

LAPORAN TUGAS BESAR DASAR PEMROGRAMAN

Program O.W.C.A



Kelas/Kelompok : K07-F
Mata Kuliah : Dasar Pemrograman IF1210
Bulan/Tahun : Mei/2024

Disusun oleh :

Theo Kurniady	19623077
Muhammad Fithra Rizki	19623057
Gabriela Jennifer Sandy	19623097
Ramadhan Abhinawa Herawanto	16523207
Kennard Hezekiah Montoya	16523057

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023/2024

Pernyataan Kejujuran

“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”

1. Nama : Theo Kurniady
NIM : 19623077
2. Nama : Muhammad Fithra Rizki
NIM : 19623057
3. Nama : Gabriela Jennifer Sandy
NIM : 19623097
4. Nama : Ramadhan Abhinawa Herawanto
NIM : 16523207
5. Nama : Kennard Hezekiah Montoya
NIM : 16523057

DAFTAR ISI

Pernyataan Kejujuran	2
DAFTAR ISI	3
DAFTAR TABEL	8
DAFTAR GAMBAR	9
BAB I	11
1.1 Deskripsi Persoalan	11
BAB II	12
2.1 Pembagian Tugas	12
2.1.1 Tabel Pembagian Tugas Fitur Spesifikasi Utama	12
2.1.2 Tabel Pembagian Tugas Fitur Spesifikasi Bonus	13
2.2 Fitur	13
2.2.1 Tabel Fitur Dan Pengerjaan Spesifikasi Utama	13
2.2.2 Tabel Fitur Dan Pengerjaan Spesifikasi Bonus	14
2.3 Laporan	14
2.2.2 Tabel Tugas Dan Pembagian Tugas	14
BAB III	15
F00 - RNG	15
F01 - Register	15
F02 - Login	15
F03 - Logout	15
F04 - Menu & Help	15
F05 - Monster	15
F06 - Potion	15
F07 - Inventory	16
F08 - Battle	16
F09 - Arena	16
F10 - Shop & Currency	16
F11 - Laboratory	16
F12 - Shop Management	16
F13 - Monster Management	16
F14 - Load	17
F15 - Save	17
F16 - Exit	17
B03 - Monster Ball	17
BAB IV	18
F00 - Random Number Generator	18
F01 - Register	18
F02 - Login	18

F03 - Logout	19
F04 - Menu & Help	19
F05 - Monster	19
F06 - Potion	19
F07 - Inventory	19
F08 - Battle	20
F09 - Arena	21
F10 - Shop & Currency	22
F11 - Laboratory	23
F12 - Shop Management	23
F13 - Monster Management	24
F14 - Load	24
F15 - Save	25
F16 - Exit	25
B03 - Monster Ball	25
BAB V	27
F00 - Random Number Generator	27
F01 - Register	29
F02 - Login	32
F03 - Logout	33
F04 - Menu & Help	34
F05 - Monster	34
F06 - Potion	34
F07 - Inventory	35
F08 - Battle	36
F09 - Arena	44
F10 - Shop & Currency	47
F11 - Laboratory	49
F12 - Shop Management	51
F13 - Monster Management	54
F14 - Load	55
F15 - Save	57
F16 - Exit	58
BAB VI	60
F00 - Random Number Generator	60
F01 - Register	60
F02 - Login	60
F03 - Logout	60
F04 - Menu & Help	61
F05 - Monster	61
F06 - Potion	61

F07 - Inventory	61
F08 - Battle	61
F09 - Arena	62
F10 - Shop & Currency	62
F11 - Laboratory	62
F12 - Shop Management	62
F13 - Monster Management	63
F14 - Load	63
F15 - Save	63
F16 - Exit	64
B03 - Monster Ball	64
BAB VII	66
F00 - Random Number Generator	66
F01 - Register	66
Gambar 7.1.1 Registrasi	66
Gambar 7.1.2 Registrasi dengan Username yang sudah terpakai	66
Gambar 7.1.3 Registrasi dengan kesalahan Username	66
Gambar 7.1.4 Registrasi gagal	66
F02 - Login	67
Gambar 7.2.1 Log in	67
Gambar 7.2.2 Log in dengan username tidak terdaftar	67
Gambar 7.2.3 Log in dengan password salah	67
Gambar 7.2.4 Log in gagal	67
F03 - Logout	67
Gambar 7.3.1 Log out	67
Gambar 7.3.2 Logout gagal	67
F04 - Menu & Help	68
Gambar 7.4.1 Tampilan Menu & Help sebelum log in	68
Gambar 7.4.2 Tampilan Menu & Help setelah login untuk user Admin	68
Gambar 7.4.3 Tampilan Menu & Help setelah login untuk user Agent	68
F05 - Monster	68
F06 - Potion	68
F07 - Inventory	69
Gambar 7.7.1 Tampilan Inventory user	69
Gambar 7.7.2 Tampilan monster user	69
Gambar 7.7.3 Tampilan item potion user	69
Gambar 7.7.4 Tampilan item monster ball user	69
Gambar 7.7.5 Tampilan keluar dari Inventory	70
F08 - Battle	70
Gambar 7.8.1 Tampilan monster liar muncul dan pilihan monster untuk bertarung	70
Gambar 7.8.2 Tampilan gagal memilih monster (pilihan tidak tersedia)	70

Gambar 7.8.3 Tampilan monster yang dipilih dan perintah	71
Gambar 7.8.4 Tampilan kabur dari pertempuran	71
Gambar 7.8.5 Tampilan monster tidak bisa ditangkap	71
Gambar 7.8.6 Tampilan monster gagal ditangkap	72
Gambar 7.8.7 Tampilan monster berhasil ditangkap	72
Gambar 7.8.8 Tampilan menggunakan potion	72
Gambar 7.8.9 Tampilan potion habis	72
Gambar 7.8.10 Tampilan batal menggunakan potion	73
Gambar 7.8.11 Tampilan monster setelah menggunakan strength potion	73
Gambar 7.8.12 Tampilan monster setelah menggunakan resilience potion	73
Gambar 7.8.13 Tampilan monster setelah menggunakan healing potion	73
Gambar 7.8.14 Tampilan pilihan potion tidak valid	73
Gambar 7.8.15 Tampilan monster setelah menggunakan potion untuk kedua kali	73
Gambar 7.8.16 Tampilan pertarungan para monster	74
Gambar 7.8.17 Tampilan monster user berhasil memenangkan pertarungan	74
Gambar 7.8.18 Tampilan monster user gagal memenangkan pertarungan	75
F09 - Arena	75
Gambar 7.9.1 Tampilan Arena	75
Gambar 7.9.2 Tampilan gagal memilih monster (pilihan tidak tersedia)	75
Gambar 7.9.3 Tampilan user mengeluarkan monster	76
Gambar 7.9.4 Tampilan tahap pertama Arena	76
Gambar 7.9.5 Tampilan gagal menaklukan tahap pertama Arena	77
Gambar 7.9.6 Tampilan berhasil memenangkan tahapan pertama dan lanjut ke tahap berikutnya	77
Gambar 7.9.7 Tampilan berhasil memenangkan seluruh tahapan dalam Arena	77
Gambar 7.9.8 Tampilan user gagal salah satu tahapan	77
F10 - Shop & Currency	78
Gambar 7.10.1 Tampilan Shop	78
Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop	78
Gambar 7.10.3 Tampilan aksi melihat item dalam Shop	78
Gambar 7.10.4 Tampilan aksi membeli monster dalam Shop	78
Gambar 7.10.5 Tampilan aksi membeli item dalam Shop	78
Gambar 7.10.6 Tampilan aksi membeli monster yang sudah dimiliki	79
Gambar 7.10.7 Tampilan aksi membeli dengan coin (OC) yang tidak cukup	79
Gambar 7.10.8 Tampilan aksi keluar dari Shop	79
F11 - Laboratory	80
Gambar 7.11.1 Tampilan Laboratory	80
Gambar 7.11.2 Tampilan menaikkan level monster	80
Gambar 7.11.3 Tampilan monster sudah level maks	81
Gambar 7.11.4 Tampilan pilihan tidak valid	81
F12 - Shop Management	81
Gambar 7.12.1 Tampilan Shop untuk Admin	81

Gambar 7.12.2 Tampilan aksi melihat monster dalam Shop untuk Admin	81
Gambar 7.12.3 Tampilan aksi melihat item dalam Shop untuk Admin	82
Gambar 7.12.4 Tampilan aksi menambah monster dalam Shop untuk Admin	82
Gambar 7.12.5 Tampilan error message bila semua item sudah ada di shop	82
Gambar 7.12.6 Tampilan menambahkan item ke shop	83
Gambar 7.12.7 Tampilan mengubah stok dan harga monster	83
Gambar 7.12.8 Tampilan mengubah stok dan harga item	83
Gambar 7.12.9 Tampilan mengubah stok monster saja	83
Gambar 7.12.10 Tampilan mengubah harga monster saja	84
Gambar 7.12.11 Tampilan mengubah harga item saja	84
Gambar 7.12.12 Tampilan mengubah stok item saja	84
Gambar 7.12.13 Tampilan error message ketika id monster kosong	85
Gambar 7.12.14 Tampilan menghapus monster dari shop	85
Gambar 7.12.15 Tampilan menghapus item dari shop	85
Gambar 7.12.16 Tampilan keluar dari shop management	85
F13 - Monster Management	86
Gambar 7.13.1 Tampilan Monster Management untuk Admin	86
Gambar 7.13.2 Tampilan database monster	86
Gambar 7.13.3 Tampilan menambah monster	86
Gambar 7.13.4 Tampilan monster sudah terdaftar	87
Gambar 7.13.5 Tampilan gagal memasukan DEF power monster	87
Gambar 7.13.6 Tampilan keluar dari Monster Management untuk Admin	87
F14 - Load	87
Gambar 7.14.1 Tampilan Load	87
Gambar 7.14.2 Tampilan Load gagal	87
Gambar 7.14.3 Tampilan Load error	88
F15 - Save	88
Gambar 7.15.1 Tampilan save data berhasil dibuat	88
Gambar 7.15.2 Tampilan data berhasil disimpan	88
F16 - Exit	88
Gambar 7.16.1 Tampilan Exit	88
Gambar 7.16.2 Tampilan Exit dan menyimpan data	88
Gambar 7.16.1 Tampilan Exit dan tidak menyimpan data	88
B03 - Monster Ball	88
Gambar 7.7.4 Tampilan item monster ball user	89
Gambar 7.8.5 Tampilan monster tidak bisa ditangkap	89
Gambar 7.8.6 Tampilan monster gagal ditangkap	89
Gambar 7.8.7 Tampilan monster berhasil ditangkap	89
Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop	90
LAMPIRAN	91

DAFTAR TABEL

2.1.1 Tabel Pembagian Tugas Fitur Spesifikasi Utama	12
2.1.2 Tabel Pembagian Tugas Fitur Spesifikasi Bonus	13
2.2.1 Tabel Fitur Dan Pengerjaan Spesifikasi Utama	13
2.2.2 Tabel Fitur Dan Pengerjaan Spesifikasi Bonus	14

DAFTAR GAMBAR

Gambar 7.1.1 Registrasi	54
Gambar 7.1.2 Registrasi dengan Username yang sudah terpakai	54
Gambar 7.1.3 Registrasi dengan kesalahan Username	54
Gambar 7.1.4 Registrasi gagal	54
Gambar 7.2.1 Log in	55
Gambar 7.2.2 Log in dengan username tidak terdaftar	55
Gambar 7.2.3 Log in dengan password salah	55
Gambar 7.2.4 Log in gagal	55
Gambar 7.3.1 Log out	55
Gambar 7.3.2 Logout gagal	55
Gambar 7.4.1 Tampilan Menu & Help sebelum log in	56
Gambar 7.4.2 Tampilan Menu & Help setelah login untuk user Admin	56
Gambar 7.4.3 Tampilan Menu & Help setelah login untuk user Agent	56
Gambar 7.7.1 Tampilan Inventory user	57
Gambar 7.7.2 Tampilan monster user	57
Gambar 7.7.3 Tampilan item potion user	57
Gambar 7.7.4 Tampilan item monster ball user	57
Gambar 7.7.5 Tampilan keluar dari Inventory	58
Gambar 7.8.1 Tampilan monster liar muncul dan pilihan monster untuk bertarung	58
Gambar 7.8.2 Tampilan gagal memilih monster (pilihan tidak tersedia)	58
Gambar 7.8.3 Tampilan monster yang dipilih dan perintah	59
Gambar 7.8.4 Tampilan kabur dari pertempuran	59
Gambar 7.8.5 Tampilan monster disimpan dalam Inventory	59
Gambar 7.8.6 Tampilan monster gagal ditangkap	60
Gambar 7.8.7 Tampilan monster berhasil ditangkap	60
Gambar 7.8.8 Tampilan menggunakan potion	60
Gambar 7.8.9 Tampilan potion habis	60
Gambar 7.8.10 Tampilan batal menggunakan potion	61
Gambar 7.8.11 Tampilan monster setelah menggunakan strength potion	61
Gambar 7.8.12 Tampilan monster setelah menggunakan resilience potion	61
Gambar 7.8.13 Tampilan monster setelah menggunakan healing potion	61
Gambar 7.8.14 Tampilan pilihan potion tidak valid	61
Gambar 7.8.15 Tampilan monster setelah menggunakan potion untuk kedua kali	61
Gambar 7.8.16 Tampilan pertarungan para monster	62
Gambar 7.8.17 Tampilan monster user berhasil memenangkan pertarungan	62
Gambar 7.8.18 Tampilan monster user gagal memenangkan pertarungan	63
Gambar 7.9.1 Tampilan Arena	63
Gambar 7.9.2 Tampilan gagal memilih monster (pilihan tidak tersedia)	63

Gambar 7.9.3 Tampilan user mengeluarkan monster	64
Gambar 7.9.4 Tampilan tahap pertama Arena	64
Gambar 7.9.5 Tampilan gagal menaklukan tahap pertama Arena	65
Gambar 7.9.6 Tampilan berhasil memenangkan tahapan pertama dan lanjut ke tahap berikutnya	65
Gambar 7.9.7 Tampilan berhasil memenangkan seluruh tahapan dalam Arena	65
Gambar 7.9.8 Tampilan user gagal salah satu tahapan	65
Gambar 7.10.1 Tampilan Shop	66
Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop	66
Gambar 7.10.3 Tampilan aksi melihat item dalam Shop	66
Gambar 7.10.4 Tampilan aksi membeli monster dalam Shop	66
Gambar 7.10.5 Tampilan aksi membeli item dalam Shop	66
Gambar 7.10.6 Tampilan aksi membeli monster yang sudah dimiliki	67
Gambar 7.10.7 Tampilan aksi membeli dengan coin (OC) yang tidak cukup	67
Gambar 7.10.8 Tampilan aksi keluar dari Shop	67
Gambar 7.11.1 Tampilan Laboratory	68
Gambar 7.11.2 Tampilan menaikkan level monster	68
Gambar 7.12.1 Tampilan Shop untuk Admin	69
Gambar 7.12.2 Tampilan aksi melihat monster dalam Shop untuk Admin	69
Gambar 7.12.3 Tampilan aksi melihat item dalam Shop untuk Admin	69
Gambar 7.12.4 Tampilan aksi menambah monster dalam Shop untuk Admin	69
Gambar 7.13.1 Tampilan Monster Management untuk Admin	70
Gambar 7.13.2 Tampilan database monster	70
Gambar 7.13.3 Tampilan menambah monster	70
Gambar 7.13.4 Tampilan monster sudah terdaftar	71
Gambar 7.13.5 Tampilan gagal memasukan DEF power monster	71
Gambar 7.13.6 Tampilan keluar dari Monster Management untuk Admin	71
Gambar 7.14.1 Tampilan Load	71
Gambar 7.14.2 Tampilan Load gagal	71
Gambar 7.14.3 Tampilan Load error	72
Gambar 7.15.1 Tampilan save data berhasil dibuat	72
Gambar 7.15.2 Tampilan data berhasil disimpan	72
Gambar 7.16.1 Tampilan Exit	72
Gambar 7.16.2 Tampilan Exit dan menyimpan data	72
Gambar 7.16.1 Tampilan Exit dan tidak menyimpan data	72
Gambar 7.8.5 Tampilan monster disimpan dalam Inventory	73
Gambar 7.8.6 Tampilan monster gagal ditangkap	73
Gambar 7.8.7 Tampilan monster berhasil ditangkap	73
Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop	73

BAB I

1.1 Deskripsi Persoalan

Dasar pemrograman merupakan hal yang penting bagi mahasiswa agar berkembang dan melatih skill untuk menciptakan suatu program. Pada tugas besar Dasar Pemrograman IF1210, kami diminta untuk membantu Agent P, Purry si Platypus, untuk mengalahkan Dr. Asep Spakbor yang menciptakan monster-monster untuk mengancam keamanan kota Danville.

Purry si Platypus, Agent P, tidak dapat menangani ancaman dari Dr. Asep Spakbor. Ia menyadari bahwa ia membutuhkan bantuan kami, para agen di markas rahasia O.W.C.A. Oleh sebab itu, ia meminta bantuan kami untuk mengalahkan Dr. Asep Spakbor dengan memulai misi pencarian monster dan menantang monster-monster yang berkeliaran di kota Danville dengan monster kami.

