LAPORAN TUGAS BESAR

**A picture containing vector graphics

Description automatically generated**

IF1210 Dasar Pemrograman

Program Fungsional Game O.W.C.A

K08-C

Disusun oleh:

Muhammad Falih Rosyidi/16523108

Mohammad Najmutstsaqib/16523228

Shanice Feodora Tjahjono/19623008

William Andrian Dharma T/19623048

Wisa Ahmaduta Dinutama/19623058

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024

# Pernyataan Kelompok

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

Yang mengeluarkan pernyataan,

Muhammad Falih Rosyidi/16523108

Mohammad Najmutstsaqib/16523228

Shanice Feodora Tjahjono/19623008

William Andrian Dharma T/19623048

Wisa Ahmaduta Dinutama/19623058

# Daftar Isi

[**Pernyataan Kelompok 2**](#_heading=h.vmp1md3hfxjj)

[**Daftar Isi 3**](#_heading=h.iupttzegepp1)

[**Daftar Tabel 4**](#_heading=h.cbr7f04v4spo)

[**Daftar Gambar 5**](#_heading=h.8yzyjnjoctr2)

[**1. Deskripsi persoalan 5**](#_heading=h.cg4wbtcqsjng)

[**2. Daftar pembagian kerja anggota kelompok 7**](#_heading=h.sp24qjsoe2e)

[**3. Checklist hasil rancangan, implementasi dan testing setiap primitif 9**](#_heading=h.xgbc88lu61vn)

[**4. Desain command 10**](#_heading=h.7m3hz55jdcy9)

[**5. Desain kamus data 10**](#_heading=h.qm2xoou9eft0)

[**6. Desain dekomposisi algoritmik dan fungsional program 10**](#_heading=h.j466h3mf74jv)

[**7. Spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat 10**](#_heading=h.bijdog8zzqo3)

[**8. Tangkapan layar hasil pengujian program berdasarkan fitur-fitur pada spesifikasi 11**](#_heading=h.y43g6tv12m3z)

[**9. Lampiran 11**](#_heading=h.iwp04y3serof)

# 

# Daftar Tabel

[**Tabel 2.1: Daftar Pembagian Pembuatan Fitur 7**](#_heading=h.2njpvgan44ik)

[**Tabel 2.2: Daftar Pembagian Pembuatan Laporan 8**](#_heading=h.ld7rov6ehpgq)

[**Tabel 3.1: Checklist hasil rancangan, implementasi dan testing 9**](#_heading=h.86vun23w4pb2)

# 

# 

# Daftar Gambar

# Deskripsi persoalan

Tugas besar ini meminta untuk membuatkan program berupa game di mana user berperan sebagai Agent yang turut membantu mengalahkan Dr. Asep Spakbor dengan mengalahkan monster-monster yang kuat. Perjalanan Agent dimulai dari mencari monster di hutan terpencil tempat tinggal para monster dan kemudian melatih monster-monster tersebut untuk dapat melawan Dr. Asep Spakbor.

Ada berbagai fungsional yang diperlukan dalam program ini sebagai berikut:

1. F00 - Random Number Generator

Fungsional ini digunakan untuk menghasilkan angka secara acak dari sebuah range

angka.

1. F01 - Register  
   Fungsional ini digunakan untuk mendaftarkan akun user terlebih dahulu, dimana user baru memasukkan username dan password akun yang hendak dibuat. User akan berperan sebagai Agent secara default dan memiliki 0 O.W.C.A Coin. Jika belum username yang dimasukkan tidak ada pada database, maka user diminta memilih salah satu monster sebagai monster pertamanya. Fungsional ini juga memastikan bahwa username yang dimasukkan hanya berisikan karakter alfabet, angka, dan simbol underscore (\_) dan strip (-).
2. F02 - Login  
   Fungsional ini digunakan untuk login pada sistem dan dapat dilakukan Agent maupun Admin. Login hanya berhasil apabila username sudah pernah terdaftar, password benar, dan tidak sedang logged in sebelumnya.
3. F03 - Logout  
   Fungsional ini hanya berlaku apabila ada akun yang sedang logged in, dan digunakan untuk logout atau keluar dari akun. Setelah logout, user tidak dapat menggunakan akunnya kembali dan harus login menggunakan akun yang berbeda.
4. F04 - Menu & Help

Fungsional Help merupakan command yang dapat membantu user selama keberjalanan game/program dengan menyediakan footnote untuk memastikan user dapat memasukkan input yang valid.

1. F05 - Monster

Fungsional Monster merupakan algoritma *attack* terhadap musuh ataupun sebaliknya yang berdasarkan pada status *ATK power, DEF power,* dan *HP*. Fungsional ini juga berisi tentang peningkatan status yang diakibatkan dari peningkatan level yang dimiliki.

1. F06 - Potion

Fungsional Potion merupakan fungsi algoritma yang berada saat fungsional battle terjadi atau teraktivasi. Fungsional Potion ini berfungsi untuk menjalankan aksi meminum potion untuk menambah *strength* (kekuatan serangan), *resilience* (kekuatan pertahanan), dan *healing* (pemulihan kesehatan atau *HP*).

1. F07 - Inventory

Fungsional Inventory merupakan fungsional yang berfungsi untuk mengetahui monster, potion, item, dan koin *OC* yang dimiliki pengguna.

1. F08 - Battle

Fungsional Battle berfungsi untuk melakukan aksi battle dengan musuh yang random dan level musuh tersebut juga secara random dari 1-5. Fungsional Battle ini juga terjadi saat fungsional Arena teraktivasi.

1. F09 - Arena  
   Fungsional Arena digunakan untuk meningkatkan kemampuan agent dan monster melalui sesi latihan yang terdiri dari 5 stage, di mana masing-masing stage merepresentasikan level monster yang di lawan. Monster di setiap stage dipilih secara acak dari monster-monster yang ada pada database, dan monster yang sejenis dapat muncul lebih dari sekali pada satu sesi latihan, sehingga jumlah monster pada database dapat kurang dari 5. Health dari monster agent akan dipulihkan di akhir setiap stage. Apabila agent kalah atau keluar dari pertarungan, agent akan mendapatkan status game over. Sebuah sesi dikatakan berakhir apabila kelima stage diselesaikan atau game over. Hadiah akan diberikan di akhir setiap stage dan meningkat sesuai dengan stage.
2. F10 - Shop & Currency  
   Fungsional Shop & Currency memuat OC (O.W.C.A. Coin) yang diperoleh Agent dengan cara memenangkan Battle, Arena, atau minigame B04 - JACKPOT! Akun Agent akan dimulai dengan 0 OC saat pertama kali mendaftarkan akun. OC juga digunakan untuk membeli monster atau potion di Shop. Shop dapat diakses Agent dengan menggunakan command ‘SHOP’. Di dalam Shop, agent dapat memilih untuk melihat, membeli, atau keluar. Adapun validasi yang perlu dilakukan dalam pembelian monster atau potion di Shop.
3. F11 - Laboratory  
   Fungsional Laboratory digunakan untuk meng-upgrade monster yang dimiliki, dan dapat diakses dengan command ‘LABORATORY’. Adapun harga dalam OC yang perlu dibayar untuk meng-upgrade monster, dan bertingkat sesuai dengan level monster yang ingin dicapai.
4. F12 - Shop Management  
   Fungsional Shop Management digunakan untuk mengatur barang-barang yang dijual pada Shop. Adapun opsi yang dapat dipilih untuk mengatur Shop, antara lain lihat, tambah, ubah, hapus, dan keluar.
5. F13 - Monster Management  
   Fungsional Monster Management digunakan untuk mengatur monster dalam database. Adapun validasi input yang perlu dilakukan sesuai database monster yang ada.
6. F14 - Load  
   Prosedur Load memuat data dari file penyimpanan saat pengguna memulai program. Prosedur ini dapat dijalankan dengan memasukkan nama folder yang berisi file penyimpanan. Dalam prosedur ini, perlu dilakukan validasi ada atau tidaknya folder.
7. F15 - Save  
   Prosedur Save bertujuan untuk menyimpan data ke file sesuai dengan struktur data eksternal, di mana user diminta memasukkan nama folder yang akan digunakan sebagai tempat penyimpanan file.
8. F16 - Exit  
   Command ini dapat digunakan Agent maupun Admin untuk meninggalkan game. Pengguna diberi pilihan untuk melakukan penyimpanan file (save) atau tidak sebelum meninggalkan game.

Terdapat spesifikasi bonus yang dikerjakan oleh kelompok kami, yaitu B04 - JACKPOT! Fungsional ini merupakan permainan berupa mesin yang memiliki 3 slot yang menunjukkan item secara random, dan dengan value yang berbeda-beda. Mesin pada fungsional ini memiliki minimal 5 item yang berbeda. User perlu membayar sejumlah OC untuk memainkan permainan ini. Mesin kemudian akan mengeluarkan 3 item secara acak. Apabila user mendapatkan Jackpot (3 item yang sama), user mendapatkan monster random. Jika tidak, user mendapatkan jumlah atau sum dari value ketiga item.

