

LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN K10-B



Dosen Pengampu :
Fariska Zakhralatifa Ruskanda, S.T., M.T.

Disusun Oleh :

Muhammad Fathur Rizky	19623149
Felix Chandra	19623289
Lidya Marthadilla	19623089
Lukas Raja Agripa	19623069
Kheitaro Ameniel Gugun M	16523249

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024

Pernyataan Kelompok

Kami, sebagai kelompok 10-B dalam mata kuliah IF1210 Dasar Pemrograman 2024, dengan ini menyatakan bahwa :

“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.”

Demikian pernyataan kami, sebagai bentuk dan komitmen terhadap integritas akademik.

Hormat kami,

Muhammad Fathur Rizky	19623149
Felix Chandra	19623289
Lidya Marthadilla	19623089
Lukas Raja Agripa	19623069
Kheitaro Ameniel Gugun M	16523249

Daftar Isi

Pernyataan Kelompok.....	2
Daftar Isi.....	3
Daftar Tabel.....	6
Daftar Gambar.....	7
I. Deskripsi Persoalan.....	8
II. Daftar Pembagian Kerja.....	9
2.1 Hasil Rancangan, Implementasi dan Testing Setiap Primitif.....	10
III. Desain Command untuk Setiap Primitif.....	11
3.1 Random Number Generator.....	11
3.2 Register.....	11
3.3 Login.....	11
3.4 Logout.....	12
3.5 Menu & Help.....	12
3.6 Monster.....	12
3.7 Potion.....	12
3.8 Inventory.....	12
3.9 Battle.....	13
3.10 Arena.....	13
3.11 Shop & Currency.....	14
3.12 Laboratory.....	14
3.13 Shop Management.....	14
3.14 Monster Management.....	15
3.15 Load.....	15
3.16 Save.....	15
3.17 Exit.....	16
IV. Desain Kamus Data.....	16
4.1 Random Number Generator.....	16
4.2 Register.....	16
4.3 Login.....	17
4.4 Logout.....	17
4.5 Menu & Help.....	17
4.6 Monster.....	17
4.7 Potion.....	18
4.8 Inventory.....	18
4.9 Battle.....	18
4.10 Arena.....	19
4.11 Shop & Currency.....	19
4.12 Laboratory.....	20
4.13 Shop Management.....	20
4.14 Monster Management.....	22
4.15 Load.....	22
4.15 Save.....	23

4.16 Exit.....	23
V. Desain Dekomposisi Algoritmik dan Fungsional Program.....	23
5.1 Random Number Generator.....	23
5.2 Register.....	24
5.3 Login.....	24
5.4 Logout.....	25
5.5 Menu & Help.....	25
5.6 Monster.....	26
5.7 Potion.....	26
5.8 Inventory.....	27
5.9 Battle.....	27
5.10 Arena.....	28
5.11 Shop & Currency.....	29
5.12 Laboratory.....	30
5.13 Shop Management.....	30
5.14 Monster Management.....	31
5.15 Load.....	31
5.16 Save.....	32
5.17 Exit.....	32
VI. Spesifikasi setiap Modul.....	33
6.1 Random Number Generator.....	33
6.2 Register.....	33
6.3 Login.....	35
6.4 Logout.....	36
6.5 Menu & Help.....	37
6.6 Monster.....	39
6.7 Potion.....	40
6.8 Inventory.....	41
6.9 Battle.....	42
6.10 Arena.....	52
6.11 Shop & Currency.....	54
6.12 Laboratory.....	57
6.13 Shop Management.....	59
6.14 Monster Management.....	67
6.15 Load.....	71
6.16 Save.....	71
6.17 Exit.....	73
VII. Tangkapan Layar Testing.....	74
7.1 Random Number Generator.....	74
7.2 Register.....	74
7.3 Login.....	75
7.4 Logout.....	76
7.5 Menu & Help.....	76
7.6 Monster.....	77

7.7	Potion.....	77
7.8	Inventory.....	77
7.9	Battle.....	79
7.10	Arena.....	84
7.11	Shop & Currency.....	89
7.12	Laboratory.....	91
7.13	Shop Management.....	92
7.14	Monster Management.....	93
7.15	Load.....	94
7.16	Save.....	95
7.17	Exit.....	95
VII.	Lampiran.....	96
8.1	Link Lampiran.....	96
8.2	Helper.....	96
8.2.1	Cipher.....	96
8.2.2	IsType.....	97
8.2.3	ListManipulation.....	97
8.2.4	ReadPeta.....	100
8.2.5	Readline.....	102
8.2.6	Splitter.....	103
8.3	Referensi.....	103
8.4	Asistensi.....	104

Daftar Tabel

II. Daftar Pembagian Kerja.....	9
2.1 Hasil Rancangan, Implementasi dan Testing Setiap Primitif.....	10

Daftar Gambar

V. Desain Dekomposisi Algoritmik dan Fungsional Program.....	23
5.1 Random Number Generator.....	23
5.2 Register.....	24
5.3 Login.....	24
5.4 Logout.....	25
5.5 Menu & Help.....	25
5.6 Monster.....	26
5.7 Potion.....	26
5.8 Inventory.....	27
5.9 Battle.....	27
5.10 Arena.....	28
5.11 Shop & Currency.....	29
5.12 Laboratory.....	30
5.13 Shop Management.....	30
5.14 Monster Management.....	31
5.15 Load.....	31
5.16 Save.....	32
5.17 Exit.....	32
VII. Tangkapan Layar Testing.....	73
7.2 Register.....	73
7.3 Login.....	74
7.4 Logout.....	75
7.5 Menu & Help.....	75
7.8 Inventory.....	77
7.9 Battle.....	78
7.10 Arena.....	84
7.11 Shop & Currency.....	89
7.12 Laboratory.....	91
7.13 Shop Management.....	92
7.14 Monster Management.....	93
7.15 Load.....	94
7.16 Save.....	95
7.17 Exit.....	95

I. Deskripsi Persoalan

Tugas besar Mata Kuliah IF1210 Dasar Pemrograman ini meminta untuk membuat program yang bisa membantu Purry si Platypus untuk menyelamatkan Kota Danville dari ancaman Dr. Asep Spakbor. Spesifikasi program utama yang dibuat antara lain, Fungsi Random Number Generator yang akan menghasilkan bilangan acak dari sebuah range angka untuk berbagai fitur terutama pada fungsi monster dan battle, Fungsi register digunakan agen untuk mendaftarkan akunnya terlebih dahulu terutama pengguna baru, Fungsi Login sebagai cara agent dan admin bisa melakukan login ke dalam sistem yang menginput username beserta passwordnya dari register yang telah dilakukan sebelumnya, Fungsi Logout memungkinkan pengguna yang sedang login dapat melakukan prosedur ini untuk keluar dari akun yang sedang dipakai, Fungsi Menu & Help dapat membantu pengguna dan admin tentang apa yang sebaiknya mereka lakukan terlebih dahulu, Fungsi Monster dalam program ini untuk memproses hal-hal yang berkaitan dengan monster maupun atributnya, Fungsi Potion dalam program ini dapat membantu agent untuk memiliki dan menggunakan potion yang dapat membantu mereka dalam *battle*, Fungsi Inventory merupakan sebuah sistem penyimpanan inventory yang dinamis dari invento-Shock dapat berupa Potion dan Monster dan Setiap item yang disimpan pada inventory dapat dimunculkan detailsnya, Fungsi Battle memungkinkan agent dapat memilih untuk bertarung melawan monster secara random yang bersifat *turn-based 1V1 game*, Fungsi Arena dibuat untuk meningkatkan kemampuan agen dan para monster, Fungsi Shop & Currency memungkinkan agent untuk membeli monster dan potion, Fungsi Laboratory memungkinkan agent dapat melakukan upgrade monster yang dimilikinya pada inventory agar para monster kemampuannya meningkat, Fungsi Shop Management adalah akses untuk admin untuk mengatur barang-barang yang dijual kepada para agent yang akan berguna dalam menjalankan misinya, Fungsi Monster Management akses untuk admin mengatur monster dalam database, Fungsi Load memuat prosedur yang akan memuat data yang sesuai dengan struktur data eksternal dan akan dijalankan sekali ketika pengguna memulai program, Fungsi Save berguna untuk menyimpan data ke dalam file yang sesuai struktur data eksternal dan program ini akan meminta nama folder yang akan digunakan sebagai tempat penyimpanan file, lalu yang terakhir, Fungsi Exit dibuat dalam memenuhi rasa kemanusiaan untuk para agent dan admin dengan cara memberikan kesempatan untuk melanjutkan petualangan di lain waktu dan akan diarahkan untuk melakukan save terlebih dahulu sebelum keluar dari program yang menyenangkan ini.

II. Daftar Pembagian Kerja

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
Asistensi, MoM	oleh : 19623089			
Laporan	oleh : 19623089, 1923069, 16523249			
User Interface	19623089	19623089 16523249	19623089 16523249	19623149
main.py	19623149	19623149	19623149	19623149
F00 - Random Number Generator	19623149	16523249	19623149	1923069
F01 - Register	16523249	16523249	19623149	1923069
F02 - Login	19623089	16523249	19623149	1923069
F03 - Logout	19623089	16523249	19623149	1923069
F04 - Menu & Help	16523249	16523249	19623149	1923069
F05 - Monster	19623089	16523249	19623069	1923069
F06 - Potion	1923069	16523249	19623069	1923069
F07 - Inventory	19623149	16523249	19623149	19623149
F08 - Battle	19623289	16523249	19623289	1923069
F09 - Arena	19623289	16523249	19623289	1923069
F10 - Shop & Currency	1923069	16523249	19623069	1923069
F11 - Laboratory	19623289	16523249	19623289	19623149
F12 - Shop Management	1923069	16523249	19623149	1923069
F13 - Monster Management	1923069	16523249	19623149	1923069
F14 - Load	19623149	16523249	19623149	1923069

F15 - Save	16523249	16523249	19623149	1923069
F16 - Exit	19623089	16523249	19623149	1923069
B01 - Git Best Practice	19623149	19623289	19623149	1923069
B02 - Typing	19623149	19623149	19623149	19623149
B03 - Monster Ball	-	-	-	-
B04 - JACKPOT!	-	-	-	-
B05 - Peta Kota Danville	-	-	-	-
BXX - Kreativitas	-	-	-	-

2.1 Hasil Rancangan, Implementasi dan Testing Setiap Primitif

Fitur	Desain	Implementasi	Testing
F00 - Random Number Generator	V	V	V
F01 - Register	V	V	V
F02 - Login	V	V	V
F03 - Logout	V	V	V
F04 - Menu & Help	V	V	V
F05 - Monster	V	V	V
F06 - Potion	V	V	V
F07 - Inventory	V	V	V
F08 - Battle	V	V	V
F09 - Arena	V	V	V
F10 - Shop & Currency	V	V	V
F11 - Laboratory	V	V	V
F12 - Shop Management	V	V	V
F13 - Monster Management	V	V	V

F14 - Load	V	V	V
F15 - Save	V	V	V
F16 - Exit	V	V	V
B01 - Git Best Practice	V	V	V
B02 - Typing	V	V	V
B03 - Monster Ball	-	-	-
B04 - JACKPOT!	-	-	-
B05 - Peta Kota Danville	-	-	-
BXX - Kreativitas	-	-	-

Keterangan: V: sudah selesai dikerjakan, X: dikerjakan, tapi belum selesai, -: tidak dikerjakan sama sekali.

III. Desain Command untuk Setiap Primitif

3.1 Random Number Generator

input :

a → batas bawah dari rentang bilangan acak yang dihasilkan
 b → batas atas dari rentang bilangan acak yang dihasilkan
 seed → nilai awal untuk generator bilangan acak

output :

mengembalikan sebuah bilangan bulat (int) acak yang berada dalam rentang [a, b]

3.2 Register

input :

s , user, user_id, pw → user memasukkan username dan password melalui input prompt
 user_data → data pengguna, sebuah mapping yang berisi matriks dari data sebuah file

output :

validasi username, instruksi dan konfirmasi pemilihan monster, pemberitahuan kesuksesan dari registrasi.

3.3 Login

input :

user, password → user memasukkan username dan password yang telah didaftarkan sebelumnya

`user_data` → Data pengguna dalam bentuk matriks, di mana setiap baris berisi informasi pengguna (ID, username, password, peran, dll)

output :

status yang menunjukkan apakah pengguna berhasil login sebagai admin, agen, gagal karena password salah, atau gagal karena username tidak ditemukan.

3.4 Logout

input :

Parameter `is_login` bertipe boolean yang menyatakan status login pengguna

`TRUE` → user sudah login

`FALSE` → user belum login

output :

Fungsi ini tidak mengembalikan nilai apapun (none). Tetapi mencetak pesan di konsol.

3.5 Menu & Help

input :

Satu argumen `folder_name` bertipe str, dan kemudian berbagai input dari pengguna selama eksekusi.

output :

Tidak ada nilai yang dikembalikan (None). Program mencetak berbagai pesan ke konsol berdasarkan input pengguna dan memanggil fungsi-fungsi terkait.

3.6 Monster

input :

`lvl` → user memasukkan level monster

`y` → nilai yang akan dimodifikasi berdasarkan level

`defend` → memasukkan nilai pertahanan monster

`att_enemy` → nilai serangan musuh

`power` → nilai kekuatan serangan monster

output :

Semua fungsi mengembalikan nilai bertipe integer, baik hasil perhitungan maupun nilai acak yang dihasilkan oleh `random_number_generator`

3.7 Potion

input :

`y` → nilai `defend`, `hp`, atau `power` pada monster

`x == 1/2` → potion power dan `defend` pada monster

`x == 3` → potion helath pada monster

output :

mengembalikan nilai `y` yang telah dimodifikasi dari nilai `x`

3.8 Inventory

input :

`idn` → ID dari item yang dicari

`user_data` → data inventori user dalam bentuk matriks
`a1` → data monster inventory
`a2` → data item lengkap
`data: List[str]` → informasi pengguna.
`user_data: Mapping` → data inventori dan monster.

output :

`get_item_data` → Mengembalikan Matrix berisi item yang cocok dengan ID.
`get_monster_data` → Mengembalikan Matrix berisi monster yang cocok dengan ID.
`concat_monster` → Tidak ada nilai yang dikembalikan (None).
`concat_monster_item` → Mengembalikan `List[Tuple[str, List[str]]]` yang berisi data monster dan item.
`inventory` → Tidak ada nilai yang dikembalikan (None). Fungsi ini mencetak inventori ke konsol dan memungkinkan interaksi pengguna.

3.9 Battle

input :

`data[0]` → User ID
`data[1]` → User name
`data[4]` → Jumlah O.W.C.A. Coin
`"monster_inventory.csv"` → Daftar monster yang dimiliki pengguna
`"monster.csv"` → Daftar monster yang tersedia di dalam game
`"item_inventory.csv"` → Daftar item (potions) yang dimiliki pengguna
`n` → Tahap pertarungan (0 untuk pertama kali masuk arena, 1 untuk tahap pertama, dan seterusnya)
`player` → Nama monster yang dipilih pemain
`power_p` → ATK power monster pemain
`def_p` → DEF power monster pemain
`hp_p` → HP monster pemain
`dmg_given` → Damage yang diberikan pemain pada musuh
`dmg_taken` → Damage yang diterima pemain dari musuh
`lvl_p` → Level monster pemain
`monster_p` → Indeks monster pemain dalam daftar

output :

User interface (print statements) dalam program menampilkan pilihan kepada pemain, status monster, dan hasil serangan dalam bentuk teks di terminal. Lalu Data updates (internal updates) menampilkan status monster dan inventory pemain diupdate berdasarkan aksi yang dipilih selama pertarungan.

3.10 Arena

input :

`data[0]` → User ID
`data[1]` → User name

data[4] → Jumlah O.W.C.A. Coin (OC)

user_data (Mapping) → Menyimpan data terkait monster dan item pengguna yang akan diteruskan ke fungsi battle

x → Status permainan

n → Tahap (stage) dalam arena

dmg_taken → Total damage yang diterima pemain

dmg_given → Total damage yang diberikan pemain

monster → Nama monster pemain

power → ATK power monster pemain

defend → DEF power monster pemain

hp → HP monster pemain

lvl_p → Level monster pemain

monster_p → Indeks monster pemain dalam daftar

output :

User Interface yang menampilkan hasil dari fungsi battle

3.11 Shop & Currency

input :

user_login[0] → User ID

user_login[4] → Jumlah OC (O.W.C.A. Coin) pengguna saat ini.

output :

Memberikan umpan balik kepada pengguna mengenai status transaksi (sukses atau gagal).

Data updates dengan berkurangnya jumlah OC pengguna jika pembelian berhasil, memperbarui stok item di toko, dan menambahkan item ke inventory pengguna.

3.12 Laboratory

input :

idx → ID dari monster.

data → Daftar data monster.

n → Jumlah total monster yang tersedia.

user_login → Data login pengguna, termasuk informasi OC.

user_data → Peta yang berisi data tentang monster, monster_inventory, dan lainnya.

output :

Mencetak pesan selamat datang dan daftar monster milik pengguna. Mencetak harga upgrade level monster. Mengambil input dari pengguna untuk memilih monster yang ingin di-upgrade.

Mengecek apakah monster sudah mencapai level maksimum. Jika belum, mencetak detail upgrade termasuk nama monster, level baru, dan harga upgrade. Mengambil konfirmasi dari pengguna apakah ingin melanjutkan upgrade. Jika pengguna setuju dan memiliki OC yang cukup: meng-upgrade level monster, mengurangi jumlah OC pengguna, dan mencetak pesan keberhasilan upgrade. Jika OC tidak cukup, mencetak pesan kegagalan.

