

**PROGRAM GAME PURRY
LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN
KELAS 05**

Kelompok C:

Muhammad Raihan Nazhim Oktana (19623215)
Athian Nugraha Muarajuang (19623055)
Muhammad Alfansya (19623065)
Dede Firman (16523245)
Aleta Edna Jessalyn (16523135)



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

Pernyataan Kelompok

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

Yang mengeluarkan pernyataan,

Muhammad Raihan Nazhim Oktana (19623215)

Athian Nugraha Muarajuang (19623055)

Muhammad Alfansya (19623065)

Dede Firman (16523245)

Aleta Edna Jessalyn (16523135)

Daftar Isi

Pernyataan Kelompok.....	2
Daftar Isi.....	3
Daftar Tabel.....	7
Daftar Gambar.....	8
Deskripsi Persoalan.....	9
Daftar Pembagian Kerja Anggota Kelompok.....	11
Checklist Hasil Pengerjaan Tugas Besar.....	12
Desain Perintah.....	13
1. F00 - Random Number Generator.....	13
2. F01 – Register.....	13
3. F02 – Login.....	13
4. F03 – Logout.....	14
5. F04 - Menu & Help.....	14
6. F05 - Monster.....	16
7. F06 - Potion.....	16
8. F07 - Inventory.....	17
9. F08 - Battle.....	19
10. F09 - Arena.....	22
11. F10 - Shop & Currency.....	26
12. F11 – Laboratory.....	27
13. F12 - Shop Management.....	28
14. F13 - Monster Management.....	28
15. F14 - Load.....	31
16. F15 - Save.....	31
17. F16 - Exit.....	31
Desain Kamus Data.....	33
1. F00 - Random Number Generator.....	33
2. F01 – Register.....	33
3. F02 – Login.....	33
4. F03 – Logout.....	33
5. F04 - Menu & Help.....	34
6. F05 - Monster.....	34
7. F06 - Potion.....	34
8. F07 - Inventory.....	34
9. F08 - Battle.....	34
10. F09 - Arena.....	36

11. F10 - Shop & Currency.....	37
12. F11 - Laboratory.....	38
13. F12 - Shop Management.....	38
14. F13 - Monster Management.....	39
15. F14 - Load.....	40
16. F15 - Save.....	41
17. F16 - Exit.....	41
Desain Dekomposisi Algoritmik Dan Fungsional Program.....	42
1. F00 - Random Number Generator.....	42
2. F01 – Register.....	42
3. F02 – Login.....	42
4. F03 – Logout.....	42
5. F04 - Menu & Help.....	42
6. F05 - Monster.....	43
7. F06 - Potion.....	43
8. F07 - Inventory.....	43
9. F08 - Battle.....	43
10. F09 - Arena.....	43
11. F10 - Shop & Currency.....	44
12. F11 - Laboratory.....	44
13. F12 - Shop Management.....	44
14. F13 - Monster Management.....	45
15. F14 - Load.....	45
16. F15 - Save.....	45
17. F16 - Exit.....	45
Spesifikasi Fungsi-Fungsi Dalam Program.....	46
1. F00 - Random Number Generator.....	46
2. F01 – Register.....	47
3. F02 – Login.....	49
4. F03 – Logout.....	51
5. F04 - Menu & Help.....	51
6. F05 - Monster.....	54
7. F06 - Potion.....	56
8. F07 - Inventory.....	57
9. F08 - Battle.....	60
10. F09 - Arena.....	69
11. F10 - Shop & Currency.....	82
12. F11 – Laboratory.....	87

13. F12 - Shop Management.....	90
14. F13 - Monster Management.....	100
15. F14 - Load.....	107
16. F15 - Save.....	110
17. F16 - Exit.....	113
Lampiran Hasil Pengujian Program.....	114
1. F00 - Random Number Generator.....	114
2. F01 – Register.....	114
3. F02 – Login.....	115
4. F03 – Logout.....	116
5. F04 - Menu & Help.....	116
6. F05 - Monster.....	118
7. F06 - Potion.....	118
8. F07 - Inventory.....	119
9. F08 - Battle.....	120
10. F09 - Arena.....	125
11. F10 - Shop & Currency.....	133
12. F11 - Laboratory.....	134
13. F12 - Shop Management.....	137
14. F13 - Monster Management.....	140
15. F14 - Load.....	143
16. F15 - Save.....	144
17. F16 - Exit.....	145
Lampiran Form Asistensi.....	147
Daftar Referensi.....	150

Daftar Tabel

Tabel 1.1 Daftar Pembagian Kerja Anggota Kelompok.....	8
Tabel 2.1 Daftar Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif..	9
Tabel 6.1 Spesifikasi fungsi dalam program Random Number Generator (F00).....	43
Tabel 6.2 Spesifikasi fungsi dalam program Register (F01).....	44
Tabel 6.3 Spesifikasi fungsi dalam program Login (F02).....	46
Tabel 6.4 Spesifikasi fungsi dalam program Logout (F03).....	48
Tabel 6.5 Spesifikasi fungsi dalam program Menu dan Help (F04).....	48
Tabel 6.6 Spesifikasi fungsi dalam program Monster (F05).....	51
Tabel 6.7 Spesifikasi fungsi dalam program Potion (F06).....	53
Tabel 6.8 Spesifikasi fungsi dalam program Inventory (F07).....	54
Tabel 6.9 Spesifikasi fungsi dalam program Battle (F08).....	57
Tabel 6.10 Spesifikasi fungsi dalam program Arena (F09).....	66
Tabel 6.11 Spesifikasi fungsi dalam program Shop & Currency (F10).....	79
Tabel 6.12 Spesifikasi fungsi dalam program Laboratory(F11).....	84
Tabel 6.13 Spesifikasi fungsi dalam program Shop Management(F12).....	87
Tabel 6.14 Spesifikasi fungsi dalam program Monster Management (F13).....	97
Tabel 6.15 Spesifikasi fungsi dalam program Load (F14).....	104
Tabel 6.15 Spesifikasi fungsi dalam program Save (F15).....	107
Tabel 6.17 Spesifikasi fungsi dalam program Battle (F16).....	110

Daftar Gambar

Gambar 7.1.1 Hasil generate RNG.....	124
Gambar 7.2.1 Hasil register dengan input yang benar.....	124
Gambar 7.2.2 Hasil jika username sudah dipakai atau ada karakter terlarang.....	125
Gambar 7.2.3 Hasil jika register namun sedang login.....	125
Gambar 7.3.1 Hasil jika username dan password benar.....	125
Gambar 7.3.2 Hasil jika username tidak terdaftar.....	126
Gambar 7.3.3 Hasil jika password salah.....	126
Gambar 7.3.4 Hasil jika username sudah login.....	126
Gambar 7.4.1 Hasil jika parameter check_login false.....	126
Gambar 7.4.2 Hasil jika parameter check_login True.....	126
Gambar 7.5.1 Tampilan help saat belum login.....	127
Gambar 7.5.2 Tampilan help saat login sebagai admin.....	127
Gambar 7.5.3 Tampilan help saat login sebagai admin.....	128
Gambar 7.5.4 Tampilan menu saat belum login.....	128
Gambar 7.5.5 Tampilan menu saat login sebagai admin.....	129
Gambar 7.5.5 Tampilan menu saat login sebagai agent.....	129
Gambar 7.6.1 Tampilan awal fungsi Monster.....	130
Gambar 7.6.2 Hasil saat validasi keluar fitur.....	130
Gambar 7.6.3 Hasil saat sudah keluar.....	130
Gambar 7.7.1 Tampilan menu potion.....	130
Gambar 7.7.2 Tampilan validasi keluar menu potion.....	130
Gambar 7.7.3 Tampilan menu potion.....	131
Gambar 7.8.1 Tampilan awal inventory.....	131
Gambar 7.8.2 Tampilan validasi id.....	131
Gambar 7.8.3 Hasil saat input id di inventory.....	131
Gambar 7.8.4 Tampilan item potion di inventory.....	132
Gambar 7.8.5 Tampilan item monster ball.....	132
Gambar 7.8.6 Tampilan validasi keluar dari inventory.....	132
Gambar 7.9.1 Tampilan awal battle.....	132
Gambar 7.9.2 Tampilan saat memilih monster.....	133
Gambar 7.9.3 Tampilan saat menggunakan potion strength.....	134
Gambar 7.9.4 Tampilan saat menggunakan potion resilience.....	134
Gambar 7.9.5 Tampilan saat menggunakan potion healing.....	135
Gambar 7.9.6 Hasil saat menyerang.....	135

Gambar 7.9.7 Tampilan saat memilih quit dari battle.....	136
Gambar 7.9.8 Tampilan saat memilih potion yang sudah diminum sebelumnya.....	136
Gambar 7.9.9 Tampilan saat potion habis.....	137
Gambar 7.9.10 Tampilan saat menang.....	137
Gambar 7.9.11 Tampilan saat kalah.....	137
Gambar 7.10.1 Tampilan memilih monster di arena.....	138
Gambar 7.10.2 Tampilan saat memasuki stage.....	139
Gambar 7.10.3 Tampilan menang stage 1 dan memasuki stage 2.....	140
Gambar 7.10.4 Tampilan menang stage 2 dan memasuki stage 3.....	141
Gambar 7.10.5 Tampilan menang stage 3 dan memasuki stage 4.....	142
Gambar 7.10.6 Tampilan menang stage 4 dan memasuki stage 5.....	143
Gambar 7.10.7 Tampilan menang 5.....	143
Gambar 7.10.8 Memenangkan semua stage.....	144
Gambar 7.10.9 Tampilan saat kabur dari permainan.....	144
Gambar 7.10.10 Tampilan saat kalah dalam suatu stage.....	145
Gambar 7.11.1 Tampilan awal yanto shop.....	145
Gambar 7.11.2 Tampilan jika memilih lihat.....	146
Gambar 7.11.3 Tampilan jika memilih lihat potion.....	146
Gambar 7.11.4 Tampilan jika memilih beli suatu monster.....	146
Gambar 7.11.5 Tampilan jika membeli potion.....	146
Gambar 7.11.6 Tampilan saat keluar yanto shop.....	146
Gambar 7.12.1 Tampilan awal laboratory Dokter Asep.....	147
Gambar 7.12.1 Tampilan saat memilih monster di list.....	148
Gambar 7.12.3 Tampilan jika berhasil mengupgrade.....	148
Gambar 7.12.4 Tampilan jika OC tidak cukup.....	148
Gambar 7.12.5 Tampilan saat membatalkan upgrade.....	149
Gambar 7.12.6 Tampilan saat monster yang dipilih sudah maksimum.....	149
Gambar 7.12.7 Tampilan saat monster yang dipilih tidak ada di list.....	149
Gambar 7.13.1 Tampilan awal Shop Management.....	150
Gambar 7.13.2 Tampilan saat melihat monster.....	150
Gambar 7.13.4 Tampilan saat menambah monster.....	150
Gambar 7.13.5 Tampilan saat menambah potion.....	151
Gambar 7.13.6 Tampilan saat mengubah monster.....	151
Gambar 7.13.7 Tampilan saat menghapus monster.....	152
Gambar 7.13.9 Tampilan saat keluar.....	152
Gambar 7.14.1 Tampilan saat memasuki Monster Management.....	153
Gambar 7.14.2 Tampilan saat aksi tidak tervalidasi.....	153
Gambar 7.14.3 Tampilan saat aksi 1.....	153

Gambar 7.14.4 Tampilan jika jawaban tidak tervalidasi.....	154
Gambar 7.14.5 Tampilan jika mau melanjutkan aktivitas.....	154
Gambar 7.14.6 Tampilan saat keluar.....	154
Gambar 7.14.7 Tampilan membuat monster baru.....	154
Gambar 7.14.8 Tampilan validasi input monster.....	154
Gambar 7.14.9 Tampilan monster tervalidasi.....	155
Gambar 7.14.10 Tampilan saat atk power monster tidak valid.....	155
Gambar 7.14.11 Tampilan saat atk power monster valid.....	155
Gambar 7.14.12 Tampilan saat def monster tidak valid.....	155
Gambar 7.14.13 Tampilan saat def monster valid.....	155
Gambar 7.14.14 Tampilan saat hp monster tidak valid.....	155
Gambar 7.14.15 Tampilan saat hp monster valid.....	156
Gambar 7.14.16 Tampilan saat penambahan tidak valid.....	156
Gambar 7.14.16 Tampilan saat menambahkan ke database.....	156
Gambar 7.14.17 Tampilan saat tidak ditambahkan ke database.....	156
Gambar 7.14.18 Tampilan saat mau melakukan hal lain.....	157
Gambar 7.14.19 Tampilan saat keluar.....	157
Gambar 7.15.1 Tampilan saat data ditemukan.....	157
Gambar 7.15.2 Tampilan saat data tidak ditemukan.....	157
Gambar 7.16.1 Tampilan saat menyimpan data.....	158
Gambar 7.15.2 Bukti data tersimpan.....	159
Gambar 7.16.1 Tampilan exit.....	160
Gambar 8.1.1 Bukti Asistensi 1.....	163
Gambar 8.1.2 Bukti Asistensi 2.....	165

Deskripsi Persoalan

Untuk membantu Purry si Platypus dalam misinya menggagalkan rencana jahat Dr. Asep Spakbor, kelompok K05-C membentuk suatu program berbahasa python dalam melakukan misi pencarian serta pelatihan pada berbagai jenis monster. Hal ini dilakukan untuk melawan monster mengerikan yang diciptakan Dr. Asep Spakbor dan demi menyelamatkan kota Danville dari ancaman keamanan.

Dalam program ini, diberikan dua jenis fitur yang dapat diakses berdasarkan jenis role dari pengguna. Role agent adalah peran yang diperoleh pengguna ketika melakukan registrasi dengan akses pada fitur-fitur utama, seperti monster, potion, inventory, battle, arena, shop & currency, dan laboratory. Role admin adalah peran default data pada program yang dapat mengakses fitur spesial, yang tidak dapat diakses role agent, seperti shop management, juga monster management. Kedua role dapat mengakses fitur logout, menu & help, load, save, dan exit.

Proses registrasi yang dilakukan oleh agen akan memasukkan id unik, username, password pengguna, role, dan O.W.C.A coin bernilai nol pada file "user.csv". File yang sama pula akan diakses kembali ketika pengguna melakukan login baik sebagai agent maupun admin.

Agent yang telah melakukan login, dapat kemudian mengakses fitur berdasarkan perintah yang dimasukkan kepada program. Dalam prosesnya, file "monster.csv", "item_inventory.csv", "monster_inventory.csv", "item_shop.csv", dan "monster_shop.csv" dapat *dipanggil* sesuai kebutuhan fitur-fitur yang diakses. File "monster.csv" memiliki fungsi sebagai daftar id unik, tipe, kekuatan penyerangan, kekuatan bertahan, serta banyaknya nyawa dari tiap-tiap monster di program. File "item_inventory.csv" adalah file khusus untuk role agen yang berisikan daftar user_id kepemilikan inventory, tipe item, dan jumlah item dari tiap-tiap agen di program. Begitu juga dengan file "monster_inventory.csv" yang berisikan daftar user_id kepemilikan inventory, monster_id yang merepresentasikan monster, serta level monster yang dimiliki agen. File "item_shop.csv" dan "monster_shop.csv" merupakan daftar dari item dan monster yang ditampilkan pada shop dengan atribut, secara berturut, tipe item, jumlah stok item, harga tiap item, dan id dari monster, jumlah stok monster, serta harga tiap monster.

Dalam program, agen akan berperan sebagai agen O.W.C.A yang bertugas dalam mencari dan melatih monster sebagai persiapan melawan monster mengerikan milik Dr. Asep Spakbor. Fitur battle, arena, dan laboratory, berguna sebagai media dalam melakukan misi. Fitur shop & currency, berfungsi sebagai bantuan tambahan bagi agen untuk menyelesaikan misi. Sementara fitur inventory dapat membantu agen dalam mengetahui item dan monster yang dimiliki.

Pengguna yang melakukan login dengan role admin, dapat memanfaatkan fitur shop management dan monster management untuk menambah item ataupun monster dalam program. Dalam hal ini, role admin memberikan lebih banyak variasi dalam program yang dapat digunakan oleh agen.

Setelah menggunakan fitur, pengguna, baik dengan role admin maupun agen, dapat melakukan logout untuk keluar dari game. Selanjutnya, fitur save juga dapat

diakses untuk menyimpan semua aktivitas yang dilakukan pada program, sehingga dapat diakses kembali dengan fitur load. Kedua fitur ini mengatur penyimpanan data pada program. Terakhir, pengguna dapat melakukan exit untuk keluar sepenuhnya dari program.

Daftar Pembagian Kerja Anggota Kelompok

Tabel 1.1 Daftar Pembagian Kerja Anggota Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00 - Random Number Generator	function random number generator	19623065	19623065	19623065
F01 - Register	function register	19623215	19623215	19623215
F02 - Login	function login	16523245	16523245	16523245
F03 - Logout	function logout	16523135	16523135	16523135
F04 - Menu & Help	procedure menu & help	19623065	19623065	19623065
F05 - Monster	procedure monster	16523135	19623065 16523135	19623065 16523135
F06 - Potion	procedure potion	16523135	19623065	19623065 16523135
F07 - Inventory	procedure inventory	16523135	16523135	16523135
F08 - Battle	function battle	19623215	19623215	19623215
F09 - Arena	function arena	19623215	19623215	19623215
F10 - Shop & Currency	function shop	19623055	19623055	19623055
F11 - Laboratory	function laboratory	16523245	16523245	16523245
F12 - Shop Management	function manage shop	19623055	19623055	19623055
F13 - Monster Management	function monster mng	16523135	16523135	16523135
F14 - Load	procedure load	19623215	19623215	19623215
F15 - Save	procedure save	19623215	19623215	19623215
F16 - Exit	function exit	19623215	19623215	19623215
MainPY	-	19623065	19623065	19623065

Checklist Hasil Pengerjaan Tugas Besar

Tabel 2.1 Daftar Checklist Hasil Rancangan, Implementasi dan Testing Setiap Primitif

Fitur	Desain	Implementasi	Testing
F00 - Random Number Generator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F01 - Register	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F02 - Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F03 - Logout	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F04 - Menu & Help	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F05 - Monster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F06 - Potion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F07 - Inventory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F08 - Battle	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F09 - Arena	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F10 - Shop & Currency	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F11 - Laboratory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F12 - Shop Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F13 - Monster Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F14 - Load	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F15 - Save	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F16 - Exit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MainPY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Desain Perintah

1. F00 - Random Number Generator

Nama command : generate(range)

Masukkan :

memberikan parameter range
menerima input range

Keluaran :

mengeluarkan angka random dari range
jika diberi range [0, 11] maka angka yang keluar berkisar antara 0 hingga 10

2. F01 – Register

Nama command: register(check_login , username_login , user , monster , inv_item , inv_monster)

Masukkan :

Memberikan parameter check_login , username_login , user , monster , inv_item , dan inv_monster.

Menerima input username, password, dan starting_monster.

Keluaran :

➤ “>>> REGISTER”

Jika parameter check_login bernilai True, maka akan dicetak pesan :

➤ “Register gagal, Anda sudah login dengan username {username_login}, silakan lakukan "LOGOUT" sebelum melakukan register.”

Jika parameter check_login bernilai False, dan username yang dimasukkan ditemukan dalam data user, maka akan dicetak pesan :

➤ “Username {username_login} sudah terpakai, silakan gunakan username lain!”

Selama terdapat karakter yang dilarang dalam username yang dimasukkan, maka akan meminta input kembali didahului dengan mencetak pesan :

➤ “Username hanya boleh berisi alfabet, angka, underscore, dan strip!”

Jika username dan password telah valid, akan dicetak ke layar pilihan monster awal dengan format :

➤ “{i}. {monster[i][1]}”

Selama input starting_monster tidak valid, maka akan meminta input kembali didahului dengan mencetak pesan :

➤ “Monster pilihan Anda tidak tersedia, silakan pilih ulang!”

Jika parameter check_login bernilai True, maka akan dicetak pesan :

➤ “Selamat datang Agent {username}. Mari kita mengalahkan Dr. Asep Spakbor dengan {monster[starting_monster][1]}!”

Jika fungsi register telah selesai dijalankan, maka akan dikembalikan nilai check_login , username_login , user , monster , inv_item , dan inv_monster.

3. F02 – Login

Nama command: login(data_user, check_login, username_login):

Masukkan :

Memberikan parameter data_user, check_login, dan username_login

Menerima input username dan password

Keluaran :

Jika parameter check_login bernilai True, dan username terdefinisi akan dicetak pesan:

➢ **“Login gagal, anda sudah login menggunakan username tersebut!”**

Jika parameter check_login bernilai False dan username tidak terdaftar akan dicetak pesan

➢ **“Username tidak terdaftar!”**

Jika parameter check_login bernilai False dan username terdaftar dan password salah akan dicetak pesan

➢ **“Password salah!”**

Jika parameter check_login bernilai False, admin bernilai true , username terdaftar, dan password benar akan dicetak pesan:

➢ **“Selamat datang admin!”**

Jika parameter check_login bernilai False, admin bernilai False, username terdaftar, dan password benar akan dicetak pesan:

➢ **“Selamat datang Agent!”**

Jika user berhasil login, maka fungsi akan mengembalikan nilai check_login, username_login, admin, user_loggedin

4. F03 – Logout

Nama command: logout(check_login)

Masukkan:

Memberikan parameter **check_login**

Keluaran:

Jika parameter check_login bernilai True, maka check_login akan berganti nilai menjadi False dan mencetak pesan:

➢ **“Logout berhasil!”**

➢ **“Logout berhasil!”**

Jika parameter check_login bernilai False, maka akan dicetak pesan:

➢ **“Logout gagal!”**

➢ **“Anda belum login, silahkan login terlebih dahulu.”**

5. F04 - Menu & Help

Nama command : help(check_login, username, admin) dan menu(check_login, username, admin)

Masukkan :

Memberikan parameter chceck_login, username, admin

Menerima input perintah `help()`
Keluaran :
Jika user belum login (`check_login = False`) akan mengeluarkan tampilan :

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

1. Register - melakukan registrasi user baru
2. Login - melakukan login ke akun ter-register
3. Exit - Keluar dari permainan

FOOTNOTE

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

Jika user sudah login dan bukan admin akan mengeluarkan tampilan :

Halo Agent {username}. Kamu memanggil command `HELP`. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian.
Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

1. Logout - Keluar dari akun
2. Monster - Melihat owca-dex yang dimiliki oleh Agent
3. Potion - Melihat jenis - jenis potion beserta efeknya
4. Inventory - Melihat isi inventory milik Agent
5. Battle - Bertarung dengan monster secara acak secara 1v1
6. Arena - Memasuki arena bertarung yang berisi 5 stage
7. Shop - Tempat Agent untuk membeli monster dan potion
8. Laboratory - Tempat Agent untuk upgrade monster yang ada di inventory
9. Load - Memuat data yang tersimpan
10. Exit - Keluar dari permainan

FOOTNOTE

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

Jika user sudah login dan user adalah admin akan mengeluarkan tampilan :

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

1. Logout - Untuk melakukan registrasi user baru
2. Shop - Mengatur isi shop, baik menambah stok maupun mengurangi stok
3. Monster - mengatur monster dalam database
4. Load - Memuat data yang tersimpan
5. Exit - Keluar dari permainan

FOOTNOTE

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

6. F05 - Monster

Nama command: monster(listmon)

Masukkan:

Memberikan parameter **listmon**

Menerima masukan untuk keluar

Keluaran:

Mencetak pesan:

- “WELCOME TO OWCA-DEX UNTUK MELIHAT MONSTER YANG TERSEDIA, AGEN”
 - “{arr1}”
 - “Informasi tambahan owca-dex mengenai level:”
 - “Level 1 : Monster menggunakan Base Attribute”
 - “Level 2 - 5: Ketika sedang Battle, Base Attribute monster ditambah dengan Base Attribute dikalikan dengan $((Level - 1) * 10\%)$. Ketika monster naik level, Base Attribute tetap disimpan, kalkulasi atribut sesuai level dilakukan saat Battle.”
 - “Tips! Cek level monstermu pada inventory, gunakan command ‘inventory’!”

Apabila validasi keluar tidak terpenuhi, akan dicetak pesan:

- “Keluar untuk kembali ke menu:”

Apabila validasi keluar terpenuhi, user keluar dari prosedur.

7. F06 - Potion

Nama command: potion()

Masukkan:

- Menerima masukan untuk keluar

Keluaran:

Mencetak pesan:

➤ “**Potions**”

- “Potion | Effect”
- “Strength Potion | Meningkatkan ATK Power sebanyak 5% dari ATK Power”
- “Resilience Potion | Meningkatkan Def Power sebanyak 5% dari DEF Power”
- “Healing Potion | Mengisi darah sebanyak 25% dari Base HP, tidak akan mengisi lebih dari maks HP”
- “Tips! Gunakan potion yang tepat di saat yang tepat untuk melawan monster yang lebih kuat!”
- “Potion bisa dibeli di shop dengan OC.”

Apabila validasi keluar tidak terpenuhi, akan dicetak pesan:

➤ “**Keluar untuk kembali ke menu:**”

Apabila validasi keluar terpenuhi, user keluar dari prosedur.

8. F07 - Inventory

Nama command: inventory(username, listinvmonster, listuser, listmonster, listinvent)

Masukkan:

- Memberikan parameter **username, listinvmonster, listuser, listmonster, listinvent**
- Menerima masukan id dari daftar inventory

Keluaran:

- “===== INVENTORY LIST (User ID: {listuser[i][0]})
=====”
- “Jumlah O.W.C.A. Coin-mu sekarang {listuser[i][4]}.”

Jika user sudah memiliki monster pada daftar inventory, maka akan dicetak dengan format:

- “{a+1}. Monster (Name: {listmonster[k][1]}, Lvl: {listinvmonster[j][2]}, HP: {listmonster[k][4]})”

Jika user sudah memiliki item pada daftar inventory, maka akan dicetak dengan format:

- “{a+1}. Potion (Type: {listinvent[l][1]}, Qty: {listinvent[l][2]})”

Jika user sudah memiliki monster ball pada daftar inventory, maka akan dicetak dengan format:

- “{a+1}. Monster ball (Qty: {listinvent[l][2]})”

Untuk melihat detail tiap item pada daftar inventory yang sebelumnya dicetak, user memasukkan id sebagai command. Jika user memasukkan id yang merujuk kepada monster, maka akan dicetak informasi mengenai monster tersebut dengan format:

```
> "array_tumbal2[m][0]"  
> "Name : {array_tumbal2[m][2]}"  
> "ATK Power : {array_tumbal2[m][3]}"  
> "DEF Power : {array_tumbal2[m][4]}"  
> "HP : {array_tumbal2[m][5]}"  
> "Level : {array_tumbal2[m][6]}"
```

Jika user memasukkan id yang merujuk kepada item potion, maka akan dicetak informasi mengenai potion tersebut dengan format:

```
> "array_tumbal3[n][0]"  
> "Type : {array_tumbal3[n][2]}"  
> "Quantity : {array_tumbal2[n][3]}"
```

Jika user memasukkan id yang merujuk kepada item monster ball, maka akan dicetak informasi mengenai monster ball dengan format:

```
> "array_tumbal4[o][0]"  
> "Quantity : {array_tumbal4[o][3]}"
```

Jika user memasukkan command “KELUAR”, maka user akan keluar dari program inventory

Dengan:

- a. **a+1** adalah nomor id daftar inventory
- b. **listmonster[k][1]** adalah nama atau tipe monster
- c. **listinvmonster[j][2]** adalah level dari monster
- d. **listmonster[k][4]** adalah hit points dari monster
- e. **listinvent[l][1]** adalah tipe item potion
- f. **listinvent[l][2]** adalah jumlah dari item
- g. **array_tumbal2[m][0]** adalah tipe item pilihan, dalam hal ini monster
- h. **array_tumbal3[n][0]** adalah tipe item pilihan, dalam hal ini potion
- i. **array_tumbal4[o][0]** adalah tipe item pilihan, dalam hal ini monster ball
- j. **array_tumbal2[m][2]** adalah nama atau tipe monster yang dipilih
- k. **array_tumbal3[n][2]** adalah tipe potion yang dipilih
- l. **array_tumbal2[m][3]** adalah nilai attack power monster yang dipilih
- m. **array_tumbal2[m][4]** adalah nilai defense power monster yang dipilih
- n. **array_tumbal2[m][5]** adalah nilai hit point monster yang dipilih
- o. **array_tumbal2[m][6]** adalah level monster yang dipilih
- p. **array_tumbal3[n][3]** adalah jumlah kuantitas potion dengan tipe sama yang dimiliki
- q. **array_tumbal4[o][3]** adalah jumlah kuantitas monsterball yang dimiliki

9. F08 - Battle

Nama command: battle(uid , username , role , oc , monster , inv_item , inv_monster)

Masukkan :

Memberikan parameter uid , username , role , oc , monster , inv_item , dan inv_monster.

Menerima input choice , perintah , pilih secara berulang.

Keluaran :

➤ “>>> BATTLE”

Di awal, akan menampilkan terlebih dahulu monster musuh yang terpilih secara acak dan langsung dengan format pesan :

➤ “RAWRRR, Monster {monster[num][1]} telah muncul !!!”
➤ “Name: {monster[num][1]}
ATK Power: {monster[num][2]} --> {temp_atk_enemy}
DEF Power: {monster[num][3]} --> {temp_def_enemy}
HP: {monster[num][4]} --> {temp_hp_enemy}
Level: {level_enemy}”

Dilanjutkan secara langsung dengan menampilkan pilihan monster yang bisa dipilih oleh pengguna dengan format :

➤ “{m}. {monster[inv_monster[i][1]][1]}”

Berdasarkan nilai m, pengguna akan diminta input choice untuk memilih monster yang akan digunakan, jika input choice terhadap nilai m tidak valid, maka akan diminta input kembali didahului dengan mencetak pesan :

➤ “Pilihan nomor tidak tersedia!”

Dilanjutkan secara langsung dengan menampilkan pilihan monster yang telah dipilih oleh pengguna dengan format :

➤ “RAWRRR, {role} {username} mengeluarkan monster {monster[inv_monster[monster_choice][1]][1]} !!!”
➤ “Name: {monster[inv_monster[monster_choice][1]][1]}
ATK Power: {monster[inv_monster[monster_choice][1]][2]} --> {temp_atk_monster}
DEF Power: {monster[inv_monster[monster_choice][1]][3]} --> {temp_def_monster}
HP: {monster[inv_monster[monster_choice][1]][4]} --> {temp_hp_monster}
Level: {level_monster}”

Dimulai pada bagian ini, selama nilai dari temp_hp_monster dan temp_hp_enemy tidak 0, maka akan dicetak pesan ke layar dengan format :

➤ “===== TURN {turn} {monster[inv_monster[monster_choice][1]][1]} =====”

➤ “1. Attack\n2. Use Potion\n3. Quit”

Pengguna akan diminta input perintah, dan jika tidak valid maka akan meminta input kembali didahului dengan menampilkan pesan :

➤ “Perintah tidak ditemukan.”

Jika perintah telah valid, maka selama pengguna memilih bagian 2, jika tidak ada potion, maka akan menampilkan pesan :

➤ “Anda tidak memiliki Potion dalam inventory.”

Tetapi jika tersedia, akan meminta input pilih dari pengguna dengan terlebih dahulu menampilkan pilihan potion dengan format :

➤ “===== POTION LIST =====”

- “1. Strength Potion (Qty: {qty_s[1]}) - Increases ATK Power
- 2. Resilience Potion (Qty: {qty_r[1]}) - Increases DEF Power
- 3. Healing Potion (Qty: {qty_h[1]}) - Restores Health
- 4. Cancel”

Jika input pilih tidak valid, maka akan meminta input kembali didahului dengan menampilkan pesan :

➤ “Perintah tidak ditemukan.”