Agar kami selalu ingat tujuan, visi, misi, barang-barang, dan para monster kami, kami menciptakan program agar dapat mengingatnya. Selain itu, kami mendapatkan beberapa bantuan dari Organisasi Warga Cool Abiez (OWCA) dalam bentuk pelatihan monster, peningkatan monster, maupun barang-barang. Program-program tersebut terdiri dari :

1. Random Number Generator
2. Register
3. Login
4. Logout
5. Menu & Help
6. Monster
7. Potion
8. Inventory
9. Battle
10. Arena
11. Shop & Currency
12. Laboratory
13. Shop Management
14. Monster Management
15. Load
16. Save
17. Exit

Kami juga mendapatkan bonus dari Agent P agar kami mudah menjelajah dan mengumpulkan monster guna mengalahkan Dr. Asep Spakbor. Bonus tersebut antara lain:

1. Git Best Practice
2. Typing
3. Monster Ball
4. Jackpot
5. Peta Kota Danville
6. Kreativitas

BAB II

Penutup

2.1 Pembagian Tugas

2.1.1 Tabel Pembagian Tugas Fitur Spesifikasi Utama

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00–RNG	Function rngLevel, rngEnemy, rngOC, rngCapture, playerAtk, enemyAtk	19623077	19623077	19623077
F01–Register	Function REGISTER	19623077	19623077	19623077
F02–Login	Function LOGIN	19623077	19623077	19623077
F03–Logout	Function LOGOUT	19623077	19623077	19623077
F04–Menu & Help	Function HELP	19623077	19623077	19623077
F05–Monster	Function MONSTER	19623097	19623097	19623077
F06–Potion	Function POTION	19623097	19623097	19623077
F07–Inventory	Function INVENTORY		19623077	19623077
F08–Battle	Function BATTLE	19623077	19623077	19623077
F09–Arena	Function ARENA	19623057	19623057	19623077
F10–Shop & Currency	Function SHOP	19623057	19623057	19623077
F11–Laboratory	Function laboratory	16523207	16523207	19623077
F12–Shop Management	Function SHOP_MANAGE MENT	19623057	19623057	19623077
F13–Monster Management	Function MONSTER	19623097	19623097	19623077
F14–Load	Function load	19623097 16523057	19623097	19623077
F15–Save	Function save	19623097	19623097	19623077

		16523057		
F16–Exit	Procedure EXIT	16523207	16523207	19623077

2.1.2 Tabel Pembagian Tugas Fitur Spesifikasi Bonus

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
B01–Git Best Practice		19623077	19623077	19623077
B02–Typing				
B03–Monster Ball		19623077	19623077	19623077
B04–JACKPOT				
B05–Peta Kota Danville				
BXX–Kreativitas				

2.2 Fitur

2.2.1 Tabel Fitur Dan Pengerjaan Spesifikasi Utama

Fitur	Pengerjaan
F00–RNG	<input checked="" type="checkbox"/>
F01–Register	<input checked="" type="checkbox"/>
F02–Login	<input checked="" type="checkbox"/>
F03–Logout	<input checked="" type="checkbox"/>
F04–Menu & Help	<input checked="" type="checkbox"/>
F05–Monster	<input checked="" type="checkbox"/>
F06–Potion	<input checked="" type="checkbox"/>
F07–Inventory	<input checked="" type="checkbox"/>
F08–Battle	<input checked="" type="checkbox"/>
F09–Arena	<input checked="" type="checkbox"/>
F10–Shop & Currency	<input checked="" type="checkbox"/>

F11–Laboratory	<input checked="" type="checkbox"/>
F12–Shop Management	<input checked="" type="checkbox"/>
F13–Monster Management	<input checked="" type="checkbox"/>
F14–Load	<input checked="" type="checkbox"/>
F15–Save	<input checked="" type="checkbox"/>
F16–Exit	<input checked="" type="checkbox"/>

2.2.2 Tabel Fitur Dan Pengerjaan Spesifikasi Bonus

Fitur	Pengerjaan
B01–Git Best Practice	<input checked="" type="checkbox"/>
B02–Typing	<input checked="" type="checkbox"/>
B03–Monster Ball	<input checked="" type="checkbox"/>
B04–JACKPOT	<input type="checkbox"/>
B05–Peta Kota Danville	<input type="checkbox"/>
BXX–Kreativitas	<input checked="" type="checkbox"/>

2.3 Laporan

2.2.2 Tabel Tugas Dan Pembagian Tugas

Tugas	NIM	Catatan
Asistensi	16523057	Asistensi 1 Asistensi 2 Kalender Meeting
Form Asistensi MOM	16523057 19623097	
Laporan	19623077 19623057 19623097 16523207 16523057	Bab I – Bab IV (16523057) Bab V (19623077, 19623057 , 19623097, 16523207, 16523057) Bab VI (16523057) Bab VII (19623077,16523057) Lampiran (16523057

BAB III

DESAIN COMMAND UNTUK SETIAP PRIMITIF

F00 - RNG

Nama command :
Masukan : -
Keluaran : {Random Number}

F01 - Register

Nama command : REGISTER ()
Masukan : username, password, pilih monster awal
Keluaran : Selamat datang Agent (username). Mari kita mengalahkan Dr. Asep Spakbor dengan (monster awal)!

F02 - Login

Nama command : LOGIN (userpas)
Masukan : username, password
Keluaran : Selamat datang, Agent (username)! Masukkan command “help” untuk daftar command yang dapat kamu panggil

F03 - Logout

Nama command : LOGOUT (currentUser, login)
Masukan : -
Keluaran : Anda berhasil logout!!

F04 - Menu & Help

Nama command : HELP (currentUser)
Masukan : role
Keluaran : {List command sesuai role pengguna}

F05 - Monster

Nama command :
Masukan : -
Keluaran : -

F06 - Potion

Nama command :
Masukan : -
Keluaran : -

F07 - Inventory

Nama command : INVENTORY (currentUser, mInv, iInv, mons)
Masukan : INVENTORY , ID
Keluaran : 'INVENTORY LIST :'
{Coin, Item, dan Monster}

F08 - Battle

Nama command : BATTLE (mons, mInv, iInv, rngEnemy, currentUser, rngLevel)

Masukan : BATTLE, Pilih Monster, Pilihan Perintah.
Keluaran : {Program Battle Monster dan Coin OWCA}

F09 - Arena

Nama command :
Masukan : ARENA, Pilih Monster, Pilihan Perintah.
Keluaran : {Program Battle Monster dan Coin OWCA}

F10 - Shop & Currency

Nama command :
Masukan : SHOP, Pilih Aksi
Keluaran : Berhasil membeli item: (Item).. Item sudah masuk ke inventory-mu!

F11 - Laboratory

Nama command :
Masukan : LABORATORY, Pilih Aksi, Pernyataan Upgrade
Keluaran : Selamat, (monster yang dipilih) berhasil di-upgrade ke level (angka) !

F12 - Shop Management

Nama command :
Masukan : SHOP, admin, pilih aksi
Keluaran : (Item) telah berhasil (aksi) ke dalam shop!

F13 - Monster Management

Nama command :
Masukan : Monster, admin, pilih aksi
Keluaran : Monster baru telah ditambahkan!

F14 - Load

Nama command :
Masukan : Menerima input nama folder
Keluaran : userData, gameData, kepemilikan Data, riwayat Data

F15 - Save

Nama command :
Masukan : Nama folder penyimpanan
Keluaran : Data csv dalam folder yang diinput

F16 - Exit

Nama command :
Masukan : y n
Keluaran : Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)

B03 - Monster Ball

Nama command :
Masukan : Pilih Perintah
Keluaran : Anda sudah memiliki monster (Nama Monster) dalam inventory!

BAB IV

DESAIN KAMUS DATA

F00 - Random Number Generator

F01 - Register

```
##### Kamus Lokal #####
# isUsernameValid = function
# isUsernameExist = function
# role = str
# oc = int
# monsterAwal = array of string
# validUsername = bool
# notExistUsername = bool
# cnt = int
# username = string
# password = string
# tempUserpasList = array of string
# angkaMonster = int
# monster = string
# tempMInv = array of string
```

F02 - Login

```
##### Kamus Argumen #####
# userpas = 2D matrix of string

##### Kamus Lokal #####
# isPasswordCorrect = function
# isLoginValid = function
# isUsernameExist = function
# currentUser = array of string
# loginBool = bool
# wrongUsername = bool
# wrongPassword = bool
# loginUsername = string
# loginPassword = string
# userList = array of string
# passList = array of string
```

F03 - Logout

```
##### Kamus Argumen #####  
# currentUser = array of string  
# login = bool
```

```
##### Kamus Lokal #####  
# LOGOUT = function
```

F04 - Menu & Help

```
##### Kamus Argumen #####  
# currentUser = array of string
```

```
##### Kamus Lokal #####  
# HELP = function
```

F05 - Monster

```
##### Kamus Argumen #####  
# atk_power = int  
# def_power = int  
# hp = int  
# level = int
```

```
##### Kamus Lokal #####  
# monster = function
```

F06 - Potion

```
##### Kamus Argumen #####  
# type_potion = string  
# atk_power = float  
# def_power = float  
# hp = float
```

```
##### Kamus Lokal #####  
# potion = function
```

F07 - Inventory

```
##### Kamus Argumen #####  
# currentUser = array of string  
# mInv = 2D matrix of string  
# iInv = 2D matrix of string
```

```
# mons = 2D matrix of string
```

```
##### Kamus Lokal #####
```

```
# invCount = int  
# mTemp = array of string  
# iTemp = array of string  
# inventoryBool = bool  
# mInvList = function  
# iInvList = function  
# INVENTORY = function
```

F08 - Battle

```
##### Kamus Argumen #####
```

```
# mons = 2D matrix of string  
# mInv = 2D matrix of string  
# iInv = 2D matrix of string  
# rngEnemy = function  
# currentUser = array of string  
# rngLevel = function  
# userpas = 2D matrix of string
```

```
##### Kamus Lokal #####
```

```
# damage_given = int  
# damage_received = int  
# ballPresence = bool  
# strengthBool = bool  
# resilienceBool = bool  
# healingBool = bool  
# chosenEnemy = array of string  
# enemy_type = string  
# enemy_atk_power = int  
# enemy_def_power = int  
# enemy_hp = int  
# enemy_level = int  
# enemy = array of string  
# tempMons = array of string  
# userMons = 2D matrix of string  
# currentMons = array of string  
# mons_type = string  
# atk_power = int  
# def_power = int  
# hp = int  
# level = int  
# currentPot = array of int  
# strengthIndex = int  
# resilienceIndex = int  
# healingIndex = int
```

```
# turnCnt = int
# win = bool
# lose = bool
# cancel = bool
# flee = bool
# end = bool
# chooseMons = function
# userPot = function
# userBall = function
# dmgCalc = function
# monsterCapture = function
# yourTurn = function
# enemyTurn = function
# BATTLE = function
```

F09 - Arena

```
##### Kamus Argumen #####
# mons = 2D matrix of string
# mInv = 2D matrix of string
# rngEnemy = function
# currentUser = array of string
# iInv = 2D matrix of string
```

```
##### Kamus Lokal #####
# reward = 2D matrix of int
# damage_received = int
# damage_given = int
# ballPresence = bool
# total_reward = int
# stage = int
# tempMons = array of int
# userMons = 2D matrix of int
# currentMons = array of int
# mons_type = string
# atk_power = int
# def_power = int
# hp = int
# hp_reset = int
# level = int
# strengthBool = bool
# resilienceBool = bool
# healingBool = bool
# enemyrng = int
# chosenEnemy = array of string
# enemy_type = string
# enemy_atk_power = int
# enemy_def_power = int
```

```
# enemy_hp = int
# enemy_level = int
# enemy = array of int
# turnCnt = int
# win = bool
# lose = bool
# currentPot = array of int
# strengthIndex = int
# resilienceIndex = int
# healingIndex = int
# mInvList = function
# ARENA = function
```

F10 - Shop & Currency

```
##### Kamus Argumen #####
# currentUser = array of string
# mShop = 2D matrix of string
# iShop = 2D matrix of string
# mons = 2D matrix of string
# mInv = 2D matrix of string
# iInv = 2D matrix of string
```

```
##### Kamus Lokal #####
# aksi = string
# shopCount = int
# msTemp = 2D matrix of int
# isTemp = 2D matrix of int
# jenis = string
# beli = string
# id_monster = int
# mons_to_inv = array of string
# id_item = int
# jumlah_item = int
# itemCek = bool
# item_to_inv = array of string
# mShopList = function
# iShopList = function
# cekMonster = function
# cekItem = function
# SHOP = function
```

F11 - Laboratory

```
##### Kamus Argumen #####
# userpas = 2D matrix of string
# mInv = 2D matrix of string
# mons = 2D matrix of string
# currentUser = array of string

##### Kamus Lokal #####
# monsters = 2D matrix of string
# currency = int
# monster_choice = int
# monster = array of string
# upgrade_price = int
# confirm = string
# breakout = bool
# i = int
# j = int
# choice = string
# playerMons = function
# display_monster = function
# upgrade_monster = function
# LABORATORY = function
```

F12 - Shop Management

```
##### Kamus Argumen #####
# currentUser: array of string
# mShop: 2D matrix of string
# iShop: 2D matrix of string
# mons: 2D matrix of string
# item: 2D matrix of string

##### Kamus Lokal #####
# action: string
# shopCount: int
# id_mons_to_shop: int
# msTemp: 2D matrix of int
# nmsTemp: 2D matrix of int
# isTemp: 2D matrix of int
# nisTemp: 2D matrix of int
# mons_to_shop: array of int
# item_to_shop: array of int
# jenis: string
# jenis_tambah: string
# jenis_ubah: string
# jenis_hapus: string
# id_mons_new: int
# stok_mons_new: string
```

```
# price_mons_new: string
# id_pot_new: int
# stok_pot_new: string
# price_pot_new: string
# id_mons_delete: int
# selectedmShopid: int
# selectedMons: string
# mons_confirm: string
# id_pot_delete: int
# selectedPot: string
# pot_confirm: string
```

```
##### Fungsi #####
# mShopList: function
# notmShopList: function
# iShopList: function
# notiShopList: function
# SHOP_MANAGEMENT: function
```

F13 - Monster Management

```
##### Kamus argumen #####
# Aksi = integer
# nama_baru = string
# atk_power = integer
# def_power = integer
# hp = integer
# add_mons = string
```

```
##### Kamus Lokal #####
# MONSTER = function
# pengulangan = bool
```

F14 - Load

```
##### Kamus Argumen #####
# filename: string
# folder_path: string
# folder: string
```

```
##### Kamus Lokal #####
# data: 2D matrix of string
# data_directory: string
# available_folders: list of string
# file_path: string
# line: string
```



```
# row: list of string
# cell: string
# i: int
# missing_file: bool
# valid_load: bool
# userpas: 2D matrix of string
# mons: 2D matrix of string
# mInv: 2D matrix of string
# iInv: 2D matrix of string
# iShop: 2D matrix of string
# mShop: 2D matrix of string
# parser: ArgumentParser
# args: Namespace
# read_csv: function
# show_folders: function
# load: function
# start: function
```

F15 - Save

```
##### Kamus Argumen #####
# folder_path: string
# csv_file: string
# array_data: 2D matrix of string
```

```
##### Kamus Lokal #####
# folder_name: string
# data_directory: string
# folder_path: string
# line: string
# i: int
# j: int
# write_csv: function
# save_data: function
```

F16 - Exit

```
###Kamus Argumen###
# response = string
# EXIT = function
```

B03 - Monster Ball

```
##### Kamus Argumen #####
# iInv = list
# userBall = function
```

```
# userMons = list
# iInv = list
# rngCapture = function
# mInv = list
# enemy_type = string
# currentUser = list
# chosenEnemy = list
# enemy_level = integer
# enemy_atk_power = integer
# enemy_def_power = integer
# enemy_hp = integer
```

```
##### Kamus Lokal #####
```

```
# end = boolean
# cancel = boolean
# monsterBall = integer
# ballIndex = integer
# alreadyHaveMons = boolean
# i = integer
# capture = boolean
```

BAB V

DESAIN DEKOMPOSISI ALGORITMIK DAN FUNSIONAL PROGRAM

F00 - Random Number Generator

ALGORITMA PROGRAM UTAMA

```
input(min_val, max_val) { input nilai minimum dan maksimum }
LCG_Init(P, min_val, max_val)
{ Contoh penggunaan }
level <- rngLevel(P)
musuh <- rngEnemy(P, ["orc", "goblin", "troll"])
oc <- rngOC(P)
capture <- rngCapture(P, level)
player_attack <- playerAtk(P, 50)
enemy_attack <- enemyAtk(P, 40)
```

```
procedure LCG_Init(output P: LCG, input min_val: integer, max_val: integer)
{ I.S. sembarang; F.S. P terdefinisi dengan m, a, c, seed, min_val, dan max_val }
```

KAMUS LOKAL

waktu_sekarang: integer

ALGORITMA

```
P.m <- 2^32
P.a <- 82
P.c <- 100
waktu_sekarang <- current_time() { asumsi ada fungsi yang mengembalikan waktu saat
ini dalam detik }
P.seed <- waktu_sekarang
P.min_val <- min_val
P.max_val <- max_val
```

```
procedure set_seed(input/output P: LCG, input seed: integer)
```

```
{ I.S. P terdefinisi; F.S. seed pada P diatur }
```

ALGORITMA

```
P.seed <- seed
```

```
function generate(input/output P: LCG) -> integer
```

```
{ I.S. P terdefinisi; F.S. menghasilkan angka acak antara min_val dan max_val }
```

KAMUS LOKAL

scaled_number: real

ALGORITMA

```
P.seed <- (P.a * P.seed + P.c) mod P.m
scaled_number <- P.min_val + (P.seed % (P.max_val - P.min_val + 1))
return scaled_number
```

```
function rngLevel(input P: LCG) -> integer
```

```
{ I.S. P terdefinisi; F.S. menghasilkan angka acak untuk level }
```

KAMUS LOKAL

random_integer: integer

ALGORITMA

```
LCG_Init(P, 1, 5)
set_seed(P, next_seed())
random_integer <- generate(P)
return random_integer
```

function rngEnemy(input P: LCG, input mons: array of string) -> integer
{ I.S. P terdefinisi; F.S. menghasilkan angka acak untuk musuh berdasarkan panjang array mons }