3.13 Shop Management

input :

`user_data` → Mapping (kamus) yang berisi data toko (misalnya, monster dan potion).
`a1` → Matrix (daftar dari daftar string).
`a2` → Matrix (daftar dari daftar string).
`data` → Matrix (daftar dari daftar string).
`idx` → String yang merupakan ID dari item yang akan dihapus.
`is_potion` → Boolean opsional (default: False) untuk menentukan apakah item adalah potion.
`s` → String opsional (default: "ubah") untuk menentukan konteks (tambah atau ubah).
`arr` → Matrix (daftar dari daftar string).
`e` → String yang dicari.
`idx` → Integer yang merupakan indeks kolom untuk pencarian.
`txt` → String yang digunakan dalam pesan input.
`c` → String yang menunjukkan jenis data yang ingin dilihat (monster atau potion).

output :

Menampilkan data yang diminta pengguna (monster atau potion) di konsol. Menambahkan item baru (monster atau potion) ke data toko dan menampilkan tabel item yang ditambahkan di konsol. Mengubah stok dan harga item (monster atau potion) yang dipilih di data toko dan mengupdate data tersebut. Menghapus item (monster atau potion) yang dipilih dari data toko setelah verifikasi. Mengelola toko dengan melakukan aksi yang dipilih (lihat, tambah, ubah, hapus, keluar). Matrix yang memiliki kolom tambahan untuk ID. Matrix yang tidak mengandung item dengan ID yang diberikan. Boolean yang menunjukkan apakah pengguna memverifikasi penghapusan. String yang merupakan stok baru yang diinput pengguna. String yang merupakan harga baru yang diinput pengguna. Boolean yang menunjukkan apakah string ditemukan di matrix. String yang merupakan ID yang valid diinput pengguna. Menampilkan data yang diminta (monster atau potion) di konsol.

3.14 Monster Management

input :

`user_data` → beberapa input dari pengguna untuk detail monster baru (Type, ATK Power, DEF Power, HP), dan konfirmasi penambahan monster ke database.

output :

mengelola data monster dalam sebuah toko, termasuk melihat, menambah, dan mengelola data monster. Menampilkan daftar monster atau memanggil fungsi `add_monster` berdasarkan pilihan aksi.

3.15 Load

input :

menerima input dari command line argument. Input tersebut adalah nama folder yang akan diverifikasi.

`python main.py <nama_folder>`

output :

Jika folder ditemukan, fungsi `menu(folder_name)` dipanggil atau menampilkan semua dari `file.py` sesuai fungsi yang ada.

3.16 Save

input :

`folder_name` → Nama folder yang akan digunakan.

`c` → Parameter integer opsional yang menentukan jenis pesan yang akan ditampilkan. Nilai default adalah 2.

`file_name` → Nama file yang akan ditulis.

`folder_name` → Nama folder dimana file akan disimpan.

`user_data` → Sebuah kamus yang memetakan nama file ke data yang akan disimpan.

`user_data` → Sebuah kamus yang memetakan nama file ke data yang akan disimpan.

output :

Fungsi ini mencetak pesan ke konsol dengan efek penundaan untuk setiap karakter dan menambahkan titik tiga di akhir pesan. Pesan yang ditampilkan bergantung pada nilai `c`. Fungsi ini menulis data dari `user_data` ke dalam file yang ditentukan di dalam folder yang ditentukan. Setiap baris data digabung menjadi satu string dengan menggunakan fungsi `join` dan diakhiri dengan new line. Fungsi ini meminta pengguna untuk memasukkan nama folder. Fungsi ini kemudian memeriksa keberadaan folder `./data` dan `./data/<folder_name>`, dan membuat folder jika tidak ada. Tergantung pada kondisi folder, fungsi ini akan memanggil `print` dengan parameter yang sesuai. Setelah itu, fungsi ini akan memanggil `write_lines` untuk setiap file dalam `user_data` untuk menulis data ke dalam file di folder yang ditentukan.

3.17 Exit

input :

`data` → Dictionary yang berisi data yang ingin disimpan.

output :

Tidak ada nilai yang dikembalikan (outputnya adalah efek samping dari operasi pada data yang disimpan). Program akan melakukan proses penyimpanan berdasarkan pilihan pengguna dan kondisi dari data. Jika pengguna memilih 'y', maka `save(data)` akan dipanggil untuk menyimpan data. Jika pengguna memilih 'n' atau input tidak valid, maka tidak ada aksi yang dilakukan dan program akan selesai.

IV. Desain Kamus Data

4.1 Random Number Generator

KAMUS LOKAL function `random_number_generator`

- `a`, `b`, `seed`, `m`, `random_number`: integer

4.2 Register

KAMUS LOKAL function `is_valid`

- `i`: char

- `s`: string

KAMUS LOKAL function `verification`

```
- data: List of string
{ 0: username is not valid
  1: username is not available
  2: user completely registered }
```

KAMUS LOKAL procedure pick_monster

- data: Matrix
- n: integer
- monster_id: string

KAMUS LOKAL function add_user

- data: Matrix
- last_id: integer

KAMUS LOKAL procedure regist

- user: string
- pw: string
- verif: integer

4.3 Login

KAMUS LOKAL function verification

- row: List of string

KAMUS LOKAL function login

- user, password: string
- result: integer

4.4 Logout

KAMUS LOKAL procedure logout

- is_login: boolean

4.5 Menu & Help

KAMUS LOKAL procedure menu

- is_login: boolean
- is_exit: boolean
- is_admin: boolean
- user_data: Mapping
- user_login: List of string

4.6 Monster

KAMUS LOKAL function stats_Monster

- x, y : integer

KAMUS LOKAL function def_monster_a

- x, y: integer

KAMUS LOKAL attack_monster_a

- x, y: integer

KAMUS LOKAL def_monster_b

- x, y: integer

KAMUS LOKAL attack_monster_b

- x, y: integer

KAMUS LOKAL def_monster_c

- x, y: integer

KAMUS LOKAL attack_monster_c

- x, y: integer

4.7 Potion

KAMUS LOKAL function potion

- x, y : int

4.8 Inventory

KAMUS LOKAL function get_item_data

- arr: Matrix

KAMUS LOKAL function get_monster_data

- arr: Matrix

KAMUS LOKAL procedure concat_monster

- a1: Matrix

- a2: Matrix

KAMUS LOKAL function concat_monster_item

- arr: List[Tuple[string, List[string]]]

4.9 Battle

KAMUS LOKAL procedure delay_print

- i : integer
- t : real
- s : string

KAMUS LOKAL procedure battle

- pilih, match, ada_potion, potions : boolean
- i, j, n, indeks, turn, oc, monster_p, monster, jlh_monster, power_m, def_m, hp_m, hp_ori, hp_p, def_p, power_p, lvl_p, att, serangan, dmg_taken, dmg_given, pilihan_p, pernah1, pernah2, pernah3 : integer
- list_lvl, list_power, list_def, list_hp, list_potion, letak_potion : list of integer
- list_nama : list of string
- nama_m, player, pilihan, pilihan_p : string

4.10 Arena

KAMUS LOKAL procedure delay_print

- i : integer
- t : real
- s : string

KAMUS LOKAL procedure arena

- reward : integer

4.11 Shop & Currency

KAMUS LOKAL function is_added

- i: List of string

KAMUS LOKAL function index_added_potion

- i: List of string

KAMUS LOKAL procedure beli

- breaked: boolean
- c: string

KAMUS LOKAL function get_jumlah()

- cc: string
- jumlah: string

KAMUS LOKAL procedure shop_and_currency

- breaked: boolean
- masukan: string

4.12 Laboratory

KAMUS LOKAL function get_name_by_id

- i: List of string

KAMUS LOKAL function get_id

- c: string

KAMUS LOKAL function verif()

- c: string

KAMUS LOKAL procedure laboratory

- monster_list: Matrix
- harga: integer
- idx: integer
- monster_name: string
- lvl: integer

4.13 Shop Management

KAMUS LOKAL procedure lihat

- breaked: boolean
- c: string

KAMUS LOKAL procedure tambah

- breaked: boolean
- c: string
- contoh: Matrix
- arr: Matrix
- monster: Matrix
- idx: string
- stock: string
- price: string
- idn: integer

KAMUS LOKAL procedure ubah

- breaked : boolean
- c, idx, ststock, price: string

- i : integer
- potion : List of string

KAMUS LOKAL procedure hapus

- breaked: boolean
- monster_shop, monster, monster_list: Matrix
- idx: string

KAMUS LOKAL procedure shop_management

- breaked : boolean
- act : string

KAMUS LOKAL function concat

- a1, a2: Matrix
- row, col_a1, col_a2, col, idx:, i integer
- new_arr: List of string

KAMUS LOKAL function write_without

- user_data, arr : Matrix
- idx : integer
- is_potion : Boolean
- arr : array

KAMUS LOKAL function verification_hapus

- idx, is_hapus: string
- user_data: Matrix

KAMUS LOKAL function get_stock

- s, stock: string

KAMUS LOKAL function get_price

- s, price: string

KAMUS LOKAL function is_in

- arr : Matrix
- e : string
- idx : integer

KAMUS LOKAL function get_idx

- idx, txt: string
- data: Matrix

KAMUS LOKAL function get_price

- s, price: string

KAMUS LOKAL procedure show

- monster_to_print : Matrix
- user_data: Mapping
- c : string

4.14 Monster Management

KAMUS LOKAL function get_type

- name: string

KAMUS LOKAL function is_added

- monster_data: Matrix
- row: List of string

KAMUS LOKAL function get_atk_power()

- atk_power: string

KAMUS LOKAL function get_def_power()

- def_power: string

KAMUS LOKAL function get_hp()

- hp: string

KAMUS LOKAL function get_last_id

- data: Matrix
- last_id: integer

KAMUS LOKAL procedure add_monster

- idn: integer
- name, atk_power, def_power, hp, choice: string

KAMUS LOKAL procedure monster_management

- choice: string

4.15 Load

KAMUS LOKAL procedure load()

- parser: argparse.ArgumentParser

- args: argparse.ArgumentParser
- folder_name: string

4.15 Save

KAMUS LOKAL procedure pprint

- i: char

KAMUS LOKAL procedure delayed_print

- i: string

KAMUS LOKAL procedure write_lines

- f: file
- data_to_write: string
- row: List of string

KAMUS LOKAL procedure save

- folder_name: string
- file_name: string

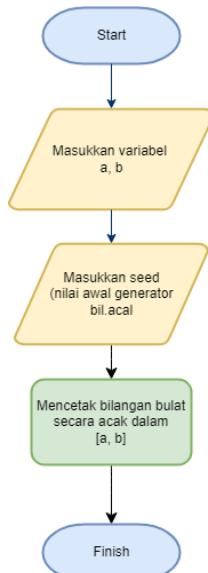
4.16 Exit

KAMUS LOKAL procedure leave

- choice: string

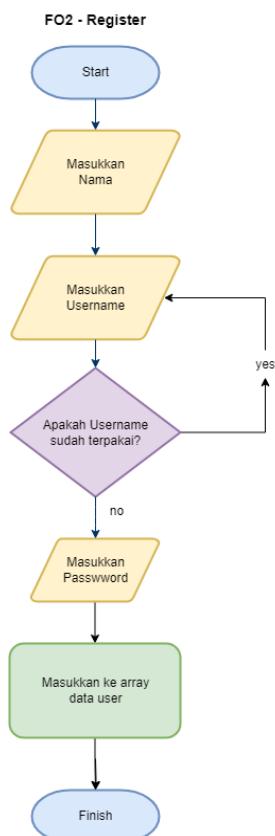
V. Desain Dekomposisi Algoritmik dan Fungsional Program

5.1 Random Number Generator



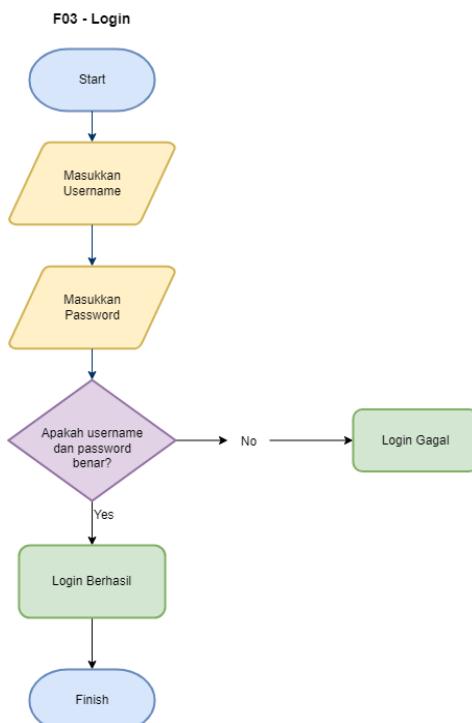
Gambar 5.1 : F01 - Random Number Generator