Jika memilih suatu potion yang jumlahnya 0, maka akan meminta input pilih kembali dengan didahului menampilkan pesan :

➤ “Wah, kamu sedang tidak memiliki ramuan ini, silakan pilih ramuan lain!”

Jika memilih suatu potion yang telah diminum sebelumnya, maka akan meminta input pilih kembali dengan didahului menampilkan pesan :

➤ “Kamu mencoba memberikan ramuan ini kepada {monster[inv_monster[monster_choice][1]][1]}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.”

Tetapi jika jumlahnya lebih dari 0 dan belum diminum sebelumnya, maka akan menampilkan pesan tertentu serta perubahan stats dengan format :

➤ “Setelah meminum ramuan ini, ...”

➤ “{stats_lama} → {stats_baru}”

Jika memilih cancel, maka akan meminta input perintah kembali dengan format :

➤ “===== TURN {turn} {({monster[inv_monster[monster_choice][1]][1]})} =====”

➤ “1. Attack\n2. Use Potion\n3. Quit”

Jika sebelumnya telah berhasil minum potion, maka turn pengguna akan di skip, sehingga musuh menyerang, dan looping berulang, didahului menampilkan pesan :

➤ “===== TURN {turn} {({monster[num][1]})} =====”

➤ “SCHWINKKK, {monster[num][1]} menyerang

{monster[inv_monster[monster_choice][1]][1]} !!!”

➤ “Name: {monster[inv_monster[monster_choice][1]][1]}

ATK Power: {temp_atk_monster}

DEF Power: {temp_def_monster}

HP: {last_hp_monster} --> {temp_hp_monster}

Level: {level_monster}”

➤ “Penjelasan Battle - ATT : Attack = {random_atk_enemy} ; Reduced by = {random_atk_enemy * (temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100}”

➤ “Penjelasan Battle - HP : Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP = {last_hp_monster} ; Final HP =

FLOOR({last_hp_monster} - {random_atk_enemy} * (100 - temp_def_monster)/100) = {temp_hp_monster}"

Jika sebelumnya belum berhasil minum potion, dan perintah yang diinput adalah Attack, maka turn pengguna akan dijalankan, sehingga saling berbalas menyerang, looping berulang, didahului menampilkan pesan :

- "SCHWINKKK, {monster[inv_monster[monster_choice][1]][1]} menyerang {monster[num][1]} !!!"
- "Name: {monster[num][1]}
ATK Power: {temp_atk_enemy}
DEF Power: {temp_def_enemy}
HP: {last_hp_enemy} --> {temp_hp_enemy}
Level: {level_enemy}"
- "Penjelasan Battle - ATT : Attack = {random_atk_monster} ; Reduced by = {random_atk_monster * (temp_def_enemy)/100} ({temp_def_enemy}%) ; Attack Results = {random_atk_monster * (100 - temp_def_enemy)/100}"
- "Penjelasan Battle - HP : Attack Results = {random_atk_monster * (100 - temp_def_enemy)/100} ; Last HP = {last_hp_enemy} ; Final HP = FLOOR({last_hp_enemy} - {random_atk_monster * (100 - temp_def_enemy)/100}) = {temp_hp_enemy}"
-
- "===== TURN {turn} ({monster[num][1]}) ====="
- "SCHWINKKK, {monster[num][1]} menyerang {monster[inv_monster[monster_choice][1]][1]} !!!"
- "Name: {monster[inv_monster[monster_choice][1]][1]}
ATK Power: {temp_atk_monster}
DEF Power: {temp_def_monster}
HP: {last_hp_monster} --> {temp_hp_monster}
Level: {level_monster}"
- "Penjelasan Battle - ATT : Attack = {random_atk_enemy} ; Reduced by = {random_atk_enemy * (temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100}"
- "Penjelasan Battle - HP : Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP = {last_hp_monster} ; Final HP = FLOOR({last_hp_monster} - {random_atk_enemy * (100 - temp_def_monster)/100}) = {temp_hp_monster}"

Jika memilih perintah Quit, maka looping akan dipaksa berhenti.

Jika musuh berhasil dikalahkan, maka akan dipilih acak jumlah oc yang didapat dan dicetak pesan :

- "Selamat, Anda berhasil mengalahkan monster {monster[num][1]} !!!"
- "Total OC yang diperoleh : {get_oc}"

Jika dikalahkan musuh, maka akan dicetak pesan :

- "Yahhh, Anda dikalahkan monster {monster[num][1]}. Jangan menyerah, coba lagi !!!"

Jika kabur, maka akan dicetak pesan :

- "Anda berhasil kabur dari BATTLE"

Jika fungsi battle telah selesai dijalankan, maka akan dikembalikan nilai uid , username , role , oc , monster , inv_item , dan inv_monster di akhir.

10. F09 - Arena

Nama command: arena(uid , username , role , oc , monster , inv_item , inv_monster)

Masukkan :

Memberikan parameter uid , username , role , oc , monster , inv_item , dan inv_monster.

Menerima input choice , perintah , pilih secara berulang.

Keluaran :

➤ “>>> ARENA”

Di awal, akan menampilkan secara langsung pilihan monster yang bisa dipilih oleh pengguna dengan format :

➤ “{m}. {monster[inv_monster[i][1]][1]}”

Berdasarkan nilai m, pengguna akan diminta input choice untuk memilih monster yang akan digunakan, jika input choice terhadap nilai m tidak valid, maka akan diminta input kembali didahului dengan mencetak pesan :

➤ “Pilihan nomor tidak tersedia!”

Dilanjutkan secara langsung dengan menampilkan pilihan monster yang telah dipilih oleh pengguna dengan format :

➤ “RAWRRR, {role} {username} mengeluarkan monster {monster[inv_monster[monster_choice][1]][1]} !!!”
➤ “Name: {monster[inv_monster[monster_choice][1]][1]}
ATK Power: {monster[inv_monster[monster_choice][1]][2]} --> {temp_atk_monster}
DEF Power: {monster[inv_monster[monster_choice][1]][3]} --> {temp_def_monster}
HP: {monster[inv_monster[monster_choice][1]][4]} --> {temp_hp_monster}
Level: {level_monster}”

Dimulai pada bagian ini, selama nilai dari stage masih kurang dari 6 dan pengguna belum kalah, maka akan dicetak pesan ke layar beserta monster musuh yang muncul dengan format :

➤ “===== STAGE {stage} =====”
➤ “RAWRRR, Monster {monster[num][1]} telah muncul !!!”
➤ “Name: {monster[num][1]}
ATK Power: {monster[num][2]} --> {temp_atk_enemy}
DEF Power: {monster[num][3]} --> {temp_def_enemy}
HP: {monster[num][4]} --> {temp_hp_enemy}
Level: {stage}”

Dimulai pada bagian ini, selama nilai dari temp_hp_monster dan temp_hp_enemy tidak 0, maka akan dicetak pesan ke layar dengan format :

➤ “===== STAGE {stage} - TURN {turn} {monster[inv_monster[monster_choice][1]][1]} =====”
➤ “1. Attack\n2. Use Potion\n3. Quit”

Pengguna akan diminta input perintah, dan jika tidak valid maka akan meminta input kembali didahului dengan menampilkan pesan :

➢ “**Perintah tidak ditemukan.**”

Jika perintah telah valid, maka selama pengguna memilih bagian 2, jika tidak ada potion, maka akan menampilkan pesan :

➢ “**Anda tidak memiliki Potion dalam inventory.**”

Tetapi jika tersedia, akan meminta input pilih dari pengguna dengan terlebih dahulu menampilkan pilihan potion dengan format :

➢ “===== **POTION LIST** =====”

➢ “**1. Strength Potion (Qty: {qty_s[1]}) - Increases ATK Power**

2. Resilience Potion (Qty: {qty_r[1]}) - Increases DEF Power

3. Healing Potion (Qty: {qty_h[1]}) - Restores Health

4. Cancel”

Jika input pilih tidak valid, maka akan meminta input kembali didahului dengan menampilkan pesan :

➢ “**Perintah tidak ditemukan.**”

Jika memilih suatu potion yang jumlahnya 0, maka akan meminta input pilih kembali dengan didahului menampilkan pesan :

➢ “**Wah, kamu sedang tidak memiliki ramuan ini, silakan pilih ramuan lain!**”

Jika memilih suatu potion yang telah diminum sebelumnya pada stage yang sama, maka akan meminta input pilih kembali dengan didahului menampilkan pesan :

➢ “**Kamu mencoba memberikan ramuan ini kepada**

{monster[inv_monster[monster_choice][1]][1]}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.”

Tetapi jika jumlahnya lebih dari 0 dan belum diminum sebelumnya, maka akan menampilkan pesan tertentu serta perubahan stats dengan format :

➢ “**Setelah meminum ramuan ini, ...**”

➢ “**{stats_lama} → {stats_baru}**”

Jika memilih cancel, maka akan meminta input perintah kembali dengan format :

➢ “===== **STAGE {stage} - TURN {turn}**
({monster[inv_monster[monster_choice][1]][1]}) =====”

➢ “**1. Attack\n2. Use Potion\n3. Quit**”

Jika sebelumnya telah berhasil minum potion, maka turn pengguna akan di skip, sehingga musuh menyerang, dan looping berulang, didahului menampilkan pesan :

➢ “===== **STAGE {stage} - TURN {turn}**
({monster[num][1]}) =====”

➢ “**SCHWINKKK, {monster[num][1]} menyerang
{monster[inv_monster[monster_choice][1]][1]} !!!**”

➢ “**Name: {monster[inv_monster[monster_choice][1]][1]}**
ATK Power: {temp_atk_monster}
DEF Power: {temp_def_monster}”

- HP: {last_hp_monster} --> {temp_hp_monster}
 Level: {level_monster}”
- “Penjelasan Battle - ATT : Attack = {random_atk_enemy} ; Reduced by = {random_atk_enemy * (temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100}”
 - “Penjelasan Battle - HP : Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP = {last_hp_monster} ; Final HP = FLOOR({last_hp_monster} - {random_atk_enemy * (100 - temp_def_monster)/100}) = {temp_hp_monster}”
- Jika sebelumnya belum berhasil minum potion, dan perintah yang diinput adalah Attack, maka turn pengguna akan dijalankan, sehingga saling berbalas menyerang, looping berulang, didahului menampilkan pesan :
- “SCHWINKKK, {monster[inv_monster[monster_choice][1]][1]} menyerang {monster[num][1]} !!!”
 - “Name: {monster[num][1]}
 ATK Power: {temp_atk_enemy}
 DEF Power: {temp_def_enemy}
 HP: {last_hp_enemy} --> {temp_hp_enemy}
 Level: {stage}”
 - “Penjelasan Battle - ATT : Attack = {random_atk_monster} ; Reduced by = {random_atk_monster * (temp_def_enemy)/100} ({temp_def_enemy}%) ; Attack Results = {random_atk_monster * (100 - temp_def_enemy)/100}”
 - “Penjelasan Battle - HP : Attack Results = {random_atk_monster * (100 - temp_def_enemy)/100} ; Last HP = {last_hp_enemy} ; Final HP = FLOOR({last_hp_enemy} - {random_atk_monster * (100 - temp_def_enemy)/100}) = {temp_hp_enemy}”
 -
 - “===== STAGE {stage} - TURN {turn} ({monster[num][1]}) =====”
 - “SCHWINKKK, {monster[num][1]} menyerang {monster[inv_monster[monster_choice][1]][1]} !!!”
 - “Name: {monster[inv_monster[monster_choice][1]][1]}
 ATK Power: {temp_atk_monster}
 DEF Power: {temp_def_monster}
 HP: {last_hp_monster} --> {temp_hp_monster}
 Level: {level_monster}”
 - “Penjelasan Battle - ATT : Attack = {random_atk_enemy} ; Reduced by = {random_atk_enemy * (temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100}”
 - “Penjelasan Battle - HP : Attack Results = {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP = {last_hp_monster} ; Final HP = FLOOR({last_hp_monster} - {random_atk_enemy * (100 - temp_def_monster)/100}) = {temp_hp_monster}”

Jika memilih perintah Quit, maka looping akan dipaksa berhenti.

Jika musuh berhasil dikalahkan, maka akan dipilih acak jumlah oc yang didapat, stage akan berlanjut, tapi terlebih dahulu dicetak pesan :

- “Selamat, Anda berhasil mengalahkan monster {monster[num][1]} !!!”
- “STAGE CLEARED! Anda akan mendapatkan {get_ocs} OC pada sesi ini.”

Jika belum selesai sampai akhir, maka akan dicetak pula pesan :

- “Memulai stage berikutnya...”

Jika dikalahkan musuh, maka arena terhenti dan akan dicetak pesan :

- “Yahhh, Anda dikalahkan monster {monster[num][1]}. Jangan menyerah, coba lagi !!!”

Jika kabur, maka arena akan terhenti dan akan dicetak pesan :

- “Anda berhasil kabur dari ARENA”

Jika berhasil memenangkan semua stage, maka akan dicetak pesan :

- “Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!”

➤

- “===== STATS =====

Total Hadiah: {get_oc} OC

Jumlah Turn: {total_turn} Turn

Jumlah Stage: {stage - 1} Stage

Potion Digunakan: {total_potion} Potion

HP Recovered: {total_healing} HP

Damage Diberikan: {total_damage_deal} Damage

Damage Diterima: {total_damage_receive} Damage”

Jika dikalahkan musuh, maka akan dicetak pesan :

- “GAME OVER! Sesi latihan berakhir pada stage {stage}!”

➤

- “===== STATS =====

Total Hadiah: {get_oc} OC

Jumlah Turn: {total_turn} Turn

Jumlah Stage: {stage - 1} Stage

Potion Digunakan: {total_potion} Potion

HP Recovered: {total_healing} HP

Damage Diberikan: {total_damage_deal} Damage

Damage Diterima: {total_damage_receive} Damage”

Jika kabur, maka akan dicetak pesan :

- “GAME OVER! Anda mengakhiri sesi latihan!”

➤

- “===== STATS =====

Total Hadiah: {get_oc} OC

Jumlah Turn: {total_turn} Turn

Jumlah Stage: {stage - 1} Stage

Potion Digunakan: {total_potion} Potion

HP Recovered: {total_healing} HP

Damage Diberikan: {total_damage_deal} Damage

Damage Diterima: {total_damage_receive} Damage”

Jika fungsi arena telah selesai dijalankan, maka akan dikembalikan nilai uid , username , role , oc , monster , inv_item , dan inv_monster di akhir.

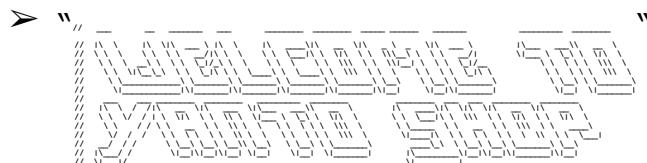
11. F10 - Shop & Currency

Nama command: shop(data_shop_monster, data_monster, data_item_shop, data_invent_item, user_loggedin, data_invent_monster)

Masukkan:

- Memberikan parameter **data_shop_monster, data_monster, data_item_shop, data_invent_item, user_loggedin, data_invent_monster**
- Menerima masukan pilihan aksi (lihat/beli/keluar)
- Menerima masukan pilihan lihat (monster/potion)
- Menerima masukan pilihan beli (monster/potion)
- Menerima masukan id monster yang mau dibeli
- Menerima masukan id potion yang mau dibeli
- Menerima masukan jumlah potion yang mau dibeli

Keluaran:



- **“Input tidak valid, mohon input ulang”**

Apabila agent memberi input aksi diluar pilihan yang telah diberikan

- **“Sudah tidak ada monster yang terjual, datang lagi di lain waktu!”**

Apabila user akan melihat atau membeli monster tetapi shop tidak menjual satupun monster

- **“Jumlah O.W.C.A. coin-mu sekarang <jumlah OC agent>”**

Menunjukkan jumlah oc agentsaat akan membeli monster atau potion

- **“ID | Type | Stok | Harga”**

Menampilkan tabel untuk potion yang terjual di shop kepada agent

- **“ID | Type | ATK Power | DEF Power | Stok | Harga”**

Menampilkan tabel untuk daftar monster yang terjual di shop kepada agent

- **“Sudah tidak ada potion yang dapat dibeli di shop, datang lagi di lain waktu!”**

Apabila agent akan melihat atau membeli potion tetapi shop telah kehabisan stok semua potion

- **“Maaf, stok potion tersebut telah habis”**

Apabila potion yang mau dibeli oleh agent telah habis

- **“Jumlah barang yang mau dibeli melebihi stok, mohon input ulang”**

Apabila agent input jumlah barang yang mau dibeli melebihi stok barang yang tersedia di shop

➢ **“OC-mu tidak cukup”**

Apabila agent tidak memiliki oc yang cukup untuk membeli suatu barang/monster

➢ **“Berhasil membeli item: <jumlah beli> <nama potion>. Item sudah masuk ke inventory-mu!”**

apabila proses pembelian potion sukses

➢ **“OC-mu tersisa <jumlah OC agent>”**

keluaran setelah agent membeli barang/monster untuk melihat sisa oc yang agent miliki

➢ **“Stok sudah habis, silakan beli di lain waktu!”**

Apabila stok monster yang mau dibeli oleh agent telah habis

➢ **“Anda telah memiliki monster tersebut, mohon pilih monster yang lain”**

Apabila agent membeli jenis monster yang sama dengan monster yang sudah dimiliki oleh agent

➢ **“Berhasil membeli monster: <nama monster>. Item sudah masuk ke inventory-mu!”**

Apabila agent telah berhasil membeli monster

➢ **“Mr. Yanto bilang terima kasih, belanja di yanto shop lagi ya nanti :)”**

Cetakan salam sebelum agent keluar dari shop

12. F11 – Laboratory

Nama_command: laboratory(monster, monster_inventory, oc_user,id_user)

Memberikan parameter monster, monster_inventory, oc_user,id_user

Fungsi akan menampilkan monster yang dimiliki oleh user dan user dapat memilih monster yang akan di upgrade.

Menerima masukkan integer No_monster:

- Jika No_monster tidak ada di list yang ditampilkan maka fungsi akan mengeluarkan : “Pilihan monster tidak valid”
- Jika Monster yang dipilih sudah level 5 maka fungsi akan mengeluarkan : “maaf monster yang anda pilih sudah memiliki level maksimal”
- Jika user ingin meng-upgrade level namun OC user kurang dari yang seharusnya, maka fungsi akan mengeluarkan : “O.W.C.A Coin tidak cukup”

- Jika No_monster valid, level monster kurang dari 5, dan OC user cukup, maka fungsi akan mengeluarkan : “Selamat, monster berhasil diupgrade” dan me-return nilai pada monster_inventory dan oc_user setelah digunakan

13. F12 - Shop Management

Nama command: manage_shop(**Monstershop, DataMonster, ItemShop, item_inventory, user_login, monsterinventory**)

Masukkan:

- Memberikan parameter **Monstershop, DataMonster, ItemShop, item_inventory, user_login, monsterinventory**
- Menerima masukan pilihan aksi (lihat/tambah/ubah/hapus/keluar)
- Menerima masukan lihat (monster/potion)
- Menerima masukan tambah (monster/potion)
-
-

14. F13 - Monster Management

Nama command: monster_mng(listmon)

Masukkan:

- Memberikan parameter **listmon**
- Menerima masukan aksi
- Menerima masukan untuk melanjutkan aksi
- Menerima masukan nama atau tipe monster baru
- Menerima masukan nilai attack power monster baru
- Menerima masukan nilai defense power monster baru
- Menerima masukan nilai hit points monster baru
- Menerima masukan untuk menambahkan monster yang telah dibuat ke database

Keluaran:

- “
- “1. Tampilkan semua Monster”
- “2. Tambah Monster baru”

Akan dilakukan validasi dari masukkan pilihan aksi. Jika agent memilih aksi dengan nilai yang tidak tersedia, maka akan dicetak:

➤ “Pilihan aksi tidak tersedia, pilih kembali”

Jika agent memilih aksi dengan nilai 1, maka akan dicetak informasi semua monster dalam permainan dengan format:

➤ “{arr1}”

Jika agent memilih aksi dengan nilai 2, maka akan dimulai pembuatan monster dan akan dicetak:

➤ “Memulai pembuatan monster baru...”

Akan dilakukan validasi dari nilai masukan nama atau tipe monster. Apabila agent mengosongkan nama atau tipe monster, akan dicetak :

➤ “Masukan input harus bernilai, coba lagi!”

Apabila agent memasukan nama atau tipe monster dengan tipe integer, akan dicetak :

➤ “Masukan input bertipe string, coba lagi!”

Apabila agent memasukkan nama atau tipe monster yang sama dengan data pada daftar monster, akan dicetak :

➤ “Nama sudah terdaftar, coba lagi!”

Apabila agent memasukkan nama atau tipe monster dengan tipe string yang belum terdaftar sebelumnya pada data daftar monster, nama tersebut akan tersimpan secara sementara.

Akan dilakukan validasi dari nilai masukan attack power monster baru. Apabila agent mengosongkan nilai attack power, akan dicetak :

➤ “Masukan input harus bernilai, coba lagi!”

Apabila agent memasukan nilai attack power dengan tipe string atau tipe integer yang bernilai negatif, akan dicetak :

➤ “Masukan input bertipe integer dan bernilai positif, coba lagi!”

Apabila agent memasukkan nilai attack power dengan tipe integer positif, nilai tersebut akan tersimpan secara sementara.

Akan dilakukan validasi dari nilai masukan defense power monster baru. Apabila agent mengosongkan nilai defense power, akan dicetak :

➤ “Masukan input harus bernilai, coba lagi!”

Apabila agent memasukan nilai defense power dengan tipe string atau tipe integer yang bernilai negatif, akan dicetak :

➤ “Masukan input bertipe integer dan bernilai positif, coba lagi!”

Apabila agent memasukan nilai defense power dengan nilai diatas batas (0-50), akan dicetak :

➤ “**DEF Power harus bernilai 0-50, coba lagi!**”

Apabila agent memasukkan nilai defense power dengan tipe integer positif dibawah batas, nilai tersebut akan tersimpan secara sementara.

Akan dilakukan validasi dari nilai masukan hit points monster baru. Apabila agent mengosongkan nilai hit points, akan dicetak :

➤ “**Masukan input harus bernilai, coba lagi!**”

Apabila agent memasukkan nilai hit points dengan tipe string atau tipe integer yang bernilai negatif, akan dicetak :

➤ “**Masukan input bertipe integer dan bernilai positif, coba lagi!**”

Apabila agent memasukkan nilai hit dengan tipe integer positif, nilai tersebut akan tersimpan secara sementara.

Setelah semua nilai yang dibutuhkan untuk membuat monster baru tervalidasi, agent memasukkan pilihan untuk melanjutkan menambahkan monster baru pada daftar monster. Jika agent mengosongkan pilihan atau memasukkan pilihan yang tidak tersedia, maka akan dicetak:

➤ “**Pilihan tidak tersedia, coba lagi!**”

Apabila agent memasukkan pilihan Y atau y, maka monster baru akan ditambahkan pada daftar monster dan akan dicetak pesan:

➤ “**Monster baru telah ditambahkan**”

Apabila agent memasukkan pilihan N atau n, maka akan dicetak :

➤ “**Monster gagal ditambahkan**”

Setelah menyelesaikan salah satu aksi, agent memasukkan pilihan untuk melanjutkan aksi. Jika agent mengosongkan pilihan atau memasukkan pilihan yang tidak tersedia, maka akan dicetak:

➤ “**Pilihan tidak tersedia, coba lagi!**”

Apabila agent memasukkan pilihan Y atau y, maka agent dapat kembali memasukkan aksi yang ingin dilakukan.

Apabila agent memasukkan pilihan N atau n, maka list daftar monster terbarukan akan dikembalikan, agent akan keluar dari program, dan akan dicetak :

➤ “
PARA MONSTER MENGIRIM PESAN
SELAMAT TINGGAL”

Dengan:

- a. **arr1** adalah informasi mengenai seluruh monster

15. F14 - Load

Nama command: load()

Masukkan :

Memberikan tidak ada parameter (None).

Menerima input folder dalam ArgumentParser.

Keluaran :

Jika folder tidak diinput, maka program akan berhenti dengan mengembalikan nilai tuple (1 , 1 , 1 , 1 , 1 , 1), didahului dengan mencetak pesan :

➢ **“Tidak ada nama folder yang diberikan!”**

➢ **“Usage : python main.py <nama_folder>”**

Jika folder tidak ditemukan, maka program akan berhenti dengan mengembalikan nilai tuple (1 , 1 , 1 , 1 , 1 , 1), didahului dengan mencetak pesan :

➢ **“Folder {folder} tidak ditemukan!”**

Jika folder ditemukan, maka akan dikembalikan nilai user , monster , inv_item , inv_monster , shop_item , dan shop_monster, didahului dengan mencetak pesan :

➢ **“Selamat datang di program OWCA!”**

16. F15 - Save

Nama command: save(user , monster , inv_item , inv_monster , shop_item , shop_monster)

Masukkan :

Memberikan parameter user , monster , inv_item , inv_monster , shop_item , dan shop_monster

Menerima input directory

Keluaran :

➢ **“>>> SAVE”**

Selama terdapat karakter yang dilarang dalam directory yang dimasukkan, maka akan meminta input kembali didahului dengan mencetak pesan :

➢ **“Nama folder yang dimasukkan mengandung karakter yang dilarang yaitu (/:*?"><|).”**

Jika input directory valid, maka akan dicetak pesan :

➢ **“Saving...”**

Jika sebelumnya folder belum ada, dengan parent bernilai ‘.data/’ , maka saat pembuatan folder akan dicetak pesan :

➢ **“”Membuat folder {parent + directory}...”**

Jika penyimpanan telah berhasil dilakukan, maka akan dicetak pesan :

➢ **“Berhasil menyimpan data di folder {parent + directory}!”**

Jika fungsi save telah selesai dijalankan, maka tidak akan dikembalikan nilai apapun (None).

17. F16 - Exit

Nama command: exit(check_logres , user , monster , inv_item , inv_monster , shop_item , shop_monster)

Masukkan :

Memberikan parameter check_logres , user , monster , inv_item , inv_monster , shop_item , dan shop_monster.

Menerima input ending.

Keluaran :

➤ “>>> EXIT”

Jika parameter check_logres bernilai False, maka program akan berhenti didahului dengan mencetak pesan :

➤ “**Tidak ada program yang sedang berjalan.**”

Jika parameter check_logres bernilai True, maka akan meminta input ending didahului dengan mencetak pesan :

➤ “**Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) :**”

Selama input ending tidak valid, maka akan meminta input kembali didahului dengan mencetak pesan :

➤ “**Masukan Anda tidak valid, silakan ulangi kembali!**”

Jika ending valid bernilai ‘y’, maka program akan melanjutkan ke subprogram save dengan mencetak pesan :

➤ “>>> SAVE”

Jika ending valid bernilai ‘n’, maka program akan berhenti dan tidak mengeluarkan pesan apa-apa.

Jika fungsi exit telah selesai dijalankan, maka akan dikembalikan nilai end yang bernilai True.