KAMUS LOKAL

random_integer: integer

panjang_mons: integer

ALGORITMA

```
panjang_mons <- length(mons)
LCG_Init(P, 1, panjang_mons - 1)
set_seed(P, next_seed())
random_integer <- generate(P)
return random_integer
```

function rngOC(input P: LCG) -> integer
{ I.S. P terdefinisi; F.S. menghasilkan angka acak untuk OC }

KAMUS LOKAL

random_integer: integer

ALGORITMA

```
LCG_Init(P, 5, 30)
set_seed(P, next_seed())
random_integer <- generate(P)
return random_integer
```

function rngCapture(input P: LCG, input enemy_level: integer) -> boolean
{ I.S. P terdefinisi; F.S. menghasilkan nilai boolean untuk menentukan keberhasilan menangkap musuh }

KAMUS LOKAL

random_integer: integer

limit: integer

ALGORITMA

```
if enemy_level = 1 then
    limit <- 75
else if enemy_level = 2 then
    limit <- 50
else if enemy_level = 3 then
    limit <- 25
else if enemy_level = 4 then
    limit <- 10
else if enemy_level = 5 then
    limit <- 5

LCG_Init(P, 1, 100)
set_seed(P, next_seed())
random_integer <- generate(P)

if random_integer <= limit then
    return true
else
    return false
```

function playerAtk(input P: LCG, input atk_power: integer) -> (integer, integer)
{ I.S. P terdefinisi; F.S. menghasilkan angka acak dan persentase untuk serangan pemain }

KAMUS LOKAL

```

    batas_bawah: real
    batas_atas: real
    random_integer: integer
    percentage: real
ALGORITMA
    batas_bawah <- atk_power * 70 / 100
    batas_atas <- atk_power * 130 / 100

    LCG_Init(P, batas_bawah, batas_atas)
    set_seed(P, next_seed())
    random_integer <- generate(P)

    percentage <- random_integer / atk_power
    if percentage < atk_power / 100 then
        percentage <- -((1 - percentage) * 100)
    else if percentage > atk_power / 100 then
        percentage <- (percentage - 1) * 100
    else
        percentage <- 0
    return (random_integer, int(percentage))

function enemyAtk(input P: LCG, input enemy_atk_power: integer) -> (integer, integer)
{ I.S. P terdefinisi; F.S. menghasilkan angka acak dan persentase untuk serangan musuh
}

```

KAMUS LOKAL

```

    batas_bawah: real
    batas_atas: real
    random_integer: integer
    percentage: real
ALGORITMA
    batas_bawah <- enemy_atk_power * 70 / 100
    batas_atas <- enemy_atk_power * 130 / 100

    LCG_Init(P, batas_bawah, batas_atas)
    set_seed(P, next_seed())
    random_integer <- generate(P)

    percentage <- random_integer / enemy_atk_power
    if percentage < enemy_atk_power / 100 then
        percentage <- -((1 - percentage) * 100)
    else if percentage > enemy_atk_power / 100 then
        percentage <- (percentage - 1) * 100
    else
        percentage <- 0
    return (random_integer, int(percentage))

```

F01 - Register

```

function isUsernameValid(input username: string) -> boolean
{ Memeriksa apakah username hanya mengandung karakter valid (alfabet, angka,
underscore, dan strip) }

function isUsernameExist(input username: string, input userpas: array of User) ->
boolean
{ Memeriksa apakah username sudah ada dalam database }

```

```
procedure REGISTER(input/output userpas: array of User, input/output mInv: array of MonsterInventory, input is_integer: function)
{ Menambahkan pengguna baru ke dalam database dan memberikan monster awal }
```

ALGORITMA PROGRAM UTAMA

```
REGISTER(userpas, mInv, is_integer)
```

```
function isUsernameValid(input username: string) -> boolean
```

```
{ Memeriksa apakah username hanya mengandung karakter valid (alfabet, angka, underscore, dan strip) }
```

KAMUS LOKAL

```
valid: boolean
```

```
tempUsername: array of char
```

ALGORITMA

```
valid <- true
```

```
tempUsername <- [char for char in username]
```

```
I traversal tempUsername
```

```
    if (ord(i) >= 0 and ord(i) <= 47 and ord(i) != 45) or (ord(i) > 57 and ord(i) < 65) or (ord(i) > 90 and ord(i) != 95 and ord(i) < 97) or ord(i) > 122 then
```

```
        valid <- false
```

```
        break
```

```
    end if
```

```
end for
```

```
return valid
```

```
function isUsernameExist(input username: string, input userpas: array of User) -> boolean
```

```
{ Memeriksa apakah username sudah ada dalam database }
```

KAMUS LOKAL

```
valid: boolean
```

```
tempList: array of string
```

```
i, j: integer
```

ALGORITMA

```
valid <- true
```

```
tempList <- []
```

```
i traversal [len(userpas)]
```

```
    j traversal [len(userpas[i])]
```

```
        if j = 1 then
```

```
            tempList.append(userpas[i][j])
```

```
        end if
```

```
    end for
```

```
end for
```

```
for i in tempList do
```

```
    if username = i then
```

```
        valid <- false
```

```
        break
```

```
    end if
```

```
end for
```

```
return valid
```

```
procedure REGISTER(input/output userpas: array of User, input/output mInv: array of MonsterInventory, input is_integer: function)
```

```
{ Menambahkan pengguna baru ke dalam database dan memberikan monster awal }
```

KAMUS LOKAL

```

role: string
oc: integer
monsterAwal: array of string
validUsername: boolean
notExistUsername: boolean
cnt: integer
username: string
password: string
angkaMonster: integer
monster: string
tempUserpasList: array of any
tempMInv: array of integer

```

ALGORITMA

```

role <- 'agent'
oc <- 0
monsterAwal <- ["Pikachow", "Bulbu", "Zeze", "Zuko", "Chacha"]
validUsername <- true
notExistUsername <- true

cnt <- len(userpas)

output("Masukkan username: ")
(username)
output("Masukkan password: ")
(password)
output("")

validUsername <- isUsernameValid(username)
while validUsername = false do
    output("Username hanya boleh berisi alfabet, angka, underscore, dan strip")
    output("Masukkan username: ")
    (username)
    output("Masukkan password: ")
    (password)
    output("")
    validUsername <- isUsernameValid(username)

notExistUsername <- isUsernameExist(username, userpas)
while notExistUsername = false do
    output("Username ", username, " sudah terpakai, silahkan gunakan username
lain!")
    output("Masukkan username: ")
    (username)
    output("Masukkan password: ")
    (password)
    output("")
    notExistUsername <- isUsernameExist(username, userpas)

tempUserpasList <- [cnt, username, password, role, oc]
userpas.append(tempUserpasList)

output("Silahkan pilih salah satu monster sebagai monster awalmu.")
output("1. Pikachow")
output("2. Bulbu")
output("3. Zeze")

```

```

output("4. Zuko")
output("5. Chacha")
output("")
while true do
  output("Monster pilihanmu: ")
  input(angkaMonster)
  if is_integer(angkaMonster) then
    angkaMonster <- int(angkaMonster)
    if angkaMonster > 5 then
      output("Masukkan angka yang valid!!")
    else
      monster <- monsterAwal[angkaMonster - 1]

      output("Selamat datang Agent ", username, ". Mari kita mengalahkan Dr.
Asep Spakbor dengan ", monster, "!")

      tempMInv <- [int(cnt), int(angkaMonster), 1]
      mInv.append(tempMInv)
      cnt <- cnt + 1
      break

  else
    output("Masukkan angka yang valid!!")

```

F02 - Login

```

from src.csv import *

```

```

function isPasswordCorrect(loginUsername, loginPassword, userList, passList) -> boolean
  Bool <- False
  i traversal [len(userList)]
    if loginUsername = userList[i] and loginPassword != passList[i] then
      Bool <- True
      stop
  return Bool

function isLoginValid(loginUsername, loginPassword, userList, passList) -> boolean
  Bool <- False
  i traversal [len(userList)]
    if loginUsername = userList[i] and loginPassword = passList[i] then
      Bool <- True
      stop
  return Bool

function isUsernameExist(wrongPassword, loginBool) -> boolean
  Bool <- False
  if wrongPassword = False and loginBool = False then
    Bool <- True
  return Bool

function LOGIN(userpas) -> tuple of (boolean, list)
  currentUser <- empty list
  loginBool <- False
  wrongUsername <- False

```



```

wrongPassword <- False

traversal [len(userpas)]
  j traversal [len(userpas[i])]
    if j = 1 then
      append userpas[i][j] to userList
    else if j = 2 then
      append userpas[i][j] to passList

wrongPassword <- isPasswordCorrect(loginUsername, loginPassword, userList,
passList)
loginBool <- isLoginValid(loginUsername, loginPassword, userList, passList)
wrongUsername <- isUsernameExist(wrongPassword, loginBool)

if loginBool then
  i traversal [len(userList)]
    j traversal [len(userList)]
      if userList[i] = loginUsername then
        append integer value of userpas[i][0] to currentUser
        append userpas[i][1] to currentUser # username
        append userpas[i][2] to currentUser # password
        append userpas[i][3] to currentUser # role
        append integer value of userpas[i][4] to currentUser
        break

      if currentUser[3] = "agent" then
        output(f"Selamat datang, Agent {loginUsername}!")
        output("Masukkan command 'help' untuk daftar command yang dapat kamu
panggil.")
      else if currentUser[3] = "admin" then
        output("Selamat datang, admin")
        output("Masukkan command 'help' untuk daftar command yang dapat kamu
panggil.")
      else if wrongUsername then
        output("Username tidak terdaftar!")
        loginBool <- False
      else if wrongPassword then
        output("Password salah!")
        loginBool <- False

output()
return loginBool, currentUser

```

F03 - Logout

```
from src.login import *
```

```

function LOGOUT(currentUser, login) -> tuple of (boolean, list)
  if login then
    currentUser.clear()
    login <- False

```

```

        output("Anda berhasil logout!!")
    else
        output("Logout gagal!")
        output("Anda belum login, silahkan login terlebih dahulu sebelum melakukan
logout")
    output()
    return login, currentUser

```

F04 - Menu & Help

```

from src.login import *

function HELP(currentUser)
    output()
    output("===== HELP =====")
    if currentUser = [] then
        output("Kamu belum login sebagai role apapun. Silahkan login terlebih
dahulu.")
        output("    1. Login: Masuk ke dalam akun yang sudah terdaftar")
        output("    2. Register: Membuat akun baru")
    else if currentUser[3] = 'agent' then
        output(f"Halo Agent {currentUser[1]}. Kamu memanggil command HELP. Kamu
memilih jalan yang benar, semoga kamu tidak sesat kemudian.")
        output("Berikut adalah hal-hal yang dapat kamu lakukan sekarang:")
        output("    1. Logout: Keluar dari akun yang sedang digunakan")
        output("    2. Monster: Melihat owca-dex yang dimiliki oleh Agent")
        output("    3. Tambah lagi ntar")
    else if currentUser[3] = 'admin' then
        output("Selamat datang, Admin. Berikut adalah hal-hal yang dapat kamu
lakukan:")
        output("    1. Logout: Keluar dari akun yang sedang digunakan")
        output("    2. Shop: Melakukan manajemen pada SHOP sebagai tempat jual beli
peralatan Agent")
        output("    3. Tambah lagi ntar")
    output("Footnote :")
    output("    1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang
terdaftar")
    output("    2. Jangan lupa untuk memasukkan input yang valid")
    output()

```

F05 - Monster

```

function monster(atk_power: integer, def_power: integer, hp: integer, level: integer)
-> tuple of (integer, integer, integer)
    if 2 <= level <= 5 then
        add <- 1 + ((level - 1) * 0.1)
        atk_power <- atk_power * add
        def_power <- def_power * add
        hp <- hp * add
    if def_power > 50 then
        def_power <- 50

    return int(atk_power), int(def_power), int(hp)

```

F06 - Potion

```

function potion(type_potion, atk_power, def_power, hp) -> tuple of (real, real, real)

```



```

        output(f"Def Power : {mTemp[i][3]}")
        output(f"HP      : {mTemp[i][4]}")
        output(f"Level   : {mTemp[i][5]}")
        output()
        exit loop
    else if len(mTemp) < int(pilihanInv) < (len(mTemp)+len(iTemp)) then
        i traversal [length of iTemp]
        if int(pilihanInv) = iTemp[i][0] then
            output("Potion")
            output(f"Type      : {iTemp[i][1]}")
            output(f"Quantity  : {iTemp[i][2]}")
            output()
            exit loop
        else if int(pilihanInv) = (len(mTemp)+len(iTemp)) then
            i traversal [length of iTemp]
            if int(pilihanInv) = iTemp[i][0] then
                output("Monster Ball")
                output(f"Quantity  : {iTemp[i][2]}")
                output()
                exit loop
            else
                output()
        until false

```

F08 - Battle

```

function chooseMons(userMons) -> tuple
    while True do
        choice <- input("Pilih monster untuk bertarung: ")
        choice <- integer(choice)
        output()
        monsNum <- length of userMons
        if choice > monsNum then
            output("Pilihan nomor tidak tersedia!")
            output()
        else
            i traversal [length of userMons]
            if choice = i+1 then
                currentMons <- [userMons[i][0], userMons[i][1], userMons[i][2],
userMons[i][3], userMons[i][4]]
                exit loop
            exit loop
        return currentMons

```

```

function userPot(iInv, currentUser) -> tuple
    strength <- 0
    resilience <- 0
    healing <- 0
    strengthIndex <- 1
    resilienceIndex <- 1
    healingIndex <- 1
    i traversal [len(iInv)]
        if int(iInv[i][0]) = int(currentUser[0]) then
            if iInv[i][1] = "strength" then
                strength <- int(iInv[i][2])

```

```

        strengthIndex <- i
        if iInv[i][1] = "resilience" then
            resilience <- int(iInv[i][2])
            resilienceIndex <- i
        if iInv[i][1] = "healing" then
            healing <- int(iInv[i][2])
            healingIndex <- i
    currentPot <- [strength, resilience, healing]
    return currentPot, strengthIndex, resilienceIndex, healingIndex

function userBall(iInv, currentUser) -> tuple
    monsterBall <- 0
    i traversal [len(iInv)]
        if int(iInv[i][0]) = int(currentUser[0]) then
            if iInv[i][1] = "monster_ball" then
                monsterBall <- int(iInv[i][2])
                ballIndex <- i
    return monsterBall, ballIndex

function dmgCalc(tempAtk, enemy_def_power, enemy_hp, percentage) -> tuple
    defcalc <- enemy_def_power/100 * tempAtk
    damagecalc <- tempAtk - defcalc
    return damagecalc, tempAtk, defcalc

function monsterCapture(userBall, userMons, iInv, rngCapture, mInv, enemy_type,
currentUser, chosenEnemy, enemy_level, enemy_atk_power, enemy_def_power, enemy_hp) ->
tuple
    end <- False
    cancel <- False
    monsterBall, ballIndex <- userBall(iInv, currentUser)
    alreadyHaveMons <- False
    i traversal [len(userMons)]
        if enemy_type = userMons[i][0] then
            alreadyHaveMons <- True
    if alreadyHaveMons then
        output(f"Anda sudah memiliki monster {enemy_type} dalam inventory!")
        output()
        cancel <- True
    else if monsterBall > 0 then
        monsterBall <- monsterBall - 1
        iInv[ballIndex][2] <- int(iInv[ballIndex][2])
        iInv[ballIndex][2] <- iInv[ballIndex][2] - 1
        output("Swoosshhhh, Anda mengeluarkan Monster Ball !!!")
        capture <- rngCapture(LCG, enemy_level)
        if capture then
            output(f"Selamat, Anda berhasil mendapatkan monster {enemy_type} !!!")
            mInv.append([currentUser[0], chosenEnemy[0], enemy_level])
            end <- True
            output(f""
Name      : {enemy_type}
ATK Power : {enemy_atk_power}
DEF Power : {enemy_def_power}
HP        : {enemy_hp}
Level     : {enemy_level}""")
            output()

```

```

        output(f"Sisa Monster Ball Anda: {monsterBall}")
        output()
    else
        output(f"Yahhh, Anda belum berhasil mendapatkan monster {enemy_type} !!!")
        output()
        output(f"Sisa Monster Ball Anda: {monsterBall}")
        output()
    else
        output("Anda tidak memiliki Monster Ball dalam inventory!")
    return end, cancel
}

function yourTurn(mons_type, atk_power, def_power, hp, strengthBool, resilienceBool,
healingBool, turnCnt, currentPot, strengthIndex, resilienceIndex, healingIndex,
enemy_type, enemy_atk_power, enemy_def_power, enemy_hp, enemy_level, dmgCalc,
chosenEnemy, userBall, userMons, ballPresence, damage_given, iInv, mInv, currentUser)
-> tuple
    end <- False
    flee <- False
    cancel <- False
    while True do
        output(f"===== TURN {turnCnt} ({mons_type}) =====")
        if ballPresence then
            output("1. Attack")
            output("2. Use Potion")
            output("3. Use Monster Ball")
            output("4. Flee")
        else
            output("1. Attack")
            output("2. Use Potion")
            output("3. Quit")
        command <- input("Pilih perintah: ")
        command <- integer(command)
        if command = 4 then
            output()
            if ballPresence then
                output("Anda berhasil kabur dari BATTLE!")
                flee <- True # keluar dari battle
                exit loop
            else
                output("Perintah tidak valid. Silahkan pilih perintah diatas!")
            else if command = 3 then
                output()
                if ballPresence then
                    end, cancel <- monsterCapture(userBall, userMons, iInv, rngCapture,
mInv, enemy_type, currentUser, chosenEnemy, enemy_level, enemy_atk_power,
enemy_def_power, enemy_hp)
                    if cancel or end then
                        exit loop
                    else
                        output("Anda berhasil kabur dari BATTLE!")
                        flee <- True # keluar dari battle
                        exit loop
                    else if command = 2 then
                        if currentPot = [0, 0, 0] then
                            output("Anda tidak memiliki potion dalam inventory!")