# Daftar pembagian kerja anggota kelompok

### Tabel 2.1: Daftar Pembagian Pembuatan Fitur

| **Fitur** | **Implementasi** | **NIM Desainer** | **NIM Coder** | **NIM Tester** |
| --- | --- | --- | --- | --- |
| F00 - Random Number Generator |  | -16523228 | -16523228 | -16523228 |
| F01 - Register |  | -19623008  -19623048 | -19623008  -19623048 | -19623048  -19623008  -19623058 |
| F02 - Login | function  login | -16523108  -19623048 | -16523108  -19623048 | -19623048  -19623008  -19623058 |
| F03 - Logout | procedure  logout | -16523108  -19623048 | -16523108  -19623048 | -19623048  -19623008  -19623058 |
| F04 - Menu & Help |  | -16523228  -19623048 | -16523228  -19623048 | -19623048  -19623008  -19623058 |
| F05 Monster |  | -16523228 | -16523228 | -16523228 |
| F06 Potion |  | -16523228 | -16523228 | -16523228 |
| F07 Inventory | function  inventory | -16523108 | -16523108 | -16523108  -19623008 |
| F08 Battle |  | -16523228 | -16523228 | -16523228  - 19623008 |
| F09 Arena | function  arena | -16523108 | -16523108 | -16523108 |
| F10 Shop & Currency | function shop | -19623058 | -19623058 | -19623008  -19623058 |
| F11 Lab | function lab | -19623058 | -19623058 | -19623058 |
| F12 Shop Manage |  | -19623008  -19623048 | -19623008  -19623048 | -19623048  - 19623008 |
| F13 Monster Manage |  | -16523228 | -16523228 | -16523228  -19623008 |
| F14 Load |  | -19623048 | -19623048 | -19623048  -19623008  -19623058 |
| F15 Save |  | -19623048 | -19623048 | -19623048  -19623008 |
| F16 Exit |  | -19623048 | -19623048 | -19623048  -19623008  -19623058 |
| B04 JACKPOT! |  | -19623048 | -19623048 | -19623048  -19623008 |

### Tabel 2.2: Daftar Pembagian Pembuatan Laporan

| **No** | **Bagian Laporan** | **NIM** |
| --- | --- | --- |
| 1 | Halaman Cover | 19623008 |
| 2 | Daftar Isi | 16523108 |
| 3 | Daftar Tabel | 16523108 |
| 4 | Daftar Gambar |  |
| 5 | Deskripsi Persoalan | 16523228,19623008 |
| 6 | Daftar Pembagian Tugas | Semua |
| 7 | Checklist Hasil Rancangan, Implementasi, dan Testing | 19623008 |
| 8 | Desain Command | 19623008, 16523108 |
| 9 | Desain Kamus Data |  |
| 10 | Desain Dekomposisi Algoritmik dan Fungsional Program | 19623008 |
| 11 | Spesifikasi |  |
| 12 | Hasil Pengujian Program | 19623058 |
| 13 | Lampiran | 19623008 |

# Checklist hasil rancangan, implementasi dan testing setiap primitif

### Tabel 3.1: Checklist hasil rancangan, implementasi dan testing

| **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 & Help | V | V | V |
| F05 Monster | V | V | V |
| F06 Potion | V | V | V |
| F07 Inventory | V | V | V |
| F08 Battle | V | V | V |
| F09 Arena | V | V | V |
| F10 Shop & Currency | V | V | V |
| F11 Lab | V | V | V |
| F12 Shop Manage | V | V | V |
| F13 Monster Manage | V | V | V |
| F14 Load | V | V | V |
| F15 Save | V | V | V |
| F16 Exit | V | V | V |
| B04 JACKPOT! | V | V | V |

# Desain *Command*

4.1 F01 - Register

| Nama Command: register  # dipilih dari instruksi awal saat memulai game yang terdiri dari tiga pilihan, yaitu login/register/exit |
| --- |
| # Ketika user memasukkan username dengan karakter yang diperbolehkan (alfabet huruf besar (A-Z) atau kecil (a-z), angka (0-9), underscore (\_), atau tanda strip (-)) dan tidak ada username yang sama dengan username yang sudah terdaftar sebelumnya.  >>> register  Register as a new user !!!  Enter username: phineasegitiga  Enter password: saenihp123  # User diharuskan memilih salah satu monster dari monster database sebagai monster awal.  Choose your first monster!  1. Pikachow  2. Bulbu  3. Zeze  4. Zuko  5. Chacha  # Pesan selamat kepada user menandakan pendaftaran yang berhasil.  Welcome Agent phineasegitiga. Let's beat Dr. Asep Spakbor with Pikachow!  # Mengarahkan user untuk menggunakan command help.  Use the command 'HELP' to see available actions  >>> |
| # Ketika user memasukkan username yang sama dengan salah satu username yang telah terdaftar.  >>> register  Register as a new user !!!  Enter username: Asep\_Spakbor  # Memberi pesan pada user bahwa username sudah terdaftar dan menawarkan user untuk login.  Username Asep\_Spakbor has already been taken.  Do you want to login instead? Y/N :  >>> |
| # Ketika user memasukkan username yang belum terdaftar, tetapi mengandung karakter di luar karakter yang diperbolehkan.  >>> register  Register as a new user !!!  Enter username: purrypurry+  # Memberi pesan pada user bahwa user memasukkan karakter di luar karakter yang diperbolehkan sebagai username.  Username must only contain letters, numbers, underscores, and dashes. |

4.2 F02 - Login

| Nama Command: login  # dipilih dari instruksi awal saat memulai game yang terdiri dari tiga pilihan, yaitu login/register/exit |
| --- |
| # Ketika user sudah memasukkan username yang ada pada database, dengan password yang sesuai.  >>> login  Login into the system !!!  Username: Agen\_P  Password: platypus123  # Memberi pesan selamat datang kembali yang menandakan proses login yang berhasil.  Welcome back, Agent Agen\_P!  # Mengarahkan user untuk menggunakan command help.  Use the command 'HELP' to see available actions |
| # Ketika user memasukkan username yang tidak terdaftar.  >>> login  Login into the system !!!  Username: sukro\_oven  Password: sukr00123  # Memberitahu kepada user bahwa username yang dimasukkan belum terdaftar.  Username is not registered!  # Memberi pesan pada user yang menawarkan untuk register apabila belum memiliki akun.  Do you want to register instead? Y/N :  >>> |
| # Ketika user memasukkan username yang benar, tetapi dengan password yang salah.  >>> login  Login into the system !!!  Username: Asep\_Spakbor  Password: inipasswordbukan  # Memberi pesan pada user bahwa password yang dimasukkan salah atau tidak sesuai.  Wrong password!  # Meminta user untuk memasukkan username (dan password) kembali.  Login into the system !!!  Username:  >>> |

4.3 F03 - Logout

| Nama Command: logout |
| --- |
| # Ketika memanggil command logout, user dinyatakan logged out dari akun  >>> logout  Successfully Logged Out  # User diberi tiga pilihan lanjut, yaitu login/register/exit.  Before entering system, please Login/Register/Exit  LOGIN / REGISTER / EXIT: |

4.4 F04 - Menu & Help

| Nama Command: help |
| --- |
| # Memanggil command help sebagai Agent.  >>> help  # Tampilan Menu & Help untuk user yang berperan sebagai Agent.  =========== HELP ===========  Hello Agent Agen\_P. Since you requested HELP, here are your available actions:  1. LOGOUT: Log out from your account  2. SHOP: Visit the Shop to purchase items or monsters  3. INVENTORY: View items and use your OWCA-Dex to see owned monsters  4. LAB: Upgrade the monsters you own  5. BATTLE: Engage in battle with enemies  6. ARENA: Fight hordes of monsters in the Arena  7. GACHA: Play and win lots of OWCA Coins !!!  8. SAVE: Save current progress and data  9. EXIT: Exit the game  # User dapat memilih salah satu commands/actions yang ada pada menu atau memasukkan command help kembali.  Use the command 'HELP' to see available actions  >>> |
| # Memanggil command help sebagai Admin.  >>> help  # Tampilan Menu & Help untuk user yang berperan sebagai Admin.  =========== HELP ===========  Welcome back, Admin. Since you requested HELP, here are your available actions:  1. LOGOUT: Log out from your account  2. SHOP: Manage the SHOP stock for agents to buy from  3. MONSTER: Add or Remove monsters from the database  4. SAVE: Save current progress and data  5. EXIT: Exit the game  # User dapat memilih salah satu commands/actions yang ada pada menu atau memasukkan command help kembali.  Use the command 'HELP' to see available actions  >>> |

4.5 F07 - Inventory

| # Memanggil command help sebagai Agent.  Nama Command: inventory (untuk Agent) |
| --- |
| # Tampilan ketika user memanggil command inventory memuat Jumlah O.W.C.A. yang dimiliki serta monster dan/atau potion yang dimiliki.  >>> inventory  ============ INVENTORY LIST (User ID: 3) ============  Jumlah O.W.C.A. Coin-mu sekarang 0.  1. Monster (Name: Bulbu, Lvl: 2, HP: 1200)  2. Monster (Name: Zeze, Lvl: 1, HP: 100)  3. Potion (Type: resilience, Qty: 7)  Ketikkan id untuk menampilkan detail item:  (99:KELUAR, 0:INVENTORY)  # Ketika user memasukkan 0, list inventory akan ditampilkan kembali.  >>> 0  ============ INVENTORY LIST (User ID: 3) ============  Jumlah O.W.C.A. Coin-mu sekarang 0.  1. Monster (Name: Bulbu, Lvl: 2, HP: 1200)  2. Monster (Name: Zeze, Lvl: 1, HP: 100)  3. Potion (Type: resilience, Qty: 7)  Ketikkan id untuk menampilkan detail item:  (99:KELUAR, 0:INVENTORY)  # Ketika user memasukkan 1-3 (tergantung pada inventory user masing-masing), informasi lebih lanjut tentang item yang dimiliki (monster/potion) ditampilkan.  >>> 1  Monster  Name : Bulbu  ATK Power : 50  DEF Power : 50  HP : 1200  Level : 2  Ketikkan id untuk menampilkan detail item:  (99:KELUAR, 0:INVENTORY)  >>> 2  Monster  Name : Zeze  ATK Power : 300  DEF Power : 10  HP : 100  Level : 1  Ketikkan id untuk menampilkan detail item:  (99:KELUAR, 0:INVENTORY)  >>> 3  Potion  Type : resilience  Quantity : 7  Ketikkan id untuk menampilkan detail item:  (99:KELUAR, 0:INVENTORY)  >>>  # Apabila user memasukkan 99, user akan keluar dari inventory dan diminta untuk memanggil command lain.  Use the command 'HELP' to see available actions  >>> |