Desain Kamus Data

1. F00 - Random Number Generator

KAMUS LOKAL

a , b , c , m, seed : integer
function create_lcg (seed : integer) → integer
function lcg() → integer
function generate(range : array of integer) → integer

2. F01 – Register

KAMUS LOKAL

function convert(string : string) → array of character
{ Menerima input berupa string, lalu memecahnya menjadi suatu list / array of character. }

function register(check_login : boolean , username_login : string , user : matrix of data , monster : matrix of data , inv_item : matrix of data , inv_monster : matrix of data) → (boolean , string , matrix of data , matrix of data , matrix of data , matrix of data)
{ Mendaftarkan pengguna baru dengan memasukkan username dan password serta pilihan monster awal. }

3. F02 – Login

KAMUS LOKAL

data_user, user: array of array of string
username_login, password :string
check_login, check1, check2, admin: boolean
i : indeks integer

function login(data_user: array of (array of string) , check_login: boolean ,
username_login: string) → (boolean , string , boolean , array of string)

{I.S. data_user, check_login , username_login terdefinisi
F.S. user berhasil login dan fungsi mengembalikan nilai untuk check_login, username_login, admin, user_loggedin}

4. F03 – Logout

KAMUS LOKAL

check_login : boolean

5. **F04 - Menu & Help**

KAMUS LOKAL

procedure help(check_login : boolean, username : string, admin : boolean)
{I.S : check_login, username, admin terdefinisi }
{F.S : menampilkan tampilan help}

procedure menu(check_login : boolean, username : string, admin : boolean)
{I.S : check_login, username, admin terdefinisi }
{F.S : menampilkan tampilan help}

6. **F05 - Monster**

KAMUS LOKAL

listmon : array of (array of string)

function samain_panjang (listmon: array of (array of string)) → array of (array of string)
{ Mengembalikan array of array of string yang telah dimodifikasi untuk dicetak}

procedure print_mons (listmons: array of (array of string))
{ I.S. : Diberikan listmonn yang telah terdefinisi}
{ F.S. : Menghasilkan cetakan dari array of array of strings yang telah dimodifikasi}

function delarraymanual (listmonns: array of (array of string), array_index: integer) →
array of (array of string)
{ Mengembalikan array of array of string yang telah mengalami pengurangan kolom}

7. **F06 - Potion**

KAMUS LOKAL

procedure potion()
{ I.S nama nama dan efek dari potion terdefinisi }
{ F.S menampilkan nama nama dan efek dari potion terdefinisi }

8. **F07 - Inventory**

KAMUS LOKAL

username : string
listinvmonster : array of (array of string)
listuser : array of (array of string)
listmonster : array of (array of string)
listinvent : array of (array of string)

function cekint (inint: string) → boolean
{ Mengecek string yang dimasukkan bernilai angka atau huruf, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

9. **F08 - Battle**

KAMUS LOKAL

procedure show_enemy(input/ouput : None)

{ I.S. Tampilan layar masih kosong. }

{ F.S. Tampilan monster yang akan dilawan muncul ke layar. }

```

procedure show_team_monster(input/ouput : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang dipilih muncul ke layar. }

function cint(data : matrix of data) → matrix of data
{ Mengubah data yang berupa string of integer menjadi integer. }

function cstr(data : matrix of data) → matrix of data
{ Mengubah data yang berupa anything menjadi string. }

function battle(uid : string , username : string , role : string , oc : string , monster :
matrix of data , inv_item : matrix of data , inv_monster : matrix of data) → (string
, string , string , matrix of data , matrix of data , matrix of data)
{ Melakukan algoritma battle sesuai spesifikasi yang telah dibuat. }

```

10. F09 - Arena

KAMUS LOKAL

```

procedure show_enemy(input/ouput : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang akan dilawan muncul ke layar. }

```

```

procedure show_team_monster(input/ouput : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang dipilih muncul ke layar. }

function cint(data : matrix of data) → matrix of data
{ Mengubah data yang berupa string of integer menjadi integer. }

function cstr(data : matrix of data) → matrix of data
{ Mengubah data yang berupa anything menjadi string. }

function arena(uid : string , username : string , role : string , oc : string , monster : matrix of data , inv_item : matrix of data , inv_monster : matrix of data) → (string , string , string , matrix of data , matrix of data , matrix of data)
{ Melakukan algoritma arena sesuai spesifikasi yang telah dibuat. }

```

11. F10 - Shop & Currency

KAMUS LOKAL

AdaItem	: <u>boolean</u>
IsDijual	: <u>boolean</u>
Monstershop	: <u>array of array of string</u>
DataMonster	: <u>array of array of string</u>
ItemShop	: <u>array of array of string</u>
item_inventory	: <u>array of array of string</u>
user_login	: <u>array of array of string</u>
monsterinventory	: <u>array of array of string</u>
Aksi	: <u>string</u>
LihatApa	: <u>string</u>
beli	: <u>string</u>
IdPotion	: <u>string</u>
JumlahBeli	: <u>string</u>
MonsterTerpilih	: <u>string</u>
IdMonster	: <u>string</u>
Urutan	: <u>integer</u>
urutan	: <u>integer</u>
urutanterpilih	: <u>integer</u>
Isi	: <u>array of string</u>
IsiMonster	: <u>array of string</u>
isiinventory	: <u>array of string</u>
MonsterID	: <u>array of string</u>
Monster	: <u>array of string</u>

```

function slicing(Data : any , Awal : integer , Akhir : integer) -> any
{Menerima input list untuk diambil hanya sebagian dari suatu list}
function IsDigit(string : string) -> bool:

```

{Menerima input string untuk cek apakah isi dari string tersebut merupakan bilangan atau bukan}

function ManualSplit (string : string, b : character) -> array of string:

{Menerima input string untuk mengubahnya menjadi list dimana isinya berupa string yang terpisah-pisah oleh}

function RemoveNewLine(string : string) -> string :

{Menerima input string untuk menghilangkan \n}

12. F11 - Laboratory

KAMUS LOKAL

monster_inventory : array of array of string

id_user : integer

oc_user : integer

list_monster : array of string

idmon : integer

No_monster : integer

konfirm : Char

nama_monster : string

upgrade_cost : array of integer

next_level : integer

current_level : integer

{fungsi untuk Mengupgrade level monster}

{I.S. user, monster_inventory, monster terdefinisi}

{F.S. level monster diupgrade jika memenuhi syarat tertentu dan mengembalikan nilai array of array of string dari monster_inventory dan mengembalikan nilai oc_user}

13. F12 - Shop Management

KAMUS LOKAL

valid, : boolean

Adakah : boolean

data_shop_monster : array of array of string

data_monster : array of array of string

data_item_shop : array of array of string

TambahTypePotion : array of array of string

monsterbelumlaku : array of array of string

potionbelumlaku : array of array of string

monsterBertahan : array of array of string

Aksi : string

LihatApa : string

```

IdPotion          : string
TambahApa        : string
MonsterLaku      : string
inputstok         : string
PilihMonster     : string
HapusId          : string
monsterdihapus  : string
hargamonster     : string
StokAwal         : string
HargaPotion      : string
UbahApa          : string
PilihIdPotion    : string
StokBaru         : string
HargaBaru        : string
PilihIdMonter   : string
MonsterDiubah   : string
HapusApa         : string
YorN              : character
Urutan           : integer
urutan           : integer
Isi               : array of string
IsiMonster        : array of string
Monster           : array of string
Dalam             : array of string
ShowMonster       : array of string
monster          : array of string
potion            : array of string
function slicing(Data : any, Awal : integer, Akhir : integer) -> any
{Menerima input list untuk diambil hanya sebagian dari suatu list}

```

```

function IsDigit(string : string) -> bool;
{Menerima input string untuk cek apakah isi dari string tersebut merupakan
bilangan atau bukan}

```

```

function ManualSplit (string : string, b : character) -> array of string;
{Menerima input string untuk mengubahnya menjadi list dimana isinya berupa
string yang terpisah-pisah oleh}

```

```

function RemoveNewLine(string : string) -> string :
{Menerima input string untuk menghilangkan \n}

```

14. F13 - Monster Management

KAMUS LOKAL

listmon : array of (array of string)

function samain_panjang (listmon: array of (array of string)) → array of (array of string)
{ Mengembalikan array of array of string yang telah dimodifikasi untuk dicetak}

procedure print_mons (listmons: array of (array of string))
{ I.S. : Diberikan listmonn yang telah terdefinisi}
{ F.S. : Menghasilkan cetakan dari array of array of strings yang telah dimodifikasi}

function delarraymanual (listmonns: array of (array of string), array_index: integer) →
array of (array of string)
{ Mengembalikan array of array of string yang telah mengalami pengurangan kolom}

function ceknama (instr: string) → boolean
{ Mengecek string yang dimasukkan bernilai angka atau huruf, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function cekint (inint: string) → boolean
{ Mengecek string yang dimasukkan bernilai angka atau huruf, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function cekdeff (indef: string) → boolean
{ Mengecek string yang dimasukkan bernilai angka dengan batasan yang ditentukan, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function ceklen (instr: string) → boolean
{ Mengecek string yang dimasukkan kosong atau tidak, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function cekar (instr: string, cekarray: array of (array of string)) → boolean
{ Mengecek string yang dimasukkan bila terdapat pada array of array of string, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function IsDigit (stri: string) → boolean
{ Mengecek string dengan panjang 1 yang dimasukkan bernilai angka, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

function cekar (instr: string, cekarray: array of (array of string)) → boolean
{ Mengecek string yang dimasukkan bernilai 1 atau 2, dan mengembalikan nilai True atau False berdasarkan hasil pengecekan}

15. F14 - Load

KAMUS LOKAL

function csplit(string : string , delimiter : character) → array of string
{ Membuat fungsi personal yang serupa dengan fungsi split() pada python. }

function cstrip(string : string , chars : character) → string
{ Membuat fungsi personal yang serupa dengan fungsi strip() pada python. }

```
procedure load(output user : matrix of data , output monster : matrix of data ,  
output inv_item : matrix of data , output inv_monster : matrix of data , output  
shop_item : matrix of data , output shop_monster : matrix of data)  
{ I.S. Menerima data parser dari input yang akan dijalankan. }  
{ F.S. Data pada csv_directory telah tersimpan ke matrix of data. }
```

16. F15 - Save

KAMUS LOKAL

```
function cmap(function : function , map_data : data) → array of data  
{ Membuat fungsi personal yang serupa dengan fungsi map() pada python. }
```

```
function cjoin(separator : character , join_data : data) → string  
{ Membuat fungsi personal yang serupa dengan fungsi join() pada python. }
```

```
procedure save(input user : matrix of data , input monster : matrix of data , input  
inv_item : matrix of data , input inv_monster : matrix of data , input shop_item :  
matrix of data , input shop_monster : matrix of data)  
{ I.S. Data user, monster, inv_item, inv_monster, shop_item, dan shop_monster  
telah terdefinisi. }  
{ F.S. Data user, monster, inv_item, inv_monster, shop_item, dan shop_monster  
telah tersimpan di suatu folder. }
```

17. F16 - Exit

KAMUS LOKAL

```
function exit(check_logres : boolean , user : matrix of data , monster : matrix of  
data , inv_item : matrix of data , inv_monster : matrix of data , shop_item : matrix  
of data , shop_monster : matrix of data) → boolean  
{ Melakukan proses keluar dari program yang sedang berjalan. }
```

Desain Dekomposisi Algoritmik Dan Fungsional Program

1. F00 - Random Number Generator

```
a , b , c , m, seed : integer
function create_lcg (seed : integer) → integer
function lcg() → integer
function generate(range : array of integer) → integer
{Memberikan random integer sesuai range yang diberikan}
```

2. F01 – Register

```
function register(check_login : boolean ,
username_login : string , user : matrix of data ,
monster : matrix of data , inv_item : matrix of data ,
inv_monster : matrix of data) → (boolean , string ,
matrix of data , matrix of data , matrix of data ,
matrix of data)

{ Mendaftarkan pengguna baru dengan memasukkan
username dan password serta pilihan monster awal. }
```

3. F02 – Login

```
function login(data_user: array of (array of string) ,
check_login: boolean , username_login: string) → (boolean ,
string , boolean , array of string)

{I.S. data_user, check_login , username_login terdefinisi
F.S. user berhasil login dan fungsi mengembalikan nilai
untuk check_login, username_login, admin, user_loggedin}
```

4. F03 – Logout

```
function logout(check_login : boolean)

{Memvalidasi keadaan login pengguna dan memberi umpan
balik berdasarkan hasil validasi}
```

5. F04 - Menu & Help

```
procedure help(check_login : boolean , admin : boolean)

{Mencetak pilihan-pilihan command untuk user dengan baik
untuk yang belum login, agent, maupun admin.

I.S. check_login dan admin terdefinisi
```

```
    F.S. mencetak pilihan-pilihan command}
```

6. F05 - Monster

```
procedure monster(listmon: array of array of string)
{Mencetak informasi mengenai seluruh monster pada program
dan menjelaskan pengaruh level pada base attribute tiap
monster
I.S. variabel listmon terdefinisi
F.S. mencetak informasi}
```

7. F06 - Potion

```
procedure potion()
{Mencetak informasi mengenai seluruh potion pada program
dan menjelaskan efeknya pada battle
I.S. Jenis-jenis potion dan efeknya terdefinisi
F.S. menampilkan jenis-jenis potion dan efeknya}
```

8. F07 - Inventory

```
procedure inventory(username : string, listinvmonster :
array of array of string, listuser : array of array of
string, listmonster: array of array of string, listinvent:
array of array of string)
{Mencetak informasi berdasarkan masukan pilihan aksi dari
user
I.S. variabel username, listuser, listinvent,
listinvmonster, dan listmonster terdefinisi
F.S. mencetak informasi sesuai pilihan-pilihan aksi}
```

9. F08 - Battle

```
function battle(uid : string , username : string ,
role : string , oc : string , monster : matrix of data
, inv_item : matrix of data , inv_monster : matrix of
data) → (string , string , string , string , matrix of
data , matrix of data , matrix of data)
{ Melakukan algoritma battle sesuai spesifikasi yang
telah dibuat. }
```

10. F09 - Arena

```
function arena(uid : string , username : string , role
: string , oc : string , monster : matrix of data ,
inv_item : matrix of data , inv_monster : matrix of
data) → (string , string , string , string , matrix of
data , matrix of data , matrix of data)
```

```
{ Melakukan algoritma arena sesuai spesifikasi yang  
telah dibuat. }
```

11. F10 - Shop & Currency

```
function shop(data_shop_monster : array of array of  
string, data_monster : array of array of string,  
data_item_shop : array of array of string,  
data_invent_item : array of array of string, user_loggedin  
: list of string, data_invent_monster : array of array of  
string)  
  
{Fungsi untuk menampilkan daftar barang yang tersedia di  
toko dan melakukan transaksi pembelian.  
  
I.S. : data_shop_monster, data_monster, data_item_shop,  
data_invent_item, user_loggedin, dan data_invent_monster  
terdefinisi  
  
F.S. : memperbarui isi data dari data_shop_monster,  
data_monster, data_item_shop, data_invent_item,  
user_loggedin, dan data_invent_monster}
```

12. F11 - Laboratory

```
function laboratory(monster: array of array of string,  
monster_inventory: array of array of string, oc_user:  
string, id_user: string) → (array of array of string,  
integer)  
  
{fungsi untuk Mengupgrade level monster}  
  
{I.S. user, monster_inventory, monster terdefinisi}  
  
{F.S. level monster diupgrade jika memenuhi syarat  
tertentu dan mengembalikan nilai array of array of string  
dari monster_inventory dan mengembalikan nilai oc_user}
```

13. F12 - Shop Management

```
function manage_shop(data_shop_monster : array of array of  
string, data_monster : array of array of string,  
data_item_shop : array of array of string)  
  
{Fungsi untuk admin mengatur shop baik menambah, mengubah,  
dan menghapus isinya.  
  
I.S. : data_shop_monster, data_monster, data_item_shop,  
dan terdefinisi
```

F.S. : memperbarui isi fata dari data_shop_monster, data_monster, dan data_item_shop}

14. F13 - Monster Management

```
function monster_mng(listmon : array of array of string)

{Mengatur isi data list monster baik menambahkan ataupun melihat isi data monster.}
```

15. F14 - Load

```
procedure load(output user : matrix of data , output
monster : matrix of data , output inv_item : matrix of data ,
output inv_monster : matrix of data , output
shop_item : matrix of data , output shop_monster :
matrix of data)

{ I.S. Menerima data parser dari input yang akan
dijalankan. }
{ F.S. Data pada csv_directory telah tersimpan ke
matrix of data. }
```

16. F15 - Save

```
procedure save(input user : matrix of data , input
monster : matrix of data , input inv_item : matrix of data ,
input inv_monster : matrix of data , input
shop_item : matrix of data , input shop_monster :
matrix of data)

{ I.S. Data user, monster, inv_item, inv_monster,
shop_item, dan shop_monster telah terdefinisi. }
{ F.S. Data user, monster, inv_item, inv_monster,
shop_item, dan shop_monster telah tersimpan di suatu
folder. }
```

17. F16 - Exit

```
function exit(check_logres : boolean , user : matrix
of data , monster : matrix of data , inv_item : matrix
of data , inv_monster : matrix of data , shop_item :
matrix of data , shop_monster : matrix of data) →
boolean

{ Melakukan proses keluar dari program yang sedang
berjalan. }
```

Spesifikasi Fungsi-Fungsi Dalam Program

1. F00 - Random Number Generator

Tabel 6.1 Spesifikasi fungsi dalam program Random Number Generator (F00)

<pre>function generate(range: <u>array of integer</u>) {Memberikan random integer sesuai range yang diberikan}</pre>
<p>KAMUS LOKAL</p> <pre>a , b , c , m, seed : <u>integer</u> function create_lcg (seed : <u>integer</u>) → <u>integer</u> {men-generate seed untuk digunakan dalam perhitungan} function lcg() → <u>integer</u> {men-generate seed untuk digunakan dalam perhitungan}</pre>
<p>ALGORITMA</p> <pre>seed ← os.getpid <u>mod</u> time.time() function create_lcg(seed : <u>integer</u>) lcg_func ← create_lcg(seed : <u>integer</u>) function generate(range: <u>array of integer</u>) → <u>integer</u> random_number = lcg_func() a ← range.0 b ← range.1 b ← b - 1 → (random_number / (m-1)*(b-a+1) + a</pre>
<p>REALISASI FUNGSI & PROSEDUR</p> <pre>function create_lcg(seed) { Fungsi ini akan mengeluarkan suatu bilangan unik tergantung dari Process ID dan Time saat digunakan.} KAMUS LOKAL {seed, a, c, m : <u>integer</u>} ALGORITMA function lcg() seed.0 ← (a * seed.0 + c) <u>mod</u> m → seed.0 → lcg()</pre>

2. F01 – Register

Tabel 6.2 Spesifikasi fungsi dalam program Register (F01)

<pre>function register(check_login : boolean , username_login : string , user : matrix of data , monster : matrix of data , inv_item : matrix of data , inv_monster : matrix of data) → (boolean , string , matrix of data , matrix of data , matrix of data , matrix of data) { Mendaftarkan pengguna baru dengan memasukkan username dan password serta pilihan monster awal. }</pre>
<p>KAMUS LOKAL</p> <pre>user , monster , inv_item , inv_monster : matrix of data array_username , allowed_character : array of character check_login , check : boolean username_login , username , password , role : string uid , oc , putar , level , quantity , accepted_character , starting_monster : integer i , j : integer {index} function convert(string : string) → array of character { Menerima input berupa string, lalu memecahnya menjadi suatu list / array of character. }</pre>
<p>ALGORITMA</p> <pre>if check_login then output("Register gagal!" , username_login) else accepted_character ← 0 array_username ← ["-"] putar ← 1 while (accepted_character ≠ len(array_username)) do if (putar > 1) then output("Username hanya boleh berisi alfabet, angka, underscore, dan strip!") accepted_character ← 0 check ← False while (not(check)) do input(username , password) uid ← str(len(user)) accepted_character ← 0 allowed_character ← ['A', 'a', 'B', 'b', 'C', 'c', 'D', 'd', 'E', 'e', 'F', 'f', 'G', 'g', 'H', 'h', 'I', 'i', 'J', 'j', 'K', 'k', 'L', 'l', 'M', 'm', 'N', 'n', 'O', 'o', 'P', 'p', 'Q', 'q', 'R', 'r', 'S', 's', 'T', 't', 'U', 'u',</pre>

```

'V', 'v', 'W', 'w', 'X', 'x', 'Y', 'y', 'Z', 'z', '0', '1',
'2', '3', '4', '5', '6', '7', '8', '9', '-', '_']
    check ← True
    i traversal [0..(len(user))]
        if (user[i][1] = username) then
            check ← False
            break
    array_username ← convert(username)
    i traversal [0..(len(array_username))]
        j traversal [0..(len(allowed_character))]
            if (array_username[i] = allowed_character[j]) then
                accepted_character ← accepted_character + 1
                break
            putar ← putar + 1
    i traversal [1..3]
        output(i , monster[i][1])
    input(starting_monster)
    while (not(1 ≤ starting_monster ≤ 3)) do
        input(starting_monster)
    role ← "Agent"
    oc ← "0"
    user ← user + [[uid , username , password , role , oc]]
    quantity ← "0"
    inv_item ← inv_item + [[uid , "strength" , quantity] ,
    [uid , "resilience" , quantity] , [uid , "healing" ,
    quantity]]
    check_login ← False
    username_login ← ""
    output("Selamat datang" , username ,
    monster[starting_monster][1])
    → (check_login , username_login , user , monster , inv_item ,
    inv_monster)

```

REALISASI FUNGSI & PROSEDUR

```

function convert(string : string) → array of character
{ Menerima input berupa string, lalu memecahnya menjadi suatu
list / array of character. }
KAMUS LOKAL
    string : string
    array : array of character
    i : integer {index}
ALGORITMA LOKAL
    array ← []
    i traversal [0..(len(string)-1)]
        array ← array + string[i]
    → array

```

3. F02 – Login

Tabel 6.3 Spesifikasi fungsi dalam program Login (F02)

<pre>function login(data_user: <u>array of (array of string)</u> , check_login: <u>boolean</u>, username_login: <u>string</u>) → (<u>boolean</u>, <u>string</u>, <u>boolean</u>, <u>array of string</u>)</pre> <p>{I.S. data_user, check_login , username_login terdefinisi F.S. user berhasil login dan fungsi mengembalikan nilai untuk check_login, username_login, admin, user_loggedin}</p>
<p>KAMUS LOKAL</p> <pre>data_user, user: <u>array of array of string</u> username_login, password :<u>string</u> check_login, check1, check2, admin: <u>boolean</u> i : indeks <u>integer</u></pre>
<p>ALGORITMA</p> <pre>function login(data_user, check_login , username_login) : <u>if</u> (check_login) <u>then</u> output ('Login gagal! Anda telah login dengan username {username_login}, silakan lakukan "LOGOUT" sebelum melakukan login kembali.')</pre> <p><u>else</u></p> <pre> user ← data_user <u>while</u> (<u>not</u>(check_login)) <u>do</u> check1 ← False check2 ← False <u>while</u> ((<u>not</u>(check1)) and (<u>not</u>(check2))) <u>do</u> output("Masukkan Username : ") username_login ← input() output("Masukkan Password : ") password ← input() i traversal [0,(len(user))] <u>if</u> (user[i][1] = username_login) <u>then</u> check1 ← True <u>if</u> (user[i][2] = password) <u>then</u> check2 ← True</pre>

```

break

if ((check1) and (check2)) then
    if (user[i][3] = 'Admin') then
        admin ← True
    else
        admin ← False
    user_loggedin ← user[i]

    elseif ((check1) and (not(check2))) then
        output("Password salah!")
    else:
        output("Username tidak terdaftar!")

    check_login ← True

    output(f'Selamat datang, Agent
{username_login}!\\nMasukkan command "help" untuk daftar
perintah yang dapat kamu panggil.')

    return (check_login, username_login, admin,
user_loggedin)

```

REALISASI FUNGSI & PROSEDUR

```

output("Masukkan pilihan aksi ")
pilihan_aksi input()
if pilihan_aksi ← 'login' then
    check_login, username, admin, user_loggedin ←
login(data_user, check_login)
    user_id ← user_loggedin[0]
    username ← user_loggedin[1]
    role ← user_loggedin[3]
    oc ← user_loggedin[4]
    if role = "admin":
        admin ← True

```

4. F03 – Logout

Tabel 6.4 Spesifikasi fungsi dalam program Logout (F03)

<pre>function logout(check_login : boolean) {Memvalidasi keadaan login pengguna dan memberi umpan balik berdasarkan hasil validasi}</pre>
<p>KAMUS LOKAL</p> <pre>check_login : boolean</pre>
<p>ALGORITMA</p> <pre>function logout(check_login: bool) if check_login then output("Logout berhasil!") check_login ← False else output("Logout gagal!") output("Anda belum login, silahkan login terlebih dahulu.") → check_login</pre>
<p>REALISASI FUNGSI & PROSEDUR</p> <pre>-</pre>

5. F04 - Menu & Help

Tabel 6.5 Spesifikasi fungsi dalam program Menu dan Help (F04)

<pre>procedure help(check_login : boolean, username : <u>string</u>, admin : <u>boolean</u>) {I.S : check_login, username, admin terdefinisi } {F.S : menampilkan tampilan help} procedure menu(check_login : <u>boolean</u>, username : <u>string</u>, admin : <u>boolean</u>) {I.S : check_login, username, admin terdefinisi } {F.S : menampilkan tampilan help}</pre>
<p>KAMUS LOKAL</p> <pre>check_admin , admin : boolean username : string</pre>
<p>ALGORITMA</p>

```

procedure help(check_login : boolean, username : string, admin
: boolean)
    if not(check_login) then
        output(

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

            1. Register - melakukan registrasi user baru
            2. Login - melakukan login ke akun ter-register
            3. Exit - Keluar dari permainan

            ===FOOTNOTE==

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

        else
            if not(admin) then
                output(

                    Halo Agent {username}. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

                    1. Logout - Keluar dari akun
                    2. Monster - Melihat owca-dex yang dimiliki oleh Agent
                    3. Potion - Melihat jenis - jenis potion beserta efeknya
                    4. Inventory - Melihat isi inventory milik Agent
                    5. Battle - Bertarung dengan monster secara acak secara lv1
                    6. Arena - Memasuki arena bertarung yang berisi 5 stage
                    7. Shop - Tempat Agent untuk membeli monster dan potion
                    8. Laboratory - Tempat Agent untuk upgrade monster yang ada di inventory
                    9. Load - Memuat data yang tersimpan
                    10. Exit - Keluar dari permainan )

            else
                output(

                    1. Logout - Untuk melakukan registrasi user baru
                    2. Shop - Mengatur isi shop, baik menambah stok maupun mengurangi stok
                    3. Monster - mengatur monster dalam database
                    4. Load - Memuat data yang tersimpan
                    5. Exit - Keluar dari permainan )

    keluar ← False
    while not(keluar) do

```

```

input(aksi)
if aksi = "keluar" then
    keluar ← True

procedure menu(check_login : boolean, username : string, admin : boolean)
    if not(check_login) then
        output(
            Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.
            1. Register - melakukan registrasi user baru
            2. Login - melakukan login ke akun ter-register
            3. Exit - Keluar dari permainan
            ===FOOTNOTE===
            1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar.
            2. Jangan lupa untuk memasukkan input yang valid!! )
    else
        if not(admin) then
            output(
                Halo Agent {username}. Berikut adalah hal hal yang kamu lakukan. Selamat berpetualang bersama monster pilihanmu!
                Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
                1. Logout - Keluar dari akun
                2. Monster - Melihat owca-dex yang dimiliki oleh Agent
                3. Potion - Melihat jenis - jenis potion beserta efeknya
                4. Inventory - Melihat isi inventory milik Agent
                5. Battle - Bertarung dengan monster secara acak secara
                1v1
                6. Arena - Memasuki arena bertarung yang berisi 5 stage
                7. Shop - Tempat Agent untuk membeli monster dan potion
                8. Laboratory - Tempat Agent untuk upgrade monster yang ada di inventory
                9. Load - Memuat data yang tersimpan
                10. Exit - Keluar dari permainan )

        else
            output(
                1. Logout - Untuk melakukan registrasi user baru
                2. Shop - Mengatur isi shop, baik menambah stok maupun mengurangi stok
                3. Monster - mengatur monster dalam database
                4. Load - Memuat data yang tersimpan
                5. Exit - Keluar dari permainan )

```

REALISASI FUNGSI & PROSEDUR
-

6. F05 - Monster

Tabel 6.6 Spesifikasi fungsi dalam program Monster (F05)

```
procedure monster(listmon: array of array of string)

{Mencetak informasi mengenai seluruh monster pada program
dan menjelaskan pengaruh level pada base attribute tiap
monster
I.S. variabel listmon terdefinisi
F.S. mencetak informasi}
```

KAMUS LOKAL

```
listmon : array of (array of string)

function samain_panjang (listmon: array of (array of
string)) → array of (array of string)
{ Mengembalikan array of array of string yang telah
dimodifikasi untuk dicetak}

procedure print_mons (listmons: array of (array of
string))
{ I.S. : Diberikan listmon yang telah terdefinisi
{ F.S. : Menghasilkan cetakan dari array of array of
strings yang telah dimodifikasi}

function delarraymanual (listmonns: array of (array of
string), array_index: integer) → array of (array of
string)
{ Mengembalikan array of array of string yang telah
mengalami pengurangan kolom}
```

ALGORITMA

```
procedure monster(listmon)
arr ← ["ID", "Type", "ATK Power", "DEF Power", "HP"]
listmonbaru ← delarraymanual(listmon, 0)
arr1 ← [arr] + listmonbaru
output("OWCA DEX")
output("WELCOME TO OWCA-DEX UNTUK MELIHAT MONSTER YANG
TERSEDIA, AGEN!")
```

```

print_mons(arr1)
output("Informasi tambahan owca-dex mengenai level:")
output("Level 1      : Monster menggunakan Base Attribute")
output("Level 2 - 5: Ketika sedang Battle, Base Attribute
monster ditambah dengan Base Attribute dikalikan dengan
((Level - 1) * 10)%.")
output("          Ketika monster naik level, Base Attribute
tetap disimpan, kalkulasi atribut sesuai level dilakukan saat
Battle.")
output("Tips! Cek level monsternya pada inventory, gunakan
command 'inventory'!")
keluar ← False
while (not(keluar))do
    output("'keluar' untuk kembali ke menu : ")
    input(a)
    if (a = 'keluar')then
        keluar ← True

```

REALISASI FUNGSI & PROSEDUR

```

function samain_panjang(listmonn: list)
    hasil ← []
    i traversal[1..(len(listmonn))]
    hasil ← hasil + ['']
    i traversal[1..(len(listmonn[0])-1)]
        b ← 0
        j traversal[1..(len(listmonn))]
            if (b < len(listmonn[j][i]))then
                b ← len(listmonn[j][i])
            j traversal[1..(len(listmonn))]
                hasil[j] ← hasil[j] + [listmonn[j][i]]
                if (b > len(listmonn[j][i]))then
                    while (b > len(listmonn[j][i]))then
                        hasil[j] ← hasil[j] + [" "]
                        listmonn[j][i] ← listmonn[j][i] + [" "]
                    hasil[j] ← hasil[j] + [" | "]
                i traversal[1..(len(listmonn))]
                    hasil[i] ← hasil[i] + [listmonn[i][len(listmonn[0])-1]]
    → hasil

procedure print_mons(listmons: list)
    listmons ← samain_panjang(listmons)
    i traversal[1..(len(listmons))]
        j traversal[1..(len(listmons[i])-1)]
            output(listmons[i][j])
            output(listmons[i][len(listmons[i])-1])

```

```

function delarraymanual(listmonns, array_index)
    listmonbaru ← []
    i traversal[1..(len(listmonns))]
        if (i ≠ array_index) then
            listmonbaru ← listmonbaru + [listmonns[i]]
    → listmonbaru

```

7. F06 - Potion

Tabel 6.7 Spesifikasi fungsi dalam program Potion (F06)

procedure potion() {Mencetak informasi mengenai seluruh potion pada program dan menjelaskan efeknya pada battle I.S. Jenis-jenis potion dan efeknya terdefinisi F.S. menampilkan jenis-jenis potion dan efeknya}
KAMUS LOKAL
-
ALGORITMA
procedure potion() <u>output</u> ("potions") <u>output</u> (" Potion Effect Strength Potion Meningkatkan ATK Power sebanyak 5% dari ATK Power Resilience Potion Meningkatkan Def Power sebanyak 5% dari DEF Power Healing Potion Mengisi darah sebanyak 25% dari Base HP, tidak akan mengisi lebih dari maks HP Tips! Gunakan potion yang tepat di saat yang tepat untuk melawan monster yang lebih kuat! Potion bisa dibeli di shop dengan OC.") keluar ← False <u>while</u> (not(keluar)) <u>do</u> <u>output</u> ("'keluar' untuk kembali ke menu : ") <u>input</u> (a) <u>if</u> (a = 'keluar') <u>then</u> keluar ← True
REALISASI FUNGSI & PROSEDUR
-

8. F07 - Inventory

Tabel 6.8 Spesifikasi fungsi dalam program Inventory (F07)

```
procedure inventory(username : string, listinvmonster :  
array of array of string, listuser : array of array of  
string, listmonster: array of array of string, listinvent:  
array of array of string)
```

{Mencetak informasi berdasarkan masukan pilihan aksi dari
user

I.S. variabel username, listuser, listinvent,
listinvmonster, dan listmonster terdefinisi

F.S. mencetak informasi sesuai pilihan-pilihan aksi}

KAMUS LOKAL

```
username : string  
listinvmonster : array of (array of string)  
listuser : array of (array of string)  
listmonster : array of (array of string)  
listinvent : array of (array of string)
```

```
function cekint ( inint: string ) → boolean  
{ Mengecek string yang dimasukkan bernilai angka atau  
huruf, dan mengembalikan nilai True atau False berdasarkan  
hasil pengecekan}
```

ALGORITMA

```
procedure inventory(username, listinvmonster, listuser,  
listmonster, listinvent)  
    i traversal [0..(len(listuser))]  
        if (listuser[i][1] = username) then  
            id ← listuser[i][0]  
            break  
        output("===== INVENTORY LIST (User ID: ",  
        listuser[i][0], " =====")  
        output("Jumlah O.W.C.A. Coin-mu sekarang ", listuser[i][4],  
        ".")  
        a ← 0  
        array_tumbal2 ← []  
        array_tumbal3 ← []  
        array_tumbal4 ← []  
        j traversal[0..(len(listinvmonster))]  
            if (listinvmonster[j][0] = id) then  
                idmon ← listinvmonster[j][1]  
                k traversal[0..(len(listmonster))]
```

```

        if (listmonster[k][0] = idmon) then
            output((a+1), ". Monster      ")
            output(" (Name: ", listmonster[k][1], ",",
            Lvl:", listinvmonster[j][2], ", HP:",
            listmonster[k][4], ")")
            a ← a+1
            array_tumbal2 ← array_tumbal2 + [[a,
            "Monster", listmonster[k][1],
            listmonster[k][2], listmonster[k][3],
            listmonster[k][4],
            listinvmonster[j][2]]]
    l traversal[0..(len(listinvent))]
        depend on(listinvent[l][0], listinvent[l][1])
        (listinvent[l][0] = id) and (listinvent[l][1] ≠
        "monster_ball") :
            output((a+1), ". Potion      ")
            output("(Type:", listinvent[l][1], ", Qty:",
            listinvent[l][2], ")")
            a ← a+1
            array_tumbal3 ← array_tumbal3 + [[a, "Potion",
            listinvent[l][1], listinvent[l][2]]]
        (listinvent[l][0] = id) and (listinvent[l][1] =
        "monster_ball") :
            output((a+1), ". Monster Ball  ")
            output("(Qty: ", listinvent[l][2], ")")
            a ← a+1
            array_tumbal4 ← [[a, "Monster Ball",
            listinvent[l][2]]]

    keluar ← False
    while (keluar = False) do
        output("Ketikkan id untuk menampilkan detail item:")
        input(ids)
        if (ids ≠ str("KELUAR")) then
            depend on(len(ids), cekint(str(ids)), int(ids))
            (len(ids) = 0) : output("id harus terisi!")
            not(cekint(str(ids)) : output("Pilihan id tidak
            tersedia!")
            output("Ketik 'KELUAR' untuk
            keluar dari inventory.")
            (int(ids) > a) : output("Pilihan id tidak
            tersedia!")
            cekint(str(ids)) :
                m traversal[0..(len(array_tumbal2))]
                if (int(ids) = array_tumbal2[m][0]) then

