

LAPORAN TUGAS BESAR

MATA KULIAH IF210 DASAR PEMROGRAMAN

PROGRAM GAME OWCA MELAWAN DR. ASEP SPAKBOR

Dosen Pengampu: Robithoh Annur, S.T., M.Eng., Ph.D.

Kelas/Kelompok: 11/D



Disusun oleh:

| | |
|--------------------------|------------|
| Nadhif Al Rozin | (19623300) |
| Nazwan Siddqi Muttaqin | (19623200) |
| Abidah Izdiyar Dzakiyyah | (16523130) |
| Hanif Kalyana Aditya | (19623120) |
| Hafizh Hananta Akbari | (13522132) |

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

2024

HALAMAN PERNYATAAN

“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.”

Anggota kelompok:

1. Nadhif Al Rozin (19623300)
2. Nazwan Siddqi Muttaqin (19623200)
3. Abidah Izdihar Dzakiyyah (19623130)
4. Hanif Kalyana Aditya (19623120)
5. Hafizh Hananta Akbari (13522132)

DAFTAR ISI

| | |
|---------------------------|---|
| LAPORAN TUGAS BESAR | 1 |
|---------------------------|---|

| | |
|--|----|
| MATA KULIAH IF210 DASAR PEMROGRAMAN..... | 1 |
| PROGRAM GAME OWCA MELAWAN DR. ASEP SPAKBOR..... | 1 |
| PAGEREF _Toc167093898 \h | 1 |
| HALAMAN PERNYATAAN..... | 2 |
| DAFTAR ISI | 2 |
| DAFTAR TABEL | 4 |
| DAFTAR GAMBAR..... | 6 |
| LAPORAN TUGAS BESAR | 7 |
| Deskripsi Persoalan | 7 |
| Daftar Pembagian Kerja | 7 |
| 1. Pembagian Kerja Program | 7 |
| 2. Pembagian Kerja Pembuatan Laporan..... | 8 |
| Checklist Hasil Rancangan | 8 |
| Desain Command..... | 9 |
| Desain Dekomposisi, Kamus Data, Spesifikasi..... | 15 |
| 1. Fungsi Main | 15 |
| 2. F00 - Random Number Generator | 17 |
| 3. F01 – Register | 18 |
| 4. F02 – Login..... | 21 |
| 5. F03 – Logout..... | 22 |
| 6. F04 - Menu & Help..... | 23 |
| 7. F05 – Monster | 25 |
| 8. F06 – Potion | 27 |
| 9. F07 – Inventory | 28 |
| 10. F08 – Battle..... | 31 |
| 11. F09 – Arena..... | 36 |
| 12. F10 - Shop & Currency..... | 41 |
| 13. F11 – Laboratory..... | 44 |
| 14. F12 - Shop Management..... | 46 |
| 15. F13 - Monster Management..... | 50 |
| 16. F14 – Load | 53 |
| 17. F15 – Save..... | 56 |
| 18. F16 – Exit..... | 59 |
| 19. B02 – Typing | 59 |
| 20. B03 - Monster Ball..... | 63 |
| 21. B04 – Jackpot..... | 64 |
| 22. B05 - Peta Kota Daville | 66 |
| Hasil Pengujian Program | 71 |

| | |
|--|----|
| 1. F00 - Random Number Generator | 71 |
| 2. F01 – Register | 71 |
| 3. F02 – Login..... | 72 |
| 4. F03 – Logout..... | 73 |
| 5. F04 - Menu & Help..... | 73 |
| 6. F05 – Monster | 73 |
| 7. F06 – Potion | 74 |
| 8. F07 – Inventory | 74 |
| 9. F08 – Battle | 76 |
| 10. F09 – Arena..... | 79 |
| 11. F10 - Shop & Currency..... | 82 |
| 12. F11 – Laboratory..... | 82 |
| 13. F12 - Shop Management..... | 83 |
| 14. F13 - Monster Management..... | 86 |
| 15. F14 – Load | 87 |
| 16. F15 – Save..... | 87 |
| 17. F16 – Exit..... | 88 |
| 18. B03 - Monster Ball..... | 88 |
| 19. B04 – Jackpot..... | 88 |
| 20. B05 - Peta Kota Daville | 90 |
| LAMPIRAN | 91 |

DAFTAR TABEL

| | |
|--|---|
| Tabel 1. Tabel Pembagian Kerja Program..... | 7 |
| Tabel 2. Tabel Pembagian Kerja Pembuatan Laporan | 8 |

| | |
|---|---|
| Tabel 3. Tabel Checklist Hasil Rancangan..... | 8 |
|---|---|

| | |
|---|-----------|
| <i>Gambar 1. Hasil Pindaian Asistensi 1</i> | <i>91</i> |
| <i>Gambar 2. Hasil Pindaian Asistensi 2</i> | <i>93</i> |

| | |
|---|-----------|
| <i>Gambar 1. Hasil Pindaian Asistensi 1</i> | <i>91</i> |
| <i>Gambar 2. Hasil Pindaian Asistensi 2</i> | <i>93</i> |

DAFTAR GAMBAR

LAPORAN TUGAS BESAR

Deskripsi Persoalan

Program ini merupakan sebuah permainan dengan tujuan utama mengalahkan monster-monster yang dibuat oleh Dr. Asep Spakbor. Pada program ini, Agen Purry akan berusaha mengalahkan Dr. Asep Spakbor menggunakan monster yang sudah dilatih menggunakan fitur-fitur pada program.

Pada program ini terdapat 16 fungsi wajib serta 3 buah fungsi bonus yang kami kerjakan. Fungsi bonus dibuat untuk membantu program agar lebih baik dijalankan. Pada proses pembuatan program, kami menggunakan bahasa pemrograman Python 3.9+ dan dibatasi oleh beberapa fungsi seperti yang tertera pada spesifikasi tugas besar.

Daftar Pembagian Kerja

1. Pembagian Kerja Program

| Fitur | Implementasi | NIM Desainer | NIM Coder | NIM Tester |
|--------------------------|---------------------|-----------------|--------------|---------------|
| F00 - RNG | random() | 19623300 | 19623300 | 16523130 |
| F01 - Register | register() | 13522132 | 13522132 | 16523130 |
| F02 - Login | login() | 16523130 | 16523130 | 16523130 |
| F03 - Logout | logout() | 16523130 | 16523130 | 16523130 |
| F04 - Menu and Help | menuHelp() | 16523130 | 16523130 | 16523130 |
| F05 - Monster | monster() | 19623300 | 19623300 | 16523130 |
| F06 - Potion | potion() | 19623200 | 19623200 | 19623200 |
| F07 - Inventory | inventory() | 19623300 | 19623300 | 19623300 |
| F08 - Battle | battle() | 19623300 | 19623300 | 16523130 |
| F09 - Arena | arena() | 19623300 | 19623300 | 19623300 |
| F10 - Shop and Currency | shop() | 19623200 | 19623200 | 19623200 |
| F11 - Laboratory | laboratory() | 19623300 | 19623300 | 19623300 |
| F12 - Shop Management | shop_management() | 19623200 | 19623200 | 19623200 |
| F13 - Monster Management | monsterManagement() | 19623300 | 19623300 | 19623300 |
| F14 - Load | load() | 19623120 | 19623120 | 19623120 |
| F15 - Save | save() | 19623120 | 19623120 | 19623120 |
| F16 - Exit | exit() | 19623120 | 19623120 | 19623120 |
| B03 - Monster Ball | catch() | 19623300 | 19623300 | 19623300 |
| B04 - Jackpot | jackpot() | 19623300 | 19623300 | 19623300 |
| B05 - Peta Kota | peta() | 19623300 | 19623300 | 19623300 |

Tabel 1. Tabel Pembagian Kerja Program

2. Pembagian Kerja Pembuatan Laporan

Tabel 2. Tabel Pembagian Kerja Pembuatan Laporan

| Bagian | NIM |
|---|--|
| Halaman Cover | 16523130 |
| Daftar Isi | 16523130 |
| Daftar Tabel | 16523130 19623200 |
| Daftar Gambar | 16523130 |
| Deskripsi Persoalan | 16523130 |
| Daftar Pembagian Tugas | 16523130 |
| Checklist Hasil Rangkaian, Implementasi, dan Testing | 16523130 |
| Desain Command | 16523130 19623200 |
| Desain Dekomposisi Algoritmik, Fungsional Program, Desain Kamus Data, Spesifikasi | 19623120 19623200 19623300 16523130 13522132 |
| Hasil Pengujian Program | 19623300 |
| Lampiran | 19623120 |

Checklist Hasil Rancangan

Tabel 3. Tabel Checklist Hasil Rancangan

| Fitur | Desain | Implementasi | Testing |
|-------------------------|--------|--------------|---------|
| F00 - RNG | V | V | V |
| F01 - Register | V | V | V |
| F02 - Login | V | V | V |
| F03 - Logout | V | V | V |
| F04 - Menu and Help | V | V | V |
| F05 - Monster | V | V | V |
| F06 - Potion | V | V | V |
| F07 - Inventory | V | V | V |
| F08 - Battle | V | V | V |
| F09 - Arena | V | V | V |
| F10 - Shop and Currency | V | V | V |
| F11 - Laboratory | V | V | V |

| | | | |
|--------------------------|---|---|---|
| F12 - Shop Management | V | V | V |
| F13 - Monster Management | V | V | V |
| F14 - Load | V | V | V |
| F15 - Save | V | V | V |
| F16 - Exit | V | V | V |
| B03 - Monster Ball | V | V | V |
| B04 - Jackpot | V | V | V |
| B05 - Peta Kota | V | V | V |

Desain Command

>>> REGISTER

procedure register_manager ()

{ I. S melakukan registrasi dengan menginput username yang belum pernah terdaftar juga memenuhi ketentuan username, password, dan pilihan monster yang ingin digunakan }

{ F. S program mengeluarkan pesan sesuai inputan yang dimasukkan, masukkan username yang memenuhi ketentuan dan tidak terdaftar pada list akan mengeluarkan pilihan monster dan pesan yang menandakan bahwa registrasi berhasil, masukkan username yang tidak memenuhi ketentuan akan mengeluarkan pesan yang menandakan bahwa username tidak terdaftar }

>>> LOGIN

function login ()

{ I. S melakukan login dengan menginput username yang terdaftar dan password yang sesuai }

{ F. S program mengeluarkan pesan sesuai masukkan, masukkan username yang terdaftar dan password yang benar akan mengeluarkan pesan yang menandakan login berhasil dan pesan untuk memasukkan command help, masukkan username yang tidak terdaftar, password yang tidak sesuai, maupun username sudah ter-login akan mengeluarkan pesan bahwa login gagal }

>>> LOGOUT

function logout ()

{I. S melakukan logout dengan menghapus username dari list username yang ter-login}

{F. S program logout berhasil jika username berada dalam list ter-login, program akan mengeluarkan pesan yang menandakan logout gagal}

>>> HELP

function menuHelp ()

{I. S menerima input dalam bentuk integer berupa pilihan menu program}

{F. S Program membuka fungsi yang telah dipilih. Program help dengan kondisi belum login hanya akan memberikan pilihan menu yang terbatas.}

>>> MONSTER

procedure monster ()

{I. S mengakses database monster yang dimiliki dan mengelola perubahan yang terjadi pada data monster}

{F. S Apabila monster melakukan serangan, nilai ATK power yang dimilikinya diambil secara acak dalam rentang $\pm 30\%$ dari ATK power. Apabila monster diserang, DEF Power (kekuatan pertahanan) akan mengurangi serangan musuh dengan perhitungan serangan musuh akan berkurang sesuai jumlah DEF Power yang dimiliki}

>>> POTION

procedure potion ()

{I. S mengelola database monster berdasarkan potion yang tersedia}

{F. S Program memiliki tiga buah potion, strenght potion akan meningkatkan ATK Power sebanyak 5%, Resillience Potion akan meningkatkan DEF Power sebanyak 5%, dan Healing Potion akan mengisi HP sebanyak 25 % dan tidak akan melebihi batas maksimal HP}

>>> INVENTORY

function inventory ()

{I. S menampilkan inventory list dari user dan menerima input integer dari data yang ingin di buka detailnya}

{F. S menampilkan detail item dari inventory list yang dipilih user}

>>> BATTLE

function battle ()

{I. S menampilkan monster lawan dan monster list yang dapat di pilih, dengan monster yang telah dipilih agent, masukkan pada program ini adalah integer berupa nomor perintah yang ingin dilakukan dalam pertarungan}

{F. S Program memulai pertarungan dan memproses pilihan perintah user dengan menampilkan detail monster setelah bertarung (opsi attack), list potion yang dimiliki (opsi potion), dan tampilan user yang meninggalkan battle (opsi quit). Setelah agent menyelesaikan battle, akan ditampilkan jumlah OC yang didapatkan. }

>>> ARENA

procedure arena ()

{I. S Agent memulai sesi di Arena dan merepresentasikan level monster yang dilawan. Artinya, agent akan melawan monster level 1 di stage 1, monster level 2 di stage 2, dst. Health dari agent dan monster dipersiapkan untuk pertarungan. Stage pertama dipersiapkan dengan monster level 1 terpilih secara acak dari database. }

{F. S Agent berhasil memenangkan kelima stage atau terjadi game over. Jika agent berhasil memenangkan kelima stage, hasil statistik sesi latihan ditampilkan pada scoreboard, termasuk total hadiah yang diterima, berhasil menang hingga stage berapa, total damage yang diberikan, dan total damage yang diterima. Jika terjadi game over (misalnya, monster milik agent mati), sesi latihan berakhir, dan status game over ditampilkan pada scoreboard.

>>> SHOP AND CURRENCY

procedure shop()

{I. S Agent dapat mengunjungi shop dengan command SHOP, setelah itu dapat memilih melihat item, membeli item, atau keluar dari shop}

{F. S Jika Agent memilih melihat item, program akan menampilkan daftar item yang dijual di shop (monster dan potion) beserta detail itemnya. Jika Agent

memilih membeli item, program akan meminta Agent memasukkan tipe item (monster/potion) dan id item yang ingin dibeli, jika membeli potion program juga akan meminta jumlah yang ingin dibeli. Program akan melakukan validasi seperti ketersediaan OC, stok item di shop, dan jika membeli monster akan divalidasi apakah monster tersebut sudah dimiliki di inventory. Jika validasi berhasil, item akan masuk ke inventory Agent (monster level 1), jika tidak program akan menampilkan pesan gagal }

>>> LABORATORY

procedure laboratory()

{I. S Agent dapat melakukan upgrade level monster yang dimiliki di inventory dengan command LABORATORY }

{F. S Program akan menampilkan daftar monster yang dimiliki Agent beserta levelnya. Agent memilih monster yang ingin di-upgrade. Jika level monster kurang dari 5, program akan menampilkan jumlah OC yang dibutuhkan untuk melakukan upgrade level. Jika level monster sudah 5, program akan menampilkan pesan bahwa monster telah mencapai level maksimum. Program akan meminta konfirmasi dari Agent apakah yakin ingin melakukan upgrade. Jika Agent mengonfirmasi dan memiliki OC yang cukup, level monster akan ter-upgrade, jika tidak program akan menampilkan pesan kegagalan upgrade }

>>> SHOP MANAGEMENT

procedure shop_management ()

{I. S Admin dapat melakukan pengelolaan item (monster dan potion) yang dijual di shop, seperti menambah, mengubah, dan menghapus item dari database shop }

{F. S Jika Admin memilih menambah item, program akan meminta Admin memasukkan detail item baru seperti tipe (monster/potion), nama, deskripsi, harga, dan stok awal. Jika Admin memilih mengubah item, program akan menampilkan daftar item yang ada di database shop, kemudian meminta Admin memilih id item yang ingin diubah dan mengubah detail item seperti harga dan stok. Jika Admin memilih menghapus item, program akan menampilkan daftar item dan meminta Admin memilih id item yang ingin dihapus dari database shop. Program akan melakukan validasi seperti keunikan nama item dan ketersediaan

stok saat penambahan/pengubahan. Jika validasi berhasil, database shop akan diperbarui, jika tidak program akan menampilkan pesan gagal}

>>> MONSTER MANAGEMENT

procedure monsterManagement()

{I. S Admin dapat mengelola monster di database}

{F. S Program akan menampilkan menu dengan opsi: 1) Lihat daftar monster, 2) Tambah monster baru, 3) Ubah monster, 4) Hapus monster. Pada opsi 1, program akan menampilkan semua monster di database beserta detailnya. Pada opsi 2, program akan meminta input detail monster baru seperti nama, ATK, DEF, HP, dan melakukan validasi input sebelum menambahkan ke database setelah konfirmasi. Pada opsi 3, program akan menampilkan daftar monster, meminta pilih id monster yang diubah, kemudian meminta input detail baru dengan validasi. Pada opsi 4, program akan menampilkan daftar monster, meminta pilih id monster yang dihapus, dan mengonfirmasi sebelum menghapus dari database}

>>> LOAD

procedure load ()

{I. S Program dapat memuat data game dari folder eksternal yang berisi file-file penyimpanan dengan struktur data tertentu}

{F. S Program akan menerima input nama folder dari pengguna saat dijalankan. Program akan melakukan validasi apakah folder tersebut ada atau tidak. Jika folder tidak ada, program akan menampilkan pesan kesalahan dan keluar. Jika folder ada, program akan membaca file-file penyimpanan di dalam folder tersebut sesuai dengan struktur data eksternal yang telah didefinisikan. Program akan memuat data dari file-file tersebut ke dalam struktur data internal program, seperti data monster, potion, inventory Agent, dan lain-lain. Setelah data berhasil dimuat, program akan menampilkan pesan selamat datang dan meminta perintah selanjutnya dari pengguna}

>>> SAVE

procedure save ()

{I. S Program dapat menyimpan data game ke dalam folder eksternal berupa file-file penyimpanan dengan struktur data tertentu}

{F. S Program akan meminta nama folder untuk menyimpan data game dan memeriksa keberadaannya. Jika folder tidak ada, program akan membuatnya dan menampilkan pesan bahwa folder baru telah dibuat serta data berhasil disimpan. Jika folder ada namun kosong, program akan membuat file baru di dalamnya dan menampilkan pesan bahwa data berhasil disimpan. Jika folder dan file sudah ada, program akan mengganti file lama dengan data baru tanpa pesan tambahan. Data game akan disimpan dari struktur data internal ke dalam file sesuai dengan struktur data eksternal yang telah ditentukan. }

>>> EXIT

procedure exit ()

{I. S Program dapat keluar setelah menyimpan atau tidak menyimpan data}

{F. S Program akan meminta konfirmasi kepada pengguna untuk menyimpan data sebelum keluar. Jika pengguna memilih 'y', program akan menjalankan fungsi save lalu keluar. Jika pengguna memilih 'n', program akan langsung keluar tanpa menyimpan. Jika input selain 'y' atau 'n', program akan meminta input valid hingga benar}

>>> JACKPOT

procedure jackpot ()

{I. S Agent dapat memainkan permainan jackpot untuk mendapatkan item atau monster menggunakan OC (O.W.C.A Coin)}

{F. S Program akan menampilkan daftar item dengan nilai OC masing-masing dan meminta konfirmasi dari Agent untuk memulai permainan. Jika Agent memilih untuk bermain, program akan memeriksa OC yang dimiliki. Jika tidak cukup, program akan menampilkan pesan. Jika cukup, program akan mengurangi OC sesuai biaya permainan, mengacak 3 item, dan menampilkan hasilnya. Jika ketiga item sama (jackpot), Agent mendapatkan monster acak yang ditambahkan ke inventory. Jika tidak jackpot, nilai OC dari ketiga item dijumlahkan dan ditambahkan ke OC Agent. Program akan menampilkan hasil permainan dan meminta konfirmasi apakah Agent ingin bermain lagi. }