5.2 Register



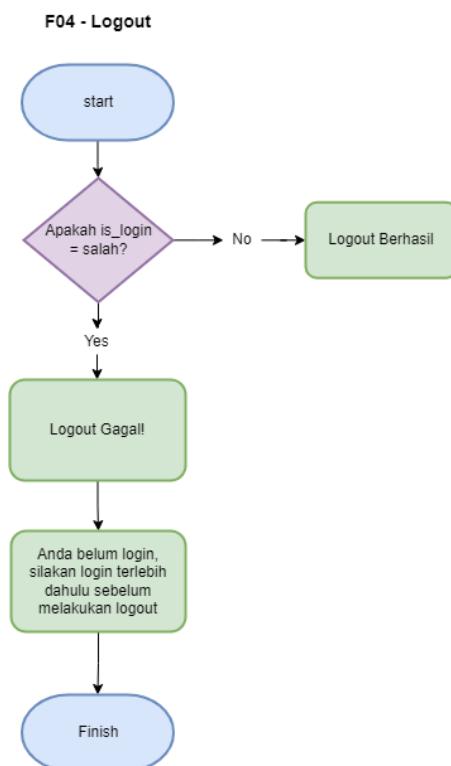
Gambar 5.2 : F02 - Register

5.3 Login



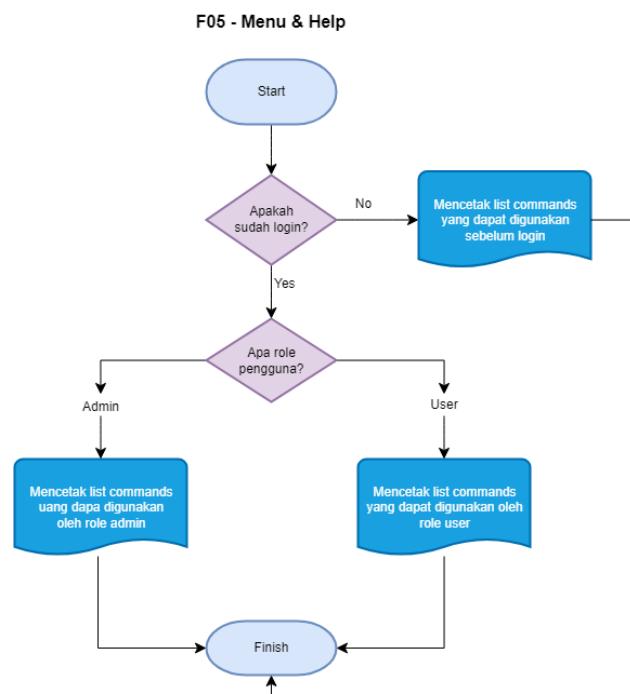
Gambar 5.3 : F03 - Login

5.4 Logout



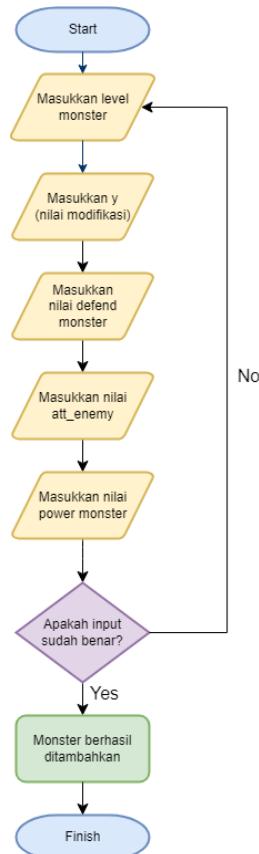
Gambar 5.4 : F04 - Logout

5.5 Menu & Help



Gambar 5.5 : F05 - Menu & Help

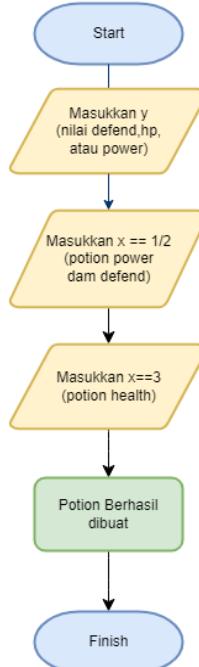
5.6 Monster



Gambar 5.6 : F06 - Monster

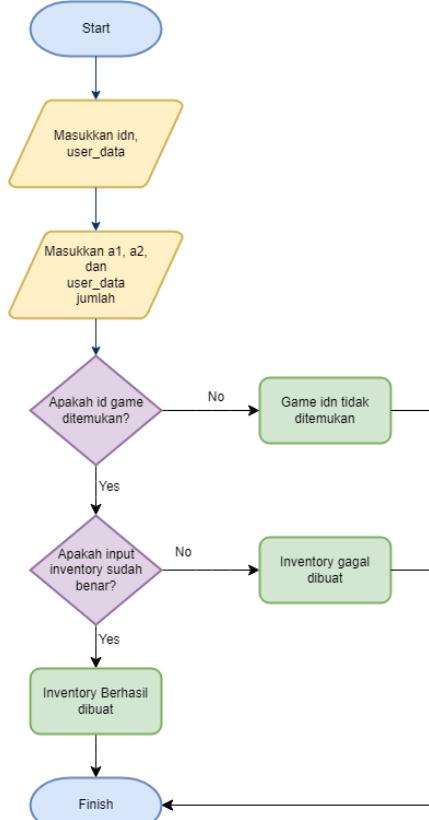
5.7 Potion

F07 - Potion



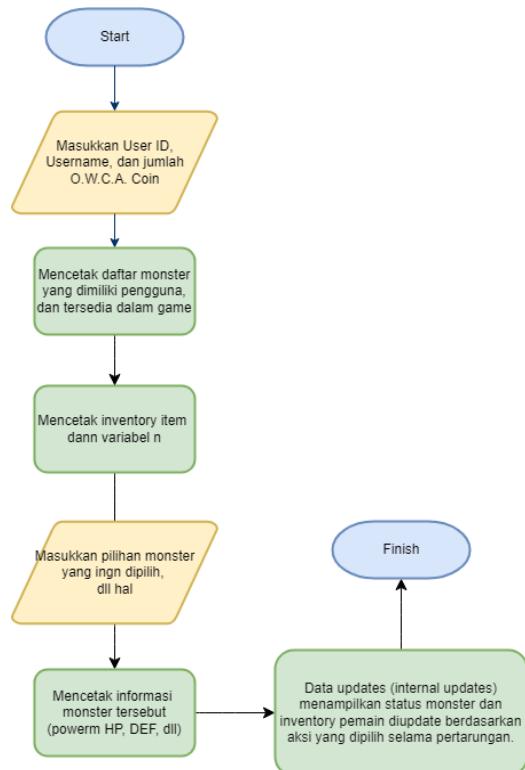
Gambar 5.7 : F07 - Potion

5.8 Inventory



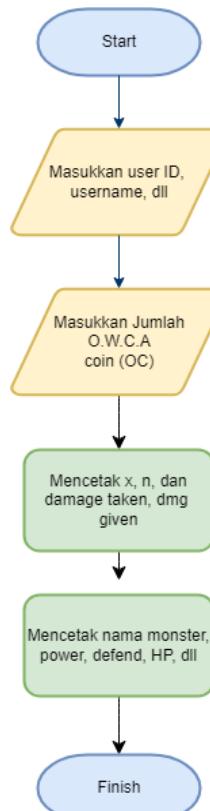
Gambar 5.8 : F08 - Inventory

5.9 Battle



Gambar 5.9 : F09 - Battle

5.10 Arena



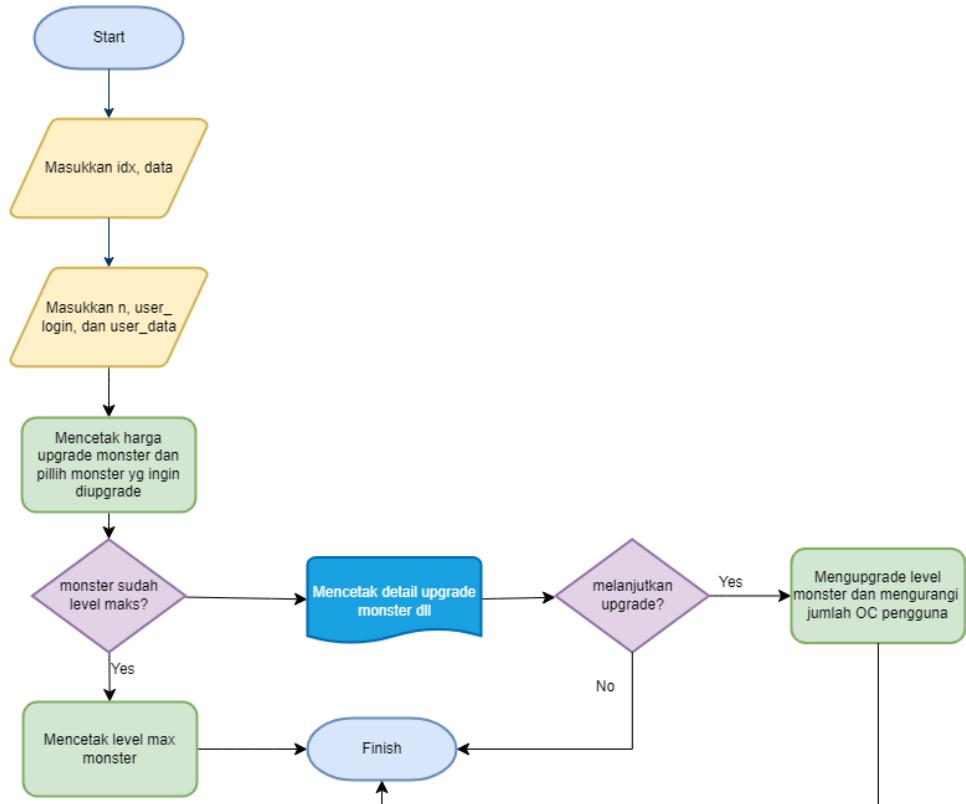
Gambar 5.10 : F10 - Arena

5.11 Shop & Currency



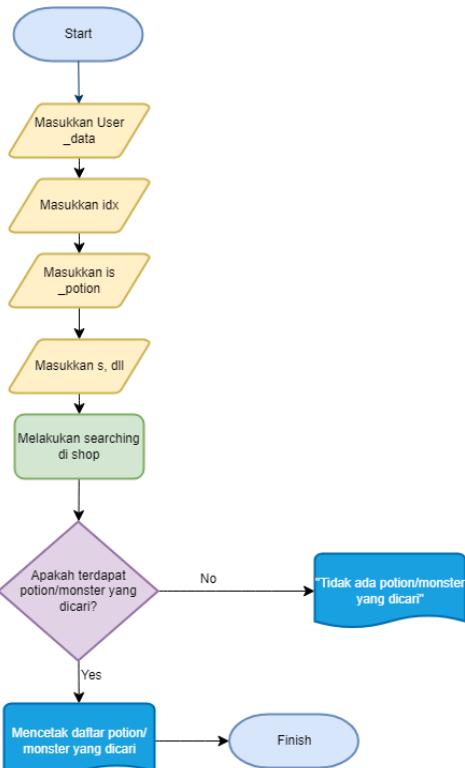
Gambar 5.11 : F11 - Shop & Currency

5.12 Laboratory



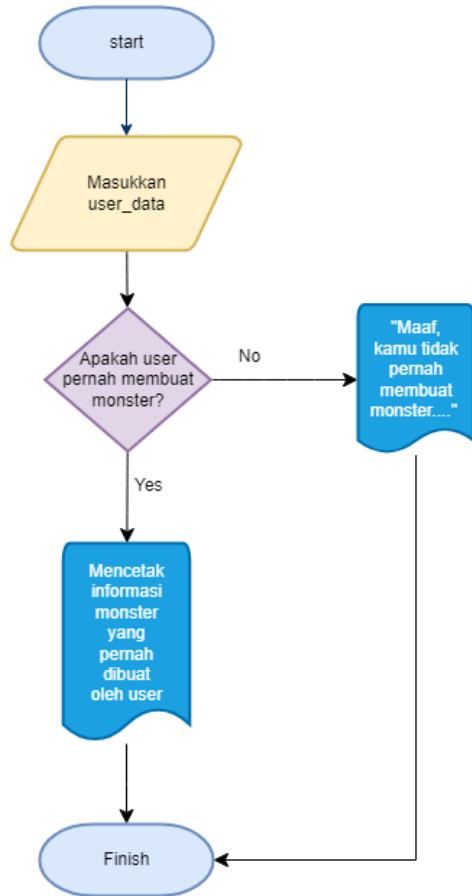
Gambar 5.12 : F12 - Laboratory

5.13 Shop Management



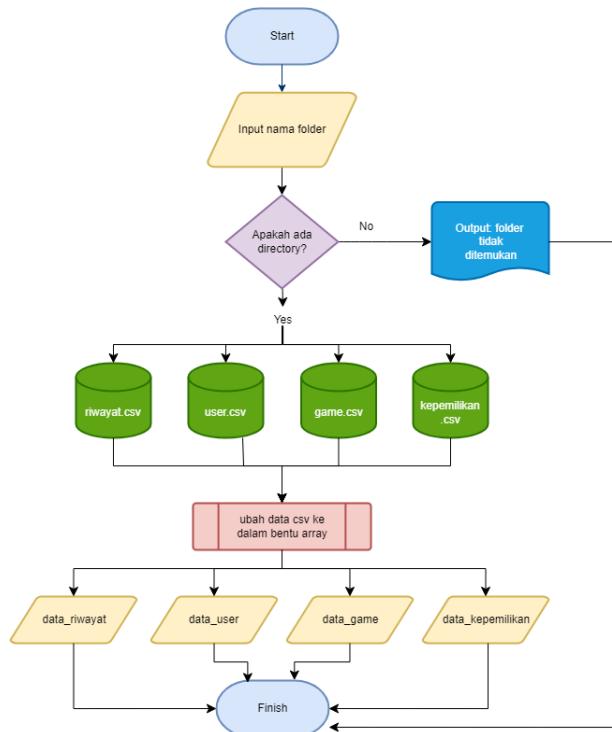
Gambar 5.13 : F13 - Shop Management

5.14 Monster Management



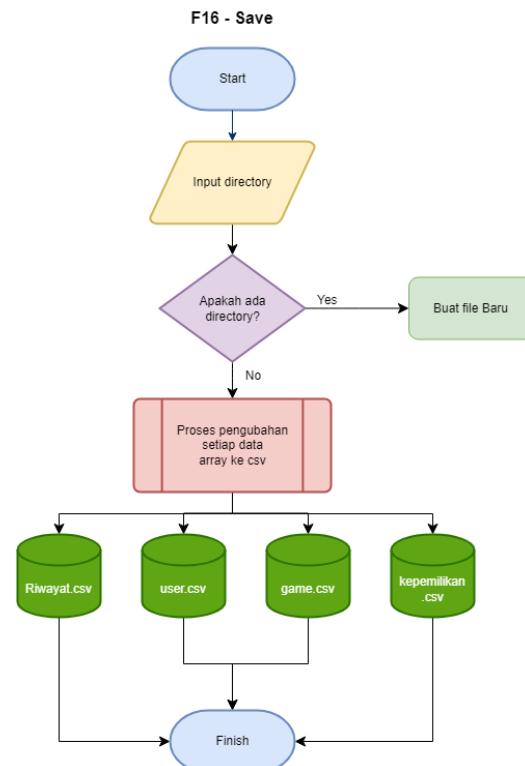
Gambar 5.14 : F14 - Monster Management

5.15 Load



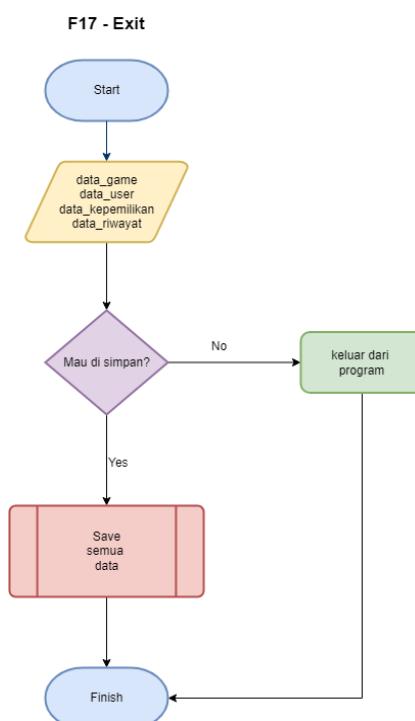
Gambar 5.15 : F15 - Load

5.16 Save



Gambar 5.16 : F16 - Save

5.17 Exit



Gambar 5.17 : F17 - Exit

VI. Spesifikasi setiap Modul

6.1 Random Number Generator

KAMUS LOKAL

- a, b, seed, m, random_number: integer

ALGORITMA

```
import time
function random_number_generator (a: integer, b: integer, seed: integer) = integer(time.time()) -> integer
    m <- 2^31 - 1
    seed <- (a * seed + b) mod m
    random_number <- a + seed mod (b - a + 1)

    return random_number
```

6.2 Register

function is_valid (s: string) -> boolean

KAMUS LOKAL

```
- i: char
- s: string
{ [A-Z], [a-z], [_.], [-] are allowed }
for each i in s
    if not (ord('A') <= ord(i) <= ord('Z') or ord('a') <= ord(i)
<= ord('z') or ord('0') <= ord(i) <= ord('9') or i = '_' or i = '-')
then
    return false
end if
end for
return true
```

function verification (user: string, user_data: Matrix) -> boolean

KAMUS LOKAL

- data: List of string

{ 0: username is not valid
1: username is not available
2: user completely registered }

```
if not is_valid(user) then
    return 0
```

```

end if

for each data in user_data do
    if data[1] = user then
        return 1
    end if
end for

return 2

procedure pick_monster (user_id: integer, user: string, user_data:
Mapping)
KAMUS LOKAL
- data: Matrix
- n: integer
- monster_id: string

data <- user_data[ "monster.csv" ]
n <- length(data)
    output('\nSilakan pilih salah satu monster sebagai monster awalmu.')
i traversal [1..n-1]
    output("%s. %s" % (data[i][0], data[i][1]))
end traversal

while true do
    input(monster_id, "\nID monster pilihanmu: ")
    i traversal [1..n-1]
        if data[i][0] = monster_id then
            append(user_data[ "monster_inventory.csv" ],
[str(user_id), monster_id, "1"])
            output('\nSelamat datang Agent %. Mari kita
mengalahkan Dr. Asep Spakbor dengan %!' %
(user, data[i][1]))
        return
    end if
    output('ID monster tidak ditemukan, silakan ulangi!')
end while
endfunction

```

```

function add_user (user: string, pw: string, user_data: Mapping) ->
void

KAMUS LOKAL
- data: Matrix
- last_id: integer

data <- user_data["user.csv"]
last_id <- int(data[length(data)][0])
append(user_data["user.csv"], [str(last_id+1), user, pw,
"Agent", "0"])

pick_monster(last_id+1, user, user_data)
endfunction

```

```

procedure regist (user_data: Mapping)
KAMUS LOKAL
- user: string
- pw: string
- verif: integer

(user, "Masukan username: ")
(pw, "Masukan password: ")
verif <- verification(user, user_data["user.csv"])
if verif = 0 then
    output('Username hanya boleh berisi alfabet, angka,
underscore, dan strip!')
else if verif = 1 then
    output('Username %s sudah terpakai, silahkan gunakan
username lain!' % (user))
else if verif = 2 then
    add_user(user, pw, user_data)
end if

```

6.3 Login

```

function verification (user: string, password: string, user_data:
Matrix) -> (List of string, integer)

```

KAMUS LOKAL

- row: List of string

for each row in user_data do

```

    if user = row[1] and password = row[2] then
        output("You are logged in as", end="")
        if "Admin" = row[3] then
            output('Admin!')
            return (row, 0)
        else
            output('Agent!')
            return(row, 1)
        end if
    else if user = row[1] and password != row[2] then
        output('Wrong password, try again!')
        return([], 1)
    end if
end for

output("Username doesn't exist!")
return([], 3)
end function

```

```

function login (user_data: Matrix) -> integer
    KAMUS LOKAL
    - user, password: string
    - result: integer

    input(user, 'Masukkan username Anda: ')
    input(password, 'Masukkan password Anda: ')

    result <- verification(user, password, user_data)
    return result

```

6.4 Logout

```

procedure logout (is_login: boolean)
    KAMUS LOKAL
    - is_login: boolean

    if not is_login then
        output("Logout gagal!")
        output("Anda belum login, silakan login terlebih dahulu
sebelum melakukan logout")
    end if

```

6.5 Menu & Help

```
procedure menu (folder_name: string)
KAMUS LOKAL
- is_login: boolean
- is_exit: boolean
- is_admin: boolean
- user_data: Mapping
- user_login: List of string

user_data <- {}
user_login <- [ ]
read_all(user_data, folder_name)
output('\nLoading...')
output("Ketik HELP untuk melihat command yang tersedia!")
is_exit <- false

while not is_exit do
    input(choice, "\n>>> ")
    if choice = "HELP" then
        if is_admin then
            help_message_admin()
        else if is_login then
            help_message_agent()
        else
            help_message_before_login()
        end if
    else if choice = "REGISTER" then
        if is_login then
            output("Register gagal!")
            output("Anda telah login dengan username %s,
silahkan lakukan \"LOGOUT\" sebelum melakukan register." %
(user_login[1]))
        else
            regist(user_data)
        end if
    else if choice = "LOGIN" then
        if is_login then
            output('You have been logged in!')
        else
```

```

        result <- login(user_data[ "user.csv" ])
        if result[1] <= 1 then
            is_login <- true
            user_login <- result[0]
        end if
        if result[1] = 0 then
            is_admin <- true
        end if
    end if
    else if choice = "INVENTORY" and is_login and not is_admin
then
    inventory(user_login, user_data)
else if choice = "MONSTER" and is_admin then
    monster_management(user_data)
else if choice = "SHOP" then
    if is_admin then
        shop_management(user_data)
    else if is_login then
        shop_and_currency(user_data, user_login)
    end if
else if choice = "LABORATORY" and is_login and not is_admin
then
    laboratory(user_login, user_data)
else if choice = "BATTLE" and is_login and not is_admin then
    battle(user_login, user_data, 0, "null", 0, 0, 0, 0, 0,
0)
    else if choice = "LOGOUT" then
        logout(is_login)
        if is_login then
            is_login <- false
            is_admin <- false
        end if
    else if choice = "EXIT" then
        is_exit <- true
        leave(user_login, user_data, 1)
    else if choice = "ARENA" and is_login and not is_admin then
        arena(user_login, user_data, 1, 1, 0, 0, "null", 0, 0,
0, 0)
    else if choice = "SAVE" then
        save(user_data)
end if