```

```

else
  while True do
    type_potion <- ""
    output()
    output("===== POTION LIST =====")
    output(f"1. Strength Potion (Qty: {currentPot[0]}) - Increases ATK
Power")
    output(f"2. Resilience Potion (Qty: {currentPot[1]}) - Increases
DEF Power")
    output(f"3. Healing Potion (Qty: {currentPot[2]}) - Restores
Health")
    output("4. Cancel")
    output()
    potCommand <- input("Pilih potion atau cancel: ")
    if int(potCommand) = 1 then
      if strengthBool = True then
        output(f"Kamu mencoba memberikan ramuan ini kepada
{mons_type}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak
bermanfaat lagi.")
      else if currentPot[0] > 0 then
        type_potion <- "strength"
        output(f"Setelah meminum ramuan ini, aura kekuatan
terlihat mengelilingi {mons_type} dan gerakannya menjadi lebih cepat dan mematikan.")
        strengthBool <- True
        currentPot <- [currentPot[0]-1, currentPot[1],
currentPot[2]]
        iInv[strengthIndex][2] <- int(iInv[strengthIndex][2])
        iInv[strengthIndex][2] <- iInv[strengthIndex][2] - 1
        exit loop
      else
        output("Wah, kamu sedang tidak memiliki ramuan ini,
silahkan pilih ramuan lain!")
      else if int(potCommand) = 2 then
        if resilienceBool = True then
          output(f"Kamu mencoba memberikan ramuan ini kepada
{mons_type}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak
bermanfaat lagi.")
        else if currentPot[1] > 0 then
          type_potion <- "resilience"
          output(f"Setelah meminum ramuan ini, muncul sebuah energi
pelindung di sekitar {mons_type} yang membuatnya terlihat semakin tangguh dan sulit
dilukai.")
          resilienceBool <- True
          currentPot <- [currentPot[0], currentPot[1]-1,
currentPot[2]]
          iInv[resilienceIndex][2] <- int(iInv[resilienceIndex][2])
          iInv[resilienceIndex][2] <- iInv[resilienceIndex][2] - 1
          exit loop
        else
          output("Wah, kamu sedang tidak memiliki ramuan ini,
silahkan pilih ramuan lain!")
        else if int(potCommand) = 3 then
          if healingBool = True then

```

```

        output(f"Kamu mencoba memberikan ramuan ini kepada
{mons_type}, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak
bermanfaat lagi.")
        else if currentPot[2] > 0 then
            type_potion <- "healing"
            output(f"Setelah meminum ramuan ini, luka-luka yang ada di
dalam tubuh {mons_type} sembuh dengan cepat. Dalam sekejap, {mons_type} terlihat
kembali prima dan siap melanjutkan pertempuran.")
            healingBool <- True
            currentPot <- [currentPot[0], currentPot[1],
currentPot[2]-1]
            iInv[healingIndex][2] <- int(iInv[healingIndex][2])
            iInv[healingIndex][2] <- iInv[healingIndex][2] - 1
            exit loop
        else
            output("Wah, kamu sedang tidak memiliki ramuan ini,
silahkan pilih ramuan lain!")
            else if int(potCommand) = 4 then
                cancel <- True
                exit loop
            else
                output("Perintah tidak valid. Silahkan pilih perintah
diatas!")
                if type_potion != "" then
                    atk_power, def_power, hp <- potion(type_potion, atk_power,
def_power, hp)
                    output()
                    exit loop
                else if command = 1 then
                    output()
                    output(f"SCHWINKKK, {mons_type} menyerang {enemy_type} !!!")
                    output(enemy_def_power)
                    tempAtk, percentage <- playerAtk(LCG, atk_power)
                    damagecalc, tempAtk, defcalc <- dmgCalc(tempAtk, enemy_def_power,
enemy_hp, percentage)
                    enemy_hp <- int(enemy_hp - damagecalc)
                    damage_given <- damage_given + damagecalc
                    output(f""
Name      : {enemy_type}
ATK Power : {enemy_atk_power}
DEF Power : {enemy_def_power}
HP        : {enemy_hp}
Level     : {enemy_level}""")
                    output()
                    output(f"Penjelasan : ATT: {tempAtk} ({percentage}%), Reduced by:
{defcalc} ({enemy_def_power}%), ATT Results: {int(damagecalc)}")
                    output()
                    exit loop
                else
                    output("Masukkan perintah yang valid")
                    if enemy_hp <= 0 then
                        win <- True
                        enemy_hp <- 0
                    else
                        win <- False

```



```

    return atk_power, def_power, hp, enemy_hp, win, currentPot, strengthBool,
    resilienceBool, healingBool, cancel, flee, end, iInv, damage_given

function enemyTurn(mons_type, atk_power, def_power, hp, level, enemy_type,
enemy_atk_power, enemy_def_power, enemy_hp, enemy_level, turnCnt, dmgCalc,
damage_received) -> tuple
    output(f"===== TURN {turnCnt} ({enemy_type}) =====")
    output()
    output(f"SCHWINKKK, {enemy_type} menyerang {mons_type} !!!")
    tempAtk, percentage <- enemyAtk(LCG, enemy_atk_power)
    damagecalc, tempAtk, defcalc <- dmgCalc(tempAtk, def_power, hp, percentage)
    hp <- int(hp - damagecalc)
    damage_received <- damage_received + damagecalc
    if hp > 0 then
        output(f""
Name      : {mons_type}
ATK Power : {int(atk_power)}
DEF Power : {int(def_power)}
HP        : {int(hp)}
Level     : {level}""")
    else
        output(f""
Name      : {mons_type}
ATK Power : {int(atk_power)}
DEF Power : {int(def_power)}
HP        : 0
Level     : {level}""")
        output()
        output(f"Penjelasan : ATT: {tempAtk} ({percentage}%), Reduced by: {defcalc}
({def_power}%), ATT Results: {int(damagecalc)}")
        output()
        if hp <= 0 then
            lose <- True
        else
            lose <- False
        return lose, hp, damage_received

function BATTLE(mons, mInv, iInv, rngEnemy, currentUser, rngLevel, userpas) -> tuple
    damage_given <- 0
    damage_received <- 0
    ballPresence <- True
    strengthBool <- False
    resilienceBool <- False
    healingBool <- False
    rngEnemy <- rngEnemy(LCG, mons)
    rngLevel <- rngLevel(LCG)
    chosenEnemy <- mons[rngEnemy]
    enemy_type <- chosenEnemy[1]
    enemy_atk_power <- (int(chosenEnemy[2]))
    enemy_def_power <- (int(chosenEnemy[3]))
    enemy_hp <- (int(chosenEnemy[4]))
    enemy_level <- rngLevel
    enemy_atk_power, enemy_def_power, enemy_hp <- monster(enemy_atk_power,
enemy_def_power, enemy_hp, enemy_level)
    enemy <- [enemy_type, enemy_atk_power, enemy_def_power, enemy_hp, enemy_level]

```

```
{-
  The pseudocode for the output statements and the remaining part of the BATTLE
  function.
  It continues from where we left off.
-}
```

```
output("""
(`'-''-/").____..--'""'\_.._
`6_6 )   `-. (   )_.`-.____`)
(_Y_.)'   ._. )   `_.`_.`-.`-.-'
  _..`--'_.-_-/_ /--'_.-'
  ((((.-' ' ((((.-' ((((.-' """)
```

```
output(f"RAWRRR, Monster {enemy[0]} telah muncul !!!")
output(f"""
Name      : {enemy_type}
ATK Power : {enemy_atk_power}
DEF Power : {enemy_def_power}
HP        : {enemy_hp}
Level     : {enemy_level}""")
```

```
output("===== MONSTER LIST =====")
tempMons <- []
userMons <- []
for i <- 0 to len(mInv) do
  for j <- 0 to len(mons) do
    if str(mInv[i][0]) == str(currentUser[0]) and str(mInv[i][1]) ==
str(mons[j][0]) then
      tempMons.append(mons[j][1])      // name
      tempMons.append(int(mons[j][2])) // atk
      tempMons.append(int(mons[j][3])) // def
      tempMons.append(int(mons[j][4])) // hp
      tempMons.append(int(mInv[i][2])) // level
      userMons.append(tempMons)
      tempMons <- []
```

```
for i <- 0 to len(userMons) do
  output(f"{i+1}. {userMons[i][0]}")
```

```
currentMons <- chooseMons(userMons)
mons_type <- currentMons[0]
atk_power <- currentMons[1]
def_power <- currentMons[2]
hp <- currentMons[3]
level <- currentMons[4]
atk_power, def_power, hp <- monster(atk_power, def_power, hp, level)
```

```
output("""
  .""".
  .-./ _=_ \.-.
  { (, (oYo), ) }
  {{ | " | } }
  { { \ (---) / } }
  {{ } '-=-' { } }
  { { } ._:_. { } }
```

```

    {{ { } } :- { } }
    { _ { } } `==` { _ }
    ((((\)      (/)))) """)

output(
    f"RAWRRR, Agent {currentUser[1]} mengeluarkan monster {mons_type} !!!")
output(f""
Name      : {mons_type}
ATK Power : {atk_power}
DEF Power : {def_power}
HP        : {hp}
Level     : {level}""")

turnCnt <- 1
win <- False
lose <- False
currentPot, strengthIndex, resilienceIndex, healingIndex <- userPot(
    iInv, currentUser)
while not win and not lose do
    atk_power, def_power, hp, enemy_hp, win, currentPot, strengthBool, resilienceBool,
    healingBool, cancel, flee, end, iInv, damage_given <- yourTurn(
        mons_type, atk_power, def_power, hp, strengthBool, resilienceBool,
        healingBool, turnCnt, currentPot, strengthIndex, resilienceIndex, healingIndex,
        enemy_type, enemy_atk_power, enemy_def_power, enemy_hp, enemy_level, dmgCalc,
        chosenEnemy, userBall, userMons, ballPresence, damage_given, iInv, mInv, currentUser)

    if not flee then
        if not end then
            if not cancel then
                if not win then
                    lose, hp, damage_received <- enemyTurn(mons_type, atk_power,
                                                                enemy_atk_power,
                                                                enemy_def_power, enemy_hp, enemy_level, turnCnt, dmgCalc, damage_received)
                    turnCnt <- turnCnt + 1
                else
                    exit loop
            else
                pass
        else
            exit loop
    else
        exit loop

if win then
    rewardOC <- rngOC(LCG)
    currentUser[4] <- int(currentUser[4])
    (currentUser[4]) <- currentUser[4] + rewardOC
    for i <- 1 to len(userpas) do
        if int(userpas[i][0]) == int(currentUser[0]) then
            userpas[i][4] <- currentUser[4]
        output(f"Selamat, Anda berhasil mengalahkan monster {enemy_type} !!!")
        output(f"Total OC yang diperoleh: {rewardOC}")
        output()
    if lose then

```

```
    output(f"Yahhh, Anda dikalahkan monster {enemy_type}. Jangan menyerah, coba lagi
!!!")
```

```
return userpas, mInv, iInv
```

```
end function
```

F09 - Arena

```
function mInvList( invCount : list, currentUser : list, mInv : list, mons: list, mTemp
: list) -> tuple
```

```
    i traversal [0..len(mInv)]
      j traversal [len(mons)]
        if str(mInv[i][0]) = str(currentUser[0]) then
          if str(mInv[i][1]) = str(mons[j][0]) then
            output(f"{invCount+1}. Monster (Name: {mons[j][1]}, Lvl:
{mInv[j][2]}, HP: {mons[j][4]})")
            invCount <- invCount + 1
            stats <- [invCount, mons[j][1], int(mons[j][2]), int(mons[j][3]),
int(mons[j][4]), int(mInv[i][2])]
            mTemp.append(stats)
          end if
        end if
      end for
    return mTemp
```

```
function ARENA( mons : list, mInv: list, rngEnemy : integer, currentUser : list, iInv:
list) -> tuple
```

```
    ballPresence <- False
    total_reward <- 0
    damage_received <- 0
    damage_given <- 0
    output("Selamat datang di Arena!")
    output("===== MONSTER LIST =====")
    tempMons <- []
    userMons <- []
    stage <- 1
    i traversal [len(mInv)]
      j traversal [len(mons)]
        if str(mInv[i][0]) = str(currentUser[0]) then
          if str(mInv[i][1]) = str(mons[j][0]) then
            tempMons.append(mons[j][1]) {name}
            tempMons.append(int(mons[j][2])) {atk}
            tempMons.append(int(mons[j][3])) {def}
            tempMons.append(int(mons[j][4])) {hp}
            tempMons.append(int(mInv[i][2])) {level}
            userMons.append(tempMons)
            tempMons <- []
          end if
        end if
      end for
    end for
```

```
    i traversal [len(userMons)]
      output(f"{i+1}. {userMons[i][0]}")
```

```
    currentMons <- chooseMons(userMons)
    mons_type <- currentMons[0]
    atk_power <- currentMons[1]
    def_power <- currentMons[2]
    hp <- currentMons[3]
    hp_reset <- hp
    level <- currentMons[4]
```

```

atk_power, def_power, hp <- monster(atk_power, def_power, hp, level)
strengthBool <- False
resilienceBool <- False
healingBool <- False
enemyrng <- rngEnemy(LCG, mons)
chosenEnemy <- mons[enemyrng]

enemy_type <- chosenEnemy[1]
enemy_atk_power <- (int(chosenEnemy[2]))
enemy_def_power <- (int(chosenEnemy[3]))
enemy_hp <- (int(chosenEnemy[4]))
enemy_level <- int(stage)
enemy_atk_power, enemy_def_power, enemy_hp <- monster(
  enemy_atk_power, enemy_def_power, enemy_hp, enemy_level)
enemy <- [enemy_type, enemy_atk_power, enemy_def_power, enemy_hp, enemy_level]

output("""
      .``.
    .-./ _=_ \.-.
    {  (, (oYo), ) }}
    {{ |   "   | }}
    { { \ (---) /  }}
    {{ } '-==-' { }}
    { { }._:_.{ }}
    {{ } -:- { }}
    { _ { } `===` { _ }
    (((\ )      (/))) """)
output(f"RAWRRR, Agent {currentUser[1]} mengeluarkan monster {mons_type} !!!")
output(f"""
Name      : {mons_type}
ATK Power : {atk_power}
DEF Power : {def_power}
HP        : {hp}
Level     : {level} """)

while stage <= 5 do
  stage_akhir <- stage-1
  output(f"===== STAGE {stage} =====")

  output("""
("`-'-'-'/" ) . ____ .--' "'`-_. _
`6_6 )    `-. (    ) .`-. ____ `)
(_Y_) '  . _ )    ` _ ` _ ` _ ` _ `
 _..`--' _.. _ /  /--' _.'
((((.-' '  ((((. '  (((.-' """))

  output(f"RAWRRR, Monster {enemy[0]} telah muncul !!!")
  output(f"""
Name      : {enemy[0]}
ATK Power : {enemy[1]}
DEF Power : {enemy[2]}
HP        : {enemy[3]}
Level     : {enemy[4]} """)
  turnCnt <- 1
  win <- False

```

```

lose <- False
currentPot, strengthIndex, resilienceIndex, healingIndex <- userPot(
  iInv, currentUser)
while (not win) and (not lose) do
  atk_power, def_power, hp, enemy_hp, win, currentPot, strengthBool,
  resilienceBool, healingBool, cancel, flee, end, iInv, damage_given <-
  yourTurn(mons_type, atk_power, def_power, hp, strengthBool,
  resilienceBool, healingBool, turnCnt, currentPot, strengthIndex,
  resilienceIndex, healingIndex, enemy_type, enemy_atk_power,
  enemy_def_power, enemy_hp, enemy_level, dmgCalc, chosenEnemy, userBall,
  userMons, ballPresence, damage_given, iInv, mInv, currentUser)

  if not flee then
    if not cancel then
      if not win then
        lose, hp, damage_received <- enemyTurn(mons_type, atk_power,
        def_power, hp, level, enemy_type,
        enemy_atk_power, enemy_def_power,
        enemy_hp, enemy_level, turnCnt,

if win then
  currentUser[4] += reward[stage-1][1]
  total_reward += reward[stage-1][1]
  output(f"Selamat, Anda berhasil mengalahkan monster {enemy_type} !!!")
  output(f"STAGE CLEARED! Anda akan mendapatkan {reward[stage-1][1]} OC pada
    sesi ini! ")
  output("Memulai stage berikutnya...")
  output()
  stage <- stage + 1
  hp <- hp_reset
  enemyrng <- rngEnemy(LCG, mons)
  chosenEnemy <- mons[enemyrng]
  enemy_type <- chosenEnemy[1]
  enemy_atk_power <- (int(chosenEnemy[2]))
  enemy_def_power <- (int(chosenEnemy[3]))
  enemy_hp <- (int(chosenEnemy[4]))
  enemy_level <- int(stage)
  enemy_atk_power, enemy_def_power, enemy_hp <- monster(
    enemy_atk_power, enemy_def_power, enemy_hp, enemy_level)
  enemy <- [enemy_type, enemy_atk_power,
    enemy_def_power,
enemy_hp, enemy_level]

  if lose then
    output(f"Yahhh, Anda dikalahkan monster {enemy_type}. Jangan menyerah, coba
      lagi !!!")
    output(f"GAME OVER! Sesi latihan berakhir pada stage {stage}!")
    stage_akhir <- stage
    stage <- 7

if stage = 6 then
  output("Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!")
  output("===== STATS =====")
  output(f"")
Total Hadiah      : {total_reward}

