

Laporan Tugas Besar
IF1210 Dasar Pemrograman
SEMESTER II 2022/2023

Program Fungsional Monster

Kelompok K02-H:

Jovandra Otniel P. S. 19623032

Muhammad Iqbal Haidar 19623242

Teresa Dillen Tshia 16523142

Muthia Nabilla Azzahra 16523012

Syauqi Muhammad Fatih 16523272



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,

Jovandra Otniel P. S. 19623032

Muhammad Iqbal Haidar 19623242

Teresa Dillen Tshia 16523142

Muthia Nabilla Azzahra 16523012

Syauqi Muhammad Fatih 16523272

DAFTAR ISI

PERNYATAAN KELOMPOK.....	i
DAFTAR ISI.....	ii
DAFTAR TABEL.....	iii
DAFTAR GAMBAR.....	iv
BAB 1.....	1
BAB 2.....	2
BAB 3.....	3
BAB 4.....	4
BAB 5.....	7
BAB 6.....	40
LAMPIRAN ASISTENSI.....	50

DAFTAR TABEL

2.1. Tabel Pembagian Kerja Kelompok.....	2
2.2. Tabel Pembagian Pembuatan Laporan.....	3
3.1. Tabel Hasil Rancangan.....	5

DAFTAR GAMBAR

6.0. Gambar Pengetesan Random Number Generator.....	51
6.1. Gambar Pengetesan Register.....	51
6.2. Gambar Pengetesan Login.....	51
6.3. Gambar Pengetesan Logout.....	51
6.4. Gambar Pengetesan Help.....	52
6.7. Gambar Pengetesan Inventory.....	53
6.8.1. Gambar Pengetesan Battle Round 1.....	54
6.8.2. Gambar Pengetesan Battle Round 2.....	55
6.9.1. Gambar Pengetesan Arena.....	56
6.9.2. Gambar Pengetesan Arena.....	57
6.10. Gambar Pengetesan Shop.....	58
6.11. Gambar Pengetesan Laboratory.....	59
6.12. Gambar Pengetesan Shop Management.....	60
6.13. Gambar Pengetesan Monster Management.....	61
6.14. Gambar Pengetesan Load.....	62
6.15. Gambar Pengetesan Penyimpanan.....	62
6.16.1. Gambar Pengetesan Exit.....	62
6.16.2. Gambar Pengetesan Exit.....	63

BAB 1

DESKRIPSI PERSOALAN

Purry si Platypus atau yang dikenal juga sebagai Agent P sedang menghadapi tantangan besar, yaitu Dr. Asep Spakbor yang merupakan seorang ilmuwan jahat yang telah menciptakan banyak monster. Monster-monster ini mengancam keamanan kota Danville. Tugas Purry adalah untuk menghentikan ancaman-ancaman ini, tetapi Purry tidak bisa melawannya sendirian sehingga dia membutuhkan bantuan dari agen-agen lain yang ia temui di markas rahasia O.W.C.A. (Organisasi Warga Cool Abiez) untuk mengalahkan monster-monster tersebut.

Untuk mengalahkan Dr. Asep Spakbor, Purry meminta bantuan untuk saling bekerja sama dan mengalahkan monster-monster kuat. Diperlukan strategi yang matang untuk menyelesaikan misi ini dengan baik. Purry juga meminta bantuan untuk mencari dan melatih monster-monster yang digunakan dalam pertempuran melawan Dr. Asep Spakbor.

Misi pencarian monster diputuskan untuk dimulai di hutan terpencil yang diyakini menjadi tempat tinggal bagi banyak jenis monster. Petualangan akan dimulai dengan segala tantangan yang muncul di perjalanan ini demi keselamatan kota Danville.

Dalam misi ini, terdapat 16 fungsi wajib dan beberapa spesifikasi bonus yang perlu dikerjakan untuk mengalahkan monster-monster jahat, tetapi untuk spesifikasi bonus tidak wajib. Spesifikasi program yang wajib dikerjakan meliputi register, login, dan logout, sampai kepada menu & help yang dilanjutkan dengan spesifikasi program yang mencakup database tentang monster, potion, dan juga inventory, lalu terdapat fungsi-fungsi untuk battle dan arena yang digunakan untuk melawan monster, lalu terdapat juga fungsi Shop Management dan Monster Management yang hanya bisa diakses admin, dan terakhir terdapat fungsi load, save, dan juga exit. Semua fungsi akan digunakan untuk menyelesaikan misi agen-agen O.W.C.A

BAB 2

DAFTAR PEMBAGIAN KERJA ANGGOTA KELOMPOK

2.1. Tabel Pembagian Kerja Kelompok

Fitur	Implementasi	NIM Desainer	NIM Coder	NIM Tester
F00-Random Number Generator (RNG)	random_num()	16523142	16523142	16523142
F01-Register	register()	16523142	16523142	19623032
F02-Login	login()	16523272	16523272	19623032
F03-Logout	logout()	16523012	16523012	19623032
F04-Menu & Help	help()	19623032	19623032	19623032
F05-Monster	adjust(); attack(); attack_arena()	19623032	19623032	19623032
F06-Potion	potion()	19623032	19623032	19623032
F07-Inventory	inventory()	19623032	19623032	19623032
F08-Battle	battle()	19623032	19623032	19623032
F09-Arena	arena()	19623032	19623032	19623032
F10-Shop & Currency	shop()	19623032	19623032	19623032
F11-Laboratory	laboratory()	16523012	16523012	19623032
F12-Shop Management	shop_management() ()	19623242	19623242	19623242
F13-Monster Management	monster_management() ment()	19623032	19623032	19623032
F14-Load	load()	19623242	19623242	19623242
F15-Save	save()	19623242	19623242	19623242
F16-Exit	exit_game()	19623032	19623032	19623032

2.2. Tabel Pembagian Pembuatan Laporan

No	Bagian Laporan	NIM
1	Halaman Cover	16523142
2	Halaman pernyataan kelompok	16523142
3	Daftar Isi	16523272
4	Daftar Tabel	16523272
5	Daftar Gambar	16523012
6	Deskripsi persoalan	16523142
7	Daftar pembagian kerja anggota kelompok	16523272 16523142 19623032
8	Checklist hasil rancangan, implementasi dan testing setiap primitif.	16523272
9	Desain command	16523012 16523142 16523272 19623242
10	Desain kamus data	19623242 16523142 16523012

11	Desain dekomposisi algoritmik dan fungsional program.	19623242 16523142 16523012
12	Spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat	19623242 16523142 16523012
13	Tangkapan layar hasil pengujian program	19623242
14	Lampiran	19623242

BAB 3

CHECKLIST HASIL RANCANGAN

3.1. Tabel Hasil Rancangan

Fungsionalitas	Design	Implementasi	Testing
F00-Random Number Generator (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-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

BAB 4

DESAIN *COMMAND*

01. F01-Register

<p>>>> REGISTER Masukan username: bangkitganteng Masukan password: inipassword124</p> <p>Silahkan pilih salah satu monster sebagai monster awalmu. 1. Charizard 2. Bulbasaur 3. Aspal</p> <p>Monster pilihanmu: 1</p> <p>Selamat datang Agent bangkitganteng. Mari kita mengalahkan Dr. Asep Spakbor dengan Charizard!</p>
<p>>>> REGISTER Masukan username: Purry Masukan password: inipassword124</p> <p>Username Purry sudah terpakai, silahkan gunakan username lain!</p>
<p>>>> REGISTER Masukan username: b@ngkitganteng Masukan password: inipassword124</p> <p>Username hanya boleh berisi alfabet, angka, underscore, dan strip!</p>
<p>>>> REGISTER Register gagal! Anda telah login dengan username Purry, silahkan lakukan "LOGOUT" sebelum melakukan register.</p>

02. F02-Login

<p>>>> LOGIN Username: Purry Password: pl@tYPu5</p> <p>Selamat datang, Agent Purry! Masukkan command "help" untuk daftar command yang dapat kamu panggil.</p>
<p>>>> LOGIN Username: Purri Password: pl@tYPu5</p> <p>Username tidak terdaftar!</p>
<p>>>> LOGIN Username: Purry</p>

Password: **platypus**

Password salah!

>>> **LOGIN**

Login gagal!

Anda telah login dengan username Purry, silahkan lakukan "LOGOUT" sebelum melakukan login kembali.

03. F03-Logout

>>> **LOGOUT**

Keluar dari akun

>>>

>>> **LOGOUT**

Logout gagal!

Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout

04. F04-Menu & Help

sebelum login

>>> **HELP**

===== HELP =====

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

1. Login: Masuk ke dalam akun yang sudah terdaftar
2. Register: Membuat akun baru

Footnote:

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

setelah login sebagai Agent

>>> **HELP**

===== HELP =====

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

1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Melihat owca-dex yang dimiliki oleh Agent

Footnote:

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

>>>

```
# setelah login sebagai Admin
>>> HELP
```

```
===== HELP =====
```

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

1. Logout: Keluar dari akun yang sedang digunakan
2. Shop: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent

Footnote:

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

05. F05-Monster

```
>>> Monster
```

```
{ I.S. aksi yang ingin dilakukan untuk melakukan manajemen database monster }
{ F.S. penambahan monster baru atau perubahan spesifikasi monster pada database monster
yang ada berhasil dilakukan apabila masukan mengenai spesifikasi monster sudah sesuai
dengan ketentuan }
```

06. F06-Potion

```
>>> MINI
```

```
{Spesifikasi: Mengembalikan nilai minimum dari dua bilangan bulat.}
{I.S.: Dua bilangan bulat, a dan b, diberikan.}
{F.S.: Nilai minimum antara a dan b ditentukan.}
```

```
>>> POTION
```

```
procedure potion(selected_potion_name: string, selected_user_monster: dictionary, login_id:
string, potion_inventory: array of dictionary, monster_data: array of dictionary)
{ I.S.: Nama potion yang dipilih, data monster pengguna, ID login pengguna, inventory
potion, dan data monster diberikan. }
{ F.S.: Atribut monster pengguna diperbarui dan inventory potion diperbarui jika potion
yang digunakan tersedia. }
```

07. F07-Inventory

```
>>> INVENTORY
```

```
Halo, user_id 2, jumlah OWCA coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
>>> potion
```

Potion yang tersedia:

1. Nama: strength, Jumlah: 5
2. Nama: resilience, Jumlah: 3

Pilih nomor urut potion untuk informasi lebih lanjut:

```
>>> 2
```

Nama: resilience, Jumlah: 3

```
Halo, user_id 2, jumlah OWCA Coin Anda 9999
```

```

>>> Lihat Barang Apa? (potion/monster/quit):
>>> monster
Monster yang tersedia:
1. Pikachow (HP: 600)
Pilih nomor urut monster untuk informasi lebih lanjut:
>>> 1
1. Pikachow (ATK: 125, DEF: 10, HP: 600)

Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
>>> potion
Potion yang tersedia:
1. Nama: strength, Jumlah: 5
2. Nama: resilience, Jumlah: 3
Pilih nomor urut potion untuk informasi lebih lanjut:
>>> 3
Pilihan tidak valid, silakan masukkan nomor yang tersedia
>>> 2
Nama: resilience, Jumlah; 3

Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
>>> quit

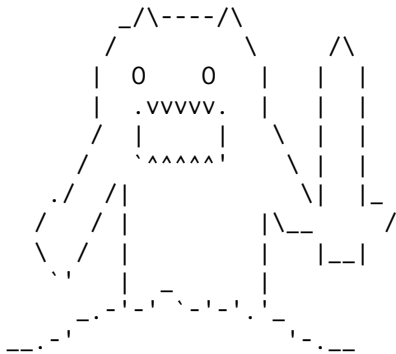
```

08. F08-Battle

```

# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
>>> BATTLE

```



RAWRRR, Monster Zuko telah muncul !!!

```

Name      : Zuko
ATK Power : 20
DEF Power : 20
HP        : 100
Level     : 1

```

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

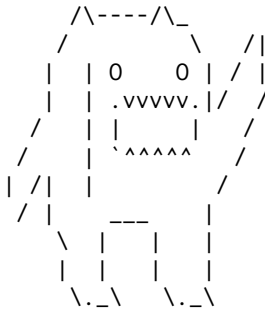
1. Chacha
2. Pikachow
3. Zeze

```
# Apabila monster yang dipilih tidak ada
Pilih monster untuk bertarung: 4
```

Pilihan nomor tidak tersedia!

```
Pilih monster untuk bertarung: _
```

```
# Apabila monster yang dipilih ada
Pilih monster untuk bertarung: 2
```



RAWRRR, Agent X mengeluarkan monster Pikachow !!!

```
Name      : Pikachow
ATK Power : 25
DEF Power : 5
HP        : 120
Level     : 1
```

===== TURN 1 (Pikachow) =====

1. Attack
2. Use Potion
3. Quit

```
# Apabila memilih Quit
Pilih perintah: 3
```

Anda berhasil kabur dari BATTLE!

>>>

```
# Apabila memilih Use Potion, namun tidak memiliki Potion
Pilih perintah: 2
```

Anda tidak memiliki Potion dalam inventory!

```
Pilih perintah: _
```

```
# Apabila memilih Use Potion
Pilih perintah: 2
```

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

Catatan: Pengambilan potion pada inventory dibebaskan boleh ambil potion pertama, terakhir, atau bahkan mengambil berdasarkan item id yang ada di inventory. Potion yang tidak dimiliki di inventory tidak perlu ditampilkan.

Apabila memilih **Strength Potion**

Pilih perintah: 1

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

===== TURN 1 (Zuko) =====

...

Apabila memilih **Resilience Potion**

Pilih perintah: 2

Setelah meminum ramuan ini, muncul sebuah energi pelindung di sekitar Pikachu yang membuatnya terlihat semakin tangguh dan sulit dilukai.

===== TURN 1 (Zuko) =====

...

Apabila memilih **Healing Potion**

Pilih perintah: 3

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

===== TURN 1 (Zuko) =====

...

Apabila memilih **Potion** yang telah diminum sebelumnya

Pilih perintah: 1

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

===== POTION LIST =====

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

Apabila memilih **Potion** yang tidak dimiliki

===== POTION LIST =====

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

Pilih perintah: 1

Wah, kamu sedang tidak memiliki ramuan ini, silahkan pilih ramuan lain!