4.6 F08 - Battle

4.7 F09 - Arena

| Nama Command: arena (untuk Agent) |
| --- |
| # Tampilan Awal, serta meminta user untuk memilih monster yang akan bertarung  >> arena  Selamat datang di Arena!!  ============ MONSTER LIST ============  1. Bulbu  2. Zeze  Pilih monster untuk bertarung: 1 #jika memilih 1-2 tergantung jumlah monster user  /\----/\\_  / \ /|  | | O O | / |  | | .vvvvv.|/ /  / | | | /  / | `^^^^^ /  | /| | /  / | \_\_\_ |  \ | | |  | | | |  \.\_\ \.\_\  RAWRRR, Agent Agen\_P mengeluarkan monster Bulbu !!!  Name : Bulbu  ATK Power : 55  DEF Power : 50  HP : 1320  Level : 2  ============= STAGE 1 =============  \_/\----/\  / \ /\  | O O | | |  | .vvvvv. | | |  / | | \ | |  / `^^^^^' \ | |  ./ /| \| |\_  / / | |\\_\_ /  \ / | | |\_\_|  `' | \_ |  \_.-'-' `-'-'.'\_  \_\_.-' '-.\_\_  RAWRRR, Monster Peroro has appeared !!!  Name : Peroro  ATK Power : 50  DEF Power : 50  HP : 1200  Level : 1  ============ TURN 1 Peroro ============  1. Attack  2. Use Potion  3. Quit  Select a command:  # battle seperti biasa  #jika berhasil menang 5 stage  STAGE CLEARED! Anda akan mendapatkan 200 OC pada sesi ini!  Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!  ============== STATS ==============  Total hadiah : 200 OC  Jumlah stage : 5  Damage diberikan : 4900  Damage diterima : 4329  #Jika menang satu stage selain stage terakhir (5)  STAGE CLEARED! Anda akan mendapatkan 20 OC pada sesi ini!  Memulai stage berikutnya...  ============= STAGE 2 =============  #battle kembali di stage berikutnya  #Jika kalah di stage tersebut  GAME OVER! Sesi latihan berakhir pada stage 1!  ============== STATS ==============  Total hadiah : 0 OC  Jumlah stage : 0  Damage diberikan : 653  Damage diterima : 1320  Use the command 'HELP' to see available actions  #Jika kabur  You have successfully escaped the BATTLE!  ============== STATS ==============  Total hadiah : 0 OC  Jumlah stage : 0  Damage diberikan : 138  Damage diterima : 97 |

4.8 F10 - Shop & Currency

| Nama Command: shop (untuk Agent) |
| --- |
| # Tampilan Awal  >> shop  >>> Karakter Ryo the Merchant <<<  Watcha buyin? Are you in the mood for [monster/item]? Or... you can also leave by typing [exit]  >> |
| # Melihat monster  >> monster  Imported directly from ragunan! type the [ID] to choose your desired monster and type [back] to get back to main shop  ID | Type | ATK Power | DEF Power | HP | Stock | Price  1 | Daz | 200 | 20 | 600 | 10 | 500  2 | Peroro | 50 | 50 | 1200 | 4 | 700  3 | Todome | 300 | 20 | 300 | 3 | 1000  4 | Bozaro | 99 | 15 | 800 | 8 | 550  5 | Ordos | 123 | 45 | 678 | 7 | 600  >> ID/BACK: |
| # Membeli monster  >> ID/BACK: 1  you currently have 9999 oc and the price is 500 oc, are you sure you want to buy this? [y/n]  >> y  thank you for your patronage! you currently have 9499 oc left in your wallet  # Menampilkan kembali list monster setelah transaksi  ID | Type | ATK Power | DEF Power | HP | Stock | Price  1 | Daz | 200 | 20 | 600 | 9 | 5002  2 | Peroro | 50 | 50 | 1200 | 4 | 700  3 | Todome | 300 | 20 | 300 | 3 | 1000  4 | Bozaro | 99 | 15 | 800 | 8 | 550  5 | Ordos | 123 | 45 | 678 | 7 | 600  >> |
| # Melihat potion  >> item  Best to supply your journey! take one of these cool and awesome items.  \*type the item\_id to choose your desired potion and type 'back' to get back to main shop  Item Id | Type | Stock | Price  1 | strength | 10 | 50  2 | resilience | 5 | 30  3 | healing | 3 | 20  >> |
| # Membeli potion  >> id / BACK : 1  how many of that item you want dear customer?  >> 2  you currently have 9499. Are you sure you want to buy this for 100 OC? y/n  >> y  # Menampilkan kembali list item setelah transaksi  Item Id | Type | Stock | Price  1 | strength | 8 | 50  2 | resilience | 5 | 30  3 | healing | 3 | 20  >> |
| # Membeli: gagal dalam validasi  >> ID/BACK : 1  you only have 0 oc lmao, come back when you have the money  >> |
| # Keluar dari shop  >> ID/BACK : back  >>> Karakter Ryo the Merchant <<<  Watcha buyin? Are you in the mood for [monster/item]? Or... you can also leave by typing [exit]  >> exit  >> |

4.9 F11 - Laboratory

4.10 F12 - Shop Management

| Nama Command: shop (hanya untuk Admin) |
| --- |
| >> shop  # Tampilan saat masuk ke Shop  >>> SHOP  Ah.. it's good to see you again. Welcome!  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: |
| # Melihat monster yang ada di Shop  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: view  >> MONSTER / ITEM: monster  ID | Type | Atk Power | Def Power | HP | Stock | Price  1 | Daz | 200 | 20 | 600 | 10 | 500  2 | Peroro | 50 | 50 | 1200 | 4 | 700  3 | Todome | 300 | 20 | 300 | 3 | 1000  4 | Bozaro | 99 | 15 | 800 | 8 | 550  5 | Ordos | 123 | 45 | 678 | 7 | 600  >> |
| # Melihat potion yang ada di Shop  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: view  >> MONSTER / ITEM: item  Item ID | Type | Stock | Price  1 | strength | 10 | 50  2 | resilience | 5 | 30  3 | healing | 3 | 20  >> |
| # Mengubah stok dan/atau harga dari monster  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: edit  >> MONSTER / ITEM monster  ID | Type | Atk Power | Def Power | HP | Stock | Price  1 | Daz | 200 | 20 | 600 | 10 | 500  2 | Peroro | 50 | 50 | 1200 | 4 | 700  3 | Todome | 300 | 20 | 300 | 3 | 1000  4 | Bozaro | 99 | 15 | 800 | 8 | 550  5 | Ordos | 123 | 45 | 678 | 7 | 600  # Mengubah stok saja  >> Enter Monster ID: 3  >> Enter new stock: 5  >> Enter new price:  Todome has been restocked with a quantity of 5!  # Mengubah harga saja  >> Enter Monster ID: 2  >> Enter new stock:  >> Enter new price: 850  Peroro has been updated with a price of 850!  # Mengubah keduanya  >> Enter Monster ID: 5  >> Enter new stock: 6  >> Enter new price: 700  Ordos has been restocked with a quantity of 6 and has been updated with a price of 700! |
| # Mengubah stok dan/atau harga dari item  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: edit  >> MONSTER / ITEM item  Item ID | Type | Stock | Price  1 | strength | 10 | 50  2 | resilience | 5 | 30  3 | healing | 3 | 20  # Mengubah stok saja  >> Enter Item id: 1  >> Enter new stock: 2  >> Enter new price:  strength has been restocked with a quantity of 2!  # Mengubah harga saja  >> Enter Item id: 3  >> Enter new stock:  >> Enter new price: 40  healing has been updated with a price of 40!  # Mengubah keduanya  >> Enter Item id: 2  >> Enter new stock: 6  >> Enter new price: 15  resilience has been restocked with a quantity of 6 and has been updated with a price of 15! |
| # Menghapus monster dari Shop  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: delete  >> MONSTER / ITEM: monster  ID | Type | ATK Power | Def Power | hp | stock | price  1 | Daz | 200 | 20 | 600 | 10 | 500  2 | Peroro | 50 | 50 | 1200 | 4 | 700  3 | Todome | 300 | 20 | 300 | 3 | 1000  4 | Bozaro | 99 | 15 | 800 | 8 | 550  5 | Ordos | 123 | 45 | 678 | 7 | 600  >> Enter Monster ID: 2  >> Are you sure you want to delete Peroro from the shop? (y/n) y  Peroro has been deleted from the shop!  >> |
| # Menghapus item dari Shop  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: delete  >> MONSTER / ITEM: item  Item ID | Type | Stock | Price  1 | strength | 10 | 50  2 | resilience | 5 | 30  3 | healing | 3 | 20  >> Enter Item id: 1  >> Are you sure you want to delete strength from the shop? (y/n) y  strength has been deleted from the shop!  >> |
| # Keluar dari Shop Management  >> OPTIONS: VIEW / ADD / EDIT / DELETE / EXIT: exit  Alrighty, safe travels! Hope to see you again~  >> |

4.11 F13 - Monster Management

| Nama Command: monster (hanya untuk Admin) |
| --- |
| >> monster  # Tampilan awal  WELCOME TO THE MONSTER DATABASE!!!  1. Display all Monsters in the database  2. Add new Monsters  3. Exit |
| # Tampilkan semua monster  >> Select a command: 1  ID | Type | ATK Power | DEF Power | HP  1 | Daz | 200 | 20 | 600  2 | Peroro | 50 | 50 | 1200  3 | Todome | 300 | 20 | 300  4 | Bozaro | 99 | 15 | 800  5 | Ordos | 123 | 45 | 678  6 | Bruaka | 68 | 32 | 750  >> |
| # Tambahkan monster baru  >> Select a command: 2  >> Enter Type / Name: Apex  >> Enter ATK Power : 250  >> Enter DEF Power: 35  >> Enter HP: 900  New monster successfully created!  Type : Apex  ATK Power : 250  DEF Power : 35  HP : 900  >> Add Monster to Database (Y/N): y  Monster added successfully!  >>  # Tambahkan monster baru: nama sudah terdaftar  >> Enter Type / Name: Ordos  Name has been registered, try again!  >>  # Tambahkan monster baru: masukan tidak valid (salah type)  >> Enter ATK Power : seratus  Enter input in the form of an Integer, try again!  >>  # Tambahkan monster baru: DEF Power tidak sesuai  >> Enter DEF Power: 60  DEF Power must be 0-50, try again!  >>  # Cek Database  Select a command: 1  ID | Type | ATK Power | DEF Power | HP  1 | Daz | 200 | 20 | 600  2 | Peroro | 50 | 50 | 1200  3 | Todome | 300 | 20 | 300  4 | Bozaro | 99 | 15 | 800  5 | Ordos | 123 | 45 | 678  6 | Bruaka | 68 | 32 | 750  7 | Apex | 250 | 35 | 900  8 | Kuromi | 100 | 35 | 1000  # Apex dan Kuromi berhasil ditambahkan  # Opsi N pada konfirmasi penambahan monster  >> Add Monster to Database (Y/N): N  Monster failed to add!  >> Select a command: 1  ID | Type | ATK Power | DEF Power | HP  1 | Daz | 200 | 20 | 600  2 | Peroro | 50 | 50 | 1200  3 | Todome | 300 | 20 | 300  4 | Bozaro | 99 | 15 | 800  5 | Ordos | 123 | 45 | 678  6 | Bruaka | 68 | 32 | 750  7 | Apex | 250 | 35 | 900  8 | Kuromi | 100 | 35 | 1000  # Database tidak berubah |