```

```

Jumlah Stage      : 5
Damage diberikan : {damage_given}
Damage diterima  : {damage_received}""")
else if stage = 7 then
    output("===== STATS =====")
    output(f""
Total Hadiah      : {total_reward}
Jumlah Stage      : {stage_akhir}
Damage diberikan : {damage_given}
Damage diterima  : {damage_received}""")
else if flee = True then
    output("===== STATS =====")
    output(f""
Total Hadiah      : {total_reward}
Jumlah Stage      : {stage_akhir}
Damage diberikan : {damage_given}
Damage diterima  : {damage_received}""")

```

F10 - Shop & Currency

function SHOP (currentUser: list, mShop: list, iShop: list, mons: list, mInv: list, iInv: list) -> tuple

```

shopCount <- 0
msTemp <- []
isTemp <- []

output("안녕하세요! Selamat Datang di SHOP!!!")
aksi <- ""
while aksi <> "keluar" do
    aksi <- input("Pilih aksi (lihat/beli/keluar): ")

    if aksi = "lihat" then
        jenis <- input("Mau lihat apa? (monster/item): ")
        output()
        if jenis = "monster" then
            mShopList(shopCount, currentUser, mShop, mons, msTemp)
        else if jenis = "item" then
            iShopList(shopCount, currentUser, iShop, isTemp)
        output()

    else if aksi = "beli" then
        output("Jumlah O.W.C.A. Coin-mu sekarang ", currentUser[4])
        output()
        beli <- input("Mau beli apa? (monster/item): ")

        if beli = "monster" then
            id_monster <- input("Masukkan id monster: ")
            if id_monster > len(mShop) - 1 then
                output("Pilihan monster tidak valid.")
                output()
            else

```

```

        if mShop[id_monster][1] = 0 then
            output("Pembelian gagal, stok monster kurang.")
            output()
        else if mShop[id_monster][2] > currentUser[4] then
            output("OC-mu tidak cukup.")
            output()
        else if cekMonster(mShop, mInv, id_monster, currentUser) then
            output("Monster ", mons[int(mShop[id_monster][0])][1], " sudah
ada dalam inventory-mu! Pembelian dibatalkan.")
        else
            output("Berhasil membeli item: ",
mons[int(mShop[id_monster][0])][1], ". Item sudah masuk ke inventory-mu!")
            output()
            currentUser[4] <- currentUser[4] - mShop[id_monster][2]
            mShop[id_monster][1] <- mShop[id_monster][1] - 1
            mons_to_inv <- [currentUser[0],
mons[int(mShop[id_monster][0])][0], 1]
            mInv.append(mons_to_inv)

    else if beli = "item" then
        id_item <- input("Masukkan id item: ")
        jumlah_item <- input("Masukkan jumlah item: ")
        if id_item > len(iShop) - 1 then
            output("Pilihan item tidak valid.")
            output()
        else
            itemCek <- cekItem(iShop, iInv, id_item, currentUser)
            if iShop[id_item][1] < jumlah_item then
                output("Pembelian gagal, stok item kurang.")
                output()
            else if iShop[id_item][2] * jumlah_item > currentUser[4] then
                output("OC-mu tidak cukup.")
                output()
            else if itemCek then
                output("Berhasil membeli item: ", jumlah_item, " ",
iShop[id_item][0], ". Item sudah masuk ke inventory-mu!")
                output()
                currentUser[4] <- currentUser[4] - iShop[id_item][2] *
jumlah_item
                iShop[id_item][1] <- iShop[id_item][1] - jumlah_item
                item_to_inv <- [currentUser[0], iShop[id_item][0],
jumlah_item]
                iInv.append(item_to_inv)
                iTemp.clear()
            else
                output("Berhasil membeli item: ", jumlah_item, " ",
iShop[id_item][0], ". Item sudah masuk ke inventory-mu!")
                output()
                currentUser[4] <- currentUser[4] - iShop[id_item][2] *
jumlah_item
                iShop[id_item][1] <- iShop[id_item][1] - jumlah_item
                for traversal in [len(iInv)]
                    if currentUser[0] = iInv[i][0] then
                        if iShop[id_item][0] = iInv[i][1] then
                            iInv[i][2] <- iInv[i][2] + jumlah_item

```



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

F11 - Laboratory

```
function playerMons( mInv: list, mons: list, currentUser: list) -> tuple
```

```
    monsters <- []
    i traversal [1 .. length of mInv]
        j traversal [1 .. length of mons]
            if int(mInv[i][0]) = int(currentUser[0]) then
                if str(mInv[i][1]) = str(mons[j][0]) then
                    monsters.append([mons[j][1], int(mInv[i][2])])
    return monsters
```

```
upgrade_prices <- {
    2: 300,
    3: 500,
    4: 800,
    5: 1000
}
```

```
function display_monster(monster: list) -> tuple
```

```
    output("\\n===== MONSTER LIST =====")
    Idx traversal, monster in enumerate[1 .. length of monsters]
        output(idx + ". " + monster[0] + " (Level: " + monster[1] + ")")
```

```
function upgrade_monster(currency: integer, monsters: list, currentUser: list, mons:
list, mInv: list, userpas: list) -> tuple
```

```
    display_monster(monsters)
    output(">>> Pilih monster: ")
    monster_choice <- input()
    if is_integer(monster_choice) then
        monster_choice <- int(monster_choice) - 1

        if (monster_choice < 0) or (monster_choice >= length of monsters) then
            output("Pilihan monster tidak valid.")

        else
            monster <- monsters[monster_choice]
            if monster[1] >= 5 then
                output("Maaf, monster sudah mencapai level maksimum.")

            else
                upgrade_price <- upgrade_prices.get(monster[1] + 1)
                output(monster[0] + " akan di-upgrade ke level " + (monster[1] + 1) +
".")
                output("Harga untuk melakukan upgrade " + monster[0] + " adalah " +
upgrade_price + " OC.")

                output(">>> Lanjutkan upgrade (Y/N): ")
                confirm <- input().upper()
                if confirm = "Y" then
                    if currency >= upgrade_price then
                        currency <- currency - upgrade_price
                        currentUser[4] <- int(currentUser[4])
```

```

        currentUser[4] <- currency
        i traversal [1 .. length of userpas]
            if userpas[i][0] = currentUser[0] then
                userpas[i][4] <- int(userpas[i][4])
                userpas[i][4] <- currency

        monster[1] <- monster[1] + 1
        breakout <- False
        i traversal [1 .. length of mInv]
            j traversal [1 .. length of mons]
                if (str(mInv[i][0]) = str(currentUser[0])) then
                    if (mons[j][1] = monster[0]) then
                        if str(mons[j][0]) = str(mInv[i][1]) then
                            mInv[i][2] <- int(mInv[i][2])
                            mInv[i][2] <- mInv[i][2] + 1
                            breakout <- True
                            break
                        if breakout then
                            break
                    output("Selamat, " + monster[0] + " berhasil di-upgrade ke
level " + monster[1] + "!!")
                else
                    output("Maaf, Anda tidak memiliki cukup OC untuk melakukan
upgrade.")
            else
                output("Upgrade dibatalkan.")
        else
            output("Masukkan angka yang valid!!")
        return currency, userpas, mInv

function LABORATORY(userpas: list, mInv: list, mons: list, currentUser: list) -> tuple
    output("Selamat datang di Lab Dokter Asep !!!")
    monsters <- playerMons(mInv, mons, currentUser)
    display_monster(monsters)

    currency <- int(currentUser[4])
    output("\\n===== UPGRADE PRICE =====")
    for level, price in upgrade_prices.items() do
        output((level - 1) + ". Level " + (level - 1) + " -> Level " + level + ": " +
price + " OC")
    output("=====")

    while True do
        output("\\n Apa yang ingin Anda lakukan?")
        output("1. Upgrade monster")
        output("2. Exit")

        output(">>> Masukkan pilihan (1/2): ")
        choice <- input()

        if choice = "1" then
            currency, userpas, mInv <- upgrade_monster(currency, monsters,
currentUser, mons, mInv, userpas, is_integer)
        else if choice = "2" then
            output("Terima kasih telah menggunakan Lab Dokter Asep!")

```

```

        break
    else
        output("Pilihan tidak valid. Silakan masukkan angka 1 atau 2.")

```

F12 - Shop Management

function mShopList(shopCount : integer, mShop : list, mons : list, msTemp : list) -> tuple

```

    output("ID | Type          | ATK Power | DEF Power | HP    | Stok | Harga")
    i traversal [1 .. length of mShop]
        output(shopCount + 1)
        output(mons[int(mShop[i][0])][1])
        output(mons[int(mShop[i][0])][2])
        output(mons[int(mShop[i][0])][3])
        output(mons[int(mShop[i][0])][4])
        output(mShop[i][1])
        output(mShop[i][2])
        shopCount <- shopCount + 1
        stats <- [shopCount, mons[int(mShop[i][0])][1],
int(mons[int(mShop[i][0])][2]), int(mons[int(mShop[i][0])][3]),
int(mons[int(mShop[i][0])][4]), int(mShop[i][2])]
        msTemp.append(stats)
    shopCount <- 0
    return msTemp, shopCount

```

function notmShopList(shopCount : integer, mShop : list, mons : list, nmsTemp : list)

```

-> tuple
    if length of mons = length of mShop then
        output("Semua monster sudah ada di shop")
    else
        id_mons_to_shop <- 0
        processed_ids <- set()
        shopCount <- length of mShop - 1
        output("ID | Type          | ATK Power | DEF Power | HP    |")
        i traversal [1 .. length of mons]
            is_in_shop <- False
            j traversal [1 .. length of mShop]
                if mons[i][0] = mShop[j][0] then
                    is_in_shop <- True
                    break
            if not is_in_shop and mons[i][0] not in processed_ids then
                output(shopCount + 1)
                output(mons[i][1])
                output(mons[i][2])
                output(mons[i][3])
                output(mons[i][4])
                shopCount <- shopCount + 1
                stats <- [shopCount, mons[i][1], int(mons[i][2]), int(mons[i][3]),
int(mons[i][4])]
                id_mons_to_shop <- mons[i][0]
                nmsTemp.append(stats)
                processed_ids.add(mons[i][0])
        shopCount <- 0
        return nmsTemp, shopCount, id_mons_to_shop

```

function iShopList(shopCount : integer, iShop : list, isTemp : list) -> tuple

```

output("ID | Type          | Stok | Harga")
i traversal [1 .. length of iShop]
  output(shopCount + 1)
  output(iShop[i][0])
  output(iShop[i][1])
  output(iShop[i][2])
  shopCount <- shopCount + 1
  potInfo <- [shopCount, iShop[i][0], int(iShop[i][1]), int(iShop[i][2])]
  isTemp.append(potInfo)
shopCount <- 0
return isTemp, shopCount

function notiShopList(shopCount : integer, iShop : list, nisTemp : list, item : list)
-> tuple
  if length of iShop = length of item then
    output("Semua item tersedia di shop.")
  else
    processed_ids <- set()
    shopCount <- length of iShop - 1
    output("ID | Type          |")
    i traversal [1 .. length of item]
      is_in_shop <- False
      j traversal [1 .. length of iShop]
        if item[i][1] = iShop[j][0] then
          is_in_shop <- True
          break
      if not is_in_shop and item[i][1] not in processed_ids then
        output(shopCount + 1)
        output(item[i][1])
        shopCount <- shopCount + 1
        stats <- [shopCount, item[i][1]]
        nisTemp.append(stats)
        processed_ids.add(item[i][1])
    shopCount <- 0
    return nisTemp, shopCount

function SHOP_MANAGEMENT(currentUser : list, mShop : list, iShop : list, mons : list,
item : list) -> tuple
  item <- [[1, "strength"], [2, "resilience"], [3, "healing"], [4, "monster_ball"],
[1, 1]]
  action <- "lihat"
  shopCount <- 0
  id_mons_to_shop <- 0
  msTemp <- []
  nmsTemp <- []
  isTemp <- []
  nisTemp <- []
  output("안녕하세요! Selamat datang kembali, " + currentUser[1])

  while action != "keluar" do
    action <- input(">>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ")

    if action = "lihat" then
      jenis <- input(">>> Mau lihat apa? (monster/item): ")
      if jenis = "monster" then

```

```

        msTemp, shopCount <- mShopList(shopCount, mShop, mons, msTemp)
    else if jenis = "item" then
        isTemp, shopCount <- iShopList(shopCount, iShop, isTemp)

    else if action = "tambah" then
        jenis_tambah <- input(">>> Mau nambahin apa? (monster/item): ")

        if jenis_tambah = "monster" then
            nmsTemp, shopCount, id_mons_to_shop <- notmShopList(shopCount, mShop,
mons, nmsTemp)
            if nmsTemp != [] then
                id_mons <- input(">>> Masukkan id monster: ")
                stok_mons <- input(">>> Masukkan stok awal: ")
                price_mons <- input(">>> Masukkan harga: ")
                output(nmsTemp[id_mons - length of mShop][1])
                output(" telah berhasil ditambahkan ke dalam shop! ")
                mShop.append([id_mons_to_shop, stok_mons, price_mons])
            else
                output("Tidak ada monster yang tersedia untuk ditambahkan.")

        else if jenis_tambah = "item" then
            nisTemp, shopCount <- notiShopList(shopCount, iShop, nisTemp, item)
            if nisTemp != [] then
                id_pot <- input(">>> Masukkan id item: ")
                stok_pot <- input(">>> Masukkan stok awal: ")
                price_pot <- input(">>> Masukkan harga: ")
                output(nisTemp[id_pot - length of iShop][1])
                output(" telah berhasil ditambahkan ke dalam shop! ")
                iShop.append([nisTemp[id_pot - length of iShop][1], stok_pot,
price_pot])
            else
                output("Tidak ada item yang tersedia untuk ditambahkan.")

    else if action = "ubah" then
        jenis_ubah <- input(">>> Mau ubah apa? (monster/item): ")

        if jenis_ubah = "monster" then
            msTemp, shopCount <- mShopList(shopCount, mShop, mons, msTemp)
            id_mons_new <- input(">>> Masukkan id monster: ")
            stok_mons_new <- input(">>> Masukkan stok baru: ")
            price_mons_new <- input(">>> Masukkan harga baru: ")
            if 1 <= id_mons_new < length of mShop then
                if stok_mons_new != "" then
                    mShop[id_mons_new][1] <- int(stok_mons_new)
                if price_mons_new != "" then
                    mShop[id_mons_new][2] <- int(price_mons_new)
                output("Data monster berhasil diubah.")
            else
                output("ID monster tidak valid.")

        else if jenis_ubah = "item" then
            isTemp, shopCount <- iShopList(shopCount, iShop, isTemp)
            id_pot_new <- input(">>> Masukkan id item: ")
            stok_pot_new <- input(">>> Masukkan stok baru: ")
            price_pot_new <- input(">>> Masukkan harga baru: ")

```

```

    if 1 <= id_pot_new < length of iShop then
        if stok_pot_new != "" then
            iShop[id_pot_new][1] <- int(stok_pot_new)
            if price_pot_new != "" then
                iShop[id_pot_new][2] <- int(price_pot_new)
                output("Data item berhasil diubah.")
            else
                output("ID item tidak valid.")
        else if action = "hapus" then
            jenis_hapus <- input(">>> Mau hapus apa? (monster/item): ")

            if jenis_hapus = "monster" then
                msTemp, shopCount <- mShopList(shopCount, mShop, mons, msTemp)
                id_mons_delete <- input(">>> Masukkan id monster: ")
                if 1 <= id_mons_delete < length of mShop then
                    selectedMons <- mons[int(mShop[id_mons_delete][0])][1]
                    mons_confirm <- input(">>> Apakah anda yakin ingin menghapus " +
selectedMons + " dari shop (y/n)? ")
                    if mons_confirm.lower() = "y" then
                        mShop.pop(id_mons_delete)
                        output(selectedMons + " berhasil dihapus dari shop.")
                    else
                        output("Penghapusan dibatalkan.")
                    else
                        output("ID monster tidak valid.")

                else if jenis_hapus = "item" then
                    isTemp, shop

```

F13 - Monster Management

procedure MONSTER(currentUser: User, userpas: array of User, mons: array of Monster, mInv: array of MonsterInventory)

ALGORITMA

```

pengulangan <- True
output("SELAMAT DATANG DI DATABASE PARA MONSTER !!!")
while pengulangan = True do
    if currentUser.role = "admin" then
        output("Available actions :")
        output("1. Tampilkan semua Monster")
        output("2. Tambah Monster baru")
        output("3. Keluar dari Monster Management")
        Aksi <- input(">>> Pilih aksi : ")

        if Aksi = 3 then
            output("Anda keluar dari monster management.")
            pengulangan <- False
            return mons

        elif Aksi = 1 then
            output("Loading database monster...")
            sleep(1.5)

```

```

    output("Database Monster")
    i traversal [len(mons)]
        output(f"{mons[i][0]:<2} | {mons[i][1]:<14} | {mons[i][2]:<9} |
{mons[i][3]:<9} | {mons[i][4]:<2}")

    output("")
    elif Aksi = 2 then
        output("Memulai pembuatan monster baru")

        nama_baru <- input(">>> Masukkan Type / Nama : ")
        I transversal (len(mons))
            while nama_baru = mons[i][1] do
                output("Nama sudah terdaftar, coba lagi!")
                nama_baru <- input(">>> Masukkan Type / Nama : ")

        atk_power <- input(">>> Masukkan ATK Power : ")
        while not is_integer(atk_power) do
            output("Masukkan input bertipe Integer, coba lagi!")
            atk_power <- input(">>> Masukkan ATK Power : ")

        atk_power <- int(atk_power)

        def_power <- input(">>> Masukkan DEF Power (0-50) : ")
        while def_power > 50 or def_power < 0 do
            output("DEF Power harus bernilai 0-50, coba lagi!")
            def_power <- input(">>> Masukkan DEF Power (0-50) : ")

        def_power <- int(def_power)

        hp <- input(">>> Masukkan HP : ")
        hp <- int(hp)

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

        add_mons <- input(">>> Tambahkan Monster ke database (Y/N) : ")
        if add_mons = "Y" then
            mons.append([len(mons), nama_baru, atk_power, def_power, hp])
            output("Monster baru telah ditambahkan!")
        elif add_mons = "N" then
            output("Monster gagal ditambahkan!")