===== POTION LIST =====

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


```

# Apabila memilih Attack
Pilih perintah: 1

SCHWINKKK, Pikachu menyerang Zuko !!!

Name      : Zuko
ATK Power : 20
DEF Power : 20
HP        : 76
Level     : 1
# Penjelasan: ATT: 30 (+20%), Reduced by: 6 (20%), ATT Results: 24
# Penjelasan battle mohon untuk ditampilkan, format bebas

===== TURN 1 (Zuko) =====

SCHWINKKK, Zuko menyerang Pikachu !!!

Name      : Pikachu
ATK Power : 25
DEF Power : 5
HP        : 102
Level     : 1
# Penjelasan: ATT: 18 (-10%), Reduced by: 0.9 (5%), ATT Results: 17.1 (jika hasil berbentuk
decimal, gunakan FLOOR pada HP terakhir, FLOOR(120 - 17.1) = 102)
# Penjelasan battle mohon untuk ditampilkan, format bebas

# Mekanisme WIN
===== TURN 99 (Pikachu) =====
1. Attack
2. Use Potion
3. Quit

Pilih perintah: 1

SCHWINKKK, Pikachu menyerang Zuko !!!

Name      : Zuko
ATK Power : 20
DEF Power : 20
HP        : 0
Level     : 1
# Misal: ATT Results: 30, HP terakhir 20, jika hasil HP akhir <= 0, hasilnya tetap 0
# Penjelasan battle mohon untuk ditampilkan, format bebas

Selamat, Anda berhasil mengalahkan monster Zuko !!!

Total OC yang diperoleh: 30

>>>

# Mekanisme LOSE
===== TURN 99 (Zuko) =====

SCHWINKKK, Zuko menyerang Pikachu !!!

Name      : Pikachu
ATK Power : 25
DEF Power : 5
HP        : 0
Level     : 1

```

Yahhh, Anda dikalahkan monster Zuko. Jangan menyerah, coba lagi !!!

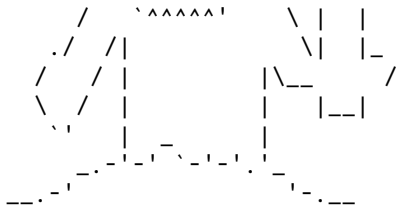
>>>

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
# Masuk Arena
>>> ARENA
```

```
===== MONSTER LIST =====
1. Chacha
2. Pikachow
3. Zeze
```

Pilih monster untuk bertarung: _

```
Name      : Pikachow
ATK Power : 25
DEF Power : 5
HP        : 120
Level     : 1
```



RAWRRR, Monster Zuko telah muncul !!!

Lanjut seperti battle biasa

Apabila stage berhasil dimenangkan dan ada stage berikutnya

Selamat, Anda berhasil mengalahkan monster Zuko !!!

STAGE CLEARED! Anda akan mendapatkan 30 OC pada sesi ini!

Memulai stage berikutnya...

===== STAGE 2 =====

Lanjut stage berikutnya

Apabila agent kalah

Yahhh, Anda dikalahkan monster Zuko. Jangan menyerah, coba lagi !!!

GAME OVER! Sesi latihan berakhir pada stage 2!

===== STATS =====

Total hadiah : 30 OC

Jumlah stage : 1

Damage diberikan : 180

Damage diterima : 250

Apabila stage 5 berhasil diselesaikan

Selamat, Anda berhasil mengalahkan monster Zeze !!!

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 : 780

Damage diterima : 400

Apabila memilih Quit

Pilih perintah: 3

GAME OVER! Anda mengakhiri sesi latihan!

===== STATS =====

Total hadiah : 0 OC

Jumlah stage : 0

Damage diberikan : 100

Damage diterima : 50

10. F10-Shop & Currency

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
# Masuk Shop
>>> SHOP
```

Irasshaimase! Selamat datang di SHOP!!

```
>>> Pilih aksi (lihat/beli/keluar):
```

```
# Melihat monster
```

```
>>> Pilih aksi (lihat/beli/keluar): lihat
```

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

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pokemon Air	10	1000	200	1	100
2	Pokemon Api	20	1000	200	5	20
3	Pokemon Tanah	30	430	100	0	300

```
>> Pilih aksi (lihat/beli/keluar):
```

```
# Melihat potion
```

```
>>> Pilih aksi (lihat/beli/keluar): lihat
```

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

ID	Type	Stok	Harga
1	Strength Potion	1	20
2	Resilience Potion	5	300

```
>> Pilih aksi (lihat/beli/keluar):
```

```
# Membeli monster
```

```
>>> Pilih aksi (lihat/beli/keluar): beli
```

Jumlah O.W.C.A. Coin-mu sekarang 900.

```
>>> Mau beli apa? (monster/potion): monster
```

```
>>> Masukkan id monster: 1
```

Berhasil membeli item: Pokemon Air. Item sudah masuk ke inventory-mu!

```
>> Pilih aksi (lihat/beli/keluar):
```

```
# Membeli potion
```

```
>>> Pilih aksi (lihat/beli/keluar): beli
```

Jumlah O.W.C.A. Coin-mu sekarang 800.

```
>>> Mau beli apa? (monster/potion): potion
```

```
>>> Masukkan id potion: 2
```

```
>>> Masukkan jumlah: 2
```

Berhasil membeli item: 2 Potion of Defense. Item sudah masuk ke inventory-mu!

```
>> Pilih aksi (lihat/beli/keluar):
```

```
# Membeli: gagal dalam validasi
```

```
>>> Pilih aksi (lihat/beli/keluar): beli
```

Jumlah O.W.C.A. Coin-mu sekarang 200.

```
>>> Mau beli apa? (monster/potion): potion
```

```
>>> Masukkan id potion: 2
```

```
>>> Masukkan jumlah: 1
```

OC-mu tidak cukup.

>> Pilih aksi (lihat/beli/keluar):
<pre># Membeli monster: gagal dalam validasi inventory >>> Pilih aksi (lihat/beli/keluar): beli Jumlah O.W.C.A. Coin-mu sekarang 200. >>> Mau beli apa? (monster/potion): monster >>> Masukkan id monster: 1 Monster Pokemon Air sudah ada dalam inventory-mu! Pembelian dibatalkan. >> Pilih aksi (lihat/beli/keluar):</pre>
<pre>>>> Pilih aksi (lihat/beli/keluar): keluar Mr. Yanto bilang makasih, belanja lagi ya nanti :) >>></pre>

11. F11-Laboratory

<pre># DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja >>> LABORATORY Selamat datang di Lab Dokter Asep !!! ===== MONSTER LIST ===== 1. Chacha (Level: 1) 2. Pikachow (Level: 2) 3. Zeze (Level: 5) ===== UPGRADE PRICE ===== 1. Level 1 -> Level 2: 300 OC 2. Level 2 -> Level 3: 500 OC 3. Level 3 -> Level 4: 800 OC 4. Level 4 -> Level 5: 1000</pre>
<pre>>>> Pilih monster: 1 # Level yang dapat dipilih hanya 2 - 5 karena level Chaca saat ini 1 Chaca akan di-upgrade ke level 2. Harga untuk melakukan upgrade Chaca adalah 300 OC. >>> Lanjutkan upgrade (Y/N): Y Selamat, Chaca berhasil di-upgrade ke level 2 !</pre>
<pre>>>> Pilih monster: 3 Maaf, monster yang Anda pilih sudah memiliki level maksimum</pre>
<pre>>>> Pilih monster: 2 # Level yang dapat dipilih hanya 2 - 5 karena level Chaca saat ini 1 Pikachow akan di-upgrade ke level 3. Harga untuk melakukan upgrade Pikachow adalah 500 OC. >>> Lanjutkan upgrade (Y/N): Y</pre>

Selamat, Pikachu berhasil di-upgrade ke level 3 !

12. F12-Shop Management

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
# Masuk Shop
>>> SHOP

Irasshaimase! Selamat datang kembali, Mr. Monogram!

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

# Melihat monster yang terdaftar di Shop
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): monster
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pokemon Air     | 10        | 1000      | 200  | 1    | 100
2  | Pokemon Api     | 20        | 1000      | 200  | 5    | 20
3  | Pokemon Tanah   | 30        | 430       | 100  | 0    | 300

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

# Melihat potion yang terdaftar di Shop
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type           | Stok | Harga
1  | Strength Potion | 1    | 20
2  | Resilience Potion | 5    | 300

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

# Menambah monster yang akan dijual
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/potion): monster
# Menampilkan seluruh monster yang ada di database tetapi belum ada pada shop
ID | Type   | ATK Power | DEF Power | HP   |
4  | Cici   | 10        | 1000      | 200  |
5  | Moskov | 20        | 1000      | 200  |
6  | Selen  | 30        | 430       | 100  |

# Jangan lupa lakukan validasi input
>>> Masukkan id monster: 4
>>> Masukkan stok awal: 5
>>> Masukkan harga: 555
Cici telah berhasil ditambahkan ke dalam shop!

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

# Menambah potion yang akan dijual
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
>>> Mau nambahin apa? (monster/potion): potion
# Menampilkan seluruh potion yang ada di database tetapi belum ada pada shop
ID | Type   |
3  | Healing Potion |

# Jangan lupa lakukan validasi input
>>> Masukkan id potion: 3
>>> Masukkan stok awal: 10
```

```

>>> Masukkan harga: 140
Healing Potion telah berhasil ditambahkan ke dalam shop!

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

# Mengubah stok / harga dari monster
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion): monster
# Menampilkan seluruh monster yang ada di shop
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pokemon Air     | 10        | 1000      | 200 | 1    | 100
2  | Pokemon Api     | 20        | 1000      | 200 | 5    | 20
3  | Pokemon Tanah  | 30        | 430       | 100 | 0    | 300

# Jangan lupa lakukan validasi input
>>> Masukkan id monster: 1
>>> Masukkan stok baru: 5
>>> Masukkan harga baru: 1200
Pokemon Air telah berhasil diubah dengan stok baru sejumlah 5 dan dengan harga baru 1200!

# Bisa dilakukan perubahan pada minimal satu atribut saja
>>> Masukkan id monster: 1
>>> Masukkan stok baru: 5
>>> Masukkan harga baru:
Pokemon Air telah berhasil diubah dengan stok baru sejumlah 5!

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

# Mengubah stok / harga dari potion
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
>>> Mau ubah apa? (monster/potion): potion
# Menampilkan seluruh potion yang ada di shop
ID | Type           | Stok | Harga
1  | Strength Potion | 1    | 20
2  | Resilience Potion | 5    | 300

# Jangan lupa lakukan validasi input
>>> Masukkan id potion: 1
>>> Masukkan stok baru: 10
>>> Masukkan harga baru: 35
Strength Potion telah berhasil diubah dengan stok baru sejumlah 10 dan dengan harga baru 35!

# Bisa dilakukan perubahan pada minimal satu atribut saja
>>> Masukkan id potion: 1
>>> Masukkan stok baru:
>>> Masukkan harga baru: 50
Strength Potion telah berhasil diubah dengan stok baru sejumlah 50!

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

# Menghapus monster dari Shop
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): monster
# Menampilkan seluruh monster yang ada di shop
ID | Type           | ATK Power | DEF Power | HP   | Stok | Harga
1  | Pokemon Air     | 10        | 1000      | 200 | 1    | 100
2  | Pokemon Api     | 20        | 1000      | 200 | 5    | 20
3  | Pokemon Tanah  | 30        | 430       | 100 | 0    | 300

```

```
# Jangan lupa lakukan validasi input
>>> Masukkan id monster: 1
>>> Apakah anda yakin ingin menghapus Pokemon Air dari shop (y/n)? y
Pokemon Air telah berhasil dihapus dari shop!

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

# Menghapus potion dari Shop
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): potion
# Menampilkan seluruh potion yang ada di shop
ID | Type | Stok | Harga
1 | Strength Potion | 1 | 20
2 | Resilience Potion | 5 | 300

# Jangan lupa lakukan validasi input
>>> Masukkan id potion: 1
>>> Apakah anda yakin ingin menghapus Strength Potion dari shop (y/n)? y
Strength Potion telah berhasil dihapus dari shop!

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

# Keluar dari Shop Management
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): keluar
Dadah Mr. Yanto, sampai jumpa lagi!

>>>
```

13. F13-Monster Management

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
>>> MONSTER
SELAMAT DATANG DI DATABASE PARA MONSTER !!!
1. Tampilkan semua Monster
2. Tambah Monster baru

# Tampilkan semua Monster
>>> Pilih Aksi : 1
ID | Type | ATK Power | DEF Power | HP
1 | Pokemon Air | 10 | 10 | 200
2 | Pokemon Api | 20 | 20 | 200
3 | Pokemon Tanah | 30 | 30 | 100

# Tambah Monster Baru
>>> Pilih Aksi : 1
Memulai pembuatan monster baru

# Duplikat di nama monster
>>> Masukkan Type / Nama : Pokemon Air
Nama sudah terdaftar, coba lagi!

>>> Masukkan Type / Nama : Pokemon Udara

# Duplikat di nama monster
>>> Masukkan ATK Power : Udara
Masukkan input bertipe Integer, coba lagi!

>>> Masukkan ATK Power : 20
```



```
# DEF Power tidak sesuai
>>> Masukkan DEF Power (0-50) : 500
DEF Power harus bernilai 0-50, coba lagi!

>>> Masukkan DEF Power (0-50) : 50
>>> Masukkan HP : 200

Monster baru berhasil dibuat!
Type : Pokemon Udara
ATK Power : 20
DEF Power : 50
HP :200
>>> Tambahkan Monster ke database (Y/N) : Y
Monster baru telah ditambahkan!

# Cek Database
>>> Pilih Aksi : 1
```

ID	Type	ATK Power	DEF Power	HP
1	Pokemon Air	10	20	200
2	Pokemon Api	20	10	200
3	Pokemon Tanah	30	30	100
4	Pokemon Udara	20	50	200

```
# Opsi N pada konfirmasi penambahan monster
>>> Tambahkan Monster ke database (Y/N) : N
Monster gagal ditambahkan!
>>> Pilih Aksi : 1
```

ID	Type	ATK Power	DEF Power	HP
1	Pokemon Air	10	20	200
2	Pokemon Api	20	10	200
3	Pokemon Tanah	30	30	100

14. F14-Load

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
~$ python main.py nama_folder
# parent folder dari nama_folder akan sama seperti dalam fungsi save

Loading...
# panggil prosedur load data
Selamat datang di program OWCA!
# meminta perintah berikutnya... (cth : register, login, dll)

# user tidak memberikan nama folder
~$ python main.py

Tidak ada nama folder yang diberikan!
Usage : python main.py <nama_folder>
# program keluar

# user memberikan folder yang tidak ada
~$ python main.py folder_palsu

Folder "folder_palsu" tidak ditemukan.
# program keluar
```

15. F15-Save

```
# nama folder tidak ditemukan, folder belum dibuat
>>> SAVE
```

```
Masukkan nama folder: 09-03-2024
# Folder 09-03-2024 belum ada
```

```
Saving...
```

```
Membuat folder data/09-03-2024...
Berhasil menyimpan data di folder data/09-03-2024!
```

```
# nama folder ditemukan
>>> SAVE
```

```
Masukkan nama folder: 09-03-2024
# Folder 09-03-2024 sudah ada
```

```
Saving...
```

```
Berhasil menyimpan data di folder data/09-03-2024!
# apabila program melakukan overwrite/replace, tidak diperlukan pesan tambahan
```

```
>>> SAVE
```

```
Masukkan nama folder: 09-03-2024
# Folder ./save belum ada
```

```
Saving...
```

```
Membuat folder data...
Membuat folder data/09-03-2024...
Berhasil menyimpan data di folder data/09-03-2024!
```

16. F16-Exit

```
>>> EXIT
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) n
# Keluar program
```