4.12 F15 - Save

| Nama Command: save |
| --- |
| # Folder yang diinput tidak ditemukan dan perlu dibuat.  >>> save  # Meminta input nama folder yang ingin digunakan untuk melakukan penyimpanan.  Enter save folder name : testingsave  Saving, do not exit program  # Membuat folder.  Creating folder at data/testingsave  # Penyimpanan berhasil.  Save succesful at data/testingsave  See you next time agent !!! |
| # Folder yang diinput sudah ada  >>> save  # Meminta input nama folder yang ingin digunakan untuk melakukan penyimpanan.  Enter save folder name : testingsave  Saving, do not exit program  # Penyimpanan berhasil.  Save succesful at data/testingsave  See you next time agent !!! |

4.13 F16 - Exit

| Nama Command: exit |
| --- |
| # Exit dengan save terlebih dahulu.  >>> exit  Do you want to save before exiting? (Y/N):y  # Menjalankan prosedur save (F15). |
| # Exit tanpa melakukan save.  >>> exit  Do you want to save before exiting? (Y/N):n  See you next time agent !!! |

4.14 B04 - JACKPOT!

| Nama Command: gacha |
| --- |
| >>> gacha  # Tampilan awal  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ Welcome to OWCA Gacha! $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ Pay 250 OC to PLAY!!!! $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  ====ITEM LIST====  1. Fedora : 40 OC  2. Watch : 60 OC  3. Gold : 100 OC  4. Dragon : 200 OC  5. Bomb : -50 OC  Get three of a kind and receive a monster!!!  There may even be a hidden reward >< !!! |
| # Ketiga item berbeda semua, agent mendapatkan sum dari value masing-masing item  >> Would you like to play (Y/N)? : Y  GET READY TO GACHAAAAAAAA  ...............................................................................................................................YOU ROLLED !!!! :  | Fedora | Bomb | Dragon  # Misalnya jumlah OC awal 9999, (9999-250+40-50+200 = 9939)  Oops, you lost 60 OC. Better luck next time  Your current balance is 9939 OC  >> |
| # Ketiga item sama, agent mendapatkan monster random  >> Would you like to play (Y/N)? : Y  GET READY TO GACHAAAAAAAA  ...............................................................................................................................YOU ROLLED !!!! :  | Watch | Watch | Watch  MONSTER JACKPOTTTT YOU'VE WON A LEVEL 1 Ordos  >> |
| # User tidak memiliki OC yang cukup untuk bermain  >> Would you like to play (Y/N)? : Y  Sorry you don't have enough OWCA Coins  Come back next time !!!  >> |

# Desain kamus data

# Desain dekomposisi algoritmik dan fungsional program

## F00 RNG

**function** lcg(modulus,a,c,seed:integer)

{menghasilkan nilai acak menggunakan metode Linear Congruential Generator (LCG)}

I.S.: Variabel modulus, a, c, dan seed terdefinisi.

F.S.: Menghasilkan nilai seed baru berdasarkan rumus LCG.

**function** random\_n(n:integer)

{menghasilkan nilai acak berdasarkan jumlah digit yang diinginkan}

I.S.: Variabel n terdefinisi.

F.S.: Menghasilkan bilangan acak dengan jumlah digit n.

**function** randint(n1:integer, n2:integer)

{menghasilkan bilangan acak di antara n1 dan n2}

I.S.: Variabel n1 dan n2 terdefinisi.

F.S.: Menghasilkan bilangan acak yang berada di antara n1 dan n2.

## F01 Register

**procedure** is\_valid\_username()

{memeriksa validitas karakter dalam username}

I.S. Variabel username terdefinisi.

F.S. Menghasilkan True jika semua karakter dalam username valid, dan False jika ada karakter yang tidak valid.

**procedure** choose\_monster()

{memilih monster pertama untuk user}

I.S. Variabel user\_id, username, monster\_data, dan inv\_monster\_data terdefinisi.

F.S. Menambahkan monster yang dipilih ke inv\_monster\_data dan mengembalikan inv\_monster\_data yang telah diperbarui.

**procedure** register\_ui()

{mendaftarkan user baru}

I.S. Variabel user\_data, monster\_data, dan inv\_monster\_data terdefinisi.

F.S. Menambahkan user baru ke user\_data dan mengembalikan informasi username, user\_id, status registrasi, dan inv\_monster\_data yang telah diperbarui.

## F02 Login

**function** is\_find(searched: string, user\_array: list of dictionary, key: string) → tuple of (boolean, integer)

{mencari sebuah string dalam array pengguna berdasarkan kunci yang diberikan dan mengembalikannya indeksnya jika ditemukan, atau -9999 jika tidak ditemukan.}

{ I.S.: searched, user\_array, dan key terdefinisi }

{ F.S.: Mengembalikan tuple (cond, index) }

**function** login(list\_user: list of dictionary) → tuple of (string, integer, boolean)

{meminta input username dan password dari pengguna dan memeriksa apakah kombinasi tersebut sesuai dengan data pengguna yang ada. Jika sesuai, mengembalikan informasi pengguna. Jika tidak, menawarkan untuk mendaftar.}

{ I.S.: list\_user terdefinisi }

{ F.S.: Mengembalikan tuple (user, user\_id, status) }

## F03 Logout

**procedure** logout()

{ I.S.: Tidak ada }

{ F.S.: Menampilkan pesan "Succesfully Logged Out" }

## F04 Menu & Help

**procedure** help\_ui()

{menampilkan bantuan (help) berdasarkan status pengguna}

I.S.: Variabel status dan username terdefinisi.

F.S.: Menampilkan daftar aksi yang tersedia sesuai dengan status pengguna.

## F05 Monster

**function** atribut(monster:dictionary, level:integer)

{menghitung atribut monster berdasarkan levelnya}

I.S.: Variabel monster dan level terdefinisi.

F.S.: Menghasilkan daftar atribut [atk\_power, def\_power, hp] yang telah ditingkatkan sesuai level.

**function** level\_hp(monster:dictionary, level:integer)

{menghitung HP monster berdasarkan levelnya}

I.S.: Variabel monster dan level terdefinisi.

F.S.: Menghasilkan nilai HP yang telah ditingkatkan sesuai level.

**procedure** attack(monster\_enemy: list of integer, atk\_power: integer)

{melakukan serangan ke monster musuh dan mengurangi HP monster musuh berdasarkan damage yang dihasilkan}

I.S.: Variabel monster\_enemy dan atk\_power terdefinisi.

F.S.: HP monster musuh berkurang berdasarkan damage yang dihasilkan.

## F06 Potion

**function** user\_id\_potion(potion\_inventory: list of dictionary, id: integer) -> list

{Mengambil potion bergantung pada user\_id}

I.S.: potion\_inventory dan id terdefinisi.

F.S.: Menghasilkan daftar potion yang dimiliki user dengan user\_id tertentu.

**procedure** up\_ability(monster\_status: list of integer, base\_hp: integer, max\_hp: integer, up: integer)

{Meningkatkan kemampuan monster berdasarkan tipe potion yang digunakan.

I.S.: monster\_status, base\_hp, max\_hp, dan up terdefinisi.

F.S.: Atribut monster ditingkatkan sesuai dengan tipe potion.

**function** find\_index(potion\_inventory: list of dictionary, user\_id: integer, type\_up: string) -> integer

{Mencari indeks potion berdasarkan user\_id dan tipe potion}

I.S.: potion\_inventory, user\_id, dan type\_up terdefinisi.

F.S.: Menghasilkan indeks potion yang sesuai dengan user\_id dan tipe potion.

**procedure** ui\_potion(potion\_inventory: list of dictionary, monster\_name: string, monster\_status: list of integer, base\_hp: integer, max\_hp: integer, id: integer, num\_potion: list of integer)

{Menampilkan daftar potion dan melakukan aksi berdasarkan pilihan pengguna}

I.S.: potion\_inventory, monster\_name, monster\_status, base\_hp, max\_hp, id, dan num\_potion terdefinisi.

F.S.: Melakukan aksi peningkatan kemampuan monster berdasarkan pilihan pengguna.