```

F14 - Load

function read_csv(filename: string, folder_path: string) -> array of array of string
{ Membaca file CSV dan mengembalikan data sebagai array dua dimensi }

ALGORITMA

```

    if folder_path[-1] != "/" then
        folder_path <- folder_path + "/"
    file_path <- folder_path + filename
    data <- []

    if not os.path.exists(file_path) then
        output("Error: The file ", filename, " was not found in the folder ",
folder_path, ".")
        return data

    open file_path in mode 'r' as file do
        for each line in file do
            row <- []
            cell <- ""
            for each char in line do
                if char = '\n' or char = '\r' then
                    continue
                end if
                if char = ';' then
                    row.append(cell)
                    cell <- ""
                else
                    cell <- cell + char

                if cell then
                    row.append(cell)
                end if
                data.append(row)
            return data

```

```

procedure show_folders()
{ Menampilkan daftar folder penyimpanan yang tersedia }

```

```

ALGORITMA
    data_directory <- os.path.dirname(__file__) + "../data/"
    available_folders <- os.listdir(data_directory)
    output("Daftar folder penyimpanan yang tersedia:")
    i traversal [len(available_folders)]
        output(i + 1, ". ", available_folders[i])

```

```

function load(folder: string) -> (array of User, array of Monster, array of
ItemInventory, array of MonsterInventory, array of ItemShop, array of MonsterShop,
boolean)
{ Memuat data dari folder yang dipilih }

```

```

ALGORITMA
    data_directory <- os.path.dirname(__file__) + "../data/"
    folder_path <- data_directory + folder
    file_csv <- ["user.csv", "monster.csv", "item_inventory.csv",
"monster_inventory.csv", "item_shop.csv", "monster_shop.csv"]
    if os.path.exists(folder_path) then
        missing_file <- false
        for each file in file_csv do
            if not os.path.exists(os.path.join(folder_path, file)) then

```



```

        missing_file <- true

        if missing_file then
            output("")
            output("Salah satu file .csv dalam folder ini tidak ada! Cek kembali isi
folder anda!")
            output("")
        else
            valid_load <- true
            output("")
            output("Loading...")
            sleep(1)

            userpas <- read_csv("user.csv", folder_path)
            mons <- read_csv("monster.csv", folder_path)
            mInv <- read_csv("monster_inventory.csv", folder_path)
            iInv <- read_csv("item_inventory.csv", folder_path)
            iShop <- read_csv("item_shop.csv", folder_path)
            mShop <- read_csv("monster_shop.csv", folder_path)

            output("Selamat datang di program OWCA!")
            output("Anda belum login, ketik 'HELP' untuk melihat command-command yang
available!")
            valid_load <- true
            return userpas, mons, iInv, mInv, iShop, mShop, valid_load
        else
            output("")
            output("Folder ", folder, " tidak dapat ditemukan!")
            show_folders()
            output("")
            sys.exit()

```

```

procedure start()
{ Memulai proses dengan parsing argumen dan memuat data dari folder }

```

ALGORITMA

```

    parser <- argparse.ArgumentParser(description="Load Data Program OWCA. Format:
python main.py <nama_folder>. Tulis NEWGAME pada <nama_folder> untuk membuat folder
baru.")
    parser.add_argument("folder", help="Nama folder tempat penyimpanan data program")
    args <- parser.parse_args()

    if not args.folder then
        output("Masukkan nama folder tempat penyimpanan data program Anda!")
        output("Pertama kali bermain? Tulis NEWGAME pada <nama_folder>!")
        output("Usage: python main.py <nama_folder>")
        output("")
        show_folders()
    else
        return load(args.folder)

```

F15 - Save

```

procedure write_csv(folder_path: string, csv_file: string, array_data: array of array
of string)

```

```

open file with write mode as file:
  i traversal [length of array_data]
    line <- ""
    j traversal [length of array_data[i]]
      line <- line + str(array_data[i][j])
      if j < length of array_data[i] - 1 then
        line <- line + ";"
      else do
        line <- line + "\n"

    write line to file

procedure save_data(userpas, mons, mShop, mInv, iShop, iInv: array of array of string)
  folder_name <- input("Masukkan nama folder : ")
  data_directory <- dirname(__file__) + "../data/"
  folder_path <- data_directory + "{folder_name}"
  if exists(data_directory) then
    if exists(folder_path) then
      output(f"Folder /data/{folder_name} sudah ada.")
      sleep(1)
    else
      make directory at folder_path
      output(f"Membuat folder /data/{folder_name}...")
      sleep(1)
      write_csv(folder_path, "user.csv", userpas)
      write_csv(folder_path, "monster.csv", mons)
      write_csv(folder_path, "monster_shop.csv", mShop)
      write_csv(folder_path, "monster_inventory.csv", mInv)
      write_csv(folder_path, "item_shop.csv", iShop)
      write_csv(folder_path, "item_inventory.csv", iInv)
      output(f"Data berhasil disimpan ke folder /data/{folder_name}.")

  else
    output("Folder data tidak ditemukan.")
    output(f"Membuat folder /data/...")
    sleep(1)
    make directory at data_directory
    output(f"Folder /data berhasil dibuat")
    output(f"Membuat folder /data/{folder_name}...")
    sleep(1)
    make directory at folder_path
    write_csv(folder_path, "user.csv", userpas)
    write_csv(folder_path, "monster.csv", mons)
    write_csv(folder_path, "monster_shop.csv", mShop)
    write_csv(folder_path, "monster_inventory.csv", mInv)
    write_csv(folder_path, "item_shop.csv", iShop)
    write_csv(folder_path, "item_inventory.csv", iInv)
    output(f"Data berhasil disimpan ke folder /data/" + folder_name)

```

F16 - Exit

```

function EXIT() ->
  while True do
    response <- input("Apakah Anda mau melakukan penyimpanan file yang sudah
      diubah? (y/n) ").lower()

```

```
if response == 'y' then  
    save()  
    output("Data telah disimpan. Selamat tinggal!")  
    break  
else if response == 'n' then  
    output("Selamat tinggal!")  
    break  
else  
    output("Input tidak valid. Silahkan coba lagi.")
```

BAB VI

SPESIFIKASI MODUL YANG DIBUAT

F00 - Random Number Generator

{Fitur ini berfungsi untuk memberikan angka secara acak bertujuan agar monster, serangan, dan level berbeda-beda}

{I.S. : Fungsi menerima nilai batas bawah dan batas atas untuk hasil acak, serta parameter LCG seperti seed, a, c, dan m. Fungsi rngLevel, rngEnemy, rngOC, rngCapture, playerAtk, dan enemyAtk menerima parameter tambahan spesifik untuk kebutuhan masing-masing.}

{F.S. : Fungsi mengembalikan nilai acak sesuai dengan parameter yang diberikan, baik itu level, tipe musuh, jumlah OC, keberhasilan tangkapan, atau hasil serangan pemain dan musuh.}

F01 - Register

{Fitur ini berfungsi untuk mendaftarkan akun agar user dapat memasuki sistem}

{I.S. : Menerima argumen userpas sebagai data seluruh pengguna yang terdaftar dalam bentuk list of lists, dimana setiap elemen list adalah data pengguna yang terdiri dari: [id, username, password, role, oc]. Menerima argumen mInv sebagai data inventaris monster yang dimiliki pengguna dalam bentuk list of lists, dimana setiap elemen list adalah data monster yang terdiri dari: [id_pengguna, id_monster, jumlah]. Akan diterima input data pengguna baru, yaitu username dan password.}

{F.S. : Data user baru akan ditambahkan ke dalam database.}

F02 - Login

{Fitur ini berfungsi untuk masuk ke program menggunakan akun yang sudah tersedia (terdaftar dalam register)}

{I.S. : userList dan passList terdefinisi, lalu menerima argumen userpas dari user.}

{F.S. : tercetak "Selamat datang,", currentUser, loginUsername, "!
Masukkan command "help" untuk melihat daftar command yang dapat kamu panggil" apabila password dan username benar. Dan tercetak "Username tidak terdaftar!" dan "Password salah!" jika username atau password salah.}

F03 - Logout

{Fitur ini berfungsi untuk user keluar dari akun yang sedang dimainkan}

{I.S. : Menerima argumen currentUser sebagai data pengguna yang saat ini login dan login sebagai status login (True jika pengguna sudah login, False jika belum login).}

{F.S. : Jika status login awalnya True, currentUser akan dikosongkan dan login diubah menjadi False. Pesan konfirmasi logout atau pesan kegagalan logout akan ditampilkan.}

F04 - Menu & Help

{Fitur ini berfungsi untuk membantu user memahami fitur dalam program yang sedang dimainkan}

{I.S. : menerima argumen currentUser sebagai data pengguna yang saat ini login.}

{F.S. : menampilkan pesan bantuan yang sesuai dengan pengguna yang sedang online, termasuk perintah-perintah yang tersedia dan petunjuk penggunaan program.}

F05 - Monster

{Fitur ini berfungsi untuk menyimpan data monster yang dimiliki user}

{I.S. : matrix monster terdefinisi}

{F.S. : menyimpan monster beserta atribut-atributnya yang dimiliki di inventory sesuai level masing-masing monster}

F06 - Potion

{Fitur ini berfungsi untuk menyimpan data jenis potion beserta efeknya dan jumlahnya yang dimiliki user}

{I.S. : Fungsi menerima tiga parameter: atk_power, def_power, dan hp, serta jenis potion (type_potion) yang digunakan (strength, resilience, atau healing).}

{F.S. : Fungsi mengembalikan atribut monster yang telah ditingkatkan sesuai dengan jenis potion yang digunakan, dengan batasan maksimal untuk def_power dan hp.}

F07 - Inventory

{Fitur ini berfungsi untuk menampilkan data item dan monster yang dimiliki user}

{I.S. : matrix mInv, mons, dan iInv terdefinisi}

{F.S. : menampilkan jumlah O.W.C.A Coin , daftar monster beserta stats, potion beserta potInfo, dan item lainnya yang dimiliki user}

F08 - Battle

{Fitur ini berfungsi untuk memainkan program dengan cara bertarung melawan monster lain menggunakan monster yang dimiliki dalam inventory}

{I.S. : Fungsi menerima data pengguna (currentUser), database monster (mons), database inventory monster pengguna (mInv), database inventory item pengguna (iInv), serta dua fungsi random generator untuk musuh (rngEnemy) dan level (rngLevel).}

{F.S. : Fungsi mengembalikan data pengguna yang mungkin telah berubah, seperti level OC, database inventory monster pengguna (mInv), dan database inventory item pengguna (iInv).}

F09 - Arena

{Fitur ini seperti fitur Battle hanya saja memiliki tingkatan dalam pertarungan untuk melatih user dan monster}

{I.S. : Fungsi ARENA menerima data monster (mons), daftar inventory monster pengguna (mInv), fungsi random generator untuk musuh (rngEnemy), data pengguna (currentUser), dan daftar inventory item pengguna (iInv). Selain itu, fungsi ini juga bergantung pada fungsi-fungsi lain seperti userPot, yourTurn, enemyTurn, monster, dmgCalc, userBall, dan chooseMons yang dimuat dari berkas-berkas terpisah.}

{F.S. : Setelah menjalankan fungsi ARENA, data pengguna (currentUser) dapat mengalami perubahan, terutama dalam jumlah O.W.C.A. Coin yang dimilikinya. Fungsi juga memberikan output statistik seperti total hadiah, jumlah stage yang berhasil diselesaikan, jumlah damage yang diberikan, dan jumlah damage yang diterima selama sesi berlangsung. Jika sesi berakhir, fungsi mengembalikan informasi tersebut sebagai bagian dari outputnya.}

F10 - Shop & Currency

{Fitur ini bertujuan menyajikan item dan monster yang dapat dijual kepada user}

{I.S. : iShopList, mShopList terdefinisi}

{F.S. : fitur ini menampilkan iShoplist jika user memberi command lihat lalu potion dan mShoplist jika user memberi command lihat lalu monster. Dan menyimpan item dari shop ke dalam Inventory jika user memberi command beli dengan syarat user memiliki OC yang cukup, terdapat stok item dalam shop, dan user tidak membeli monster yang sudah dimiliki dalam inventory.}

F11 - Laboratory

{Fitur ini berfungsi untuk melakukan upgrade level monster yang dimiliki di inventory}

{I.S. : matrix monsters yang terdiri dari monster dan level terdefinisi}

{F.S. : Apabila user memiliki currency yang cukup untuk melakukan upgrade monster yang dipilih, maka level monster akan bertambah. Jika tidak, akan mencetak "Maaf, anda tidak memiliki OC yang cukup untuk melakukan upgrade}

F12 - Shop Management

{Fitur ini bertujuan agar user admin dapat mengubah data item dan monster dalam shop}

{I.S. : Fungsi SHOP_MANAGEMENT menerima data pengguna (currentUser), daftar monster yang tersedia di toko (mShop), daftar item yang tersedia di toko (iShop), daftar monster yang ada dalam permainan (mons), dan daftar item yang ada dalam permainan (item).}

{F.S. : Setelah menjalankan fungsi SHOP_MANAGEMENT, data mengenai daftar monster dan item yang tersedia di toko (mShop dan iShop) dapat mengalami perubahan berdasarkan aksi yang dilakukan pengguna seperti menambah, mengubah, atau menghapus entri. Fungsi mengembalikan data baru tentang daftar monster dan item yang terupdate setelah aksi dilakukan.}

F13 - Monster Management

{Fitur ini bertujuan agar user admin dapat mengubah dan menambah data monster dalam program}

{I.S. : Modul MONSTER menerima input dari pengguna untuk memilih aksi yang ingin dilakukan, seperti menampilkan daftar monster, menambahkan monster baru, atau keluar dari manajemen monster. Modul juga menerima data pengguna (currentUser) yang digunakan untuk memverifikasi hak akses.}

{F.S. : Setelah pengguna memilih aksi yang diinginkan, modul MONSTER akan menampilkan daftar monster sesuai permintaan pengguna, menambahkan monster baru jika dipilih, atau keluar dari manajemen monster. Jika pengguna adalah admin, modul akan memungkinkan pengguna untuk melakukan manajemen penuh terhadap database monster.}

F14 - Load

{Fitur ini berguna membaca dan menampilkan data csv yang tersimpan ke dalam program menggunakan Argumentparser}

{I.S. : Fungsi read_csv menerima dua parameter: filename yang merupakan nama file CSV yang akan dibaca, dan folder_path yang merupakan path folder tempat file CSV disimpan. Fungsi show_folders tidak menerima parameter masukan.

{F.S. : Setelah menjalankan fungsi read_csv, fungsi mengembalikan sebuah list data yang berisi data dari file CSV yang telah dibaca. Jika file tidak ditemukan, fungsi akan mencetak pesan error dan mengembalikan list kosong. Setelah menjalankan fungsi show_folders, fungsi akan mencetak daftar folder penyimpanan yang tersedia beserta nomor indeksinya. Tidak ada perubahan pada data atau variabel lain setelah fungsi dijalankan.

F15 - Save

{Fitur ini bertujuan menyimpan data ke dalam file setelah terjadi perubahan akibat penggunaan program oleh user}

{I.S. : Fungsi write_csv menerima tiga parameter: folder_path yang merupakan path ke folder tempat file CSV akan disimpan, csv_file yang merupakan nama file CSV yang akan ditulis, dan array_data yang merupakan data yang akan ditulis ke dalam file CSV. Fungsi save_data menerima enam parameter: userpas, mons, mShop, mInv, iShop, dan iInv, yang masing-masing merupakan data pengguna,

```

monster, item toko monster, inventori monster, toko item, dan
inventori item.}
{F.S. : Setelah menjalankan fungsi write_csv, fungsi akan menulis data
dari array_data ke dalam file CSV yang sesuai dengan nama
csv_file di folder yang ditentukan oleh folder_path. Tidak ada
nilai kembalian yang dihasilkan dari fungsi ini.Setelah
menjalankan fungsi save_data, fungsi akan meminta pengguna untuk
memasukkan nama folder penyimpanan. Jika folder tersebut belum
ada, fungsi akan membuat folder baru. Kemudian, fungsi akan
menyimpan data pengguna, monster, toko monster, inventori
monster, toko item, dan inventori item ke dalam file CSV di dalam
folder tersebut. Setelah proses penyimpanan selesai, fungsi akan
mencetak pesan berhasil disimpan ke folder yang sesuai. Tidak ada
nilai kembalian yang dihasilkan dari fungsi ini.}

```

F16 - Exit

```

{Fitur ini bertujuan keluar dari program atau permainan}
{I.S. : Modul EXIT akan meminta input dari pengguna untuk menentukan
apakah ingin menyimpan perubahan data sebelum keluar dari
program.}
{F.S. : Jika pengguna memilih untuk menyimpan perubahan (input 'y'),
perubahan data akan disimpan menggunakan modul SAVE, kemudian
aplikasi akan keluar dengan pesan "Data telah disimpan. Selamat
tinggal!". Jika pengguna memilih untuk tidak menyimpan perubahan
(input 'n'), aplikasi akan keluar dengan pesan "Selamat
tinggal!". Jika input tidak valid, pengguna akan diminta untuk
memasukkan input lagi.}

```

B03 - Monster Ball

```

{Fitur ini merupakan fitur tambahan dalam fitur Battle yang bertujuan
untuk menangkap monster liar}
{I.S. : Fungsi menerima inventori pengguna (iInv), inventori monster
pengguna (userMons), fungsi untuk mengecek jumlah Monster Ball
(userBall), fungsi random capture (rngCapture), database
inventori monster global (mInv), tipe monster musuh (enemy_type),
data pengguna (currentUser), data monster musuh yang dipilih
(chosenEnemy), level monster musuh (enemy_level), dan statistik
monster musuh (enemy_atk_power, enemy_def_power, enemy_hp).}
{F.S. : Fungsi mengembalikan status pertempuran yang diperbarui
berdasarkan hasil percobaan menangkap monster: end yang
menunjukkan apakah pertempuran berakhir, dan cancel yang
menunjukkan apakah aksi ditunda. Jika monster berhasil ditangkap,
data monster ditambahkan ke inventori monster pengguna (mInv).}

```


BAB VII

HASIL PENGUJIAN PROGRAM

F00 - Random Number Generator

F01 - Register

Modul untuk melakukan registrasi user baru.