```

```

        output(array_tumbal2[m][1])
        output("Name      :",
        array_tumbal2[m][2])
        output("ATK Power :",
        array_tumbal2[m][3])
        output("DEF Power :",
        array_tumbal2[m][4])
        output("HP       :",
        array_tumbal2[m][5])
        output("Level     :",
        array_tumbal2[m][6])
        break
    n traversal[0..(len(array_tumbal3))]
    if (int(ids) = array_tumbal3[n][0])then
        output(array_tumbal3[n][1])
        output("Type      :",
        array_tumbal3[n][2])
        output("Quantity  :",
        array_tumbal3[n][3])
        break
    o traversal[0..(len(array_tumbal4))]then
        if (int(ids) = array_tumbal4[o][0])then
            output(array_tumbal4[o][1])
            output("Quantity  :",
            array_tumbal4[o][2])
            break

    else
        keluar ← True

```

REALISASI FUNGSI & PROSEDUR

```

function cekint(inint)
    i traversal[1..len(inint)]
    cekinte[i] ← ord(inint[i])

    f ← True
    i ← 0
    while (i<len(cekinte))do
        if (48≤cekinte[i]≤57) then
            i ← i+1
        else
            f ← False
            i ← i+1
    → f

```

9. F08 - Battle

Tabel 6.9 Spesifikasi fungsi dalam program Battle (F08)

<pre> function battle(uid : <u>string</u> , username : <u>string</u> , role : <u>string</u> , oc : <u>string</u> , monster : <u>matrix of data</u> , inv_item : <u>matrix of data</u> , inv_monster : <u>matrix of data</u>) → (<u>string</u> , <u>string</u> , <u>string</u> , <u>matrix of data</u> , <u>matrix of data</u> , <u>matrix of data</u>) { Melakukan algoritma battle sesuai spesifikasi yang telah dibuat. } </pre>
<p>KAMUS LOKAL</p> <pre> user , monster , inv_item , inv_monster : <u>matrix of data</u> username , role : <u>string</u> uid , oc , get_oc , num , turn , m : <u>integer</u> level_enemy , level_monster , choice , monster_choice , perintah , pilih : <u>integer</u> temp_atk_enemy , temp_def_enemy , temp_hp_enemy , last_hp_enemy : <u>integer</u> temp_atk_monster , temp_def_monster , temp_hp_monster , leave_atk_monster , leave_def_monster , leave_hp_monster , start_hp_monster , last_hp_monster , random_atk_monster , random_atk_enemy : <u>integer</u> true_choice : <u>array of character</u> choice_option : <u>matrix of integer</u> qty_s , qty_r , qty_h : <u>tuple of integer</u> minum_s , minum_r , minum_h , minum_round , fight , run , check_potion , cek : <u>boolean</u> i : <u>integer</u> {index} procedure show_enemy (<u>input/output</u> : None) { I.S. Tampilan layar masih kosong. } { F.S. Tampilan monster yang akan dilawan muncul ke layar. } procedure show_team_monster(<u>input/ouput</u> : None) { I.S. Tampilan layar masih kosong. } { F.S. Tampilan monster yang dipilih muncul ke layar. } function cint(data : <u>matrix of data</u>) → <u>matrix of data</u> { Mengubah data berupa string of integer menjadi integer. } function cstr(data : <u>matrix of data</u>) → <u>matrix of data</u> { Mengubah data yang berupa anything menjadi string. } </pre>
<p>ALGORITMA</p> <pre> uid ← int(uid) oc ← int(oc) monster ← cint(monster) inv_item ← cint(inv_item) inv_monster = ←(inv_monster) show_enemy() </pre>

```

num ← generate([1 , len(monster)])
level_enemy ← generate([1 , 6])
temp_atk_enemy ← int(monster[num][2] * (1 + (level_enemy - 1)
* 0.1))
temp_def_enemy ← int(monster[num][3] * (1 + (level_enemy - 1)
* 0.1))
temp_hp_enemy ← int(monster[num][4] * (1 + (level_enemy - 1)
* 0.1))
output("RAWRRR, Monster {monster[num][1]} telah muncul !!!")
output("Name\t\t: {monster[num][1]}\nATK Power\t:
{monster[num][2]} --> {temp_atk_enemy}\nDEF Power\t:
{monster[num][3]} --> {temp_def_enemy}\nHP\t\t:
{monster[num][4]} --> {temp_hp_enemy}\nLevel\t\t:
{level_enemy}\n")
output("===== MONSTER LIST =====")
m ← 1
choice_option ← []
i traversal [1 , (len(inv_monster)-1)]
  if (uid = inv_monster[i][0]) then
    output("{m}. {monster[inv_monster[i][1]][1]}")
    choice_option ← choice_option + [(m - 1) , i ,
inv_monster[i][1]]
    m ← m + 1
true_choice ← [i traversal [1 , (m - 1)]]
input(choice)
while (not(choice in true_choice)) do
  input(choice)
monster_choice ← choice_option[choice - 1][1]
level_monster ← inv_monster[monster_choice][2]
temp_atk_monster ←
int(monster[inv_monster[monster_choice][1]][2] * (1 +
(level_monster - 1) * 0.1))
temp_def_monster ←
int(monster[inv_monster[monster_choice][1]][3] * (1 +
(level_monster - 1) * 0.1))
temp_hp_monster ←
int(monster[inv_monster[monster_choice][1]][4] * (1 +
(level_monster - 1) * 0.1))
start_hp_monster ← temp_hp_monster
show_team_monster()
output("RAWRRR, {role} {username} mengeluarkan monster
{monster[inv_monster[monster_choice][1]][1]} !!!")
output("Name: {monster[inv_monster[monster_choice][1]]}")

```

```

output("ATK Power:
{monster[inv_monster[monster_choice][1]][2]} -->
{temp_atk_monster}")
output("DEF Power:
{monster[inv_monster[monster_choice][1]][3]} -->
{temp_def_monster}")
output("HP: {monster[inv_monster[monster_choice][1]][4]} -->
{temp_hp_monster}")
output("Level: {level_monster}")

turn ← 1
minum_s ← False
minum_r ← False
minum_h ← False
while ((temp_hp_monster ≠ 0) and (temp_hp_enemy ≠ 0)) do
    output("===== TURN {turn}
({monster[inv_monster[monster_choice][1]][1]}) =====")
    output("1. Attack\n2. Use Potion\n3. Quit")
    input(perintah)
    while (not(1 ≤ perintah ≤ 3)) do
        input(perintah)
    fight ← False
    minum_round ← False
    run ← False
    while ((perintah ≠ 3) and (not(fight))) do
        while (perintah = 2) do
            cek ← False
            check_potion ← False
            i traversal [1..(len(item)-1)]
            if ((inv_item[i][0] = uid) and (inv_item[i][2] ≠
0) and ((inv_item[i][1] = "strength") or (inv_item[i][1] =
"resilience") or (inv_item[i][1] = "healing"))) then
                cek ← True
                break
            if (not(cek)) then
                output("Anda tidak memiliki Potion dalam
inventory")
            else
                output("===== POTION LIST =====")
            qty_s ← (0 , 0)
            qty_r ← (0 , 0)
            qty_h ← (0 , 0)
            i traversal [1..(len(inv_item)-1)]
            if (inv_item[i][0] == uid) then
                depend on (inv_item[i][1])
                (inv_item[i][1] = "strength") :

```

```

        qty_s = (i , inv_item[i][2])
        (inv_item[i][1] = "resilience") :
            qty_r = (i , inv_item[i][2])
            (inv_item[i][1] = "healing") :
                qty_h = (i , inv_item[i][2])
        output("1. Strength Potion (Qty: {qty_s[1]}) - Increases ATK Power")
        output("2. Resilience Potion (Qty: {qty_r[1]}) - Increases DEF Power")
        output("3. Healing Potion (Qty: {qty_h[1]}) - Restores Health")
        output("4. Cancel")
        input(pilih)
        while (not(1 ≤ pilih ≤ 3)) do
            input(pilih)
            while (not(check_potion)) do
                depend on (pilih)
                pilih = 1 :
                    if (qty_s[1] == 0) then
                        output("Wah, kamu sedang tidak memiliki ramuan ini, silakan pilih ramuan lain!")
                    else
                        if (minum_s) then
                            output("Kamu mencoba memberikan ramuan ini kepada {monster[inv_monster[monster_choice][1]][1]}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.")
                    else
                        inv_item[qty_s[0]][2] ←
                        inv_item[qty_s[0]][2] - 1
                        qty_s ← (qty_s[0] ,
                        inv_item[qty_s[0]][2])
                        check_potion ← True
                        minum_round ← True
                        minum_s ← True
                        leave_atk_monster ←
                        temp_atk_monster
                        temp_atk_monster ←
                        int(temp_atk_monster * 1.05)
                        output("Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi {monster[inv_monster[monster_choice][1]][1]} dan gerakannya menjadi lebih cepat dan mematikan.")

```

```

        output("Update ATK Power :
{leave_atk_monster} --> {temp_atk_monster}")
        pilih = 2 :
        if (qty_r[1] == 0) then
            output("Wah, kamu sedang tidak
memiliki ramuan ini, silakan pilih ramuan lain!")
        else
            if (minum_r) then
                output("Kamu mencoba memberikan
ramuan ini kepada
{monster[inv_monster[monster_choice][1]][1]}, namun dia
menolaknya seolah-olah dia memahami ramuan tersebut sudah
tidak bermanfaat lagi.")
            else
                inv_item[qty_r[0]][2] ←
inv_item[qty_r[0]][2] - 1
                qty_r ← (qty_r[0] ,
inv_item[qty_r[0]][2])
                check_potion ← True
                minum_round ← True
                minum_r ← True
                leave_atk_monster ←
temp_atk_monster
                temp_def_monster ←
int(temp_def_monster * 1.05)
                output("Setelah meminum ramuan
ini, muncul sebuah energi pelindung di sekitar
{monster[inv_monster[monster_choice][1]][1]} yang membuatnya
terlihat semakin tangguh dan sulit dilukai.")

        output("Update DEF Power :
{leave_def_monster} --> {temp_def_monster}")
        pilih = 3 :
        if (qty_h[1] == 0) then
            output("Wah, kamu sedang tidak
memiliki ramuan ini, silakan pilih ramuan lain!")
        else
            if (minum_h) then
                output("Kamu mencoba memberikan
ramuan ini kepada
{monster[inv_monster[monster_choice][1]][1]}, namun dia
menolaknya seolah-olah dia memahami ramuan tersebut sudah
tidak bermanfaat lagi.")
            else
                inv_item[qty_h[0]][2] ←
inv_item[qty_h[0]][2] - 1

```

```

        qty_h ← (qty_h[0] ,
inv_item[qty_h[0]][2])
        check_potion ← True
        minum_round ← True
        minum_h ← True
        leave_hp_monster ← temp_hp_monster
        temp_hp_monster ←
        int(temp_atk_monster +
monster[inv_monster[monster_choice][1]][4] * 1/4)
        if (temp_hp_monster >
start_hp_monster) then temp_hp_monster ← start_hp_monster
        output("Setelah meminum ramuan
ini, aura kekuatan terlihat mengelilingi
{monster[inv_monster[monster_choice][1]][1]} dan gerakannya
menjadi lebih cepat dan mematikan.")
        output("Update HP :
{leave_hp_monster} --> {temp_hp_monster}")
        pilih = 4 :
        output("")
        break
        if (not(check_potion)) then
        output("===== POTION LIST
=====")
        output("1. Strength Potion (Qty:
{qty_s[1]} - Increases ATK Power")
        output("2. Resilience Potion (Qty:
{qty_r[1]} - Increases DEF Power")
        output("3. Healing Potion (Qty: {qty_h[1]}) -
Restores Health")
        output("4. Cancel")
        input(pilih)
        while (not(1 ≤ pilih ≤ 3)) do
        input(pilih)
        if (check_potion) then
        perintah ← 1
        else
        output("===== TURN {turn}
({monster[inv_monster[monster_choice][1]][1]}) =====")
        output("1. Attack\n2. Use Potion\n3. Quit")
        input(perintah)
        while (not(1 ≤ perintah ≤ 3)) do
        input(perintah)
        if (perintah = 1) then
        fight ← True
        if (minum_round) then

```

```

        output("===== TURN {turn}
({monster[num][1]}) =====")
        last_hp_monster ← temp_hp_monster
        random_atk_enemy ← int(temp_atk_enemy * (100 +
generate([-30 , 31]))/100)
        temp_hp_monster ← int(temp_hp_monster -
(random_atk_enemy * (100 - temp_def_monster)/100))
        if (temp_hp_monster ≤ 0) then
            temp_hp_monster ← 0
        output("SCHWINKKK, {monster[num][1]} menyerang
{monster[inv_monster[monster_choice][1]][1]} !!!")
        output("Name\t\t:
{monster[inv_monster[monster_choice][1]][1]}\nATK Power\t:
{temp_atk_monster}\nDEF Power\t: {temp_def_monster}\nHP\t\t:
{last_hp_monster} --> {temp_hp_monster}\nLevel\t\t:
{level_monster}")
        output("#Penjelasan Battle - ATT : Attack =
{random_atk_enemy} ; Reduced by = {random_atk_enemy *
(temp_def_monster)/100} ({temp_def_monster}%) ; Attack
Results = {random_atk_enemy * (100 - temp_def_monster)/100}")
        output("#Penjelasan Battle - HP : Attack Results
= {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP
= {last_hp_monster} ; Final HP = FLOOR({last_hp_monster} -
{random_atk_enemy * (100 - temp_def_monster)/100}) =
{temp_hp_monster}")
        if (temp_hp_monster = 0) then
            break
        else
            last_hp_enemy ← temp_hp_enemy
            random_atk_monster ← int(temp_atk_monster * (100 +
generate([-30 , 31]))/100)
            temp_hp_enemy ← int(temp_hp_enemy -
(random_atk_monster * (100 - temp_def_enemy)/100))
            if (temp_hp_enemy ≤ 0) then
                temp_hp_enemy ← 0
            output("
\nSCHWINKKK,
{monster[inv_monster[monster_choice][1]][1]} menyerang
{monster[num][1]} !!!")
            output("Name\t\t: {monster[num][1]}\nATK Power\t:
{temp_atk_enemy}\nDEF Power\t: {temp_def_enemy}\nHP\t\t:
{last_hp_enemy} --> {temp_hp_enemy}\nLevel\t\t:
{level_enemy}")
            output("#Penjelasan Battle - ATT : Attack =
{random_atk_monster} ; Reduced by = {random_atk_monster *

```

```

(temp_def_enemy)/100} ({temp_def_enemy}%) ; Attack Results =
{random_atk_monster * (100 - temp_def_enemy)/100}")
    output("#Penjelasan Battle - HP : Attack Results
= {random_atk_monster * (100 - temp_def_enemy)/100} ; Last HP
= {last_hp_enemy} ; Final HP = FLOOR({last_hp_enemy} -
{random_atk_monster * (100 - temp_def_enemy)/100}) =
{temp_hp_enemy}\n")
    if (temp_hp_enemy = 0) then
        break
    else
        output("===== TURN {turn}
({monster[num][1]}) =====")
        last_hp_monster ← temp_hp_monster
        random_atk_enemy ← int(temp_atk_enemy * (100 +
generate([-30 , 31]))/100)
        temp_hp_monster ← int(temp_hp_monster -
(random_atk_enemy * (100 - temp_def_monster)/100))
        if (temp_hp_monster ≤ 0) then
            temp_hp_monster ← 0
        output("SCHWINKKK, {monster[num][1]} menyerang
{monster[inv_monster[monster_choice][1]][1]} !!!")
        output("Name\t\t:
{monster[inv_monster[monster_choice][1]][1]}\nATK Power\t:
{temp_atk_monster}\nDEF Power\t: {temp_def_monster}\nHP\t\t:
{last_hp_monster} --> {temp_hp_monster}\nLevel\t\t:
{level_monster}")
        output("#Penjelasan Battle - ATT : Attack =
{random_atk_enemy} ; Reduced by = {random_atk_enemy *
(temp_def_monster)/100} ({temp_def_monster}%) ; Attack
Results = {random_atk_enemy * (100 - temp_def_monster)/100}")
        output("#Penjelasan Battle - HP : Attack Results
= {random_atk_enemy * (100 - temp_def_monster)/100} ; Last HP
= {last_hp_monster} ; Final HP = FLOOR({last_hp_monster} -
{random_atk_enemy * (100 - temp_def_monster)/100}) =
{temp_hp_monster}")
        if (temp_hp_monster = 0) then
            break
        else if (perintah ≠ 3) then
            output("Perintah tidak ditemukan.")
            input(perintah)
            while ((perintah ≠ 1) and (perintah ≠ 2) and
(perintah ≠ 3)) do
                input(perintah)
            if (perintah = 3) then
                run ← True

```

```

    output("Anda berhasil kabur dari BATTLE")
    break
else
    turn ← turn + 1
if ((temp_hp_enemy = 0) and (not(run))) then
    output("Selamat, Anda berhasil mengalahkan monster
{monster[num][1]} !!!")
    get_oc ← generate([5 , 31])
    oc ← oc + get_oc
    output("Total OC yang diperoleh : {get_oc}")
else if ((temp_hp_monster = 0) and (not(run))) then
    output("Yahhh, Anda dikalahkan monster {monster[num][1]}.
Jangan menyerah, coba lagi !!!")
    uid ← str(uid)
    oc ← str(oc)
    monster ← cstr(monster)
    inv_item ← cstr(inv_item)
    inv_monster ← cstr(inv_monster)
→ (uid , username , role , oc , monster , inv_item ,
inv_monster)

```

REALISASI FUNGSI & PROSEDUR

```

procedure show_enemy (input/output : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang akan dilawan muncul ke layar. }
KAMUS LOKAL
-

```

ALGORITMA LOKAL

```

    output("")
    output("      / \----/ \      ")
    output("      / \      \ / \ ")
    output("      |   O   O   |   |   |   ")
    output("      |   .vvvvv.   |   |   |   ")
    output("      /   |   |   \   |   |   ")
    output("      /   ^^^^^^   \   |   |   ")
    output("      . /   / |   |   \   |   |   ")
    output("      /   / |   |   | \   |   / ")
    output("      \   / |   |   |   | \   |   ")
    output("      \   |   |   |   |   |   ")
    output("      . -' -' -' -' -' . ' ")
    output("      _ -' -' -' -' -' . ' ")
→ None

```

```

procedure show_team_monster(input/outut : None)

```

```

{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang dipilih muncul ke layar. }
KAMUS LOKAL
-

```

ALGORITMA LOKAL

```

    output("")

```

```

        /\---/\_   ")
output("   /     \   / | ")
output("   | | O   O | / | ")
output("   | | .vvvvv.| / / ")
output("   / | |   |   / ")
output("   / | `^^^^^   / ")
output("   | / |   |   / ")
output("   / |   |   |   | ")
output("   \ |   |   |   | ")
output("   | |   |   |   | ")
output("   \_\ \_\ \_\ \_\ " )
→ None
function cint(data : matrix of data) → matrix of data
{ Mengubah data berupa string of integer menjadi integer. }
KAMUS LOKAL
    data : matrix of data
    num_char : list of character
    i , j : integer {index}
ALGORITMA LOKAL
    num_char ← '0123456789'
    i traversal [1..(len(data)-1)]
        j traversal [0..(len(data[i])-1)]
            if ((data[i][j][0]) in (num_char)) then
                data[i][j] ← int(data[i][j])
            else
                if (data[i][j][0] = '-') then
                    if ((data[i][j][1]) in (num_char)) then
                        data[i][j] ← int(data[i][j])
    → data
function cstr(data : matrix of data) → matrix of data
{ Mengubah data yang berupa anything menjadi string. }
KAMUS LOKAL
    data : matrix of data
    i , j : integer {index}
ALGORITMA LOKAL
    output("")
    i traversal [1..(len(data)-1)]
        j traversal [0..(len(data[i])-1)]
            data[i][j] ← str(data[i][j])
    → data

```

10. F09 - Arena

Tabel 6.10 Spesifikasi fungsi dalam program Arena (F09)

```

function arena(uid : string , username : string , role :
string , oc : string , monster : matrix of data , inv_item :
matrix of data , inv_monster : matrix of data) → (string ,
string , string , matrix of data , matrix of data ,
matrix of data)

```

```
{ Melakukan algoritma arena sesuai spesifikasi yang telah dibuat. }
```

KAMUS LOKAL

```
user , monster , inv_item , inv_monster : matrix of data
username , role : string
uid , oc , get_oc , num , turn , m : integer
level_enemy , level_monster , choice , monster_choice ,
    perintah , pilih : integer
temp_atk_enemy , temp_def_enemy , temp_hp_enemy ,
    last_hp_enemy : integer
temp_atk_monster , temp_def_monster , temp_hp_monster ,
    leave_atk_monster , leave_def_monster , leave_hp_monster ,
    start_hp_monster , last_hp_monster , random_atk_monster ,
    random_atk_enemy : integer
true_choice : array of character
choice_option : matrix of integer
qty_s , qty_r , qty_h : tuple of integer
minum_s , minum_r , minum_h , minum_round , fight , run ,
    check_potion , cek : boolean
i : integer {index}
procedure show_enemy (input/output : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang akan dilawan muncul ke layar. }
procedure show_team_monster(input/output : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang dipilih muncul ke layar. }
function cint(data : matrix of data) → matrix of data
{ Mengubah data berupa string of integer menjadi integer. }
function cstr(data : matrix of data) → matrix of data
{ Mengubah data yang berupa anything menjadi string. }
```

ALGORITMA

```
import time
function arena(uid , username , role , oc , monster , inv_item ,
inv_monster)
    output("Selamat datang di Arena !!!")
    output("")
    uid ← int(uid)
    oc ← int(oc)
    monster ← cint(monster)
    inv_item ← cint(inv_item)
    inv_monster ← cint(inv_monster)
    time.sleep(1)
    output("===== MONSTER LIST =====")
    m ← 1
    choice_option ← []
    i traversal[2..len(inv_monster)]
```

```

if (uid = inv_monster[i][0]) then
    output(m, ".", monster[inv_monster[i][1]][1])
    choice_option ← choice_option + [[m - 1, i,
    inv_monster[i][1]]]
    m ← m + 1
output("Pilih monster untuk bertarung (masukkan angka saja)
: ")
input(choice)
while not(1 ≤ choice ≤ m - 1) do
    output("Pilih monster untuk bertarung (masukkan angka
saja): ")
    input(choice)
monster_choice ← choice_option[choice - 1][1]
level_monster ← inv_monster[monster_choice][2]
temp_atk_monster ←
int(monster[inv_monster[monster_choice][1]][2] * (1 +
(level_monster - 1) * 0.1))
temp_def_monster ←
int(monster[inv_monster[monster_choice][1]][3] * (1 +
(level_monster - 1) * 0.1))
temp_hp_monster ←
int(monster[inv_monster[monster_choice][1]][4] * (1 +
(level_monster - 1) * 0.1))
start_hp_monster ← temp_hp_monster
time.sleep(1)
show_team_monster()
time.sleep(1)
output("")
output("RAWRRR,", role, username, "mengeluarkan monster",
monster[inv_monster[monster_choice][1]][1], "!!!!")
output("Name      : ",
monster[inv_monster[monster_choice][1]][1])
output("ATK Power  : ",
monster[inv_monster[monster_choice][1]][2], "-->",
temp_atk_monster)
output("DEF Power  : ",
monster[inv_monster[monster_choice][1]][3], "-->",
temp_def_monster)
output("HP         : ",
monster[inv_monster[monster_choice][1]][4], "-->",
temp_hp_monster)
output("Level      : ", level_monster")
output("")
stage ← 1
get_oc ← 0

```

```

total_turn ← 0
total_potion ← 0
total_healing ← 0
total_damage_deal ← 0
total_damage_receive ← 0
over ← False
while (stage ≤ 5) and not(over) do
    turn ← 1
    minum_s ← False
    minum_r ← False
    minum_h ← False
    temp_hp_monster ← start_hp_monster
    time.sleep(2)
    output("===== STAGE", stage, " =====")
    time.sleep(1)
    show_enemy()
    num ← generate([1, len(monster)])
    level_enemy ← stage
    temp_atk_enemy ← int(monster[num][2] * (1 + (level_enemy
        - 1) * 0.1))
    temp_def_enemy ← int(monster[num][3] * (1 + (level_enemy
        - 1) * 0.1))
    temp_hp_enemy ← int(monster[num][4] * (1 + (level_enemy
        - 1) * 0.1))
    time.sleep(1)
    output("")
    output("RAWRRR, Monster", monster[num][1] "telah muncul
    !!!")
    output("Name : ", monster[num][1])
    output("ATK Power : ", monster[num][2], "-->",
    temp_atk_enemy)
    output("DEF Power : ", monster[num][3], "-->",
    temp_def_enemy)
    output("HP : ", monster[num][4], "-->",
    temp_hp_enemy)
    output("Level : ", level_enemy)
    while ((temp_hp_monster ≠ 0) and (temp_hp_enemy ≠ 0))do
        time.sleep(1)
        output("")
        output("===== STAGE ", stage, "- TURN ",
        turn, (monster[inv_monster[monster_choice][1]][1]),
        "=====")
        output("1. Attack")
        output("2. Use Potion")
        output("3. Quit")

```

```

output("Pilih perintah (masukkan angka saja) : ")
input(perintah)
while not (1 ≤ perintah ≤ 3) do
    output("Perintah tidak ditemukan.")
    output("Pilih perintah (masukkan angka saja) :
    ")
    input(perintah)
fight ← False
minum_round ← False
run ← False
while (perintah ≠ 3) and (not(fight)) do
    while (perintah = 2) do
        cek ← False
        check_potion ← False
        i traversal[2.. len(inv_item) ]
        depend on(inv_item, cek)
            (inv_item[i][0] = uid) and
            (inv_item[i][2] ≠ 0) and
            ((inv_item[i][1] = "strength") or
            (inv_item[i][1] = "resilience") or
            (inv_item[i][1] = "healing")) :
                cek ← True
                break
        not(cek) :
            output("Anda tidak memiliki Potion
            dalam inventory.")
    else
        time.sleep(1)
        output("")
        output("===== POTION LIST
=====")
        qty_s ← (0 , 0)
        qty_r ← (0 , 0)
        qty_h ← (0 , 0)
        i traversal[2..len(inv_item) ]
        depend on(inv_item)
            (inv_item[i][0] = uid) and
            (inv_item[i][1] = "strength") :
                qty_s ← (i , inv_item[i][2])
            (inv_item[i][0] = uid) and
            (inv_item[i][1] = "resilience") :
                qty_r ← (i , inv_item[i][2])
            (inv_item[i][0] == uid) and
            (inv_item[i][1] == "healing") :
                qty_h ← (i , inv_item[i][2])

```

```

output("1. Strength Potion (Qty: ",
qty_s[1], " ) - Increases ATK Power")
output("2. Resilience Potion (Qty: ",
qty_r[1], " ) - Increases DEF Power")
output("3. Healing Potion (Qty: ",
qty_h[1], " ) - Restores Health")
output("4. Cancel")
output("Pilih perintah (masukkan angka
saja) : ")
while not (1 ≤ pilih ≤ 4) do
    output("Perintah tidak ditemukan.")
    output("Pilih perintah (masukkan
    angka saja) : ")
    while (not(check_potion)) do
    depend on(pilih)
    (pilih = 1):
        depend on(qty_s)
        (qty_s[1] = 0) :
            output("Wah, kamu sedang
            tidak memiliki ramuan ini,
            silakan pilih ramuan lain!")
            output("")
        (qty_s[1] ≠ 0) and (minum_s) :
            output("Kamu mencoba
            memberikan ramuan ini kepada
            Pikachu, namun dia
            menolaknya seolah-olah dia
            memahami ramuan tersebut
            sudah tidak bermanfaat
            lagi")
            output("")
    else
        inv_item[qty_s[0]][2] ←
        inv_item[qty_s[0]][2] - 1
        qty_s ← (qty_s[0] ,
        inv_item[qty_s[0]][2])
        total_potion ← total_potion +1
        check_potion ← True
        minum_round ← True
        minum_s ← True
        leave_atk_monster ←
        temp_atk_monster
        temp_atk_monster =

```

```

        int(temp_atk_monster * 1.05)
        output("")
        output("Setelah meminum ramuan
ini, aura kekuatan terlihat
mengelilingi", monster[inv_
_monster[monster_choice][1]][1
], "dan gerakannya menjadi
lebih cepat dan mematikan.")
        output("Update ATK Power :",
leave_atk_monster, "-->",
temp_atk_monster")
        output("")
        time.sleep(1)
(pilih = 2) :
    depend_on(qty_r)
    (qty_r[1] = 0) :
        output("Wah, kamu sedang
tidak memiliki ramuan ini,
silakan pilih ramuan lain!")
        output("")
    (qty_r[1] ≠ 0) and (minum_r) :
        output("Kamu mencoba
memberikan ramuan ini kepada
Pikachow, namun dia
menolaknya seolah-olah dia
memahami ramuan tersebut
sudah tidak bermanfaat
lagi.")
        output("")
else
    inv_item[qty_r[0]][2] ←
    inv_item[qty_r[0]][2] - 1
    qty_r ← (qty_r[0] ,
    inv_item[qty_r[0]][2])
    total_potion ← total_potion +
    1
    check_potion ← True
    minum_round ← True
    minum_r ← True
    leave_def_monster ←
    temp_def_monster
    temp_def_monster ←
    int(temp_def_monster * 1.05)
    output("")
    output("Setelah meminum ramuan

```

```

ini, muncul sebuah energi
pelindung di sekitar",
monster[inv_monster[monster_c
hoic][1]][1], "yang
membuatnya terlihat semakin
tangguh dan sulit dilukai.")
output("Update DEF Power :
leave_def_monste, "-->",
temp_def_monster)
output("")
time.sleep(1)
(pilih == 3) :
depend_on(qty_h)
(qty_h[1] = 0) :
output("Wah, kamu sedang
tidak memiliki ramuan ini,
silakan pilih ramuan
lain!")
output("")
(qty_h[1] ≠ 0) and (minum_h) :
output("Kamu mencoba
memberikan ramuan ini kepada
Pikachow, namun dia
menolaknya seolah-olah dia
memahami ramuan tersebut
sudah tidak bermanfaat
lagi.")
else
    inv_item[qty_h[0]][2] ←
    inv_item[qty_h[0]][2] - 1
    qty_h ← (qty_h[0] ,
    inv_item[qty_h[0]][2])
    total_potion ← total_potion +
    1
    check_potion ← True
    minum_round ← True
    minum_h ← True
    leave_hp_monster ←
    temp_hp_monster
    temp_hp_monster ←
    int(temp_hp_monster +
    monster[inv_monster[monster_
choice][1]][4] * 1/4)
    if (temp_hp_monster >
    start_hp_monster) :

```

```

        temp_hp_monster ←
        start_hp_monster
        total_healing ← total_healing
        + temp_hp_monster -
        leave_hp_monster
        output("")
        output("Setelah meminum
        ramuan ini, luka-luka yang ada
        di dalam tubuh", monster[inv_
        monster[monster_choice][1]]
        [1], "sembuh dengan
        cepat.")
        output("Dalam sekejap",
        monster[inv_monster[monster_c
        hoice][1]][1], "terlihat
        kembali prima dan siap
        melanjutkan pertempuran.")
        output("Update HP : ",
        leave_hp_monster, "-->",
        temp_hp_monster)
        output("")
        time.sleep(1)
else
        output("")
        break
if not(check_potion) then
        time.sleep(1)
        output("===== POTION LIST
=====")
        output("1. Strength Potion
        (Qty:", qty_s[1], ") - Increases
        ATK Power")
        output("2. Resilience Potion
        (Qty:", qty_r[1], ") - Increases
        DEF Power")
        output(f"3. Healing Potion
        (Qty:", qty_h[1], ") - Restores
        Health")
        output("4. Cancel")
        output("Pilih perintah (masukkan
        angka saja) : ")
        input(pilih)
while not (1 ≤ pilih ≤ 4) do
        output("Perintah tidak
        ditemukan.")