## F07 Inventory

**function** isnumeric(kata: string) → boolean

{ memeriksa apakah setiap karakter dalam string kata adalah angka}

{ I.S.: kata terdefinisi }

{ F.S.: Mengembalikan True jika semua karakter dalam kata adalah angka, sebaliknya False }

**function** ext\_dict(data: dictionary, keys\_other: list of string) → dictionary

{membuat dictionary baru berdasarkan kunci yang diberikan dari kamus input data}

{ I.S.: data dan keys\_other terdefinisi }

{ F.S.: Mengembalikan kamus baru yang berisi pasangan kunci-nilai yang disaring dari data }

**function** oc(user\_id: integer, user: list of dictionary) → integer

{mencari jumlah OC (koin) dari pengguna berdasarkan user\_id}

{ I.S.: user\_id dan user terdefinisi }

{ F.S.: Mengembalikan jumlah OC dari pengguna dengan user\_id }

**function** item\_inv(user\_id: integer, item\_inventory: list of dictionary) → list of dictionary

{mendapatkan inventaris item untuk pengguna berdasarkan user\_id}

{ I.S.: user\_id dan item\_inventory terdefinisi }

{ F.S.: Mengembalikan daftar item yang dimiliki pengguna }

**function** monster\_inv(user\_id: integer, monster\_inventory: list of dictionary, monster: list of dictionary) → list of dictionary

{mendapatkan inventaris monster untuk pengguna berdasarkan user\_id}

{ I.S.: user\_id, monster\_inventory, dan monster terdefinisi }

{ F.S.: Mengembalikan daftar monster yang dimiliki pengguna }

**procedure** ui\_main\_inventory(user\_id: integer, oc: integer, item: list of dictionary, monster: list of dictionary)

{tampilan utama inventory}

{ I.S.: user\_id, oc, item, dan monster terdefinisi }

{ F.S.: Menampilkan inventaris utama pengguna }

**procedure** ui\_monster\_inventory(monster: dictionary)

{tampilan detail monster}

{ I.S.: monster terdefinisi }

{ F.S.: Menampilkan detail inventaris monster }

**procedure** ui\_item\_inventory(item: dictionary)

{tampilan detail potion}

{ I.S.: item terdefinisi }

{ F.S.: Menampilkan detail inventaris item }

**function** inventory(user\_id: integer, user: list of dictionary, item\_inventory: list of dictionary, monster\_inventory: list of dictionary, monster: list of dictionary) → dictionary

{menampilkan dan mengelola inventaris pengguna, termasuk interaksi pengguna untuk melihat detail item atau monster.}

{ I.S.: user\_id, user, item\_inventory, monster\_inventory, dan monster terdefinisi }

{ F.S.: Menampilkan dan mengelola inventaris pengguna, mengembalikan dictionary berisi OC, potion, dan monster }

## F08 Battle

**procedure** user\_id\_monster()

{Mengambil monster bergantung pada user\_id}

I.S.: monster\_inventory dan id terdefinisi.

F.S.: Menghasilkan daftar monster yang dimiliki user dengan user\_id tertentu.

**procedure** name\_user()

{Menentukan nama dari id yang diberikan}

I.S.: id dan user terdefinisi.

F.S.: Menghasilkan nama user berdasarkan id yang diberikan.

**procedure** select\_monster()

{Memilih monster untuk bertarung}

I.S.: id, user, list\_monster\_user, dan monster\_dict terdefinisi.

F.S.: Menghasilkan pilihan monster dan atribut monster yang dipilih.

## F09 Arena

**procedure** ui\_stats()

{Menampilkan statistik hasil permainan}

I.S.: result, oc\_received, stage, damage\_dealt, dan damage\_received terdefinisi.

F.S.: Menampilkan statistik permainan.

**procedure** ui\_arena()

{Menampilkan UI arena dan memanggil ui\_stats}

I.S.: result, oc\_received, stage, damage\_dealt, dan damage\_received terdefinisi.

F.S.: Menampilkan UI arena dan memanggil ui\_stats.

**function** arena(user\_id: integer, user\_data: list of dictionary, item\_inventory: list of dictionary, monster\_inventory: list of dictionary, monster: list of dictionary) -> integer

{Program utama untuk arena pertempuran}

I.S.: user\_id, user\_data, item\_inventory, monster\_inventory, dan monster terdefinisi.

F.S.: Menghasilkan OC yang diterima setelah pertempuran.

## F10 Shop & Currency

**procedure** exist()

{Memeriksa apakah suatu elemen tertentu ada dalam list of dictionaries berdasarkan kategori yang diberikan.}

I.S.: list\_of\_dict terdefinisi sebagai list of dictionaries. n dan category terdefinisi sebagai nilai dan kategori yang akan dicari.

F.S.: Mengembalikan True jika elemen ditemukan dalam list of dictionaries, False jika tidak ditemukan.

**procedure** index()

{Mengembalikan indeks dari suatu elemen yang dicari dalam list of dictionaries berdasarkan kategori yang diberikan.}

I.S.: list\_of\_dict terdefinisi sebagai list of dictionaries. n dan category terdefinisi sebagai nilai dan kategori yang akan dicari.

F.S.: Mengembalikan indeks elemen dalam list of dictionaries jika ditemukan.

**procedure** textbox()

{Membuat textbox (hiasan) dengan teks yang diberikan}

I.S. text terdefinisi

F.S. Menampilkan teks dalam bentuk textbox di console

**procedure** user\_id\_monster()

{Mengambil monster bergantung pada user\_id}

I.S. monster\_inventory dan user\_id terdefinisi

F.S. Mengembalikan list of dictionaries yang berisi monster dengan user\_id yang diberikan

**procedure** monster\_shop()

{Membuka shop khusus transaksi monster}

I.S. monster\_data, shop\_monster\_data, inv\_monster\_data, users\_data, dan user\_id terdefinisi

F.S. Menampilkan toko monster dan memungkinkan pengguna untuk membeli monster

**procedure** item\_shop()

{Membuka shop khusus transaksi item}

I.S. display\_item\_shop, inv\_item\_data, shop\_item\_data, users\_data, dan user\_id terdefinisi

F.S. Menampilkan toko item dan memungkinkan pengguna untuk membeli item

**procedure** shop()

{Program utama shop yang dipanggil di main}

I.S. shop\_monster\_data, monster\_data, inv\_monster\_data, shop\_item\_data, inv\_item\_data, users\_data, dan user\_id terdefinisi

F.S. Menampilkan toko utama dan memungkinkan pengguna untuk memilih toko monster atau item, atau keluar

## F11 Lab

**procedure** laboratory()

{Menangani peningkatan level monster di laboratorium}

I.S. inv\_monster\_data, monster\_data, users\_data, dan user\_id terdefinisi

F.S. Mengizinkan pengguna untuk meningkatkan level monster mereka jika mereka memiliki cukup uang

## F12 Shop Manage

**procedure** view()

{Menampilkan data monster atau item yang tersedia di shop}

I.S.: monster\_data, shop\_item\_data, shop\_monster\_data, display\_monster\_shop, dan display\_item\_shop terdefinisi

F.S.: Menampilkan tabel monster atau item berdasarkan pilihan user

**procedure** add()

{Menambahkan monster atau item baru ke shop}

I.S.: monster\_data, shop\_item\_data, shop\_monster\_data, display\_monster\_shop, dan display\_item\_shop terdefinisi

F.S.: Monster atau item baru ditambahkan ke shop sesuai input user

**procedure** edit()

{Mengedit stok atau harga monster atau item di shop}

I.S.: monster\_data, shop\_item\_data, shop\_monster\_data, display\_monster\_shop, dan display\_item\_shop terdefinisi

F.S.: Stok atau harga monster atau item di shop diupdate sesuai input user

**procedure** delete()

{Menghapus monster atau item dari shop}

I.S.: monster\_data, shop\_item\_data, shop\_monster\_data, display\_monster\_shop, dan display\_item\_shop terdefinisi

F.S.: Stok monster atau item di shop diupdate menjadi 0 sesuai input user

**procedure** shop\_management(monster\_data, shop\_item\_data, shop\_monster\_data)

{Program Utama Shop Management yang dipanggil di main}

I.S.: monster\_data, shop\_item\_data, dan shop\_monster\_data terdefinisi

F.S.: Menjalankan fungsi manajemen shop berdasarkan input user

## F13 Monster Manage

**procedure** tabel\_monster(monster\_dict)

{Menampilkan tabel monster dengan format yang rapi}

I.S. monster\_dict terdefinisi

F.S. Tabel monster ditampilkan di layar

**procedure** ui\_monster(monster\_dict)

{Menampilkan menu interaktif untuk mengelola database monster}

I.S. monster\_dict terdefinisi

F.S. User dapat melihat semua monster, menambah monster baru, atau keluar dari menu

## F14 Load

**procedure** csv\_to\_dict()

{Mengubah file CSV menjadi list of dictionaries}

I.S. file\_name terdefinisi

F.S. Mengembalikan list of dictionaries yang berisi data dari file CSV

**procedure** convert\_realvalues\_dict()

{Mengubah nilai dari key tertentu dalam dictionary menjadi integer}

I.S. arr\_key\_int dan arr\_dict terdefinisi

F.S. Mengembalikan list of dictionaries dengan nilai yang telah diubah menjadi integer

**procedure** load\_files()

{Memuat file CSV dari folder yang ditentukan dan mengubahnya menjadi list of dictionaries}

I.S.: -

F.S.: Mengembalikan enam list of dictionaries yang berisi data dari file CSV

## F15 Save

**procedure** csv\_saver()

{Menyimpan data ke dalam file CSV

I.S. folder\_path, data\_array, dan file\_name terdefinisi

F.S. Data disimpan ke dalam file CSV dengan nama file\_name di dalam folder\_path

**procedure** save()

{Menyimpan semua data ke dalam file CSV di folder yang ditentukan}

I.S. Semua data terdefinisi

F.S. Data disimpan ke dalam file CSV di dalam folder yang ditentukan oleh pengguna

## F16 Exit

**procedure** exit\_kill()

{Digunakan untuk keluar dari game}

I.S. Semua data terdefinisi

F.S. Program keluar setelah menyimpan data jika dipilih oleh pengguna

## B04 JACKPOT!

**function** gacha()

{Melakukan permainan gacha di mana pengguna dapat memenangkan hadiah berupa OC atau monster}

I.S. user\_data, monster\_data, dan inv\_monster\_data terdefinisi

F.S. user\_data dan inv\_monster\_data diperbarui berdasarkan hasil permainan gacha

# Spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat

## F00 RNG

## F01 Register

**function** is\_valid\_username (username: string) → boolean

{ Memeriksa validitas karakter dalam username }

**KAMUS LOKAL**

validif : string

char : character

**ALGORITMA**

validif ← "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890\_-"

for each char in username do

if char not in validif then

→ False

→ True

**procedure** choose\_monster (input user\_id: integer, input username: string, input monster\_data: list of dictionary, input/output inv\_monster\_data: list of dictionary)

{ Memilih monster pertama }

**KAMUS LOKAL**

monster\_choice : integer

row : dictionary

**ALGORITMA**

output("Choose your first monster!")

for each row in monster\_data do

output(row["id"], ". ", row["type"])

while True do

output("Enter entry number: "), input(monster\_choice)

if monster\_choice is integer then

monster\_choice ← to\_integer(monster\_choice)

if 0 < monster\_choice < length(monster\_data)+1 then

exit

else

output("Enter a valid number")

else

output("Enter a valid number")

append(inv\_monster\_data, {'user\_id': user\_id, 'monster\_id': monster\_choice, 'level': 1})

output("Welcome Agent ", username, ". Let's beat Dr. Asep Spakbor with ", monster\_data[monster\_choice-1]['type'], "!")