Gambar 7.1.1 Registrasi

```
>>> register
Masukkan username: ilhamku
Masukkan password: gatau

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachu
2. Bulbu
3. Zeze
4. Zuko
5. Chacha

Monster pilihanmu: 1
Selamat datang Agent ilhamku. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachu!
```

Gambar 7.1.2 Registrasi dengan Username yang sudah terpakai

```
Monster pilihanmu: 1
Selamat datang Agent ilhamku. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachu!

>>> register
Masukkan username: Agen_P
Masukkan password: platypus123

Username Agen_P sudah terpakai, silahkan gunakan username lain!
Masukkan username: 
```

Gambar 7.1.3 Registrasi dengan kesalahan Username

```
>>> register
Masukkan username: agen&
Masukkan password: ilham

Username hanya boleh berisi alfabet, angka, underscore, dan strip
Masukkan username: 
```

Gambar 7.1.4 Registrasi gagal

```
>>> register
Register gagal!
Anda telah login dengan username Agen_P, silahkan lakukan "LOGOUT" sebelum melakukan register.
```

F02 - Login

Modul untuk melakukan login user

Gambar 7.2.1 Log in

```
>>> login
Masukkan username : Agen_P
Masukkan password : platypus123
Selamat datang, Agent Agen_P!
Masukkan command "help" untuk daftar command yang dapat kamu panggil.
```

Gambar 7.2.2 Log in dengan username tidak terdaftar

```
>>> login
Masukkan username : Agen
Masukkan password : platypus123
Username tidak terdaftar!
```

Gambar 7.2.3 Log in dengan password salah

```
>>> login
Masukkan username : Agen_P
Masukkan password : platy
Password salah!
```

Gambar 7.2.4 Log in gagal

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

F03 - Logout

Modul untuk melakukan logout user

Gambar 7.3.1 Log out

```
>>> logout
Anda berhasil logout!!
```

Gambar 7.3.2 Logout gagal

```
>>> logout
Logout gagal!
Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout
```

F04 - Menu & Help

Modul untuk memberikan penjelasan dan bantuan atas menu yang ada

Gambar 7.4.1 Tampilan Menu & Help sebelum log in

```
>>> help

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

Gambar 7.4.2 Tampilan Menu & Help setelah login untuk user Admin

```
>>> help

===== HELP =====
Selamat datang, Admin. Berikut adalah hal-hal yang dapat kamu lakukan:
  1. Logout: Keluar dari akun yang sedang digunakan
  2. Shop: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
  3. Monster: Melakukan manajemen monster
  4. Save: Menyimpan semua progres
  5. Exit: Keluar dari program
Footnote :
  1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
  2. Jangan lupa untuk memasukkan input yang valid
```

Gambar 7.4.3 Tampilan Menu & Help setelah login untuk user Agent

```
>>> help

===== HELP =====
Halo Agent Agen P. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian.
Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
  1. Logout: Keluar dari akun yang sedang digunakan
  2. Monster: Melihat owca-dex yang dimiliki oleh Agent
  3. Battle: Melawan satu musuh secara random
  4. Arena: Memasuki arena yang memiliki 5 tingkat, yang mana player akan melawan monster dengan level sesuai tingkat secara melanjut
  5. Shop: Memasuki tempat membeli monster dan potion
  6. Laboratory: Memasuki tempat meningkatkan level monster dengan biaya tertentu
  7. Save: Menyimpan semua progres
  8. Exit: Keluar dari program
Footnote :
  1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
  2. Jangan lupa untuk memasukkan input yang valid
```

F05 - Monster

Modul untuk menyimpan data-data para monster

F06 - Potion

Modul untuk menyimpan data-data potion

F07 - Inventory

Modul untuk memaparkan data item dan monster yang dimiliki user

Gambar 7.7.1 Tampilan Inventory user

```
>>> inventory
===== INVENTORY LIST (User ID: 3) =====
Jumlah O.W.C.A. Coin-mu sekarang 10000
1. Monster      (Name: Bulbu, Lvl: 2, HP: 1200)
2. Monster      (Name: Zeze, Lvl: 1, HP: 100)
3. Monster      (Name: Theo, Lvl: 2, HP: 1200000)
4. Monster      (Name: Chacha, Lvl: 2, HP: 700)
5. Monster      (Name: Zuko, Lvl: 3, HP: 800)
6. Potion       (Type: resilience, Qty: 7)
7. Monster Ball (Qty: 9999999999996)

Masukkan pilihan sesuai angka diatas atau 'KELUAR'
>> █
```

Gambar 7.7.2 Tampilan monster user

```
Masukkan pilihan sesuai angka diatas atau 'KELUAR'
>> 1
Monster
Name      : Bulbu
ATK Power : 50
Def Power : 50
HP        : 1200
Level     : 2
```

Gambar 7.7.3 Tampilan item potion user

```
Masukkan pilihan sesuai angka diatas atau 'KELUAR'
>> 6
Potion
Type      : resilience
Quantity   : 7
```

Gambar 7.7.4 Tampilan item monster ball user

```
Masukkan pilihan sesuai angka diatas atau 'KELUAR'
>> 7
Monster Ball
Quantity   : 9999999999996
```

Gambar 7.7.5 Tampilan keluar dari Inventory

```
Masukkan pilihan sesuai angka diatas atau 'KELUAR'  
>> keluar  
Keluar dari inventory...
```

F08 - Battle

Modul untuk bertarung mengalahkan monster lainnya

Gambar 7.8.1 Tampilan monster liar muncul dan pilihan monster untuk bertarung

```
>>> battle  
  
("'-''-/").____.---''''_.--_.  
`6_6 )  `-. ( )`-.____.)  
(_Y_.)'  ._)  `_.--'`--'---'  
_.--'--'_.--'/ /--'_.'  
((((_.--' ((((_.--' (((_.--'  
RAWRRR, Monster Bulbu telah muncul !!!  
  
Name      : Bulbu  
ATK Power : 55  
DEF Power : 50  
HP        : 1320  
Level     : 2  
===== MONSTER LIST =====  
1. Bulbu  
2. Zeze  
3. Theo  
4. Chacha  
5. Zuko  
Pilih monster untuk bertarung: █
```

Gambar 7.8.2 Tampilan gagal memilih monster (pilihan tidak tersedia)

```
===== MONSTER LIST =====  
1. Bulbu  
2. Zeze  
3. Theo  
4. Chacha  
5. Zuko  
Pilih monster untuk bertarung: 6  
  
Pilihan nomor tidak tersedia!  
  
Pilih monster untuk bertarung: █
```

Gambar 7.8.3 Tampilan monster yang dipilih dan perintah

```
Pilih monster untuk bertarung: 1

      .""".
     .-./ _=_ \.-.
    { (,(oYo),) }}
   {{ |  "  |}}
  { { \(___)/  }}
 {{  }'--'{} }
 { { }.:_._{  }}
 {{  } -:- {  }}
 {_{ }`===`{ _}
 ((((\      (/)))

RAWRRR, Agent Agen_P mengeluarkan monster Bulbu !!!

Name      : Bulbu
ATK Power : 55
DEF Power : 50
HP        : 1320
Level     : 2
===== TURN 1 (Bulbu) =====
1. Attack
2. Use Potion
3. Use Monster Ball
4. Flee
Pilih perintah: █
```

Gambar 7.8.4 Tampilan kabur dari pertempuran

```
===== TURN 1 (Bulbu) =====
1. Attack
2. Use Potion
3. Use Monster Ball
4. Flee
Pilih perintah: 4

Anda berhasil kabur dari BATTLE!
```

Gambar 7.8.5 Tampilan monster tidak bisa ditangkap

```
Anda telah memilih monster Zeze
Pilih perintah: 3

Anda sudah memiliki monster Zeze dalam inventory!
```

Gambar 7.8.6 Tampilan monster gagal ditangkap

```
Pilih perintah: 3

Swoosshhhh, Anda mengeluarkan Monster Ball !!!
Yahhh, Anda belum berhasil mendapatkan monster Pikachu !!!

Sisa Monster Ball Anda: 999999999995

===== TURN 1 (Bulbu) =====
```

Gambar 7.8.7 Tampilan monster berhasil ditangkap

```
Swoosshhhh, Anda mengeluarkan Monster Ball !!!
Selamat, Anda berhasil mendapatkan monster Pikachu !!!

Name      : Pikachu
ATK Power : 125
DEF Power : 10
HP        : 600
Level     : 1

Sisa Monster Ball Anda: 999999999994
```

Gambar 7.8.8 Tampilan menggunakan potion

```
===== TURN 1 (Theo) =====
1. Attack
2. Use Potion
3. Use Monster Ball
4. Flee
Pilih perintah: 2

===== POTION LIST =====
1. Strength Potion (Qty: 0) - Increases ATK Power
2. Resilience Potion (Qty: 7) - Increases DEF Power
3. Healing Potion (Qty: 0) - Restores Health
4. Cancel

Pilih potion atau cancel: █
```

Gambar 7.8.9 Tampilan potion habis

```
Pilih potion atau cancel: 1
Wah, kamu sedang tidak memiliki ramuan ini, silahkan pilih ramuan lain!

===== POTION LIST =====
```


Gambar 7.8.10 Tampilan batal menggunakan potion

```
Pilih potion atau cancel: 4
```

```
===== TURN 1 (Theo) =====
```

Gambar 7.8.11 Tampilan monster setelah menggunakan strength potion

```
Pilih potion atau cancel: 1
Setelah meminum ramuan ini, aura kekuatan terlihat mengelilingi Bulbu dan gerakannya menjadi lebih cepat dan mematikan.

===== TURN 2 (Bulbu) =====
```

Gambar 7.8.12 Tampilan monster setelah menggunakan resilience potion

```
Pilih potion atau cancel: 2
Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Bulbu yang membuatnya terlihat semakin tangguh dan sulit dilukai.

===== TURN 3 (Bulbu) =====
```

Gambar 7.8.13 Tampilan monster setelah menggunakan healing potion

```
Pilih potion atau cancel: 3
Setelah meminum ramuan ini, luka-luka yang ada di dalam tubuh Bulbu sembuh dengan cepat. Dalam sekejap, Bulbu terlihat kembali prima dan siap melanjutkan pertempuran.

===== TURN 2 (Pikachow) =====
```

Gambar 7.8.14 Tampilan pilihan potion tidak valid

```
===== POTION LIST =====
1. Strength Potion (Qty: 9) - Increases ATK Power
2. Resilience Potion (Qty: 5) - Increases DEF Power
3. Healing Potion (Qty: 2) - Restores Health
4. Cancel

Pilih potion atau cancel: 5
Perintah tidak valid. Silahkan pilih perintah diatas!

===== POTION LIST =====
```

Gambar 7.8.15 Tampilan monster setelah menggunakan potion untuk kedua kali

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

Pilih potion atau cancel: 3
Kamu mencoba memberikan ramuan ini kepada Bulbu, namun dia menolaknya seolah-olah dia memahami ramuan tersebut sudah tidak bermanfaat lagi.

===== POTION LIST =====
```

Gambar 7.8.16 Tampilan pertarungan para monster

```
Pilih perintah: 1

SCHWINKKK, Bulbu menyerang Zeze !!!

Name      : Zeze
ATK Power : 300
DEF Power : 10
HP        : 58
Level     : 1

Penjelasan : ATT: 46.5 (-15%), Reduced by: 4.65 (10%), ATT Results: 41

False
===== TURN 1 (Zeze) =====

SCHWINKKK, Zeze menyerang Bulbu !!!

Name      : Bulbu
ATK Power : 55
DEF Power : 50
HP        : 1160
Level     : 2

Penjelasan : ATT: 319.0 (6%), Reduced by: 159.5 (50%), ATT Results: 159

===== TURN 2 (Bulbu) =====
```

Gambar 7.8.17 Tampilan monster user berhasil memenangkan pertarungan

```
Pilih perintah: 1

SCHWINKKK, Theo menyerang Zuko !!!

Name      : Zuko
ATK Power : 110
DEF Power : 27
HP        : 0
Level     : 2

Penjelasan : ATT: 14672850.0 (11%), Reduced by: 3961669.5000000005 (27%), ATT Results: 10711180

Selamat, Anda berhasil mengalahkan monster Zuko !!!
Total OC yang diperoleh: 29
```

Gambar 7.8.18 Tampilan monster user gagal memenangkan pertarungan

```
Pilih perintah: 1

SCHWINKKK, Bulbu menyerang Theo !!!

Name      : Theo
ATK Power : 13200000
DEF Power : 50
HP        : 1319964
Level     : 2

Penjelasan : ATT: 70.5 (28%), Reduced by: 35.25 (50%), ATT Results: 35

===== TURN 1 (Theo) =====

SCHWINKKK, Theo menyerang Bulbu !!!

Name      : Bulbu
ATK Power : 55
DEF Power : 50
HP        : 0
Level     : 2

Penjelasan : ATT: 12065381.0 (-8%), Reduced by: 6032690.5 (50%), ATT Results: 6032690

Yahhh, Anda dikalahkan monster Theo. Jangan menyerah, coba lagi !!!
```

F09 - Arena

Modul yang mirip dengan battle dengan tambahan tahapan

Gambar 7.9.1 Tampilan Arena

```
>>> arena
Selamat datang di Arena!!
===== MONSTER LIST =====
1. Bulbu
2. Zeze
3. Theo
4. Chacha
5. Zuko
Pilih monster untuk bertarung: █
```

Gambar 7.9.2 Tampilan gagal memilih monster (pilihan tidak tersedia)

```
Pilih monster untuk bertarung: 6

Pilihan nomor tidak tersedia!

Pilih monster untuk bertarung: █
```

Gambar 7.9.3 Tampilan user mengeluarkan monster

```

3. Leko
Pilih monster untuk bertarung: 1

      .~.
    .-./ _=_ \.-.
  {  (,(oYo),)  }
  {{ |  "  | } }
  { { \(---)/  } }
  {{ }'--' { } }
  { { }._:_. { } }
  {{ } -:- { } }
  { { }^=== { } }
  ((((\)      (/))))

RAWRRR, Agent Agen_P mengeluarkan monster Bulbu !!!

Name      : Bulbu
ATK Power : 55
DEF Power : 50
HP         : 1320
Level     : 2

```

Gambar 7.9.4 Tampilan tahap pertama Arena

```

===== STAGE 1 =====

("~-''-/")._____.'''~_._
`6_ 6 )   `-. (   ).`-._._.)
(_Y_.)'   _ )   `._.'`-._._.'
_..`-.'-._./ /--'_.
(((.-'' (((.-' (((.-'
RAWRRR, Monster Bulbu telah muncul !!!

      Name      : Bulbu
      ATK Power : 50
      DEF Power : 50
      HP         : 1200
      Level     : 1
===== TURN 1 (Bulbu) =====
1. Attack
2. Use Potion
3. Quit
Pilih perintah: █

```

Gambar 7.9.5 Tampilan gagal menaklukkan tahap pertama Arena

```
GAME OVER! Anda mengakhiri sesi latihan!  
===== STATS =====  
  
Total Hadiah      : 0  
Jumlah Stage      : 0  
Damage diberikan  : 0  
Damage diterima   : 0
```

Gambar 7.9.6 Tampilan berhasil memenangkan tahapan pertama dan lanjut ke tahap berikutnya

```
Selamat, Anda berhasil mengalahkan monster Theo !!!  
STAGE CLEARED! Anda akan mendapatkan 30 OC pada sesi ini!  
Memulai stage berikutnya...  
  
===== STAGE 2 =====
```

Gambar 7.9.7 Tampilan berhasil memenangkan seluruh tahapan dalam Arena

```
Selamat, Anda berhasil mengalahkan monster Bulbu !!!  
STAGE CLEARED! Anda akan mendapatkan 250 OC pada sesi ini!  
Memulai stage berikutnya...  
  
Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!  
===== STATS =====  
  
Total Hadiah      : 590  
Jumlah Stage      : 5  
Damage diberikan  : 51185258.45999999  
Damage diterima   : 0
```

Gambar 7.9.8 Tampilan user gagal salah satu tahapan

```
Yahhh, Anda dikalahkan monster Zuko. Jangan menyerah, coba lagi !!!  
GAME OVER! Sesi latihan berakhir pada stage 1!  
===== STATS =====  
  
Total Hadiah      : 0  
Jumlah Stage      : 1  
Damage diberikan  : 517.5  
Damage diterima   : 162.9
```

F10 - Shop & Currency

Modul yang memberikan user item maupun monster yang dapat dibeli beserta koin user

Gambar 7.10.1 Tampilan Shop

```
>>> shop
안녕하세요! Selamat Datang di SHOP!!!
Pilih aksi (lihat/beli/keluar):
```

Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop

```
Pilih aksi (lihat/beli/keluar): lihat
Mau lihat apa? (monster/item): monster

ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pikachow       | 125 | 10 | 600 | 10 | 500
2  | Zeze           | 300 | 10 | 100 | 3 | 1000
3  | Zuko           | 100 | 25 | 800 | 8 | 550
4  | Chacha         | 80 | 30 | 700 | 6 | 600
```