```
>>> EXIT
# Contoh input tidak valid
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) a
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) Y
# Menjalankan prosedur save (F15) dan keluar program
```

BAB 5

DESAIN KAMUS DATA, DEKOMPOSISI, SPESIFIKASI UNTUK TIAP MODUL/PROSEDUR/FUNGSI

F00

PROGRAM RANDOM NUMBER GENERATOR

{Spesifikasi : Memberikan angka random sesuai dengan algoritma LCG }

KAMUS

max_value, min_value, seed, random_number : integer

constant a : integer = 110351524512

constant c : integer = 12345

constant m : integer = 2**31

ALGORITMA

import time

seed = time.time()

a = 110351524512

c = 12345

m = 2**31

random_number <- ((seed * a + c) % m) / (m - 1)

return int(random_number * (max_value - min_value + 1)) + min_value

F01

PROGRAM REGISTER

{ Spesifikasi : Mendaftarkan pengguna baru dengan nama pengguna unik dan menetapkan monster awal kepada pengguna tersebut. }

{ Asumsi : semua array, kecuali diatur, telah memiliki banyak elemen yang benar, fungsi bawaan Python sudah ada di notal. }

KAMUS

{ Deklarasi variabel }

type userline :

< id: integer;

username: string;

password: string;

role: string;

oc: string >

type monsterline :

< id: string;

type: string>

type inventoryline :

< user_id: string;

monster_id: string;

level: string >

```

list_user : array of userline
list_monster: array of monsterline
list_monster_inventory : array of inventoryline
user_data : array of userline
monster_data: array of monsterline
monster_inventory: array of inventoryline
new_user_id: integer
username: string
password: string
selected_monster_id: string
full: boolean
monster_benar: boolean

```

{ Deklarasi Fungsi dan Prosedur }

```

function check_valid (username : string) -> boolean
{ SPEK : Mengembalikan true jika username hanya mengandung karakter yang diizinkan (0-9, a-z, A-Z, _, -).}
{ I.S. username }
{ F.S. sebuah boolean }

```

```

function custom_zip (headers: array of string, row: array of string) -> dict
{ SPEK : Menggabungkan headers dan row menjadi dictionary (asumsi import dari src.F05).}
{ I.S. headers dan row }
{ F.S. dictionary }

```

```

function custom_isdigit (value: string) -> boolean
{ SPEK : Mengembalikan true jika string hanya berisi digit (asumsi import dari src.F05).}
{ I.S. string }
{ F.S. sebuah boolean }

```

ALGORITMA

{ Inisialisasi Data }

```

convert list_user, list_monster, list_monster_inventory to array of string.
convert list_user, list_monster, list_monster_inventory to array of dictionary

```

{ Inisialisasi new_user_id }

```
new_user_id <- 1
```

{ Proses Registrasi User }

```

while True do
  output ("Masukkan username: "), input (username)
  if not check_valid(username) then
    output ("Username hanya boleh mengandung huruf kapital/kecil, angka, garis bawah _, dan garis hubung -")
    continue
  full <- False
  for user in user_data do
    if user['username'] == username then
      output ("Username sudah terdaftar! Silakan coba masukkan username lain")
      full <- True
      continue
  if full then
    continue
  output ("Masukkan password: "), input (password)

```

```

        for user in user_data do
            while user['id'] == str(new_user_id) do
                new_user_id <- new_user_id + 1
            break
        { Pilih Monster }

while True do
    output ("Pilih monster sebagai monster awalmu:")
    for monster in monster_data do
        output (monster['id']. " ". monster['type'])
    monster_benar <- False
    output ("Masukkan ID monster yang ingin dipilih: "), input
(selected_monster_id)
    if not custom_isdigit(selected_monster_id) then
        output ("Masukkan format valid!")
        continue
    for monster in monster_data do
        if monster['id'] == selected_monster_id then
            monster_benar <- True
    if not monster_benar then
        output ("Monster ID tidak ada! Silakan masukkan ID monster yang
tersedia.")
        continue
    new_user <- {'id': str(new_user_id), 'username': username, 'password':
password, 'role': 'agent', 'oc': '0'}
    append new_user to user_data

    new_inventory <- {'user_id': new_user_id, 'monster_id': selected_monster_id,
'level': '1'}
    append new_inventory to monster_inventory
    break

{ Tulis Data yang Diperbarui}

headers <- keys of user_data[0]
list_user <- [headers] + [[d[key] for key in headers] for d in user_data]
headers <- keys of monster_inventory[0]
list_monster_inventory <- [headers] + [[d[key] for key in headers] for d in
monster_inventory]
convert elements of list_user and list_monster_inventory to integer

{ Output Hasil Registrasi }

output ("User " + username + " successfully registered with user ID " +
str(new_user_id) + " and selected monster ID " + selected_monster_id.)
return

return list_user, list_monster_inventory

```

F02

PROGRAM LOGIN

{ Spesifikasi: Memeriksa login user berdasarkan username dan password. }
{ Asumsi: Semua array telah memiliki banyak elemen yang benar, fungsi bawaan Python sudah ada di notal. }

KAMUS

```

{ Deklarasi variabel }
  type userline:
    < id: integer;
      username: string;
      password: string;
      role: string;
      oc: string >

  list_user : array of userline
  username : string
  password : string
  user_id : integer
  i: integer

ALGORITMA
  output ("Masukkan username: "), input (username)

  { Pencarian Username }

  for i <- 1 to len(list_user) - 1 do
    if username == list_user[i][1] then
      output ("Masukkan password: "), input (password)
    if password == list_user[i][2] then
      output ("Selamat datang " + list_user[i][3] + " " + list_user[i][1])
      user_id <- list_user[i][0]
      return user_id
    else
      output ("Password salah")
      break
    else
      output ("Username tidak terdaftar")
      break
  return

```

F03

```

PROGRAM LOGOUT
{ Spesifikasi: Melakukan logout untuk user yang sedang login. }
{ Asumsi: Fungsi bawaan Python sudah ada di notal. }

```

KAMUS

login_id: integer or None

ALGORITMA

```

{ Cek status login }
if login_id then
  output ("Kamu telah logout")

else
  output ("Tidak bisa logout karena kamu belum login")

```

F04

PROGRAM HELP

```

{ Spesifikasi: Menampilkan bantuan berdasarkan status login dan role pengguna. }
{ Asumsi: Fungsi bawaan Python sudah ada di notal. }

```

KAMUS

login_id: integer or None
list_user: array of array of string
headers: array of string
data: array of array of string
user_data: array of dictionary
role: string
i: integer

ALGORITMA

```
{ Konversi list_user menjadi list of dictionary }

headers <- list_user[0]
data <- []
for i <- 1 to panjang(list_user) - 1 do
  data.append(list_user[i])
user_data <- []
user_data <- for row in data do
  dict(zip(headers, row))
user_data <- u for u in user_data if u['id'] == str(login_id)

if user_data kosong then
  output ("Kamu belum login sebagai role apapun. Silahkan login
  terlebih dahulu.\n")
  output ("Login: Masuk ke dalam akun yang sudah terdaftar\n")
  output ("Register: Membuat akun baru\n")
  output ("\nFootnote:")
  output ("Untuk menggunakan aplikasi, silahkan masukkan nama fungsi
  yang terdaftar\n")
  output ("Jangan lupa untuk memasukkan input yang valid")
  return
{ Ambil data user yang terlogin }

user_data <- user_data[0]

{ Cek role user }

role <- lower(user_data['role'])

{ Tampilkan bantuan berdasarkan role }

if role == 'agent' then
  output ("Halo Agent Purry. Kamu memanggil command HELP. \nKamu
  memilih jalan yang benar, semoga kamu tidak sesat kemudian. \nBerikut
  adalah hal-hal yang dapat kamu lakukan sekarang:")
  output ("Logout: Keluar dari akun yang sedang digunakan\n")
  output ("Inventory: Melihat owca-dex yang dimiliki oleh Agent\n")
  output ("\nFootnote:")
  output ("Untuk menggunakan aplikasi, silahkan masukkan nama fungsi
  yang terdaftar\n")
  output ("Jangan lupa untuk memasukkan input yang valid")
if role == 'admin' then
  output ("Selamat datang, Admin. Berikut adalah hal-hal yang dapat
  lakukan:\n")
  output ("Logout: Keluar dari akun yang sedang digunakan\n")
  output ("Shop: Melakukan manajemen pada SHOP sebagai tempat jual beli
  peralatan Agent\n")
  output ("\nFootnote:")
  output ("Untuk menggunakan aplikasi, silahkan masukkan nama fungsi
  yang terdaftar")
```

```

        output ("Jangan lupa untuk memasukkan input yang valid")
    else
        output ("Kamu belum login sebagai role apapun. Silahkan login
        terlebih dahulu.")
        output ("Login: Masuk ke dalam akun yang sudah terdaftar")
        output ("Register: Membuat akun baru")
        output ("\nFootnote:")
        output ("Untuk menggunakan aplikasi, silahkan masukkan nama fungsi
        yang terdaftar")
        output ("Jangan lupa untuk memasukkan input yang valid")

```

F05

PROGRAM MAXI

{ Spesifikasi: Mengembalikan nilai maksimum antara dua angka. }

KAMUS

```

a: integer
b: integer

```

function maxi (a,b)

{ Spesifikasi: Mengembalikan nilai maksimum antara dua angka. }

{ I.S. integer }

{ F.S. nilai maksimum }

ALGORITMA

```

    if a > b then
        a
    else
        b

```

PROGRAM MINI

{ Spesifikasi: Mengembalikan nilai minimum antara dua angka. }

KAMUS

```

a: integer
b: integer

```

function mini (a,b)

{ Spesifikasi: Mengembalikan nilai minimum antara dua angka. }

{ I.S. integer }

{ F.S. nilai minimum }

ALGORITMA

```

    if a < b then
        a
    else
        b

```

PROGRAM CUSTOM_ZIP

{ Spesifikasi: Menggabungkan elemen dari beberapa iterable menjadi tuple. }

KAMUS

```

args: array of iterable
iterators: array of iterator
sentinel: object
result: tuple

```


ALGORITMA

```

{ Inisialisasi iterators }

iterators <- [iter(iterable) for iterable in args]
sentinel <- object()

while True do
    result <- tuple(next(iterator, sentinel) for iterator in iterators)
    if any(val is sentinel for val in result) then
        return
    yield result

```

PROGRAM CUSTOM_ISDIGIT

{ Spesifikasi: Memeriksa apakah sebuah string merupakan digit. }

KAMUS

s: string or integer

ALGORITMA

```

if isinstance(s, int) then
    return True
return all('0' <= char <= '9' for char in str(s))

```

PROGRAM ADJUST

{ Spesifikasi: Menyesuaikan atribut monster berdasarkan levelnya. }

KAMUS

monster: dictionary
 level: integer
 factor: float

ALGORITMA

```

level <- int(monster['level'])
if level == 1 then
    return monster
else
    factor <- 1 + (level - 1) * 0.1
    monster['atk_power'] <- int(int(monster['atk_power']) * factor)
    monster['def_power'] <- int(int(monster['def_power']) * factor)
    monster['hp'] <- int(int(monster['hp']) * factor)
    return monster

```

PROGRAM ATTACK

{ Spesifikasi: Melakukan serangan dari satu monster ke monster lain. }

KAMUS

attacker: dictionary
 defender: dictionary
 n: integer
 atk_multiplier: float
 attacker_atk_power: integer
 defender_def_power: integer
 damage: integer

ALGORITMA

```

import random_num from src.F00
n <- random_num(-30, 30)
atk_multiplier <- 1 + n / 100
attacker_atk_power <- int(attacker['atk_power']) * atk_multiplier
defender_def_power <- mini(int(defender['def_power']), 50)
damage <- int(attacker_atk_power * (1 - defender_def_power / 100))

```

```
defender['hp'] <- maxi(int(defender['hp']) - maxi(damage, 0), 0)
return attacker, defender
```

PROGRAM ATTACK_ARENA

{ Spesifikasi: Melakukan serangan dari satu monster ke monster lain di arena. }

KAMUS

```
attacker: dictionary
defender: dictionary
n: integer
atk_multiplier: float
attacker_atk_power: integer
defender_def_power: integer
damage: integer
```

ALGORITMA

```
import random_num from src.F00
n <- random_num(-30, 30)
atk_multiplier <- 1 + n / 100
attacker_atk_power <- int(attacker['atk_power']) * atk_multiplier
defender_def_power <- mini(int(defender['def_power']), 50)
damage <- int(attacker_atk_power * (1 - defender_def_power / 100))
defender['hp'] <- maxi(int(defender['hp']) - maxi(damage, 0), 0)
return attacker, defender, damage
```

F06

PROGRAM MINI

{ Spesifikasi: Mengembalikan nilai terkecil antara dua bilangan. }

{ Asumsi: Fungsi bawaan Python sudah ada di notal. }

KAMUS

```
a: integer
b: integer
```

function mini (a,b)

{ Spesifikasi: Mengembalikan nilai terkecil antara dua bilangan. }

{ I.S. integer }

{ F.S. nilai minimum }

ALGORITMA

```
{ Cek nilai `a` dan `b` }
  if `a` < `b` then
    return `a`
  else
    return `b`
```

PROGRAM POTION

{ Spesifikasi: Menggunakan potion pada monster pengguna yang sedang login. }

{ Asumsi: Fungsi bawaan Python sudah ada di notal. }

KAMUS

```
selected_potion_name : string
selected_user_monster : dictionary
login_id : string
potion_inventory : array of dictionary
monster_data : array of dictionary
potion : dictionary
```

```
level : integer
factor : float
monster : dictionary
```

```
function potion (selected_potion_name: string, selected_user_monster: dictionary, login_id:
string, potion_inventory: array of dictionary, monster_data: array of dictionary)
{ Spesifikasi : Mengubah menjadi dictionary dan array of dictionary }
{ I.S. string, dictionary, array of dictionary }
{ F.S. dictionary, array of dictionary }
```

ALGORITMA

```
  for each potion in potion_inventory :
    if potion['user_id'] = login_id and potion['type'] = selected_potion_name then
      if int(potion['quantity']) > 0 then
        if selected_potion_name.lower() = 'strength' then
          selected_user_monster['atk_power'] <-
            str(int(int(selected_user_monster['atk_power']) * 1.05))
          potion['quantity'] <- str(int(potion['quantity']) - 1)
        else if selected_potion_name.lower() = 'resilience' then
          selected_user_monster['def_power']
            str(int(int(selected_user_monster['def_power']) * 1.05))
          potion['quantity'] <- str(int(potion['quantity']) - 1)
        else if selected_potion_name.lower() = 'healing' then
          level <- int(selected_user_monster['level'])
          factor <- 1 + (level - 1) * 0.1
          monster <- [m for m in monster_data if selected_user_monster['id'] =
            m['id']][0]
          selected_user_monster['hp'] <-
            str(mini(int(int(selected_user_monster['hp']) * 1.25),
              int(int(monster['hp']) * factor)))
          potion['quantity'] <- str(int(potion['quantity']) - 1)
        else
          output("Potion habis")
  return selected_user_monster, potion_inventory
```

F07

PROGRAM INVENTORY