**procedure** register\_ui (input/output user\_data: list of dictionary, input monster\_data: list of dictionary, input/output inv\_monster\_data: list of dictionary)

{ Mendaftarkan pengguna baru dan memilih monster }

**KAMUS LOKAL**

username : string

password : string

user\_id : integer

new\_user : dictionary

new\_inv\_monster : list of dictionary

**ALGORITMA**

*output*("Register as a new user !!!")

while True do

output("Enter username: ")

input(username)

if not is\_valid\_username(username) then

output("Username must only contain letters, numbers,

underscores, and dashes.")

else

if not is\_find(username, user\_data, 'username')[0] then

output("Enter password: "), input(password)

user\_id ← length(user\_data) + 1

new\_user ← {'id': user\_id, 'username': username, 'password':

password, 'role': 'agent', 'oc': 0}

append(user\_data, new\_user)

new\_inv\_monster ← choose\_monster(user\_id, username,

monster\_data, inv\_monster\_data)

return (username, user\_id, True, new\_inv\_monster)

else

output("Username ", username, " has already been taken.")

while True do

output("Do you want to login instead? Y/N : ")

input(login)

if login = "Y" then

login

elif login = "N" then

repeat register\_ui

else

output("Enter a valid option")

## F02 Login

**function** is\_find(searched: string, user\_array: list of dictionary, key: string) → tuple of (boolean, integer)

{ I.S.: searched, user\_array, dan key terdefinisi }

{ F.S.: Mengembalikan tuple (cond, index) }

KAMUS LOKAL

cond: boolean

index: integer

i: integer

ALGORITMA

cond ← False

index ← -9999

for i traversal [0..length(user\_array)-1] do

if searched = user\_array[i][key] then

cond ← True

index ← i

→ (cond, index)

**function** login(list\_user: list of dictionary) → tuple of (string, integer, boolean)

{ I.S.: list\_user terdefinisi }

{ F.S.: Mengembalikan tuple (user, user\_id, status) }

KAMUS LOKAL

user: string

pw: string

user\_exists: boolean

index: integer

register\_move: string

user\_id: integer

ALGORITMA

while True do

output("Login into the system !!!")

input(user, "Username: ")

input(pw, "Password: ")

(user\_exists, index) ← is\_find(user, list\_user, 'username')

if user\_exists then

if list\_user[index]['password'] = pw then

output("Welcome back, Agent ", user, "!")

user\_id ← list\_user[index]['id']

→ (user, user\_id, True)

else

output("Wrong password!")

else

output("Username is not registered!")

while True do

input(register\_move, "Do you want to register instead? Y/N :

")

if register\_move = "Y" then

→ ("invalid\_user", -999, False)

else if register\_move = "N" then

break

else

output("Enter a valid option")

## F03 Logout

**procedure** logout()

{ I.S.: Tidak ada }

{ F.S.: Menampilkan pesan "Succesfully Logged Out" }

ALGORITMA

output("Succesfully Logged Out")

## F04 Menu & Help

## F05 Monster

## F06 Potion

## F07 Inventory

**function** isnumeric(kata: string) → boolean

{ I.S.: kata terdefinisi }

{ F.S.: Mengembalikan True jika semua karakter dalam kata adalah angka, sebaliknya False }

KAMUS LOKAL

cond: boolean

i: integer

ALGORITMA

cond ← True

for i traversal [0..length(kata)-1] do

if not (kata[i] = "1" or kata[i] = "2" or kata[i] = "3" or kata[i] = "4" or kata[i] = "5" or kata[i] = "6" or kata[i] = "7" or kata[i] = "8" or kata[i] = "9" or kata[i] = "0") then

cond ← False

break

→ cond

**function** ext\_dict(data: dictionary, keys\_other: list of string) → dictionary

{ I.S.: data dan keys\_other terdefinisi }

{ F.S.: Mengembalikan kamus baru yang berisi pasangan kunci-nilai yang disaring dari data }

KAMUS LOKAL

row\_dict: dictionary

j: string

ALGORITMA

row\_dict ← dictionary kosong

for j traversal keys\_other do

row\_dict[j] ← data[j]

→ row\_dict

**function** oc(user\_id: integer, user: list of dictionary) → integer

{ I.S.: user\_id dan user terdefinisi }

{ F.S.: Mengembalikan jumlah OC dari pengguna dengan user\_id }

KAMUS LOKAL

oc: integer

i: integer

ALGORITMA

oc ← 0

for i traversal [0..length(user)-1] do

if int(user[i]['id']) = user\_id then

oc ← user[i]['oc']

break

→ oc

**function** item\_inv(user\_id: integer, item\_inventory: list of dictionary) → list of dictionary

{ I.S.: user\_id dan item\_inventory terdefinisi }

{ F.S.: Mengembalikan daftar item yang dimiliki pengguna }

KAMUS LOKAL

item: list of dictionary

i: integer

ALGORITMA

item ← list kosong

for i traversal [0..length(item\_inventory)-1] do

if int(item\_inventory[i]['user\_id']) = user\_id then

item.append(ext\_dict(item\_inventory[i], ['type', 'quantity']))

→ item

**function** monster\_inv(user\_id: integer, monster\_inventory: list of dictionary, monster: list of dictionary) → list of dictionary

{ I.S.: user\_id, monster\_inventory, dan monster terdefinisi }

{ F.S.: Mengembalikan daftar monster yang dimiliki pengguna }

KAMUS LOKAL

monster\_id: list of integer

monster\_level: list of integer

inv\_monster: list of dictionary

i, j: integer

ALGORITMA

monster\_id ← list kosong

monster\_level ← list kosong

for i traversal [0..length(monster\_inventory)-1] do

if user\_id = int(monster\_inventory[i]['user\_id']) then

monster\_id.append(int(monster\_inventory[i]['monster\_id']))

monster\_level.append(int(monster\_inventory[i]['level']))

if length(monster\_id) > 0 then

inv\_monster ← list kosong

for i traversal [0..length(monster\_id)-1] do

for j traversal [0..length(monster)-1] do

if monster\_id[i] = int(monster[j]['id']) then

inv\_monster.append(ext\_dict(monster[j], ['type', 'atk\_power', 'def\_power', 'hp']))

for i traversal [0..length(monster\_level)-1] do

inv\_monster[i]['level'] ← monster\_level[i]

else

inv\_monster ← list kosong

→ inv\_monster

**procedure** ui\_main\_inventory(user\_id: integer, oc: integer, item: list of dictionary, monster: list of dictionary)

{ I.S.: user\_id, oc, item, dan monster terdefinisi }

{ F.S.: Menampilkan inventaris utama pengguna }

KAMUS LOKAL

number: integer

i: integer

ALGORITMA

output("============ INVENTORY LIST (User ID: ", user\_id, ") ============")

output("Jumlah O.W.C.A. Coin-mu sekarang ", oc, ".")

if length(item) + length(monster) > 0 then

number ← 0

for i traversal [0..length(monster)-1] do

number ← number + 1

output(number, ". Monster (Name: ", monster[i]['type'], ", Lvl: ", monster[i]['level'], ", HP: ", monster[i]['hp'], ")")

for i traversal [0..length(item)-1] do

number ← number + 1

output(number, ". Potion (Type: ", item[i]['type'], ", Qty: ", item[i]['quantity'], ")")

else

output("!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!")

output("Tidak ada barang di Inventory kamu. (┬┬﹏┬┬)")

output()

**procedure** ui\_monster\_inventory(monster: dictionary)

{ I.S.: monster terdefinisi }

{ F.S.: Menampilkan detail inventaris monster }

ALGORITMA

output("Monster")

output("Name : ", monster['type'])

output("ATK Power : ", monster['atk\_power'])

output("DEF Power : ", monster['def\_power'])

output("HP : ", monster['hp'])

output("Level : ", monster['level'])

**procedure** ui\_item\_inventory(item: dictionary)

{ I.S.: item terdefinisi }

{ F.S.: Menampilkan detail inventaris item }

ALGORITMA

output("Potion")

output("Type : ", item['type'])

output("Quantity : ", item['quantity'])

**function** inventory(user\_id: integer, user: list of dictionary, item\_inventory: list of dictionary, monster\_inventory: list of dictionary, monster: list of dictionary) → dictionary

{ I.S.: user\_id, user, item\_inventory, monster\_inventory, dan monster terdefinisi }

{ F.S.: Menampilkan dan mengelola inventaris pengguna, mengembalikan dictionary berisi OC, potion, dan monster }

KAMUS LOKAL

user\_oc: integer

user\_item: list of dictionary

user\_monster: list of dictionary

id\_input: string

ALGORITMA

user\_oc ← oc(user\_id, user)

user\_item ← item\_inv(user\_id, item\_inventory)

user\_monster ← monster\_inv(user\_id, monster\_inventory, monster)

ui\_main\_inventory(user\_id, user\_oc, user\_item, user\_monster)

if length(user\_item) + length(user\_monster) = 0 then

→ {'oc': user\_oc, 'potion': user\_item, 'monster': user\_monster}

else

while True do

output("Ketikkan id untuk menampilkan detail item:")

output("(99:KELUAR, 0:INVENTORY)")

input(id\_input, ">>> ")

if isnumeric(id\_input) then

id\_input ← int(id\_input)

if id\_input = 99 then

output()

break

else if id\_input = 0 then

ui\_main\_inventory(user\_id, user\_oc, user\_item, user\_monster)

else if id\_input > length(user\_monster) + length(user\_item) or id\_input <= 0 then

output("Maaf id yang dimasukkan tidak ada dipilihan.")

else if id\_input > length(user\_monster) then

ui\_item\_inventory(user\_item[(id\_input) - length(user\_monster) - 1])

else

ui\_monster\_inventory(user\_monster[id\_input - 1])

else

output("""Maaf input yang anda masukkan tidak sesuai !!