```

```
end while
```

6.6 Monster

```
from src.RandomNumberGenerator import random_number_generator
import time
function stats_Monster (lvl: integer, y; integer) -> integer
    KAMUS LOKAL
        - x, y : integer

        x <- y * (1+(lvl-1)*1/10
        return x

function def_monster_a (defend: integer, att_enemy: integer) ->
integer
    KAMUS LOKAL
        - x,y: integer

        x <- random_number_generator(1,defend+1)
        x <- x-1
        y <- att_enemy*(100-x)/100
        return y

function attack_monster_a (power: integer) -> integer
    KAMUS LOKAL
        - x,y : integer

        x <- power*70/100
        y <- power*130/100
        return random_number_generator(x,y)

function def_monster_b (defend: integer, att_enemy: integer) ->
integer
    KAMUS LOKAL
        - x,y: integer

        x <- random_number_generator(1 , defend+1 , time.time() +
def_monster_a (defend,att_enemy)+ 1)
        x <- x-1
        y <- att_enemy*(100-x)/100
        return y
```

```

function attack_monster_b (power: integer) -> integer
    KAMUS LOKAL
    - x,y : integer

        x <- power*70/100
        y <- power*130/100
        return random_number_generator( x ,y, time.time() +
attack_monster_a(power))

function def_monster_c (defend: integer, att_enemy: integer) ->
integer
    KAMUS LOKAL
    - x,y: integer

        x <- random_number_generator(1 , defend+1 , time.time() +
def_monster_b (defend,att_enemy)+ 1)
        x <- x-1
        y <- att_enemy*(100-x)/100
        return y

function attack_monster_c (power: integer) -> integer
    KAMUS LOKAL
    - x,y : integer

        x <- power*70/100
        y <- power*130/100
        return random_number_generator( x ,y, time.time() +
attack_monster_b(power))

```

6.7 Potion

```

function potion (x: integer, y: integer) -> integer
    KAMUS LOKAL
    - x,y : integer

    if (x==1) or (x==2) then
        y <- y*105/100
    else
        y <- y/4
    return y

```

6.8 Inventory

ALGORITMA

```
from .helper.IsType import is_number
from typing import List, Tuple, Dict
from src.Monster import stats_Monster
function get_item_data (idn: string, user_data: Matrix) -> Matrix
```

KAMUS LOKAL

- arr: Matrix

```
arr <- []
for each i in user_data do
    if i[0] = idn then
        arr.append(i)
    end if
end for
return arr
```

```
function get_monster_data (idn: string, user_data: Matrix) -> Matrix
```

KAMUS LOKAL

- arr: Matrix

```
arr <- []
for each i in user_data do
    if i[0] = idn then
        arr.append(i)
    end if
end for
return arr
```

```
procedure concat_monster (a1: Matrix, a2: Matrix)
```

KAMUS LOKAL

- a1: Matrix
- a2: Matrix

```
i traversal [1..length(a1)]
for each j in a2 do
    if a1[i][1] = j[0] then
        for k from 1 to 4 do
            a1[i].append(j[k])
        end for
    end if
```

```

    end for
end for

function concat_monster_item (a1: Matrix, a2: Matrix) ->
List[Tuple[string, List[string]]]

KAMUS LOKAL
- arr: List[Tuple[string, List[string]]]

arr <- []
for each i in a1 do
    arr.append(('monster', i))
end for
for each i in a2 do
    arr.append(('potion', i))
end for
return arr

```

6.9 Battle

```

procedure delay_print (s : string, t <- 0.015):
KAMUS LOKAL
- i : integer
- t : real
- s : string

i traversal[0,len(s)]
    output(i,end <- "")
    sys.stdout.flush()
    time.sleep(t)
output("")

procedure battle (data: Matrix, user_data:Mapping, n: integer,
player: string, power_p: integer, def_p: integer, hp_p : integer,
dmg_given: integer, dmg_taken : integer, lvl_p : integer, monster_p
: integer)
KAMUS LOKAL
-pilih, match, ada_potion, potions : boolean
-i,j,n,indeks,turn,oc,monster_p,monster,jlh_monster,power_m,de
f_m,hp_m,hp_ori,hp_p,def_p,power_p,lvl_p,att,serangan,dmg_taken,dmg_
given,pilihan_p,pernah1,pernah2,pernah3 : integer

```

```

-list_lvl, list_power, list_def, list_hp, list_potion ,
letak_potion : list of integer
-list_nama : list of string
-nama_m, player, pilihan, pilihan_p : string

if (n=1) then
    delay_print("Selamat datang di Arena\n")
    delay_print(" ===== MONSTER LIST ===== \n")
    j <- integer(1)
    list_lvl : array[0..0] of integer
    list_power : array [0..0] of integer
    list_def : array [0..0] of integer
    list_hp : array [0..0] of integer
    list_nama : array [0..0] of string
    i traversal [0..len(user_data["monster_inventory.csv"])]
        if (user_data["monster_inventory.csv"][i][0]=data[0])
then

indeks<-integer(user_data["monster_inventory.csv"][i][1])+1
    delay_print(f"{j}.

{user_data['monster.csv'][indeks][1]}, level:
{user_data['monster_inventory.csv'][i][2]}")
    j <- j+ 1

list_lvl.append(integer(user_data["monster_inventory.csv"][i][2]))

list_power.append(integer(user_data["monster.csv"][indeks][2]))

list_def.append(integer(user_data["monster.csv"][indeks][3]))

list_hp.append(integer(user_data["monster.csv"][indeks][4]))

list_nama.append(user_data['monster.csv'][indeks][1])

pilih <- True
while (pilih):
    delay_print("Pilih monster untuk bertarung: ")
    input (monster_p)
    while not is_number(monster_p):
        delay_print("Masukkan harus bertipe Integer, coba
lagi! ")

```

```

        input (monster_p)
monster_p <- integer(monster_p)
if (monster_p>j-1) then
    delay_print("Pilihan nomor tidak tersedia!\n")
else
    pilih <- False
    ascii_battle(monster_p)
    lvl_p <- list_lvl[monster_p]
    power_p <-
stats_Monster(lvl_p,list_power[monster_p])
    def_p <- stats_Monster(lvl_p,list_def[monster_p])
    hp_ori <- stats_Monster(lvl_p,list_hp[monster_p])
    hp_p <- hp_ori
    player <- list_nama[monster_p]
    delay_print(f"RAWRRR, AGENT {data[1]} mengeluarkan
monster {player}\n")
    delay_print (f"Name : {player}")
    delay_print (f"ATK Power : {power_p}")
    delay_print (f"DEF Power : {def_p}")
    delay_print (f"HP : {hp_p}")
    delay_print (f"Level : {lvl_p}\n")

if (n!=0) then
    output(f" ===== STAGE {n} ===== \n")
jh_monster <- len(user_data["monster.csv"])
monster <-
random_number_generator(1,jlh_monster-1,integer(time.time())+random_
number_generator(1,100))
ascii_battle(monster)
output("")
delay_print(f"RAWRRR, Monster
{user_data['monster.csv'][monster][1]} telah muncul !!! \n")
lvl_m <-
random_number_generator(1,4,integer(time.time())+random_number_generator(1,100))
nama_m <- user_data['monster.csv'][monster][1]
power_m <-
stats_Monster(lvl_m,integer(user_data['monster.csv'][monster][2]))
def_m <-
stats_Monster(lvl_m,integer(user_data['monster.csv'][monster][3]))

```

```

hp_m <-
stats_Monster(lvl_m, integer(user_data[ 'monster.csv' ][monster][4]))
delay_print (f"Name      : {nama_m}")
delay_print (f"ATK Power : {power_m}")
delay_print (f"DEF Power : {def_m}")
delay_print (f"HP       : {hp_m}")
delay_print (f"Level     : {lvl_m}\n")

if (n=0) then
    delay_print(" ===== MONSTER LIST ===== \n")
    j <- integer(1)
    list_lvl : array [0..0] of integer
    list_power : array [0..0] of integer
    list_def : array [0..0] of integer
    list_hp : array [0..0] of integer
    list_nama <- : array [0..0] of string
    i traversal[0..len(user_data[ "monster_inventory.csv" ])]
        if(user_data[ "monster_inventory.csv" ][i][0]=data[0])
then
    indeks <-
    integer(user_data[ "monster_inventory.csv" ][i][1])+1
        delay_print(f"{j} .
{user_data[ 'monster.csv' ][indeks][1]}, level:
{user_data[ 'monster_inventory.csv' ][i][2]})"
        j <- j+ 1

list_lvl.append(integer(user_data[ "monster_inventory.csv" ][i][2]))

list_power.append(integer(user_data[ "monster.csv" ][indeks][2]))

list_def.append(integer(user_data[ "monster.csv" ][indeks][3]))

list_hp.append(integer(user_data[ "monster.csv" ][indeks][4]))

list_nama.append(user_data[ 'monster.csv' ][indeks][1])

pilih <- True
while (pilih):
    delay_print("Pilih monster untuk bertarung: ")
    input (monster_p)
    while not is_number(monster_p):

```

```

        delay_print("Masukkan harus bertipe Integer, coba
lagi!")
        input (monster_p)
monster_p <- integer(monster_p)
if (monster_p>j-1) then
    delay_print("Pilihan nomor tidak tersedia!\n")
else
    pilih <- False
    ascii_battle(monster_p)
    lvl_p <- list_lvl[monster_p]
    power_p <-
stats_Monster(lvl_p,list_power[monster_p])
    def_p <- stats_Monster(lvl_p,list_def[monster_p])
    hp_ori <- stats_Monster(lvl_p,list_hp[monster_p])
    hp_p <- hp_ori
    player <- list_nama[monster_p]
    delay_print(f"RAWRRR, AGENT {data[1]} mengeluarkan
monster {player}\n")
    delay_print (f"Name : {player}")
    delay_print (f"ATK Power : {power_p}")
    delay_print (f"DEF Power : {def_p}")
    delay_print (f"HP : {hp_p}")
    delay_print (f"Level : {lvl_p}\n")
else
    hp_ori <- hp_p
match <- True
turn <- 1
pernah1 <- 0
pernah2 <- 0
pernah3 <- 0
while(match):
    if (turn%2=0) then
        ascii_battle(monster)
        delay_print(f" ====== TURN {turn} ({nama_m})
===== \n")
        att <- attack_monster_c(power_m)
        serangan <- def_monster_c(def_p,att)
        hp_p <- hp_p -serangan
        dmg_taken <- dmg_taken + serangan
        if (hp_p<0):
            dmg_taken <- dmg_taken + hp_p

```

```

        hp_p <- 0
        delay_print (f"SCHWINKKK, {nama_m} menyerang
{player}\n")
        delay_print ("DETAILED SERANGAN:")
        delay_print (f"{nama_m} mencoba untuk menyerang {player}
dengan power sebesar {att}
({integer(real((att-power_m)/power_m*100))}%}")
        delay_print (f"{player} menahan serangan dari {nama_m}
sebesar {att-serangan} ({integer(real((att-serangan)/att*100))}%}")
        delay_print (f"sehingga damage yang diterima oleh
{player} sebesar {serangan}\n")

        delay_print (f"Name : {player}")
        delay_print (f"ATK Power : {power_p}")
        delay_print (f"DEF Power : {def_p}")
        delay_print (f"HP : {hp_p}")
        delay_print (f"Level : {lvl_p}\n")

turn <- turn +1
if (hp_p=0) then
    if (n>=1) then

arena(data,user_data,0,n,dmg_taken,dmg_given,player,power_p,def_p,hp
_ori,lvl_p,monster_p)
else
    delay_print (f"Yahhh, Anda dikalahkan monster
{nama_m}. Jangan menyerah, coba lagi !!!")
    match <- False
    time.sleep(0.2)
else
    if(n!=0)or(n=0 and turn!=1) then
        ascii_battle(monster_p)
        output(f" ===== TURN {turn} ({player})
===== ")
        delay_print("1. Attack")
        delay_print("2. Use Potion")
        delay_print("3. Quit")
        input (pilihan)
        time.sleep(1)
        output("")
if(pilihan=="3") then

```

```

        if (n>=1) then

arena(data,user_data,2,n,dmg_taken,dmg_given,player,power_p,def_p,hp
_ori,lvl_p,monster_p)
        else
            delay_print("Anda berhasil kabur dari BATTLE!")
            match <- False
        else if(pilihan = "1") then
            delay_print(f"SCHWINKKK, {player} menyerang {nama_m}
\n")
            att <- attack_monster_b(power_p)
            serangan <- def_monster_b(def_m,att)
            hp_m <- hp_m - serangan
            dmg_given <- dmg_given + serangan
            delay_print ("DETAILED SERANGAN:")
            delay_print (f"{player} mencoba untuk menyerang
{nama_m} dengan power sebesar {att}
({integer(real((att-power_p)/power_p*100))}%}")
            delay_print (f"{nama_m} menahan serangan dari
{player} sebesar {att-serangan}
({integer(real((att-serangan)/att*100))}%}")
            delay_print (f"sehingga damage yang diterima oleh
{nama_m} sebesar {serangan}\n")
            if (hp_m<0) then
                dmg_given <- dmg_given + hp_m
                hp_m <- 0
                delay_print (f"Name : {nama_m}")
                delay_print (f"ATK Power : {power_m}")
                delay_print (f"DEF Power : {def_m}")
                delay_print (f"HP : {hp_m}")
                delay_print (f"Level : {lvl_m}\n")
                if (hp_m=0) then
                    delay_print (f"Selamat, Anda berhasil
mengalahkan monster {nama_m} !!! \n")
                    if (n>=1) then
                        n <- n + 1

arena(data,user_data,1,n,dmg_taken,dmg_given,player,power_p,def_p,hp
_ori,lvl_p,monster_p)
        else
            oc <- random_number_generator(5,30)

```

```

        delay_print (f"Total OC yang diperoleh:
{oc} ")

        data[4] <- str(integer(data[4]))+oc
        match <- False
        turn <- turn + 1
    else if (pilihan = "2") then
        ada_potion <- False
        list_potion : array [0..3] of integer
        letak_potion : array [0..3] of integer
        i traversal[1,len(user_data["item_inventory.csv"])]
        if
            (user_data["item_inventory.csv"][i][0]=data[0]) then
                if
                    (user_data["item_inventory.csv"][i][1]="strength") then
                        list_potion[1] <-
                            integer(user_data["item_inventory.csv"][i][2])
                        letak_potion[1] <- i
                    else if
                        (user_data["item_inventory.csv"][i][1]="resilience") then
                            list_potion[2] <-
                                integer(user_data["item_inventory.csv"][i][2])
                            letak_potion[2] <- i
                        else if
                            (user_data["item_inventory.csv"][i][1]="healing") then
                                list_potion[3] <-
                                    integer(user_data["item_inventory.csv"][i][2])
                                letak_potion[3] <- i
                                ada_potion <- True
                            if(ada_potion=False) then
                                delay_print ("Anda tidak memiliki Potion dalam
inventory!")
                            else
                                potions <- True
                                while_(potions):
                                    output (" ===== POTION LIST
===== ")
                                    delay_print (f"1. Strength Potion (Qty:
{list_potion[1]}) - Increases ATK Power")
                                    delay_print (f"2. Resilience Potion (Qty:
{list_potion[2]}) - Increases DEF Power")

```

```

        delay_print (f"3. Healing Potion (Qty:
{list_potion[3]}) - Restores Health")
        delay_print ("4. Cancel")
        input(pilihan_p)
        while not is_number(pilihan_p):
            delay_print("Masukkan harus bertipe
Integer, coba lagi!")
            input(pilihan_p)
            pilihan_p <- integer(pilihan_p)
            output("")
            if(pilihan_p = 1) then
                if(list_potion[1]=0) then
                    output("Wah, kamu sedang tidak
memiliki ramuan ini, silahkan pilih ramuan lain!")
                else if (pernah1=1) then
                    delay_print(f"Kamu mencoba
memberikan ramuan ini kepada {player}, namun dia menolaknya
seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat
lagi. \n")
                else
                    power_p <- potion(1,power_p)#fungsi
potion
                    delay_print(f"Setelah meminum ramuan
ini, aura kekuatan terlihat mengelilingi {player} dan gerakannya
menjadi lebih cepat dan mematikan. \n")
                    pernah1 <- 1
                    delay_print (f"Name      :
{player}")
                    delay_print (f"ATK Power   :
{power_p}")
                    delay_print (f"DEF Power   :
{def_p}")
                    delay_print (f"HP          : {hp_p}")
                    delay_print (f"Level       :
{lvl_p}\n")

user_data["item_inventory.csv"][letak_potion[1]][2] <-
integer(user_data["item_inventory.csv"][letak_potion[1]][2])-1
        potions <- False
        turn <- turn + 1
        else if (pilihan_p = 2) then

```

```

        if (list_potion[2]=0)then
            output("Wah, kamu sedang tidak
memiliki ramuan ini, silahkan pilih ramuan lain!")
        else if (pernah2=1) then
            delay_print(f"Kamu mencoba
memberikan ramuan ini kepada {player}, namun dia menolaknya
seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat
lagi. \n")
        else
            delay_print(f"Setelah meminum ramuan
ini, muncul sebuah energi pelindung di sekitar {player} yang
membuatnya terlihat semakin tangguh dan sulit dilukai \n")
        def_p <- potion(2,def_p)#fungsi
        pernah2 <- 1
        delay_print (f"Name      :
{player}")
        delay_print (f"ATK Power   :
{power_p}")
        delay_print (f"DEF Power   :
{def_p}")
        delay_print (f"HP          : {hp_p}")
        delay_print (f"Level      :
{lvl_p}\n")

user_data["item_inventory.csv"][letak_potion[2]][2] <-
integer(user_data["item_inventory.csv"][letak_potion[2]][2])-1
        potions <- False
        turn <- turn + 1
    else if (pilihan_p = 3) then
        if (list_potion[3]=0)then
            output("Wah, kamu sedang tidak
memiliki ramuan ini, silahkan pilih ramuan lain!")
        else if (pernah3=1) then
            delay_print(f"Kamu mencoba
memberikan ramuan ini kepada {player}, namun dia menolaknya
seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat
lagi.")
        else
            delay_print (f"Setelah meminum
ramuan ini, luka-luka yang ada di dalam tubuh {player} sembuh dengan