```

```

        output("Pilih perintah (masukkan
        angka saja) : ")
        input(pilih)
        if (check_potion) then
            perintah ← 1
        else
            time.sleep(1)
            output("===== STAGE", stage, "-
            TURN", turn, "(",
            monster[inv_monster[monster_choice]
            [1]][1], ") =====")
            output("1. Attack")
            output("2. Use Potion")
            output("3. Quit")
            output("Pilih perintah (masukkan angka
            saja) : ")
            input(perintah)
            while not (1 ≤ perintah ≤ 3) do
                output("Perintah tidak ditemukan.")
                output("Pilih perintah (masukkan
                angka saja) : ")
                input(perintah)
            if (perintah = 1) then
                fight ← True
                if (minum_round) then
                    output("===== STAGE", stage, "-
                    TURN", turn, "(, monster[num][1], ")
                    =====")
                    last_hp_monster ← temp_hp_monster
                    temp_hp_monster ← int(temp_hp_monster -
                    (temp_atk_enemy * (100 -
                    temp_def_monster)/100))
                    if (temp_hp_monster ≤ 0) then
                        temp_hp_monster ← 0
                    total_damage_receive ←
                    total_damage_receive + last_hp_monster -
                    temp_hp_monster
                    output("SCHWINKKK,", monster[num][1],
                    "menyerang", monster[inv_monster
                    [monster_choice][1]][1], "!!!!")
                    output("Name : ",
                    monster[inv_monster[monster_choice][1]][1])
                    output("ATK Power : ",
                    temp_atk_monster)

```

```

        output("DEF Power      : ",
temp_def_monster)
        output("HP           : ",
temp_hp_monster)
Level: {level_monster}")
        output(f"#Penjelasan Battle - ATT :
Attack = {temp_atk_enemy} ; Reduced by = {temp_atk_enemy *
(temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results =
{temp_atk_enemy * (100 - temp_def_monster)/100}")
        output(f"#Penjelasan Battle - HP :
Attack Results = {temp_atk_enemy * (100 - temp_def_monster)/100}
; Last HP = {last_hp_monster} ; Final HP =
FLOOR({last_hp_monster} - {temp_atk_enemy * (100 -
temp_def_monster)/100}) = {temp_hp_monster}")
        time.sleep(1)
        if (temp_hp_monster == 0)then
            break
        else :
            time.sleep(1)
            last_hp_enemy = temp_hp_enemy
            temp_hp_enemy = int(temp_hp_enemy -
(temp_atk_monster * (100 - temp_def_enemy)/100))
            if (temp_hp_enemy ≤ 0)then
                temp_hp_enemy = 0
                total_damage_deal = total_damage_deal +
last_hp_enemy - temp_hp_enemy
            output(f"\nSCHWINKKK,
{monster[inv_monster[monster_choice][1]][1]} menyerang
{monster[num][1]} !!!")
            output(f"Name\t\t:
{monster[num][1]}\nATK Power\t: {temp_atk_enemy}\nDEF Power\t:
{temp_def_enemy}\nHP\t\t: {temp_hp_enemy}\nLevel\t\t:
{level_enemy}")
            output(f"#Penjelasan Battle - ATT :
Attack = {temp_atk_monster} ; Reduced by = {temp_atk_monster *
(temp_def_enemy)/100} ({temp_def_enemy}%) ; Attack Results =
{temp_atk_monster * (100 - temp_def_enemy)/100}")
            output(f"#Penjelasan Battle - HP :
Attack Results = {temp_atk_monster * (100 - temp_def_enemy)/100}
; Last HP = {last_hp_enemy} ; Final HP = FLOOR({last_hp_enemy} -
{temp_atk_monster * (100 - temp_def_enemy)/100}) =
{temp_hp_enemy}\n")
            time.sleep(1)
            if (temp_hp_enemy == 0)then :
                break

```

```

    else
        output(f"===== STAGE {stage}
- TURN {turn} ({monster[num][1]}) =====")
        last_hp_monster = temp_hp_monster
        temp_hp_monster =
        int(temp_hp_monster - (temp_atk_enemy * (100 -
        temp_def_monster)/100))
            if (temp_hp_monster ≤ 0)then
                temp_hp_monster = 0
                total_damage_receive =
                total_damage_receive + last_hp_monster - temp_hp_monster
                output(f"SCHWINKKK,
{monster[num][1]} menyerang
{monster[inv_monster[monster_choice][1]][1]} !!!!")
                output(f"Name\t\t:
{monster[inv_monster[monster_choice][1]][1]}\nATK Power\t:
{temp_atk_monster}\nDEF Power\t: {temp_def_monster}\nHP\t\t:
{temp_hp_monster}\nLevel\t\t: {level_monster}")
                output(f"#Penjelasan Battle - ATT :
Attack = {temp_atk_enemy} ; Reduced by = {temp_atk_enemy *
(temp_def_monster)/100} ({temp_def_monster}%) ; Attack Results =
{temp_atk_enemy * (100 - temp_def_monster)/100}")
                output(f"#Penjelasan Battle - HP :
Attack Results = {temp_atk_enemy * (100 - temp_def_monster)/100}
; Last HP = {last_hp_monster} ; Final HP =
FLOOR({last_hp_monster} - {temp_atk_enemy * (100 -
temp_def_monster)/100}) = {temp_hp_monster}")
                time.sleep(1)
                if (temp_hp_monster == 0)then
                    break
                else if (perintah != 3) then
                    print("Perintah tidak ditemukan.")
                    perintah = int(input("Pilih perintah
(masukkan angka saja) : "))
                if (perintah == 3) then
                    run = True
                    over = True
                    output("Anda berhasil kabur dari BATTLE")
                    break
                else
                    turn <- turn + 1
                    total_turn <- total_turn + 1
                if ((temp_hp_enemy = 0) and (not(run)))then
                    time.sleep(1)

```

```

        print(f"Selamat, Anda berhasil mengalahkan monster
{monster[num][1]} !!!")
        total_healing = total_healing + start_hp_monster -
temp_hp_monster
        get_ocs = generate([20 * stage , 40 * stage + 1])
        get_oc = get_oc + get_ocs
        stage = stage + 1
        print(f"STAGE CLEARED! Anda akan mendapatkan
{get_ocs} OC pada sesi ini.\n")
        print("Memulai stage berikutnya...\n")
elif ((temp_hp_monster == 0) and (not(run))) :
    time.sleep(1)
    over = True
    print(f"\nYahhh, Anda dikalahkan monster
{monster[num][1]}. Jangan menyerah, coba lagi !!!")
    oc = oc + get_oc
    time.sleep(1)
if ((not(run)) and (not(over))) :
    print("\nSelamat, Anda berhasil menyelesaikan seluruh
stage Arena !!!\n")
    time.sleep(1)
    print("===== STATS =====")
    print(f"Total Hadiyah\t\t: {get_oc} OC")
    print(f"Jumlah Turn\t\t: {total_turn} Turn")
    print(f"Jumlah Stage\t\t: {stage - 1} Stage")
    print(f"Potion Digunakan\t\t: {total_potion} Potion")
    print(f"HP Recovered\t\t: {total_healing} HP")
    print(f"Damage Diberikan\t\t: {total_damage_deal} Damage")
    print(f"Damage Diterima\t\t: {total_damage_receive}
Damage")
elif ((not(run)) or (not(over))) :
    print(f"GAME OVER! Sesi latihan berakhir pada stage
{stage}!\n")
    time.sleep(1)
    print("===== STATS =====")
    print(f"Total Hadiyah\t\t: {get_oc} OC")
    print(f"Jumlah Turn\t\t: {total_turn} Turn")
    print(f"Jumlah Stage\t\t: {stage - 1} Stage")
    print(f"Potion Digunakan\t\t: {total_potion} Potion")
    print(f"HP Recovered\t\t: {total_healing} HP")
    print(f"Damage Diberikan\t\t: {total_damage_deal} Damage")
    print(f"Damage Diterima\t\t: {total_damage_receive}
Damage")
else :
    print("\nGAME OVER! Anda mengakhiri sesi latihan!\n")

```

```
time.sleep(1)
print("===== STATS =====")
print(f"Total Hadiah\t\t: {get_oc} OC")
print(f"Jumlah Turn\t\t: {total_turn} Turn")
print(f"Jumlah Stage\t\t: {stage - 1} Stage")
print(f"Potion Digunakan\t: {total_potion} Potion")
print(f"HP Recovered\t\t: {total_healing} HP")
print(f"Damage Diberikan\t: {total_damage_deal} Damage")
print(f"Damage Diterima\t\t: {total_damage_receive} Damage")
Damage")
uid = str(uid)
oc = str(oc)
monster = cstr(monster)
inv_item = cstr(inv_item)
inv_monster = cstr(inv_monster)
return (uid , username , role , oc , monster , inv_item ,
inv_monster)
```

REALISASI FUNGSI & PROSEDUR

```
procedure show_enemy (input/output : None)
{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang akan dilawan muncul ke layar. }
KAMUS LOKAL
```

ALGORITMA LOKAL

```
output("")  
output("      / \---/ \      ")  
output("      /   \   \ / \  ")  
output("      |   O   O   |   |   | ")  
output("      |   .vvvvv.   |   |   | ")  
output("      /   |   |   \   |   |   | ")  
output("      /   `^^^^^'   \   |   |   | ")  
output("      . /   / |           \   |   |_ ")  
output("      /   /   |           | \_   / ")  
output("      \   /   |           |   | \_ |_ ")  
output("      ,   |           |   |   | ")  
output("      . - - ' - - ' - - . ' - ")  
output("      _ . - ' - - ' - - . ' - . _ ")  
→ None
```

```
procedure show team monster (input/ouput : None)
```

{ I.S. Tampilan layar masih kosong. }
{ F.S. Tampilan monster yang dipilih muncul ke layar. }
KAMUS LOKAL

ALGORITMÁ LOKAL

```
output("")  
output("      /\\---/\\_      ")  
output("      /      \\      / | ")  
output("      |      O      O      | / | ")
```

```

output("   | | .vvvvv. | / /")
output("   / | |   | /")
output("   / | ^^^^^^ /")
output(" | / | |   /")
output(" / | |   | |")
output(" \ | | | |")
output(" | | | | |")
output(" \_. \ \_. \ ")
→ None
function cint(data : matrix of data) → matrix of data
{ Mengubah data berupa string of integer menjadi integer. }
KAMUS LOKAL
    data : matrix of data
    num_char : list of character
    i , j : integer {index}
ALGORITMA LOKAL
    num_char ← '0123456789'
    i traversal [1..(len(data)-1)]
        j traversal [0..(len(data[i])-1)]
            if ((data[i][j][0]) in (num_char)) then
                data[i][j] ← int(data[i][j])
            else
                if (data[i][j][0] = '-') then
                    if ((data[i][j][1]) in (num_char)) then
                        data[i][j] ← int(data[i][j])
    → data
function cstr(data : matrix of data) → matrix of data
{ Mengubah data yang berupa anything menjadi string. }
KAMUS LOKAL
    data : matrix of data
    i , j : integer {index}
ALGORITMA LOKAL
    output("")
    i traversal [1..(len(data)-1)]
        j traversal [0..(len(data[i])-1)]
            data[i][j] ← str(data[i][j])
    → data

```

11. F10 - Shop & Currency

Tabel 6.11 Spesifikasi fungsi dalam program Shop & Currency (F10)

```

function (Monstershop : array of array of string,
DataMonster : array of array of string, ItemShop : array of
array of string, item_inventory : array of array of string,
user_login: array of string, monsterinventory : array of
array of string)

```

```
{Fungsi untuk menampilkan daftar barang yang tersedia di  
toko dan melakukan transaksi pembelian.}
```

KAMUS LOKAL

```
AdaItem : boolean  
IsDijual : boolean  
Monstershop : array of array of string  
DataMonster : array of array of string  
ItemShop : array of array of string  
item_inventory : array of array of string  
user_login : array of array of string  
monsterinventory : array of array of string  
Aksi : string  
LihatApa : string  
beli : string  
IdPotion : string  
JumlahBeli : string  
MonsterTerpilih : string  
IdMonster : string  
Urutan, urutan, urutanterpilih : integer  
Isi : array of string  
IsiMonster : array of string  
isiinventory : array of string  
MonsterID : array of string
```

```

Monster : array of string

function slicing(Data : any, Awal : integer , Akhir : integer) -> any

{Menerima input list untuk diambil hanya sebagian dari suatu
list}

function IsDigit(string : string) -> bool:

{Menerima input string untuk cek apakah isi dari string
tersebut merupakan bilangan atau bukan}

function ManualSplit (string : string, b : character) -> array of string:

{Menerima input string untuk mengubahnya menjadi list dimana
isinya berupa string yang terpisah-pisah oleh}

function RemoveNewLine(string : string) -> string :

{Menerima input string untuk menghilangkan \n}

```

ALGORITMA

```

input(Aksi)
while True do
    while Aksi.lower() ≠ "lihat" and Aksi.lower() ≠ "beli" and
Aksi.lower() ≠ "keluar" do
        output("\nInput tidak valid, mohon input ulang\n")
        input(Aksi)
    if Aksi.lower() = "keluar" then
        break
    else
        if Aksi.lower() = "lihat" then
            input(LihatApa)
            while LihatApa.lower() ≠ "monster" and LihatApa.lower() ≠
"potion" do
                output("\nInput tidak valid, mohon input ulang\n")
                input(LihatApa)
            if LihatApa.lower() = "monster" then
                if len(data_shop_monster) ≤ 1 then
                    output("\nSudah tidak ada monster yang terjual, datang
lagi di lain waktu!\n")

```



```

        if data_item_shop[1][1] = "0" and data_item_shop[2][1] =
"0" and data_item_shop[3][1] = "0" then
            output("\nSudah tidak ada potion yang dapat dibeli di
shop, datang lagi di lain waktu!\n")
        else
            input(IdPotion)
            if not IsDigit(IdPotion) or IdPotion not in (chr(i)
i traversal[49..51] then
                output("\nInput tidak valid, mohon input ulang\n")
            else if data_item_shop[int(IdPotion)][1] = "0" then
                output("\nMaaf, stok potion tersebut telah
habis\n")
            else
                IdPotion ← int(IdPotion)
                input(JumlahBeli)
                if not IsDigit(JumlahBeli) then
                    output("\ninput invalid, mohon input ulang\n")
                else
                    JumlahBeli ← int(JumlahBeli)
                    if int(JumlahBeli) < 0 or int(JumlahBeli) >
int(data_item_shop[IdPotion][1]) then
                        if int(JumlahBeli) < 0 then
                            output("\ninput invalid, mohon input
ulang\n")
                        else
                            output("\nJumlah barang yang mau dibeli
melebihi stok, mohon input ulang\n")
                        else if int(JumlahBeli) *
int(data_item_shop[IdPotion][2]) > int(user_loggedin[4]) then
                            output("\nOC-mu tidak cukup\n")
                        else
                            user_loggedin[4] ← str(int(user_loggedin[4]) -
(int(JumlahBeli) * int(data_item_shop[IdPotion][2]))) # Mulai
proses pembelian Potion
                            data_item_shop[IdPotion][1] ←
str(int(data_item_shop[IdPotion][1]) - JumlahBeli)
                            output(f"\nBerhasil membeli item: {JumlahBeli}
{data_item_shop[IdPotion][0]}. Item sudah masuk ke
inventory-mu!")
                            output(f"OC-mu tersisa
{int(user_loggedin[4])}\n")
                            AdaItem ← False
                            isininventory in data_invent_item
                            if isininventory[0] = user_loggedin[0] then
# cek id user di inventory dengan id user ter-login

```

```

        if isiinventory[1] =
data_item_shop[IdPotion][0] then # Kepemilikan potion yang
dibelii
            isiinventory[2] ←
str(int(isiinventory[2]) + JumlahBeli)
            AdaItem ← True
            if not AdaItem then
                TambahTypePotion ← []
                TambahTypePotion ← [user_loggedin[0],
data_item_shop[IdPotion][0], str(JumlahBeli)]
                data_invent_item.append(TambahTypePotion)
            else if beli = "monster" then # Beli MONSTER
                if len(data_shop_monster) ≤ 1 then
                    output("\nSudah tidak ada monster yang bisa dibeli
dari shop, datang lagi di lain waktu!\n")
                else
                    input(IdMonster)
                    IsDijual ← False
                    urutan traversal [1..len(data_shop_monster)]
                    MonsterID ← data_shop_monster[urutan]
                    if MonsterID[0] = IdMonster then
                        IsDijual ← True
                        MonsterTerpilih ← MonsterID
                        urutanterpilih ← urutan
                        break
                    depend on (IsDijual, MonsterTerpilih,
user_loggedin[0])
                    not IsDijual:
                        output("\nInput tidak valid, mohon input
ulang\n")
                    MonsterTerpilih[1] = "0":
                        output("\nStok sudah habis, silakan beli di lain
waktu!\n")
                    [user_loggedin[0], MonsterTerpilih[0]] in
(slicing(data_invent_monster[i], 0, 2) i traversal
[1..(len(data_invent_monster))]):
                        output("\nAnda telah memiliki monster tersebut,
mohon pilih monster yang lain\n")
                    else
                        if user_loggedin[4] < MonsterTerpilih[2] then
                            output("\nOOC-mu tidak cukup\n")
                        else # Proses Pembelian monster
                            user_loggedin[4] ← str(int(user_loggedin[4]) -
int(MonsterTerpilih[2]))

```

```

        data_shop_monster[urutanterpilih][1] ←
str(int(data_shop_monster[urutanterpilih][1]) - 1)
        Monster in data_monster:
            if MonsterTerpilih[0] = Monster[0] then
                output(f"Berhasil membeli monster:
{Monster[1]}. Item sudah masuk ke inventory-mu!")
                break
            data_invent_monster.append([user_loggedin[0],
MonsterTerpilih[0], "1"])
            input(Aksi)
            output("\nMr. Yanto bilang terima kasih, belanja di yanto
shop lagi ya nanti :)\n")

→ data_shop_monster, data_monster, data_item_shop,
data_invent_item, user_loggedin, data_invent_monster

```

REALISASI FUNGSI & PROSEDUR

```

function slicing(Data : any, Awal : integer , Akhir : integer)
-> any:
    {SPESIFIKASI LOKAL}
    {Menerima input list untuk diambil hanya sebagian dari suatu
list}
    {I.S. : menerima input data yang dapat berupa array ataupun
string}
    {F.S. : mengeluarkan sebagian dari input awal array atau
string}

    {KAMUS LOKAL}
    {Data : any}
    {Awal, Akhir : integer}
    {String, IsiArray : string}
    {Urut : integer}
    {Array : array of string}

    {ALGORITMA LOKAL}
    if (type(Data) = string) then
        String ← ""
        Urut traversal [1..len(Data)]
        char = Data[Urut]
        if Urut ≥ Awal and Urut < Akhir then
            String ← String + Char
    → String

```

```

else if (type(Data) = array) then
    Array ← []
    Urut traversal [1..len(Data)]
    IsiArray = Data[Urut]
    if (Urut ≥ Awal) and (Urut < Akhir) then
        Array.append(IsiArray)
    → Array

function IsDigit(string : string) -> bool:
    {SPESIFIKASI LOKAL}
    {Menerima input string untuk cek apakah isi dari string
    tersebut merupakan bilangan atau bukan}
    {I.S. : menerima input data berupa string}
    {F.S. : mengeluarkan boolean setelah pengecekan semua char
    suatu string}

    {KAMUS LOKAL}
    {string : string}
    {char : char}

    {ALGORITMA LOKAL}
    char in string:
        if char not in (chr(i) i traversal[48..57]) then
            → False
        → True

function ManualSplit (string : string, b : char) -> array:
    {SPESIFIKASI LOKAL}
    {Menerima input string untuk mengubahnya menjadi list dimana
    isinya berupa string yang terpisah-pisah oleh}
    {I.S. : string terdefinisi dan b merupakan separator}
    {F.S. : mengeluarkan list hasil memisahkan string dengan
    separator}

    KAMUS LOKAL
    string : string
    b : char
    list : array of string
    e, i : integer
    slicing : function

    ALGORITMA LOKAL
    list ← []
    e ← 0
    i traversal [0..len(string)-1]

```

```

        if ((ord(string[i]) = ord(b)) and (i ≠ len(string) - 1))
then
        list.append(slicing(string , e , i))
        e ← i + 1
    else if (i = len(string) - 1) then
        if (ord(string[i]) ≠ ord(b)) then
            list.append(slicing(string , e , i + 1))
        else
            list.append(slicing(string , e , i))
→ list

function RemoveNewLine(string : string) -> string :
{SPESIFIKASI LOKAL}
{Menerima input string untuk menghilangkan \n}
{I.S. : string terdefinisi}
{F.S. : menghilangkan char \n pada suatu string}

KAMUS LOKAL
string : string
ArrayTumbal : array of character
hasil : string
char : char
slicing : function

ALGORITMA LOKAL
ArrayTumbal ← []
hasil ← ""
char in string:
    ArrayTumbal.append(char)
char in slicing(ArrayTumbal , 0 , len(ArrayTumbal) - 1):
    hasil ← hasil + char
→ hasil

```

12. F11 – Laboratory

Tabel 6.12 Spesifikasi fungsi dalam program Laboratory(F11)

```

function laboratory(monster: array of array of string,
monster_inventory: array of array of string, oc_user:

```

```

string, id_user: string) → (array of array of string,
integer)

{fungsi untuk Mengupgrade level monster}

{I.S. user, monster_inventory, monster terdefinisi}

{F.S. level monster diupgrade jika memenuhi syarat tertentu
dan mengembalikan nilai array of array of string dari
monster_inventory dan mengembalikan nilai oc_user}

```

KAMUS LOKAL

```

monster_inventory : array of array of string
id_user           : integer
oc_user           : integer
list_monster      : array of string
idmon             : integer
No_monster        : integer
konfirm            : Char
nama_monster      : string
upgrade_cost       : array of integer
next_level         : integer
current_level      : integer

```

ALGORITMA

```

function laboratory(monster: array of array of string,
monster_inventory: array of array of string, oc_user:
string, id_user: string) → (array of array of string,
integer)

oc_user ← int(oc_user)

output("Selamat Datang Di Laboratorium Dokter Asep")

while True do

    output('OC yang anda miliki :', oc_user)
    output("===== Monster List =====")
    list_monster ← []
    j traversal [0, lenght(monster_inventory)
        if monster_inventory[j][0] = id_user then
            idmon ← monster_inventory[j][1]
            k traversal [0, lenght(monster)]
                if monster[k][0] = idmon then
                    output(lenght(list_monster) + 1,

```

```

        monster[k][1], "(Level:
        {monster_inventory[j][2]}))")
            list_monster.append([length(list_monster) +
1, monster[k][1], monster_inventory[j][2], j])

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

while True do
    No_monster ← output(">>> Pilih Nomor Monster yang akan
    di Upgrade (0 untuk keluar): ")
    depend on (No_monster, length(list_monster))
    No_monster = 0 :
        output("Terima kasih telah menggunakan Lab
        Dokter Asep!")
        return (monster_inventory, oc_user) # Keluar
        dari fungsi
        break
    1 ≤ No_monster ≤ length(list_monster):
        break
    No_monster > length(list_monster):
        output("Pilihan Monster tidak valid")

    current_level ← int(list_monster[No_monster - 1][2])
    next_level ← current_level + 1
    upgrade_cost ← [300, 500, 800, 1000, 0][next_level -
2]
    nama_monster ← list_monster[No_monster - 1][1]

    if current_level > 4 then
        output("Maaf, monster yang Anda pilih sudah
        memiliki level maksimum")
    else
        output(f"{nama_monster} akan di-upgrade ke level
        {next_level}")
        output(f"Harga untuk melakukan upgrade
        {nama_monster} adalah {upgrade_cost} OC")

```

```

while True do
    konfirm ← output(f"\n>>> Lanjutkan upgrade
(Y/N) : ")
    if konfirm.upper() = 'Y':
        if (oc_user) ≥ upgrade_cost then
            output("Selamat, {nama_monster}
berhasil di-upgrade ke level
{next_level}!")

        list_monster[No_monster - 1][2] ←
next_level
        oc_user ← oc_user - upgrade_cost
        output(f"O.W.C.A Coin anda sekarang:
{oc_user}")
        output(f"Level {nama_monster} sekarang:
{list_monster[No_monster - 1][2]}")

        # Update monster level IN
        monster_inventory
        monster_inventory[list_monster[No_monster-1][3]] [2] ← next_level
        break
    else
        output("O.W.C.A Coin tidak cukup")
        break

elseif konfirm.upper() = 'N' then
    output("Upgrade dibatalkan")
    break
else
    output("Masukkan pilihan yang valid")

```

REALISASI FUNGSI & PROSEDUR

```

output("Masukkan pilihan aksi ")
pilihan_aksi ← input()
if pilihan_aksi ← 'laboratory' then
    monster_inventory, oc ← laboratory(data_monster,
data_invent_monster, oc, user_id)
    user_loggedin[4] ← oc

```

13. F12 - Shop Management

Tabel 6.13 Spesifikasi fungsi dalam program Shop Management(F12)

<pre>function manage_shop(data_shop_monster : <u>array</u> of <u>array</u> of <u>string</u>, data_monster : <u>array</u> of <u>array</u> of <u>string</u>, data_item_shop : <u>array</u> of <u>array</u> of <u>string</u>) {Fungsi untuk admin mengatur shop baik menambah, mengubah, dan menghapus isinya. }</pre>
<p>KAMUS LOKAL</p> <p>valid, : Boolean</p> <p>Adakah : Boolean</p> <p>data_shop_monster : <u>array</u> of <u>array</u> of <u>string</u></p> <p>data_monster : <u>array</u> of <u>array</u> of <u>string</u></p> <p>data_item_shop : <u>array</u> of <u>array</u> of <u>string</u></p> <p>TambahTypePotion : <u>array</u> of <u>array</u> of <u>string</u></p> <p>monsterbelumlaku : <u>array</u> of <u>array</u> of <u>string</u></p> <p>potionbelumlaku : <u>array</u> of <u>array</u> of <u>string</u></p> <p>monsterBertahan : <u>array</u> of <u>array</u> of <u>string</u></p> <p>Aksi : <u>string</u></p> <p>LihatApa : <u>string</u></p> <p>IdPotion : <u>string</u></p> <p>TambahApa : <u>string</u></p> <p>MonsterLaku : <u>string</u></p> <p>inputstok : <u>string</u></p> <p>PilihMonster : <u>string</u></p>

```
HapusId : string
monsterdihapus : string
hargamonster : string
StokAwal : string
HargaPotion : string
UbahApa : string
PilihIdPotion : string
StokBaru : string
HargaBaru : string
PilihIdMonter : string
MonsterDiubah : string
HapusApa: string
YorN : character
Urutan : integer
urutan : integer
Isi : array of string
IsiMonster : array of string
Monster : array of string
Dalam : array of string
ShowMonster : array of string
monster : array of string
potion : array of string
```

```

function slicing(Data : any, Awal : integer , Akhir :
integer) -> any

{Menerima input list untuk diambil hanya sebagian dari suatu
list}

function IsDigit(string : string) -> bool:

{Menerima input string untuk cek apakah isi dari string
tersebut merupakan bilangan atau bukan}

function ManualSplit (string : string, b : character) ->
array of string:

{Menerima input string untuk mengubahnya menjadi list dimana
isinya berupa string yang terpisah-pisah oleh}

function RemoveNewLine(string : string) -> string :

{Menerima input string untuk menghilangkan \n}

```

ALGORITMA

```

input(Aksi)
while Aksi.lower() ≠ "keluar" do
    while Aksi.lower() ≠ "lihat" and Aksi.lower() ≠ "tambah" and
    Aksi.lower() ≠ "ubah" and Aksi.lower() ≠ "hapus" and
    Aksi.lower() ≠ "keluar" do
        output("\\nInput tidak valid, mohon input ulang\\n")
        input(Aksi)
        depend on : Aksi.lower()
        Aksi.lower() = "lihat" :
            input(LihatApa)
            while LihatApa.lower() ≠ "monster" and LihatApa.lower() ≠
            "potion" do
                output("\\nInput tidak valid, mohon input ulang\\n")
                input(LihatApa)

            if LihatApa.lower() = "monster" then
                if len(data_shop_monster) ≤ 1 then
                    output("\\nTidak ada monster yang terjual dalam shop
ini\\n")
                else

```

```

    output("ID\t| Type\t\t\t| ATK Power\t| DEF Power\t|
Stok\t| Harga")
    ID traversal [1..len(slicing(data_shop_monster, 1,
len(data_shop_monster)))]
        Isi = slicing(data_shop_monster, 1,
len(data_shop_monster))[ID]
        output(f"{Isi[0]}\t|", end=" ")
        Urutan traversal [1..len(slicing(data_monster, 1,
len(data_monster)))]
        IsiMonster = slicing(data_monster, 1,
len(data_monster))[Urutan]
        if Isi[0] = IsiMonster[0] then
            depend on (len(IsiMonster[1]))
            len(IsiMonster[1]) > 13 :
                output(f"{IsiMonster[1]}\t| {IsiMonster[2]}\t\t|{IsiMonster[3]}\t\t|", end=" ")
                len(IsiMonster[1]) > 5 :
                    output(f"{IsiMonster[1]}\t\t|{IsiMonster[2]}\t\t| {IsiMonster[3]}\t\t|", end=" ")
            else
                output(f"{IsiMonster[1]}\t\t\t|{IsiMonster[2]}\t\t| {IsiMonster[3]}\t\t|", end=" ")
                output(f"{Isi[1]}\t|{Isi[2]}")
        else
            if data_item_shop[1][1] = "0" and data_item_shop[2][1] =
"0" and data_item_shop[3][1] = "0" then
                output("\nSudah tidak ada potion yang terjual di shop,
isiin stoknya dong bang\n")
            else
                output("ID\t| Type\t\t\t| Stok\t| Harga")
                ID traversal [1..len(slicing(data_item_shop, 1, 4))]
                    Isi = slicing(data_item_shop, 1, 4)[ID]
                    if int(Isi[1]) > 0 then
                        output(f"{ID + 1}\t| {Isi[0]}\t\t| {Isi[1]}\t|{Isi[2]}")
                    else
                        continue
Aksi.lower() = "tambah" :
    input(TambahApa)
    depend on (TambahApa.lower())
    TambahApa.lower() ≠ "monster" and TambahApa.lower() ≠
"item":
        output("\nInput tidak valid, mohon input ulang\n")
    TambahApa.lower() = "monster":
        monsterbelumlaku ← []