Ketikkan id untuk menampilkan detail item:

(99:KELUAR, 0:INVENTORY)""")

id\_input ← (input(">>> "))

→ {'oc': user\_oc, 'potion': user\_item, 'monster': user\_monster}

## F08 Battle

## F09 Arena

## F10 Shop & Currency

## F11 Lab

## F12 Shop Manage

## F13 Monster Manage

## F14 Load

## F15 Save

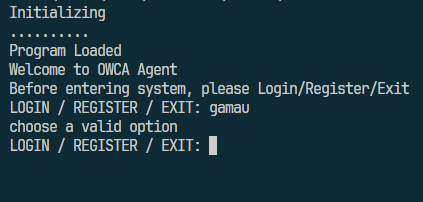
## F16 Exit

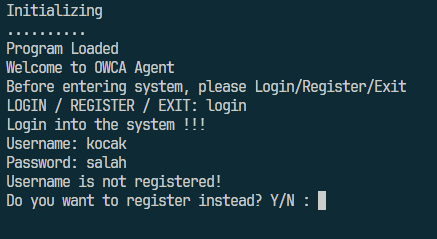
## B04 JACKPOT!

# Tangkapan layar hasil pengujian program berdasarkan fitur-fitur pada spesifikasi

Pada setiap fitur, tangkapan layar minimal berisi: (1) data masukan, (2) data keluaran untuk *input* yang valid dan tidak valid jika terdapat validasi.

1. **F01 - Register**
2. Input invalid



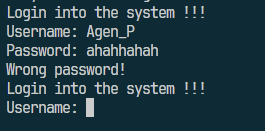
1. Input: login > username salah  
   

\* Y: memanggil fungsi register

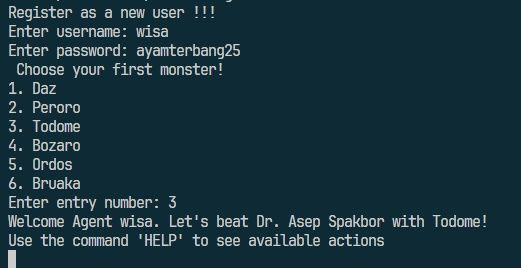
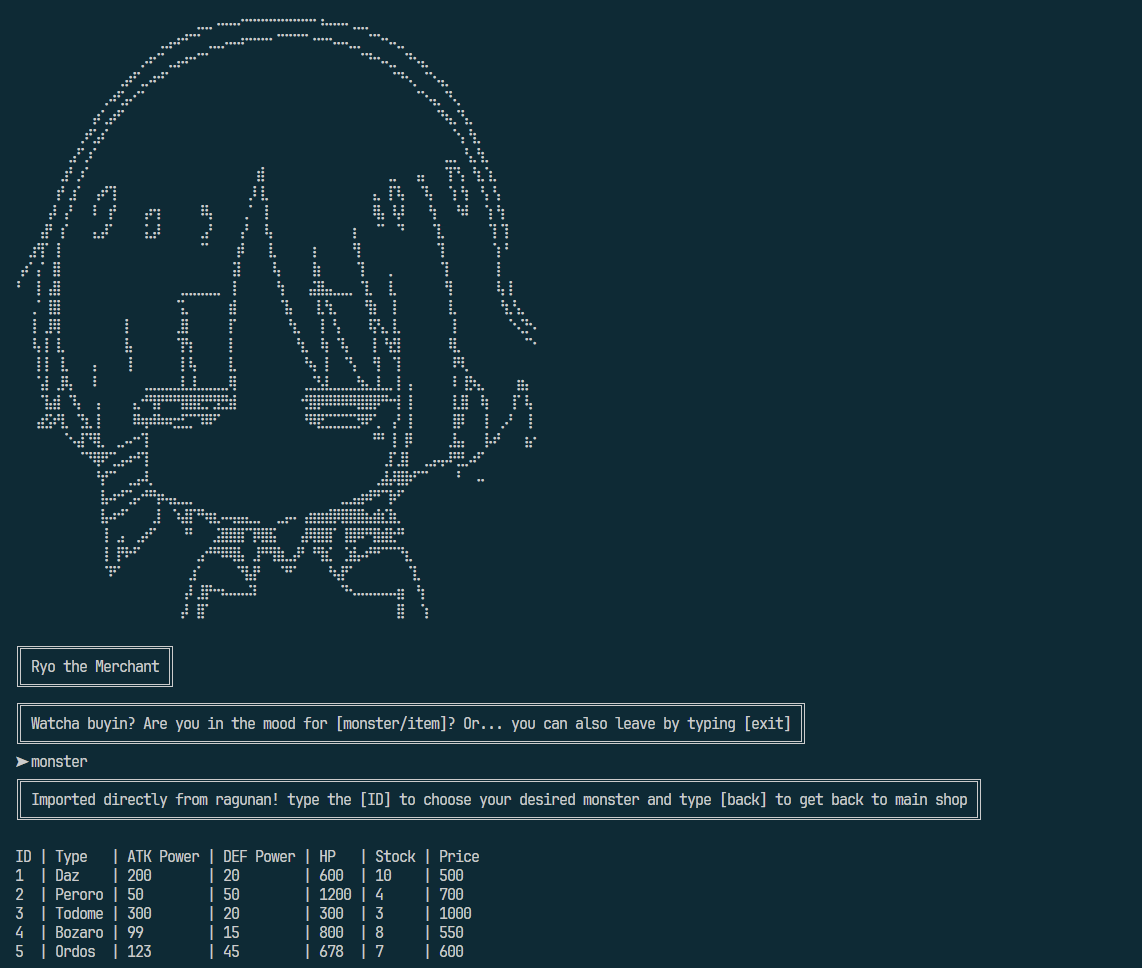
N: kembali ke loop login

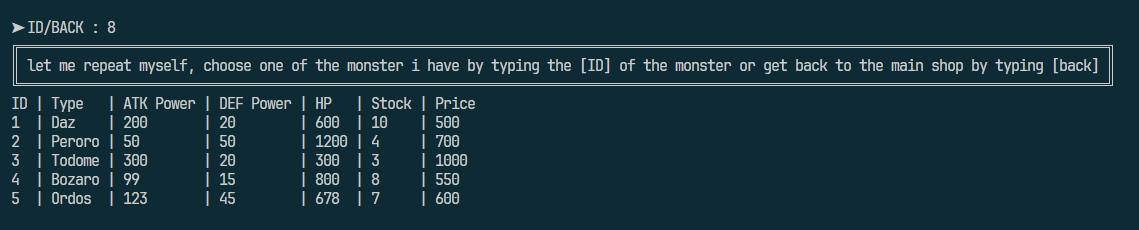
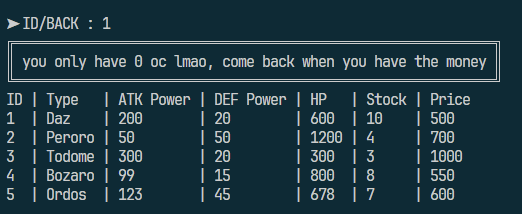
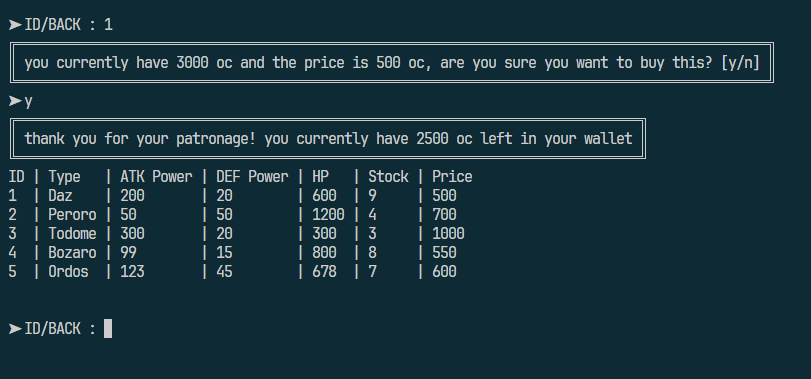
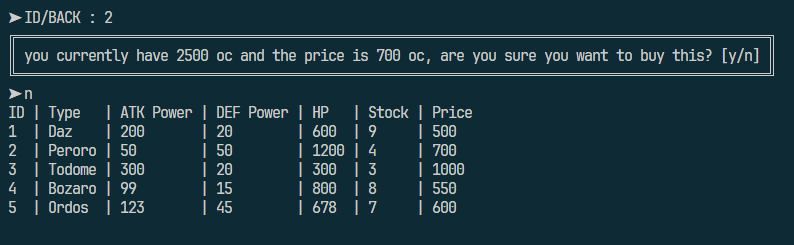
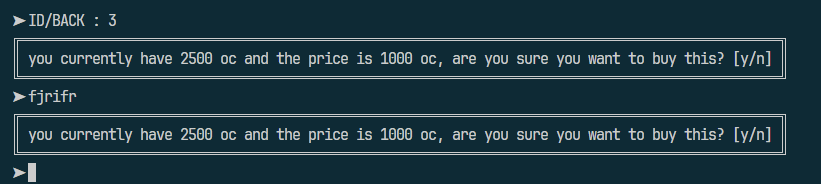
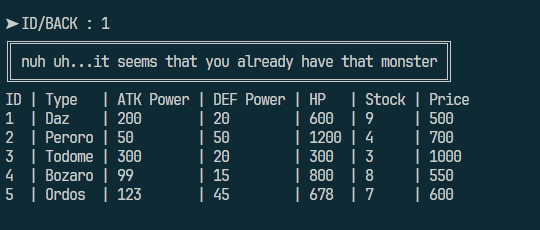
(input invalid): konfirmasi ulang