```

cepat. Dalam sekejap, {player} terlihat kembali prima dan siap melanjutkan pertempuran. \n")

```

hp_p <- hp_p+potion(3,hp_ori)
pernah3 <- 1
if(hp_p>hp_ori) then
    hp_p <- hp_ori
    delay_print (f"Name      :
{player}")

delay_print (f"ATK Power   :
{power_p}")

delay_print (f"DEF Power   :
{def_p}")

delay_print (f"HP          : {hp_p}")
delay_print (f"Level      :
{lvl_p}\n")

```

user_data["item_inventory.csv"][letak_potion[3]][2] <-
integer(user_data["item_inventory.csv"][letak_potion[3]][2])-1
 turn <- turn + 1
 potions <- False
else_if (pilihan_p=4) then
 potions <- False
else
 delay_print("Pilihan Anda tidak valid! \n")
 time.sleep(0.2)

6.10 Arena

```

procedure delay_print (s : string, t <- 0.015 : ):

KAMUS LOKAL
- i : integer
- t : real
- s : string

i traversal[0,len(s)]
    output(i,end <- "")
    sys.stdout.flush()
    time.sleep(t)
output("")

procedure arena (data : Matrix, user_data: Mapping, x : integer, n :
integer, dmg_taken : integer, dmg_given : integer, monster : string,

```

```
power : integer, defend : integer, hp : integer, lvl_p,monster_p :  
integer)
```

KAMUS LOKAL

```
- reward : integer
```

```
reward <- 0  
if(n>1):  
    if(n=2) then  
        reward <- 30  
    else if(n=3) then  
        reward <- 50  
    else if(n=4) then  
        reward <- 100  
    else if(n=5) then  
        reward <- 150  
    else if(n=6) then  
        reward <- 200  
    if(x=1) then  
        delay_print (f"STAGE CLEARED! Anda akan mendapatkan  
{reward} OC pada sesi ini !\n")  
    from src.Battle import battle  
    if(x=1) and (n<=5) then  
  
battle(data,user_data,n,monster,power,defend,hp,dmg_given,dmg_taken,  
lvl_p,monster_p)  
    else  
        if(x=2) then  
            delay_print ("GAME OVER! Anda mengakhiri sesi  
latihan!\n")  
        else  
            if(n<6) then  
                delay_print(f"GAME OVER! Sesi latihan berakhir pada  
stage {n-1}\n")  
            else  
                delay_print (f"Selamat! Anda berhasil menyelesaikan  
seluruh stage Arena !!!\n")  
                delay_print ("===== STATS =====")  
                delay_print (f"Total hadiah : {reward} OC")  
                delay_print (f"Jumlah stage : {n-1}")  
                delay_print (f"Damage diberikan: {dmg_given}")  
                delay_print (f"Damage diterima : {dmg_taken}")
```

```
data[4] <- string(integer(data[4])+reward)
```

6.11 Shop & Currency

```
function is_added (userdata: Matrix, user_id: string, idn: string)
-> boolean
```

KAMUS LOKAL

```
- i: List of string
```

```
for each i in userdata do
    if i[0] = user_id and i[1] = idn then
        return True
    end if
return False
end for
return None
```

```
function index_added_potion (potion_inventory: Matrix, potion_name:
string, idn: string, user_id) -> integer
```

KAMUS LOKAL

```
- i : List of string
```

```
for each i, row in enumerate(potion_inventory) do
    if row[0] = user_id and row[1] = potion_name then
        return i
    end if
return -1
end for
```

```
procedure beli (user_data: Mapping, user_login : List of string)
```

KAMUS LOKAL

```
- breaked: boolean
```

```
- c: string
```

```
- monster: Matrix
```

```
- monster_shop: Matrix
```

```
- inventory: Matrix
```

```
- idn: string
```

```
- row_index: integer
```

```
- price: integer
```

```
output("Jumlah O.W.C.A Coin-mu sekarang %s.\n"
    %(user_login[4]))
```

```
breaked <- False
```

```

while not breaked do
    c <- input(">>> Mau beli apa? (monster/potion): ")
    if c in ['monster', 'potion'] then
        breaked <- True
    end while
    depend on (c)
        c = 'monster':
            monster <- user_data["monster.csv"]
            monster_shop <- user_data["monster_shop.csv"]
            inventory <- user_data["monster_inventory.csv"]

            idn <- get_idx(user_data["monster.csv"], "monster")
            row_index <- is_in(monster_shop, idn, 0)
            price <- integer(monster_shop[row_index][2])
            depend on ()
                monster_shop[row_index][1] = '0' :
                    output("Monster %s sedang tidak tersedia
(kosong)." % (monster[is_in(monster, idn, 0)][1]))
                    is_added(inventory, user_login[0], idn):
                        output("Monster %s sudah ada dalam
inventory-mu! Pembelian dibatalkan." % (monster[is_in(monster,
idn, 0)][1]))
                        integer(user_login[4]) < price:
                            output("OC-mu tidak cukup.")
                    else
                        user_login[4] = string(integer(user_login[4])
- price)
                            inventory.append([user_login[0], idn, '1'])
                            monster_shop[row_index][1] = string(
integer(monster_shop[row_index][1]) - 1)
                            output("Berhasil membeli item: %s. Item sudah
masuk ke inventory-mu!" %(monster[is_in(monster, idn, 0)][1]))

            c = "potion" :
                potion <- user_data["item_shop.csv"]
                potion_inventory <- user_data["item_inventory.csv"]
                idn <- get_idx(add_id(potion), 'potion')
                price <- integer(potion[integer(idn)][2])

function get_jumlah() -> string
KAMUS LOKAL

```

```

- cc: string
- jumlah: string
cc <- input(">>> Masukkan jumlah: ")
while not is_number(cc) do
    output("Masukkan input bertipe Integer, coba lagi!")
    output()
    cc <- input(">>> Masukkan jumlah: ")
return integer(cc)
end while
jumlah <- get_jumlah()
depend on
    integer(user_login[4]) < jumlah * price:
        output("OC-mu tidak cukup.")
    jumlah > integer(potion[integer(idn)][1]):
        output("Jumlah yang diminta melebihi stok.")

else
    potion[integer(idn)][1]<-string(integer(potion[integer(idn)][1
]) - jumlah)
    user_login[4] <- string(integer(user_login[4]) - jumlah * 
price)
    user_inventory_idx <- index_added_potion(potion_inventory,
potion[integer(idn)][0], idn, user_login[0])
    if user_inventory_idx = -1 then
        potion_inventory.append([user_login[0], potion[integer(idn
)][0], string(jumlah)])
    else
        potion_inventory[user_inventory_idx][2] <-
        string(integer(potion_inventory[user_inventory_idx][2]) +
jumlah)
    end if

output("Berhasil membeli item: %s Potion of %. Item sudah masuk ke
inventory-mu!" % (jumlah, potion[int(idn)][0]))

```

procedure shop_and_currency(user_data, user_login):

KAMUS LOKAL

- breaked: boolean
- masukan: string

output("Irasshaimase! Selamat datang di SHOP!!")

breaked <- False

```

while not brokead:
    masukan <- (input("\\n>>> Pilih aksi (lihat/beli/keluar) :
"))
    depend on (masukan)
        masukan.lower() = "lihat":
            lihat(user_data)
        masukan.lower() = "beli":
            beli(user_data, user_login)
        masukan.lower() = "keluar":
            output("Mr. Yanto bilang makasih, belanja lagi
ya nanti :)")
            brokead <- True
end while

```

6.12 Laboratory

```

from typing import List, Dict
from .helper.IsType import is_number
function get_name_by_id (idx: string, data: Matrix) -> string
    KAMUS LOKAL
    - i: List of string

    for each i in data do
        if i[0] = idx then
            return i[1]
        end if
    end for
    return None

```

```

function get_id (n: integer) -> integer
    KAMUS LOKAL
    - c: string

    c <- input(">>> Pilih monster: ")
    while not is_number(c) or not (1 <= integer(c) <= n) do
        if not is_number(c) then
            output("Masukkan input bertipe Integer, coba lagi!")
        else if not (1 <= integer(c) <= n) then
            output("Urutan tidak ditemukan!")
        end if
        c <- input(">>> Pilih monster: ")

```

```

end while
return integer(c)
end function

function verif() -> boolean
    KAMUS LOKAL
    - c: string
    while True do
        input(">>> Lanjutkan upgrade (Y/N): ")
        if c.lower() = 'y' then
            return True
        elif c.lower() = 'n' then
            return False
        end if
    end while

```

```

procedure laboratory (user_login: List of string, user_data:
Mapping)

```

```

    KAMUS LOKAL
    - monster_list: Matrix
    - harga: integer
    - idx: integer
    - monster_name: string
    - lvl: integer

    output("Selamat datang di Lab Dokter Asep !!!\n")
    output("===== MONSTER LIST =====\n")
    monster_list <- []
    for each row in user_data["monster_inventory.csv"] do
        if row[0] = user_login[0] then
            monster_list.append(row)
        end if
    end for
    i traversal [1..length(monster_list)]
        output(i, " . ", get_name_by_id(monster_list[i-1][1],
user_data["monster.csv"]), " (Level: ", monster_list[i-1][2], ")\n")
    end for
    output("\n")
    output("===== UPGRADE PRICE =====\n")
    output("1. Level 1 -> Level 2: 300 OC\n")
    output("2. Level 2 -> Level 3: 500 OC\n")

```

```

output("3. Level 3 -> Level 4: 800 OC\n")
output("4. Level 4 -> Level 5: 1000 OC\n")

harga <- {
    "2": 300,
    "3": 500,
    "4": 800,
    "5": 1000
}
idx <- get_id(length(monster_list))
if monster_list[idx-1][2] = '5' then
    output("Maaf, monster yang Anda pilih sudah memiliki level
maksimum\n")
else
    monster_name <- get_name_by_id(monster_list[idx-1][1],
user_data["monster.csv"])
    lvl <- integer(monster_list[idx-1][2]) + 1
    output("\\n", monster_name, " akan di-upgrade ke level ",
lvl, ".\\n")
    output("Harga untuk melakukan upgrade ", monster_name, " "
adalah ", harga[str(lvl)], " OC.\\n")
    if verif() then
        output("\\n")
        if integer(user_login[4]) < harga[str(lvl)] then
            output("OC tidak cukup\\n")
        else
            monster_list[idx-1][2] <- str(lvl)
            user_login[4] <- str(integer(user_login[4]) -
harga[str(lvl)])
            output("Selamat, ", monster_name, " berhasil
di-upgrade ke level ", lvl, " !\\n")
        end if
    end if
end if

```

6.13 Shop Management

procedure lihat (user_data: Mapping)

KAMUS LOKAL

- breaked: boolean
- c: string

```

breaked <- false
while not breaked do
    output(">>> Mau lihat apa? (monster/potion): ")
    input(c)
    panggil show(c, user_data)
    if c in ['monster', 'potion'] then
        breaked <- true
    end if
end while
end procedure

```

procedure tambah (user_data: Mapping)

KAMUS LOKAL

- breaked: boolean
- c, idx, stock, price: string
- contoh, arr, monster: Matrix
- idn: integer
- potion : Matrix

```

breaked <- false
while not breaked do
    output(">>> Mau lihat apa? (monster/potion): ")
    input(c)
    if c in ['monster', 'potion'] then
        breaked <- true
    end if
end while

```

```

if c = 'monster' then
    contoh <- [ ["ID", "Type", "ATK Power", "DEF Power", "HP"],
                ["4", "Cici", "10", "1000", "200"],
                ["5", "Moskov", "20", "1000", "200"],
                ["6", "Selena", "30", "430", "100"] ]
    arr <- []
    monster <- user_data["monster.csv"]
    for each i in contoh do
        if is_in(monster, i[0], 0) = -1 then
            arr.append(i)
        end if

```

```

    end for
    panggil table_print(arr)
    idx <- get_idx(arr, 'monster')
    stock <- get_stock("tambah")
    price <- get_price("tambah")
    idn <- is_in(arr, idx, 0)
    user_data["monster.csv"].append(arr[idn])
    user_data["monster_shop.csv"].append([arr[idn][0], stock,
price])
else if c = 'potion' then
    contoh <- [[ "ID", "Type" ],
                [ "3", "healing" ]]
    arr <- []
    potion <- user_data["item_shop.csv"]
    for each i in contoh do
        if is_in(potion, i[0], 0) = -1 then
            arr.append(i)
        end if
    end for
    panggil table_print(arr)
    idx <- get_idx(arr, 'potion')
    stock <- get_stock()
    price <- get_price()
    idn <- is_in(arr, idx, 0)
    user_data["item_shop.csv"].append([arr[idn][1], stock,
price])
end if

```

procedure ubah(user_data: Mapping)

KAMUS LOKAL

- breaked : boolean
- c, idx, stock, price: string
- i : integer
- potion : List of string

```

breaked <- False
while not breaked do
    c: string <- input(">>> Mau lihat apa? (monster/potion): ")
    show(c, user_data)
    if c in ['monster', 'potion'] then
        breaked <- True
end while

```

```

if c = 'monster' then
    monster_shop: Matrix <- user_data[ "monster_shop.csv" ]
    monster_list: Matrix <- concat(user_data[ "monster.csv" ],
user_data[ "monster_shop.csv" ])
    idx <- get_idx(monster_list, 'monster')
    stock <- get_stock()
    price<- get_price()
    i traversal [0..len(monster_shop)-1]
        row <- monster_shop[i]
        if row[0] = idx then
            if stock != "" then
                row[1] <- stock
            if price != "" then
                row[2] <- price
            return
        end if
    end if
else if c = 'potion' then
    potion: Matrix <- add_id(user_data[ "item_shop.csv" ])
    idx <- get_idx(potion, 'potion')
    stock <- get_stock()
    price <- get_price()
    potion <- user_data[ "item_shop.csv" ]
    i traversal [0..len(potion)-1]
        data <- potion[i]
        if i = integer(idx) then
            if stock != "" then
                data[1] <- stock
            if price != "" then
                data[2] <- price
            end if
        return
    end if
return

```

procedure hapus (user_data: Mapping)

KAMUS LOKAL

- breaked: boolean
- monster_shop, monster, monster_list: Matrix
- idx: string

```

breaked <- False
while not breaked do
    output(">>> Mau lihat apa? (monster/potion): ")
    input(c)
    show(c, user_data)
    if c in ['monster', 'potion'] then
        breaked <- True
    endif
end while
if c = 'monster' then
    monster_shop <- user_data["monster_shop.csv"]
    monster <- user_data["monster.csv"]
    monster_list <- concat(monster, monster_shop)
    idx<- get_idx(monster_list, 'monster')
    if verification_hapus(idx, monster_list) then
        user_data["monster_shop.csv"] <-
write_without(user_data["monster_shop.csv"], idx)
        user_data["monster.csv"] <-
write_without(user_data["monster.csv"], idx)
    else if c = 'potion' then
        potion <- add_id(user_data["item_shop.csv"])
        idx <- get_idx(potion, 'potion')
        if verification_hapus(idx, potion) then
            user_data["item_shop.csv"] <-
write_without(user_data["item_shop.csv"], idx, True)
    return

```

procedure shop_management (user_data: Mapping)

KAMUS LOKAL

- breaked : boolean
- act : string

output("Irasshaimase! Selamat datang kembali, Mr. Admin!\n")

breaked <- False

while not breaked do

output(">>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ")

input(act)

if act.lower() = 'lihat' then

 lihat(user_data)

 breaked <- True

else if act.lower() = 'tambah' then

```

        tambah(user_data)
        breaked <- True
else if act.lower() = 'ubah' then
        ubah(user_data)
        breaked <- True
else if act.lower() = 'hapus' then
        hapus(user_data)
        breaked <- True
else if act.lower() = 'keluar' then
        output("Dadah Mr. Admin")
        breaked <- True
end if
return
end while

```

function concat(a1: Matrix, a2: Matrix) -> Matrix

KAMUS LOKAL

```

- a1, a2: Matrix
- row, col_a1, col_a2, col, idx:, i integer
- new_arr: List of string
row <- len(a1)
col_a1 <- len(a1[0])
col_a2 <- len(a2[0])
col <- col_a1 + col_a2 - 1
new_arr <- [[" for _ in range(col)] for _ in range(row)]
i traversal [0]
    j traversal [0..col_a1-1]
        new_arr[i][j] <- a1[i][j]
    j traversal [1..col_a2-1]
        new_arr[i][col_a1 + j - 1] <- a2[i][j]
i traversal [1..row-1]
    j traversal [0..col_a1-1]
        new_arr[i][j] <- a1[i][j]
    idx <- -1
    j traversal [0..len(a2)-1]
        if new_arr[i][0] = a2[j][0] then
            idx <- j
        if idx != -1 then
            j traversal [1..col_a2-1]
            new_arr[i][col_a1 + j - 1] <- a2[idx][j]
    end if

```

```
return new_arr
```

```
function write_without(user_data, idx, is_potion = False) -> Matrix
```