```



```

        output("\nInput tidak valid, mohon input ulang\n")
    else
        input(inputstok)
        if not IsDigit(inputstok) or inputstok = "" then
            output("\nInput tidak valid, mohon input ulang\n")
        else
            input(hargamonster)
            if not IsDigit(hargamonster) or hargamonster = ""
    then
        output("\nInput tidak valid, mohon input ulang\n")
    else
        data_shop_monster.append([MonsterLaku[0],
inputstok, hargamonster])
        output(f"{MonsterLaku[1]} telah berhasil
ditambahkan ke dalam shop!")
    else {Tambah Potion}
        potionbelumlaku ← []
        ID traversal [1..data_item_shop]:
            potion ← data_item_shop[ID]
            if potion[1] = "0" then
                potionbelumlaku.append(potion)
            else
                continue
        if potionbelumlaku = [] then
            output("\nSemua potion sedang terjual sekarang\n")
        else
            output("ID\t| Type\t|")
            for ID traversal [1..data_item_shop]
                potion ← data_item_shop[ID]
                if potion[1] = "0" then
                    output(f"{ID}\t| {potion[0]}\t|")
                else
                    continue
            input(IdPotion)
            if not IsDigit(IdPotion) or IdPotion not in (chr(i) i
traversal [49, 51]) or data_item_shop[int(IdPotion)][1] ≠ "0"
    then
        output("\nInput tidak valid, mohon input ulang\n")
    else
        input(StokAwal)
        if not IsDigit(StokAwal) or StokAwal = "" then
            output("\nInput tidak valid, mohon input ulang\n")
        else
            input(HargaPotion)
            if not IsDigit(HargaPotion) or HargaPotion = "" then

```

```

        output("\\nInput tidak valid, mohon input ulang\\n")
    else
        output(f"\\n{data_item_shop[int(IdPotion)][0]}")
Potion telah berhasil ditambahkan ke dalam shop!\\n")
        data_item_shop[int(IdPotion)] ←
[data_item_shop[int(IdPotion)][0], str(StokAwal),
str(HargaPotion)]


    elif Aksi.lower() = "ubah" then
        UbahApa ← input("">>>> Mau ubah apa? (monster/potion):")
        if UbahApa.lower() ≠ "potion" and UbahApa.lower() ≠
"monster" then
            output("\\nInput tidak valid, mohon input ulang\\n")
        elif UbahApa.lower() = "potion" then
            if data_item_shop[1][1] = "0" and data_item_shop[2][1] =
"0" and data_item_shop[3][1] = "0" then
                output("\\nSudah tidak ada potion yang terjual di shop,
isiin stoknya dong bang\\n")
            else
                output("ID\\t| Type\\t\\t\\t| Stok\\t| Harga")
                for ID, Isi in enumerate(slicing(data_item_shop, 1, 4)):
                    if int(Isi[1]) > 0 then
                        output(f"{ID + 1}\\t| {Isi[0]}\\t\\t| {Isi[1]}\\t|
{Isi[2]}")
                PilihIdPotion ← input("">>>> masukkan id potion: ")
                depend on
                    not IsDigit(PilihIdPotion) or PilihIdPotion not in
(str(chr(i)) i traversal[49..52]
                    output("\\nInput tidak valid, mohon input ulang\\n")
                    data_item_shop[int(PilihIdPotion)][1] = "0" :
                        output("\\nInput tidak valid, mohon input ulang\\n")
                    else
                        input(StokBaru)
                        if not IsDigit(StokBaru) and StokBaru ≠ "" then
                            output("\\nInput tidak valid, mohon input ulang\\n")
                        else
                            input(HargaBaru)
                            if not IsDigit(StokBaru) and HargaBaru ≠ "" then
                                output("\\nInput tidak valid, mohon input ulang\\n")
                            else
                                depend on HargaBaru, StokBaru
                                HargaBaru = "" and StokBaru ≠ "":
                                    data_item_shop[int(PilihIdPotion)] ←
[data_item_shop[int(PilihIdPotion)][0], StokBaru,
data_item_shop[int(PilihIdPotion)][2]]
```

```

        output(f"{{data_item_shop[int(PilihIdPotion)][0]}}
Potion telah berhasil diubah dengan stok baru sejumlah
{StokBaru} dan dengan harga baru
{{data_item_shop[int(PilihIdPotion)][2]}!"})
        StokBaru = "" and HargaBaru != "":
            data_item_shop[int(PilihIdPotion)] ←
[{{data_item_shop[int(PilihIdPotion)][0]}, {{data_item_shop[int(PilihIdPotion)][1]}, HargaBaru}]
            output(f"{{data_item_shop[int(PilihIdPotion)][0]}}
Potion telah berhasil diubah dengan stok baru sejumlah
{{data_item_shop[int(PilihIdPotion)][1]}} dan dengan harga baru
{HargaBaru}!"")
        StokBaru != "" and HargaBaru != "":
            data_item_shop[int(PilihIdPotion)] ←
[{{data_item_shop[int(PilihIdPotion)][0]}, StokBaru, HargaBaru]
            output(f"{{data_item_shop[int(PilihIdPotion)][0]}}
Potion telah berhasil diubah dengan stok baru sejumlah
{StokBaru} dan dengan harga baru {HargaBaru}!")
        else
            data_item_shop[int(PilihIdPotion)] ←
[{{data_item_shop[int(PilihIdPotion)][0]}, {{data_item_shop[int(PilihIdPotion)][1]}, {{data_item_shop[int(PilihIdPotion)][2]}]
            output(f"Tidak ada perubahan pada
{{data_item_shop[int(PilihIdPotion)][0]}} potion")
-----
        else # Ubah Monster
        if len(data_shop_monster) ≤ 1 then
            output("\\nTidak ada monster yang terjual dalam shop
ini\\n")
        else
            output("ID\t| Type\t\t\t| ATK Power\t| DEF Power\t|
Stok\t| Harga")
            ID traversal [1..slicing(data_shop_monster, 1,
len(data_shop_monster))]
            Isi ← slicing(data_shop_monster, 1,
len(data_shop_monster))[ID]
            output(f"{{Isi[0]}}\\t|", end=" ")
            for Urutan, IsiMonster in enumerate(data_monster) do
                if Isi[0] = IsiMonster[0] then
                    if len(IsiMonster[1]) > 5 then
                        output(f"{{IsiMonster[1]}}\\t\\t|
{{IsiMonster[2]}}\\t\\t| {{IsiMonster[3]}}\\t\\t|", end=" ")
                    else

```

```

        output(f"\{IsiMonster[1]\}\t\t\t|"
{IsiMonster[2]}\t\t| {IsiMonster[3]}\t\t|", end=" ")
        output(f"\{Isi[1]\}\t|\{Isi[2]\} ")
        input(PilihIdMonster)
        if not IsDigit(PilihIdMonster) or PilihIdMonster not
in (str(data_shop_monster[i][0])) i traversal [2,
len(data_shop_monster)]) then
            output("\nInput tidak valid, mohon input ulang\n")
        else
            Monster in data_monster
            if PilihIdMonster = Monster[0] then
                MonsterDiubah ← Monster[1]
                break
            input(StokBaru)
            if not IsDigit(StokBaru) then
                output("\nInput tidak valid, mohon input ulang\n")
            else
                input(HargaBaru)
                if not IsDigit(HargaBaru) then
                    output("\nInput tidak valid, mohon input
ulang\n")
                else
                    if StokBaru ≠ "" and HargaBaru ≠ "" then
                        for urutan, Monster in
enumerate(data_shop_monster) do
                            if PilihIdMonster = Monster[0] then
                                output(f"\n{MonsterDiubah} telah berhasil
diubah dengan stok baru sejumlah {StokBaru} dan dengan harga
baru {HargaBaru}!\n")
                                data_shop_monster[urutan] ←
[PilihIdMonster, StokBaru, HargaBaru]
                                break
                            else if StokBaru = "" and HargaBaru ≠ "" then
                                urutan traversal [1..data_shop_monster]
                                Monster ← data_shop_monster[urutan]
                                if PilihIdMonster = Monster[0] then
                                    output(f"\n{MonsterDiubah} telah berhasil
diubah dengan stok baru sejumlah {Monster[1]} dan dengan harga
baru {HargaBaru}!\n")
                                    data_shop_monster[urutan] ←
[PilihIdMonster, Monster[1], HargaBaru]
                                    break
                                else if StokBaru ≠ "" and HargaBaru = "" then
                                    urutan traversal [1..len(data_shop_monster)]

```

```

        Monster ← data_shop_monster[urutan]
        if PilihIdMonster = Monster[0] then
            output(f"\n{MonsterDiubah} telah berhasil
diubah dengan stok baru sejumlah {StokBaru} dan dengan harga
baru {Monster[2]}!\n")
            data_shop_monster[urutan] ←
[PilihIdMonster, StokBaru, Monster[2]]
            break
        else
            output(f"\n{MonsterDiubah} tidak mengalami
perubahan stok ataupun harga!\n")
        else:
            {Hapus}
            input(HapusApa)
            if HapusApa.lower() ≠ "monster" and HapusApa.lower() ≠
"potion" then
                output('\nInput tidak valid, mohon input ulang\n')
            elif HapusApa.lower() = "monster" then
                # Hapus Monster
                if len(data_shop_monster) ≤ 1 then
                    output('\nSudah tidak ada monster yang bisa dihapus
dari shop\n')
                else
                    ID traversal [1..slicing(data_shop_monster, 1,
len(data_shop_monster))]
                    Isi ← data_shop_monster[ID]
                    output(f'{Isi[0]}\t')
                    Urutan traversal [1..len(data_monster)]
                    Isi ← data_shop_monster[ID]
                    if Isi[0] = IsiMonster[0] then
                        if len(IsiMonster[1]) > 5 then
                            output(f'{IsiMonster[1]}\t\t|
{IsiMonster[2]}\t\t| {IsiMonster[3]}\t\t|')
                        else
                            output(f'{IsiMonster[1]}\t\t\t|
{IsiMonster[2]}\t\t| {IsiMonster[3]}\t\t|')
                            output(f'{Isi[1]}\t{Isi[2]}')
                            input(HapusId)
                            if not IsDigit(HapusId) or HapusId not in
(str(data_shop_monster[i][0]) i traversal[2,
len(data_shop_monster))) then
                                output('\nInput tidak valid, mohon input ulang\n')
                            else
                                Urutan traversal [1..len(data_monster)]

```

```

IsiMonster ← data_monster[Urutan]
if HapusId = IsiMonster[0] then
    monsterdihapus ← IsiMonster[1]
    break
input(YorN)
while YorN.lower() ≠ 'y' and YorN.lower() ≠ 'n' do
    output('\nInput tidak valid, mohon input ulang\n')
    input(YorN)
if YorN.lower() = 'n' then
    output('\nProses penghapusan monster dari shop
dibatalkan.\n')
else
    {YorN.lower() = 'y'}
    MonsterBertahan ← []
    ID traversal [1..len(data_shop_monster))
    Isi ← data_shop_monster[ID]
    if Isi[0] = HapusId then
        continue
    MonsterBertahan.append(Isi)
    data_shop_monster ← MonsterBertahan
    output(f'\n{monsterdihapus} telah berhasil dihapus
dari shop!\n')
else
    # Hapus Potion
    if data_item_shop[1][1] = '0' and data_item_shop[2][1] =
    '0' and data_item_shop[3][1] = '0' then
        output('\nSudah tidak ada potion yang terjual di shop,
datang lagi di lain waktu!\n')
    else
        output(f'''ID\t| Type\t\t| Stok\t| Harga''')
        ID traversal [1..len(slicing(data_item_shop, 1, 4))]
        Isi ← slicing(data_item_shop, 1, 4)[ID]
        if int(Isi[1]) > 0 then
            output(f'''{ID + 1}\t| {Isi[0]}\t\t| {Isi[1]}\t|
{Isi[2]}''')
            HapusId ← input('">>>> Masukkan Id potion: ')
            if not IsDigit(HapusId) or HapusId not in (str(chr(i))
i traversal[49..52]) then
                output('\nInput tidak valid, mohon input ulang\n')
            elif data_item_shop[int(HapusId)][1] = '0' then
                output('\nInput tidak valid, mohon input ulang\n')
            else
                input(YorN)
                while YorN.lower() ≠ 'y' and YorN.lower() ≠ 'n' do
                    output('\nInput tidak valid, mohon input ulang\n')

```

```

        input(YorN)
        if YorN.lower() = 'n' then
            output('\nProses penghapusan monster dari shop
dibatalkan.\n')
        else
            {YorN.lower() = 'y'}
            data_item_shop[int(HapusId)][2] ← '0'
            output(f'\n{data_item_shop[int(HapusId)][0]} telah
berhasil dihapus dari shop!\n')
        input(Aksi)
        output('\nMr. Yanto bilang terima kasih sudah bantu mengatur
tokonya menjadi lebih baik :)\n')
    → data_shop_monster, data_monster, data_item_shop

```

REALISASI FUNGSI & PROSEDUR

```

function slicing(Data : any, Awal : integer , Akhir : integer)
-> any:
    {SPESIFIKASI LOKAL}
    {Menerima input list untuk diambil hanya sebagian dari suatu
list}
    {I.S. : menerima input data yang dapat berupa array ataupun
string}
    {F.S. : mengeluarkan sebagian dari input awal array atau
string}

    {KAMUS LOKAL}
    {Data : any}
    {Awal, Akhir : integer}
    {String, IsiArray : string}
    {Urut : integer}
    {Array : array of string}

    {ALGORITMA LOKAL}
    if (type(Data) = string) then
        String ← ""
        Urut traversal [1..len(Data)]
        char = Data[Urut]
        if Urut ≥ Awal and Urut < Akhir then
            String ← String + Char
        → String
    else if (type(Data) = array) then
        Array ← []
        Urut traversal [1..len(Data)]
        IsiArray = Data[Urut]

```

```

        if (Urut ≥ Awal) and (Urut < Akhir) then
            Array.append(IsiArray)
        → Array

function IsDigit(string : string) -> bool:
    {SPESIFIKASI LOKAL}
    {Menerima input string untuk cek apakah isi dari string
    tersebut merupakan bilangan atau bukan}
    {I.S. : menerima input data berupa string}
    {F.S. : mengeluarkan boolean setelah pengecekan semua char
    suatu string}

    {KAMUS LOKAL}
    {string : string}
    {char : char}

    {ALGORITMA LOKAL}
    char in string :
        if char not in (chr(i) i traversal[48..57]) then
            → False
        → True

function ManualSplit (string : string, b : char) -> array:
    {SPESIFIKASI LOKAL}
    {Menerima input string untuk mengubahnya menjadi list dimana
    isinya berupa string yang terpisah-pisah oleh}
    {I.S. : string terdefinisi dan b merupakan separator}
    {F.S. : mengeluarkan list hasil memisahkan string dengan
    separator}

    KAMUS LOKAL
    string : string
    b : char
    list : array of string
    e, i : integer
    slicing : function

    ALGORITMA LOKAL
    list ← []
    e ← 0
    i traversal [0..len(string)-1]
        if ((ord(string[i]) = ord(b)) and (i ≠ len(string) - 1))
    then
        list.append(slicing(string , e , i))
        e ← i + 1

```

```

        else if (i = len(string) - 1) then
            if (ord(string[i]) ≠ ord(b)) then
                list.append(slicing(string , e , i + 1))
            else
                list.append(slicing(string , e , i))
        → list

function RemoveNewLine(string : string) -> string :
    {SPESIFIKASI LOKAL}
    {Menerima input string untuk menghilangkan \n}
    {I.S. : string terdefinisi}
    {F.S. : menghilangkan char \n pada suatu string}

    KAMUS LOKAL
    string : string
    ArrayTumbal : array of character
    hasil : string
    char : char
    slicing : function

    ALGORITMA LOKAL
    ArrayTumbal ← []
    hasil ← ""
    char in string :
        ArrayTumbal.append(char)
    char slicing(ArrayTumbal , 0 , len(ArrayTumbal) - 1) :
        hasil ← hasil + char
    → hasil

```

14. F13 - Monster Management

Tabel 6.14 Spesifikasi fungsi dalam program Monster Management (F13)

```

function monster_mng(listmon : array of array of string)
    {Mengatur isi data list monster baik menambahkan ataupun
    melihat isi data monster.

    I.S. variabel listmon terdefinisi
    F.S. mengembalikan data listmon dengan data terbaru}

```

KAMUS LOKAL

```
listmon      : array of (array of string)  
  
function samain_panjang (listmonn: array of (array of string)) → array of (array of string)  
{ Mengembalikan array of array of string yang telah  
dimodifikasi untuk dicetak}  
  
procedure print_mons (listmons: array of (array of string))  
{ I.S. : Diberikan listmonn yang telah terdefinisi}  
{ F.S. : Menghasilkan cetakan dari array of array of  
strings yang telah dimodifikasi}  
  
function delarraymanual (listmonns: array of (array of string), array_index: integer) → array of (array of string)  
{ Mengembalikan array of array of string yang telah  
mengalami pengurangan kolom}  
  
function ceknama (instr: string) → boolean  
{ Mengecek string yang dimasukkan bernilai angka atau  
huruf, dan mengembalikan nilai True atau False berdasarkan  
hasil pengecekan}  
  
function cekint (inint: string) → boolean  
{ Mengecek string yang dimasukkan bernilai angka atau  
huruf, dan mengembalikan nilai True atau False berdasarkan  
hasil pengecekan}  
  
function cekdeff (indef: string) → boolean  
{ Mengecek string yang dimasukkan bernilai angka dengan  
batasan yang ditentukan, dan mengembalikan nilai True atau  
False berdasarkan hasil pengecekan}  
  
function ceklen (instr: string) → boolean  
{ Mengecek string yang dimasukkan kosong atau tidak, dan  
mengembalikan nilai  
True atau False berdasarkan hasil pengecekan}  
  
function cekar (instr: string, cekarray: array of (array of string)) → boolean  
{ Mengecek string yang dimasukkan bila terdapat pada array  
of array of string, dan mengembalikan nilai True atau  
False berdasarkan hasil pengecekan}  
  
function IsDigit (stri: string) → boolean
```

```

{ Mengecek string dengan panjang 1 yang dimasukkan
bernilai angka, dan mengembalikan nilai True atau False
berdasarkan hasil pengecekan}

function cekar (instr: string, cekarray: array of (array
of string)) → boolean
{ Mengecek string yang dimasukkan bernilai 1 atau 2, dan
mengembalikan nilai True atau False berdasarkan hasil
pengecekan}

```

ALGORITMA

```

function monster_mng(listmon: list)
    output("Selamat datang di database para monster")
    output("1. Tampilkan semua Monster")
    output("2. Tambah Monster baru")
    arr ← ["ID", "Type", "ATK Power", "DEF Power", "HP"]
    listmonbaru ← delarraymanual(listmon, 0)
    arr1 ← [arr] + listmonbaru
    output("Pilih aksi: ")
    input(a)
    while (str(a) ≠ "KELUAR") do
        repeat
            output("Pilihan aksi tidak tersedia, pilih kembali!")
            a = input("Pilih aksi: ")
            IsDigit(a)
            cekinputawal(a)
            until (IsDigit(a)) or (cekinputawal(int(a)))
            depend on(int(a))
            int(a) = 1:
            print_mons(arr1)
            lanjut ← False
            output("Apakah mau melakukan hal lain? (Y/N): ")
            input(confirm)
            while not(lanjut) do
                depend on(str(confirm))
                (str(confirm)='y') or (str(confirm)='Y'):
                    output("Pilih aksi: ")
                    input(a)
                    lanjut ← True
                (str(confirm)='n') or (str(confirm)='N'):
                    a ← "KELUAR"
                    lanjut ← True
            else
                output("Pilihan tidak tersedia, pilih kembali!")
                output("Apakah mau melakukan hal lain? (Y/N): ")
                input(confirm)

```

```

int(a) = 2:
    output("Memulai pembuatan monster baru..")
    output("Masukkan Type / Nama: ")
    input(nama)
    repeat
        depend on(cekar(nama, arr1), ceknama(nama),
        ceklen(nama))
        not(cekar(nama, arr1)):
            output("Nama sudah terdaftar, coba lagi!")
        not(ceknama(nama)) :
            output("Masukkan input bertipe string, coba
            lagi!")
        not(ceklen(nama)) and cekar(nama, arr1) and
        ceknama(nama):
            output("Masukkan input harus bernilai, coba
            lagi!")
        output("Masukkan Type / Nama: ")
        input(nama)
    until (not(cekar(nama, arr1))) or (not(ceknama(nama)))
    or (not(ceklen(nama)))

    output("Masukkan ATK Power : ")
    input(atk)
    repeat
        depend on(cekint(atk), ceklen(atk))
        not(cekint(atk)):
            output("Masukkan input bertipe integer dan
            bernilai positif, coba lagi!")
        not(ceklen(atk)):
            output("Masukkan input harus bernilai, coba
            lagi!")
        output("Masukkan ATK Power : ")
        input(atk)
    until not(cekint(atk)) or not(ceklen(atk))

    output("Masukkan DEF Power : ")
    input(deff)
    repeat
        depend on(ceklen(deff), cekint(deff), cekdef(deff))
        not(ceklen(deff)):
            output("Masukkan input harus bernilai, coba
            lagi!")
        not(cekdef(deff)) and cekint(deff):
            output("DEF Power harus bernilai 0-50, coba
            lagi!")

```

```

not(cekint(deff)) :
    output("Masukkan bertipe integer dan bernilai
    positif, coba lagi!")
    output("Masukkan DEF Power : ")
    input(deff)
until not(cekdef(deff)) or not(ceklen(deff)) or
not(cekint(deff))

output("Masukkan HP : ")
input(hp)
repeat
    depend on(ceklen(hp), cekint(hp))
    not(cekint(hp)) :
        output("Masukkan input bertipe integer dan
        bernilai positif, coba lagi!")
    not(ceklen(hp)) :
        output("Masukkan input harus bernilai, coba
        lagi!")
        output("Masukkan HP : ")
        input(hp)
until not(cekint(hp)) or not(ceklen(hp))

output("Monster baru berhasil dibuat!")
output("Type : ", nama)
output("ATK Power : ", atk)
output("DEF Power : ", deff)
output("HP : ", hp)
output("Tambahkan Monster ke database (Y/N) : ")
input(k)
tambah ← False
repeat
    depend on(str(k))
    str(k) = 'y' or str(k) = 'Y':
        h ← arr1[len(arr1)-1][0]
        MonsterBaru ← [str(int(h)+1), nama, atk, deff,
        hp]
        arr1 ← arr1 + [MonsterBaru]
        tambah ← True
        output("Monster baru telah ditambahkan!")
    str(k) = 'n' or str(k) = 'N':
        output("Monster gagal ditambahkan!")
        tambah ← True
else
    output("Pilihan tidak tersedia, pilih kembali!")
    output("Tambahkan Monster ke database (Y/N) : ")

```

```

input(k)
until not(tambah)

lanjut ← False
output("Apakah mau melakukan hal lain? (Y/N) : ")
input(confirm)
repeat
    depend on(str(confirm))
        str(confirm) = 'y' or str(confirm)= 'Y':
            output("Pilih aksi: ")
            input(a)
            lanjut ← True
        str(confirm) = 'n' or str(confirm)= 'N':
            a ← "KELUAR"
            lanjut ← True
    else
        output("Pilihan tidak tersedia, pilih kembali!")
        output("Apakah mau melakukan hal lain? (Y/N) : ")
        input(confirm)
    until not(lanjut)
output("Para Monster mengucapkan Selamat Tinggal")
→ arr1

```

REALISASI FUNGSI & PROSEDUR

```

function samain_panjang(listmonn: list)
    hasil ← []
    i traversal[1..(len(listmonn))]
    hasil ← hasil + ['']
    i traversal[1..(len(listmonn[0])-1)]
        b ← 0
        j traversal[1..(len(listmonn))]
            if (b < len(listmonn[j][i]))then
                b ← len(listmonn[j][i])
            j traversal[1..(len(listmonn))]
                hasil[j] ← hasil[j] + [listmonn[j][i]]
                if (b > len(listmonn[j][i]))then
                    while (b > len(listmonn[j][i]))then
                        hasil[j] ← hasil[j] + [" "]
                        listmonn[j][i] ← listmonn[j][i] + [" "]
                hasil[j] ← hasil[j] + [" | "]
    i traversal[1..(len(listmonn))]
        hasil[i] ← hasil[i] + [listmonn[i][len(listmonn[0])-1]]
→ hasil

procedure print_mons(listmons: list)
    listmons ← samain_panjang(listmons)

```

```

i traversal[1..(len(listmons))]
    j traversal[1..(len(listmons[i])-1)]
        output(listmons[i][j])
        output(listmons[i][len(listmons[i])-1])

function delarraymanual(listmonns, array_index)
    listmonbaru ← []
    i traversal[1..(len(listmonns))]
        if (i ≠ array_index) then
            listmonbaru ← listmonbaru + [listmonns[i]]
    → listmonbaru

function ceknama(instr)
    i traversal[1..len(instr)]
        ceknem[i] ← ord(instr[i])
    i ← 0
    f ← True
    while (i<(len(ceknem))) do
        if (65≤ceknem[i]≤90) or (97≤ceknem[i]≤122) then
            i ← i+1
        else
            f ← False
            i ← 1
    → f

function cekint(inint)
    i traversal[1..len(inint)]
        cekinte[i] ← ord(inint[i])
    i ← 0
    f ← True
    while (i<(len(cekinte))) do
        if (48≤cekinte[i]≤57) then
            i ← i+1
        else
            f ← False
            i ← 1
    → f

function cekdef(indef)
    i traversal[1..len(indef)]
        cekdeff[i] ← ord(indef[i])
    depend on(cekdeff)
        (len(cekdeff)=2) :
            depend on(cekdeff)
            (49≤cekdeff[0]≤52) :

```

```

        → True
(cekdeff[0]= 53) and (cekdeff[1]=48):
        → True
else
        → False
(len(cekdeff)=1):
        → True
else
        → False

function ceklen(instr)
if (len(instr)=0)then
        → False

function cekar(instr, cekarray)
f ← True
i ← 0
while (i<(len(cekarray))) do
    if (str(instr) ≠ cekarray[i][1]) then
        i ← i + 1
    else
        f ← False
        i ← i + 1
    → f

function IsDigit(stri)
f ← true
i traversal[1..len(stri)]
j traversal[48..57]
if stri[i] ≠ chr[j]then
    f ← False
    break
→ f

function cekinputawal(inte)
if (inte = 1) or (inte = 2)then
        → True
else
        → False

```

15. F14 - Load

Tabel 6.15 Spesifikasi fungsi dalam program Load (F14)

<pre>procedure load(<u>output</u> user : <u>matrix</u> of <u>data</u> , <u>output</u> monster : <u>matrix</u> of <u>data</u> , <u>output</u> inv_item : <u>matrix</u> of <u>data</u> , <u>output</u> inv_monster : <u>matrix</u> of <u>data</u> , <u>output</u> shop_item : <u>matrix</u> of <u>data</u> , <u>output</u> shop_monster : <u>matrix</u> of <u>data</u>)</pre> <p>{ I.S. Menerima data parser dari input yang dijalankan. } { F.S. Data pada csv_directory telah tersimpan ke matrix of data. }</p>
<p>KAMUS LOKAL</p> <pre>parser , args : <u>ArgumentParser</u> folder , path_folder , csv_directory , row path_user , path_monster , path_inv_item , path_inv_monster , path_shop_item , path_shop_monster : <u>string</u> path_data : <u>array</u> of <u>string</u> content : <u>matrix</u> of <u>string</u> data : <u>data</u> user , monster , inv_item , inv_monster , shop_item , shop_monster : <u>matrix</u> of <u>data</u> i , j : <u>integer</u> {index} function csplit(string : <u>string</u> , delimiter : <u>character</u>) → <u>array</u> of <u>string</u> { Membuat fungsi personal yang serupa dengan fungsi split() pada python. } function cstrip(string : <u>string</u> , chars : <u>character</u>) → <u>string</u> { Membuat fungsi personal yang serupa dengan fungsi strip() pada python. }</pre>
<p>ALGORITMA</p> <pre>parser ← argparse.ArgumentParser(description = "Argument Parser") parser.add_argument("folder" , type = str , nargs = '?' , default = None , help = "Directory folder data CSV yang akan digunakan.") args ← parser.parse_args() if (args.folder = None) <u>then</u> <u>output</u>("Tidak ada nama folder yang diberikan!") <u>output</u>("Usage : python main.py <nama_folder>") → (1 , 1 , 1 , 1 , 1 , 1) else folder ← args.folder path_folder ← 'data/' + str(folder) if (<u>not</u>(os.path.exists(path_folder))) <u>then</u> <u>output</u>("Folder {folder} tidak ditemukan!") → (1 , 1 , 1 , 1 , 1 , 1)</pre>

```

else
    path_user ← path_folder + '/user.csv'
    path_monster ← path_folder + '/monster.csv'
    path_inv_item ← path_folder + '/item_inventory.csv'
    path_inv_monster ← path_folder +
    '/monster_inventory.csv'
    path_shop_item ← path_folder + '/item_shop.csv'
    path_shop_monster ← path_folder + '/monster_shop.csv'
    path_data ← [path_user , path_monster , path_inv_item ,
path_inv_monster , path_shop_item , path_shop_monster]
    content ← []
    i traversal [0..5]
        assign(data , path_data[i])
        open(data)
        j traversal [0..(len(data)-1)]
            row ← csplit(cstrip(data[j])) , ',')
            content[i] ← content[i] + [row]
        close (Data)
    user = content[0]
    monster = content[1]
    inv_item = content[2]
    inv_monster = content[3]
    shop_item = content[4]
    shop_monster = content[5]
    output("Selamat datang di program OWCA!")
    → user , monster , inv_item , inv_monster , shop_item ,
shop_monster

```

REALISASI FUNGSI & PROSEDUR

```

function csplit(string : string , delimiter : character) → array
of string
{ Membuat fungsi personal yang serupa dengan fungsi split() pada
python. }
KAMUS LOKAL
    result : array of string
    string , column : string
    delimiter : character
    i : integer {index}
ALGORITMA LOKAL
    result ← []
    column ← ""
    i traversal [0..(len(string)-1)]
        if (string[i] = delimiter) then
            result ← result + [column]
        else
            column ← column + string[i]
    result ← result + [column]

```

```

→ result
function cstrip(string : string , chars : character) → string
{ Membuat fungsi personal yang serupa dengan fungsi strip() pada
python. }
KAMUS LOKAL
    separator : character
    string , output : string
    chars : array of character
    start , end : integer {index}
ALGORITMA LOKAL
    if (chars = None) then
        chars = '\t\n\r\x0b\x0c'
    start ← 0
    end ← len(string) - 1
    while ((start ≤ end) and (string[start] in chars)) do
        start ← start + 1
    while ((end ≥ start) and (string[end] in chars)) do
        end ← end - 1
    output ← string[start : (end + 1)]
→ output

```

16. F15 - Save

Tabel 6.15 Spesifikasi fungsi dalam program Save (F15)

```

procedure save(input user : matrix of data , input monster : matrix of data ,
input inv_item : matrix of data , input
inv_monster : matrix of data , input shop_item : matrix of
data , input shop_monster : matrix of data)

{ I.S. Data user, monster, inv_item, inv_monster, shop_item,
dan shop_monster telah terdefinisi. }
{ F.S. Data user, monster, inv_item, inv_monster, shop_item,
dan shop_monster telah tersimpan di suatu folder. }

KAMUS LOKAL
    user , monster , inv_item , inv_monster , shop_item ,
        shop_monster : matrix of data
    f_user , f_monster , f_inv_item , f_inv_monster , f_shop_item
        , f_shop_monster : SEQFILE of (*) <list : array of stringarray of string
    forbidden_character : array of character
    nama , path , parent , directory : string
    check , check_directory : boolean
    roll : integer
    i : integer {index}
    function cmap(function : function , map_data : data) → array
of data

```

```

{ Membuat fungsi personal yang serupa dengan fungsi map()
pada python. }
function cjoin(separator : character , join_data : data) →
string
{ Membuat fungsi personal yang serupa dengan fungsi join()
pada python. }

```

ALGORITMA

```

import os
parent ← '.data/'
check_directory ← False
while (not(check_directory)) do
    check_directory ← True
    check ← False
    input(directory)
    forbidden_character ← '/\:.*?><|'
    i traversal [0..(len(directory)-1)]
    j traversal [0..(len(forbidden_character)-1)]
    if (directory[i] = forbidden_cahracter[j]) then
        check_directory ← False
        check ← True
        output("Nama folder dilarang")
        break
    if (check) then
        break
path ← os.path.join(parent , directory)
make ← False
if (not(os.path.exists(path))) then
    os.mkdir(path)
    make ← True
os.chdir(parent + directory)
assign(f_user , "user.csv")
h_user ← ["id" , "username" , "password" , "role" , "oc"]
rewrite(f_user)
write(f_user , cjoin(';' , cmap(str , h_user)))
i traversal [1..(len(user) - 1)]
    write(f_user , cjoin(';' , cmap(str , user[i])))
close(f_user)
assign(f_monster , "monster.csv")
h_monster ← ['id' , 'type' , 'atk_power' , 'def_power' ,
'hp']
rewrite(f_monster)
write(f_monster , cjoin(';' , cmap(str , h_monster)))
i traversal [1..(len(monster) - 1)]
    write(f_user , cjoin(';' , cmap(str , monster[i])))
close(f_monster)

```

```

assign(f_inv_item , "item_inventory.csv")
h_inv_item ← ["user_id" , "type" , "quantity"]
rewrite(f_inv_item)
write(f_inv_item , cjoin(';' , cmap(str , h_inv_item)))
i traversal [1..(len(inv_item) - 1)]
    write(f_inv_item , cjoin(';' , cmap(str , inv_item[i])))
close(f_inv_item)
assign(f_inv_monster , "monster_inventory.csv")
h_inv_monster ← ["user_id" , "monster_id" , "level"]
rewrite(f_inv_monster)
write(f_inv_monster , cjoin(';' , cmap(str , h_inv_monster)))
i traversal [1..(len(inv_monster) - 1)]
    write(f_inv_monster , cjoin(';' , cmap(str ,
inv_monster[i])))
close(f_inv_monster)
assign(f_shop_item , "item_shop.csv")
h_shop_item ← ["type" , "stock" , "price"]
rewrite(f_shop_item)
write(f_shop_item , cjoin(';' , cmap(str , h_shop_item)))
i traversal [1..(len(shop_item) - 1)]
    write(f_shop_item , cjoin(';' , cmap(str , shop_item[i])))
close(f_shop_item)
assign(f_shop_monster , "monster_shop.csv")
h_shop_monster ← ["monster_id" , "stock" , "price"]
rewrite(f_shop_monster)
write(f_shop_monster , cjoin(';' , cmap(str ,
h_shop_monster)))
i traversal [1..(len(shop_monster) - 1)]
    write(f_shop_monster , cjoin(';' , cmap(str ,
shop_monster[i])))
close(f_shop_monster)
output("Saving...")
if (make) then
    output("Membuat folder {parent + directory}...")
output("Berhasil menyimpan data di folder {parent +
directory}!")
os.chdir('..')