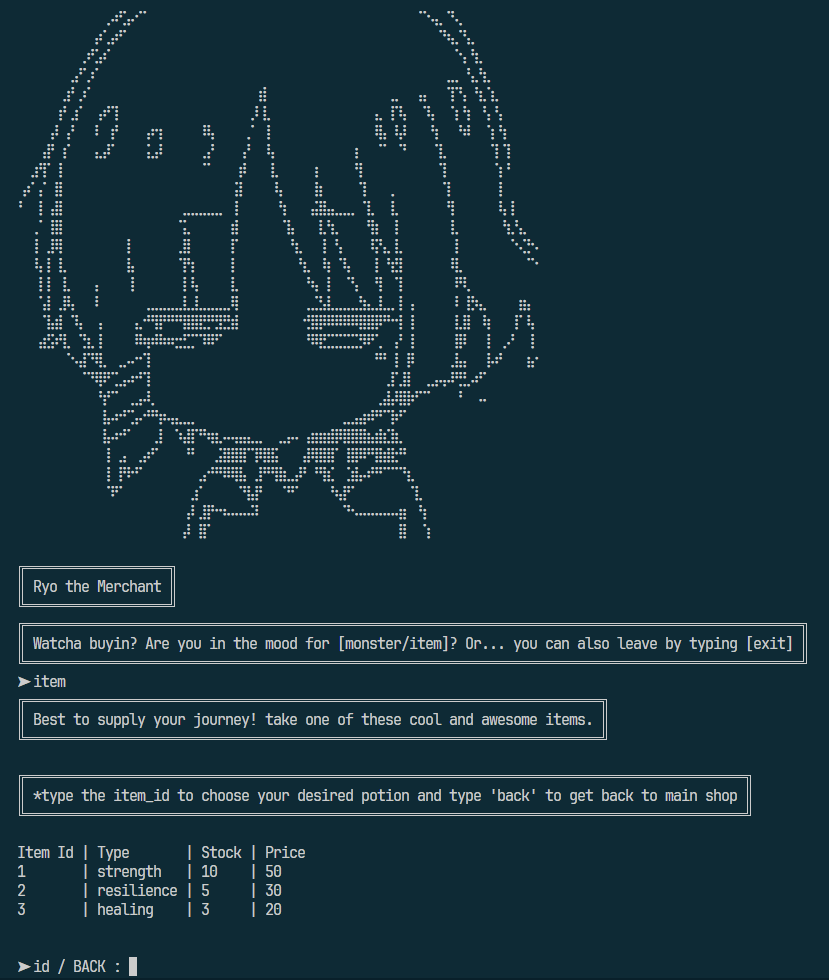
1. Input: login > password salah

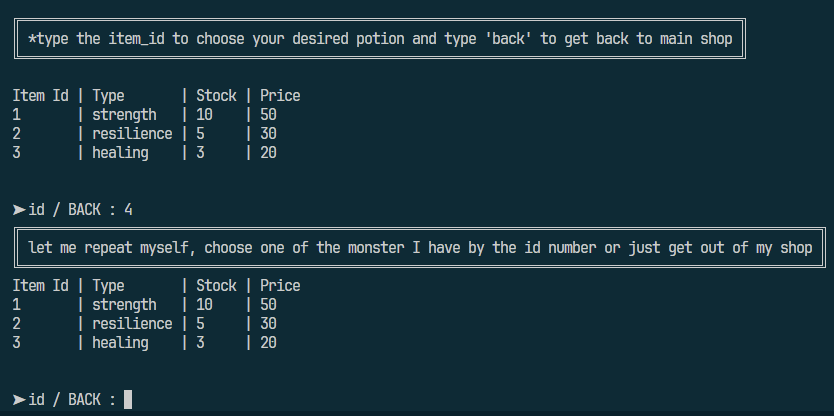
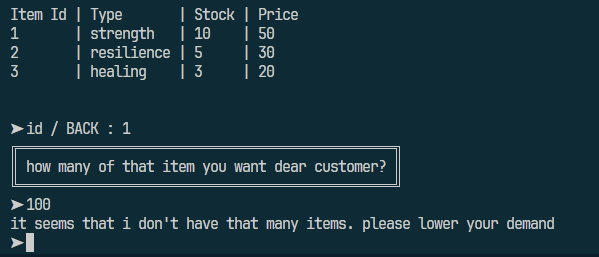
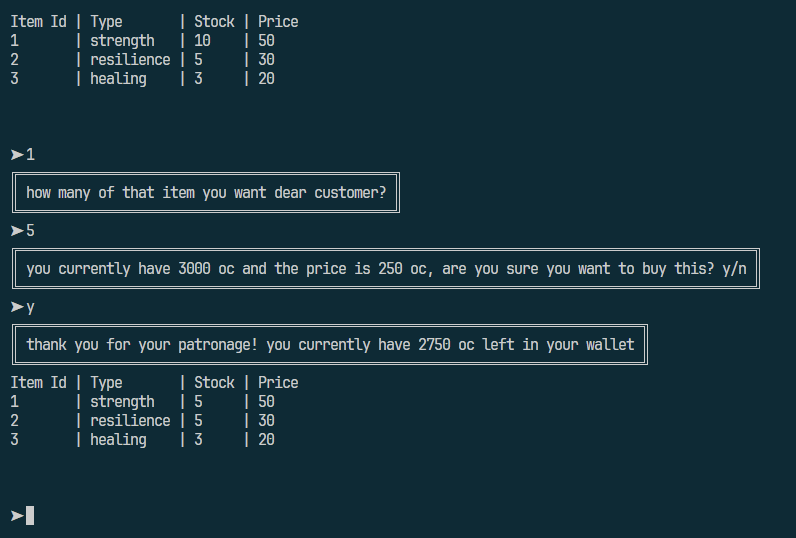
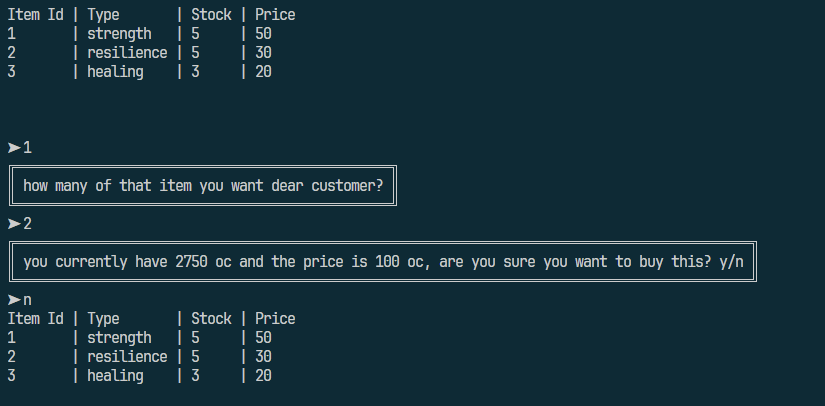
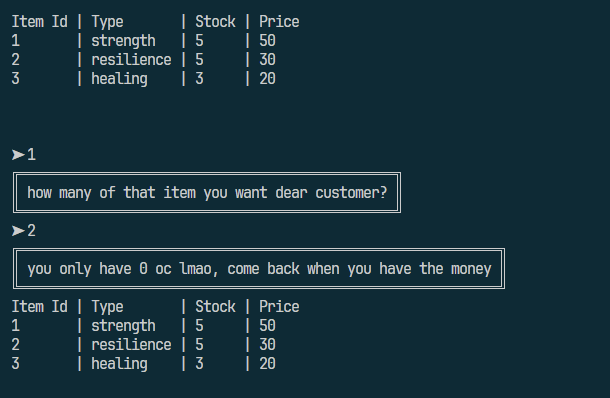
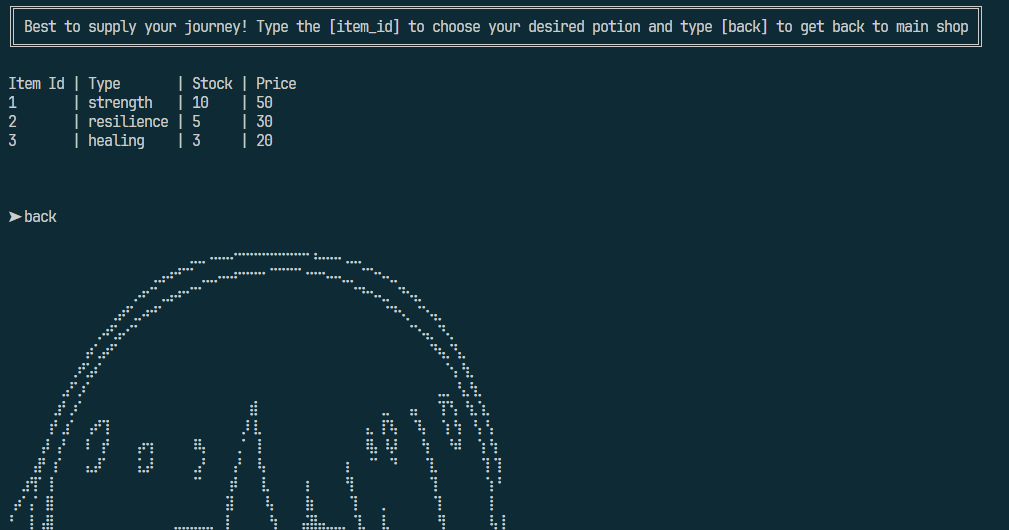


input

1. Input: register > valid username > valid monster  
   
2. F10 - Shop & Currency
3. Input: Monster  
   

* Input: Invalid ID  
  
* Input: Valid ID (Insufficient Money)  
  
* Input: Valid ID (Sufficient Money) > Y  
  
* Input: Valid ID (Sufficient money) > N  
  
* Input: Valid ID (Sufficient money) > Invalid Input  
  
* Input: Valid ID (Monster already in inventory)  
  
* Input: Monster > Back  
  

1. Input: Item  
   

* Input: Invalid ID  
  
* Input: Valid ID > Invalid amount  
  
* Input: Valid ID > Valid amount (Sufficient Money) > Y  
  
* Input: Valid ID > Valid amount (Sufficient Money) > N  
  
* Input: Valid ID > Valid Amount (Insufficient Money)  
  
* Input: Back  
  

1. Input: Exit  
   

# Lampiran

<https://docs.google.com/document/d/1vdMYdmMiIA78J1QkOKQSOcjIMJu7DzGM/edit#heading=h.gjdgxs>