```
{ Spesifikasi: Menampilkan inventarisasi barang dan monster untuk user yang sedang login. }
{ Asumsi: Fungsi bawaan Python sudah ada di notal. }
```

KAMUS

```
function custom_zip (arr1 : array, arr2 : array) -> array
{ Spesifikasi: Menjalankan sesuai dengan fungsi bawaan python zip() }
{ I.S.: tuple }
{ F.S.: tuple hasil }
function custom_isdigit (s : char) -> boolean
{ Spesifikasi: Memeriksa apakah digit angka }
{ I.S.: character }
{ F.S.: boolean }
procedure display_shop_items (items : array)
{ Spesifikasi: Menampilkan daftar item dengan nomor urut }
{ I.S.: Daftar item }
{ F.S.: Daftar item ditampilkan di layar }
procedure inventory (login_id : string, list_user : 2D array, list_monster : 2D array,
list_item_inventory : 2D array, list_monster_inventory : 2D array)
{ Spesifikasi: Menampilkan inventory berdasarkan login_id }
```

```

{ I.S.: Data pengguna, monster, item inventory, dan monster inventory }
{ F.S.: Inventory ditampilkan di layar, atau pesan kesalahan jika login_id tidak valid }
{ Deklarasi variabel }
    login_id: string
    list_user: list of lists
    list_monster: list of lists
    list_item_inventory: list of lists
    list_monster_inventory: list of lists
    headers: list
    data: list
    user_data: list of dictionaries
    user_login: dictionary
    role: string
    monster_data: list of dictionaries
    potion_inventory: list of dictionaries
    monster_inventory: list of dictionaries
    item_type: string
    potion_info: list of strings
    selected_potion_info: string
    monster_inventory_details: list of strings
    mns_info: list of strings
    mns_inv: list of dictionaries
    selected_monster_info: string
    idx: string

```

ALGORITMA

```

output("\\nHalo, user_id ", login_id, " selamat datang di inventory Anda!")
exit <- False
while not exit do
    action <- input("\\nPilih aksi (lihat/quit): ")
    while not isActionValid(action) do
        action <- input("Input tidak valid. Ulangi\\nPilih aksi (lihat/quit): ")

    if action.lower() = "lihat" then
        choice <- input("Mau lihat apa? (potion/monster): ")
        while not isMonsterOrPotion(choice) do
            choice <- input("Input tidak valid. Ulangi\\nMau lihat apa? (potion/monster): ")

        if choice.lower() = "potion" then
            output("Potion yang tersedia: ")
            potion_info <- [f>Nama: ", potion['type'], ", Jumlah: ", potion['quantity']]
            for potion in potion_inventory if potion['user_id'] = str(login_id)
            if len(potion_info) > 0 then
                display_shop_items(potion_info)
                output("Pilih nomor urut potion untuk informasi lebih lanjut: ")
                while True do
                    idx <- input()
                    if not custom_isdigit(idx) then
                        output("Pilihan tidak valid, silakan masukkan angka")
                        continue
                    idx <- int(idx) - 1
                    if idx < 0 or idx >= len(potion_info) then
                        output("Pilihan tidak valid, silakan masukkan nomor yang tersedia")
                        continue
                    selected_potion_info <- potion_info[idx]
                    output(selected_potion_info)
                    break
            else
                output("Anda tidak mempunyai potion apapun di inventory")

```

```

else if choice.lower() = "monster" then
  output("Monster yang tersedia: ")
  monster_inventory_details <- []
  mns_info <- []
  mns_inv <- [monster for monster in monster_inventory if monster['user_id']
str(login_id)]
  for monster in mns_inv :
    monster_info <- [m for m in monster_data if m['id'] =
monster['monster_id']][0]
    monster_inventory_details.append(f"{monster['monster_id']}.
{monster_info['type']} (HP: {monster_info['hp']})")
    mns_info.append(f"{monster['monster_id']}. {monster_info['type']} (ATK:
{monster_info['atk_power']}, DEF: {monster_info['def_power']}, HP:
{monster_info['hp']})")
  display_shop_items(monster_inventory_details)
  output("Pilih nomor urut monster untuk informasi lebih lanjut: ")
  while True do
    idx <- input()
    if not custom_isdigit(idx) then
      output("Pilihan tidak valid, silakan masukkan angka")
      continue
    idx <- int(idx) - 1
    if idx < 0 or idx >= len(mns_info) then
      output("Pilihan tidak valid, silakan masukkan nomor yang tersedia")
      continue
    selected_monster_info <- mns_info[idx]
    output(selected_monster_info)
    break

else if action.lower() = "quit" then
  output("Sampai jumpa lagi!")
  exit <- True
else
  output("Input tidak valid, silakan pilih aksi yang tersedia.")
return

```

F08

PROGRAM BATTLE

{Spesifikasi: Menjalankan pertarungan antara monster pengguna dan monster lawan dalam arena.}

{I.S.: Data list pengguna, monster, inventory item, dan inventory monster sebelum pertarungan.}

{F.S.: Data list pengguna, monster, inventory item, dan inventory monster setelah pertarungan.}

KAMUS

function random_num (min_val : integer, max_val : integer) -> integer

{ Spesifikasi: Menghasilkan angka acak antara min_val dan max_val }

{ I.S.: Rentang angka }

{ F.S.: Angka acak }

function adjust (monster : dictionary) -> dictionary

{ Spesifikasi: Menyesuaikan atribut monster berdasarkan level }

{ I.S.: Monster dengan atribut sebelum disesuaikan }

{ F.S.: Monster dengan atribut setelah disesuaikan }

function attack (attacker : dictionary, defender : dictionary) -> attacker : dictionary,

```

defender : dictionary
{ Spesifikasi: Menjalankan serangan dari attacker ke defender }
{ I.S.: Attacker dan defender sebelum serangan }
{ F.S.: Attacker dan defender setelah serangan }
function custom_zip (headers : array, row : array) -> dictionary
{ Spesifikasi: Menggabungkan dua array menjadi dictionary }
{ I.S.: Headers dan row }
{ F.S.: Dictionary hasil penggabungan headers dan row }
function custom_isdigit (s : string) -> boolean
{ Spesifikasi: Memeriksa apakah string merupakan digit }
{ I.S.: String }
{ F.S.: Boolean }
function potion (selected_potion_name : string, selected_user_monster : dictionary,
login_id : string, potion_inventory : 2D array, monster_data : 2D array) ->
selected_user_monster : dictionary, potion_inventory : 2D array
{ Spesifikasi: Menggunakan potion untuk meningkatkan atribut monster }
{ I.S.: Potion yang tersedia, monster yang dipilih }
{ F.S.: Atribut monster yang diperbarui, inventory potion yang diperbarui }
procedure sleep (duration : integer)
{ Spesifikasi: Menghentikan eksekusi program selama durasi tertentu }
{ I.S.: Durasi dalam detik }
{ F.S.: Program berhenti sementara }
{ Deklarasi variabel }
    login_id : string
    list_user : 2D array
    list_monster : 2D array
    list_item_inventory : 2D array
    list_monster_inventory : 2D array
    headers : array
    data : array
    user_data : array of dictionaries
    user_login : dictionary
    role : string
    monster_data : array of dictionaries
    potion_inventory : array of dictionaries
    monster_inventory : array of dictionaries
    mns : array of dictionaries
    opponent_monster : dictionary
    user_monsters : array of dictionaries
    selected_user_monster : dictionary
    user_choice : integer
    is_strength_used : boolean
    is_healing_used : boolean
    is_resilience_used : boolean
    user_quit : boolean
    ronde : integer
    action : string
    user_potions : array of dictionaries
    potion_choice : integer
    selected_potion : string
    oc_coin : integer

ALGORITMA
output("nSelamat datang di arena pertarungan, user_id ", login_id, "!")
exit <- False
while not exit do
    # Konversi List ke String dan Buat Dictionary
    list_user = [[str(item) for item in row] for row in list_user]
    list_monster = [[str(item) for item in row] for row in list_monster]
    list_item_inventory = [[str(item) for item in row] for row in list_item_inventory]

```

```

        list_monster_inventory = [[str(item) for item in row] for row in
list_monster_inventory]

headers <- list_user[0]
data <- []
for i from 1 to len(list_user)-1 :
    data.append(list_user[i])
user_data <- [dict(custom_zip(headers, row)) for row in data]

user_login <- [u for u in user_data if u['id'] == str(login_id)]
if user_login then
    user_login <- user_login[0]
else:
    output("User_id tidak terdaftar!")
    list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row
in list_user]
    list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for
row in list_monster]
    list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in
row] for row in list_item_inventory]
    list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item
row] for row in list_monster_inventory]
    return list_user, list_monster, list_item_inventory, list_monster_inventory

role <- str(user_login['role']).lower()
if role != 'agent' then
    output("Yah, hanya agent saja yang boleh masuk Battle.")
    list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row
in list_user]
    list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for
row in list_monster]
    list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in
row] for row in list_item_inventory]
    list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in
row] for row in list_monster_inventory]
    return list_user, list_monster, list_item_inventory, list_monster_inventory

headers <- list_monster[0]
data <- []
for i from 1 to len(list_monster)-1 :
    data.append(list_monster[i])
monster_data <- [dict(custom_zip(headers, row)) for row in data]

headers <- list_item_inventory[0]
data <- []
for i from 1 to len(list_item_inventory)-1 :
    data.append(list_item_inventory[i])
potion_inventory <- [dict(custom_zip(headers, row)) for row in data]

headers <- list_monster_inventory[0]
data <- []
for i from 1 to len(list_monster_inventory)-1 :
    data.append(list_monster_inventory[i])
monster_inventory <- [dict(custom_zip(headers, row)) for row in data]

mns <- []
for monster_inv in monster_inventory :
    for monster_data_entry in monster_data :
        if monster_inv['monster_id'] == monster_data_entry['id'] then
            monster <- {**monster_data_entry, 'level': monster_inv['level']}

```

```

        monster <- adjust(monster)
        mns.append(monster)

opponent_idx <- random_num(0, len(mns)-1)
opponent_monster <- mns[opponent_idx]
opponent_monster['level'] <- str(random_num(1, 5))
opponent_monster <- adjust(opponent_monster)
sleep(1)
output("\\nMonster Lawan:")
output("Nama: ", opponent_monster['type'])
output("ATK Power: ", opponent_monster['atk_power'])
output("DEF Power: ", opponent_monster['def_power'])
output("HP: ", opponent_monster['hp'])
output("Level: ", opponent_monster['level'])
sleep(1)

user_monsters <- []
for monster_inv in monster_inventory :
    for monster_data_entry in monster_data :
        if monster_inv['monster_id'] == monster_data_entry['id'] and
        monster_inv['user_id'] == login_id then
            monster <- {**monster_data_entry, 'level': monster_inv['level']}
            monster <- adjust(monster)
            user_monsters.append(monster)

output("\\nMonster Anda:")
for idx from 0 to len(user_monsters)-1 :
    user_monster <- user_monsters[idx]
    output(f"{idx + 1}. Name: {user_monster['type']} | ATK Power:
    {user_monster['atk_power']} | DEF Power: {user_monster['def_power']} |
    HP: {user_monster['hp']} | Level: {user_monster['level']}")
while True do
    user_choice <- input("\\nPilih monster untuk dipertarungkan: ")
    if custom_isdigit(user_choice) then
        user_choice <- int(user_choice) - 1
        if len(user_monsters) - 1 >= user_choice then
            break

selected_user_monster <- user_monsters[user_choice]
output("\\nAnda memilih:")
output("Nama: ", selected_user_monster['type'])
output("ATK Power: ", selected_user_monster['atk_power'])
output("DEF Power: ", selected_user_monster['def_power'])
output("HP: ", selected_user_monster['hp'])
output("Level: ", selected_user_monster['level'])
sleep(1)

is_strength_used <- False
is_healing_used <- False
is_resilience_used <- False
user_quit <- False
ronde <- 0
while int(opponent_monster['hp']) > 0 and int(selected_user_monster['hp']) > 0 do
    sleep(1)
    output(f"\\nRONDE {ronde + 1}")
    sleep(1)
    output("\\nGiliran Anda menyerang monster lawan!")
    sleep(1)
    action <- input("Pilih aksi Anda - Attack, Potion, Quit: ").lower()

```



```

if action == 'attack' then
    selected_user_monster, opponent_monster <- attack(selected_user_monster,
    opponent_monster)
    output(f"{selected_user_monster['type']} menyerang monster
    {opponent_monster['type']}. HYAH!")
    sleep(1)
    output(f"MONSTER LAWAN ({opponent_monster['type']}): ")
    output(f"ATK Power: {opponent_monster['atk_power']}")
    output(f"DEF Power: {opponent_monster['def_power']}")
    output(f"HP: {opponent_monster['hp']}")
    output(f"Level: {opponent_monster['level']}")
    sleep(1)
    if opponent_monster['hp'] <= 0 then
        break
else if action == 'potion' then
    output("\nPotion Tersedia:")
    user_potions <- [potion for potion in potion_inventory if potion['user_id'] ==
    login_id]
    if not user_potions:
        output("Anda tidak mempunyai potion apapun")
        continue
    for idx from 0 to len(user_potions)-1 :
        user_potion <- user_potions[idx]
        output(f"{idx + 1}. {user_potion['type']} - Quantity:
        {user_potion['quantity']}")
    while True do
        potion_choice <- input("\nPilih potion yang akan digunakan: ")
        if custom_isdigit(potion_choice) then
            potion_choice <- int(potion_choice) - 1
            if potion_choice <= len(user_potions) - 1 then
                break
        selected_potion <- user_potions[potion_choice]['type']

    if selected_potion == 'strength' and not is_strength_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
        selected_user_monster, login_id, potion_inventory, monster_data)
        is_strength_used <- True
    else if selected_potion == 'healing' and not is_healing_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
        selected_user_monster, login_id, potion_inventory, monster_data)
        is_healing_used <- True
    else if selected_potion == 'resilience' and not is_resilience_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
        selected_user_monster, login_id, potion_inventory, monster_data)
        is_resilience_used <- True
    else:
        output("You cannot use that potion again in this battle.")
        output("\nMonster Anda:")
        output("Name: ", selected_user_monster['type'])
        output("ATK Power: ", selected_user_monster['atk_power'])
        output("DEF Power: ", selected_user_monster['def_power'])
        output("HP: ", selected_user_monster['hp'])
        output("Level: ", selected_user_monster['level'])
        sleep(1)
        continue
else if action == 'quit' then
    user_quit <- True
    break
else:
    continue

```