Desain Dekomposisi, Kamus Data, Spesifikasi

1. Fungsi Main

a. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}
kondisi, userId : integer
isExit, isLogout, changePotionInvent : boolean
numRange : array of integer
ItemList : array of string
LokasiUser : tuple of integer
dataSemuaUser, dataMonster, dataMonsterShop, dataItemShop,
dataUser, monsterUser, potionInvent, ballUser : dictionary

b. Desain Spesifikasi

```
from src.load import load, loadInvent, loadData, showDict
from src.battle import battle
from src.arena import arena
from src.peta import peta
from src.jackpot import jackpot
from src.lab import laboratory
from src.potion import getPotion, updatePotionUser, updatePotionInvent
from src.share import sleep, display
from src.inventory import inventory
from src.shopMng import shop_management
from src.monsterMng import monsterManagement
from src.shop import shop
from src.save import save, updateUser, updateItem, updateMonster
from src.exit import exit
from src.menuHelp import menu
```

ALGORITMA

```
{Pertama masuk}
kondisi <- 1
dataUser <- {
    "ID": "",
    "Username": "",
    "Password": "",
    "Role": "",
    "OC": ""
}

[dataSemuaUser, dataMonster, dataMonsterShop, dataItemShop] <- load()
[kondisi, isExit, isLogout, dataUser] <- menu(dataUser)
userId <- dataUser["ID"]
[monsterUser, potionInvent, ballUser] <- loadInvent(userId)
potionUser <- getPotion(userId)
itemList <- [itemType for itemType in dataItemShop["Type"]]

{main loop}
while kondisi != 0 do
    if isExit and kondisi = 0 then {memilih exit dan tidak save}
        break
    if dataUser["Role"] = "admin" then
        if kondisi = 1 then
            shop_management(dataMonsterShop, dataItemShop, dataMonster,
            itemList)
        else if kondisi = 2 then
```

```

        itemInvent <- updateItem(potionInvent, ballUser, userId)
        monsterInvent <- updateMonster(monsterUser, userId)
        updateUser(dataUser, dataSemuaUser)
        save(dataSemuaUser, dataMonster, dataMonsterShop,
monsterInvent, dataItemShop, itemInvent)
        if isExit then
            break
        if isLogout then
            column transversal dataUser
            dataUser[column] <- ""
            [dataSemuaUser, dataMonster, dataMonsterShop, dataItemShop]
<- load()
            userId <- dataUser["ID"]
            [monsterUser, potionInvent, ballUser] <- loadInvent(userId)
            potionUser <- getPotion(userId)
            itemList <- [itemType for itemType in dataItemShop["Type"]]
        else if kondisi = 3 then
            monsterManagement(dataMonster)
    else
        userId <- dataUser["ID"]
        if kondisi = 1 then
            lokasiUser <- (0,0)
            agent(lokasiUser, dataUser, dataMonster, dataMonsterShop,
dataItemShop, potionUser, monsterUser, ballUser, potionInvent)
        else if kondisi = 2 then
            if dataUser["ID"] = "" then
                output("Anda belum login, tidak ada data yang bisa
disimpan")
                break
            else
                itemInvent <- updateItem(potionInvent, ballUser, userId)
                monsterInvent <- updateMonster(monsterUser, userId)
                updateUser(dataUser, dataSemuaUser)
                save(dataSemuaUser, dataMonster, dataMonsterShop,
monsterInvent, dataItemShop, itemInvent)
                if isExit then
                    break
                if isLogout then
                    column transversal dataUser
                    dataUser[column] <- ""
                    [dataSemuaUser, dataMonster, dataMonsterShop,
dataItemShop] <- load()
                    userId <- dataUser["ID"]
                    [monsterUser, potionInvent, ballUser] <-
loadInvent(userId)
                    potionUser <- getPotion(userId)
                    [kondisi, isExit, isLogout, dataUser] <- menu(dataUser)
exit()

```

```

from src.load import load, loadInvent, loadData, showDict
from src.battle import battle
from src.arena import arena
from src.peta import peta
from src.jackpot import jackpot
from src.lab import laboratory
from src.potion import getPotion, updatePotionUser, updatePotionInvent
from src.share import sleep, display
from src.inventory import inventory
from src.shopMng import shop_management
from src.monsterMng import monsterManagement

```



```

from src.shop import shop
from src.save import save, updateUser, updateItem, updateMonster
from src.exit import exit
from src.menuHelp import menu

procedure agent(lokasidUser, dataUser, dataMonster, dataMonsterShop,
dataItemShop, potionUser, monsterUser, ballUser, potionInvent)
    changePotionInvent <- True
    while True do
        if changePotionInvent then
            updatePotionInvent(potionUser, potionInvent)
            updatePotionUser(potionUser, potionInvent)
            [aksi, lokasidUser] <- peta(lokasi=lokasidUser)
            if aksi = "M" then
                break
            else if aksi = "B" then
                changePotionInvent <- True
                isMenang <- battle(dataUser, potionUser, monsterUser,
ballUser)
                if not isMenang then
                    output("Sedang respawn...")
                    sleep()
                    lokasidUser <- (0,0)
                else if aksi = "AM" then
                    changePotionInvent <- True
                    display("Anda terkena serangan tiba tiba, bersiaplah !!!")
                    sleep()
                    isMenang <- battle(dataUser, potionUser, monsterUser,
ballUser, ambush=True)
                    if not isMenang then
                        output("Sedang respawn...")
                        sleep()
                        lokasidUser <- (0,0)
                else if aksi = "LA" then
                    laboratory(userId, dataUser, monsterUser)
                else if aksi = "I" then
                    inventory(dataUser, monsterUser, potionUser, ballUser)
                else if aksi = "J" then
                    jackpot(dataUser, dataMonster, monsterUser)
                else if aksi = "A" then
                    changePotionInvent <- True
                    arena(dataUser, potionUser, monsterUser)
                else if aksi = "S" then
                    shop(dataUser, dataMonsterShop, dataItemShop, potionInvent,
dataMonster, monsterUser, ballUser)
                    changePotionInvent <- False

```

2. F00 - Random Number Generator

a. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}

a,c,m,n : integer

x0,seed : float

numRange : array of integer

b. Desain Spesifikasi

```

import os
import time

```

```
function random(a=48271, c=0, m=2**31-1,n=None, seed=None, numRange=None)
-> integer: {C++11's minstd_rand}
```

ALGORITMA

```
if seed is None then
  x0 <- int(os.getpid() + time.time())
else
  x0 <- seed
if n is None then
  n <- random(n=1, numRange=[2,10])
repeat n times
  x0 <- (a * x0 + c) % m
if numRange is None then
  return x0
else
  hasil <- int((x0 / (m - 1)) * (numRange[1] - numRange[0]) +
numRange[0])
  if hasil = numRange[1] then
    hasil <- hasil - 1
  return hasil
```

3. F01 – Register

a. Desain Kamus

```
KAMUS LOKAL {Deklarasi variabel}:
userFile, monsterFile: string
inputName, inputPass: string
lastID: integer
inputID, monInPlayerID, inputRole, inputCoin, inputMonster:
string
monster_types: array of string
row: array of char
is_name: boolean
i, x: integer
monInMonsterID, monInLevelID: integer
userData, monInventoryData: array of string
```

b. Desain Spesifikasi

```
function concatenate_list(lst, delimiter) -> string
  result <- ""
  item traversal [lst]
    result <- result + str(item) + delimiter
  return result
```

```
function append_to_row(row, list)
  if IsList(item) then
    i traversal [len(item)]
      row.append(item[i])
  else
    row.append(item)
```

```
function write_to_csv(csvFile, data)
```

```

with open(csvFile, mode='a', newline='') as file
    row <- ""
    item traversal [data]
    row <- row + str(item) + ";"
    row <- row[:-1] + '\n'
    file.write(row)

```

```

function check_valid_input(username: string, csvFile: string) -> boolean
    char traversal [username]
    if char not in allowed_characters then
        output("Username hanya boleh berisi alfabet, angka,
underscore, dan strip!")
        return False

    with open(csvFile, mode='r') as file
        line traversal (file)
        row <- []
        is_name <- False
        char traversal [line]
        if char = ';' then
            if is_name == True then
                break
            is_name <- True
        else if is_name = False then
            append_to_row(row, [char])
        if username = concatenate_list(row, '') then
            output("Username ", username, " sudah terpakai, silahkan
gunakan username lain!")
            return False

    return True

```

KAMUS LOKAL {Deklarasi variabel}:

```

userFile, monsterFile: string
inputName, inputPass: string
lastID: integer
inputID, monInPlayerID, inputRole, inputCoin, inputMonster: string
monster_types: array of string
row: array of char
is_name: boolean
i, x: integer
monInMonsterID, monInLevelID: integer
userData, monInventoryData: array of string

```

ALGORITMA

```

function register_manager() -> array of string
    userFile <- 'data\user.csv'
    monsterFile <- 'data\monster.csv'

    while True do
        inputName <- input("Masukkan nama: ")
        inputPass <- input("Masukkan password: ")

```

```

    if not check_valid_input(inputName, userFile) then
        sleep(2)
        continue

lastID <- 0
with open(userFile, mode='r') as file
    next(file)
    line traversal [file]
    row <- []
    char traversal [line]
        if char = ';' then
            break
        append_to_row(row, [char])
    lastID <- max(lastID, int(concatenate_list(row, '')))

inputID <- str(lastID + 1)
monInPlayerID <- inputID

inputRole <- 'agent'
inputCoin <- '0'

monster_types <- []
with open(userFile, mode='r') as file
    next(file)
    line traversal [file]
    row <- []
    is_name <- False
    char traversal [line]
        if char = ';' then
            if is_name = False then
                break
            is_name <- True
        else if is_name = False then
            append_to_row(row, [char])
    monster_types.append(concatenate_list(row, ''))
break

while True do
    output("\nSilahkan pilih salah satu monster sebagai monster
awalmu.")
    i, monster_types traversal [enumerate(monster_types, start=1)]
        output(i, ". ", monster_type)

    x <- int(input("\nMonster pilihanmu: "))
    if x <= 0 or x > len(monster_types) maka
        output("\nPilihan monster tidak valid.")
        sleep(2)
        continue

    monInMonsterID <- x
    monInLevelID <- 1

    inputMonster <- monster_types[x - 1]

```

```

        output("\\nSelamat datang Agent ", inputName, ". Mari kita
mengalahkan Dr. Asep Spakbor dengan ", inputMonster, "!")
        sleep(2)

        userData <- [inputID, inputName, inputPass, inputRole,
inputCoin]
        monInventoryData <- [monInPlayerID, monInMonsterID,
monInLevelID]

        write_to_csv("data\\monster_inventory.csv", monInventoryData)

        if userData != [] then
            write_to_csv("data\\user.csv", userData)

        return userData

```

4. F02 – Login

a. Desain Kamus

```

KAMUS LOKAL {Deklarasi variabel}
    username, password : string
    isExit : boolean
    dataUser : dictionary

```

b. Desain Spesifikasi

```

from src.load import getDataUser, loadData
from src.share import YesOrNo, clear, sleep

```

```

KAMUS LOKAL {Deklarasi variabel}
    username, password : string
    isExit : boolean
    dataUser : dictionary

```

ALGORITMA

```

function login(dataUser) -> dictionary
    dataSemuaUser <- loadData("user")
    while True do
        clear()
        if isLogin(dataUser) then
            output("Anda berhasil login dengan username",
dataUser["Username"])
            sleep(3)
            return dataUser
        userName <- ("Masukan username: ")
        input <- userName
        if userName in dataSemuaUser["Username"] then
            for [i, _] in enumerate(dataSemuaUser["ID"]):
                if dataSemuaUser["Username"][i] = userName:
                    password <- ("Masukan password: ")
                    input <- password
                    if dataSemuaUser["Password"][i] = password then

```

```

        userId <- i+1
        dataUser <- getDataUser(userId)
        break
    else
        output("Password salah!")
        sleep(2)
        continue
else
    output("Username tidak ditemukan")
    isExit <- YesOrNo(input("Keluar (Y/N): "))
    if isExit then
        return dataUser

function isLogin(dataUser):
    return not(dataUser["ID"] == "")

```

5. F03 – Logout

a. Desain Kamus

```

KAMUS LOKAL {Deklarasi variabel}
    logout : Boolean
    dataUser : dictionary
    save : boolean

```

b. Desain Spesifikasi

```

from src.login import isLogin
from src.share import YesOrNo

KAMUS LOKAL {Deklarasi variabel}
    logout : Boolean
    dataUser : dictionary
    save : boolean

function logout(dataUser:dict) -> Boolean
    if not isLogin(dataUser) then
        output("Logout gagal!")
        output("Anda belum login sebagai user")
    else

        save <- YesOrNo(input("Apa anda ingin menyimpan file sebelum
logout ? (Y/N): "))
        output("Logout berhasil")
        if save then
            return True
        else
            column traversal [dataUser]
            dataUser[column] = ""
            output("Anda tidak melakukan penyimpanan file. Perubahan
tidak tersimpan.")
            return False

```

6. F04 - Menu & Help

a. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}
kondisi, choice : integer
role : string
isExit, isLogout, isSave : boolean
dataUser : dictionary

b. Desain Spesifikasi

```
from src.login import login, isLogin
from src.logout import logout
from src.register import register_manager
from src.share import pilihanValid, sleep, displayBar, clear, YesOrNo

ALGORITMA
# MENU AND HELP
# sebelum login
procedure beforeLogin()
    output("===== HELP =====")
    output("Kamu belum login sebagai role apapun. Silahkan login
    terlebih dahulu.")
    output("          1. Login   : Masuk ke dalam akun yang sudah
    terdaftar")
    output("          2. Register: Membuat akun baru")
    output("          3. Exit    : Keluar dari game")
    output("")
    output("Footnote:")
    output("          1. Untuk menggunakan aplikasi, silahkan masukkan
    nama fungsi yang terdaftar")
    output("          2. Jangan lupa untuk memasukkan input yang valid")

function afterLogin(role, dataUser)
    if role = 'agent' then
        displayBar("MAIN MENU")
        output(f""""Halo Agent {dataUser["Username"]}. Kamu memasuki
        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. Bermain  : Memasuki peta kota
        3. Save     : Menyimpan data saat ini
        4. Exit     : Keluar dari game""")
        choice <- pilihanValid(input("<///> Pilih perintah (1/2/3/4) "),
        ["1", "2", "3", "4"])
        isLogout <- False
        if choice = "1" then
            isSave <- logout(dataUser)
            isLogout <- True
            isExit <- YesOrNo(input("Exit (Y/N): "))
            sleep(2)
            if isSave then
                return [2, isExit, isLogout]
            else
                return [0, isExit, isLogout]
        else if choice = "2" then
            return [1, False, isLogout]
        else if choice = "3" then
            return [2, False, isLogout]
        else
            isSave <- YesOrNo(input("Ingin melakukan save ? (Y/N): "))
```

```

        if isSave then
            return[2, True, isLogout]
        else
            return [0, True, isLogout]
    else if role = 'admin' then
        displayBar("MENU")
        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. Monster : Menambahkan Monster baru")
        output("4. Exit : Keluar dari game")
        choice <- pilihanValid(input("<///> Pilih perintah (1/2/3) "),
["1", "2", "3", "4"])
        isLogout <- False
        if choice = '1' then
            isSave <- logout(dataUser)
            isLogout <- True
            isExit <- YesOrNo(input("Exit (Y/N): "))
            sleep(2)
            if isSave then
                return [2, isExit, isLogout]
            else
                return [0, isExit, isLogout]
        else if choice = '2' then
            return [1, False, isLogout]
        else if choice = '3' then
            return [3, False, isLogout]
        else
            isSave <- YesOrNo(input("Ingin melakukan save ? (Y/N): "))
            if isSave then
                return[2, True, isLogout]
            return [0, True, isLogout]

function menu(dataUser:dict)
    kondisi <- 0
    isExit <- False
    while kondisi = 0 do
        clear()
        if not isLogin(dataUser) then
            beforeLogin()
            choice <- pilihanValid(input("Masukkan pilihan (1/2/3): "),
["1", "2", "3"])
            if choice = '1' then
                dataUser <- login(dataUser)
            else if choice = '2' then
                register_manager()
            else
                isSave <- YesOrNo(input("Ingin melakukan save ? (Y/N):
"))
                if isSave then
                    return[2, True, True, dataUser]
                return [0, True, True, dataUser]
            else
                user_role <- dataUser["Role"]
                [kondisi, isExit, isLogout] <- afterLogin(user_role,
dataUser)
            if isExit then
                break

```



```
return [kondisi, isExit, isLogout, dataUser]
```

7. F05 – Monster

a. Desain Kamus

```
KAMUS LOKAL {Deklarasi variabel}  
monsterId, statIndex: integer  
data: matrix of string  
hasil: array of string
```

b. Desain Spesifikasi

```
from src.share import arraycsv, search  
function statMonster(monsterId, statIndex) -> string:  
  
{Spesifikasi : fungsi statMonster digunakan untuk mengambil stat tertentu  
dari database Monster}  
{I.S : user memasukkan monsterId dan statindex yang diinginkan untuk  
diambil}  
F.S : user mendapat string stat monster tersebut}  
  
ALGORITMA  
data <- arraycsv("monster")  
hasil <- search(0, str(monsterId), data)[0]  
return hasil[statIndex]
```

```
function get_stats(id, level) -> dictionary:  
  
{Spesifikasi : fungsi get_stats menerima input monsterId dan levelMonster  
untuk menghasilkan dictionary stat monster}  
{I.S : user memasukkan monsterId dan level Monster yang diinginkan untuk  
diambil stat nya}  
F.S : user mendapat dictionary stat monster tersebut}  
  
KAMUS LOKAL {Deklarasi variabel}  
id, level : integer  
pengaliLevel : float  
stat : dictionary  
  
ALGORITMA  
if level = 1 then  
    pengaliLevel <- 1  
else  
    pengaliLevel <- ((level - 1) * 10 + 100) / 100  
stat <- {  
    "ID" : id,  
    "Name": statMonster(id, 1),  
    "Atk": int(int(statMonster(id, 2)) * pengaliLevel),  
    "Def": int(int(statMonster(id, 3)) * pengaliLevel),  
    "HP": int(int(statMonster(id, 4)) * pengaliLevel),  
    "Level": level  
}  
return stat
```

```
from src.share import displayBar
from src.load import loadInvent
procedure monsterList(userId, data=None):
```

{Spesifikasi : fungsi monsterList akan menampilkan list monster yang dimiliki user}

{I.S : user memasukkan userId dan data bila ada, jika tidak akan mengambil langsung dari database}

F.S : user mendapat tampilan semua monster yang dimiliki}

KAMUS LOKAL {Deklarasi variabel}

```
userId, levelMonster, space : integer
monsterID, monsterName, monsterList : string
stat, data : dictionary
```

ALGORITMA

```
if data is None then
    data <- loadInvent(userId, "monster")
displayBar("MONSTER LIST")
[i, element] traversal enumerate(data["MonsterID"])
    monsterID <- element
    levelMonster <- level(monsterID, data)
    stat <- get_stats(monsterID, levelMonster)
    monsterName <- stat["Name"]
    monsterlist <- f"{i+1}. {monsterName}"
    space <- 15 - len(monsterlist)
    output(monsterlist, " " * space + f"(Lvl: {levelMonster})")
```

```
from src.share import clear, pilihanValid
function pilihMonster(userId, monsterUser, withList=False) -> integer:
```

{Spesifikasi : fungsi pilihMonster menerima input userId, monsterUser dan withList untuk menentukan monster mana yang user pilih}

{I.S : user mendapat tampilan monster list jika withlist true, setelah itu user akan diminta input, jika input tidak valid akan terus diminta hingga valid}

F.S : user memasukkan input yang valid dan menghasilkan pilihan}

KAMUS LOKAL {Deklarasi variabel}

```
userId, maxPilihan, pilihan : integer
WithList : boolean
monsterUser, data : dictionary
```

ALGORITMA

```
maxPilihan <- len(monsterUser["MonsterID"]) + 1
if withList then
    monsterList(userId, monsterUser)
    while True do
        pilihan <- int(pilihanValid(input("<///> Pilih monster untuk bertarung : "), [f'{i+1}' i traversal range(maxPilihan-1)]))
        clear()
        return int(monsterUser["MonsterID"][pilihan-1])
```

```
from src.load import loadData
```

```
function banyakMonster() -> integer:
```