```

REALISASI FUNGSI & PROSEDUR

```

function cmap(function : function , map_data : data) → array of
data
{ Membuat fungsi personal yang serupa dengan fungsi map() pada
python. }
KAMUS LOKAL
    function : function
    map_data : data

```

```

map_result : array of map_data
i : integer {index}

ALGORITMA LOKAL
map_result ← []
i traversal [0..(len(map_data)-1)]
map_result ← map_result + function(map_data[i])
→ map_result
function cjoin(separator : character , join_data : data) →
string
{ Membuat fungsi personal yang serupa dengan fungsi join() pada
python. }
KAMUS LOKAL
separator : character
join_data , item : data
join_result : string
i : integer {index}

ALGORITMA LOKAL
join_result ← ""
i traversal [0..(len(join_data)-1)]
join_result ← join_result + str(join_data[i])
if (i < (len(join_data) - 1) then
    join_result ← join_result + separator
→ join_result

```

17. F16 - Exit

Tabel 6.17 Spesifikasi fungsi dalam program Battle (F16)

```

function exit(check_login : boolean , user : matrix of data
, monster : matrix of data , inv_item : matrix of data ,
inv_monster : matrix of data , shop_item : matrix of data ,
shop_monster : matrix of data) → boolean

{ Melakukan proses keluar dari program yang berjalan. }

KAMUS LOKAL
user , monster , inv_item , inv_monster , shop_item ,
shop_monster : matrix of data
check_logres , end : boolean
ending : character

ALGORITMA
if (check_logres) then
    input(ending)
    while (not((ending = 'y') or (ending = 'n'))) do
        input(ending)
    if (ending = 'y') then
        save(user, monster , inv_item , inv_monster , shop_item
        , shop_monster)
    else

```

```
    output("Tidak ada program yang sedang berjalan")
end ← True
→ end
```

Lampiran Hasil Pengujian Program

1. F00 - Random Number Generator

hasil generate dari range 0 hingga 9 (10 tidak termasuk) :

```
61     print(generate([0, 10]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\ACER\OneDrive\Documents\GitHub\if1210-2024-tubes-k05-c\src\rng.py
5
- PS C:\Users\ACER\OneDrive\Documents\GitHub\if1210-2024-tubes-k05-c\src\rng.py
3
- PS C:\Users\ACER\OneDrive\Documents\GitHub\if1210-2024-tubes-k05-c\src\rng.py
6
- PS C:\Users\ACER\OneDrive\Documents\GitHub\if1210-2024-tubes-k05-c\src\rng.py
0

Gambar 7.1.1 Hasil generate RNG

2. F01 – Register

Saat input normal dan benar :

```
>>> REGISTER
```

Masukkan Username : TubesKeren
Masukkan Password : TubesMantap123
Silakan pilih salah satu monster sebagai awalmu!
1. Pikachu
2. Bulbu
3. Zeze
Masukkan Pilihanmu (masukkan angka saja) : 1
Selamat datang Agent TubesKeren. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachu!

Gambar 7.2.1 Hasil register dengan input yang benar

Saat username telah terpakai atau ada karakter terlarang :

```
Masukkan Username : TubesKeren&
Masukkan Password : CintaTubes
Username hanya boleh berisi alfabet, angka, underscore, dan strip!
Masukkan Username : raihan
Masukkan Password : geprek
Username raihan sudah terpakai, silakan gunakan username lain!
```

Gambar 7.2.2 Hasil jika username sudah dipakai atau ada karakter terlarang

Saat ingin register dalam keadaan ada yang sedang login :

```
>>> REGISTER
[REDACTED]
Register gagal!
Anda telah login dengan username TimKeren, silakan lakukan "LOGOUT" sebelum melakukan register.
```

Gambar 7.2.3 Hasil jika register namun sedang login

3. F02 – Login

jika login username dan password benar:

```
>>> login
[REDACTED]
Masukkan Username : defir
Masukkan Password : 12345678
Selamat datang, Agent defir!
Masukkan command "help" untuk daftar perintah yang dapat kamu panggil.
```

Gambar 7.3.1 Hasil jika username dan password benar

jika username tidak terdaftar:

```
[REDACTED]
Masukkan Username : 1234
Masukkan Password : 123
Username tidak terdaftar!
Silakan register terlebih dahulu!
```

Gambar 7.3.2 Hasil jika username tidak terdaftar

jika username terdaftar dan passwordnya salah:

```
>>> login
Masukkan Username : defir
Masukkan Password : 123
Password salah!
```

Gambar 7.3.3 Hasil jika password salah

jika username sudah login:

```
>>> login
Login gagal!
Anda telah login dengan username defir, silakan lakukan "LOGOUT" sebelum melakukan login kembali.
```

Gambar 7.3.4 Hasil jika username sudah login

4. F03 – Logout

Jika logout memiliki parameter bernilai False:

```
Logout gagal!
Anda belum login, silahkan login terlebih dahulu.
```

Gambar 7.4.1 Hasil jika parameter check_login false

Jika logout memiliki parameter bernilai True:

```
Logout berhasil!
```

Gambar 7.4.2 Hasil jika parameter check_login True

5. F04 - Menu & Help

Help saat belum login :

Gambar 7.5.1 Tampilan help saat belum login

Help saat login admin :

Gambar 7.5.2 Tampilan help saat login sebagai admin

Help saat login non admin :

Gambar 7.5.3 Tampilan help saat login sebagai admin

Menu saat belum login :

Selamat datang di program OWCA!

The image shows a large, bold, blocky font word "WELCOME" centered on a black background. The letters are white with black outlines, giving them a 3D, stepped appearance. Above the word, there are several "<!--" and "-->" symbols, likely from a code editor. To the right of the word, there is a vertical series of right-pointing arrows (-->).

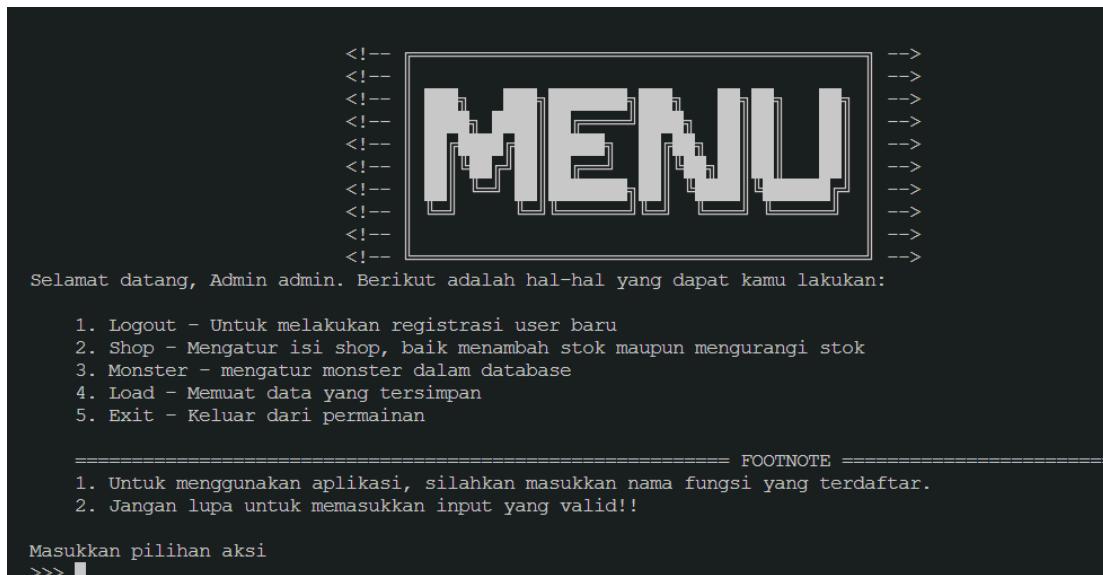
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

1. Register - melakukan registrasi user baru
2. Login - melakukan login ke akun ter-register
3. Exit - Keluar dari permainan

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

Gambar 7.5.4 Tampilan menu saat belum login

Menu saat login admin :



Gambar 7.5.5 Tampilan menu saat login sebagai admin

Menu saat login non admin :



Gambar 7.5.5 Tampilan menu saat login sebagai agent

6. F05 - Monster

Gambar 7.6.1 Tampilan awal fungsi Monster

Validasi keluar fitur:

'keluar' untuk kembali ke menu : NO
'keluar' untuk kembali ke menu : █

Gambar 7.6.2 Hasil saat validasi keluar fitur

'keluar' untuk kembali ke menu : Keluar
(base) aleta@iMac TUBES DASPRO %

Gambar 7.6.3 Hasil saat sudah keluar

7. F06 - Potion

Potion	
Strength Potion	Meningkatkan ATK Power sebanyak 5% dari ATK Power
Resilience Potion	Meningkatkan Def Power sebanyak 5% dari DEF Power
Healing Potion	Mengisi darah sebanyak 25% dari Base HP, tidak akan mengisi lebih dari maks HP
Tips! Gunakan potion yang tepat di saat yang tepat untuk melawan monster yang lebih kuat! Potion bisa dibeli di shop dengan OC.	

Gambar 7.7.1 Tampilan menu potion

Validasi keluar fitur

'keluar' untuk kembali ke menu : NO
'keluar' untuk kembali ke menu : █

Gambar 7.7.2 Tampilan validasi keluar menu potion

```
'keluar' untuk kembali ke menu : keluar
(base) aleta@iMac TUBES DASPRO %
```

Gambar 7.7.3 Tampilan menu potion

8. F07 - Inventory

```
===== INVENTORY LIST (User ID: 1) =====
Jumlah O.W.C.A. Coin-mu sekarang 10.
1. Monster      (Name: Pikachu, Lvl: 2, HP: 600)
2. Monster      (Name: Zeze, Lvl: 5, HP: 100)
3. Potion       (Type: strength, Qty: 5)
4. Potion       (Type: healing, Qty: 2)
5. Potion       (Type: resilience, Qty: 7)
6. Monster Ball (Qty: 2)
Ketikkan id untuk menampilkan detail item:
```

Gambar 7.8.1 Tampilan awal inventory

Jika id tidak valid:

```
Ketikkan id untuk menampilkan detail item:
id harus terisi!
Ketikkan id untuk menampilkan detail item:
d
Pilihan id tidak tersedia!
Ketik 'KELUAR' untuk keluar dari inventory.
Ketikkan id untuk menampilkan detail item:
7
Pilihan id tidak tersedia!
Ketikkan id untuk menampilkan detail item:
█
```

Gambar 7.8.2 Tampilan validasi id

Jika id merujuk pada monster (misal id bernilai 1):

```
Ketikkan id untuk menampilkan detail item:
1
Monster
Name      : Pikachu
ATK Power : 125
DEF Power : 10
HP        : 600
Level     : 2
Ketikkan id untuk menampilkan detail item:
█
```

Gambar 7.8.3 Hasil saat input id di inventory

Jika id merujuk pada potion (misal id bernilai 3):

```
Ketikkan id untuk menampilkan detail item:
3
Potion
Type      : strength
Quantity  : 5
Ketikkan id untuk menampilkan detail item:
█
```

Gambar 7.8.4 Tampilan item potion di inventory

Jika id merujuk pada monster ball (misal id bernilai 6):

```
6
Monster Ball
Quantity : 2
Ketikkan id untuk menampilkan detail item:
```

Gambar 7.8.5 Tampilan item monster ball

Jika id bernilai “KELUAR”:

```
Ketikkan id untuk menampilkan detail item:  
KELUAR  
(base) aleta@iMac TUBES DASPRO %
```

Gambar 7.8.6 Tampilan validasi keluar dari inventory

9. F08 - Battle

Tampilan awal :

Gambar 7.9.1 Tampilan awal battle

Saat memilih monster :

```
===== MONSTER LIST =====
1. Pikachu
2. Zee
3. Falcon
Pilih monster untuk bertarung (masukkan angka saja) : 4
Pilihan nomor tidak tersedia!
Pilih monster untuk bertarung (masukkan angka saja) : 3

      /----\_
      /        \  /|
      | 0 0 | / | | | |
      | .vvvvv.| / |
      | | :^_____| / |
      | | | | | |
      \ | | | | |
      \_\_\_\_\_\_\\

RAWRRR, Agent raihan mengeluarkan monster Falcon !!!
Name      : Falcon
ATK Power : 120 --> 168
DEF Power : 30 --> 42
HP       : 2200 --> 3080
Level     : 5

===== TURN 1 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : █
```

Gambar 7.9.2 Tampilan saat memilih monster

Saat memilih perintah dan menggunakan potion strength :

```
===== TURN 1 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 4
Perintah tidak ditemukan.
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 4) - Increases ATK Power
2. Resilience Potion (Qty: 6) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 5
Perintah tidak ditemukan.
Pilih perintah (masukkan angka saja) : 4

===== TURN 1 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 4) - Increases ATK Power
2. Resilience Potion (Qty: 6) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 1

Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Falcon dan gerakannya menjadi lebih cepat dan mematikan.
Update ATK Power : 168 --> 176

===== TURN 1 (Zuko) =====
SCHWINKKK, Zuko menyerang Falcon !!!
Name : Falcon
ATK Power : 176
DEF Power : 42
HP : 3080 --> 3006
Level : 5
#Penjelasan Battle - ATT : Attack = 126 ; Reduced by = 52.92 (42%) ; Attack Results = 73.08
#Penjelasan Battle - HP : Attack Results = 73.08 ; Last HP = 3080 ; Final HP = FLOOR(3080 - 73.08) = 3006
```

Gambar 7.9.3 Tampilan saat menggunakan potion strength

Saat memilih perintah dan menggunakan potion resilience :

```
===== TURN 2 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 3) - Increases ATK Power
2. Resilience Potion (Qty: 6) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 2

Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Falcon yang membuatnya terlihat semakin tangguh dan sulit dilukai.
Update DEF Power : 42 --> 44

===== TURN 2 (Zuko) =====
SCHWINKKK, Zuko menyerang Falcon !!!
Name : Falcon
ATK Power : 176
DEF Power : 44
HP : 3006 --> 2935
Level : 5
#Penjelasan Battle - ATT : Attack = 126 ; Reduced by = 55.44 (44%) ; Attack Results = 70.56
#Penjelasan Battle - HP : Attack Results = 70.56 ; Last HP = 3006 ; Final HP = FLOOR(3006 - 70.56) = 2935
```

Gambar 7.9.4 Tampilan saat menggunakan potion resilience

Saat memilih perintah dan menggunakan potion healing :

```

===== TURN 3 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 3) - Increases ATK Power
2. Resilience Potion (Qty: 5) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 3

Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh Falcon sembuh dengan cepat.
Dalam sekejap, Falcon terlihat kembali prima dan siap melanjutkan pertempuran.
Update HP : 2935 --> 3080

===== TURN 3 (Zuko) =====
SCHWINKKK, Zuko menyerang Falcon !!!
Name : Falcon
ATK Power : 176
DEF Power : 44
HP : 3080 --> 3009
Level : 5
#Penjelasan Battle - ATT : Attack = 126 ; Reduced by = 55.44 (44%) ; Attack Results = 70.56
#Penjelasan Battle - HP : Attack Results = 70.56 ; Last HP = 3080 ; Final HP = FLOOR(3080 - 70.56) = 3009

```

Gambar 7.9.5 Tampilan saat menggunakan potion healing

Saat memilih perintah dan memilih menyerang :

```

===== TURN 4 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 1

SCHWINKKK, Falcon menyerang Zuko !!!
Name : Zuko
ATK Power : 126
DEF Power : 36
HP : 2365 --> 2252
Level : 2
#Penjelasan Battle - ATT : Attack = 176 ; Reduced by = 63.36 (36%) ; Attack Results = 112.64
#Penjelasan Battle - HP : Attack Results = 112.64 ; Last HP = 2365 ; Final HP = FLOOR(2365 - 112.64) = 2252

===== TURN 4 (Zuko) =====
SCHWINKKK, Zuko menyerang Falcon !!!
Name : Falcon
ATK Power : 176
DEF Power : 44
HP : 3009 --> 2938
Level : 5
#Penjelasan Battle - ATT : Attack = 126 ; Reduced by = 55.44 (44%) ; Attack Results = 70.56
#Penjelasan Battle - HP : Attack Results = 70.56 ; Last HP = 3009 ; Final HP = FLOOR(3009 - 70.56) = 2938

===== TURN 5 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : ■

```

Gambar 7.9.6 Hasil saat menyerang

Saat memilih quit :

```
===== TURN 5 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 3
Anda berhasil kabur dari BATTLE
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> █
```

Gambar 7.9.7 Tampilan saat memilih quit dari battle

Saat memilih potion yang telah diminum sebelumnya :

```
===== TURN 1 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 4) - Increases ATK Power
2. Resilience Potion (Qty: 6) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 1

Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Falcon dan gerakannya menjadi lebih cepat dan mematikan.
Update ATK Power : 168 --> 176

===== TURN 1 (Pikachow) =====
SCHWINNKKK, Pikachu menyerang Falcon !!!
Name : Falcon
ATK Power : 176
DEF Power : 42
HP : 3080 --> 3022
Level : 5
#Penjelasan Battle - ATT : Attack = 100 ; Reduced by = 42.0 (42%) ; Attack Results = 58.0
#Penjelasan Battle - HP : Attack Results = 58.0 ; Last HP = 3080 ; Final HP = FLOOR(3080 - 58.0) = 3022

===== TURN 2 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 3) - Increases ATK Power
2. Resilience Potion (Qty: 6) - Increases DEF Power
3. Healing Potion (Qty: 5) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 1
Kamu mencoba memberikan ramuan ini kepada Pikachu, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.
```

Gambar 7.9.8 Tampilan saat memilih potion yang sudah diminum sebelumnya

Saat memilih potion yang habis :

```
===== TURN 1 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 2

===== POTION LIST =====
1. Strength Potion (Qty: 0) - Increases ATK Power
2. Resilience Potion (Qty: 2) - Increases DEF Power
3. Healing Potion (Qty: 1) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : 1
Wah, kamu sedang tidak memiliki ramuan ini, silakan pilih ramuan lain!

===== POTION LIST =====
1. Strength Potion (Qty: 0) - Increases ATK Power
2. Resilience Potion (Qty: 2) - Increases DEF Power
3. Healing Potion (Qty: 1) - Restores Health
4. Cancel
Pilih perintah (masukkan angka saja) : █
```

Gambar 7.9.9 Tampilan saat potion habis

Saat menang :

```
===== TURN 20 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 1

SCHWINKKK, Falcon menyerang Bulbu !!!
Name : Bulbu
ATK Power : 105
DEF Power : 37
HP : 36 --> 0
Level : 1
#Penjelasan Battle - ATT : Attack = 168 ; Reduced by = 62.16 (37%) ; Attack Results = 105.84
#Penjelasan Battle - HP : Attack Results = 105.84 ; Last HP = 36 ; Final HP = FLOOR(36 - 105.84) = 0

Selamat, Anda berhasil mengalahkan monster Bulbu !!!
Total OC yang diperoleh : 10
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> █
```

Gambar 7.9.10 Tampilan saat menang

Saat kalah :

```
===== TURN 28 (Pikachow) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 1

SCHWINKKK, Pikachu menyerang Geprek !!!
Name : Geprek
ATK Power : 169
DEF Power : 32
HP : 858 --> 776
Level : 4
#Penjelasan Battle - ATT : Attack = 120 ; Reduced by = 38.4 (32%) ; Attack Results = 81.6
#Penjelasan Battle - HP : Attack Results = 81.6 ; Last HP = 858 ; Final HP = FLOOR(858 - 81.6) = 776

===== TURN 28 (Geprek) =====
SCHWINKKK, Geprek menyerang Pikachu !!!
Name : Pikachu
ATK Power : 120
DEF Power : 48
HP : 24 --> 0
Level : 3
#Penjelasan Battle - ATT : Attack = 169 ; Reduced by = 81.12 (48%) ; Attack Results = 87.88
#Penjelasan Battle - HP : Attack Results = 87.88 ; Last HP = 24 ; Final HP = FLOOR(24 - 87.88) = 0
Yahhh, Anda dikalahkan monster Geprek. Jangan menyerah, coba lagi !!!
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> █
```

Gambar 7.9.11 Tampilan saat kalah

10. F09 - Arena

Saat awal dan memilih monster :

```
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-6) ITB\if1210-2024-tubes-k05-c> python tes_arena.py default
Selamat datang di program OWCA!
>>> ARENA

      /\---/\_
      /  o   o \ / |
      |  .vvvv. | / |
      /  | | ^^^^ | /
      | /| |   / |
      / | |   |
      \ | |   |
      \..\\ \..\\

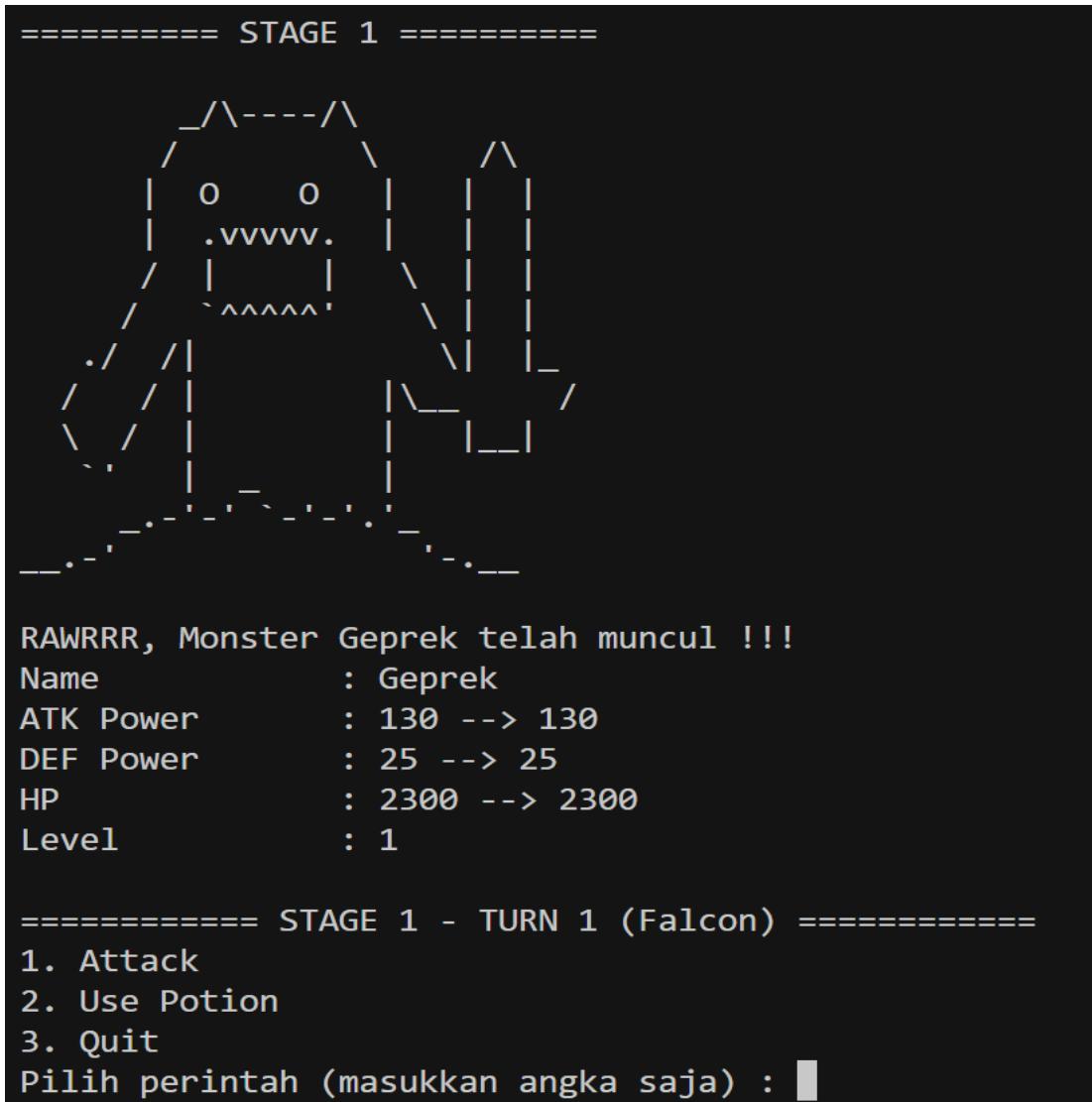
Selamat datang di Arena !!!
=====
1. Pikachuow
2. Zeze
3. Falcon
Pilih monster untuk bertarung (masukkan angka saja) : 4
Pilihan nomor tidak tersedia!
Pilih monster untuk bertarung (masukkan angka saja) : 3

      /\---/\_
      /  o   o \ / |
      |  .vvvv. | / |
      /  | | ^^^^ | /
      | /| |   / |
      / | |   |
      \ | |   |
      \..\\ \..\\

RAWRR, Agent raihan mengeluarkan monster Falcon !!!
Name      : Falcon
ATK Power : 120 --> 168
DEF Power : 30 --> 42
HP        : 2200 --> 3080
Level     : 5
```

Gambar 7.10.1 Tampilan memilih monster di arena

Saat memasuki stage 1 :



Gambar 7.10.2 Tampilan saat memasuki stage

Saat menang stage 1 dan memasuki stage 2 :

Gambar 7.10.3 Tampilan menang stage 1 dan memasuki stage 2

Saat menang stage 2 dan memasuki stage 3 :

Gambar 7.10.4 Tampilan menang stage 2 dan memasuki stage 3

Saat menang stage 3 dan memasuki stage 4 :

Gambar 7.10.5 Tampilan menang stage 3 dan memasuki stage 4

Saat menang stage 4 dan memasuki stage 5 :

Gambar 7.10.6 Tampilan menang stage 4 dan memasuki stage 5

Saat menang stage 5 :

```
===== STAGE 5 - TURN 32 (Falcon) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 1

SCHWINKKK, Falcon menyerang Zuko !!!
Name : Zuko
ATK Power : 161
DEF Power : 46
HP : 70 --> 0
Level : 5
#Penjelasan Battle - ATT : Attack = 193 ; Reduced by = 88.78 (46%) ; Attack Results = 104.22
#Penjelasan Battle - HP : Attack Results = 104.22 ; Last HP = 70 ; Final HP = FLOOR(70 - 104.22) = 0

Selamat, Anda berhasil mengalahkan monster Zuko !!!
STAGE CLEARED! Anda akan mendapatkan 149 OC pada sesi ini.
```

Gambar 7.10.7 Tampilan menang 5

Saat memenangkan semua stage :

```
Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!  
===== STATS =====  
Total Hadiah : 446 OC  
Jumlah Turn : 125 Turn  
Jumlah Stage : 5 Stage  
Potion Digunakan : 5 Potion  
HP Recovered : 9786 HP  
Damage Diberikan : 12900 Damage  
Damage Diterima : 9786 Damage
```

Gambar 7.10.8 Memenangkan semua stage

Saat kabur dari permainan :

```
===== STAGE 2 - TURN 2 (Falcon) =====  
1. Attack  
2. Use Potion  
3. Quit  
Pilih perintah (masukkan angka saja) : 3  
Anda berhasil kabur dari BATTLE  
  
GAME OVER! Anda mengakhiri sesi latihan!  
===== STATS =====  
Total Hadiah : 36 OC  
Jumlah Turn : 22 Turn  
Jumlah Stage : 1 Stage  
Potion Digunakan : 3 Potion  
HP Recovered : 1619 HP  
Damage Diberikan : 2513 Damage  
Damage Diterima : 1696 Damage  
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> █
```

Gambar 7.10.9 Tampilan saat kabur dari permainan

Saat kalah dalam suatu stage :

```

===== STAGE 2 - TURN 27 (Bulbu) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah (masukkan angka saja) : 1

SCHWINKKK, Bulbu menyerang Zeze !!!
Name : Zeze
ATK Power : 121
DEF Power : 38
HP : 660 --> 594
Level : 2
#Penjelasan Battle - ATT : Attack = 105 ; Reduced by = 39.9 (38%) ; Attack Results = 65.1
#Penjelasan Battle - HP : Attack Results = 65.1 ; Last HP = 660 ; Final HP = FLOOR(660 - 65.1) = 594

===== STAGE 2 - TURN 27 (Zeze) =====
SCHWINKKK, Zeze menyerang Bulbu !!!
Name : Bulbu
ATK Power : 105
DEF Power : 38
HP : 53 --> 0
Level : 1
#Penjelasan Battle - ATT : Attack = 121 ; Reduced by = 45.98 (38%) ; Attack Results = 75.02
#Penjelasan Battle - HP : Attack Results = 75.02 ; Last HP = 53 ; Final HP = FLOOR(53 - 75.02) = 0

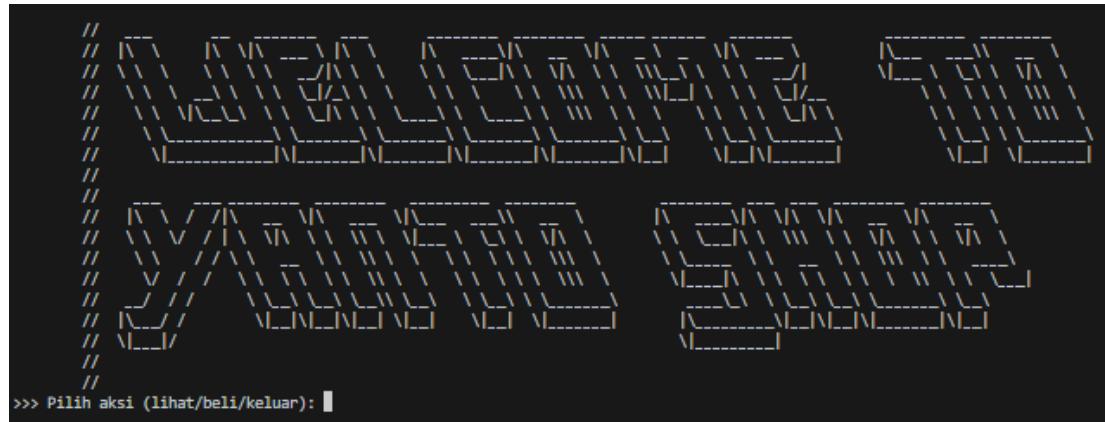
Yahhh, Anda dikalahkan monster Zeze. Jangan menyerah, coba lagi !!!
GAME OVER! Sesi latihan berakhir pada stage 2!