```

if not user_quit then
  output(f"Sekarang giliran monster {opponent_monster['type']} melawan
  monstermu! HYAH!")
  opponent_monster, selected_user_monster <- attack(opponent_monster,
  selected_user_monster)
  sleep(1)
  output(f"MONSTER ANDA ({selected_user_monster['type']}): ")
  output(f"ATK Power: {selected_user_monster['atk_power']}")
  output(f"DEF Power: {selected_user_monster['def_power']}")
  output(f"HP: {selected_user_monster['hp']}")
  output(f"Level: {selected_user_monster['level']}")
  sleep(1)
  if opponent_monster['hp'] <= 0 then
    break
  else:
    ronde <- ronde + 1

if not user_quit then
  if opponent_monster['hp'] <= 0 then
    oc_coin <- random_num(10, 100)
    for user in user_data :
      if user['id'] == login_id then
        user['oc'] <- str(int(user['oc']) + oc_coin)
        output(f"Selamat, Anda Menang! Anda memperoleh {oc_coin} OC coins!")
        # Konversi balik ke List of List untuk menyimpan data
        headers <- list(user_data[0].keys())
        list_user <- [headers] + [[d[key] for key in headers] for d in user_data]

        headers <- list(monster_data[0].keys())
        list_monster <- [headers] + [[d[key] for key in headers] for d in monster_data]

        headers <- list(potion_inventory[0].keys())
        list_item_inventory <- [headers] + [[d[key] for key in headers] for d in
        potion_inventory]

        headers <- list(monster_inventory[0].keys())
        list_monster_inventory <- [headers] + [[d[key] for key in headers] for d in
        monster_inventory]

        list_user <- [[int(item) if custom_isdigit(item) else item for item in row]
        for row in list_user]
        list_monster <- [[int(item) if custom_isdigit(item) else item for item in row]
        for row in list_monster]
        list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item
        in row] for row in list_item_inventory]
        list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for
        item in row] for row in list_monster_inventory]

      return list_user, list_monster, list_item_inventory, list_monster_inventory
    else:
      output("Yah, Anda dikalahkan monster lawan! Jangan menyerah, terus kembangkan
      monstermu!")
      list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for
      row in list_user]
      list_monster <- [[int(item) if custom_isdigit(item) else item for item in row]
      for row in list_monster]
      list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item
      in row] for row in list_item_inventory]

```

```

        list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for
        item in row] for row in list_monster_inventory]
        return list_user, list_monster, list_item_inventory, list_monster_inventory
    else:
        output("Anda berhasil kabur dari pertarungan!")
        list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row
        in list_user]
        list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for
        row in list_monster]
        list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in
        row] for row in list_item_inventory]
        list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in
        row] for row in list_monster_inventory]
        return list_user, list_monster, list_item_inventory, list_monster_inventory

```

F09

PROGRAM ARENA

```

{Spesifikasi : Menjalankan opsi arena dalam program}
{I.S : Data list database sebelum update}
{F.S : Data list database sesudah update}

```

KAMUS

```

User_choice, potion_choice : integer
action : string
function random_num (min_value : integer, max_value : integer) -> integer
{ Spesifikasi : Mengembalikan angka random sesuai dengan LCG }
{ I.S. range angka }
{ F.S. angka random }
function adjust (monster : array) -> array
{ Spesifikasi : Menyesuaikan stat monster atk, def, dan hp }
{ I.S. stat monster sebelum update }
{ F.S. stat monster setelah update }
function attack_arena (attacker : array, defender : array) -> attacker : array, defender :
array, damage : integer
{ Spesifikasi : Menjalankan fungsi attack arena pada game }
{ I.S. musuh dan penyerang }
{ F.S. musuh, penyerang, dan damage yang diterima }
function custom_zip (arr1 : array, arr2 : array) -> array
{ Spesifikasi : Menjalankan sesuai dengan fungsi bawaan python zip() }
{ I.S. tuple }
{ F.S. tuple hasil }
function custom_isdigit (s : char) -> boolean
{ Spesifikasi : Memeriksa apakah digit angka }
{ I.S. character }
{ F.S. boolean }
function potion (selected_potion_name : string, selected_user_monster : array, login_id :
integer, potion_inventory : 2D array, monster_data : 2D array) -> selected_user_monster :
array, potion_inventory : 2D array
{ Spesifikasi : Mengatur potion yang dimiliki }
{ I.S. database }
{ F.S. database yang sudah diupdate }

```

ALGORITMA

```

list_user <- [[str(item) for item in row] for row in list_user]
list_monster <- [[str(item) for item in row] for row in list_monster]
list_item_inventory <- [[str(item) for item in row] for row in list_item_inventory]
list_monster_inventory <- [[str(item) for item in row] for row in list_monster_inventory]

headers <- list_user[0]
data <- []
for i from 1 to len(list_user) - 1 do
    data.append(list_user[i])
user_data <- [dict(custom_zip(headers, row)) for row in data]

user_login <- [u for u in user_data if u['id'] = str(login_id)]
if user_login then
    user_login <- user_login[0]
else
    output("User_id tidak terdaftar!")

list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row in
list_user]
list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for row in
list_monster]
list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in row] for
row in list_item_inventory]
list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in row]
for row in list_monster_inventory]
    return list_user, list_monster, list_item_inventory, list_monster_inventory

role <- str(user_login['role']).lower()
if role != 'agent' then
    output("Yah, hanya agent saja yang boleh masuk Arena.")
    list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row in
list_user]
    list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for row
in list_monster]
    list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in row]
for row in list_item_inventory]
    list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in
row] for row in list_monster_inventory]
    return list_user, list_monster, list_item_inventory, list_monster_inventory

headers <- list_monster[0]
data <- []
for i from 1 to len(list_monster) - 1 do
    data.append(list_monster[i])
monster_data <- [dict(custom_zip(headers, row)) for row in data]

headers <- list_item_inventory[0]
data <- []
for i from 1 to len(list_item_inventory) - 1 do
    data.append(list_item_inventory[i])
potion_inventory <- [dict(custom_zip(headers, row)) for row in data]

headers <- list_monster_inventory[0]
data <- []
for i from 1 to len(list_monster_inventory) - 1 do
    data.append(list_monster_inventory[i])
monster_inventory <- [dict(custom_zip(headers, row)) for row in data]

user_monsters <- []
for monster_inv in monster_inventory do

```

```

    for monster_data_entry in monster_data do
      if monster_inv['monster_id'] = monster_data_entry['id'] and monster_inv['user_id'] =
login_id then
        monster <- {**monster_data_entry, 'level': monster_inv['level']}
        monster <- adjust(monster)
        user_monsters.append(monster)

output("\nMonster Anda:")
for idx traversal len(user_monsters) - 1 do
  user_monster <- user_monsters[idx]
  output(f"{idx + 1}. Name: {user_monster['type']} | ATK Power:{user_monster['atk_power']}
| DEF Power: {user_monster['def_power']} | HP: {user_monster['hp']} | Level:
{user_monster['level']}")

user_choice <- int(input("\nPilih monster untuk dipertarungkan: ")) - 1
selected_user_monster <- user_monsters[user_choice]
output("\nAnda memilih:")
output("Nama: " + selected_user_monster['type'])
output("ATK Power: " + selected_user_monster['atk_power'])
output("DEF Power: " + selected_user_monster['def_power'])
output("HP: " + selected_user_monster['hp'])
output("Level: " + selected_user_monster['level'])
sleep(1)

oc_coin <- 0
stage <- 0
attacks <- 0
defends <- 0
user_quit <- False
ronde <- 0

while stage <= 4 and not user_quit do
  mns <- []
  for monster_inv in monster_inventory do
    for monster_data_entry in monster_data do
      if monster_inv['monster_id'] = monster_data_entry['id'] then
        monster <- {**monster_data_entry, 'level': monster_inv['level']} monster <-
adjust(monster)
        mns.append(monster)

opponent_idx <- random_num(0, len(mns) - 1)
opponent_monster <- mns[opponent_idx]
opponent_monster['level'] <- str(stage + 1)
opponent_monster <- adjust(opponent_monster)
output(f"----- STAGE {stage + 1} -----")
sleep(1)
output("\nMonster Lawan:")
output("Nama: " + opponent_monster['type'])
output("ATK Power: " + opponent_monster['atk_power'])
output("DEF Power: " + opponent_monster['def_power'])
output("HP: " + opponent_monster['hp'])
output("Level: " + opponent_monster['level'])
sleep(1)

is_strength_used <- False
is_healing_used <- False
is_resilience_used <- False

while int(opponent_monster['hp']) > 0 and int(selected_user_monster['hp']) > 0 do
  sleep(1)

```

```

output(f"\nRONDE {ronde + 1}")
sleep(1)
output("\nGiliran Anda menyerang monster lawan!")
sleep(1)
action <- input("Pilih aksi Anda - Attack, Potion, Quit: ")

if action.lower() = 'attack' then
  selected_user_monster, opponent_monster, damage <-
  attack_arena(selected_user_monster, opponent_monster)
  attacks <- attacks + damage
  output(f"{selected_user_monster['type']} menyerang monster
{opponent_monster['type']}. HYAH!")
  sleep(1)
  output(f"MONSTER LAWAN ({opponent_monster['type']}): ")
  output(f"ATK Power: {opponent_monster['atk_power']}")
  output(f"DEF Power: {opponent_monster['def_power']}")
  output(f"HP: {opponent_monster['hp']}")
  output(f"Level: {opponent_monster['level']}")
  sleep(1)
  if opponent_monster['hp'] <= 0 then
    break
  else if action.lower() = 'potion' then
    output("\nPotion Tersedia:")
    user_potions <- [potion for potion in potion_inventory if potion['user_id'] =
login_id]
    if not user_potions then
      output("Anda tidak mempunyai potion apapun")
      continue
    for idx from 0 to len(user_potions) - 1 do
      user_potion <- user_potions[idx]
      output(f"{idx + 1}. {user_potion['type']} - Quantity:
{user_potion['quantity']}")
      potion_choice <- int(input("\nPilih potion yang akan digunakan: ")) - 1
      selected_potion <- user_potions[potion_choice]['type']

      if selected_potion = 'strength' and not is_strength_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
selected_user_monster, login_id, potion_inventory, monster_data)
        is_strength_used <- True
      else if selected_potion = 'healing' and not is_healing_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
selected_user_monster, login_id, potion_inventory, monster_data)
        is_healing_used <- True
      else if selected_potion = 'resilience' and not is_resilience_used then
        selected_user_monster, potion_inventory <- potion(selected_potion,
selected_user_monster, login_id, potion_inventory, monster_data)
        is_resilience_used <- True
      else
        output("You cannot use that potion again in this battle.")
        output("\nMonster Anda:")
        output("Name: " + selected_user_monster['type'])
        output("ATK Power: " + selected_user_monster['atk_power'])
        output("DEF Power: " + selected_user_monster['def_power'])
        output("HP: " + selected_user_monster['hp'])
        output("Level: " + selected_user_monster['level'])
        sleep(1)
        continue

    else if action.lower() = 'quit' then
      user_quit <- True

```

```

        break
    else
        continue

    if not user_quit then
        output(f"Sekarang giliran monster {opponent_monster['type']} melawan
monstermu! HYAH!")
        opponent_monster, selected_user_monster, damage <-
attack_arena(opponent_monster, selected_user_monster)
        defends <- defends + damage
        sleep(1)
        output(f"MONSTER ANDA ({selected_user_monster['type']}): ")
        output(f"ATK Power: {selected_user_monster['atk_power']}")
        output(f"DEF Power: {selected_user_monster['def_power']}")
        output(f"HP: {selected_user_monster['hp']}")
        output(f"Level: {selected_user_monster['level']}")
        sleep(1)
        if selected_user_monster['hp'] <= 0 then
            break
        else
            ronde <- ronde + 1
    else
        break

    if not user_quit then
        if opponent_monster['hp'] <= 0 then
            oc_coin <- oc_coin + 10 * (stage + 1) ^ 2
            output(f"Selamat, Anda Lolos Stage {stage + 1} ! Anda memperoleh {oc_coin} OC
coins!")
            stage <- stage + 1
        else
            break

    if user_quit then
        output("Anda berhasil kabur dari pertarungan!")
    else if stage = 5 then
        output("Selamat, Anda berhasil melalui kelima stage arena ini.")
    else if 0 < stage < 5 then
        output(f"Selamat, Anda telah menempuh stage {stage}. Semangat!")
    else
        output("Yah, monster Anda kalah putaran pertama. Semangat, pasti bisa!")

    output("----- STATS -----")
    output(f"Jumlah stage lolos: {stage}")
    output(f"Total hadiah koin: {oc_coin} OC")
    output(f"Serangan diberikan: {attacks}")
    output(f"Serangan diterima: {defends}")

    for user in user_data do
        if user['id'] = login_id then
            user['oc'] <- str(int(user['oc']) + oc_coin)

    headers <- list(user_data[0].keys())
    list_user <- [headers] + [[d[key] for key in headers] for d in user_data]

    headers <- list(monster_data[0].keys())
    list_monster <- [headers] + [[d[key] for key in headers] for d in monster_data]

    headers <- list(potion_inventory[0].keys())
    list_item_inventory <- [headers] + [[d[key] for key in headers] for d in potion_inventory]

```

```

headers <- list(monster_inventory[0].keys())
list_monster_inventory <- [headers] + [[d[key] for key in headers] for d in
monster_inventory]

list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row in
list_user]
list_monster <- [[int(item) if custom_isdigit(item) else item for item in row] for row in
list_monster]
list_item_inventory <- [[int(item) if custom_isdigit(item) else item for item in row] for
row in list_item_inventory]
list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in row]
for row in list_monster_inventory]

return list_user, list_monster, list_item_inventory, list_monster_inventory

```

F10

PROGRAM SHOP & CURRENCY

{ Spesifikasi : Mengelola proses belanja di toko dan menampilkan currency }

KAMUS

procedure shop (login_id, list_user, list_monster, list_item_inventory,
list_monster_inventory, list_item_shop, list_monster_shop)
{ Spesifikasi: Mengelola proses belanja di toko, termasuk melihat dan membeli item oleh
user yang berperan sebagai agent. }
{ I.S.: Array dari list_user, list_monster, list_item_inventory, list_monster_inventory,
list_item_shop, list_monster_shop. }
{ F.S.: Data diperbarui sesuai transaksi di toko. }
{ Deklarasi variabel }

- login_id: integer
- list_user: array of user
- list_monster: array of monster
- list_item_inventory: array of item
- list_monster_inventory: array of monster
- list_item_shop: array of item
- list_monster_shop: array of monster
- headers: array of string
- data: array of array of string
- user_data, monster_data, potion_inventory, monster_inventory, potion_shop,
- monster_shop: array of dictionary
- user_login: dictionary
- user_id, role, action, item_type, idx, selected_potion_idx, quantity: string
- selected_potion, user_potion, user_coins, selected_monster, existing_monster,
- monster_info: dictionary
- monster_shop_details: array of string

ALGORITMA