{Spesifikasi : fungsi banyak monster digunakan untuk mencari tahu berapa jumlah monster di database}

```
{I.S : system meminta berapa banyak monster yang ada di database}  
{F.S : system menerima jumlah monster}
```

```
return len(loadData("monster") ["ID"])
```

```
from src.share import index
```

```
function level(monsterId, data) -> integer:
```

```
{Spesifikasi : fungsi level digunakan untuk menentukan level monster  
dengan monsterId tertentu dalam suatu data}
```

```
{I.S : user ingin mendapat level dari monster milik user}
```

```
{F.S : user mendapat level monster dari dataUser}
```

KAMUS LOKAL {Deklarasi variabel}

monsterId : integer

hasil : string

data : dictionary

ALGORITMA

```
hasil <- data["Level"][index(str(monsterId), data["MonsterID"])]
```

```
return int(hasil)
```

8. F06 – Potion

A. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}

userId, quantityPotion, quantity, space, mac : integer

data, potionUser, status, dataPotion, potionInvent, temp :

dictionary

type, potion, typePotion : string

B. Desain Spesifikasi

```
from src.load import loadInvent
```

```
function getPotion(userId:int) -> dict
```

ALGORITMA

```
potionUser <- loadInvent(userId, "potion")
```

```
data <- {}
```

```
for [i ,potion] in enumerate(potionUser["Type"])
```

```
    quantityPotion <- int(potionUser["Quantity"][i])
```

```
    if quantityPotion > 0 then
```

```
        data[potion] <- str(quantityPotion)
```

```
return data
```

```
function potionStatus(potionUser:dict) -> dict
```

ALGORITMA

```
status <- {}
```

```
for potion in potionUser
```

```
    status[potion] <- 0
```

```
return status
```

```
function tambahPotion(type:str, quanntity:int, dataPotion:dict) -> dict
```

ALGORITMA

```
dataPotion[type] + quanntity
return dataPotion
```

```
from src.share import displayBar
function potionList(potionUser:dict) -> int
```

ALGORITMA

```
displayBar("POTION LIST")
for [i, potion] in enumerate(potionUser)
    space <- 12 - len(potion)
    output(f"{i+1}. {potion} Potion" + space * " " + f"(Qty:
{potionUser[potion]})", end="")
    if potion = "Strength" then
        output(" - Increase ATK Power")
    else if potion = "Resilience" then
        output(" - Increase DEF Power")
    else if potion = "Healing" then
        output(" - Restore Health")
    else
        output()
max <- len(potionUser) + 1
output(f"{max}. Cancel")
return max
```

```
procedure updatePotionUser(potionUser:dict, potionInvent:dict)
```

ALGORITMA

```
temp <- {}
for [i, typePotion] in enumerate(potionInvent["Type"])
    quantityPotion <- int(potionInvent["Quantity"][i])
    if quantityPotion > 0 then
        temp[typePotion] <- str(quantityPotion)
    if typePotion in potionUser then
        if potionUser[typePotion] = "0" then
            del potionUser[typePotion]
for typePotion in temp
    potionUser[typePotion] <- temp[typePotion]
```

```
procedure updatePotionInvent(potionUser:dict, potionInvent:dict)
```

ALGORITMA

```
for [i, typePotion] in enumerate(potionInvent["Type"])
    if typePotion in potionUser then
        quantityPotion <- potionUser[typePotion]
    else
        quantityPotion <- 0
    potionInvent["Quantity"][i] <- str(quantityPotion)
```

9. F07 – Inventory**A. Desain Kamus**

KAMUS LOKAL {Deklarasi variabel}

quantity, monsterId, levelMonster, maxHp, nomor, pilihan :

integer

PotionUser, desc, inventPotionUser, invent, inventMonsterUser,
inventPotionUser, inventBall : array of string

```
DataPotion, dataStatMonster, dataStatPotion : array of
dictionary
hasil, namaMonster, inventBall : string
monsterUser, stat, monsterBall, dataUser, potionUser :
dictionary
IsExit : float
```

B. Desain Spesifikasi

```
from src.battle import showStat
from src.share import clear, displayBar, pilihanValid, YesOrNo, display
procedure inventory(dataUser, monsterUser, potionUser, monsterBall):
{Spesifikasi : prosedur inventory digunakan untuk menampilkan semua hal
yang ada di inventory}
{I.S : user memilih untuk masuk ke inventory, dan ditampilkan list
barang}
F.S : user memilih keluar inventory}
```

ALGORITMA

```
    while True do
        clear()
        [inventMonsterUser, dataStatMonster] <-
monsterInventory(monsterUser)
        [inventPotionUser, dataStatPotion] <- potionInventory(potionUser)
        inventBall <- ballInventory(monsterBall)
        i transversal inventPotionUser:
            inventMonsterUser.append(i)
        if monsterBall["Ball"] != 0 then
            inventMonsterUser.append(inventBall)
        invent <- inventMonsterUser
        {Menampilkan hasil ke terminal}
        displayBar("User Info")
        output(
f"""User ID : {dataUser["ID"]}
Nama      : {dataUser["Username"]}
OC        : {dataUser["OC"]}"""
        displayBar("Inventory List")
        barang transversal enumerate(invent):
            output(f"{barang[0]+1}. {barang[1]}")
            nomor <- len(invent) + 1
            output(f"{nomor}. Keluar")
            output("Ketikkan id untuk menampilkan item")
            pilihan <- int(pilihanValid(input("<///> : "), [str(i) i
traversal(1...nomor+1)]))
            clear()
            if pilihan = nomor then
                break
            else if invent[pilihan-1][:12] = "Monster Ball" then
                output("Monster Ball")
                output(f"Quantity : {monsterBall["Ball"]}")
            else if invent[pilihan-1][:7] = "Monster" then
                output("Monster")
                showStat(dataStatMonster[pilihan-1])
            else if invent[pilihan-1][:6] = "Potion" then
                data <- dataStatPotion[pilihan - len(dataStatMonster) - 1]
                output("Potion")
                display(
f"""Type      : {data[0]}
Quantity : {data[1]}"""
                isExit <- YesOrNo(input("<///> Keluar Inventory (Y/N): "))
                if isExit then
```

```
clear()
break
```

```
from src.monster import get_stats, level
function monsterInventory(monsterUser) -> array:
{Spesifikasi : fungsi untuk melakukan update pada inventory dengan
mengaplikasikan perubahan dari monsterUser}
{I.S : perubahan monsterUser belum ada di monsterInventory}
F.S : monsterInvent sudah up to date}
```

ALGORITMA

```
desc <- []
dataStatMonster <- []
monster tranversal monsterUser["MonsterID"]:
    monsterId <- int(monster)
    levelMonster <- level(monsterId, monsterUser)
    stat <- get_stats(monsterId, levelMonster)
    maxHp <- stat["HP"]
    namaMonster <- stat["Name"]
    hasil <- "Monster" + " " * (15- len("Monster")) + f"(Name:
{namaMonster}, Lvl: {levelMonster}, HP: {maxHp})"
    desc.append(hasil)
    dataStatMonster.append(stat)
return [desc, dataStatMonster]
```

```
function ballInventory(monsterBall) -> string:
{Spesifikasi : fungsi untuk melakukan update pada inventory dengan
mengaplikasikan perubahan dari monsterBall}
{I.S : perubahan monsterBall belum ada di monsterInventory}
F.S : data Inventory sudah up to date}
```

ALGORITMA

```
hasil <- "Monster Ball" + " " * (15 - len("Monster Ball")) + f"(Qty:
{monsterBall["Ball"]})"
return hasil
```

```
function potionInventory(potionUser) -> array:
{Spesifikasi : fungsi untuk melakukan update pada inventory dengan
mengaplikasikan perubahan dari potionUser}
{I.S : perubahan potionUser belum ada di monsterInventory}
F.S : data Inventory sudah up to date}
```

ALGORITMA

```
inventPotionUser <- []
dataPotion <- []
potionType tranversal potionUser:
    depend on potionType
        potionType = "Strength":
            type <- "ATK"
        potionType = "Resilience" then
            type <- "DEF"
        potionType = "Healing" then
            type <- "Heal"
    quantity <- potionUser[potionType]
    hasil <- "Potion" + " " * (15- len("Potion")) + f"(Type: {type},
Qty: {quantity})"
    dataPotion.append([type, quantity])
```

```
inventPotionUser.append(hasil)
return [inventPotionUser, dataPotion]
```

10. F08 – Battle

A. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}

kondisi, hadiah, ronde, userId, maxHpMusuh, maxHpAgent, levelMusuh, idMusuh, monsterId, levelMonste, maxHp, number, pilihan, LowATK, highATK, Atk, Def, rngAtk, damage, quantity : integer

NamaUser, AttackerName, defenderName, typePotion : string

ambush, isEscape, isCatch, isCancel, hasil : boolean

dataPotion : array of dictionary

status, dataUser, potionUser, monsterUser, monsterBall, statMusuh, statAgent, stat, allies, enemies, defender, agent, musuh: dictionary

B. Desain Spesifikasi

```
from src.share import clear
from src.potion import potionStatus
function battle(dataUser, potionUser, monsterUser, monsterBall,
ambush=False) -> boolean:
{Spesifikasi : fungsi battle menerima input dataUser, potionUser,
monsterUser, monsterBall, dan ambush. ambush opsional. Melakukan battle
dengan monster yang digenerate secara random baik id dan levelnya}
{I.S : user memilih untuk bertarung atau terkena ambush}
{F.S : user menang, kalah atau escape dari pertempuran}
```

ALGORITMA

```
clear()
kondisi <- 0
userId <- dataUser["ID"]
namaUser <- dataUser["Username"]
[statAgent, statMusuh] <- encounter(userId, namaUser, monsterUser)
ronde:int <- 0
status <- potionStatus(potionUser)
maxHpMusuh <- statMusuh["HP"]
maxHpAgent <- statAgent["HP"]
levelMusuh <- statMusuh["Level"]
clear()
if ambush then
    [isEscape, isCatch] <- turnEnemy(ronde, statAgent, statMusuh)
    [hasil, kondisi] <- check(isEscape, isCatch, statAgent,
statMusuh)
    while kondisi = 0 do
        ronde + 1
        [isEscape, isCatch] <- turnAlly(ronde, monsterUser, monsterBall,
statAgent, statMusuh, status, potionUser, maxHpMusuh, maxHpAgent)
        [hasil, kondisi] <- check(isEscape, isCatch, statAgent,
statMusuh)
        if kondisi != 0 then
            break
        [isEscape, isCatch] <- turnEnemy(ronde, statAgent, statMusuh)
        [hasil, kondisi] <- check(isEscape, isCatch, statAgent,
statMusuh)
        clear()
        showStat(statMusuh, maxHpMusuh)
    output ("                VS                ")
```

[illegible]


```
clear()
return hasil
```

[illegible]

```

from src.share import display
function showStat(stat, maxHp=None) -> string:
{Spesifikasi : fungsi showStat akan menampilkan stat monster ke layar}
{I.S : user ingin menampilkan stat}
{F.S : system menampilkan stat ke layar}

```

ALGORITMA

```

    if maxHp is None then
        maxHp <- stat["HP"]
    display (
f"""Name          : {stat["Name"]}
ATK Power : {stat["Atk"]}
DEF Power : {stat["Def"]}
HP        : {stat["HP"]}/{maxHp}
Level     : {stat["Level"]}"""

```

```

from src.share import clear, pilihanValid, sleep, displayBar
from src.potion import potionList
function turnAlly(number, monsterUser, monsterBall, allies, enemies,
status, potionUser, maxHpMusuh, maxHpAgent) -> array of boolean:
{Spesifikasi : fungsi turnAlly adalah user akan memilih aksi dalam
turnya, dapat berupa attack, potion, monster ball atau escape}
{I.S : user memasuki turn nya}
{F.S : user menyelesaikan turnnya dengan melakukan suatu aksi}

```

ALGORITMA

```

    while True do
        clear()
        showStat(enemies, maxHpMusuh)
        output("                                VS")
    ")
        showStat(allies, maxHpAgent)
        displayBar(f"Turn {number} ({allies["Name"]})")
        output(
f"""1. Attack
2. Use Potion
3. Use Monster Ball
4. Escape"""
        pilihan <-int(pilihanValid(input("<///> Pilih perintah: "), ["1",
"2", "3", "4"]))
        depend on pilihan
            pilihan = 1:
                attack(allies["Atk"], enemies["Def"], allies["Name"],
enemies["Name"], enemies)
                return [False, False]
            pilihan = 2:
                maxPilihan <- potionList(potionUser)
                isCancel <- usePotion(status, potionUser, maxPilihan,
allies, maxHpAgent)
                if isCancel then
                    continue
                return [False, False]
            pilihan = 3:
                isCatch <- catch(enemies, monsterUser, monsterBall)
                sleep(3)

```

```

        if isCatch then
            return [False, True]
        else
            return [True, False]

```

```

from src.share import displayBar, clear
function turnEnemy(number, allies, enemies) -> array of boolean:
{Spesifikasi : fungsi turnEnemy adalah monster musuh akan melakukan serangan attack}
{I.S : user menyelesaikan turnnya dan musuh belum mati}
{F.S : musuh telah menyerang user}

```

ALGORITMA

```

    displayBar(f"Turn {number} ({enemies["Name"]})")
    attack(enemies["Atk"], allies["Def"], enemies["Name"],
allies["Name"], allies)
    clear()
    return[False, False]

```

```

from src.rng import random
from src.share import sleep
procedure attack(Atk:int, Def:int, attackerName, defenderName, defender):
{Spesifikasi : fungsi attack untuk melakukan damage pada defender}
{I.S : attacker memilih untuk menyerang defender}
{F.S : defender menerima damage}

```

ALGORITMA

```

    lowATK <- int(Atk * 7/10)
    highATK <- int(Atk * 13/10)
    rngATK <- random(numRange=[lowATK, highATK])
    DEF <- rngATK * (Def/100)
    damage <- int(rngATK - DEF)
    output(f"{attackerName} attack {defenderName} dealing {damage} damage
!!!")
    defender["HP"] - damage
    if defender["HP"] < 0 then
        defender["HP"] <- 0
    sleep(3)

```

```

from src.share import clear, pilihanValid, sleep
function usePotion(status, potionUser, maxPilihan, allies, maxHp):
{Spesifikasi : fungsi usepotion saat user memilih untuk menggunakan potion
dan user memilih potion mana yang ingin digunakan dari list potion user}
{I.S : user memilih menggunakan potion}
{F.S : user memilih aksi entah menggunakan potion atau cancel}

```

ALGORITMA

```

    while True do
        pilihan <- int(pilihanValid(input("<///> Pilih potion: "),
[str(i+1) i traversal(maxPilihan)]))
        if pilihan-1 = len(status) then
            clear()
            return True
        typePotion <- [potion for potion in potionUser][pilihan-1]

```

```

quantity <- int(potionUser[typePotion])
if quantity = 0 then
    output(f"{typePotion} potion sudah habis")
else if status[typePotion] = "1" then
    output("sudah digunakan")
else:
    output(f"{typePotion} potion digunakan")
    quantity <- quantity - 1
    potionUser[typePotion] <- str(quantity)
    depend on typePotion
        TypePotion = "Strength":
            allies["Atk"] + int(5 / 100 * allies["Atk"])
        typePotion = "Resilience":
            allies["Def"] + int(5 / 100 * allies["Def"])
        typePotion = "Healing":
            allies["HP"] + int(25 / 100 * maxHp)
            if allies["HP"] > maxHp then
                allies["HP"] = maxHp
    status[typePotion] = str(1)
    sleep(2)
    return False

```

```

function check(isEscape, isCatch, agent, musuh) -> array:
{Spesifikasi : fungsi check untuk melakukan check pada kondisi battle
apakah sudah selesai atau tidak}
{I.S : setiap selesai turn}
{F.S : saat kondisi yang direturn bukan 0 maka fungsi battle selesai}

```

ALGORITMA

```

if isEscape then
    return [False, 1]
else if isCatch then
    return [True, 4]
else if agent["HP"] = 0 then
    return [False, 2]
else if musuh["HP"] = 0 then
    return [True, 3]
else
    return [False, 0]

```

11. F09 – Arena

A. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}

kondisi, ronde, idMusuh, hadiah, levelMusuh, arena, maxHpMusuh, maxHpAgent, maxPilihan, maxHp, pilihan, quantity, monsterId, levelMonster, userId, number, dmg, LowATK, highATK, Atk, Def, rngAtk, damage: **integer**

IsMenang, hasil, isEscape, isCancel : **boolean**

AttackerName, defenderName : **string**

dataRonde : **array of integer**

dataUser, dataPotion, potionUser, monsterUser, allies, enemies, statAgent, hasilLatihan, statMusuh, status, defender : **dictionary**

B. Desain Spesifikasi

```

from src.potion import potionStatus
from src.load import showDict
from src.share import display, sleep, clear

function arena(dataUser, potionUser, monsterUser) -> boolean
{Spesifikasi : fungsi arena digunakan untuk memasuki arena dan
menantang 5 stage yang akan mendapat hadiah. Dan data hasil latihan
akan ditampilkan di akhir}
{I.S : user memilih memasuki arena }
{F.S : user menyelesaikan latihan}

```

ALGORITMA

```

clear()
statAgent <- enterArena(dataUser, monsterUser)
hasilLatihan = {
    "Total Hadiah": 0,
    "Jumlah Stage": 0,
    "Damage diberikan": 0,
    "Damage diterima": 0,
    "Potion digunakan": 0,
    "Total turn": 0,
    "Battle terlama": 0,
    "Battle tersingkat": 0
}
dataRonde <- []
dataPotion <- potionStatus(potionUser)
for traversal [1...6]
    output (f"Memasuki Ronde {i}...")
    sleep(3)
    [isMenang, kondisi] <- battleArena(potionUser, statAgent, i,
hasilLatihan, dataRonde, dataPotion, dataUser)
    if isMenang then
        continue
    else
        sleep(3)
        break
    hasilLatihan["Potion digunakan"] <- dataPotion["Strength"] +
dataPotion["Resilience"] + dataPotion["Healing"]
    hasilLatihan["Battle terlama"] <- maxEle(dataRonde)
    hasilLatihan["Battle tersingkat"] <- minEle(dataRonde)
    depend on kondisi
        kondisi = 1
            display("GAME OVER !, Anda mengakhiri battle")
            showDict(hasilLatihan)
            return isMenang
        kondisi = 2:
            display("GAME OVER !, Anda kalah")
            showDict(hasilLatihan)
            return isMenang
        kondisi = 3:
            display("CONGRATS !, Anda menang ")
            showDict(hasilLatihan)
            return isMenang

```

```

from src.battle import pilihMonster, showStat
from src.monster import level, get_stats

```



```

      /  |  ^^^^  |  /      \  |  /  |  ./.  .vvvvv.  |  \
     /  /  |      |  /      \  |  /  |  ^vvvv'  |  /  \
    /  /  |      |  /      \  |  /  |  ~      |  /      \
   /  /  |      |  /      \  |  /  |  7      |  /      \
  /  /  |      |  /      \  |  /  |  .      |  /      \
 /  /  |      |  /      \  |  /  |  -  |  /  \  |  /  \
--'---'---'---'---'---'---'---'---'---'---'---'---'---'---'
                                     \uU \UU/      |      |      :F_P:"")