KAMUS LOKAL

- user_data, arr : Matrix
- idx : integer
- is_potion : Boolean
- arr : array

```
arr <- []
```

```
if is_potion then
```

```
    i trasversal [from 0 to length(user_data) - 1] do
```

```
        if i is not equal to integer(idx) then
            append user_data[i] to arr
```

```
else
```

```
    i traversal [0..length(user_data)-1]
```

```
        if user_data[i][0] /= idx then
            append user_data[i] to arr
```

```
end if
```

```
return arr
```

```
function verification_hapus(idx, user_data) -> boolean
```

KAMUS LOKAL

- idx, is_hapus: string

- user_data: Matrix

```
is_hapus <- ""
```

```
while is_hapus.lower() is not_in ['y', 'n'] do
```

```
    is_hapus <- input(">>> Apakah anda yakin ingin menghapus %s  
dari shop (y/n)? " % user_data[is_in(user_data, idx, 0)][1])
```

```
    if is_hapus.lower() = 'y' then
```

```
        return True
```

```
    return False
```

```
function get_stock(s = "ubah") -> string
```

KAMUS LOKAL

- s, stock: string

```
output(">>> Masukkan stok baru: ")
```

```
input(stock)
```

```
if s = 'tambah' then
```

```
    while not (is_number(stock)) or length(stock) <= 0 do
```

```

    output("Masukkan input bertipe Integer nonnegatif, coba
lagi!\n")
    output(">>> Masukkan stok baru: ")
    input(stock)
else if s = 'ubah' then
    while not (is_number(stock)) and length(stock) > 0 do
        output("Masukkan input bertipe Integer nonnegatif, coba
lagi!\n")
        output(">>> Masukkan stok baru: ")
        input(stock)
return stock

```

function get_price(s = "ubah") -> string

KAMUS LOKAL

```

- s, price: string
output(">>> Masukkan harga baru: ")
input(price)
if s = 'tambah' then
    while not (is_number(price)) or length(price) <= 0 do
        output("Masukkan input bertipe Integer nonnegatif, coba
lagi!\n")
        output(">>> Masukkan harga baru: ")
        input(price)
else if s = 'ubah' then
    while not (is_number(price)) and length(price) > 0 do
        output("Masukkan input bertipe Integer nonnegatif, coba
lagi!\n")
        output(">>> Masukkan harga baru: ")
        input(price)
return price

```

function is_in(arr: Matrix, e: string, idx: integer) -> boolean

KAMUS LOKAL

```

- arr : Matrix
- e : string
- idx : integer
i traversal [0 to length(arr)-1]
    if arr[i][idx] = e then
        return i
return -1

```

```

function get_idx(data , txt) -> string
KAMUS LOKAL
- idx, txt: string
- data: Matrix
idx <- input(">>> Masukkan id " + txt + ": ")
while not (is_number(idx) and is_in(data, idx, 0) != -1) do
    if not is_number(idx) then
        output("Masukkan input bertipe Integer, coba lagi!")
    else
        output("ID tidak ada dalam data!")
    output()
    output(">>> Masukkan id " + txt + ": ")
    output(idx)
return idx

```

procedure show(c, user_data)

```

KAMUS LOKAL
- monster_to_print : Matrix
- user_data: Mapping
- c : string
if c = 'monster' then
    monster_to_print <- concat(user_data["monster.csv"],
user_data["monster_shop.csv"])
    table_print(monster_to_print)
else if c = 'potion' then
    table_print(add_id(user_data["item_shop.csv"]))

```

6.14 Monster Management

ALGORITMA

function get_type (user_data: Mapping) -> string

KAMUS LOKAL

- name: string

function is_added (name: string) -> boolean

KAMUS LOKAL

- monster_data: Matrix
- row: List of string

```

monster_data <- user_data["monster.csv"]
for each row in monster_data do
    if name = row[1] then

```

```

        return true
    end if
end for
return false

output( ">>> Masukkan Type / Nama: " )
input(name)
while is_added(name) do
    output( "Nama sudah terdaftar, coba lagi!" )
    output()
    output( ">>> Masukkan Type / Nama: " )
    input(name)
end while
return name

```

function get_atk_power() -> string

KAMUS LOKAL

- atk_power: string

```

output( ">>> Masukkan ATK Power: " )
input(atk_power)
while not is_number(atk_power) do
    output( "Masukkan input bertipe Integer, coba lagi!" )
    output()
    output( ">>> Masukkan ATK Power: " )
    input(atk_power)
end while
return atk_power

```

function get_def_power() -> string

KAMUS LOKAL

- def_power: string

```

function is_valid (def_power: string) -> boolean
    if not is_number(def_power) then
        return false
    end if
    if not (0 <= integer(def_power) <= 50) then
        return false
    end if
    return true

```

```
end function

output(">>> Masukkan DEF Power (0-50): ")
input(def_power)
while not is_valid(def_power) do
    if not is_number(def_power) then
        output("Masukkan input bertipe Integer, coba lagi!")
    else if not (0 <= integer(def_power) <= 50) then
        output("DEF Power harus bernilai 0-50, coba lagi!")
    end if
    output()
    output(">>> Masukkan DEF Power (0-50): ")
    input(def_power)
end while
return def_power
```

```
function get_hp() -> string
```

KAMUS LOKAL

- hp: string

```
output(">>> Masukkan HP: ")
input(hp)
while not is_number(hp) do
    output("Masukkan input bertipe Integer, coba lagi!")
    output()
    output(">>> Masukkan HP: ")
    input(hp)
end while
return hp
```

```
function get_last_id (user_data: Mapping) -> integer
```

KAMUS LOKAL

- data: Matrix
- last_id: integer

```
data <- user_data["monster.csv"]
last_id <- integer(data[length(data)][0])
return last_id
```

```
procedure add_monster (user_data: Mapping)
```

KAMUS LOKAL

```

- idn: integer
- name, atk_power, def_power, hp, choice: string

output("Memulai pembuatan monster baru")
output()

idn <- get_last_id(user_data)
name <- get_type(user_data)
atk_power <- get_atk_power()
def_power <- get_def_power()
hp <- get_hp()

output("Monster baru berhasil dibuat!")
output("Type:", name)
output("ATK Power:", atk_power)
output("DEF Power:", def_power)
output("HP:", hp)

while choice.upper() ≠ 'Y' and choice.upper() ≠ 'N' do
    end while
if choice.upper() = 'Y' then
    user_data["monster.csv"].append([str(idn+1), name,
atk_power, def_power, hp])
    output("Monster baru telah ditambahkan!")
end if
end procedure

procedure monster_management (user_data: Mapping)
KAMUS LOKAL
- choice: string

output("SELAMAT DATANG DI DATABASE PARA MONSTER!!!")
output("1. Tampilkan semua Monster")
output("2. Tambah Monster baru")
choice <- input(">>> Pilih aksi: ")
if choice = '1' then
    table_print(user_data["monster.csv"])
else if choice = '2' then
    add_monster(user_data)

```

```

else
    monster_management(user_data)
end if

```

6.15 Load

```

import os
import argparse
from src.MenuAndHelp import menu
procedure load()
    KAMUS LOKAL
        - parser: argparse.ArgumentParser
        - args: argparse.ArgumentParser
        - folder_name: string

    function folder_verification (folder_name: string) -> boolean
        return is_directory("./data/" + folder_name)

    parser <- create_argument_parser(description='Verifies if a
folder exists.')
        parser.add_argument('folder_name', nargs='?', help='Nama
folder')
    args <- parse_args(parser)

    folder_name <- args.folder_name
    if folder_name = None then
        output('Masukkan nama folder Anda.')
        output('usage: python main.py <nama_folder>')
    else if not folder_verification(args.folder_name) then
        output('Folder ' + folder_name + ' tidak ditemukan, silakan
ulangi!')
    else
        output('Folder ' + folder_name + ' ditemukan!')
        menu(folder_name)
    end if

```

6.16 Save

```

import os
import time
import sys
from typing import List, Dict, TextIO
from .helper.ListManipulation import join

```

```
procedure pprint (folder_name: string, c: integer = 2)
```

KAMUS LOKAL

- i: U

```
procedure delayed_print (s: string)
```

KAMUS LOKAL

- i: string

```
for each i in s do
```

```
    output(i, end="")
```

```
    flush_output()
```

```
    sleep(0.05)
```

```
end for
```

```
for each i in "..." do
```

```
    output(i, end="")
```

```
    flush_output()
```

```
    sleep(0.75)
```

```
end for
```

```
end procedure
```

```
delayed_print("Saving")
```

```
output()
```

```
if c = 0 then
```

```
    delayed_print("Membuat folder data")
```

```
    output()
```

```
else_if c <= 1 then
```

```
    delayed_print("Membuat folder data/" + folder_name)
```

```
    output()
```

```
end if
```

```
output("Berhasil menyimpan data di folder data/" + folder_name +
```

```
"!")
```

```
procedure write_lines (file_name: string, folder_name: string,  
user_data: Mapping)
```

KAMUS LOKAL

- f: file

- data_to_write: string

- row: List of string

```

f <- open("./data/" + folder_name + "/" + file_name, 'w')
data_to_write <- ""
for each row in user_data[file_name] do
    data_to_write <- data_to_write + join(row) + "\n"
end for
f.write(data_to_write)
f.close()
end procedure

```

procedure save (user_data: Mapping)

KAMUS LOKAL

- folder_name, file_name: string

```



```

```

if not is_directory("./data") then
    output(folder_name, 0)
    make_directory("if1210-2024-tubes-k10-b/data")
else if not is_directory("./data/" + folder_name) then
    output(folder_name, 1)
    make_directory("./data/" + folder_name)
else if is_directory("./data/" + folder_name) then
    output(folder_name)
end if

```

```

for each file_name in user_data do
    write_lines(file_name, folder_name, user_data)
end for

```

6.17 Exit

```

from src.Save import save
procedure leave (data: Mapping)
KAMUS LOKAL
- choice: string

```

```



```

```

if to_lower(choice) = 'y' then
    save(data)

```

```
else
    if to_lower(choice) ≠ 'n' then
        leave(data)
    end if
end if
```

VII. Tangkapan Layar Testing

berisi: (1) data masukan, (2) data keluaran untuk *input* yang valid dan tidak valid jika terdapat validasi.

7.1 Random Number Generator

Random Number Generator merupakan program pembantu sehingga tidak ada validasi yang dapat dilihat.

7.2 Register

Gambar 7.1 : F02 - Register
– register berhasil –

```
>>> REGISTER
Masukan username: Fathur
Masukan password: Ganteng

Silakan pilih salah satu monster sebagai monster awalmu.
0. Dokumon Api
1. Dokumon Air
2. Dokumon Tanah

ID monster pilihanmu: 0

Selamat datang Agent Fathur. Mari kita mengalahkan Dr. Asep Spakbor dengan Dokumon Api!
```

Gambar 7.2 : F02 - Register
– register gagal (username sudah terpakai) –

```
>>> REGISTER
Masukan username: Admin
Masukan password: Admin
Username Admin sudah terpakai, silahkan gunakan username lain!
```

Gambar 7.3 : F02 - Register
– register gagal (username tidak sesuai prosedur) –

```
>>> REGISTER  
Masukan username: f@a7hur  
Masukan password: 64n73ng  
Username hanya boleh berisi alfabet, angka, underscore, dan strip!
```

Gambar 7.4 : F02 - Register

– register gagal (user sedang dalam kondisi login) –

```
>>> LOGIN  
Masukkan username Anda: b  
Masukkan password Anda: b  
You are logged in as Agent!  
  
>>> REGISTER  
Register gagal!  
Anda telah login dengan username b, silahkan lakukan "LOGOUT" sebelum melakukan register.
```

7.3 Login

Gambar 7.5 : F03 - Login

– login berhasil –

```
>>> LOGIN  
Masukkan username Anda: fathur  
Masukkan password Anda: fathur  
You are logged in as Agent!
```

Gambar 7.6 : F03 - Login

– login gagal (password salah) –

```
>>> LOGIN  
Masukkan username Anda: fathur  
Masukkan password Anda: withyou  
Wrong password, try again!
```

Gambar 7.7 : F03 - Login

– login gagal (username tidak terdaftar) –

```
>>> LOGIN  
Masukkan username Anda: orangganteng  
Masukkan password Anda: hello  
Username doesn't exist!
```

Gambar 7.8 : F03 - Login

– login gagal (user sudah login) –

```
>>> LOGIN  
Masukkan username Anda: fathur  
Masukkan password Anda: fathur  
You are logged in as Agent!  
  
>>> LOGIN  
You have been logged in!
```

7.4 Logout

Gambar 7.9 : F04 - Logout
– logout dengan skema validasi –

```
>>> LOGIN  
Masukkan username Anda: fathur  
Masukkan password Anda: fathur  
You are logged in as Agent!  
  
>>> LOGOUT  
  
>>> LOGOUT  
Logout gagal!  
Anda belum login, silakan login terlebih dahulu sebelum melakukan logout
```

7.5 Menu & Help

Gambar 7.10 : F05 - Menu & Help
– help message untuk guest –

```
>>> HELP  
===== HELP =====  
  
Kamu belum login sebagai role apapun. Silakan login terlebih dahulu.  
Menu yang tersedia:  
[HELP] Membantu untuk melihat command yang ada  
[REGISTER] User melakukan registrasi akun  
[LOGIN] User melakukan login akun  
[EXIT] Guest keluar dari game
```

Gambar 7.11 : F05 - Menu & Help
– help message untuk agent –

```
>>> HELP
===== HELP =====

Halo Agent b. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
Menu yang tersedia:
[HELP] Membantu untuk melihat command yang ada
[INVENTORY] User melihat penyimpanan
[SHOP] User Agent membeli monster dan potion
[LABORATORY] Agent melakukan upgrade monster
[ARENA] Agent melatih monster
[BATTLE] Agent melakukan battle
[LOGOUT] Agent melakukan logout akun
[SAVE] Agent melakukan save progres
[EXIT] User keluar dari game
```

Gambar 7.12 : F05 - Menu & Help
– help message untuk admin –

```
>>> HELP
===== HELP =====

Selamat datang, Admin. Berikut adalah hal-hal yang dapat kamu lakukan:
Menu yang tersedia:
[HELP] Membantu untuk melihat command yang ada
[SHOP] Admin mengatur manajemen Shop and Currency
[MONSTER] Admin mengatur manajemen Monster
[SAVE] Admin melakukan save progres
[LOGOUT] Admin melakukan logout akun
[EXIT] Admin keluar dari game
```

7.6 Monster

Bagian Monster merupakan bagian informasi mengenai aturan monster dan sudah disatukan dengan Battle dan Arena sehingga validasi tidak dapat dinilai.

7.7 Potion

Bagian Potion merupakan bagian informasi mengenai aturan potion dan sudah disatukan dengan Battle dan Arena sehingga validasi tidak dapat dinilai.

7.8 Inventory

Gambar 7.13 : F08 - Inventory
– detail item dan list dari inventory –

```
>>> INVENTORY
===== INVENTORY LIST (User ID: 4) =====
Jumlah O.W.C.A. Coin-mu sekarang 992.
1. Monster      (Name: Dokumon Air, Lvl: 2, HP: 75)
2. Monster      (Name: Dokumon Tanah, Lvl: 3, HP: 1200)
3. Monster      (Name: Dokumon Api, Lvl: 1, HP: 135)
4. Potion       (Type: Heal, Qty: 3)
5. Potion       (Type: DEF, Qty: 4)
6. Potion       (Type: ATK, Qty: 10)
```

Ketikkan id untuk menampilkan detail item atau [KELUAR] untuk keluar dari inventory:

```
>>> 1
Monster
Name      : Dokumon Air
ATK Power : 75
DEF Power : 36
HP        : 75
Level     : 2
```

Ketikkan id untuk menampilkan detail item atau [KELUAR] untuk keluar dari inventory:

```
>>> 4
Potion
Type      : Heal
Quantity  : 3
```

Ketikkan id untuk menampilkan detail item atau [KELUAR] untuk keluar dari inventory:

```
>>> KELUAR
```

7.9 Battle

Gambar 7.14 : F09 - Battle
– tampilan awal menu battle –

```
>>> BATTLE

          .+####+.          .+####-
-#- ..+#.          .#+.. -#+
+#+.-+ .#-          -#+ +- ++

.##++##+. .+#####+. .+##++##.
.-.#####-. .-#####-
      #####      #####
.+- .  #####-##  ##-+###  . -#.
##. +- .#-  ###+ .#.  .#. +###  -#. +- .#-
-#. -+.#+ -###  .+  +.  +###  -#. -+.#+
-####+ +##.
-###  .###  ####-  ####-
+#+  -###  ####-  -#+
-#- .+###  +###.  -#+
+#####+
      +##  ####
      +###  .+  +.  +###  .#.  +###
      +###+ -##  ##-+#####
.##-#####  #####-##.
+#+ -#####+. .+#####- +#
+##  +#####+
      +#+  -#####+  +#
      .###  +##.
.#####.  .#####.
-+#####+-
```

RAWRRR, Monster Dokumon Tanah telah muncul !!!