```

list <- login_id, list_user, list_monster, list_item_inventory,
list_monster_inventory, list_item_shop, list_monster_shop

headers <- list_user[0]
data <- list_user[1:]
user_data <- [dict(custom_zip(headers, row)) for row in data]
user_login <- [u for u in user_data if u['id'] = str(login_id)]

if not user_login then
    output ("User_id tidak terdaftar!")

```



```

        return list_user, list_monster, list_item_inventory, list_monster_inventory,
        list_item_shop, list_monster_shop

user_login ← user_login[0]
role ← user_login['role'].lower()

if role ≠ 'agent' then
    output ("Yah, hanya agent saja yang boleh masuk Shop and Currency.")
    return list_user, list_monster, list_item_inventory, list_monster_inventory,
    list_item_shop, list_monster_shop

headers ← list_monster[0]
data ← list_monster[1:]
monster_data ← [dict(custom_zip(headers, row)) for row in data]

headers ← list_item_inventory[0]
data ← list_item_inventory[1:]
potion_inventory ← [dict(custom_zip(headers, row)) for row in data]

headers ← list_monster_inventory[0]
data ← list_monster_inventory[1:]
monster_inventory ← [dict(custom_zip(headers, row)) for row in data]

headers ← list_item_shop[0]
data ← list_item_shop[1:]
potion_shop ← [dict(custom_zip(headers, row)) for row in data]

headers ← list_monster_shop[0]
data ← list_monster_shop[1:]
monster_shop ← [dict(custom_zip(headers, row)) for row in data]

function display_shop_items(items)
    for idx, item IN ENUMERATE(items, start=1) do
        output ("{idx}. {item}")

{ Iterasi Shop and Currency oleh User }
while True do
    output (">>> Selamat datang di Toko Mr. Yanto! Pilih aksi (beli/lihat/keluar)
    :"), input (action)
    action ← action.lower()

    if action = "lihat" then
        WHILE True DO
            output (">>> Mau lihat apa? (potion/monster):"), input (item_type)
            item_type ← item_type.lower()

            if item_type = "potion" then
                output ("Potion yang tersedia:")
                display_shop_items([f"{potion['type']}" (Stok: {potion['stock']},
                Harga: {potion['price']} koin)" for potion in potion_shop])
                break

            if item_type = "monster" then
                output ("Monster yang tersedia:")
                monster_shop_details ← []
                for monster IN monster_shop do
                    monster_info ← [m for m in monster_data if m['id'] ==
                    monster['monster_id']][0]
                    monster_shop_details.APPEND(f"{monster['monster_id']}.
                    {monster_info['type']}" (ATK:

```

```

                                {monster_info['atk_power']], DEF:
                                {monster_info['def_power']], HP: {monster_info['hp']],
                                Stok: {monster['stock']], Harga: {monster['price']]
                                koin)")
    display_shop_items(monster_shop_details)
    break
else
    continue

if action = "beli" then
    user_id ← str(login_id)
    output (f"Username Anda: {user_id}")
    while True do
        output (">>> Mau beli apa? (potion/monster):"), input (item_type)
        item_type ← item_type.lower()

        if item_type = "potion" then
            while True do
                for user IN user_data do
                    IF user['id'] = user_id THEN
                        user_coins ← int(user['oc'])

                        output (f"Jumlah koin Anda: {user['oc']}")
                        output ("Potion yang dijual:")
                        output (">>> Pilih nomor urut potion yang ingin dibeli:"),
                        input (idx)

                        if NOT custom_isdigit(idx) then
                            output ("Pilihan tidak valid. Silakan masukkan format
                                yang benar.")
                            continue

                        selected_potion_idx ← int(idx) - 1

                        if selected_potion_idx > len(potion_shop) - 1 then
                            output ("Pilihan tidak valid, silakan masukkan nomor
                                pilihan yang tersedia")
                            continue

                        input ("Masukkan banyaknya potion yang ingin dibeli: ")

                        if NOT custom_isdigit(quantity) then
                            output ("Pilihan tidak valid. Silakan masukkan format
                                yang benar.")
                            continue

                        quantity ← int(quantity)
                        selected_potion ← potion_shop[selected_potion_idx]

                        if int(selected_potion['stock']) > 0 and
                        int(selected_potion['stock']) - quantity >= 0 then
                            for user IN user_data do
                                if user['id'] = user_id then
                                    user_coins ← int(user['oc'])
                                    if user_coins >= int(selected_potion['price'])
                                    * quantity THEN
                                        user_coins -= int(selected_potion['price'])

```

```

        * quantity
        selected_potion_name ← selected_potion['type']
        user_potion ← [p for p in potion_inventory if
            p['user_id'] ← user_id AND p['type'] =
            selected_potion_name]

        if user_potion then
            user_potion[0]['quantity'] ←
                str(int(user_potion[0]['quantity']) +
                    quantity)
        else
            potion_inventory.APPEND({'user_id':
                user_id, 'type': selected_potion_name,
                'quantity': f'{quantity}'})

            selected_potion['stock'] ←
                str(int(selected_potion['stock']) -
                    quantity)
            user['oc'] ← str(user_coins)
            output (f"Berhasil membeli
                {selected_potion_name}.")
        else
            output ("Koin Anda tidak mencukupi.")
            break
        break
    else
        output ("Stok potion habis atau tidak mencukupi.")
        break
    break

if item_type = "monster" then
    for user IN user_data do
        if user['id'] = user_id then
            user_coins ← int(user['oc'])

            output (f"Jumlah koin Anda: {user['oc']}")
            output ("Monster yang dijual:")
            monster_shop_details ← []

            for monster in monster_shop do
                monster_info ← [m for m in monster_data if m['id'] =
                    monster['monster_id']][0]
                monster_shop_details.APPEND(f"{monster['monster_id']}.
                    {monster_info['type']} (ATK: {monster_info['atk_power']},
                    DEF: {monster_info['def_power']}, HP: {monster_info['hp']},
                    Stok: {monster['stock']}, Harga: {monster['price']} koin)")
            display_shop_items(monster_shop_details)

            while True do
                output (">>> Pilih nomor urut monster yang ingin dibeli:")
                , input (idx)

                if NOT custom_isdigit(idx) then
                    output ("Pilihan tidak valid. Silakan masukkan format
                        yang benar.")
                    continue

                selected_monster_idx ← int(idx) - 1

```

```

IF selected_monster_idx > len(monster_shop) - 1 THEN
    output ("Pilihan tidak valid, silakan masukkan nomor
    pilihan yang tersedia")
    continue

selected_monster ← monster_shop[selected_monster_idx]
existing_monster ← [m for m in monster_inventory if
m['user_id'] = user_id AND m['monster_id'] =
selected_monster['monster_id']]

if int(selected_monster['stock']) > 0 then
    if existing_monster then
        output ("Anda sudah memiliki monster ini.")
    else
        for user IN user_data DO
            if user['id'] = user_id then
                user_coins ← int(user['oc'])

```

F11

PROGRAM LABORATORY

{ Spesifikasi : melakukan upgrade monster yang dimiliki di inventory. }

KAMUS

{ Deklarasi Variabel }

login_id : integer

list_user, list_monster, list_monster_inventory : array of array of string

procedure laboratory (login_id, list_user, list_monster, list_monster_inventory)

{ SPEK : melakukan upgrade monster yang dimiliki di inventory. }

{ I.S. list_monster, list_user, list_monster_inventory }

{ F.S. Mencetak hasil upgrade level monster }

ALGORITMA

{ Konversi Elemen ke String }

list_user <- [[str(item) for item in row] for row in list_user]

list_monster <- [[str(item) for item in row] for row in list_monster]

list_monster_inventory <- [[str(item) for item in row] for row in
list_monster_inventory]

{ Persiapan Data User }

headers <- list_user[0]

data <- []

i traversal [1...len(list_user) - 1]

data.append(list_user[i])

user_data <- [dict(custom_zip(headers, row)) for row in data]

{ Validasi User Login }

user_login <- [u for u in user_data if u['id'] == str(login_id)]

if user_login tidak kosong then

user_login <- user_login[0]

else

output ("User_id tidak terdaftar!")

list_user <- [[int(item) if custom_isdigit(item) else item for item in row]
for row in list_user]

list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for
item in row] for row in list_monster_inventory]

return list_user, list_monster_inventory

```

role <- str(user_login['role']).lower()
if role != 'agent' then
    output ("Yah, hanya agent saja yang boleh masuk Laboratory.")
    list_user <- [[int(item) if custom_isdigit(item) else item for item in row]
    for row in list_user]
    list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for
    item in row] for row in list_monster_inventory]
    return list_user, list_monster_inventory

{ Persiapan Data Monster dan Inventori }
headers <- list_monster[0]
data <- []
i traversal [1...len(list_monster) - 1]
    data.append(list_monster[i])
monster_data <- [dict(custom_zip(headers, row)) for row in data]

headers <- list_monster_inventory[0]
data <- []
i traversal [1...len(list_monster_inventory) - 1]
    data.append(list_monster_inventory[i])
monster_inventory <- [dict(custom_zip(headers, row)) for row in data]

{ Menampilkan Daftar Monster }
while True do
    output ("===== MONSTER LIST =====")
    index <- 1
    user_monsters <- [monster for monster in monster_inventory if
    monster['user_id'] == login_id]
    for monster in user_monsters:
        monster_id <- monster['monster_id']
        level <- monster['level']
        monster_name <- ""
        for monster in monster_data:
            if monster['id'] == monster_id then
                monster_name <- monster['type']
                break
        output (f"{index}. {monster_name} (Level: {level})")
        index <- index + 1

{ Menampilkan Daftar Harga Upgrade }
output ("\n===== UPGRADE PRICE =====")
output ("1. Level 1 -> Level 2: 300 OC")
output ("2. Level 2 -> Level 3: 500 OC")
output ("3. Level 3 -> Level 4: 800 OC")
output ("4. Level 4 -> Level 5: 1000 OC")

{ Meminta Input Pilihan Monster }
monster_choice <- input(">>> Pilih monster: ")
if not custom_isdigit(monster_choice) then
    output ("Format tidak valid")
    continue
monster_choice <- int(monster_choice)

if monster_choice < 1 or monster_choice > len(user_monsters) then
    output ("Pilihan monster tidak valid. Silakan pilih lagi.\n")
    continue

{ Mencari Data Monster yang Dipilih User }
selected_monster_data <- user_monsters[monster_choice - 1]
monster_id <- selected_monster_data['monster_id']
current_level <- int(selected_monster_data['level'])

```

```

{ Menghitung Biaya Upgrade }
  if current_level == 1 then
    upgrade_cost <- 300
  if current_level == 2 then
    upgrade_cost <- 500
  if current_level == 3 then
    upgrade_cost <- 800
  if current_level == 4 then
    upgrade_cost <- 1000
  if current_level >= 5 then
    output ("Maaf, monster yang Anda pilih sudah memiliki level maksimum.\n")
    break

{ Memeriksa Kembali Data Monster }
  for monster in monster_data:
    if monster['id'] == monster_id then
      monster_name <- monster['type']
      break
    output (f"\n{monster_name} akan di-upgrade ke level {current_level + 1}.")
    output (f"Harga untuk melakukan upgrade adalah {upgrade_cost} OC.")
    confirm <- input(">>> Lanjutkan upgrade (Y/N): ").upper()

{ Memeriksa Hasil Konfirmasi }
  if confirm == 'Y':
    for user in user_data:
      if user['id'] == login_id then
        user_coins <- int(user['oc'])
        if user_coins >= upgrade_cost then
          user_coins <- str(user_coins - upgrade_cost)
          selected_monster_data['level'] <- str(current_level + 1)
          user['oc'] <- user_coins
          output (f"Selamat, {monster_name} berhasil di-upgrade ke level {selected_monster_data['level']}!")
          output (f"Besar OC setelah upgrade: {user_coins}\n")
        else:
          output ("Maaf, OC Anda tidak mencukupi untuk melakukan upgrade.\n")
          break
    break
  if confirm == 'N':
    output ("Upgrade dibatalkan.\n")
    break
  else
    output ("Input tidak valid. Upgrade dibatalkan.\n")
    break

{ Mengembalikan Data ke Bentuk Awal }
  headers <- list(user_data[0].keys())
  list_user <- [headers] + [[d[key] for key in headers] for d in user_data]

  headers <- list(monster_inventory[0].keys())
  list_monster_inventory <- [headers] + [[d[key] for key in headers] for d in monster_inventory]

  list_user <- [[int(item) if custom_isdigit(item) else item for item in row] for row in list_user]
  list_monster_inventory <- [[int(item) if custom_isdigit(item) else item for item in row] for row in list_monster_inventory]

{ Return Data }
  return list_user, list_monster_inventory

```

PROGRAM SHOP MANAGEMENT

{Spesifikasi : Melakukan operasi manajemen shop sesuai input pengguna}

{I.S : Data shop sebelum update}

{F.S : Data shop sesudah update}

KAMUS

exit : boolean

action, choice : string

function load_tli (li_monster, li_item, li_monster_shop, li_item_shop : 2D array) -> tli_monster, tli_item : 2D array)

{ Spesifikasi : Mengembalikan list yang sesuai dengan spesifikasi tabel }

{ I.S. list data file.csv }

{ F.S. list data sesuai table }

function load_tlinot (li_monster, li_item, li_monster_shop, li_item_shop : 2D array) -> tlinot_monster, tlinot_item : 2D array)

{ Spesifikasi : Mengembalikan list yang tidak ada di shop sesuai dengan spesifikasi tabel }

{ I.S. list data file.csv }

{ F.S. list data sesuai table }

procedure tableprint (tli : 2D array)

{ Spesifikasi : Mengeluarkan table sesuai dengan list }

{ I.S. list table }

{ F.S. table }

function tambahmonster (li_monster_shop, tlinot_monster : 2D array) -> li_monster_shop : 2D array

{ Spesifikasi : Menjalankan opsi tambah monster }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function tambahitem (li_item, li_item_shop, tlinot_item : 2D array) -> li_item_shop : 2D array

{ Spesifikasi : Menjalankan opsi tambah item }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function ubahmonster (li_monster_shop, tli_monster : 2D array) -> li_monster_shop : 2D array

{ Spesifikasi : Menjalankan opsi ubah monster }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function ubahitem (li_item, li_item_shop, tli_item : 2D array) -> li_item_shop : 2D array

{ Spesifikasi : Menjalankan opsi ubah item }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function hapusmonster (li_monster_shop, tli_monster : 2D array) -> li_monster_shop : 2D array

{ Spesifikasi : Menjalankan opsi hapus monster }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function hapusitem (li_item, li_item_shop, tli_item : 2D array) -> li_item_shop : 2D array

{ Spesifikasi : Menjalankan opsi hapus item }

{ I.S. list database file.csv }

{ F.S. updated list database file.csv }

function isActionValid (input : string) -> boolean

{ Spesifikasi : Mengembalikan nilai boolean hasil input action }

{ I.S. input action }

{ F.S. boolean }

function isMonsterOrPotion (input : string) -> boolean

```
{ Spesifikasi : Mengembalikan nilai boolean hasil input choice }
{ I.S. input choice }
{ F.S. boolean }
```

ALGORITMA