```

```

output(f"RAWRR, Monster {statMusuh["Name"]} telah muncul !!!")
sleep(3)
ronde <- 0
status <- potionStatus(potionUser)
maxHpMusuh <- statMusuh["HP"]
maxHpAgent <- statAgent["HP"]
levelMusuh <- statMusuh["Level"]
clear()
while kondisi = 0 do
  ronde + 1
  isEscape <- turnArena(hasilLatihan, dataPotion, ronde,
statAgent, statMusuh, status, potionUser, maxHpMusuh, maxHpAgent)
  [hasil, kondisi] <- check(isEscape, statAgent, statMusuh)
  if kondisi != 0 then
    break
  turnMusuhArena(hasilLatihan, ronde, statAgent, statMusuh)
  [hasil, kondisi] <- check(isEscape, statAgent, statMusuh)
clear()
if kondisi = 3 then
  hadiah <- dataHadiah[arena-1]
  hasilLatihan["Total Hadiah"] + hadiah
  hasilLatihan["Total turn"] + ronde
  hasilLatihan["Jumlah Stage"] <- arena
  dataRonde.append(ronde)
  dataUser["OC"] <- str(int(dataUser["OC"]) + hadiah)
  display(f"Selamat anda telah menyelesaikan ronde {arena},
mendapatkan {hadiah} OC !!!!")
  sleep(3)
return [hasil, kondisi]

```

```

from src.battle import showStat
from src.share import display, clear
function turnArena(hasilLatihan, dataPotion, number, allies, enemies,
status, potionUser, maxHpMusuh, maxHpAgent) -> boolean:
{Spesifikasi : fungsi turnArena adalah saat user bergerak pada
turnnya user dapat memilih untuk attack, potion atau escape dan akan
direkam di hasil latihan}
{I.S : user memasuki turnnya}
{F.S : user sudah memilih aksi}
ALGORITMA
  while True do
    clear()
    showStat(enemies, maxHpMusuh)
    output("                               VS                               ")
    showStat(allies, maxHpAgent)
    displayBar(f"Turn {number} ({allies["Name"]})")

```

```

        output(
f"""1. Attack
2. Use Potion
3. Escape""")

        pilihan <-int(pilihanValid(input("<///> Pilih perintah: "),
["1", "2", "3"]))
        if pilihan = 1 then
            dmg <- attackArena(allies["Atk"], enemies["Def"],
allies["Name"], enemies["Name"], enemies)
            hasilLatihan["Damage diberikan"] + dmg
            return False
        else if pilihan = 2 then
            maxPilihan <- potionList(potionUser)
            isCancel <- usePotionArena(status, dataPotion,
potionUser, maxPilihan, allies, maxHpAgent)
            if isCancel then
                continue
            return False
        else
            return True

```

```

from src.rng import random
from src.share import sleep
procedure attackArena(Atk:int, Def:int, attackerName, defenderName,
defender):
{Spesifikasi : procedure attackArena adalah defender akan menerima
sejumlah damage dari attacker dan akan direkam di hasil latihan}
{I.S : attacker memilih untuk melakukan attack pada defender}
{F.S : defender menerima sejumlah damage}

```

ALGORITMA

```

        lowATK <- int(Atk * 7/10)
        highATK <- int(Atk * 13/10)
        rngATK <- random(numRange=[lowATK, highATK])
        DEF <- rngATK * (Def/100)
        damage <- int(rngATK - DEF)
        output(f"{attackerName} attack {defenderName} dealing {damage} damage
!!!")
        defender["HP"] - damage
        if defender["HP"] < 0 then
            defender["HP"] <- 0
        sleep(3)

```

```

from src.share import displayBar, clear
function turnEnemy(hasilLatihan, number, allies, enemies) -> array of
boolean:
{Spesifikasi : fungsi=i turnEnemy adalah saat musuh akan melakukan
turnnya dengan melakukan attack dan akan direkam di hasil latihan}
{I.S : saat user menyelesaikan turnnya}
{F.S : saat user menerima damage}

```

ALGORITMA

```

        displayBar(f"Turn {number} ({enemies["Name"]})")
        dmg <- attackArena(enemies["Atk"], allies["Def"],
enemies["Name"], allies["Name"], allies)

```



```
hasilLatihan["Damage diterima"] + dmg  
clear()
```

```
from src.share import clear, pilihanValid, sleep  
function usePotionArena(status, dataPotion, potionUser, maxPilihan,  
allies, maxHp) -> boolean:  
{Spesifikasi : fungsi userPotionArena adalah saat user memilih untuk  
menggunakan potion dan memilih dari list potion user dan penggunaanya  
akan direkam di hasil latihan }  
{I.S : saat user memilih untuk menggunakan potion}  
{F.S : user memilih menggunakan potion atau cancel}
```

ALGORITMA

```
while True do  
    pilihan <- int(pilihanValid(input("<///> Pilih potion: "),  
[str(i+1) for i traversal [maxPilihan]]))  
    if pilihan-1 = len(status)t then  
        clear()  
        return True  
    typePotion <- [potion for potion in potionUser][pilihan-1]  
    quantity <- int(potionUser[typePotion])  
    if quantity = 0 then  
        output(f"{typePotion} potion sudah habis")  
    else if status[typePotion] = "1" then  
        output("sudah digunakan")  
    else  
        output(f"{typePotion} potion digunakan")  
        quantity <- quantity - 1  
        potionUser[typePotion] <- str(quantity)  
        depend on typePotion  
            TypePotion = "Strength":  
                allies["Atk"] + int(5 / 100 * allies["Atk"])  
            typePotion = "Resilience":  
                allies["Def"] + int(5 / 100 * allies["Def"])  
            typePotion = "Healing":  
                allies["HP"] + int(25 / 100 * maxHp)  
                if allies["HP"] > maxHp then  
                    allies["HP"] = maxHp  
        status[typePotion] <- str(1)  
        dataPotion[typePotion] + 1  
        sleep(2)  
    return False
```

12. F10 - Shop & Currency

a. Desain Kamus

```
KAMUS LOKAL {Deklarasi variabel}  
dataUser, monsterShop, itemShop, potionInvent, dataMonster,  
monsterInvent, ballUser, monster_data : dictionary  
owca_coin, userId, pilihan, item_id, quantity, counter :  
integer  
monsterIndex, monster_type, atk_power, def_power, hp, stock,  
price action, item_type : string
```

b. Desain Spesifikasi

```
from src.share import clear, pilihanValid, index
from src.load import monster
procedure shop(dataUser:dict, monsterShop:dict, itemShop:dict,
potionInvent:dict, dataMonster:dict, monsterInvent:dict, ballUser:dict)

ALGORITMA
    owca_coin <- int(dataUser["OC"])
    userId <- int(dataUser["ID"])
    clear()

output("""<=====
==>
    Irasshaimase! Selamat datang di SHOP!!""")
    output()
    while True do
        action <- input(">>> Pilih aksi (lihat/beli/keluar): ").lower()
        if action = "lihat" then
            item_type <- input(">>> Mau lihat apa? (monster/item):
").lower()
            if item_type = "monster" then
                show_monsters(monsterShop, dataMonster)
            else if item_type = "item" then
                show_items(itemShop)
            else if action = "beli" then
                output(f"Jumlah O.W.C.A. Coin-mu sekarang {owca_coin}.")
                item_type <- input(">>> Mau beli apa? (monster/item):
").lower()
                if item_type = "monster" then
                    pilihan <- int(pilihanValid(input(">>> Masukkan id
monster: "), [str(i + 1) for i in range(len(monsterShop["MonsterID"])])))
                    owca_coin <- buy_monster(pilihan, owca_coin, monsterShop,
dataMonster, monsterInvent)
                else if item_type = "item" then
                    item_id <- int(pilihanValid(input(">>> Masukkan id item:
"), [str(i + 1) for i in range(len(itemShop["Type"])])))
                    quantity <- int(input(">>> Masukkan jumlah: "))
                    owca_coin <- buy_item(item_id, quantity, owca_coin,
itemShop, potionInvent, ballUser, userId)
                else if action = "keluar" then
                    output("Mr. Yanto bilang makasih, belanja lagi ya nanti :)")
                    break
                else
                    output("Aksi tidak valid.")
            dataUser["OC"] <- str(owca_coin)
```

```
procedure show_monsters(monsterShop:dict, monster_data:dict)
```

ALGORITMA

```
    output("ID | Type          | ATK Power | DEF Power | HP   | Stok |
Harga")
    for i, monster_id in enumerate(monsterShop["MonsterID"], start=1)
        monsterIndex <- index(str(monster_id), monster_data["ID"])
        monster_type <- monster_data["Type"][monsterIndex]
        atk_power <- monster_data["ATK_power"][monsterIndex]
        def_power <- monster_data["DEF_power"][monsterIndex]
        hp <- monster_data["HP"][monsterIndex]
        stock <- monsterShop["Stock"][i-1]
        price <- monsterShop["Price"][i-1]
```

```

    output(f"{i:3}| {monster_type:14}| {atk_power:10}|
{def_power:10}| {hp:4} | {stock:4} | {price:5}")
    output()

```

```

procedure show_items(itemShop:dict)

```

ALGORITMA

```

    output("ID | Type | Stok | Harga")
    for counter, i in enumerate(range(len(itemShop["Type"])), start=1)
        output(f"{counter:<3}| {itemShop['Type'][i]:20}|
{itemShop['Stock'][i]:5}| {itemShop['Price'][i]}")
    output()

```

```

function buy_monster(pilihan: int, owca_coin: int, monsterShop: dict,
monster_data: dict, userId: int, monsterInventory: dict) -> int

```

KAMUS LOKAL {Deklarasi variabel}

ALGORITMA

```

    monsterId <- monsterShop["MonsterID"][pilihan-1]
    monsterIndex <- index(str(monsterId), monster_data["ID"])
    monster_type <- monster_data["Type"][monsterIndex]
    stock <- monsterShop["Stock"][index(str(monsterId),
monsterShop["MonsterID"])]
    price <- monsterShop["Price"][index(str(monsterId),
monsterShop["MonsterID"])]

    if int(stock) > 0 and int(price) <= owca_coin then
        if str(monsterId) not in monsterInventory["MonsterID"] then
            monsterShop["Stock"][index(str(monsterId),
monsterShop["MonsterID"])] <- str(int(stock) - 1)
            output(f"Berhasil membeli item: {monster_type}. Item sudah
masuk ke inventory-mu!")
            monsterInventory["MonsterID"].append(str(monsterId))
            monsterInventory["Level"].append("1")
            return owca_coin - int(price)
        else
            output(f"Monster {monster_type} sudah ada dalam inventory-mu!
Pembelian dibatalkan.")
        else if int(stock) = 0 then
            output("Stok monster ini habis!")
        else
            output("OC-mu tidak cukup.")

    return owca_coin

```

```

function buy_item(id:int, quantity:int, owca_coin:int, itemShop:dict,
potionInvent:dict, ballUser:dict, userId:int) -> int

```

ALGORITMA

```

    item <- next((p for p in itemShop["Type"] if p = itemShop["Type"][id-
1]), None)
    if item then
        stock <- itemShop["Stock"][id-1]
        price <- itemShop["Price"][id-1]
        total_price <- int(price) * quantity

```

```

    if int(stock) >= quantity and total_price <= owca_coin then
        itemShop["Stock"][id-1] <- str(int(stock) - quantity)
        if item in potionInvent["Type"] then
            index <- potionInvent["Type"].index(item)
            potionInvent["Quantity"][index] <-
str(int(potionInvent["Quantity"][index]) + quantity)
        else if item = "Monster Ball" then
            ballUser["Ball"] <- str(int(ballUser["Ball"]) + quantity)
            output(ballUser)
            input()
        else # Jika item tidak dimiliki sama sekali oleh user
            potionInvent["Type"].append(item)
            potionInvent["Quantity"].append(str(quantity))
            output(f"Berhasil membeli item: {quantity} {item}. Item sudah
masuk ke inventory-mu!")
        return owca_coin - total_price
    else if int(stock) < quantity then
        output("Stok item tidak cukup!")
    else
        output("OC-mu tidak cukup.")
else
    output('item tidak ditemukan.')
return owca_coin

```

13. F11 – Laboratory

a. Desain Kamus

KAMUS LOKAL {Deklarasi variabel}
 UserId, OC, jumlahPilihan, pilihan, monsterId, levelMonster: integer
 userName : string
 IsExit, IsUpgrade : boolean
 dataUser, monsterUser, data, statMonste: dictionary

b. Desain Spesifikasi

```

from src.share import YesOrNo, clear
procedure laboratory(userId, dataUser, monsterUser)

{Spesifikasi : procedure laboratory dilakukan dengan menampilkan mosnter
user dan harga upgrade untuk user melakukan upgrade level monster}
{I.S : saat user memilih untuk masuk lab}
{F.S : user memilih keluar}

ALGORITMA
    clear()
    while True do
        userName <- dataUser["Username"]
        OC <- int(dataUser["OC"])
        jumlahPilihan <- len(monsterUser["MonsterID"]) + 1
        labMenu(userId, userName, OC, monsterUser, jumlahPilihan)
        [pilihan, monsterId, levelMonster] <-
pilihMonsterLab(monsterUser, jumlahPilihan)
        if pilihan = 0 then
            break
        upgrade(pilihan, monsterId, levelMonster, OC, dataUser,
monsterUser)
        isExit = YesOrNo(input("<///> Keluar (Y/N): "))
        if isExit then
            break

```

```

from src.share import displayBar
from src.monster import monsterList
procedure labMenu(userId, userName, OC, monsterUser, jumlahPilihan)

{Spesifikasi : procedure labMenu dilakukan dengan menampilkan monster
user dan harga upgrade dan OC user yang dimiliki}
{I.S : saat user memilih untuk masuk lab}
{F.S : user ditampilkan informasi harga upgrade}

```

ALGORITMA

```

    output(f"Selamat datang di Lab Dokter Asep. Agent {userName} !!!")
    monsterList(userId, monsterUser)
    output(f"{jumlahPilihan}. Cancel")
    displayBar("UPGRADE PRICE")
    output(
        ""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""
    )
    output(f"Anda memiliki {OC} OC ")

```

```

from src.share import YesOrNo, display, pilihanValid, clear, writcsv,
displayBar, index
from src.monster import get_stats, monsterList, level
function pilihMonsterLab(monsterUser, jumlahPilihan) -> array of integer:

```

```

{Spesifikasi : fungsi pilih monster lab untuk user memilih monster yang
ingin di upgrade}
{I.S : layar menampilkan pilihan}
{F.S : user memilih monster yang diupgrade}

```

ALGORITMA

```

    while True do
        pilihan <- int(pilihanValid(input("<///> Pilih monster: "),
[str(i+1) i traversal range(jumlahPilihan)]))
        if pilihan = jumlahPilihan then
            return[0, 0, 0]
        monsterId <- monsterUser["MonsterID"][pilihan-1]
        levelMonster <- level(monsterId, monsterUser)
        if levelMonster = 5 then
            output("max level")
        else
            clear()
            return [pilihan, monsterId, levelMonster]

```

```

from src.share import display, clear, YesOrNo
from src.monster import get_stats
procedure upgrade(pilihan, monsterId, levelMonster, OC, data,
monsterUser)

{Spesifikasi : procedure upgrade user melakukan upgrade level monster}
{I.S : saat user sudah memilih monster}
{F.S : user memilih untuk melanjutkan upgrade atau tidak}

```

ALGORITMA

```

    statMonster <- get_stats(monsterId, levelMonster)
    namaMonster <- statMonster["Name"]
    depend on levelMonster

```

```

    levelMonster = 1:
        hargaUpgrade <- 300
    levelMonster = 2 then
        hargaUpgrade <- 500
    levelMonster = 3 then
        hargaUpgrade <- 800
    levelMonster = 4 then
        hargaUpgrade <- 1000
    if hargaUpgrade <= OC then
        display(
f""{namaMonster} akan di-upgrade ke level {levelMonster + 1}
Harga untuk melakukan upgrade {namaMonster} adalah {hargaUpgrade} OC
Saat ini anda memiliki {OC} OC""")
        isUpgrade <- YesOrNo(input("<///> Lanjutkan upgrade (Y/N): "))
        clear()
        if isUpgrade then
            data["OC"] <- OC - hargaUpgrade
            monsterUser["Level"][pilihan-1] <- str(levelMonster + 1)
            display(f'Selamat, {namaMonster} berhasil di-upgrade ke level
{levelMonster + 1} !')
        else
            output(
f""Anda hanya memiliki {OC} OC
butuh {hargaUpgrade} OC untuk mengupgrade {namaMonster}""")

```

14. F12 - Shop Management

a. Desain Kamus

KAMUS LOKAL {Deklarasi Variabel}

monsterList, monsterShop, itemList, itemShop : dictionary

monsterId, monster_index, potionId, potion_index index,
stok_baru, harga_baru, stok, harga, itemId, action, item_type :
integer

monster_type, potion_type, item : string

confirm : boolean

b. Desain Spesifikasi

```

from src.share import clear, pilihanValid, YesOrNo, pop_element
from src.shop import show_monsters, show_items
procedure shop_management(monsterShop:dict, itemShop:dict,
monsterList:dict, itemList:dict)
ALGORITMA
    clear()

output ("""<=====
==>
    Irasshaimase! Selamat datang kembali, Mr. Monogram!""")
    output()
    while True do
        action <- input(">>> Pilih aksi (lihat/tambah/ubah/hapus/keluar):
").lower()
        if action = "lihat" then
            item_type <- input(">>> Mau lihat apa? (monster/item):
").lower()
            if item_type = "monster" then
                show_monsters(monsterShop, monsterList)
            else if item_type = "item" then
                show_items(itemShop)

```

```

        else if action = "tambah" then
            item_type <- input(">>> Mau nambahin apa? (monster/item):
").lower()
            if item_type = "monster" then
                add_monster(monsterList, monsterShop)
            else if item_type = "item" then
                add_item(itemList, itemShop)
        else if action = "ubah" then
            item_type <- input(">>> Mau ubah apa? (monster/item):
").lower()
            if item_type = "monster" then
                update_monster(monsterList, monsterShop)
            else if item_type = "item" then
                update_item(itemShop)
        else if action = "hapus" then
            item_type <- input(">>> Mau hapus apa? (monster/item):
").lower()
            if item_type = "monster" then
                delete_monster(monsterList, monsterShop)
            else if item_type = "item" then
                delete_item(itemShop)
        else if action = "keluar" then
            output("Dadah Mr. Yanto, sampai jumpa lagi!")
            break
        else
            output("Aksi tidak valid.")