Gambar 7.10.3 Tampilan aksi melihat item dalam Shop

```
Pilih aksi (lihat/beli/keluar): lihat
Mau lihat apa? (monster/item): item

ID | Type           | Stok | Harga
1  | strength       | 20 | 40
2  | resilience     | 5 | 30
3  | healing        | 3 | 20
4  | monster_ball   | 0 | 10
```

Gambar 7.10.4 Tampilan aksi membeli monster dalam Shop

```
Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 10000

Mau beli apa? (monster/item): monster
Masukkan id monster: 1
Berhasil membeli item: Pikachow. Item sudah masuk ke inventory-mu!
```

Gambar 7.10.5 Tampilan aksi membeli item dalam Shop

```
Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 9000

Mau beli apa? (monster/item): item
Masukkan id item: 2
Masukkan jumlah item: 2
Berhasil membeli item: 2 strength. Item sudah masuk ke inventory-mu!
```

Gambar 7.10.6 Tampilan aksi membeli monster yang sudah dimiliki

```
Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 8920

Mau beli apa? (monster/item): monster
Masukkan id monster: 2
Monster Zeze sudah ada dalam inventory-mu! Pembelian dibatalkan.
```

Gambar 7.10.7 Tampilan aksi membeli dengan coin (OC) yang tidak cukup

```
Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 10

Mau beli apa? (monster/item): item
Masukkan id item: 1
Masukkan jumlah item: 10
OC-mu tidak cukup.
```

Gambar 7.10.8 Tampilan aksi keluar dari Shop

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

>>>
```

F11 - Laboratory

Modul untuk meningkatkan level monster user

Gambar 7.11.1 Tampilan Laboratory

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

===== MONSTER LIST =====
1. Bulbu (Level: 2)
2. Zeze (Level: 1)
3. Theo (Level: 2)
4. Chacha (Level: 2)
5. Zuko (Level: 3)

===== 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
=====

Apa yang ingin Anda lakukan?
1. Upgrade monster
2. Exit
>>> Masukkan pilihan (1/2): █
```

Gambar 7.11.2 Tampilan menaik level monster

```
Apa yang ingin Anda lakukan?
1. Upgrade monster
2. Exit
>>> Masukkan pilihan (1/2): 1

===== MONSTER LIST =====
1. Bulbu (Level: 2)
2. Zeze (Level: 1)
3. Theo (Level: 2)
4. Chacha (Level: 2)
5. Zuko (Level: 3)
>>> Pilih monster: 1
Bulbu akan di-upgrade ke level 3.
Harga untuk melakukan upgrade Bulbu adalah 500 OC.
>>> Lanjutkan upgrade (Y/N): y
Selamat, Bulbu berhasil di-upgrade ke level 3!
```


Gambar 7.11.3 Tampilan monster sudah level maks

```
>>> Masukkan pilihan (1/2): 1

===== MONSTER LIST =====
1. Bulbu (Level: 5)
2. Zeze (Level: 1)
3. Theo (Level: 2)
4. Chacha (Level: 2)
5. Zuko (Level: 3)
>>> Pilih monster: 1
Maaf, monster sudah mencapai level maksimum.
```

Gambar 7.11.4 Tampilan pilihan tidak valid

```
===== MONSTER LIST =====
1. Bulbu (Level: 2)
2. Zeze (Level: 1)
3. Theo (Level: 2)
4. Chacha (Level: 2)
5. Zuko (Level: 3)
>>> Pilih monster: 6
Pilihan monster tidak valid.
```

F12 - Shop Management

Modul untuk admin yang bertujuan mengubah data item dalam shop

Gambar 7.12.1 Tampilan Shop untuk Admin

```
>>> shop
안녕하세요! Selamat datang kembali, Mr_Monogram
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.12.2 Tampilan aksi melihat monster dalam Shop untuk Admin

```
>>> shop
안녕하세요! Selamat datang kembali, Mr_Monogram
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/item): monster
ID | Type          | ATK Power | DEF Power | HP   | Stok | Harga
1 | Pikachow       | 125 | 10 | 600 | 10 | 500
2 | Zeze           | 300 | 10 | 100 | 3 | 1000
3 | Zuko           | 100 | 25 | 800 | 8 | 550
4 | Chacha         | 80 | 30 | 700 | 6 | 600
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.12.3 Tampilan aksi melihat item dalam Shop untuk Admin

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/item): item
ID | Type                | Stok | Harga
1  | strength              | 20   | 40
2  | resilience             | 5    | 30
3  | healing                | 3    | 20
4  | monster_ball           | 0    | 10
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
```

Gambar 7.12.4 Tampilan aksi menambah monster dalam Shop untuk Admin

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): monster
ID | Type                | ATK Power | DEF Power | HP
5  | Bulbu               | 50        | 50        | 1200
6  | Theo                | 12000000  | 50        | 1200000
>>> Masukkan id monster: 5
>>> Masukkan stok awal: 2
>>> Masukkan harga: 5000
Bulbu telah berhasil ditambahkan ke dalam shop!
```

Gambar 7.12.5 Tampilan error message bila semua item sudah ada di shop

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): item
Semua item tersedia di shop.
```

Gambar 7.12.6 Tampilan menambahkan item ke shop

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): item
ID | Type
4 | strength
>>> Masukkan id item: 4
>>> Masukkan stok awal: 10
>>> Masukkan harga: 400
strength telah berhasil ditambahkan ke dalam shop!
```

Gambar 7.12.7 Tampilan mengubah stok dan harga monster

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): monster
ID | Type | ATK Power | DEF Power | HP | Stok | Harga
1 | Pikachu | 125 | 10 | 600 | 5 | 400
2 | Zeze | 300 | 10 | 100 | 3 | 1000
3 | Zuko | 100 | 25 | 800 | 8 | 550
4 | Chacha | 80 | 30 | 700 | 6 | 600
>>> Masukkan id monster: 1
>>> Masukkan stok baru: 3
>>> Masukkan harga baru: 1000
Bulbu telah berhasil diubah dengan stok baru sejumlah 3 dan dengan harga baru 1000!
```

Gambar 7.12.8 Tampilan mengubah stok dan harga item

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): item
ID | Type | Stok | Harga
1 | resilience | 5 | 30
2 | healing | 3 | 20
3 | monster_ball | 0 | 10
4 | strength | 10 | 400
>>> Masukkan id item: 3
>>> Masukkan stok baru: 1000
>>> Masukkan harga baru: 30
monster_ball telah berhasil diubah dengan stok baru sejumlah 1000 dan dengan harga baru 30!
```

Gambar 7.12.9 Tampilan mengubah stok monster saja

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): monster
ID | Type | ATK Power | DEF Power | HP | Stok | Harga
1 | Pikachu | 125 | 10 | 600 | 50 | 0
2 | Zeze | 300 | 10 | 100 | 3 | 1000
3 | Zuko | 100 | 25 | 800 | 8 | 550
4 | Chacha | 80 | 30 | 700 | 6 | 600
>>> Masukkan id monster: 1
>>> Masukkan stok baru: 50
>>> Masukkan harga baru:
Bulbu telah berhasil diubah dengan stok baru sejumlah 50!
```

Gambar 7.12.10 Tampilan mengubah harga monster saja

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1 | Pikachu          | 125 | 10 | 600   | 50 | 0
2 | Zeze             | 300 | 10 | 100   | 3 | 1000
3 | Zuko             | 100 | 25 | 800   | 8 | 550
4 | Chacha           | 80 | 30 | 700   | 6 | 600
>>> Masukkan id monster: 1
>>> Masukkan stok baru:
>>> Masukkan harga baru: 50
Bulbu telah berhasil diubah dengan harga baru 50!
```

Gambar 7.12.11 Tampilan mengubah harga item saja

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): item
ID | Type           | Stok | Harga
1 | strength        | 20 | 40
2 | resilience       | 5 | 30
3 | healing          | 3 | 20
4 | monster_ball    | 100000 | 30
>>> Masukkan id item: 1
>>> Masukkan stok baru:
>>> Masukkan harga baru: 30
strength telah berhasil diubah dengan harga baru 30!
```

Gambar 7.12.12 Tampilan mengubah stok item saja

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): item
ID | Type           | Stok | Harga
1 | strength        | 20 | 30
2 | resilience       | 5 | 30
3 | healing          | 3 | 20
4 | monster_ball    | 100000 | 30
>>> Masukkan id item: 1
>>> Masukkan stok baru: 50
>>> Masukkan harga baru:
strength telah berhasil diubah dengan stok baru sejumlah 50!
```

Gambar 7.12.13 Tampilan error message ketika id monster kosong

```
>>> shop
안녕하세요! Selamat datang kembali, 1
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): monster
ID | Type          | ATK Power | DEF Power | HP   |
5 | Bulbu          | 50 | 50 | 1200
6 | Theo           | 12000000 | 50 | 1200000
7 | ilham          | 10 | 10 | 10
>>> Masukkan id monster:
ID monster harus berupa integer!!
```

Gambar 7.12.14 Tampilan menghapus monster dari shop

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/item): montser
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/item): monster
ID | Type          | ATK Power | DEF Power | HP   | Stok | Harga
1 | Pikachow      | 125 | 10 | 600 | 3 | 1000
2 | Zeze          | 300 | 10 | 100 | 3 | 1000
3 | Zuko          | 100 | 25 | 800 | 8 | 550
4 | Chacha        | 80 | 30 | 700 | 6 | 600
>>> Masukkan id monster: 1
>>> Apakah anda yakin ingin menghapus Pikachow dari shop (y/n)? y
Berhasil menghapus Pikachow dari shop
```

Gambar 7.12.15 Tampilan menghapus item dari shop

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/item): item
ID | Type          | Stok | Harga
1 | resilience    | 5 | 30
2 | healing       | 3 | 20
3 | monster_ball  | 1000 | 30
4 | strength      | 10 | 400
>>> Masukkan id item: 3
>>> Apakah anda yakin ingin menghapus monster_ball dari shop (y/n)? y
Item telah berhasil dihapus dari shop!
```

Gambar 7.12.16 Tampilan keluar dari shop management

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): keluar
Anda keluar dari shop management.
```

F13 - Monster Management

Modul untuk admin yang bertujuan mengubah data monster dalam shop

Gambar 7.13.1 Tampilan Monster Management untuk Admin

```
>>> monster
SELAMAT DATANG DI DATABASE PARA MONSTER !!!
Available actions :
1. Tampilkan semua Monster
2. Tambah Monster baru
3. Keluar dari Monster Management
>>> Pilih aksi : █
```

Gambar 7.13.2 Tampilan database monster

```
>>> Pilih aksi : 1
Loading database monster...
Database Monster
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
5  | Chacha       | 80        | 30        | 700
6  | Theo         | 12000000  | 50        | 1200000
```

Gambar 7.13.3 Tampilan menambah monster

```
>>> Pilih aksi : 2
Memulai pembuatan monster baru
>>> Masukkan Type / Nama : ilham
>>> Masukkan ATK Power : 10
>>> Masukkan DEF Power (0-50) : 10
>>> Masukkan HP : 10
Monster baru berhasil dibuat!
Type : ilham
ATK Power : 10
DEF Power : 10
HP : 10
>>> Tambahkan Monster ke database (Y/N) : Y
Monster baru telah ditambahkan!
Available actions :
1. Tampilkan semua Monster
2. Tambah Monster baru
3. Keluar dari Monster Management
>>> Pilih aksi : █
```

Gambar 7.13.4 Tampilan monster sudah terdaftar

```
>>> Pilih aksi : 2
Memulai pembuatan monster baru
>>> Masukkan Type / Nama : Pikachow
Nama sudah terdaftar, coba lagi!
>>> Masukkan Type / Nama : █
```

Gambar 7.13.5 Tampilan gagal memasukan DEF power monster

```
>>> Masukkan DEF Power (0-50) : 60
DEF Power harus bernilai 0-50, coba lagi!
>>> █
```

Gambar 7.13.6 Tampilan keluar dari Monster Management untuk Admin

```
>>> Pilih aksi : 3
Anda keluar dari monster management.
```

F14 - Load

Modul untuk melakukan load data dari csv menjadi sebuah

Gambar 7.14.1 Tampilan Load

```
$ python main.py save

Loading...
Selamat datang di program OWCA!
Anda belum login, ketik 'HELP' untuk melihat command-command yang available!

>>> █
```

Gambar 7.14.2 Tampilan Load gagal

```
$ python main.py load

Folder load tidak dapat ditemukan!
Daftar folder penyimpanan yang tersedia:
1. .gitkeep
2. default
3. ilham
4. newsave
5. newsavebro
6. newsavefolder
7. save
```

Gambar 7.14.3 Tampilan Load error

```
$ python main.py
usage: main.py [-h] folder
main.py: error: the following arguments are required: folder
```

F15 - Save

Modul untuk menyimpan data ke dalam folder csv

Gambar 7.15.1 Tampilan save data berhasil dibuat

```
>>> save
Masukkan nama folder : newsavefolder
Membuat folder /data/newsavefolder...
Data berhasil disimpan ke folder /data/newsavefolder.
```

Gambar 7.15.2 Tampilan data berhasil disimpan

```
>>> save
Masukkan nama folder : save
Folder /data/save sudah ada.
Data berhasil disimpan ke folder /data/save.
```

F16 - Exit

Modul untuk keluar dari program

Gambar 7.16.1 Tampilan Exit

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

Gambar 7.16.2 Tampilan Exit dan menyimpan data

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) y
Masukkan nama folder : save
Folder /data/save sudah ada.
Data berhasil disimpan ke folder /data/save.
Data telah disimpan. Selamat tinggal!
```

Gambar 7.16.1 Tampilan Exit dan tidak menyimpan data

```
>>> exit
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) n
Selamat tinggal!
```

B03 - Monster Ball

Modul untuk menambahkan perintah dalam Battle berupa menangkap monster

Gambar 7.7.4 Tampilan item monster ball user

```
Masukkan pilihan sesuai angka diatas atau 'KELUAR'  
>> 7  
Monster Ball  
Quantity : 9999999999996
```

Gambar 7.8.5 Tampilan monster tidak bisa ditangkap

```
Pilih perintah: 3  
  
Anda sudah memiliki monster Zeze dalam inventory!
```

Gambar 7.8.6 Tampilan monster gagal ditangkap

```
Pilih perintah: 3  
  
Swoosshhhhh, Anda mengeluarkan Monster Ball !!!  
Yahhh, Anda belum berhasil mendapatkan monster Pikachow !!!  
  
Sisa Monster Ball Anda: 9999999999995  
  
===== TURN 1 (Bulbu) =====
```

Gambar 7.8.7 Tampilan monster berhasil ditangkap

```
Swoosshhhhh, Anda mengeluarkan Monster Ball !!!  
Selamat, Anda berhasil mendapatkan monster Pikachow !!!  
  
Name      : Pikachow  
ATK Power : 125  
DEF Power : 10  
HP        : 600  
Level     : 1  
  
Sisa Monster Ball Anda: 9999999999994
```

Gambar 7.10.2 Tampilan aksi melihat monster dalam Shop

```
Pilih aksi (lihat/beli/keluar): lihat
Mau lihat apa? (monster/item): monster
```

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	10	500
2	Zeze	300	10	100	3	1000
3	Zuko	100	25	800	8	550
4	Chacha	80	30	700	6	600

LAMPIRAN

Nomor Asistensi : 01
No. Kelompok/Kelas : F/ K07
Tanggal asistensi : 02 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16523207/ Ramadhan Abhinawa Herawanto
	2	16523057/ Kennard Hezekiah Montoya
	3	19623057/ Muhammad Fithra Rizki
	4	19623077/ Theo Kurniady
	5	19623097/ Gabriela Jennifer Sandy
	6	
Asisten pembimbing	NIM / Nama	
	18221066/ Aulia Nadhirah	

Catatan Asistensi:

Rangkuman Diskusi
<ol style="list-style-type: none">1. Pembahasan spesifikasi tugas besar2. File CSV untuk register bertambah ketika sudah di <i>save</i>.3. File inventory memanggil 2 file CSV (item dan monster).4. CSV hanya bisa dibaca ketika di <i>load</i> dan di <i>write</i> ketika sudah di <i>save</i>.5. Admin memiliki akses lebih dalam item di permainan (melihat, menambah, mengubah dan menghapus serta tidak melakukan apa-apa).6. Saran pembagian tugas.
Tindak Lanjut
<ol style="list-style-type: none">1. Semua perubahan disimpan ke variabel arr/dict yang sudah ada setelah itu disimpan.2. Tips inventory, shop, monster, dll diambil dari CSV hanya sekali setiap kali program dijalankan (load saja) dan tidak setiap kali pengguna membuka shop/inventory.3. Dilarang mengimport CSV (CSV dibaca menggunakan open dan for loop).
Dokumentasi

Nomor Asistensi : 02
No. Kelompok/Kelas : F/K07
Tanggal asistensi : 11 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	16523207/ Ramadhan Abhinawa Herawanto
	2	16523057/ Kennard Hezekiah Montoya
	3	19623057/ Muhammad Fithra Rizki
	4	19623077/ Theo Kurniady
	5	19623097/ Gabriela Jennifer Sandy
Asisten pembimbing	6	
	NIM / Nama	
	18221066/ Aulia Nadhirah	

Catatan Asistensi:

Rangkuman Diskusi
<ol style="list-style-type: none">1. Pengecekan pembagian tugas2. Pengecekan progress per individu (progress report)3. Demonstrasi kode4. Pertanyaan (QnA)
Tindak Lanjut
<ol style="list-style-type: none">1. Mengerjakan program utama terlebih dahulu karena merupakan nilai utama.2. Kerjakan bonus apabila dinilai bisa mengerjakan bonus bersama program utama.3. Penamaan variabel dan comment mudah dimengerti dan jelas.4. Menghilangkan data.py dari csv.py dan data.py5. Save ke CSV saat mengesave6. Programnya mudah dibaca, modular bisa dipakai kemana-mana, dan git best practice agar commitnya sesuai.
Dokumentasi