```
output("\nIrasshaimase! Selamat datang kembali, Mr. Monogram!")
exit <- False
while not exit do
  action <- input("\nPilih aksi (lihat/tambah/ubah/hapus/keluar): ")
  while not isActionValid(action) do
    action <- input("Input tidak valid. Ulangi\nPilih aksi
(lihat/tambah/ubah/hapus/keluar): ")

  tli_monster, tli_item <- load_tli(li_monster, li_item, li_monster_shop, li_item_shop)
  tlinot_monster, tlinot_item <- load_tlinot(li_monster, li_item, li_monster_shop,
li_item_shop)

  if action.lower() = "lihat" then
    choice <- input("Mau lihat apa? (monster/potion): ")
    while not isMonsterOrPotion(choice) do
      choice <- input("Input tidak valid. Ulangi\nMau lihat apa? (monster/potion): ")
    if choice.lower() = "monster" then
      tableprint(tli_monster)
    else
      tableprint(tli_item)
  else if action.lower() = "tambah" then
    choice <- input("Mau nambahin apa? (monster/potion): ")
    while not isMonsterOrPotion(choice) do
      choice <- input("Input tidak valid. Ulangi\nMau nambahin apa? (monster/potion):
")

    if choice.lower() = "monster" then
      if len(tlinot_monster) = 1 then
        output("Tidak ada monster yang tersedia")
      else
        tableprint(tlinot_monster)
        li_monster_shop <- tambahmonster(li_monster_shop, tlinot_monster)
    else
      if len(tlinot_item) = 1 then
        output("Tidak ada potion yang tersedia")
      else
        tableprint(tlinot_item)
        li_item_shop <- tambahitem(li_item, li_item_shop, tlinot_item)
  else if action.lower() = "ubah" then
    choice <- input("Mau ubah apa? (monster/potion): ")
    while not isMonsterOrPotion(choice) do
      choice <- input("Input tidak valid. Ulangi\nMau ubah apa? (monster/potion): ")
    if choice.lower() = "monster" then
      if len(tli_monster) = 1 then
        output("Tidak ada monster yang tersedia")
      else
        tableprint(tli_monster)
        li_monster_shop <- ubahmonster(li_monster_shop, tli_monster)
    else
      if len(tli_item) = 1 then
        output("Tidak ada item yang tersedia")
      else
        tableprint(tli_item)
        li_item_shop <- ubahitem(li_item, li_item_shop, tli_item)
  else if action.lower() = "hapus" then
```



```

choice <- input("Mau hapus apa? (monster/potion): ")
while not isMonsterOrPotion(choice) do
  choice <- input("Input tidak valid. Ulangi\nMau hapus apa? (monster/potion): ")
if choice.lower() = "monster" then
  if len(tli_monster) = 1 then
    output("Tidak ada monster yang tersedia")
  else
    tableprint(tli_monster)
    li_monster_shop <- hapusmonster(li_monster_shop, tli_monster)
  else
    if len(tli_item) = 1 then
      output("Tidak ada item yang tersedia")
    else
      tableprint(tli_item)
      li_item_shop <- hapusitem(li_item, li_item_shop, tli_item)
    else
      output("Dadah Mr. Yanto, sampai jumpa lagi!\n")
      exit <- True
return li_monster, li_monster_shop, li_item_shop

```

F13 (on process)

PROGRAM MONSTER MANAGEMENT

{ Spesifikasi : Mengatur monster dalam database. }
{ Asumsi : semua array, kecuali diatur, telah memiliki banyak elemen yang benar, fungsi bawaan Python sudah ada di notal }

KAMUS

{ Deklarasi Variabel }
list_monster: array of array of string

{ Deklarasi Prosedur }
procedure monster_management
{ SPEK : Mengatur monster dalam database }
{ I.S. list of element }
{ F.S. list of string }

ALGORITMA

```
{ Konversi Elemen ke String }
list_monster <- [[str(item) for item in row] for row in list_monster]

{ Persiapan Data Monster }
headers <- list_monster[0]
data <- []
i traversal [1...len(list_monster) - 1]
  data.append(list_monster[i])
monster_data <- [dict(custom_zip(headers, row)) for row in data]

{ Menu Utama }
while True:
  output ("\nMonster Management Menu:")
  output ("1. Tampilkan semua monster")
  output ("2. Tambahkan monster baru")
  output ("3. Keluar\n")
  choice <- input ("Masukkan pilihan (1, 2, or 3): ")

{ Pilihan 1: Tampilkan Semua Monster }
  if choice == "1" then
    output ("Semua Monster:")
    for monster in monster_data:
      output (f"{monster['id']}. {monster['type']} (ATK:
        {monster['atk_power']}, DEF: {monster['def_power']}, HP:
        {monster['hp']})")
    continue

{ Pilihan 2: Tambahkan Monster Baru }
  if choice == "2" then
    new_monster_name <- input ("Masukkan nama monster baru: ")

{ Validasi Nama Monster Baru }
    monster_full <- False
    for monster in monster_data:
      if monster['type'] == new_monster_name then
        output ("Monster ini sudah tersedia di menu. Silakan masukkan nama
          yang berbeda.\n")
        monster_full <- True
    if monster_full then
      Continue

{ Mengumpulkan Detail Monster Baru }
    new_monster_id <- 1
```

```

    for monster in monster_data:
        if monster['id'] == str(new_monster_id) then
            new_monster_id <- new_monster + 1

    new_hp <- input ("Masukkan hit points (integer positif): ")
    new_attack_power <- input ("Masukkan attack power (integer positif): ")
    new_defence_power <- input ("Masukkan defence power (integer 1-50): ")

{ Validasi Input }

    if custom_isdigit(new_attack_power) and custom_isdigit(new_defence_power) and
    custom_isdigit(new_hp) then
        new_hp <- int(new_hp)
        new_attack_power <- int(new_attack_power)
        new_defence_power <- int(new_defence_power)
        if not (0 < new_defence_power <= 50) then
            output ("Defence power harus di dalam rentang 1-50")
            continue
        if new_hp <= 0 dan new_attack_power <= 0 dan new_defence_power <= 0
        then
            output ("Masukan harus bilangan bulat positif!")
            continue
    else
        output ("Silakan masukkan format yang benar!")
        Continue

{ Menambahkan Monster Baru }
    new_monster <- {
        'id': str(new_monster_id),
        'type': str(new_monster_name),
        'atk_power': str(new_attack_power),
        'def_power': str(new_defence_power),
        'hp': str(new_hp)
    }
    monster_data.append(new_monster)
    output ("Monster baru berhasil ditambahkan.")
    Continue

{ Pilihan 3: Keluar }
    if choice == "3" then
        output ("Menutup monster management...")
        headers <- list(monster_data[0].keys())
        list_monster <- [headers] + [[d[key] for key in headers] for d in
        monster_data]
        list_monster <- [[int(item) if custom_isdigit(item) else item for item in
        row] for row in list_monster]
        return list_monster

{ Pilihan Tidak Valid }
    else
        output ("Silakan masukkan pilihan yang tersedia")
        continue

```

PROGRAM LOAD

```
{Spesifikasi : melakukan load}
{I.S : program belum terload file csvnya}
{F.S : file csv terload}
```

KAMUS

```
path      : string
li_item   : 2D Array
```

```
function csvtolist (csv_file : string, index_columnint : Array of Integer)
{ Spesifikasi : Mengembalikan 2D array berisi data dari file.csv}
{ I.S. file.csv }
{ F.S. 2D array }
```

ALGORITMA

```
import argparse
import sys
import os

parser <- argparse.ArgumentParser()
parser.add_argument('folder')

if len(sys.argv) != 2 then
    output("Tidak ada nama folder yang diberikan!\nUsage : python/py main.py
<nama_folder>")
    sys.exit(1)
else
    args <- parser.parse_args()
    path <- 'data/' + args.folder
    if not os.path.exists(path) then
        output(f"\nFolder {args.folder} tidak ditemukan.")
        sys.exit(1)
    else
        output("Loading...\n")
        li_user <- csvtolist(path + '/user.csv', [0, 4])
        li_monster <- csvtolist(path + '/monster.csv', [0, 2, 3, 4])
        li_item_inventory <- csvtolist(path + '/item_inventory.csv', [0, 2])
        li_monster_inventory <- csvtolist(path + '/monster_inventory.csv', [0, 1, 2])
        li_item_shop <- csvtolist(path + '/item_shop.csv', [1, 2])
        li_monster_shop <- csvtolist(path + '/monster_shop.csv', [0, 1, 2])
        li_item <- [['potion_id', 'type'], [1, 'strength'], [2, 'resilience'], [3, 'healing']]
        output("Selamat datang di program OWCA!")

    return li_user, li_monster, li_item, li_item_inventory, li_monster_inventory, li_item_shop,
    li_monster_shop
```

F15

PROGRAM SAVE

```
{Spesifikasi : melakukan penyimpanan data ke dalam folder}
{I.S : program belum menyimpan data ke dalam folder}
{F.S : data tersimpan dalam folder yang ditentukan}
```

KAMUS

```
parentsave, path : string
procedure listtocsv (li_name : 2D Array, csv_file : string)
{ Spesifikasi : Menulis data dari 2D Array ke file.csv }
{ I.S. 2D array }
```

```
{ F.S. file.csv }
```

ALGORITMA

```
import os
```

```
parentsave <- 'data'  
folder_name <- input("Masukkan nama folder: ")  
path <- parentsave + '/' + str(folder_name)  
output("\\nSaving...\\n")
```

```
if not os.path.exists(parentsave) then  
  os.makedirs(parentsave)  
  output(f"Membuat folder {parentsave}...")
```

```
if not os.path.exists(path) then  
  os.makedirs(path)  
  output(f"Membuat folder {path}...")
```

```
listtocsv(li_user, path + '/user.csv')  
listtocsv(li_monster, path + '/monster.csv')  
listtocsv(li_item_inventory, path + '/item_inventory.csv')  
listtocsv(li_monster_inventory, path + '/monster_inventory.csv')  
listtocsv(li_item_shop, path + '/item_shop.csv')  
listtocsv(li_monster_shop, path + '/monster_shop.csv')  
output(f"Berhasil menyimpan data di folder {path}")
```

F16

PROGRAM EXIT

```
{ Spesifikasi : Exit program secara umum dengan pilihan save }  
{ import fungsi dari f15 as f15 }
```

KAMUS

```
procedure exit_game (list_user, list_monster, list_item_inventory, list_monster_inventory,  
list_item_shop, list_monster_shop)  
{ Spesifikasi: Mengelola proses keluar dari game, termasuk opsi untuk menyimpan data  
terlebih dahulu. }  
{ I.S.: Array dari list_user, list_monster, list_item_inventory, list_monster_inventory,  
list_item_shop, list_monster_shop. }  
{ F.S.: Program keluar dengan atau tanpa menyimpan data. }
```

```
{ Deklarasi Variabel }
```

```
  list_user : array of user  
  list_monster : array of monster  
  list_item_inventory : array of item  
  list_monster_inventory : array of monster  
  list_item_shop : array of item  
  list_monster_shop : array of monster  
  user_input : string
```

ALGORITMA

```
  while True do  
    output("Apakah Anda mau melakukan penyimpanan file yang sudah diubah?  
(y/n)"), input(user_input)  
    user_input <- to_lowercase(user_input)  
    if user_input = 'y' then  
      {Menjalankan prosedur save}
```

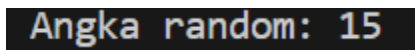
```
        save(list_user,          list_monster,          list_item_inventory,  
            list_monster_inventory, list_item_shop, list_monster_shop)  
        exit  
if user_input = 'n' then  
    {Keluar program}  
    exit  
else  
    continue
```

BAB 6

TANGKAPAN LAYAR PENGUJIAN PROGRAM

F00 demo?

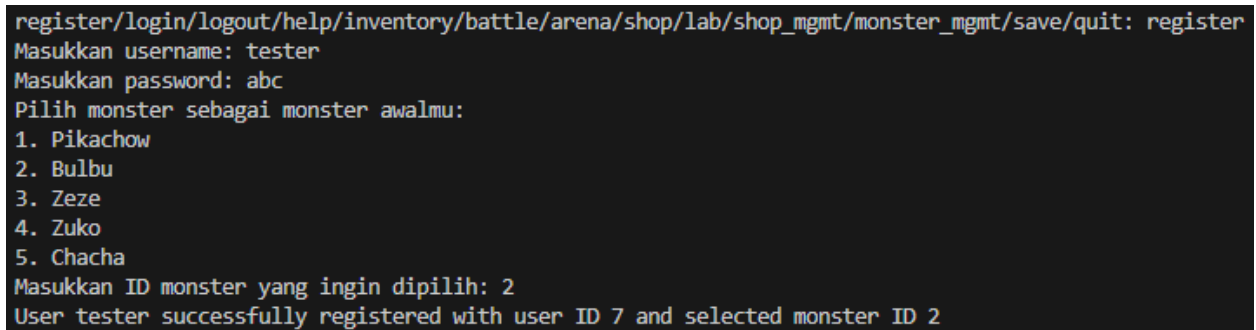
6.0. Gambar Pengetesan *Random Number Generator*

A screenshot of a terminal window with a black background and white text. The text reads "Angka random: 15".

Angka random: 15

F01 Register

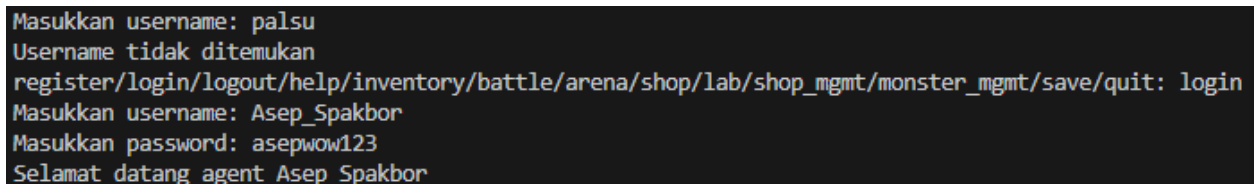
6.1. Gambar Pengetesan *Register*

A screenshot of a terminal window with a black background and white text. The text shows a command prompt menu, followed by registration details for a user named 'tester' with password 'abc', a list of monster choices (Pikachow, Bulbu, Zeze, Zuko, Chacha), and a confirmation message for selecting monster ID 2.

```
register/login/logout/help/inventory/battle/arena/shop/lab/shop_mgmt/monster_mgmt/save/quit: register
Masukkan username: tester
Masukkan password: abc
Pilih monster sebagai monster awalmu:
1. Pikachow
2. Bulbu
3. Zeze
4. Zuko
5. Chacha
Masukkan ID monster yang ingin dipilih: 2
User tester successfully registered with user ID 7 and selected monster ID 2
```

F02 Login

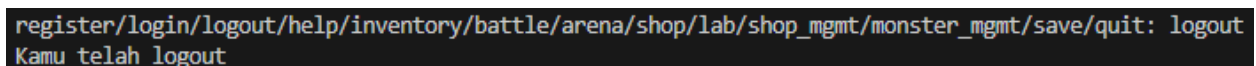
6.2. Gambar Pengetesan *Login*

A screenshot of a terminal window with a black background and white text. It shows a failed login attempt with a fake username, followed by a successful login for 'Asep Spakbor' with password 'asepow123', and a welcome message.