```

procedure add_monster(monsterList:dict, monsterShop:dict)
 {Spesifikasi : Fungsi untuk menambahkan monster ke dalam shop }

ALGORITMA

```

        output("ID | Type | ATK Power | DEF Power | HP |")
        for monster_id, monster_type, atk_power, def_power, hp in
zip(monsterList["ID"], monsterList["Type"], monsterList["ATK_power"],
monsterList["DEF_power"], monsterList["HP"])
            if monster_id not in monsterShop["MonsterID"] then
                output(f"{monster_id:3}| {monster_type:14}| {atk_power:10}|
{def_power:10}| {hp:4} |")
                output()

        monsterId <- int(input(">>> Masukkan id monster: "))
        if str(monsterId) in monsterList["ID"] then
            index <- monsterList["ID"].index(str(monsterId))
            monster_type <- monsterList["Type"][index]
            if str(monsterId) in monsterShop["MonsterID"] then
                output(f"{monster_type} sudah ada dalam shop. Operasi tambah
dibatalkan.")
            else
                stok <- int(input(">>> Masukkan stok awal: "))
                harga <- int(input(">>> Masukkan harga: "))

                monsterShop["MonsterID"].insert(index, str(monsterId))
                monsterShop["Stock"].insert(index, str(stok))
                monsterShop["Price"].insert(index, str(harga))
                output(f"{monster_type} telah berhasil ditambahkan ke dalam
shop!")
        else

```

```
output("ID monster tidak ditemukan. Operasi tambah dibatalkan.")  
output()
```

```
procedure add_item(itemList:dict, itemShop:dict)  
{Spesifikasi : Fungsi untuk menambahkan potion ke dalam shop }
```

ALGORITMA

```
  for item in itemList  
    if item not in itemShop["Type"] then  
      sudahAda <- False  
      break  
    sudahAda <- True  
  if sudahAda then  
    output("Semua item sudah ada dalam shop. Tidak ada yang perlu  
ditambahkan.")  
    return  
  
  output("ID | Type |")  
  for counter, item in enumerate(itemList, start=1)  
    if item not in itemShop["Type"] then  
      output(f"{counter:<3}| {item:16}|")  
  output()  
  
  itemId <- int(input(">>> Masukkan id item: "))  
  if 1 <= itemId <= counter then  
    item <- itemList[itemId-1] # Ambil item berdasarkan indeks  
    stok <- int(input(">>> Masukkan stok awal: "))  
    harga <- int(input(">>> Masukkan harga: "))  
    itemShop["Type"].append(item)  
    itemShop["Stock"].append(str(stok))  
    itemShop["Price"].append(str(harga))  
    output(f"{item} telah berhasil ditambahkan ke dalam shop!")  
  else  
    output("ID item tidak valid. Operasi tambah dibatalkan.")  
  output()
```

```
from src.shop import show_monsters  
procedure update_monster(monsterList:dict, monsterShop:dict)  
{Spesifikasi : Fungsi untuk mengubah nilai stok atau harga dari monster}
```

ALGORITMA

```
  show_monsters(monsterShop)  
  monsterId <- int(input(">>> Masukkan id monster: "))  
  monster_index <- next((index for index, monster in  
enumerate(monsterShop["MonsterID"]) if monster == str(monsterId)), None)  
  
  if monster_index is not None then  
    index <- monsterList["ID"].index(str(monsterId))  
    monster_type <- monsterList["Type"][index]  
    stok_baru <- int(input(">>> Masukkan stok baru: "))  
    harga_baru <- int(input(">>> Masukkan harga baru: "))  
  
    if stok_baru > 0 then  
      monsterShop["Stock"][monster_index] <- str(stok_baru)  
  
    if harga_baru > 0 then  
      monsterShop["Price"][monster_index] <- str(harga_baru)
```



```

    if stok_baru > 0 and harga_baru > 0 then
        monsterShop["Stock"][monster_index] <- str(stok_baru)
        monsterShop["Price"][monster_index] <- str(harga_baru)
        output(f"{monster_type} telah berhasil diubah dengan stok
baru sejumlah {stok_baru} dan dengan harga baru {harga_baru}!")
    else if stok_baru > 0 then
        monsterShop["Stock"][monster_index] <- str(stok_baru)
        output(f"{monster_type} telah berhasil diubah dengan stok
baru sejumlah {stok_baru}!")
    else if harga_baru > 0 then
        monsterShop["Price"][monster_index] <- str(harga_baru)
        output(f"{monster_type} telah berhasil diubah dengan harga
baru {harga_baru}!")
    else
        output("Tidak ada perubahan yang dilakukan.")
else
    output("ID monster tidak ditemukan.")
output()

```

```

from src.shop import show_items
procedure update_item(itemShop:dict)
{Spesifikasi : Fungsi untuk mengubah nilai stok atau harga dari potion}

```

ALGORITMA

```

    show_items(itemShop)
    potionId <- int(pilihanValid(input(">>> Masukkan id item: "),
[str(i+1) for i in range(len(itemShop["Type"])) ]))

    if 1 <= potionId <= len(itemShop["Type"]) then
        potion_index <- potionId - 1
        potion_type <- itemShop["Type"][potion_index]
        stock_baru <- int(input(">>> Masukkan stok baru: "))
        harga_baru <- int(input(">>> Masukkan harga baru: "))

        if stock_baru > 0 and harga_baru > 0 then
            itemShop["Stock"][potion_index] <- str(stock_baru)
            itemShop["Price"][potion_index] <- str(harga_baru)
            output(f"{potion_type} telah berhasil diubah dengan stok baru
sejumlah {stock_baru} dan dengan harga baru {harga_baru}!")
        else if stock_baru > 0 then
            itemShop["Stock"][potion_index] <- str(stock_baru)
            output(f"{potion_type} telah berhasil diubah dengan stok baru
sejumlah {stock_baru}!")
        else if harga_baru > 0 then
            itemShop["Price"][potion_index] <- str(harga_baru)
            output(f"{potion_type} telah berhasil diubah dengan harga
baru {harga_baru}!")
        else
            output("Tidak ada perubahan yang dilakukan.")
            output()
    else
        output("ID item tidak valid!")

```

```

from src.shop import show_monsters
from src.share import YesOrNo, pop_element
procedure delete_monster(monsterList:dict, monsterShop:dict)
{Spesifikasi : Fungsi untuk menghapus monster dari shop}

```

ALGORITMA

```

show_monsters(monsterShop)
monsterId <- int(input(">>> Masukkan id monster: "))
monster_index <- next((index for index, monsterIdShop in
enumerate(monsterShop["MonsterID"]) if monsterIdShop == str(monsterId)),
None)

    if monster_index is not None then
        index <- monsterList["ID"].index(str(monsterId))
        monster_type <- monsterList["Type"][index]
        confirm <- YesOrNo(input(f">>> Apakah anda yakin ingin menghapus
{monster_type} dari shop (y/n)? "))

        if confirm then
            pop_element(monsterShop, monster_index)
        else if confirm.lower() == "n" then
            output("Operasi hapus dibatalkan.")
    else
        output("ID monster tidak ditemukan. Silakan coba lagi.")
    output()

```

```

from src.shop import show_items
from src.share import YesOrNo, pop_element
procedure delete_item(itemShop:dict)
{Spesifikasi : Fungsi untuk menghapus potion dari shop }

```

ALGORITMA

```

show_items(itemShop)
potionId <- int(input(">>> Masukkan id item: "))

potion_index <- potionId - 1
if 0 <= potion_index < len(itemShop["Type"]) then
    potion_type <- itemShop["Type"][potion_index]
    confirm <- YesOrNo(input(f">>> Apakah anda yakin ingin menghapus
{potion_type} dari shop (y/n)? "))

    if confirm then
        pop_element(itemShop, potion_index)
        output(f"{potion_type} telah berhasil dihapus dari shop!")
    else
        output("Operasi hapus dibatalkan.")
    else
        output("ID item tidak ditemukan atau bukan potion. Silakan coba
lagi.")
    output()

```

15. F13 - Monster Management**a. Desain Kamus**

```

KAMUS LOKAL {Deklarasi Variabel}
monsterId, pilihan, ATK, DEF, HP, id, space : integer
namaMonster, nilai, nama ; string
isKeluar, isMade, isTambah, isExit : boolean
columnLen : array of integer
dataMonster, file : dictionary

```

b. Desain Spesifikasi

```

from src.share import pilihanValid, YesOrNo, clear
procedure monsterManagement(dataMonster)

```

{Spesifikasi : procedure monster management adalah admin akan mendapat pilihan untuk melihat data monster atau membuat monster baru}

{I.S : saat admin memilih monster management}

{F.S : admin memilih keluar}

ALGORITMA

```
while True do
    clear()
    output(
        ""<=====>
        SELAMAT DATANG DI DATABASE PARA MONSTER !!!
        1. Tampilkan semua Monster
        2. Tambah Monster Baru""")
        pilihan <- int(pilihanValid(input("<///> Pilih perintah: "),
        ["1", "2"]))

        if pilihan = 1 then
            showMonsterData(dataMonster, columnLen<-[2, 12, 10, 10,
6])

        else
            [isMade, [monsterID, namaMonster, ATK, DEF, HP]] <-
buatMonster(dataMonster)
            if isMade then
                dataMonster["ID"].append(str(monsterID))
                dataMonster["Type"].append(str(namaMonster))
                dataMonster["ATK_power"].append(str(ATK))
                dataMonster["DEF_power"].append(str(DEF))
                dataMonster["HP"].append(str(HP))

            else
                break
            isKeluar <- YesOrNo(input("<///> Keluar (Y/N): "))
            if isKeluar then
                break
```

procedure showMonsterData(file, columnLen=None)

{Spesifikasi : procedure showMonsterData akan menampilkan data monster dalam bentuk kolom dengan panjang kolom berada di columnLen}

{I.S : saat admin memilih untuk melihat monster}

{F.S : system menampilkan data ke layar}

ALGORITMA

```
if columnLen is None then
    columnLen: <- [(len(category) + 2) for category in file]
    column transversal enumerate(file)
    nilai <- str(column[1])
    makeRow(nilai, columnLen, id=column[0])
    output()
    i transversal file["ID"]
    category transversal enumerate(file)
    nilai <- str(file[category[1]][int(i)-1])
    makeRow(nilai, columnLen, id=category[0])
    output()
```

```
procedure makeRow(nilai, columnLen, id)
```

```
{Spesifikasi : procedure makerow akan membuat baris pada tabel saat  
lihat monster}
```

```
{I.S : saat ingin membuat kolom pada tabel}
```

```
{F.S : system menampilkan kolom}
```

ALGORITMA

```
space <- (columnLen[id]-len(nilai))
```

```
output(nilai, space * " " + "|", end=" ")
```

```
from src.share import display, YesOrNo, clear, sleep, isDigit
```

```
function buatMonster(data) -> array:
```

```
{Spesifikasi : function buatMonster akan menghasilkan array yang  
berisi pilihan apakah ingin dibuat atau tidak dan data stat monster  
yang dibuat}
```

```
{I.S : saat ingin membuat monster}
```

```
{F.S : admin memilih untuk menyimpan monster ke database atau tidak}
```

ALGORITMA

```
namaMonster <- data["Type"]
```

```
display("Memulai pembuatan Monster...")
```

```
sleep(2)
```

```
while True do
```

```
nama:str <- input("Masukkan Type / Nama : ")
```

```
if nama in namaMonster then
```

```
output("Nama sudah terdaftar, coba lagi!")
```

```
else
```

```
monsterID:int <- len(namaMonster) + 1
```

```
ATK:int <- isDigit(input("Masukkan ATK Power : "))
```

```
DEF:int <- inputDEF()
```

```
HP:int <- isDigit(input("Masukkan HP : "))
```

```
output("Sedang membuat monster...")
```

```
sleep(3)
```

```
clear()
```

```
display(
```

```
f"""Monster baru berhasil dibuat!
```

```
Type : {nama}
```

```
ATK Power : {ATK}
```

```
DEF Power : {DEF}
```

```
HP : {HP}""")
```

```
isTambah:bool <- YesOrNo(input("Tambahkan Monster ke  
database (Y/N):"))
```

```
if isTambah then
```

```
return [True, [monsterID, nama, ATK, DEF, HP]]
```

```
else
```

```
isExit:bool <- YesOrNo(input("<///> Keluar (Y/N): "))
```

```
clear()
```

```
if isExit then
```

```
return [False, [0, 0, 0, 0, 0]]
```

```
from src.share import isDigit
```

```
procedure inputDEF() -> integer:
```

```
{Spesifikasi : procedure inputDEF akan meminta input dari user untuk
stat DEF dengan aturan input harus integer dan diantara 0 dan 50}
{I.S : saat ingin memasukkan def}
{F.S : saat user sudah menginput def}
```

ALGORITMA

```
while True do
  DEF <- isDigit(input("Masukkan DEF Power (0-50) :"))
  if 0 <= DEF <= 50 then
    return DEF
  else
    output("DEF Power harus bernilai 0-50, coba lagi!")
```

16. F14 – Load

a. Desain Kamus

```
KAMUS LOKAL {Deklarasi Variabel}
nama_folder, cnt_kolom, nama_file, args, data : string
hasil, value, res, quantityList : array of integer
monster, monsterShop, data, itemShop, user, monsterInventory,
potionInventory, ballInventory : dictionary
userId, : integer
typeList : array of string
```

b. Desain Spesifikasi

```
from src.share import split, search, display, loadingmsg, find_path
import os
import argparse
from time import sleep
procedure monsterManagement(dataMonster)
```

ALGORITMA

```
function read_csv(nama_folder, cnt_kolom, nama_file)
  with open(os.path.join(nama_folder, nama_file)) as file
    # Inisialisasi sebuah variabel bertipe tabel kosong yang akan
    memuat data dari csv
    res <- []
    for line in file
      items <- split(line, cnt_kolom) # parse CSV
      res.append(items) # masukkan baris hasil parsing ke tabel res
    return res
```

```
# FUNGSI LOAD
```

```
procedure load()
  parser <- argparse.ArgumentParser()
  parser.add_argument("nama_folder", type=str)
  args <- parser.parse_args()
  nama_folder <- args.nama_folder
  if find_path(nama_folder) = "" then
    output("Folder " + nama_folder + " tidak ditemukan.")
    exit()
  else
    loadingmsg("Loading")
    output()
    sleep(3)
    output('{: ^120s}'.format(" " * 120))
    output('{: ^120s}'.format("Selamat Datang di.."))
    output(""
```

```
W   CCCC      AAA
```

```
W       W   C      A   A
```

```
O O O      W
O          O
```

```

O      O      W W W      C      AAAAAAA
O      O      W W W      C      A      A
O O O      W W      CCCC A      A      """)
output('{: ^120s}'.format("*" * 120))
sleep(3)
output()
# Masukkan hasil parsing csv ke dalam array [dataSemuaUser,
dataMonster, dataMonsterShop, dataItemShop] <- loadData()
return [dataSemuaUser, dataMonster, dataMonsterShop, dataItemShop]

```

```

function readcsvData(fileName:str) -> list[list[str]]
with open(f'data\\{fileName}.csv', 'r') as file
    hasil:list <- []
    for line in file
        row <- list(split(line, ";"))
        hasil.append(row)
    value <- []
    for data in enumerate(hasil[0])
        temp <- [hasil[int(j)][data[0]] for j in range(len(hasil))]
        value.append(temp)
    return value

```

```

function readcsvInvent(fileName:str, userId:int=None) ->
list[list[str]]
with open (f"data\\{fileName}.csv", "r") as file
    hasil:list <- []
    for line in file
        row <- list(split(line, ";"))
        hasil.append(row)
    data <- []
    if userId is None then
        for barang in hasil
            data.append([barang[0], barang[1], barang[2]])
        return data[1:]
    else
        invent <- search(0, str(userId), hasil)
        for barang in invent
            data.append([barang[1], barang[2]])
        return data

```

```

function monster() -> dict
data <- readcsvData("monster")
monster <- {
    "ID":data[0][1:],
    "Type": data[1][1:],
    "ATK_power":data[2][1:],
    "DEF_power":data[3][1:],
    "HP": data[4][1:]
}
return monster

```

```

function monsterShop() -> dict
data <- readcsvData("monster_shop")
monsterShop <- {
    "MonsterID":data[0][1:],
    "Stock": data[1][1:],
    "Price":data[2][1:]
}
return monsterShop

```

```

function user(userId:int = None) -> dict
data <- readcsvData("user")
user <- {

```

| |
|---|
| <pre> "ID":data[0][1:], "Username": data[1][1:], "Password":data[2][1:], "Role":data[3][1:], "OC": data[4][1:] } <u>if</u> userId is not None then user <- { "ID":data[0][userId], "Username": data[1][userId], "Password":data[2][userId], "Role":data[3][userId], "OC": data[4][userId] } return user </pre> |
| <pre> <u>function</u> itemShop() -> dict data <- readcsvData("item_shop") itemShop <- { "Type":data[0][1:], "Stock": data[1][1:], "Price":data[2][1:] } return itemShop </pre> |
| <pre> <u>function</u> monsterInventory(userId:int) -> dict data <- readcsvInvent("monster_inventory", userId) monsterInventory:dict <- { "MonsterID": [barang[0] for barang in data], "Level": [barang[1] for barang in data] } return monsterInventory </pre> |
| <pre> <u>function</u> potionInventory(userId:int) -> dict data <- readcsvInvent("item_inventory", userId) typeList <- [] quantityList <- [] potionInventory:dict <- { "Type": typeList, "Quantity": quantityList } for barang in data <u>if</u> barang[0] != "Ball" then typeList.append(barang[0]) quantityList.append(barang[1]) return potionInventory </pre> |
| <pre> <u>function</u> ballInventory(userId:int) -> dict data <- readcsvInvent("item_inventory", userId) quantity <- 0 for barang in data <u>if</u> barang[0] = "Ball" then quantity <- int(barang[1]) ballInventory:dict <- { "Ball": quantity } return ballInventory </pre> |
| <pre> <u>function</u> loadData(data:str=None, userId:int=None) -> dict <u>if</u> data is None then return[user(), monster(), monsterShop(), itemShop()] <u>else</u> <u>if</u> data = "monster" then </pre> |

```

    return monster()
  else if data = "monster_shop" then
    return monsterShop()
  else if data = "user" then
    return user(userId)
  else if data = "item_shop" then
    return itemShop()

function loadInvent(userId:int, data:str=None) -> dict
  if data is None then
    return[monsterInventory(userId), potionInventory(userId),
ballInventory(userId)]
  else
    if data = "monster" then
      return monsterInventory(userId)
    else if data = "potion" then
      return potionInventory(userId)
    else if data = "ball" then
      return ballInventory(userId)

function getDataUser(userId:int) -> dict
  data <- loadData("user")
  hasil <- {
    "ID": userId,
    "Username": data["Username"][userId-1],
    "Password": data["Password"][userId-1],
    "Role": data["Role"][userId-1],
    "OC": data["OC"][userId-1]    }
  return hasil

function getBall(userId:int) -> dict
  data <- loadInvent(userId, "ball")
  return data

procedure showDict(data:dict)
  arr <- []
  text <- ""
  for i in data
    arr.append([i, data[i]])
  for [category, value] in arr
    space <- 8 - len(category)
    row <- f"{category}" + space * " " + ": " + str(value)
    text + f""{row}""
  display(text[:-1])

```

17. F15 – Save

a. Desain Kamus

KAMUS LOKAL {Deklarasi Variabel}

```

nama_folder, nama_file : string
arr_file, user, monster_shop, monster_inventory, item_shop,
item_inventory, dataUser, dataSemuaUser, potionInvent,
ballUser, userId, monsterUser, r, potionInvent,
ballUser, potionInvent, ballUser, userId, monsterUser, r,
potionInvent, ballUser, userId, monsterUser, : dictionary

```

b. Desain Spesifikasi

```

import os
from src.share import my_len, find_path, loadingmsg
from src.load import readcsvInvent

```



```

procedure write_csv (nama_folder, arr_file, nama_file)
  path_folder <- find_path(nama_folder)
  # path_folder akan mengembalikan "" jika tidak nama_folder tidak
  ditemukan
  if(path_folder=="") then # jika nama_folder tidak ditemukan
    path_folder <- os.path.join(os.getcwd(), nama_folder)
    os.makedirs(path_folder) # buat folder tersebut
    path_file <- os.path.join(path_folder, nama_file)
    if (os.path.exists(path_file)) then # jika file sudah ada dalam
    folder
      f <- open(path_file, "w+") # overwrite
      [i, column] traversal(arr_file)
      f.write(str(column))
      if (i != my_len(arr_file)-1) then
        f.write(";")
      f.write("\n")
      temp <- [i for i in arr_file]
      column <- temp[0]
      i traversal(len(arr_file[column]))
      [j, column] traversal(arr_file)
      f.write(str(arr_file[column][i]))
      if (j != my_len(arr_file)-1) then
        f.write(";")
      f.write("\n")
      f.close()
    else # jika file belum ada
      f <- open(path_file, "x") # buat file tersebut
      i traversal (my_len (arr_file))
      j traversal (my_len (arr_file[i]))
      f.write(str(arr_file[i][j]))
      if (j!=my_len(arr_file[i])-2) then
        f.write(";")
      f.write("\n")
      f.close()

# tes fungsi write_csv
# arr <- [[0 for j in range (2)] for i in range (4)]
# nama_folder <- input()
# arr <- [["game", "user_id"],["12", "2"]]
# write_csv(nama_folder, arr, nama_file)

```

```

# Fungsi save
procedure save (user, monster, monster_shop, monster_inventory,
item_shop, item_inventory)
  os.system('cls' if os.name='nt' else 'clear')
  nama_folder <- input("Masukkan nama folder penyimpanan: ")
  loadingmsg("Saving")
  write_csv (nama_folder, item_inventory, "item_inventory.csv")
  write_csv (nama_folder, item_shop, "item_shop.csv")
  write_csv (nama_folder, monster_inventory,
"monster_inventory.csv")
  write_csv (nama_folder, monster_shop, "monster_shop.csv")
  write_csv (nama_folder, monster, "monster.csv")
  write_csv (nama_folder, user, "user.csv")

  output("Data telah tersimpan pada folder", nama_folder)