Name : Dokumon Tanah
ATK Power : 130
DEF Power : 65
HP : 1300
Level : 4

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

1. Dokumon Api, level: 3
2. Dokumon Air, level: 2
3. Dokumon Tanah, level: 4

Pilih monster untuk bertarung:

Gambar 7.15 : F09 - Battle
– pesan error saat masukan salah –

```
===== MONSTER LIST =====

1. Dokumon Api, level: 3
2. Dokumon Air, level: 2
3. Dokumon Tanah, level: 4
Pilih monster untuk bertarung:
daspro
Masukkan harus bertipe Integer, coba lagi!
```

Gambar 7.16 : F09 - Battle
- tampilan setelah player memilih monster -

```
===== MONSTER LIST =====

1. Dokumon Api, level: 3
2. Dokumon Air, level: 2
3. Dokumon Tanah, level: 4
Pilih monster untuk bertarung:
daspro
Masukkan harus bertipe Integer, coba lagi!
1

-+---.
+- .+.. ..-++++++-.. .+.. +-+
+. -+. .++-. .++. .+- .+-
+- -+- -+- -+- -+- -+- -+- -+-
.-+ #+. .+#
.+. #####. .+#####. .+.
+. ...+##- -##+... .#
.+ ..... +.
+. .-..##+ +##..-. .#
.+. .#- -##++-+##- -+- .+.
+- -++##- ##. -##+- .-+
.-+. ....-+----- .+-
.-+ .+###+#+##+ .+
.+ .+- -+. +.
+++ -+-
```

RAWRRR, AGENT b mengeluarkan monster Dokumon Api

Name : Dokumon Api
ATK Power : 235
DEF Power : 36
HP : 162
Level : 3

===== TURN 1 (Dokumon Api) =====

1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: █

Gambar 7.17 : F09 - Battle
– pesan error saat perintah salah –

```
===== TURN 1 (Dokumon Api) =====
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: Daspro

Pilihan Anda tidak valid!

===== TURN 1 (Dokumon Api) =====
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: █
```

Gambar 7.18 : F09 - Battle
– tampilan saat player melakukan serangan –

```
===== TURN 1 (Dokumon Api) =====
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: 1

SCHWINKKK, Dokumon Api menyerang Dokumon Tanah

DETAIL SERANGAN:
Dokumon Api mencoba untuk menyerang Dokumon Tanah dengan power sebesar 222 (-5%)
Dokumon Tanah menahan serangan dari Dokumon Api sebesar 94 (42%)
sehingga damage yang diterima oleh Dokumon Tanah sebesar 128

Name      : Dokumon Tanah
ATK Power : 130
DEF Power : 65
HP        : 1172
Level     : 4
```

Gambar 7.19 : F09 - Battle
– tampilan saat player diserang oleh lawan –

```
-----+
-+###+.          .+#####
-#- ..+#.          .#+.. -#+
+#+.-+ .#-          -#+ -+ #+
.#+#+#++. .+#####+. .+#+#+#.
.-.#####-. -#####
#####          #####
.+- . #####-##    ##-+#####
#. .+- .#- #####+ .#. .#+## -#. -+ .#-
-#.-+.#+ -###+ .+    +. +##- -#.-+.#+
-#####+ +##.          .### +#####
-### .###          ###- ####-
+#+ -###          ###- -#+
-#- .+###          +##+. -#+
+#####          #####
+##          #####
+### .+    +. +##+
###+ .#.          .#. +##+
+###+-##          ##-+#####
.##-#####          #####-##.
++- -#####+. .+###+- +##+
### +#####          ##+
++- -#####+          +##
.###          +##.
.#####          .#####
-#####+-          -#####
===== TURN 2 (Dokumon Tanah)=====
SCHWINKKK, Dokumon Tanah menyerang Dokumon Api

DETAIL SERANGAN:
Dokumon Tanah mencoba untuk menyerang Dokumon Api dengan power sebesar 95 (-26%)
Dokumon Api menahan serangan dari Dokumon Tanah sebesar 24 (25%)
sehingga damage yang diterima oleh Dokumon Api sebesar 71

Name      : Dokumon Api
ATK Power : 235
DEF Power : 36
HP        : 91
Level     : 3
```

Gambar 7.20 : F09 - Battle
– tampilan saat player memilih perintah “use potion” –

```
===== TURN 3 (Dokumon Api) =====
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: 2

===== POTION LIST =====
1. Strength Potion (Qty: 10) - Increases ATK Power
2. Resilience Potion (Qty: 4) - Increases DEF Power
3. Healing Potion (Qty: 3) - Restores Health
4. Cancel
Pilihan Potion: █
```

Gambar 7.21 : F09 - Battle
– pesan error saat masukan pilihan potion salah –

```
===== POTION LIST =====
1. Strength Potion (Qty: 10) - Increases ATK Power
2. Resilience Potion (Qty: 4) - Increases DEF Power
3. Healing Potion (Qty: 3) - Restores Health
4. Cancel
Pilihan Potion: potion power
Masukkan harus bertipe Integer, coba lagi!
Pilihan Potion: █
```

Gambar 7.22 : F09 - Battle
– tampilan saat player menggunakan “Healing Potion” –

```
Pilihan Potion: 3

Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh Dokumon Api sembuh dengan cepat. Dalam sekejap, Dokumon Api terlihat kembali prima dan siap melanjutkan pertempuran.

Name      : Dokumon Api
ATK Power : 235
DEF Power : 36
HP        : 131
Level     : 3
```

Gambar 7.23 : F09 - Battle
– tampilan saat player menggunakan “Resilience Potion” –

```
===== POTION LIST =====
1. Strength Potion (Qty: 10) - Increases ATK Power
2. Resilience Potion (Qty: 4) - Increases DEF Power
3. Healing Potion (Qty: 2) - Restores Health
4. Cancel
Pilihan Potion: 2

Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Dokumon Api yang membuatnya terlihat semakin tangguh dan sulit diluka

Name      : Dokumon Api
ATK Power : 235
DEF Power : 37
HP        : 53
Level     : 3
```

Gambar 7.24 : F09 - Battle
– tampilan saat player menggunakan “Strength Potion” –

```
===== POTION LIST =====
1. Strength Potion (Qty: 10) - Increases ATK Power
2. Resilience Potion (Qty: 3) - Increases DEF Power
3. Healing Potion (Qty: 2) - Restores Health
4. Cancel
Pilihan Potion: 1

Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Dokumon Api dan gerakannya menjadi lebih cepat dan mematikan.

Name      : Dokumon Api
ATK Power : 246
DEF Power : 36
HP        : 162
Level     : 3
```

Gambar 7.25 : F09 - Battle
– tampilan saat player mencoba untuk menggunakan potion yang sudah pernah digunakan –

```
W cancel
Pilihan Potion: 1

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

===== POTION LIST =====
1. Strength Potion (Qty: 9) - Increases ATK Power
2. Resilience Potion (Qty: 3) - Increases DEF Power
3. Healing Potion (Qty: 2) - Restores Health
4. Cancel
Pilihan Potion: ■
```

Gambar 7.26 : F09 - Battle
– tampilan saat player dikalahkan oleh lawan –

```

+#+#+#+.          .+#####
-#- ..+#.          .#+.. -#+
+#+ -+ .#+-        -#+ -+ #+
.#+++#+. .+#####+. .+#+#+#+.
.-.#####. .-#####-
###### ######
.+- .   ####-##   ##-+###   . -#.
##. +- .-#   ###+ .#.   .#. +### -#. -+ .##
-#, -+.#- -### .+    +. +##- -#. -+ .#+
-#####+ +##.          .##+ +#####
-### .###          ##- ##-
+#+ -###          ##- -#+
-#- .+###          +##+. -#+
+#####.          #######+
+##          #####
+##. .+    +. +##+
###+ .#.          .#. +###
+##+-###          ##-+#####
.##-#####.          #######.
+##- -#####+. .-#####- +##
+## +#####.          ##+
+##- -#####+. +##
.##.          +##.
.#####.          .#####.
--#####-          -#####-
```

===== TURN 6 (Dokumon Tanah)=====

SCHWINKKK, Dokumon Tanah menyerang Dokumon Api

DETAIL SERANGAN:
Dokumon Tanah mencoba untuk menyerang Dokumon Api dengan power sebesar 121 (-6%)
Dokumon Api menahan serangan dari Dokumon Tanah sebesar 44 (36%)
sehingga damage yang diterima oleh Dokumon Api sebesar 77

Name : Dokumon Api
ATK Power : 235
DEF Power : 37
HP : 0
Level : 3

Yahhh, Anda dikalahkan monster Dokumon Tanah. Jangan menyerah, coba lagi !!!

>>> ■

Gambar 7.27 : F09 - Battle

– tampilan saat player memenangkan pertarungan –

```
SCHWINKKK, Dokumon Api menyerang Dokumon Air

DETAIL SERANGAN:
Dokumon Api mencoba untuk menyerang Dokumon Air dengan power sebesar 214 (-8%)
Dokumon Air menahan serangan dari Dokumon Api sebesar 22 (10%)
sehingga damage yang diterima oleh Dokumon Air sebesar 192

Name      : Dokumon Air
ATK Power : 75
DEF Power : 36
HP        : 0
Level     : 2

Selamat, Anda berhasil mengalahkan monster Dokumon Air !!!
Total OC yang diperoleh: 16
>>> █
```

Gambar 7.28 : F09 - Battle

– tampilan saat player memberi perintah quit –

```
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: 3

Anda berhasil kabur dari BATTLE!
>>> █
```

7.10 Arena

Gambar 7.29 : F10 - Arena

– tampilan awal menu arena –

```
>>> ARENA
Selamat datang di Arena

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

1. Dokumon Api, level: 3
2. Dokumon Air, level: 2
3. Dokumon Tanah, level: 4
Pilih monster untuk bertarung:
█
```

Gambar 7.30 : F10 - Arena

– pesan error saat pilihan monster tidak valid –

```
===== MONSTER LIST =====
1. Dokumon Api, level: 3
2. Dokumon Air, level: 2
3. Dokumon Tanah, level: 4
Pilih monster untuk bertarung:
5
Pilihan nomor tidak tersedia!

Pilih monster untuk bertarung:
daspro
Masukkan harus bertipe Integer, coba lagi!
|
```

Gambar 7.31 : F10 - Arena
– tampilan setelah player memilih monster

```
3. Dokumon Tanah, level: 4
Pilih monster untuk bertarung:
3

. +###+. . +###-
-#- .+. . #+ -#
+#+- . #- -#- +#
. ##+#+#. . +####+. . +#+#+#.
. - . +#####-. . -#####
# ##### # #####
. +- . #####-# ##-+### . -#.
##. +- . #- ##+ .#. . #. +### -#- . +- . ##
-#-.-+.#+ -### .+ +. +### -#-.-+.#+
-###+ +###. . ###- +###-
-### . ### # ##- # ##
+#+ -### # ##- -#+
-#- . +### +###+. -#+
+#####+
+## # ##
+### .+ +. +###+
###+. .#. . # ##
+###-+## # ##-+#####
. ##- ##### # ##.
+## -###+. . +####- +##
+## +#####-# ##+ ##+
+#+ -#####+ +##
. ##+ +##.
. #####. . #####.
-+#####-
```

RAWRRR, AGENT b mengeluarkan monster Dokumon Tanah

Name	:	Dokumon Tanah
ATK Power	:	130
DEF Power	:	65
HP	:	1300
Level	:	4

```
===== STAGE 1 =====
```



Gambar 7.32 : F10 - Arena

– tampilan saat player memenangkan stage 1 –



Gambar 7.33 : F10 - Arena
– tampilan saat player memenangkan stage 2 –

Gambar 7.34 : F10 - Arena
– tampilan saat player memenangkan stage 3 –

Gambar 7.35 : F10 - Arena
– tampilan saat player memenangkan stage 4 –

```
Selamat, Anda berhasil mengalahkan monster Dokumon Api !!!  
STAGE CLEARED! Anda akan mendapatkan 150 OC pada sesi ini !  
===== STAGE 5 =====
```

Gambar 7.36 : F10 - Arena
– tampilan saat player memenangkan semua stage di arena –

```
Selamat, Anda berhasil mengalahkan monster Dokumon Tanah !!!  
STAGE CLEARED! Anda akan mendapatkan 200 OC pada sesi ini !  
Selamat! Anda berhasil menyelesaikan seluruh stage Arena !!!  
===== STATS ======  
Total hadiah      : 200 OC  
Jumlah stage     : 5  
Damage diberikan: 2326  
Damage diterima  : 1746  
>>> █
```

Gambar 7.37 : F10 - Arena
– tampilan saat player kalah di arena –

```
===== STATS ======  
Total hadiah      : 50 OC  
Jumlah stage     : 2  
Damage diberikan: 531  
Damage diterima  : 162  
>>> █
```

Gambar 7.38 : F10 - Arena
– tampilan saat player kabur dari arena –

```

===== TURN 1 (Dokumon Api) =====
1. Attack
2. Use Potion
3. Quit
Pilihan Perintah: 3

GAME OVER! Anda mengakhiri sesi latihan!

===== STATS =====
Total hadiah    : 50 OC
Jumlah stage    : 2
Damage diberikan: 204
Damage diterima : 110

>>> █

```

7.11 Shop & Currency

Gambar 7.39 : F11 - Shop & Currency
– melihat potion –

```

>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
| ID | type      | stock | price |
| 1  | strength   | 199   | 9999  |
| 2  | resilience | 10    | 100   |
| 3  | healing    | 100   | 1000  |

```

Gambar 7.40 : F11 - Shop & Currency
– tampilan potion saat OC tidak cukup –

```

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 1000.

>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 1
>>> Masukkan jumlah: 1
OC-mu tidak cukup.

```

Gambar 7.41 : F11 - Shop & Currency
– tampilan saat potion berhasil dibeli –

```

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 1000.

>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 2
>>> Masukkan jumlah: 5
Berhasil membeli item: 5 Potion of resilience. Item sudah masuk ke inventory-mu!

```

Gambar 7.42 : F11 - Shop & Currency
– melihat monster yang ada di shop –

```

>>> SHOP
Irasshaimase! Selamat datang di SHOP!!

>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
| id | type          | atk_power | def_power | hp    | stock | price |
| 0  | Dokumon Api   | 196       | 30        | 135   | 5     | 198   |
| 1  | Dokumon Air   | 69        | 33        | 69    | 10    | 1000  |
| 2  | Dokumon Tanah | 100       | 50        | 1000  | 6     | 202   |

```

Gambar 7.43 : F11 - Shop & Currency
– tampilan saat monster yang dibeli sudah terdapat di inventory –

```

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 992.

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 0
Monster Dokumon Api sudah ada dalam inventory-mu! Pembelian dibatalkan.

```

Gambar 7.44 : F11 - Shop & Currency
– tampilan saat monster yang dibeli belum terdapat di inventory –

```

>>> SHOP
Irasshaimase! Selamat datang di SHOP!!

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 1000.

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 0
Berhasil membeli item: Dokumon Api. Item sudah masuk ke inventory-mu!

```

Gambar 7.45 : F11 - Shop & Currency
– tampilan saat OC tidak cukup untuk membeli –

```

>>> SHOP
Irasshaimase! Selamat datang di SHOP!!

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 190.

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 0
OC-mu tidak cukup.

```

Gambar 7.46 : F11 - Shop & Currency
– tampilan saat keluar dari fungsi ini –

```
>>> Pilih aksi (lihat/beli/keluar): keluar  
Mr. Yanto bilang makasih, belanja lagi ya nanti :)
```

7.12 Laboratory

Gambar 7.47 : F12 - Laboratory
– jika OC mencukupi –

```
>>> LABORATORY  
Selamat datang di Lab Dokter Asep !!!  
  
===== MONSTER LIST =====  
1. Dokumon Air (Level: 2)  
2. Dokumon Tanah (Level: 3)  
3. Dokumon Api (Level: 1)  
  
===== UPGRADE PRICE =====  
1. Level 1 -> Level 2: 300 OC  
2. Level 2 -> Level 3: 500 OC  
3. Level 3 -> Level 4: 800 OC  
4. Level 4 -> Level 5: 1000 OC  
>>> Pilih monster: 1  
  
Dokumon Air akan di-upgrade ke level 3.  
Harga untuk melakukan upgrade Dokumon Air adalah 500 OC.  
>>> Lanjutkan upgrade (Y/N): y  
  
Selamat, Dokumon Air berhasil di-upgrade ke level 3 !
```

Gambar 7.48 : F12 - Laboratory
– jika OC tidak mencukupi –

```
>>> LABORATORY
Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====
1. Dokumon Air (Level: 3)
2. Dokumon Tanah (Level: 3)
3. Dokumon Api (Level: 1)

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster: 1

Dokumon Air akan di-upgrade ke level 4.
Harga untuk melakukan upgrade Dokumon Air adalah 800 OC.
>>> Lanjutkan upgrade (Y/N): y

OC tidak cukup
```

Gambar 7.49 : F12 - Laboratory

– jika level monster yang akan diupgrade sudah maksimum –

```
>>> LABORATORY
Selamat datang di Lab Dokter Asep !!!

===== MONSTER LIST =====
1. Dokumon Air (Level: 2)
2. Dokumon Tanah (Level: 3)
3. Dokumon Api (Level: 5)

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster: 3
Maaf, monster yang Anda pilih sudah memiliki level maksimum
```

7.13 Shop Management

Gambar 7.50 : F13 - Shop Management
– tampilan awal –

```

>>> SHOP
Irasshaimase! Selamat datang kembali, Mr. Admin!

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): L1h4t
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): a
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): █

```

Gambar 7.51 : F13 - Shop Management
– tampilan ketika memilih monster –

```

>>> Mau lihat apa? (monster/potion): mastah
>>> Mau lihat apa? (monster/potion): monster
+---+-----+-----+-----+-----+-----+
| id | type      | atk_power | def_power | hp     | stock   | price   |
+---+-----+-----+-----+-----+-----+
| 0  | Dokumon Api | 196       | 30        | 135    | 5        | 198     |
| 1  | Dokumon Air  | 69        | 33        | 69     | 10       | 1000    |
| 2  | Dokumon Tanah | 100       | 50        | 1000   | 6        | 202     |

```

7.14 Monster Management

Gambar 7.52 : F14 - Monster Management
– menampilkan data monster beserta atributnya –

```

>>> MONSTER
SELAMAT DATANG DI DATABASE PARA MONSTER!!!
1. Tampilkan semua Monster
2. Tambah Monster baru
>>> Pilih aksi: 1
+---+-----+-----+-----+-----+
| id | type      | atk_power | def_power | hp     |
+---+-----+-----+-----+-----+
| 0  | Dokumon Api | 196       | 30        | 135    |
| 1  | Dokumon Air  | 69        | 33        | 69     |
| 2  | Dokumon Tanah | 100       | 50        | 1000   |