===== STATS =====
Total Hadiah : 27 OC
Jumlah Turn : 59 Turn
Jumlah Stage : 1 Stage
Potion Digunakan : 2 Potion
HP Recovered : 2170 HP
Damage Diberikan : 3816 Damage
Damage Diterima : 4220 Damage
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> █

```

Gambar 7.10.10 Tampilan saat kalah dalam suatu stage

11. F10 - Shop & Currency



Gambar 7.11.1 Tampilan awal yanto shop

```

>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID | Type | ATK Power | DEF Power | Stok | Harga
2  | Bulbu | 50        | 50        | 4    | 700
3  | Zeze  | 300       | 10        | 3    | 1000
4  | Zuko  | 100       | 25        | 8    | 550
5  | Chacha| 80        | 30        | 7    | 600
>>> Pilih aksi (lihat/beli/keluar): █

```

Gambar 7.11.2 Tampilan jika memilih lihat

```
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID | Type | Stok | Harga
1 | strength | 10 | 50
2 | resilience | 5 | 30
>>> Pilih aksi (lihat/beli/keluar):
```

Gambar 7.11.3 Tampilan jika memilih lihat potion

```
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID | Type | ATK Power | DEF Power | Stok | Harga
2 | Bulbu | 50 | 50 | 4 | 700
3 | Zeze | 300 | 10 | 3 | 1000
4 | Zuko | 100 | 25 | 8 | 550
5 | Chacha | 80 | 30 | 7 | 600
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. coin-mu sekarang 9090

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 5
Berhasil membeli monster: Chacha. Item sudah masuk ke inventory-mu!
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID | Type | ATK Power | DEF Power | Stok | Harga
2 | Bulbu | 50 | 50 | 4 | 700
3 | Zeze | 300 | 10 | 3 | 1000
4 | Zuko | 100 | 25 | 8 | 550
5 | Chacha | 80 | 30 | 6 | 600
>>> Pilih aksi (lihat/beli/keluar):
```

Gambar 7.11.4 Tampilan jika memilih beli suatu monster

```
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. coin-mu sekarang 9090

>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 1
>>> Masukkan jumlah: 9

Berhasil membeli item: 9 strength. Item sudah masuk ke inventory-mu!
OC-mu tersisa 8640

>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID | Type | Stok | Harga
1 | strength | 1 | 50
2 | resilience | 5 | 30
>>> Pilih aksi (lihat/beli/keluar):
```

Gambar 7.11.5 Tampilan jika membeli potion

```
>>> Pilih aksi (lihat/beli/keluar): keluar
Mr. Yanto bilang terima kasih, belanja di yanto shop lagi ya nanti :)
PS C:\Users\User\Downloads\TBIF1210-09-12\TBIF1210-09-12>
```

Gambar 7.11.6 Tampilan saat keluar yanto shop

12. F11 - Laboratory

Tampilan awal saat masuk laboratory Dokter Asep

Gambar 7.12.1 Tampilan awal laboratory Dokter Asep

Jika memilih monster yang ada di list:

```
OC yang anda miliki : 5000
===== Monster List =====
1. Pikachu (Level: 4)
2. Bulbu (Level: 3)
3. Zeze (Level: 2)
===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih Nomor Monster yang akan di Upgrade (0 untuk keluar) : 2

Bulbu akan di-upgrade ke level 4
Harga untuk melakukan upgrade Bulbu adalah 800 OC

>>> Lanjutkan upgrade (Y/N) : █
```

Gambar 7.12.1 Tampilan saat memilih monster di list

Jika konfirmasi ‘y’ untuk meng-upgrade dan OC mencukupi:

```
>>> Lanjutkan upgrade (Y/N) : Y
Selamat, Bulbu berhasil di-upgrade ke level 4!
O.W.C.A Coin anda sekarang: 4200
Level Bulbu sekarang: 4
```

Gambar 7.12.3 Tampilan jika berhasil mengupgrade

Jika konfirmasi ‘y’ untuk meng-upgrade namun OC kurang:

```
OC yang anda miliki : 300
===== Monster List =====
1. Pikachu (Level: 2)
2. Bulbu (Level: 3)
3. Zuko (Level: 4)
===== 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 Nomor Monster yang akan di Upgrade (0 untuk keluar) : 3

Zuko akan di-upgrade ke level 5
Harga untuk melakukan upgrade Zuko adalah 1000 OC

>>> Lanjutkan upgrade (Y/N) : Y
O.W.C.A Coin tidak cukup
OC yang anda miliki : 300
```

Gambar 7.12.4 Tampilan jika OC tidak cukup

Jika konfirmasi ‘n’:

```
>>> Lanjutkan upgrade (Y/N) : N
Upgrade dibatalkan
OC yang anda miliki : 4200
```

Gambar 7.12.5 Tampilan saat membatalkan upgrade

Jika monster yang dipilih sudah level maksimal:

```
===== Monster List =====
1. Pikachu (Level: 5)
2. Bulbasaur (Level: 4)
3. Zebra (Level: 2)
===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih Nomor Monster yang akan di Upgrade (0 untuk keluar): 1
Maaf, monster yang Anda pilih sudah memiliki level maksimum
OC yang anda miliki = 2200
```

Gambar 7.12.6 Tampilan saat monster yang dipilih sudah maksimum

Jika nomor monster yang dipilih tidak ada di list:

```
OC yang anda miliki : 3200
===== Monster List =====
1. Pikachu (Level: 5)
2. Bulbasaur (Level: 4)
3. Zebra (Level: 2)
===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih Nomor Monster yang akan di Upgrade (0 untuk keluar): 4
Pilihan Monster tidak valid
>>> Pilih Nomor Monster yang akan di Upgrade (0 untuk keluar): 1
```

Gambar 7.12.7 Tampilan saat monster yang dipilih tidak ada di list

13. F12 - Shop Management

Gambar 7.13.1 Tampilan awal Shop Management

```
>>> Lihat apa? (potion/monster): monster
ID      | Type           | ATK Power | DEF Power | Stok | Harga
1       | Pikachuow    | 125      | 10        | 0    | 500
2       | Bulbu          | 50       | 50        | 4    | 700
3       | Zeze           | 300      | 10        | 3    | 1000
4       | Zuko            | 100      | 25        | 8    | 550
5       | Chacha          | 80       | 30        | 7    | 600
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.2 Tampilan saat melihat monster

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID      | Type           | Stok | Harga
1       | strength       | 10   | 50
2       | resilience     | 5    | 30
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.3 Tampilan saat melihat potion

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau tambah apa? (monster/item): monster
ID      | Type           | ATK Power | DEF Power | hp
6       | Virgo          | 120      | 28        | 420
7       | Sagittarius    | 139      | 49        | 20
>>> Masukkan id monster: 7
>>> Masukkan stok baru: 77
>>> Masukkan harga monster: 777
Sagittarius telah berhasil ditambahkan ke dalam shop!
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID      | Type           | ATK Power | DEF Power | Stok | Harga
1       | Pikachuow    | 125      | 10        | 0    | 500
2       | Bulbu          | 50       | 50        | 4    | 700
3       | Zeze           | 300      | 10        | 3    | 1000
4       | Zuko            | 100      | 25        | 8    | 550
5       | Chacha          | 80       | 30        | 7    | 600
7       | Sagittarius    | 139      | 49        | 77   | 777
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.4 Tampilan saat menambah monster

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau tambah apa? (monster/item): item
ID      | Type           |
3       | healing         |
>>> masukkan id potion: 3
>>> Masukkan stok awal: 17
>>> Masukkan harga: 99

healing Potion telah berhasil ditambahkan ke dalam shop!

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID      | Type           | Stok | Harga
1       | strength       | 10   | 50
2       | resilience     | 5    | 30
3       | healing          | 17   | 99
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.5 Tampilan saat menambah potion

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion):monster
ID | Type          | ATK Power | DEF Power | Stok | Harga
1  | Pikachuow   | 125       | 10        | 0    | 500
2  | Bulbu        | 50        | 50        | 4    | 700
3  | Zeze         | 300       | 10        | 3    | 1000
4  | Zuko          | 100       | 25        | 8    | 550
5  | Chacha       | 80        | 30        | 7    | 600
>>> masukkan id monster: 3
>>> Masukkan stok baru:
>>> Masukkan harga baru: 900

Zeze telah berhasil diubah dengan stok baru sejumlah 3 dan dengan harga baru 900!

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID | Type          | ATK Power | DEF Power | Stok | Harga
1  | Pikachuow   | 125       | 10        | 0    | 500
2  | Bulbu        | 50        | 50        | 4    | 700
3  | Zeze         | 300       | 10        | 3    | 900
4  | Zuko          | 100       | 25        | 8    | 550
5  | Chacha       | 80        | 30        | 7    | 600
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.6 Tampilan saat mengubah monster

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion):potion
ID | Type          | Stok | Harga
1  | strength     | 10   | 20
2  | healing       | 30   | 15
3  | resilience   | 20   | 10
>>> masukkan id potion: 3
>>> Masukkan stok baru:
>>> Masukkan harga baru: 90
resilience Potion telah berhasil diubah dengan stok baru sejumlah 20 dan dengan harga baru 90!
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID | Type          | Stok | Harga
1  | strength     | 10   | 20
2  | healing       | 30   | 15
3  | resilience   | 20   | 90
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.13.7 Tampilan saat mengubah potion

```

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): monster
1 | Pikachu | 100 | 30 | 3 | 200
2 | Bulbu | 110 | 27 | 4 | 300
3 | Zeze | 120 | 25 | 5 | 400
4 | Zuko | 130 | 22 | 6 | 500
5 | Falcon | 140 | 20 | 7 | 600
6 | Geprek | 200 | 40 | 10 | 1000
7 | Mumbul | 250 | 50 | 10 | 2000
>>> Masukkan Id monster: 6
Apakah anda yakin ingin menghapus Geprek dari shop (y/n)? y
Geprek telah berhasil dihapus dari shop!

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): monster
ID | Type | ATK Power | DEF Power | Stok | Harga
1 | Pikachu | 100 | 30 | 3 | 200
2 | Bulbu | 110 | 27 | 4 | 300
3 | Zeze | 120 | 25 | 5 | 400
4 | Zuko | 130 | 22 | 6 | 500
5 | Falcon | 140 | 20 | 7 | 600
7 | Mumbul | 250 | 50 | 10 | 2000
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): █

```

Gambar 7.13.7 Tampilan saat menghapus monster

```

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): potion
ID | Type | Stok | Harga
1 | strength | 10 | 20
2 | healing | 30 | 15
3 | resilience | 20 | 10
>>> Masukkan Id potion: 3
Apakah anda yakin ingin menghapus resilience Potion dari shop (y/n)? y
resilience telah berhasil dihapus dari shop!

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Lihat apa? (potion/monster): potion
ID | Type | Stok | Harga
1 | strength | 10 | 20
2 | healing | 30 | 15
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): █

```

Gambar 7.13.8 Tampilan saat menghapus potion

```

>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): keluar
Mr. Yanto bilang terima kasih sudah bantu mengatur tokonya menjadi lebih baik :)
```

Gambar 7.13.9 Tampilan saat keluar

*kembali ke main program

14. F13 - Monster Management



Gambar 7.14.1 Tampilan saat memasuki Monster Management

Jika aksi tidak tervalidasi:

```
Pilih aksi: f
Pilihan aksi tidak tersedia, pilih kembali!
Pilih aksi: s
Pilihan aksi tidak tersedia, pilih kembali!
Pilih aksi: █
```

Gambar 7.14.2 Tampilan saat aksi tidak tervalidasi

Jika aksi bernilai 1:

Pilih aksi: 1					
ID	Type	ATK Power	DEF Power	HP	
1	Pikachow	125	10	600	
2	Bulbu	50	50	1200	
3	Zeze	300	10	100	
4	Zuko	100	25	800	

Apakah mau melakukan hal lain? (Y/N): █

Gambar 7.14.3 Tampilan saat aksi 1

Jika jawaban melanjutkan aktivitas pada fungsi tidak tervalidasi:

```
Apakah mau melakukan hal lain? (Y/N): h
Pilihan tidak tersedia, pilih kembali!
Apakah mau melakukan hal lain? (Y/N): g
Pilihan tidak tersedia, pilih kembali!
Apakah mau melakukan hal lain? (Y/N): █
```

Gambar 7.14.4 Tampilan jika jawaban tidak tervalidasi

Jika jawaban melanjutkan aktivitas pada fungsi bernilai ‘Y’ atau ‘y’:

```
Apakah mau melakukan hal lain? (Y/N): Y
Pilih aksi: █
```

Gambar 7.14.5 Tampilan jika mau melanjutkan aktivitas

Jika jawaban melanjutkan aktivitas pada fungsi bernilai ‘N’ atau ‘n’:

```
Apakah mau melakukan hal lain? (Y/N): N
TERIMA KASIH SUDAH MENGGUNAKAN
SELAMAT TINGGAL
```

Gambar 7.14.6 Tampilan saat keluar

Jika aksi bernilai 2:

```
Pilih aksi: 2
Memulai pembuatan monster baru..
Masukkan Type / Nama : █
```

Gambar 7.14.7 Tampilan membuat monster baru

Jika nama monster tidak valid:

```
Masukkan Type / Nama : 78a
Masukkan input bertipe string, coba lagi!
Masukkan Type / Nama : Pikachuow
Nama sudah terdaftar, coba lagi!
Masukkan Type / Nama :
Masukkan input harus bernilai, coba lagi!
Masukkan Type / Nama : █
```

Gambar 7.14.8 Tampilan validasi input monster

Jika nama monster valid:

```
Masukkan Type / Nama : rawr
Masukkan ATK Power : 
```

Gambar 7.14.9 Tampilan monster tervalidasi

Jika atk power monster tidak valid:

```
Masukkan ATK Power : js9
Masukkan input bertipe integer dan bernilai positif, coba lagi!
Masukkan ATK Power :
Masukkan input harus bernilai, coba lagi!
Masukkan ATK Power : 
```

Gambar 7.14.10 Tampilan saat atk power monster tidak valid

Jika atk power monster valid:

```
Masukkan ATK Power : 60
Masukkan DEF Power : 
```

Gambar 7.14.11 Tampilan saat atk power monster valid

Jika def power monster tidak valid:

```
Masukkan DEF Power :
Masukkan input harus bernilai, coba lagi!
Masukkan DEF Power : jj90
Masukkan bertipe integer dan bernilai positif, coba lagi!
Masukkan DEF Power : 8392
DEF Power harus bernilai 0-50, coba lagi!
Masukkan DEF Power : 
```

Gambar 7.14.12 Tampilan saat def monster tidak valid

Jika def power monster valid:

```
Masukkan DEF Power : 45
Masukkan HP : 
```

Gambar 7.14.13 Tampilan saat def monster valid

Jika hp monster tidak valid:

```
Masukkan HP : asia8
Masukkan input bertipe integer dan bernilai positif, coba lagi!
Masukkan HP :
Masukkan input harus bernilai, coba lagi!
Masukkan HP : 
```

Gambar 7.14.14 Tampilan saat hp monster tidak valid

Jika hp monster valid:

```
Masukkan HP : 67
Monster baru berhasil dibuat!
Type : rawr
ATK Power : 60
DEF Power : 45
HP : 67
Tambahkan Monster ke database (Y/N) : █
```

Gambar 7.14.15 Tampilan saat hp monster tidak valid

Jika jawaban penambahan monster tidak valid:

```
Tambahkan Monster ke database (Y/N) :
Pilihan tidak tersedia, pilih kembali!
Tambahkan Monster ke database (Y/N) : asj
Pilihan tidak tersedia, pilih kembali!
Tambahkan Monster ke database (Y/N) : █
```

Gambar 7.14.16 Tampilan saat penambahan tidak valid

Jika jawaban penambahan monster bernilai ‘Y’ atau ‘y’:

```
Tambahkan Monster ke database (Y/N) : Y
Monster baru telah ditambahkan!
Apakah mau melakukan hal lain? (Y/N): █
```

Gambar 7.14.16 Tampilan saat menambahkan ke database

Jika jawaban penambahan monster bernilai ‘N’ atau ‘n’:

```
Tambahkan Monster ke database (Y/N) : N
Monster gagal ditambahkan!
Apakah mau melakukan hal lain? (Y/N): █
```

Gambar 7.14.17 Tampilan saat tidak ditambahkan ke database

Jika aksi 1 kembali dilakukan setelah jawaban penambahan monster bernilai ‘Y’ atau ‘y’:

```
Pilih aksi: 1
ID | Type      | ATK Power | DEF Power | HP
1  | Pikachuow | 125       | 10         | 600
2  | Bulbu     | 50        | 50         | 1200
3  | Zeze      | 300       | 10         | 100
4  | Zuko      | 100       | 25         | 800
5  | rawr      | 60        | 45         | 67
Apakah mau melakukan hal lain? (Y/N): █
```

Gambar 7.14.18 Tampilan saat mau melakukan hal lain

Jika aksi 1 kembali dilakukan setelah jawaban penambahan monster bernilai 'N' atau 'n':

Pilih aksi: 1

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

Apakah mau melakukan hal lain? (Y/N):

Gambar 7.14.19 Tampilan saat keluar

15. F14 - Load

Ketika folder data ditemukan :

Gambar 7.15.1 Tampilan saat data ditemukan

Secara berurutan, ketika folder data tidak ditemukan dan ketika tidak ada nama folder yang diberikan :

```
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> python src/tes_load.py aneh
Folder aneh tidak ditemukan!
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> python src/tes_load.py
Tidak ada nama folder yang diberikan!
Usage : python main.py <nama folder>
```

Gambar 7.15.2 Tampilan saat data tidak ditemukan

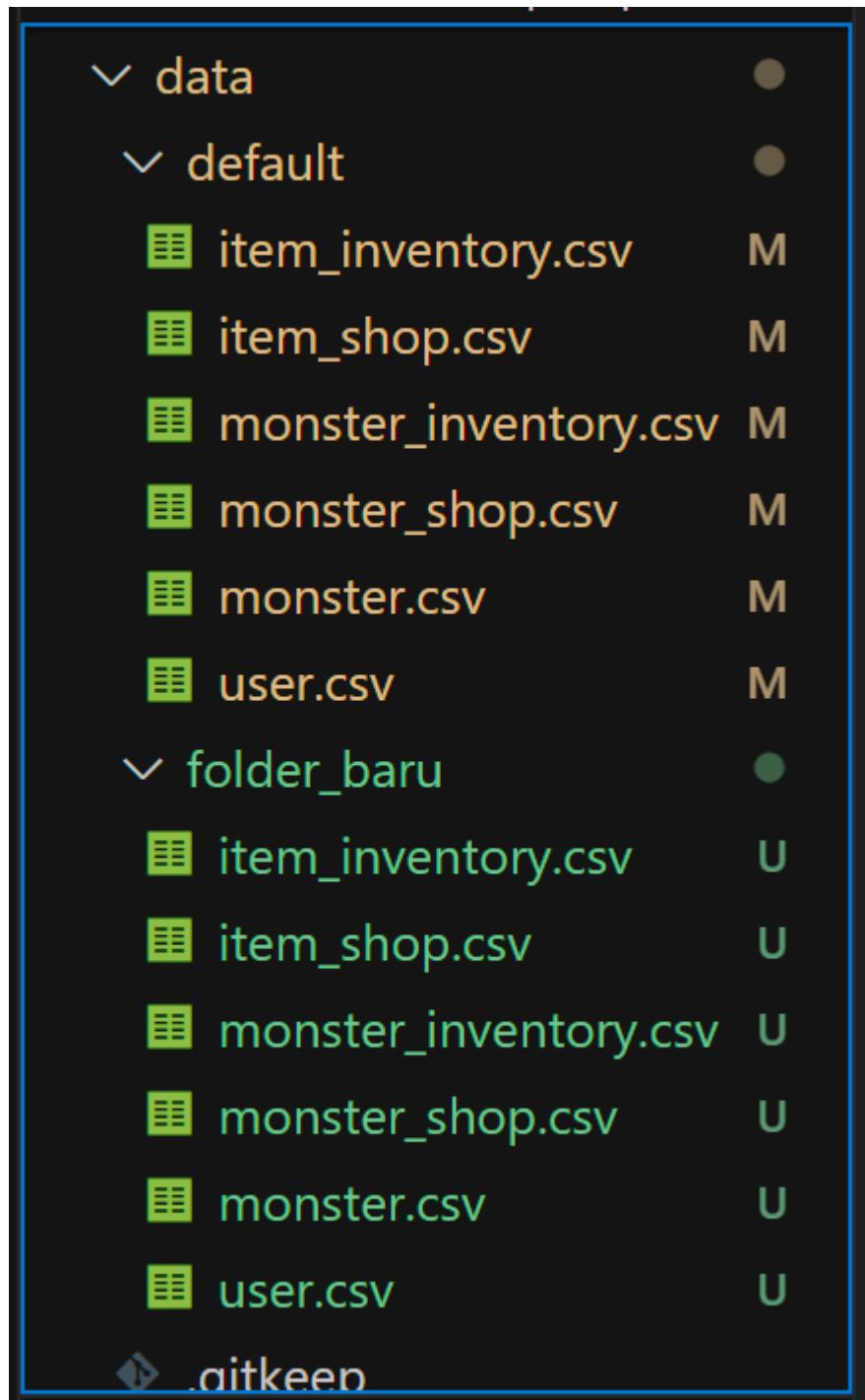
16. F15 - Save

Secara berurutan, ketika folder sudah ada sebelumnya, ketika nama folder belum ada sebelumnya, dan ketika nama folder tidak valid :

```
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> python src/tes_save.py default
Selamat datang di program OWCA!
>>> SAVE
Masukkan Nama Folder : default
Saving...
Berhasil menyimpan data di folder ./data/default!
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> python src/tes_save.py default
Selamat datang di program OWCA!
>>> SAVE
Masukkan Nama Folder : folder_baru
Saving...
Membuat folder ./data/folder_baru...
Berhasil menyimpan data di folder ./data/folder_baru!
PS D:\Muhammad Raihan Nazhim Oktana\Kuliah STEI-K (IF-G) ITB\if1210-2024-tubes-k05-c> python src/tes_save.py default
Selamat datang di program OWCA!
>>> SAVE
Masukkan Nama Folder : apaiyah?
Nama folder yang dimasukkan mengandung karakter yang dilarang yaitu (/:\*?"><|).
```

Gambar 7.16.1 Tampilan saat menyimpan data

Bukti data tersimpan :



Gambar 7.15.2 Bukti data tersimpan

17. F16 - Exit

Secara berurutan, ketika input n dan ketika input y :

Gambar 7.16.1 Tampilan exit

Lampiran *Form* Asistensi

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

Nomor Asistensi : 1
No. Kelompok/Kelas : 3 / K-05
Tanggal asistensi : Jumat, 3 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	<u>19623215 / Muhammad Raihan Nazhim Oktana</u>
2	<u>19623055 / Athian Nugraha Muarajuang</u>
3	<u>19623065 / Muhammad Alfansya</u>
4	<u>16523245 / Dede Firman</u>
5	<u>16523135 / Aleta Edna Jessalyn</u>
6	<u>-</u>

Asisten pembimbing	NIM / Nama
	<u>18221066 / Aulia Nadhirah Yasmin Badrulkamal</u>

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none">- Kerjakan tugas bagiannya per orang di file berbeda lalu dipanggil- Global isLogin tidak boleh- Boleh menambahkan kolom pada csv sebagai fungsi kreativitas- Ada default CSV lalu di update- Git add push commit- Load di awal, save di akhir- Github pakai desktop- Format commit message
Tindak Lanjut
<ul style="list-style-type: none">- Mengerjakan file sesuai bagian masing-masing dan dipanggil di main.py- Mengatur start awal isLogin sebagai False lalu diupdate saat dijalankan- Tidak menambah kolom, mengikuti spesifikasi saja- Sudah dibuat default data CSV- Sudah dipelajari caranya menggunakan Github- Sudah dibuat program sesuai spesifikasi- Menggunakan Github via desktop dan web keduanya- Menggunakan format commit message yang ditentukan
Dokumentasi
<ul style="list-style-type: none">- Waktu Meeting : 35 Menit- Screenshot :

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



Gambar 8.1.1 Bukti Asistensi 1

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

Nomor Asistensi : 2
No. Kelompok/Kelas : 3 / K-05
Tanggal asistensi : Sabtu, 11 Mei 2024

Anggota kelompok

NIM / Nama (Hanya yang Hadir)	
1	19623215 / Muhammad Raihan Nazhim Oktana
2	19623055 / Athian Nugraha Muarajuang
3	19623065 / Muhammad Alfansya
4	16523245 / Dede Firman
5	16523135 / Aleta Edna Jessalyn
6	-

Asisten pembimbing

NIM / Nama	
18221066 / Aulia Nadhirah Yasmin Badrulkamal	

Catatan Asistensi:

Rangkuman Diskusi
1. Github: tidak ada fungsi yang ilegal
2. Progress :
a. Raihan: load save udah aman tgl debugging, error handling dalam tahap pembentukan
b. athian: main program udh ada beberapa yg bs jalan
c. alfan: program sudah berjalan
d. defir: lab sudah berjalan sesuai fungsinya
e. aleta: f03, f07,f13 udh aman
3. Demo:
a. tidak ada fungsi ilegal
4. Pertanyaan:
a. semua fungsi baiknya dibuat secara manual
b. changes: masih mau dilakukan perubahan, stage changes: udh di tahap add file, atur dgn unstate untuk mengurangi atau menambah jumlah yg ingin di push.
c. Style itu ga ngubah kodennya tapi mengubah indent (hal-hal tata kode) , refactor fungsinya untuk mengubah kode sehingga lebih efisien (dari segi fungsi)
d. merename, gunakan style
e. defense di fungsi 13 diharuskan untuk bernilai 0-50
f. design kamus data, desain command: diberi referensi untuk bagian laporan
g. ketentuan font dan ukuran: times new roman, 12, asal ga comic sans
h. tubes ini dr yg membuat spesifikasi, asistensi, dan menilai, dipegang oleh dosen
i. laporan jangan terlalu deadliner, karena lumayan banyak
j. deadline daspro, pasti keluar kok (WKKWK) 

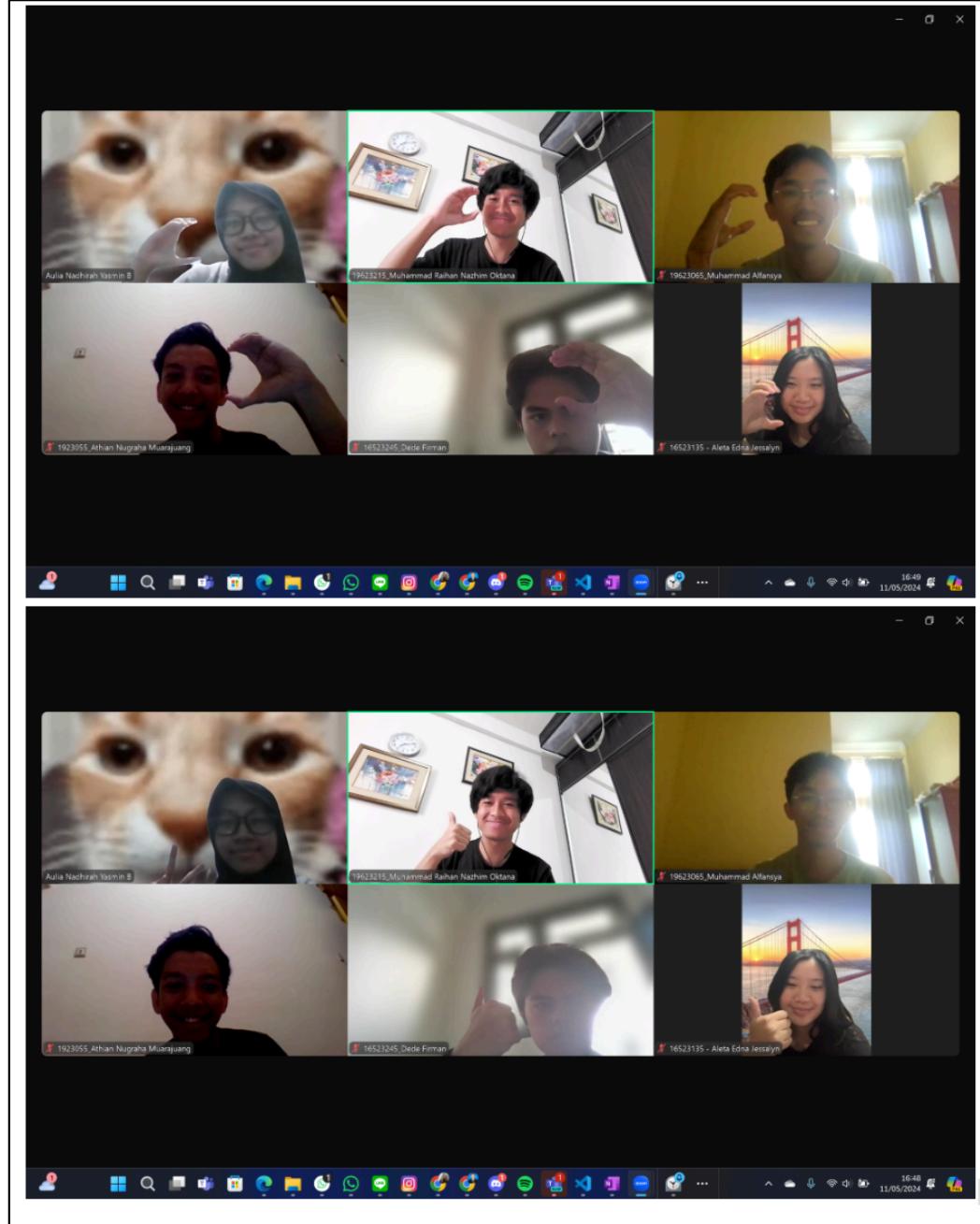
Tindak Lanjut

1. LANJUT BUAT LAPORAN YEY 
2. Semua aman, tinggal dilanjutkan.

Dokumentasi

Waktu Meeting : 50 Menit
Screenshot :

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



Gambar 8.1.2 Bukti Asistensi 2

Daftar Referensi

1.  IF1210 Spesifikasi Tugas Besar
2.  IF1210_FormAsistensiTB_1_K05-C.docx
3.  IF1210_FormAsistensiTB_2_K05-C.docx
4.  IF1210_QNA
5.  IF1210 Jadwal Demo