```

```

procedure updateUser(dataUser:dict, dataSemuaUser:dict)
  userId <- str(dataUser["ID"])
  userName <- dataUser["Username"]

```

```

password <- dataUser["Password"]
role <- dataUser["Role"]
oc <- dataUser["OC"]
if userId not in dataSemuaUser["ID"] then
    dataSemuaUser["ID"].append(userId)
    dataSemuaUser["Username"].append(userName)
    dataSemuaUser["Password"].append(password)
    dataSemuaUser["Role"].append(role)
    dataSemuaUser["OC"].append(oc)
else
    index <- int(userId)-1
    dataSemuaUser["Username"][index] <- userName
    dataSemuaUser["Password"][index] <- password
    dataSemuaUser["Role"][index] <- role
    dataSemuaUser["OC"][index] <- oc

```

```

function updateItem(potionInvent:dict, ballUser:dict, userId:int)
    invent <- readcsvInvent("item_inventory")
    listUser <- []
    listType <- []
    listQuantity <- []
    for barang in invent
        listUser.append(barang[0])
        listType.append(barang[1])
        listQuantity.append(barang[2])
    itemUser <- potionInvent
    itemUser["Type"].append("Monster Ball")
    itemUser["Quantity"].append(str(ballUser["Ball"]))
    [i, id] traversal (listUser)
        if id = str(userId) then
            [j, typeItem] traversal (itemUser["Type"])
                if listType[i] = typeItem then
                    listQuantity[i] <- itemUser["Quantity"][j]
                    break
    itemInvent <- {
        "UserID" : listUser,
        "Type" : listType,
        "Quantity" : listQuantity
    }
    return itemInvent

```

```

function updateMonster(monsterUser:dict, userId:int)
    invent <- readcsvInvent("monster_inventory")
    listUser <- []
    listMonster <- []
    listLevel <- []
    monsterInvent <- [] #list untuk menampung monster user yang ada
    di database
    barang traversal invent
        listUser.append(barang[0])
        listMonster.append(barang[1])
        listLevel.append(barang[2])
    [i, id] traversal (listUser)
        if id = str(userId) then
            monsterInvent.append(listMonster[i])
    [i, id] traversal (listUser)
        if id = str(userId) then
            [j, monsterID] traversal (monsterUser["MonsterID"])
                if monsterID not in monsterInvent then
                    listUser.append(id)

```

```

        listMonster.append(monsterID)
        listLevel.append(monsterUser["Level"][j])
        monsterInvent.append(monsterID)
        break
    if listMonster[i] == monsterID then
        listLevel[i] <- monsterUser["Level"][j]
monsterInvent <- {
    "UserID" : listUser,
    "MonsterID": listMonster,
    "Level": listLevel
}
return monsterInvent

```

18. F16 – Exit

a. Desain Spesifikasi

```

from src.share import loadingmsg
from time import sleep
procedure exit():

```

ALGORITMA

```

    output()
    sleep(2)
    loadingmsg("Exiting")
    output('{: ^120s}'.format(""*120))
    output()
    output('{: ^120s}'.format("Terima kasih telah bermain!"))
    output('{: ^120s}'.format("""
SSS eee eee yy yy ooo uu uu S eeee eeee
yy yy oo oo uu uu SSS eee eee yyy oo oo uu
uu S e e y oo oo uu uu SSS eee
eee y ooo uuu uuu """))
    output('{: ^120s}'.format(""*120))

```

19. B02 – Typing

```

> def arena(dataUser:dict, potionUser:dict, monsterUser:dict): ...

> def enterArena(dataUser:dict, monsterUser:dict) -> dict: # Dipanggil saat memasuki arena pertama kali ...

    def battleArena(potionUser:dict,
> | | | | statAgent:dict, arena:int, hasilLatihan:dict, dataRonde:list, dataPotion:dict, dataUser:dict, maxHpAgent:int) -> bool: ...

    def turnArena(hasilLatihan:dict,
> | | | | dataPotion:dict, number:int, allies:dict, enemies:dict, status:list, potionUser:dict, maxHpMusuh:int, maxHpAgent:int) -> bool: ...

> def turnMusuhArena(hasilLatihan:dict, number:int, allies:dict, enemies:dict): ...

> def attackArena(Atk:int, Def:int, attackerName:str, defenderName:str, defender:dict) -> int: ...

> def usePotionArena(status:list, dataPotion:dict, potionUser:dict, maxPilihan:int, allies:dict, maxHp:int) -> bool: ...

> def check(isEscape:bool, agent:dict, musuh:dict) -> list[bool, int]: ...

> def battle(dataUser:dict, potionUser:dict, monsterUser:dict, monsterBall:dict, ambush:bool=False) -> bool: ...

> def encounter(userId:int, namaUser:str, monsterUser:dict) -> list: ...

> def showStat(stat:dict, maxHp:int=None) -> str: ...

    def turnAlly(number:int,
> | | | | monsterUser:dict, monsterBall:dict, allies:dict, enemies:dict, status:list, potionUser:dict, maxHpMusuh:int, maxHpAgent:int) -> bool: ...

> def turnEnemy(number:int, allies:dict, enemies:dict) -> bool: ...

> def catch(enemies:dict, monsterUser:dict, monsterBall:dict): ...

> def randomCatch(levelMonster:int) -> bool: ...

> def attack(Atk:int, Def:int, attackerName:str, defenderName:str, defender:list): ...

> def usePotion(status:list, potionUser:dict, maxPilihan:int, allies:dict, maxHp:int) -> bool: ...

> def check(isEscape:bool, isCatch:bool, agent:dict, musuh:dict) -> list[bool, int]: ...

> def inventory(dataUser:dict, monsterUser:dict, potionUser:dict, monsterBall:dict[int]): ...

> def monsterInventory(monsterUser:dict) -> list[list[str], list[dict]]: ...

> def ballInventory(monsterBall:dict) -> str: ...

> def potionInventory(potionUser:dict) -> list[list[str], list[list[str]]]: ...

def jackpot(dataUser:dict, dataMonster:dict, monsterUser:dict): ...

def displayJackpot(hasil1:str, hasil2:str, hasil3:str): ...

> def laboratory(userId:int, dataUser:dict, monsterUser:dict): ...

> def labMenu(userId:int, userName:str, OC:int, monsterUser:dict, jumlahPilihan:int): ...

> def pilihMonsterLab(monsterUser:dict, jumlahPilihan:int) -> int: ...

> def upgrade(pilihan:int, monsterId:int, levelMonster:int, OC:int, data:dict, monsterUser:list): ...

> def statMonster(monsterId:int, statIndex:int) -> str: ...

> def get_stats(id:int, level:int) -> dict: ...

> def monsterList(userId:int, data:dict=None) -> list[str]: ...

> def pilihMonster(userId:int, monsterUser:dict, withList:bool=False) -> int: ...

> def banyakMonster() -> int: ...

> def level(monsterId:int, data:dict) -> int: ...

```

```

> def monsterManagement(dataMonster:dict): ...
> def showMonsterData(file:dict, columnLen:list[int]=None): ...
> def makeRow(nilai:str, columnLen:int, id:int): ...
> def buatMonster(data:dict) -> list[bool, list[int, str]]: ...
> def inputDEF() -> int: ...

```

```
def peta(lokasituple[int]=(0,0)) -> list[str, tuple[int]]: ...
```

```
def isAmbush() -> bool: ...
```

```
def drawPeta(lokasituple[int]) -> list[str]: ...
```

```
def move(peta:list, arah:str, lokasituple[int]) -> bool: ...
```

```
def showOption(options:list[str]): ...
```

```
def setOptionMove(lokasituple[int], dataPilihan:list): ...
```

```
def locate(peta:list[str]) -> tuple[int]: ...
```

```
def pilihanSalah(pilihan:str): ...
```

```
def pilihanSalahArah(pilihan:str): ...
```

```
def surrounding(peta:list, lokasituple[int], dataPilihan:list): ...
```

```
def random(a:int=48271, c:int=0, m:int=2**31-1, n:int=None, seed=None, numRange:list[int]=None) -> int: #
```

```
def write_csv (nama_folder:str, data:dict, nama_file:str): ...
```

```
def save (user:dict, monster:dict, monster_shop:dict, monster_inventory:dict, item_shop:dict, item_inventory:dict): ...
```

```
def updateUser(dataUser:dict, dataSemuaUser:dict): ...
```

```
def updateItem(potionInvent:dict, ballUser:dict, userId:int): ...
```

```
def updateMonster(monsterUser:dict, userId:int): ...
```

```
> def login(dataUser) -> dict: ...
```

```
> def isLogin(dataUser): ...
```

```
def logout(dataUser:dict): ...
```

```
> def beforeLogin(): ...

> def afterLogin(role, dataUser): ...

> def menu(dataUser:dict): ...
```

```
> def read_csv(nama_folder:str, cnt_kolom:str, nama_file:str) -> list: ...

> def load() -> dict: ...

> def readcsvData(fileName:str) -> list[list[str]]: ...

> def readcsvInvent(fileName:str, userId:int=None) -> list[list[str]]: ...

> def monster() -> dict: ...

> def monsterShop() -> dict: ...

> def user(userId:int = None) -> dict: ...

> def itemShop() -> dict: ...

> def monsterInventory(userId:int) -> dict: ...

> def potionInventory(userId:int) -> dict: ...

> def ballInventory(userId:int) -> dict: ...

> def loadData(data:str=None, userId:int=None) -> dict: ...

> def loadInvent(userId:int, data:str=None) -> dict: ...

> def getDataUser(userId:int) -> dict: ...

> def getBall(userId:int) -> dict: ...

> def showDict(data:dict): ...
```

```
def exit(): ...
```

```
def shop_management(monsterShop:dict, itemShop:dict, monsterList:dict, itemList:dict): ...

def add_monster(monsterList:dict, monsterShop:dict): ...

def add_item(itemList:dict, itemShop:dict): ...

def update_monster(monsterList:dict, monsterShop:dict): ...

def update_item(itemShop:dict): ...

def delete_monster(monsterList:dict, monsterShop:dict): ...

def delete_item(itemShop:dict): ...
```

```

def shop(dataUser:dict, monsterShop:dict, itemShop:dict, potionInvent:dict, dataMonster:dict, monsterInvent:dict, ballUser:dict) -> dict: ...
def show_monsters(monsterShop:dict, monster_data:dict): ...
def show_items(itemShop:dict): ...
def buy_monster(pilihan: int, owca_coin: int, monsterShop: dict, monster_data: dict, userId: int, monsterInventory: dict) -> int: ...
def buy_item(id:int, quantity:int, owca_coin:int, itemShop:dict, potionInvent:dict, ballUser:dict, userId:int) ->int: ...

def getPotion(userId:int): ...

def potionStatus(potionUser): ...

def tambahPotion(type:str, quanntity:int, dataPotion:dict): ...

def potionList(potionUser:dict) -> int: ...

def updatePotionUser(potionUser:dict, potionInvent:dict): ...

def updatePotionInvent(potionUser:dict, potionInvent:dict): ...

```

20. B03 - Monster Ball

a. Desain Kamus

KAMUS LOKAL {Deklarasi Variabel}

namaMonster : string

monsterId, levelMonster, levelMonster, chance, number : integer

isCatch : boolean

enemies, monsterUser, monsterBall, data : dictionary

b. Desain Spesifikasi

```

from src.monster import get_stats
from src.battle import showStat
function catch(enemies, monsterUser, monsterBall) -> boolean:
{Spesifikasi : function catch digunakan saat user ingin menangkap monster
yang diencounter }
{I.S : saat user memilih menggunakan monster ball}
{F.S : saat user berhasil/tidak berhasil menangkap monster}

```

ALGORITMA

```

levelMonster <- enemies["Level"]
namaMonster <- enemies["Name"]
monsterId <- str(enemies["ID"])
while True do
    if monsterBall["Ball"] = 0 then
        output("Anda tidak memiliki Monster Ball dalam inventory!")
        return False
    else if monsterId in monsterUser["MonsterID"] then
        output(f"Anda sudah memiliki monster {namaMonster} dalam
inventory!")
        return False
    else
        isCatch <- randomCatch(levelMonster)
        monsterBall["Ball"] - 1
        if isCatch then
            output(f"{namaMonster} Berhasil diitangkap !!! ")
            monsterUser["MonsterID"].append(monsterId)
            monsterUser["Level"].append(str(levelMonster))
            stat <- get_stats(monsterId, levelMonster)
            showStat(stat)

```

```

        output(f"Sisa Monster Ball Anda : {monsterBall["Ball"]}")
        return True
    else
        output(f"Yah anda belum berhasil mendapatkan monster
{namaMonster} !!! ")
        output(f"Sisa Monster Ball Anda : {monsterBall["Ball"]}")
        return False

```

function randomCatch(levelMonster) -> **boolean**:
 {Spesifikasi : function randomCatch akan menentukan apakah user berhasil atau tidak dalam penangkapan }
 {I.S : saat user memilih untuk melakukan penangkpan monster}
 {F.S : mengembalikan hasil penangkapan}
ALGORITMA
 number <- random(numRange=[0,100])
 data <- {
 "1": 75,
 "2": 50,
 "3": 25,
 "4": 10,
 "5": 5
 }
 chance <- data[str(levelMonster)]
return number < chance

21. B04 – Jackpot

a. Desain Kamus

KAMUS LOKAL {Deklarasi Variabel}
 banyakMonster, max, hasil1, hasil2, hasil3, monsterId, count,
 hadiah, side : **integer**
 barang1, barang2, barang3, hasil1, hasil2, hasil3, bar,
 hasil, content : **string**
 isMain : **boolean**
 barang : **array of string**
 dataUser, data, dataMonster, monsterUser : **dictionary**

b. Desain Spesifikasi

```

from src.load import showDict, load, loadInvent
from src.share import displayBar, YesOrNo, clear
from src.rng import random
procedure jackpot(dataUser, dataMonster, monsterUser):

{Spesifikasi : prosedur jackpot akan meminta input user untuk bermain dan
setiap kali bermain OC user akan dikurangi 600 dan user mendapat hadiah
secara random}
{I. S Agent dapat memainkan permainan jackpot untuk mendapatkan item atau
monster menggunakan OC (O.W.C.A Coin)}
{F. S Program akan menampilkan daftar item dengan nilai OC masing-masing
dan meminta konfirmasi dari Agent untuk memulai permainan. Jika Agent
memilih untuk bermain, program akan memeriksa OC yang dimiliki. Jika
tidak cukup, program akan menampilkan pesan. Jika cukup, program akan
mengurangi OC sesuai biaya permainan, mengacak 3 item, dan menampilkan
hasilnya. Jika ketiga item sama (jackpot), Agent mendapatkan monster acak
yang ditambahkan ke inventory. Jika tidak jackpot, nilai OC dari ketiga
item dijumlahkan dan ditambahkan ke OC Agent. Program akan menampilkan
hasil permainan dan meminta konfirmasi apakah Agent ingin bermain lagi.}

ALGORITMA
    clear()

```


[illegible]


```

setOptionMove(lokalasi, dataPilihan)
surrounding(peta, lokasi, dataPilihan)
showOption(dataPilihan)
while True do
    pilihan <- str.upper(input("<///> "))
    if pilihan not in dataPilihan then
        pilihanSalah(pilihan)
        continue
    break
if pilihan in ["U", "D", "L", "R"] then
    isValidMove <- move(peta, pilihan, lokasi)
    if not isValidMove then
        continue
    if isAmbush() then
        return ["AM", lokasi]
    continue
return [pilihan, lokasi]

```

```

from src.rng import random
function isAmbush() -> boolean:
{Spesifikasi : function isAmbush akan menentukan apakah user terkena
ambush pada move ini}
{I.S : saat user bergerak pada peta}
{F.S : saat monster dikalahkan/ditangkap atau user kalah}
ALGORITMA
    rngAmbush <- random(numRange=[0,100])
    chance <- 5
    return rngAmbush < chanc

```

```

function drawPeta(lokalasi) -> array of string:
{Spesifikasi : fungsi untuk menaruh posisi player pada peta}
{I.S : sebelum melakukan move}
{F.S : setelah player dimasukkan pada data}
ALGORITMA
    (x, y) <- lokasi
    data <- ["#####",
            "####S##X##",
            "#####X##",
            "####J##X#L",
            "#####XXXX##",
            "#X#####",
            "#X#####",
            "#XXX##A###",
            "#####",
            "#####XXXXX#"]
    data[x] <- data[x][:y] + "P" + data[x][y+1:]
    return data

```

```

function move(peta, arah, lokasi) -> boolean:
{Spesifikasi : function untuk melakukan pergerakan pada peta}
{I.S : saat user memasukkan L R U atau D}
{F.S : saat posisi player bergerak}
ALGORITMA
    (x, y) <- lokasi
    depend on arah
    arah = "U":
        if peta[x-1][y] = "#" then

```

```

        peta[x] <- peta[x][:y] + "#" + peta[x][y+1:]
        peta[x-1] <- peta[x-1][:y] + "P" + peta[x-1][y+1:]
        return True
    else
        pilihanSalahArah(peta[x-1][y])
        return False
arah = "D":
    if peta[x+1][y] == "#" then
        peta[x] <- peta[x][:y] + "#" + peta[x][y+1:]
        peta[x+1] <- peta[x+1][:y] + "P" + peta[x+1][y+1:]
        return True
    else
        pilihanSalahArah(peta[x+1][y])
        return False
arah = "L":
    if peta[x][y-1] == "#" then
        peta[x] <- peta[x][:y] + "#" + peta[x][y+1:]
        peta[x] <- peta[x][:y-1] + "P" + peta[x][y:]
        return True
    else
        pilihanSalahArah(peta[x][y-1])
        return False
arah = "R":
    if peta[x][y+1] == "#" then
        peta[x] <- peta[x][:y] + "#" + peta[x][y+1:]
        peta[x] <- peta[x][:y+1] + "P" + peta[x][y+2:]
        return True
    else
        pilihanSalahArah(peta[x][y+1])
        return False

```

procedure showOption(options):

{Spesifikasi : procedure showOption akan menampilkan ke layar opsi yang dimiliki user }

{I.S : sebelum melakukan move}

{F.S : menampilkan data ke layar}

ALGORITMA

```

    hasil <- ""
    opsi traversal options
    depend on opsi:
        opsi = "U":
            hasil + "U : Move Up, "
        opsi = "D":
            hasil + "D : Move Down, "
        opsi = "L":
            hasil + "L : Move Left, "
        opsi = "R":
            hasil + "R : Move Right, "
        opsi = "B":
            hasil + "B : Enter Battle, "
        opsi = "S":
            hasil + "S : Enter Shop, "
        opsi = "LA":
            hasil + "LA : Enter Laboratory, "
        opsi = "A":
            hasil + "A : Enter Arena, "
        opsi = "J":
            hasil + "J : Enter Jackpot, "
        opsi = "I":
            hasil + "I : Open Inventory, "