```

Gambar 7.53: F14 - Monster Management
– menampilkan penambahan monster baru beserta atributnya –

```
>>> MONSTER
SELAMAT DATANG DI DATABASE PARA MONSTER!!!
1. Tampilkan semua Monster
2. Tambah Monster baru
>>> Pilih aksi: 2
Memulai pembuatan monster baru

>>> Masukkan Type / Nama: Godzilla
>>> Masukkan ATK Power: 1000
>>> Masukkan DEF Power (0-50): 51
DEF Power harus bernilai 0-50, coba lagi!

>>> Masukkan DEF Power (0-50): 50
>>> Masukkan HP: 500
Monster baru berhasil dibuat!
Type: Godzilla
ATK Power: 1000
DEF Power: 50
HP: 500
Tambahkan Monster ke database (Y/N): y
Monster baru telah ditambahkan!
```

7.15 Load

Gambar 7.54 : F15 - Load

– Load berhasil –

```
$ python main.py testing
Folder testing ditemukan!

Loading...
Ketik HELP untuk melihat command yang tersedia!
```

Gambar 7.55 : F15 - Load

– load tanpa memasukkan nama folder –

```
$ python main.py
Masukkan nama folder Anda.
usage: python main.py <nama_folder>
```

Gambar 7.56 : F15 - Load

– Load ketika folder tidak ditemukan –

```
$ python main.py save123
Folder save123 tidak ditemukan, silakan ulangi!
```

7.16 Save

Gambar 7.57 : F16 - Save
– save ketika folder sudah dibuat –

```
>>> SAVE
Masukkan nama folder: testdoang

Saving...

Berhasil menyimpan data di folder data/testdoang!
```

Gambar 7.58 : F15 - Load
– save ketika folder belum terbuat –

```
>>> SAVE
Masukkan nama folder: belumada

Saving...

Membuat folder data/belumada...
Berhasil menyimpan data di folder data/belumada!
```

7.17 Exit

Gambar 7.59 : F17 - Exit
– exit dengan skema validasi –

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

– lanjutan dari skema di atas diakhiri dengan save folder/file yang sudah diubah –

```
>>> EXIT
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) Y
Masukkan nama folder: fathur

Saving...

Membuat folder data/fathur...
Berhasil menyimpan data di folder data/fathur!
```

VII. Lampiran

8.1 Link Lampiran

Link Repository:

[if1210-2024-tubes-k10-b/src at main · Labpro-21/if1210-2024-tubes-k10-b](https://github.com/Labpro-21/if1210-2024-tubes-k10-b)

8.2 Helper

Helper ini dibuat sebagai inisiatif untuk memudahkan satu sama lainnya dalam membuat program dalam tugas besar ini.

8.2.1 Cipher

```
def encrypt(s: str, key: int) -> str:
    text: str = ""
    for i in range(len(s)):
        if ord('A') <= ord(s[i]) <= ord('Z'):
            text += chr(((ord(s[i]) - ord('A')) + key) % 26) + ord('A'))
        elif ord('a') <= ord(s[i]) <= ord('z'):
            text += chr(((ord(s[i]) - ord('a')) + key) % 26) + ord('a'))
        else:
            text += s[i]
    return text
```

```
def decrypt(s: str, key: int) -> str:
    text: str = ""
    for i in range(len(s)):
        if ord('A') <= ord(s[i]) <= ord('Z'):
            text += chr(((ord(s[i]) - ord('A')) - key) % 26) + ord('A'))
        elif ord('a') <= ord(s[i]) <= ord('z'):
            text += chr(((ord(s[i]) - ord('a')) - key) % 26) + ord('a'))
        else:
            text += s[i]
    return text
```

8.2.2 IsType

```
def is_number(s: str) -> bool:  
    for i in s:  
        if not (ord('0') <= ord(i) <= ord('9')):  
            return False  
    return True and len(s) > 0
```

8.2.3 ListManipulation

```
from typing import List, TextIO, Dict  
from .Splitter import splitter  
import os  
  
Matrix = List[List[str]]  
  
  
def table_print(data: Matrix) -> None:  
    """  
    Prosedur ini membutuhkan parameter list 2 dimensi yang nantinya  
    akan diproses menjadi sebuah tabel yang rapih dengan mengeluarkan  
    output di terminal.  
    """  
    n: int = len(data)  
    m: int = len(data[0])  
    table_length: List[int] = [0 for _ in range(m)]  
    for i in range(n):  
        for j in range(m):  
            table_length[j] = max(table_length[j], len(data[i][j]))  
  
    for i in range(n):  
        for j in range(m):  
            element: str = data[i][j]  
            print("| %s " % (element), end="")  
            for k in range(table_length[j]-len(element)):  
                print(end=" ")  
        print("|")
```

```
def to_list(data: List[str]) -> List[List[str]]:
    """
        Fungsi ini digunakan untuk memproses list 1 dimensi menjadi 2 dimensi

        Contoh input: ["ID;Type;Stock", "0;Ayam;1"]
        Contoh output: [["ID","Type","Stock"], ["0","Ayam","1"]]
    """

    row: int = len(data)
    col: int = len(splitter(data[0]))
    arr: List[List[str]] = [["" for _ in range(col)] for _ in range(row)]
    for i in range(row):
        tmp: List[str] = splitter(data[i])
        for j in range(col):
            arr[i][j] = tmp[j]
    return arr

def bubble_sort(data: List[str]) -> None:
    """
        Prosedur bubble_sort digunakan untuk mengurutkan list menggunakan
        algoritma bubble sort.

        contoh penggunaan:
        >>> arr = [1,2,5,4,3]
        >>> bubble_sort(arr)
        >>> print(arr)
        [1,2,3,4,5]
    """

    n: int = len(data)
    for i in range(n):
        for j in range(0, n-i-1):
            if data[j] > data[j+1]:
                data[j], data[j+1] = data[j+1], data[j]
```

```
def join(data: List[str]) -> str:
    txt: str = ""
    col: int = len(data)
    for i in range(col):
        txt += str(data[i])
        if i != col-1:
            txt += ';'
    return txt

def readlines(path: str) -> List[str]:
    data: TextIO = open(path, 'r')
    n: int = 0
    for _ in data:
        n += 1
    data.close()
    data: TextIO = open(path, 'r')
    arr: List[str] = [""] * n
    idx: int = 0
    for i in data:
        arr[idx] = i
        idx += 1
    data.close()
    return arr

def read_all(user_data: Dict[str, Matrix], folder_name: str) -> None:
    for file_name in os.listdir("./data/%s" % (folder_name)):
        user_data[file_name] = to_list(
            readlines("./data/%s/%s" % (folder_name, file_name)))
```

8.2.4 ReadPeta

```
from typing import List, TextIO, Tuple

Matrix = List[List[str]]


def get_peta(folder_name: str) -> Tuple[int, int, Matrix]:
    file: TextIO = open("./data/%s/peta.txt" % (folder_name))
    arr: List[str] = []
    for i in file:
        arr.append(i)
    file.close()
    n: int = int(arr[0])-2
    m: int = int(arr[1])-1
    peta: Matrix = [["" for _ in range(m)] for _ in range(n)]
    for i in range(2, 2+n):
        for j in range(m):
            peta[i-2][j] = arr[i][j]
    return (n, m, peta)
```

```
def mapify(peta: Matrix) -> None:
    row: int = len(peta)
    col: int = len(peta[0])
    for i in range(col+2):
        print(end="* ")
    print()
    for i in range(row):
        for j in range(col):
            if j == 0:
                print(end="* ")
            print(peta[i][j] if peta[i][j] != '#' else " ", end=" ")
            if j == col-1:
                print(end="* ")
        print()
    for i in range(col+2):
        print(end="* ")
```

8.2.5 Readline

```
from typing import List, TextIO

def readlines(path: str) -> List[str]:
    data: TextIO = open(path, 'r')
    n: int = 0
    for _ in data:
        n += 1
    data.close()
    data: TextIO = open(path, 'r')
    arr: List[str] = [""] * n
    idx: int = 0
    for i in data:
        arr[idx] = i
        idx += 1
    data.close()
    return arr
```

8.2.6 Splitter

```
from typing import List

def count_delimiter(data: str) -> int:
    count: int = 0
    for i in data:
        if i == ';':
            count += 1
    return count

def result(n: int, arr: List[str], idx: int, data: str, pos: int) -> None:
    if pos == n+1:
        return
    elem: str = ""
    while data[idx] != ';' and data[idx] != '\n':
        elem += data[idx]
        idx += 1
    arr[pos] = elem
    result(n, arr, idx+1, data, pos+1)

def splitter(data: str) -> List[str]:
    n: int = count_delimiter(data)
    arr: List[str] = [" " for _ in range(n+1)]
    result(n, arr, 0, data+";", 0)
    return arr
```

8.3 Referensi

[Linear Congruential Generator](#). Wikipedia.
[Caesar Cipher](#). Wikipedia.

8.4 Asistensi

Form MoM Asistensi Tugas Besar IF1210/Dasar Pemrograman Sem. 2 2021/2022

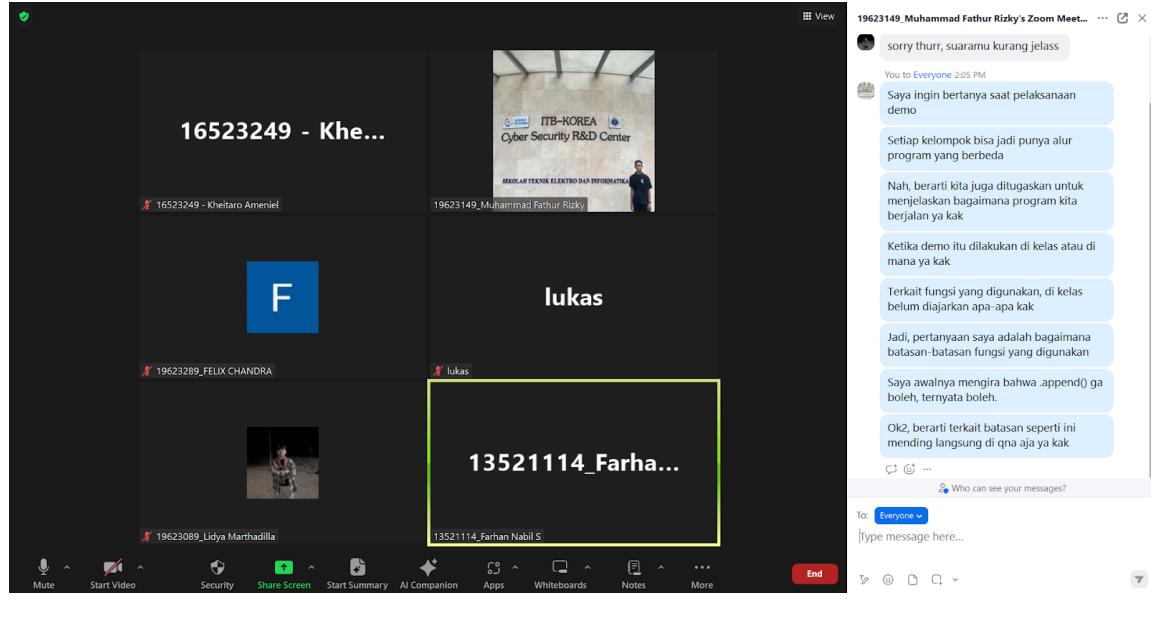
Nomor Asistensi	:	1														
No. Kelompok/Kelas	:	B/10														
Tanggal asistensi	:	1 Mei 2024														
Anggota kelompok		<table border="1"><thead><tr><th colspan="2">NIM / Nama (Hanya yang Hadir)</th></tr></thead><tbody><tr><td>1</td><td>19623149/Muhammad Fathur Rizky</td></tr><tr><td>2</td><td>19623089/Lidya Marthadilla</td></tr><tr><td>3</td><td>19623069/Lukas Raja</td></tr><tr><td>4</td><td>19623289/Felix Chandra</td></tr><tr><td>5</td><td>16523249/Kheitaro Ameniel Gugun M</td></tr><tr><td>6</td><td></td></tr></tbody></table>	NIM / Nama (Hanya yang Hadir)		1	19623149/Muhammad Fathur Rizky	2	19623089/Lidya Marthadilla	3	19623069/Lukas Raja	4	19623289/Felix Chandra	5	16523249/Kheitaro Ameniel Gugun M	6	
NIM / Nama (Hanya yang Hadir)																
1	19623149/Muhammad Fathur Rizky															
2	19623089/Lidya Marthadilla															
3	19623069/Lukas Raja															
4	19623289/Felix Chandra															
5	16523249/Kheitaro Ameniel Gugun M															
6																
Asisten pembimbing		<table border="1"><thead><tr><th colspan="2">NIM / Nama</th></tr></thead><tbody><tr><td colspan="2">13521114 / Farhan Nabil Suryono</td></tr></tbody></table>	NIM / Nama		13521114 / Farhan Nabil Suryono											
NIM / Nama																
13521114 / Farhan Nabil Suryono																

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">Diskusi pertama membahas perihal F15 - Save. Kami membahas bagaimana mekanisme penyimpanan file yang ada-apa saja file yang disimpan dalam suatu folder.Diskusi kedua membahas perihal pelaksanaan demo. Kami membahas bagaimana mekanisme pelaksanaan demo.Diskusi ketiga membahas terkait fungsi yang digunakan belum diajarkan di kelas.Diskusi keempat membahas terkait batasan fungsi yang digunakan.
Tindak Lanjut
<ul style="list-style-type: none">Tindak lanjut dari diskusi pertama, mekanisme penyimpanan file disarankan untuk menyimpan 6 file csv dalam suatu folder yang daripada membedakan akses admin dan agent.Tindak lanjut dari diskusi kedua, dijelaskan bahwa ketika pelaksanaan demo nanti kami harus menjalankan dan menjelaskan apa yang telah kami kerjakan. Untuk mekanisme pelaksanaan demo akan diumumkan di kemudian hari.Tindak lanjut dari diskusi ketiga, dijelaskan bahwa fungsi atau materi yang digunakan memang belum dijelaskan di kelas, dan nantinya ketika di jurusan pun akan banyak mengeksplor sendiri.

- Tindak lanjut dari diskusi keempat, semua terkait batasan-batasan fungsi yang digunakan dapat ditanyakan di QnA agar kelompok lain juga tau informasinya.

Dokumentasi



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

Nomor Asistensi : 2
No. Kelompok/Kelas : B/10
Tanggal asistensi : 12 Mei 2024

Anggota kelompok	
1	NIM / Nama (Hanya yang Hadir)
2	19623149/Muhammad Fathur Rizky
3	19623089/Lidya Marthadilla
4	19623289/Felix Chandra
5	19623069/Lukas Raja
6	16523249/Kheitaro Ameniel Gugun M
Asisten pembimbing	NIM / Nama
	13521114 / Farhan Nabil Suryono

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">• Diskusi pertama, kami meminta saran terkait User Interface untuk di terminal, dan bertanya bagaimana penilaian kreativitas untuk User Interface.• Diskusi kedua, Kami membahas terkait monster shop dan monster csv. Dan Kami menanyakan menanyakan apakah secara tidak langsung mengenalkan konsep memory reference?• Diskusi ketiga, Kami bertanya berapa nilai maksimal dari tugas besar ini? dan bagaimana penilaian orang yang nge push di github (kontribusi individual) berdasarkan commit atau file yang diurus?• Diskusi keempat, Kami menanyakan terkait penulisan features apakah ada referensi?• Diskusi kelima, Kami menanyakan bagaimana cara asistensi yang benar?• Diskusi keenam, Kami menanyakan bagaimana pendekatan antar anggota tim atau cara me manage penugasan yang baik dan benar? lalu, pertanyaan selanjutnya yaitu apakah setiap tugas besar harus ada yg jadi ketua?

Tindak Lanjut
<ul style="list-style-type: none">• Tindakan dari diskusi pertama, untuk User Interface jika basic (seperti contoh yang ada di spesifikasi tugas) juga tidak masalah karena memang tidak dituntut untuk bagus. Penilaian untuk User Interface yaitu untuk menambah nilai, akan tetapi bobot nilai nya tidak besar.• Tindak lanjut dari diskusi kedua, jawaban yang diberikan asisten terkait pertanyaan yang diajukan yaitu overall sudah baik, hanya diperlukan penambahan minor.• Tindak lanjut dari diskusi ketiga, untuk penilaian tugas besar ini akan disesuaikan di akhir. Lalu untuk penilaian kontribusi akan di data bagaimana keseluruhan kontribusinya dan nantinya akan ada nilai kelompok maupun individu.• Tindak lanjut dari diskusi keempat, tidak ada referensi. Jika bisa di features ditulis siapa pembagian tugasnya kerjain. Dan terkait deskripsi, deskripsi jika tidak dikerjakan juga tidak apa apa, tapi sebaiknya diisi saja karena akan di public dan agar terlihat lebih rapi di githubnya.• Tindak lanjut dari diskusi kelima, selama tujuan dari asistensi tercapai, maka asistensi sudah benar.

- Tindak lanjut dari diskusi keenam, belajar seiring berjalannya waktu, karena tipe org berbeda beda. Untuk setiap tugas besar seperti ini sebenarnya tidak harus memiliki ketua (karena semua setara), hanya saja terkadang dibutuhkan orang yang bisa menjadi penengah dan memimpin terkait masalah yang dihadapi atau hanya sekedar diskusi.

Dokumentasi