```

```
opsi = "M":  
    hasil + "M : Open Menu, "  
output(f"({hasil[:2]})")
```

```
procedure setOptionMove(lokalasi, dataPilihan):  
{Spesifikasi : procedure setOptionMove akan menentukan move yang valid  
berdasarkan lokasi user}  
{I.S : sebelum melakukan showOption}  
{F.S : data pilihan di update}  
ALGORITMA  
    if lokasi[0] = 0 then  
        dataPilihan.append("D")  
    else if lokasi[0] = 9 then  
        dataPilihan.append("U")  
    else  
        dataPilihan.append("U")  
        dataPilihan.append("D")  
    if lokasi[1] = 0 then  
        dataPilihan.append("R")  
    else if lokasi[1] = 9 then  
        dataPilihan.append("L")  
    else  
        dataPilihan.append("R")  
        dataPilihan.append("L")
```

```
function locate(peta) -> tuple of integer:  
{Spesifikasi : function locate untuk menentukan posisi user pada peta}  
{I.S : setelah melakukan move}  
{F.S : menghasilkan tuple lokasi}  
ALGORITMA  
    [i, row] tranversal enumerate(peta)  
    [j, ele] tranversal enumerate(row)  
    if ele = "P" then  
        return (i, j)
```

```
from src.share import sleep  
procedure pilihanSalah(pilihan):  
{Spesifikasi : saat user memasukkan input yang tidak ada pada showOption}  
{I.S : saat user melakukan move}  
{F.S : menampilkan pesan kesalahan}  
ALGORITMA  
    depend on pilihan:  
    pilihan = "U":  
        output("You Can't Move Up")  
    pilihan = "D":  
        output("You Can't Move Down")  
    pilihan = "L":  
        output("You Can't Move Left")  
    pilihan = "R":  
        output("You Can't Move Right")  
    pilihan = "B":  
        output("You Can't Enter Battle")  
    pilihan = "S":
```

```

    output("You Can't Enter Shop")
    pilihan = "LA":
    output("You Can't Enter Laboratory")
    pilihan = "A":
    output("You Can't Enter Arena")
    pilihan = "J":
    output("You Can't Enter Jackpot")
    pilihan = "I":
    output("You Can't Open Inventory")
    pilihan = "M":
    output("You Can't Open Menu")
    default:
    output("Pilihan tidak valid")
    sleep(2)

```

```

from src.share import sleep
procedure pilihanSalahArah(pilihan):
{Spesifikasi : procedure pilihanSalahArah akan menampilkan pesan
kesalahan saat user melakukan move ke suatu obstacle }
{I.S : saat user melakukan move}
{F.S : pesan kesalahann ditampilkan ke layar}
ALGORITMA
    depend on pilihan:
    pilihan = "X":
    output("You Can't Enter a Bush")
    pilihan = "S":
    output("Your Head Bump into Shop. To access it enter command")
    pilihan = "LA":
    output("Your Head Bump into Laboratory. To access it enter
command")
    pilihan = "J":
    output("Your Head Bump into Jackpot. To access it enter command")
    pilihan = "A":
    output("Your Head Bump into Arena. To access it enter command")
    sleep(2)

```

```

procedure surrounding(peta, lokasi, dataPilihan):
{Spesifikasi : procedure surrounding untuk menentukan tempat yang ada
disekitar user untuk ditambahkan ke data pilihan }
{I.S : saat user akan move}
{F.S : mengupdate data pilihan}
ALGORITMA
    (x, y) <- lokasi
    if x = 0 or x = 9 then
        if x = 9 then
            up <- peta[x-1][y]
            down <- "#"
        else
            up <- "#"
            down <- peta[x+1][y]
    else
        up <- peta[x-1][y]
        down <- peta[x+1][y]

    if y = 0 or y = 9 then
        if y = 9 then
            left <- peta[x][y-1]
            righth <- "#"

```

```

        else
            left <- "#"
            righth <- peta[x][y+1]
        else
            left <- peta[x][y-1]
            righth <- peta[x][y+1]
        data <- [up, down, left, righth]
        if "X" in data then
            dataPilihan.append("B")
        if "S" in data then
            dataPilihan.append("S")
        if "L" in data then
            dataPilihan.append("LA")
        if "A" in data then
            dataPilihan.append("A")
        if "J" in data then
            dataPilihan.append("J")

```

Hasil Pengujian Program

1. F00 - Random Number Generator

Hasil generate fungsi random dengan range 1 sampai dengan 100.

```

if1210-2024-tubes-k11-d python -u "c:\00_Nadhif\03_Kuliah\01_Kampus\02_Semester 2\01_Dasar Pemograman\Tubes\if1210-2024-tubes-k11-d\src\rng.py"
30
89
50
10
70
file:C:\00_Nadhif\03_Kuliah\01_Kampus\02_Semester 2\01_Dasar Pemograman\Tubes\if1210-2024-tubes-k11-d (ctrl + click)
Asus Store Sakura if1210-2024-tubes-k11-d integrasi ~4

```

2. F01 – Register

Tampilan saat user berhasil register:

===== 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. Exit : Keluar dari game

Footnote:

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

Masukkan pilihan (1/2/3): 2

Masukkan nama: tes

Masukkan password: 123

Silahkan pilih salah satu monster sebagai monster awalmu.

1. Pikachu
2. Bulbasaur
3. Charmander
4. Squirtle
5. Jigglypuff
6. Eevee
7. Snorlax
8. Mewtwo
9. Psyduck
10. Meowth
11. tes

Monster pilihanmu: 2

Selamat datang Agent tes. Mari kita mengalahkan Dr. Asep Spakbor dengan Bulbasaur!

█

Tampilan saat memasukkan nama dari user yang sudah ada

Masukkan pilihan (1/2/3): 2

Masukkan nama: tes

Masukkan password: 123

Username tes sudah terpakai, silahkan gunakan username lain!

Masukkan nama: █

3. F02 – Login

Tampilan saat akan login:

Masukan username: t

Masukan password: █

Tampilan saat berhasil login:

Anda berhasil login dengan username t

█

Tampilan saat username tidak ada:


```
Masukan username: te
Username tidak ditemukan
Keluar (Y/N):
```

Tampilan saat password salah:

```
Masukan username: t
Masukan password: sasd
Password salah!
```

4. F03 – Logout

Tampilan saat logout:

```
1. Logout : Keluar dari akun yang sedang digunakan
2. Bermain : Memasuki peta kota
3. Save : Menyimpan data saat ini
4. Exit : Keluar dari game
<///> Pilih perintah (1/2/3/4) 1
Apa anda ingin menyimpan file sebelum logout ? (Y/N): n
Logout berhasil
Anda tidak melakukan penyimpanan file. Perubahan tidak tersimpan.
Exit (Y/N):
```

5. F04 - Menu & Help

Tampilan menu saat belum login:

```
===== 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. Exit : Keluar dari game

Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
Masukkan pilihan (1/2/3):
```

Tampilan menu saat login sebagai agent:

```
<===== MAIN MENU <=====>
Halo Agent t. Kamu memasuki 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. Bermain : Memasuki peta kota
3. Save : Menyimpan data saat ini
4. Exit : Keluar dari game
<///> Pilih perintah (1/2/3/4)
```

Tampilan menu saat login sebagai admin:

```
<===== MENU <=====>
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 : Menambahkan Monster baru
4. Exit : Keluar dari game
<///> Pilih perintah (1/2/3)
```

6. F05 – Monster

Tampilan monster list:

```

<=====> MONSTER LIST <=====>
1. Bulbasaur    (Lvl: 5)
2. Charmander   (Lvl: 4)
3. Squirtle     (Lvl: 5)
4. Jigglypuff   (Lvl: 1)
5. Pikachu     (Lvl: 1)
6. tes         (Lvl: 1)
<///> Pilih monster untuk bertarung : 

```

Tampilan data Monster:

```

<///> Pilih perintah: 1
ID | Type       | ATK_power | DEF_power | HP   |
1  | Pikachu    | 50        | 40        | 320  |
2  | Bulbasaur  | 250       | 45        | 900  |
3  | Charmander | 52        | 43        | 320  |
4  | Squirtle   | 48        | 25        | 310  |
5  | Jigglypuff | 45        | 20        | 570  |
6  | Eevee      | 55        | 50        | 340  |
7  | Snorlax    | 110       | 20        | 680  |
8  | Mewtwo     | 110       | 10        | 680  |
9  | Psyduck    | 52        | 48        | 320  |
10 | Meowth     | 45        | 35        | 290  |
11 | tes        | 20        | 10        | 20   |
<///> Keluar (Y/N): 

```

7. F06 – Potion

Tampilan potion list:

```

<///> Pilih perintah: 2
<=====> POTION LIST <=====>
1. Healing Potion    (Qty: 5) - Restore Health
2. Strength Potion   (Qty: 4) - Increase ATK Power
3. Resilience Potion (Qty: 3) - Increase DEF Power
4. Cancel
<///> Pilih potion: 1
Healing potion digunakan

```

8. F07 – Inventory

Tampilan inventory:

```

<=====> User Info <=====>
User ID : 6
Nama    : t
OC      : 321

<=====> Inventory List <=====>
1. Monster      (Name: Bulbasaur, Lvl: 5, HP: 1260)
2. Monster      (Name: Charmander, Lvl: 4, HP: 416)
3. Monster      (Name: Squirtle, Lvl: 5, HP: 434)
4. Monster      (Name: Jigglypuff, Lvl: 1, HP: 570)
5. Monster      (Name: Pikachu, Lvl: 1, HP: 320)
6. Monster      (Name: tes, Lvl: 1, HP: 20)
7. Potion       (Type: Heal, Qty: 5)
8. Potion       (Type: ATK, Qty: 4)
9. Potion       (Type: DEF, Qty: 3)
10. Monster Ball (Qty: 100)
11. Keluar
Ketikkan id untuk menampilkan item
<///> : 

```

Tampilan saat memilih tidak valid:

```

Ketikkan id untuk menampilkan item
<///> : 11
Masukan tidak valid
<///> Pilih perintah: 

```

Tampilan saat memilih monster:

```

Monster
<=====>
Name      : Bulbasaur
ATK Power : 350
DEF Power : 62
HP        : 1260/1260
Level     : 5
<=====>
<///> Keluar Inventory (Y/N): 

```

Tampilan saat memilih potion:

```

Potion
<=====>
Type      : Heal
Quantity  : 5
<=====>
<///> Keluar Inventory (Y/N): 

```

Tampilan saat memilih monster ball:

```
Monster Ball
Quantity : 100
<///> Keluar Inventory (Y/N): █
```

9. F08 – Battle

Tampilan saat musuh spawn:

```

      .-.-.-.
     /      \
    | 0  0   |
    |.vvvvv. |
    |~~~~~|
   /  /  /  \
  /  /  /  \
 /  /  /  \
/  /  /  \
-.-.-.-.-
      \uU \uU/
      :F_P:

RAWRR, Monster Squirtle telah muncul !!!
<=====>
Name      : Squirtle
ATK Power : 52
DEF Power : 27
HP        : 341/341
Level     : 2
<=====>
```

Tampilan saat memilih monster:

```

<=====> MONSTER LIST <=====>
1. Bulbasaur   (Lvl: 5)
2. Charmander  (Lvl: 4)
3. Squirtle    (Lvl: 5)
4. Jigglypuff  (Lvl: 1)
5. Pikachu     (Lvl: 1)
6. tes         (Lvl: 1)
<///> Pilih monster untuk bertarung : █
```

Tampilan saat memasuki turn:

```

<=====>
Name      : Mewtwo
ATK Power : 143
DEF Power : 13
HP        : 884/884
Level     : 4
<=====>

VS

<=====>
Name      : Bulbasaur
ATK Power : 350
DEF Power : 62
HP        : 1204/1260
Level     : 5
<=====>
<=====> Turn 1 (Bulbasaur) <=====>
1. Attack
2. Use Potion
3. Use Monster Ball
4. Escape
<///> Pilih perintah: 2

```

Tampilan saat memilih attack:

```

<///> Pilih perintah: 1
Bulbasaur attack Squirtle dealing 193 damage !!!
<=====> Turn 1 (Squirtle) <=====>
Squirtle attack Bulbasaur dealing 22 damage !!!

```

Tampilan saat memilih menggunakan potion:

```

<///> Pilih perintah: 2
<=====> POTION LIST <=====>
1. Healing Potion      (Qty: 5) - Restore Health
2. Strength Potion     (Qty: 4) - Increase ATK Power
3. Resilience Potion  (Qty: 3) - Increase DEF Power
4. Cancel
<///> Pilih potion: 1
Healing potion digunakan

```

Tampilan saat user memilih potion yang sudah habis:

```
<///> Pilih perintah: 2
<=====> POTION LIST <=====>
1. Resilience Potion (Qty: 7) - Increase DEF Power
2. Healing Potion (Qty: 0) - Restore Health
3. Cancel
<///> Pilih potion: 2
Healing potion sudah habis
<///> Pilih potion: 
```

Tampilan saat user memilih potion yang sudah digunakan:

```
<///> Pilih perintah: 2
<=====> POTION LIST <=====>
1. Healing Potion      (Qty: 4) - Restore Health
2. Strength Potion     (Qty: 4) - Increase ATK Power
3. Resilience Potion  (Qty: 3) - Increase DEF Power
4. Cancel
<///> Pilih potion: 1
sudah digunakan
<///> Pilih potion: 
```

Tampilan attack pertama saat terkena ambush:

```
<=====> Turn 0 (Mewtwo) <=====>
Mewtwo attack Bulbasaur dealing 56 damage !!!
```

Tampilan saat user escape:

```
<=====>
Anda berhasil kabur dari Battle!
<=====>
Sedang respawn...
```

Tampilan saat user kalah:

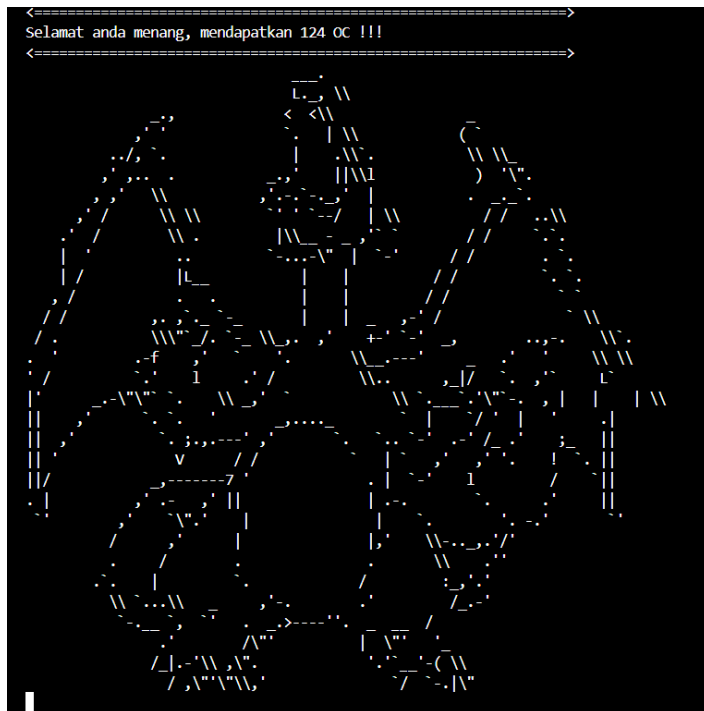
```

<=====
Sayang sekali anda kalah
<=====

      ;-.
     / \  .... / \  ....
    /   \      /   \
   /()   () \   /   \
  /)      ;   () \   /
 /  -'    ;;    <  \
;  _      ;;    >  \
/ ,      / ,    |  >
( /      ( /    ;; |  <
 \       \      ;
  >       \      /
 ( -'    >      /
      \      /

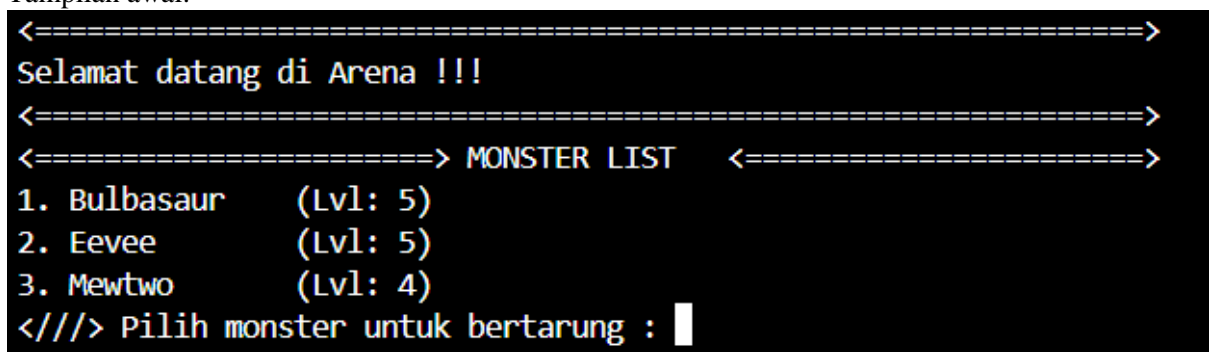
```

Tampilan saat user menang:

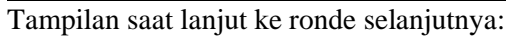


10. F09 – Arena

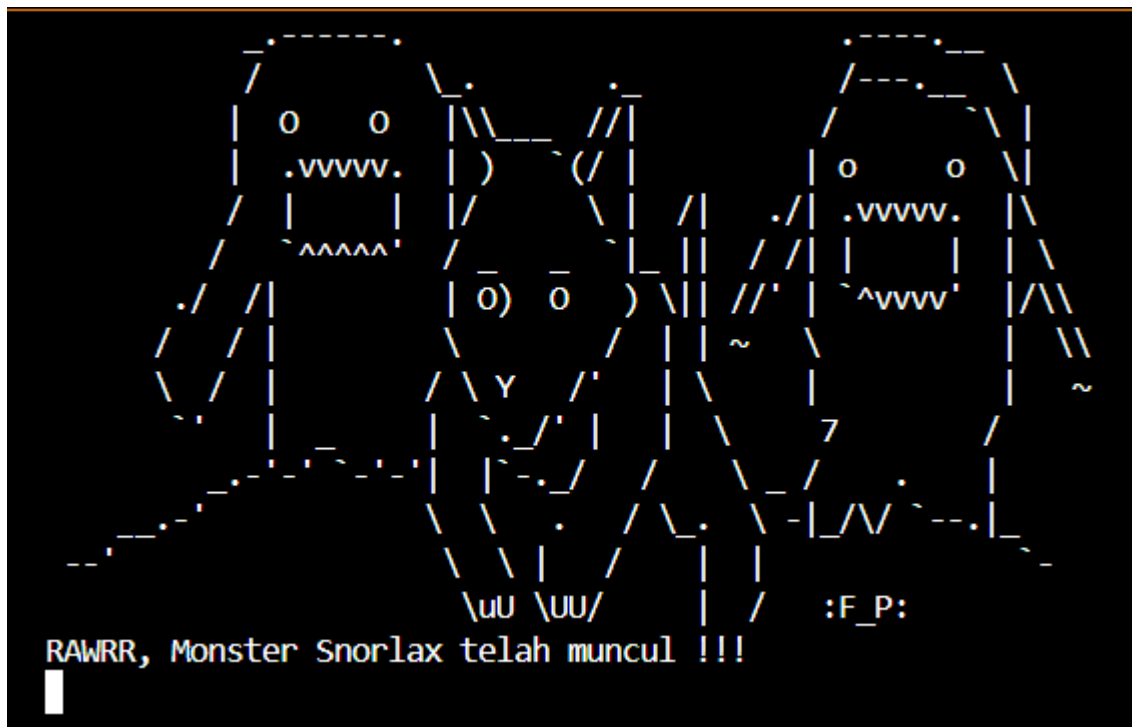
Tampilan awal:



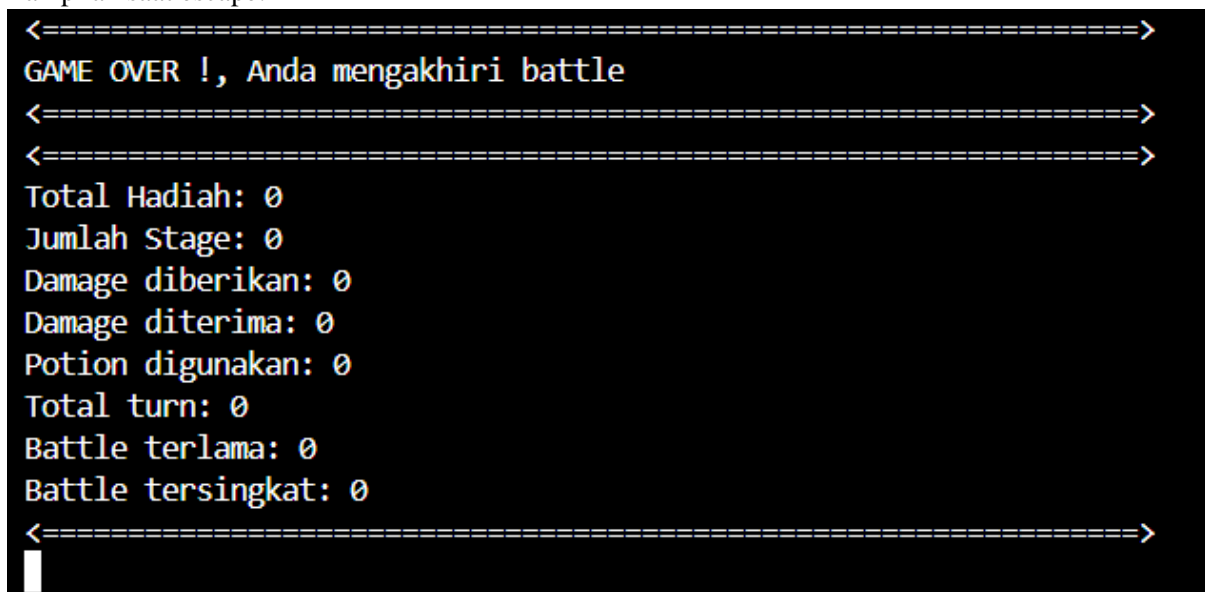
Tampilan memasuki ronde:



```
<=====>
Selamat anda telah menyelesaikan ronde 1, mendapatkan 30 OC !!!
<=====>
Memasuki Ronde 2...
█
```

Tampilan saat escape:



Tampilan saat kalah:

```

<=====>
GAME OVER !, Anda kalah
<=====>
<=====>
Total Hadiah: 0
Jumlah Stage: 0
Damage diberikan: 12
Damage diterima: 47
Potion digunakan: 0
Total turn: 0
Battle terlama: 0
Battle tersingkat: 0
<=====>
█

```

Tampilan saat Menang:

```

<=====>
CONGRATS !, Anda Menang
<=====>
<=====>
Total Hadiah: 780
Jumlah Stage: 5
Damage diberikan: 3037
Damage diterima: 242
Potion digunakan: 0
Total turn: 14
Battle terlama: 4
Battle tersingkat: 2
<=====>
█

```

11. F10 - Shop & Currency

Tampilan saat membeli item:

```

>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 1321.
>>> Mau beli apa? (monster/item): item
>>> Masukkan id item: 4
>>> Masukkan jumlah: 100
Berhasil membeli item: 100 Monster Ball. Item sudah masuk ke inventory-mu!
>>> Pilih aksi (lihat/beli/keluar): █

```

12. F11 – Laboratory

Tampilan awal:

```

Selamat datang di Lab Dokter Asep. Agent t !!!
<=====> MONSTER LIST <=====>
1. Bulbasaur (Lvl: 5)
2. Charmander (Lvl: 4)
3. Squirtle (Lvl: 5)
4. Jigglypuff (Lvl: 1)
5. Pikachu (Lvl: 1)
6. tes (Lvl: 1)
7. Cancel
<=====> 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
Anda memiliki 553 OC
<///> Pilih monster: █

```

Tampilan jika monster sudah max level:

```

<///> Pilih monster: 1
max level
<///> Pilih monster: █

```

Tampilan jika OC tidak cukup:

```

Anda hanya memiliki 553 OC
butuh 1000 OC untuk mengupgrade Charmander
<///> Keluar (Y/N): █

```

Tampilan jika monster diupgrade:

```

<=====>
Jigglypuff akan di-upgrade ke level 2
Harga untuk melakukan upgrade Jigglypuff adalah 300 OC
Saat ini anda memiliki 397 OC
<=====>
<///> Lanjutkan upgrade (Y/N): █

```

13. F12 - Shop Management

Tampilan awal shop:

```

<=====>
      Irasshaimase! Selamat datang kembali, Mr. Monogram!

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

```

Tampilan saat lihat:

```
>>> Mau lihat apa? (monster/item): monster
```

| ID | Type | ATK Power | DEF Power | HP | Stok | Harga |
|----|------------|-----------|-----------|-----|------|-------|
| 1 | Pikachu | 50 | 40 | 320 | 7 | 500 |
| 2 | Charmander | 52 | 43 | 320 | 3 | 1000 |
| 3 | Squirtle | 48 | 25 | 310 | 8 | 550 |
| 4 | Jigglypuff | 45 | 20 | 570 | 5 | 600 |
| 5 | tes | 20 | 10 | 20 | 19 | 200 |
| 6 | Meowth | 45 | 35 | 290 | 10 | 500 |

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

```
>>> Mau lihat apa? (monster/item): item
```

| ID | Type | Stok | Harga |
|----|--------------|------|-------|
| 1 | Resilience | 5 | 30 |
| 2 | Healing | 20 | 20 |
| 3 | Monster Ball | 60 | 10 |
| 4 | Strength | 10 | 30 |

Tampilan saat ubah item:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/item): item
```

| ID | Type | Stok | Harga |
|----|--------------|------|-------|
| 1 | Strength | 1 | 50 |
| 2 | Resilience | 5 | 30 |
| 3 | Healing | 20 | 20 |
| 4 | Monster Ball | 60 | 10 |

```
>>> Masukkan id item: 2
>>> Masukkan stok baru: 10
>>> Masukkan harga baru: 40
Resilience telah berhasil diubah dengan stok baru sejumlah 10 dan dengan harga baru 40!
```

Tampilan saat ubah 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 | 50 | 40 | 320 | 7 | 500 |
| 2 | Bulbasaur | 250 | 45 | 900 | 4 | 700 |
| 3 | Charmander | 52 | 43 | 320 | 3 | 1000 |
| 4 | Squirtle | 48 | 25 | 310 | 8 | 550 |
| 5 | Jigglypuff | 45 | 20 | 570 | 5 | 600 |
| 6 | tes | 20 | 10 | 20 | 19 | 200 |

```
>>> Masukkan id monster: 1
>>> Masukkan stok baru: 10
>>> Masukkan harga baru: 400
Pikachu telah berhasil diubah dengan stok baru sejumlah 10 dan dengan harga baru 400!
```

Tampilan saat hapus monster:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/item): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1 | Pikachu         | 50        | 40        | 320  | 7    | 500
2 | Bulbasaur       | 250       | 45        | 900  | 4    | 700
3 | Charmander      | 52        | 43        | 320  | 3    | 1000
4 | Squirtle        | 48        | 25        | 310  | 8    | 550
5 | Jigglypuff      | 45        | 20        | 570  | 5    | 600
6 | tes             | 20        | 10        | 20   | 19   | 200

>>> Masukkan id monster: 2
>>> Apakah anda yakin ingin menghapus Bulbasaur dari shop (y/n)? y
```

Tampilan saat hapus item:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/item): item
ID | Type           | Stok | Harga
1 | Strength       | 1    | 50
2 | Resilience    | 5    | 30
3 | Healing        | 20   | 20
4 | Monster Ball   | 60   | 10

>>> Masukkan id item: 1
>>> Apakah anda yakin ingin menghapus Strength dari shop (y/n)? y
Strength telah berhasil dihapus dari shop!
```

Tampilan saat tambah monster:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): monster
ID | Type           | ATK Power | DEF Power | HP   |
2 | Bulbasaur       | 250       | 45        | 900  |
6 | Eevee          | 55        | 50        | 340  |
7 | Snorlax        | 110       | 20        | 680  |
8 | Mewtwo         | 110       | 10        | 680  |
9 | Psyduck        | 52        | 48        | 320  |
10 | Meowth         | 45        | 35        | 290  |

>>> Masukkan id monster: 10
>>> Masukkan stok awal: 10
>>> Masukkan harga: 500
Meowth telah berhasil ditambahkan ke dalam shop!
```

Tampilan saat tambah item:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): item
ID | Type      |
1  | Strength   |

>>> Masukkan id item: 1
>>> Masukkan stok awal: 10
>>> Masukkan harga: 30
Strength telah berhasil ditambahkan ke dalam shop!
```

Tampilan saat tidak ada yang bisa ditambah:

```
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/item): item
Semua item sudah ada dalam shop. Tidak ada yang perlu ditambahkan.
```

14. F13 - Monster Management

Tampilan awal:

```
<=====>
SELAMAT DATANG DI DATABASE PARA MONSTER !!!
1. Tampilkan semua Monster
2. Tambah Monster Baru
<///> Pilih perintah: 1
```

Tampilan list monster:

```
<///> Pilih perintah: 1
ID | Type      | ATK_power | DEF_power | HP      |
1  | Pikachu   | 50        | 40        | 320     |
2  | Bulbasaur | 250       | 45        | 900     |
3  | Charmander | 52       | 43        | 320     |
4  | Squirtle  | 48        | 25        | 310     |
5  | Jigglypuff | 45       | 20        | 570     |
6  | Eevee     | 55        | 50        | 340     |
7  | Snorlax   | 110       | 20        | 680     |
8  | Mewtwo    | 110       | 10        | 680     |
9  | Psyduck   | 52        | 48        | 320     |
10 | Meowth    | 45        | 35        | 290     |
11 | tes       | 20        | 10        | 20      |
<///> Keluar (Y/N):
```

Tampilan pembuatan monster:

```

<=====>
Memulai pembuatan Monster...
<=====>
Masukkan Type / Nama : monstertes
Masukkan ATK Power : kdosfd
Masukkan input bertipe Integer, coba lagi!
<///> Masukan : 100
Masukkan DEF Power (0-50) :hsad
Masukkan input bertipe Integer, coba lagi!
<///> Masukan : 51
DEF Power harus bernilai 0-50, coba lagi!
Masukkan DEF Power (0-50) :49
Masukkan HP : 500
Sedang membuat monster...

```

Tampilan konfirmasi penambahan:

```

<=====>
Monster baru berhasil dibuat!
Type      : monstertes
ATK Power : 100
DEF Power : 49
HP        : 500
<=====>
Tambahkan Monster ke database (Y/N):

```

15. F14 – Load

Tampilan loading:

```

Loading...
*****
                        Selamat Datang di..
*****

      O O O   W       W   CCCC   AAA
      O   O   W       W   C      A  A
      O   O   W   W   W   C      AAAAAA
      O   O   W   W   W   C      A      A
      O O O       W W   CCCC   A      A
*****

```

16. F15 – Save

Tampilan input nama folder untuk menyimpan data:

```

Masukkan nama folder penyimpanan: data

```

Tampilan saving :

```
Saving...
Data telah tersimpan pada folder data
```

17. F16 – Exit

Tampilan exit:

```
Exiting...
*****
                                     Terima kasih telah bermain!

  SSS  eeee  eeee  yy  yy  ooo  uu  uu  a  gg  gg  eeee  nn  nn  ttt
S      e    e    yy yy  oo  oo  uu  uu  a a  g    e    nnn  nn  t
  SSS  eeee  eeee  yyy  oo  oo  uu  uu  aaaa  g  gg  eee  nn  n  nn  t
  S    e    e    y    oo  oo  uu  uu  a  a  g  g  e    nn  nnn  t
  SSS  eeee  eeee  y    ooo  uu  uu  a  a  gg  g  eeee  nn  nn  t
*****
```

18. B03 - Monster Ball

Tampilan saat memilih menggunakan monster ball dan tidak berhasil:

```
<///> Pilih perintah: 3
Yah anda belum berhasil mendapatkan monster Mewtwo !!!
Sisa Monster Ball Anda : 17
```

Tampilkan berhasil menangkap monster:

```
<///> Pilih perintah: 3
Mewtwo Berhasil diitangkap !!!
<=====>
Name      : Mewtwo
ATK Power : 143
DEF Power : 13
HP        : 884/884
Level     : 4
<=====>
Sisa Monster Ball Anda : 13
```

Tampilan saat user tidak punya monster ball:

```
<///> Pilih perintah: 3
Anda tidak memiliki Monster Ball dalam inventory!
```

19. B04 – Jackpot

Tampilan awal:

[illegible]

Tampilan saat tidak mendapat jackpot:

[illegible]

Tampilan saat mendapat jackpot:

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$ Potion | Potion | Potion $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

JACKPOT!!! Selamat, Anda mendapatkan monster Psyduck.

Monster telah ditambahkan ke inventory Anda.

Your OC : 5919

Bermain lagi (Y/N): █

Tampilan saat sudah memiliki semua monster

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$ Monster | Monster | Monster $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
Anda telah mempunyai semua monster. Sebagai gantinya anda mendapatkan 1500 OC !!!
Your OC : 15219
Bermain lagi (Y/N): y

```

Tampilan saat OC tidak cukup:

```
Your OC : 321
Mulai bermain (Y/N): y
Maaf, anda tidak memiliki cukup OC untuk bermain JACKPOT.
```

20. B05 - Peta Kota Daville

Tampilan peta saat user masuk:

```
* * * * *
* P *
* S X *
* X *
* J X L *
* X X X X *
* X *
* X *
* X X X A *
* *
* X X X X X *
* * * * *
Your Location : (0, 0)
(I : Open Inventory, M : Open Menu, D : Move Down, R : Move Right)
<///> █
```

Tampilan saat user bergerak ke batas:

```
* * * * *
* P *
* S X *
* X *
* J X L *
* X X X X *
* X *
* X *
* X X X A *
* *
* X X X X X *
* * * * *
Your Location : (0, 4)
(I : Open Inventory, M : Open Menu, D : Move Down, R : Move Right, L : Move Left, S : Enter Shop)
<///> u
You Can't Move Up
█
```

Tampilan saat user disekitar suatu tempat:

```
* * * * *
* S X *
* P X *
* J X L *
* X X X X *
* X *
* X *
* X X X A *
* *
* X X X X X *
* * * * *
Your Location : (2, 4)
(I : Open Inventory, M : Open Menu, U : Move Up, D : Move Down, R : Move Right, L : Move Left, S : Enter Shop, J : Enter Jackpot)
<///> █
```

Tampilan saat menabrak tempat:

```
* * * * *
*      P      *
*      S      X  *
*              X  *
*      J      X  L *
*      X X X X  *
*      X      *
*      X      *
*      X X X    A  *
*              *
*      X X X X X  *
* * * * *
Your Location : (0, 4)
(I : Open Inventory, M : Open Menu, D : Move Down, R : Move Right, L : Move Left, S : Enter Shop)
<///> d
Your Head Bump into Shop. To acces it enter the command
```

Tampilan saat terkena ambush:

```
<=====>
Anda terkena serangan tiba tiba, bersiaplah !!!
<=====>
```

LAMPIRAN

Asistensi 1

Gambar 1. Hasil Pindaian Asistensi 1

| | | | | | | | | | | | | | |
|--|--|--------------------------|---|---|---------------------------------|---|-----------------------------------|---|---------------------------------|---|-------------------------------------|---|----------------------------------|
| Form MoM Asistensi Tugas Besar IF1210/Dasar Pemrograman Sem. 2 2023/2024 | | | | | | | | | | | | | |
| Nomor Asistensi | : 1 | | | | | | | | | | | | |
| No. Kelompok/Kelas | : K11-D | | | | | | | | | | | | |
| Tanggal asistensi | : 1 Mei 2024 | | | | | | | | | | | | |
| Anggota kelompok | <table><tr><td></td><td>NIM / Nama (Hanya yang Hadir)</td></tr><tr><td>1</td><td>19623300 / Nadhif Al Rozin</td></tr><tr><td>2</td><td>19623200 / Nazwan Siddqi Muttaqin</td></tr><tr><td>3</td><td>19623120 / Hanif Kalyana Aditva</td></tr><tr><td>4</td><td>16523130 / Abidah Izdiyar Dzakiyyah</td></tr><tr><td>5</td><td>13522132 / Hafizh Hananta Akbari</td></tr></table> | | NIM / Nama (Hanya yang Hadir) | 1 | 19623300 / Nadhif Al Rozin | 2 | 19623200 / Nazwan Siddqi Muttaqin | 3 | 19623120 / Hanif Kalyana Aditva | 4 | 16523130 / Abidah Izdiyar Dzakiyyah | 5 | 13522132 / Hafizh Hananta Akbari |
| | NIM / Nama (Hanya yang Hadir) | | | | | | | | | | | | |
| 1 | 19623300 / Nadhif Al Rozin | | | | | | | | | | | | |
| 2 | 19623200 / Nazwan Siddqi Muttaqin | | | | | | | | | | | | |
| 3 | 19623120 / Hanif Kalyana Aditva | | | | | | | | | | | | |
| 4 | 16523130 / Abidah Izdiyar Dzakiyyah | | | | | | | | | | | | |
| 5 | 13522132 / Hafizh Hananta Akbari | | | | | | | | | | | | |
| Asisten pembimbing | <table><tr><td></td><td>NIM / Nama</td></tr><tr><td></td><td>13521114 / Farhan Nabil Survono</td></tr></table> | | NIM / Nama | | 13521114 / Farhan Nabil Survono | | | | | | | | |
| | NIM / Nama | | | | | | | | | | | | |
| | 13521114 / Farhan Nabil Survono | | | | | | | | | | | | |
| Catatan Asistensi: - | | | | | | | | | | | | | |
| <table><tr><td>Rangkuman Diskusi</td></tr><tr><td>Penjelasan spesifikasi tubes dan penjelasan tentang data csv yang digunakan, serta penjelasan tentang comment dalam program yang jelas dan baik. dan karena beberapa anggota tidak dapat masuk github classroom dibuat repository sementara</td></tr></table> | | Rangkuman Diskusi | Penjelasan spesifikasi tubes dan penjelasan tentang data csv yang digunakan, serta penjelasan tentang comment dalam program yang jelas dan baik. dan karena beberapa anggota tidak dapat masuk github classroom dibuat repository sementara | | | | | | | | | | |
| Rangkuman Diskusi | | | | | | | | | | | | | |
| Penjelasan spesifikasi tubes dan penjelasan tentang data csv yang digunakan, serta penjelasan tentang comment dalam program yang jelas dan baik. dan karena beberapa anggota tidak dapat masuk github classroom dibuat repository sementara | | | | | | | | | | | | | |

Asistensi 2

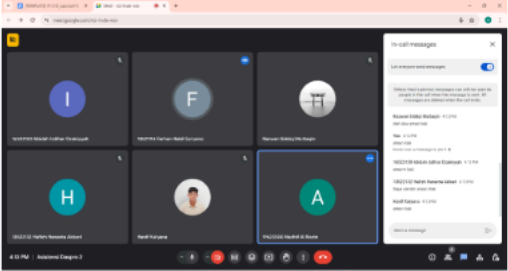
Gambar 2. Hasil Pindaian Asistensi 2

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

Nomor Asistensi : 2
No. Kelompok/Kelas : Kelompok D/K-11
Tanggal asistensi : 12 Mei 2024

| Anggota kelompok | NIM / Nama (Hanya yang Hadir) |
|--------------------|-------------------------------------|
| 1 | 19623300 / Nadhif Al Rozin |
| 2 | 19623200 / Nazwan Siddqi Muttaqin |
| 3 | 19623120 / Hanif Kalyana Aditya |
| 4 | 16523130 / Abidah Izdiyar Dzakiyyah |
| 5 | 13522132 / Hafizh Hananta Akbari |
| 6 | |
| Asisten pembimbing | NIM / Nama |
| | 13521114 / Farhan Nabil Suryono |

Catatan Asistensi:

| |
|---|
| <p>Rangkuman Diskusi</p> <p>Mengecheck progress dan hasil pengerjaan masing masing, persiapan melakukan integrasi ke fungsi main, penentuan waktu diskusi kelompok selanjutnya dan pembagian tugas pengerjaan laporan</p> |
| <p>Tindak Lanjut</p> <p>Senin, 13 Mei 2024 jam 7 malam, dijadwalkan melakukan zoom untuk integrasi awal, dialnutkan dengan pembagian tugas yang dikerjakan mandiri. Melanjutkan pengerjaan dan mencil apa yang bisa digabung terlebih dahulu</p> |
| <p>Dokumentasi</p>  |