```
Masukkan username: palsu
Username tidak ditemukan
register/login/logout/help/inventory/battle/arena/shop/lab/shop_mgmt/monster_mgmt/save/quit: login
Masukkan username: Asep_Spakbor
Masukkan password: asepow123
Selamat datang agent Asep Spakbor
```

F03 Logout

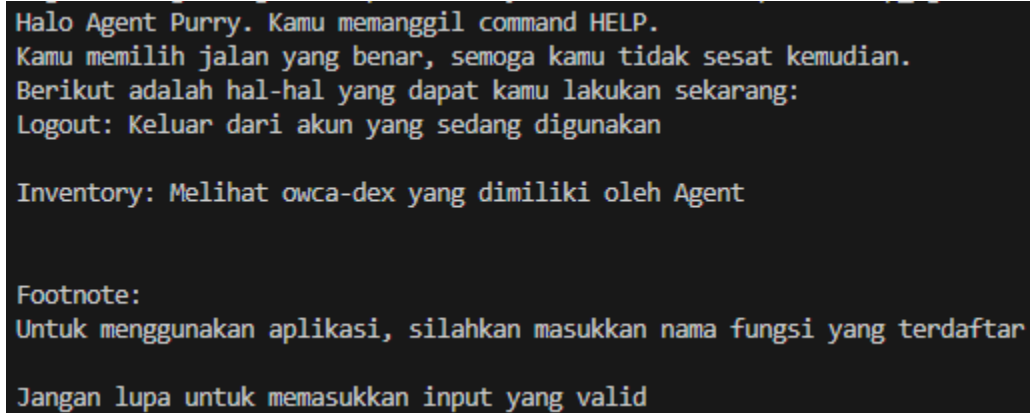
6.3. Gambar Pengetesan *Logout*

A screenshot of a terminal window with a black background and white text. It shows the 'logout' command being entered and the response 'Kamu telah logout'.

```
register/login/logout/help/inventory/battle/arena/shop/lab/shop_mgmt/monster_mgmt/save/quit: logout
Kamu telah logout
```

F04 Help

6.4. Gambar Pengetesan *Help*



```
Halo Agent Purry. Kamu memanggil command HELP.  
Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian.  
Berikut adalah hal-hal yang dapat kamu lakukan sekarang:  
Logout: Keluar dari akun yang sedang digunakan  
  
Inventory: Melihat owca-dex yang dimiliki oleh Agent  
  
Footnote:  
Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar  
Jangan lupa untuk memasukkan input yang valid
```

F05/F06 demo? (demonya di dalam battle sama arena (F08 dan F09), karena fungsi F05 (attack dan adjust) dan F06 (potion) itu cuman bisa dipake di bagian (F08 dan F09) itu)

F07 Inventory

6.7. Gambar Pengetesan *Inventory*

```
Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
potion
Potion yang tersedia:
1. Nama: strength, Jumlah: 5
2. Nama: resilience, Jumlah: 3
Pilih nomor urut potion untuk informasi lebih lanjut:
2
Nama: resilience, Jumlah: 3
Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
monster
Monster yang tersedia:
1. 1. Pikachow (HP: 600)
Pilih nomor urut monster untuk informasi lebih lanjut:
1
1. Pikachow (ATK: 125, DEF: 10, HP: 600)
Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
potion
Potion yang tersedia:
1. Nama: strength, Jumlah: 5
2. Nama: resilience, Jumlah: 3
Pilih nomor urut potion untuk informasi lebih lanjut:
3
Pilihan tidak valid, silakan masukkan nomor yang tersedia
2
Nama: resilience, Jumlah: 3
Halo, user_id 2, jumlah OWCA Coin Anda 9999
>>> Lihat Barang Apa? (potion/monster/quit):
quit
```

F08 Battle

6.8.1. Gambar Pengetesan *Battle Round 1*

```
Monster Lawan:
Nama: Pikachu
ATK Power: 125
DEF Power: 10
HP: 600
Level: 1

Monster Anda:
1. Name: Pikachu | ATK Power: 125 | DEF Power: 10 | HP: 600 | Level: 1

Pilih monster untuk dipertarungkan: 1

Anda memilih:
Nama: Pikachu
ATK Power: 125
DEF Power: 10
HP: 600
Level: 1

RONDE 1

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: attack
Pikachu menyerang monster Pikachu. HYAH!
MONSTER LAWAN (Pikachu):
ATK Power: 125
DEF Power: 10
HP: 468
Level: 1
Sekarang giliran monster Pikachu melawan monstermu! HYAH!
MONSTER ANDA (Pikachu):
atk Power: 125
def Power: 10
HP: 476
Level: 1

RONDE 2

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: potion

Potion Tersedia:
1. strength - Quantity: 5
2. resilience - Quantity: 3

Pilih potion yang akan digunakan: 1

Monster Anda:
Name: Pikachu
ATK Power: 131
DEF Power: 10
HP: 476
Level: 1

RONDE 2
```

6.8.2. Gambar Pengetesan *Battle Round 2*

```
RONDE 2  
  
Giliran Anda menyerang monster lawan!  
Pilih aksi Anda - Attack, Potion, Quit: quit  
Anda berhasil kabur dari pertarungan!
```

F09 Arena

6.9.1. Gambar Pengetesan *Arena*

```
Monster Anda:
1. Name: Pikachu | ATK Power: 125 | DEF Power: 10 | HP: 600 | Level: 1

Pilih monster untuk dipertarungkan: 1

Anda memilih:
Nama: Pikachu
ATK Power: 125
DEF Power: 10
HP: 600
Level: 1
----- STAGE 1 -----

Monster Lawan:
Nama: Pikachu
ATK Power: 125
DEF Power: 10
HP: 600
Level: 1

RONDE 1

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: attack
Pikachow menyerang monster Pikachu. HYAH!
MONSTER LAWAN (Pikachow):
ATK Power: 125
DEF Power: 10
HP: 462
Level: 1
Sekarang giliran monster Pikachu melawan monsternya! HYAH!
MONSTER ANDA (Pikachow):
atk Power: 125
def Power: 10
HP: 469
Level: 1

RONDE 2

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: potion

Potion Tersedia:
1. strength - Quantity: 5
2. resilience - Quantity: 3

Pilih potion yang akan digunakan: 2

Monster Anda:
Name: Pikachu
ATK Power: 125
DEF Power: 10
HP: 469
Level: 1

RONDE 2
```

6.9.2. Gambar Pengetesan *Arena*

```
RONDE 2

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: attack
Pikachow menyerang monster Pikachow. HYAH!
MONSTER LAWAN (Pikachow):
ATK Power: 125
DEF Power: 10
HP: 352
Level: 1
Sekarang giliran monster Pikachow melawan monstermu! HYAH!
MONSTER ANDA (Pikachow):
atk Power: 125
def Power: 10
HP: 367
Level: 1

RONDE 3

Giliran Anda menyerang monster lawan!
Pilih aksi Anda - Attack, Potion, Quit: quit
Anda berhasil kabur dari pertarungan!
----- STATS -----
Jumlah stage lolos: 0
Total hadiah koin: 0 OC
Serangan diberikan: 248
Serangan diterima: 233
```

F10 Shop

6.10. Gambar Pengetesan *Shop*

```
>>> Mau beli apa? (potion/monster):
monster
Jumlah koin Anda: 6699
Monster yang dijual:
1. 1. Pikachu (ATK: 125, DEF: 10, HP: 600, Stok: 10, Harga: 500 koin)
2. 2. Bulbu (ATK: 50, DEF: 50, HP: 1200, Stok: 4, Harga: 700 koin)
3. 3. Zeze (ATK: 300, DEF: 10, HP: 100, Stok: 3, Harga: 1000 koin)
4. 4. Zuko (ATK: 100, DEF: 25, HP: 800, Stok: 8, Harga: 550 koin)
5. 5. Chacha (ATK: 80, DEF: 30, HP: 700, Stok: 7, Harga: 600 koin)
>>> Pilih nomor urut monster yang ingin dibeli:
4
Berhasil membeli monster.
>>> Selamat datang di Toko Mr. Yanto! Pilih aksi (beli/lihat/keluar):
lihat
>>> Mau lihat apa? (potion/monster):
potion
Potion yang tersedia:
1. strength (Stok: 10, Harga: 50 koin)
2. resilience (Stok: 5, Harga: 30 koin)
3. healing (Stok: 3, Harga: 20 koin)
>>> Selamat datang di Toko Mr. Yanto! Pilih aksi (beli/lihat/keluar):
4
>>> Selamat datang di Toko Mr. Yanto! Pilih aksi (beli/lihat/keluar):
palsu
>>> Selamat datang di Toko Mr. Yanto! Pilih aksi (beli/lihat/keluar):
keluar
Terima kasih telah berbelanja. Sampai jumpa lagi!
```

F11 Laboratory

6.11. Gambar Pengetesan *Laboratory*

```
===== MONSTER LIST =====
1. Pikachu (Level: 1)
2. Zuko (Level: 1)

===== UPGRADE PRICE =====
1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000 OC
>>> Pilih monster: 3
Pilihan monster tidak valid. Silakan pilih lagi.

===== MONSTER LIST =====
1. Pikachu (Level: 1)
2. Zuko (Level: 1)

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

Pikachow akan di-upgrade ke level 2.
Harga untuk melakukan upgrade adalah 300 OC.
>>> Lanjutkan upgrade (Y/N): Y
Selamat, Pikachu berhasil di-upgrade ke level 2!
Besarnya OC setelah upgrade: 9149
```

F12 Shop Management

6.12. Gambar Pengetesan *Shop Management*

Irasshaimase! Selamat datang kembali, Mr. Monogram!

Pilih aksi (lihat/tambah/ubah/hapus/keluar): salah
Input tidak valid. Ulangi

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

ID	Type	ATK Power	DEF Power	HP	Stock	Price
1	Pikachow	11	111	1111	10	100
2	Bulbu	22	222	2222	20	200
3	Zeze	33	333	3333	30	300

Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
Mau nambahin apa? (monster/potion): potion

ID	Type
3	Healing Potion

Masukkan id potion: 3

Masukkan stok awal: 12

Masukkan harga: 12

Healing Potion telah berhasil ditambahkan ke dalam shop!

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

ID	Type	ATK Power	DEF Power	HP	Stock	Price
1	Pikachow	11	111	1111	10	100
2	Bulbu	22	222	2222	20	200
3	Zeze	33	333	3333	30	300

Masukkan id monster: 2

Masukkan stok awal: 12

Masukkan harga: 12

Bulbu telah berhasil diubah dengan stok baru sejumlah 12 dan dengan harga baru 12!

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

ID	Type	Stok	Harga
1	Strength Potion	10	50
2	Resilience Potion	5	30
3	Healing Potion	12	12

Masukkan id potion: 3

Apakah anda yakin ingin menghapus Healing Potion dari shop (y/n)? y

Healing Potion telah berhasil dihapus dari shop!

Pilih aksi (lihat/tambah/ubah/hapus/keluar): keluar
Dadah Mr. Yanto, sampai jumpa lagi!

F13 Monster Management

6.13. Gambar Pengetesan *Monster Management*

```
Monster Management Menu:
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

Masukkan pilihan (1, 2, or 3): 1
Semua Monster:
1. Pikachu (ATK: 125, DEF: 10, HP: 600)
2. Bulbu (ATK: 50, DEF: 50, HP: 1200)
3. Zeze (ATK: 300, DEF: 10, HP: 100)
4. Zuko (ATK: 100, DEF: 25, HP: 800)
5. Chacha (ATK: 80, DEF: 30, HP: 700)

Monster Management Menu:
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

Masukkan pilihan (1, 2, or 3): 2
Masukkan nama monster baru: test
Masukkan hit points (integer positif): 12
Masukkan attack power (integer positif): 12
Masukkan defence power (integer 1-50): 12
Monster baru berhasil ditambahkan.

Monster Management Menu:
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

Masukkan pilihan (1, 2, or 3):
Silakan masukkan pilihan yang tersedia

Monster Management Menu:
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

Masukkan pilihan (1, 2, or 3): keluar
Silakan masukkan pilihan yang tersedia

Monster Management Menu:
1. Tampilkan semua monster
2. Tambahkan monster baru
3. Keluar

Masukkan pilihan (1, 2, or 3): 3
Menutup monster management...
```

F14 Load

6.14. Gambar Pengetesan *Load*

```
PS C:\Users\iqbal\Downloads\TBIF1210> py F14_Load.py

Tidak ada nama folder yang diberikan!
Usage : python/py main.py <nama_folder>
PS C:\Users\iqbal\Downloads\TBIF1210> py F14_Load.py data

Folder data tidak ditemukan.
PS C:\Users\iqbal\Downloads\TBIF1210> py F14_Load.py dataset_default

Loading...

Selamat datang di program OWCA!
```

F15 Save

6.15. Gambar Pengetesan Penyimpanan

```
Masukkan nama folder:
Input tidak valid

Masukkan nama folder: dataset4

Saving...

Membuat folder data/dataset4...
Berhasil menyimpan data di folder data/dataset4
```

F16 Exit

6.16.1. Gambar Pengetesan *Exit*

```
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) y
Masukkan nama folder: dataset6

Saving...

Membuat folder data/dataset6...
Berhasil menyimpan data di folder data/dataset6
```

6.16.2. Gambar Pengetesan *Exit*

```
register/login/logout/help/inventory/battle/arena/shop/lab/shop_mgmt/monster_mgmt/save/quit: quit  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n)  
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) n
```

LAMPIRAN ASISTENSI

Asistensi 1

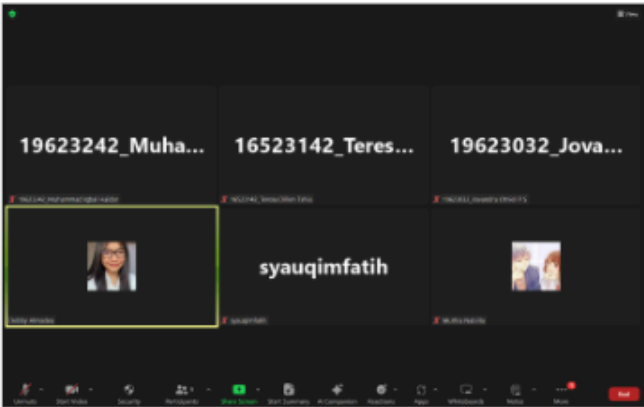
Gambar Lembar Asistensi 1

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

Nomor Asistensi : 1
 No. Kelompok/Kelas : H/K-02
 Tanggal asistensi : Kamis, 03/05/2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	19623032/Jovandra Otniel P. S.
	2	19623242/Muhammad Iqbal Haidar
	3	16523142/Teresa Dillen Tshia
	4	16523012/Muthia Nabilla Azzahra
	5	16523272/Syauqi Muhammad Fatih
Asisten pembimbing	6	
	NIM / Nama	
	13521153 / Made Debby Almadea Putri	

Catatan Asistensi:

Rangkuman Diskusi
<ul style="list-style-type: none"> • Untuk logout tidak perlu input username kembali • Pada F00 - RNG disesuaikan pada algoritma LCG dan untuk seed disesuaikan pada waktu • Perhatikan ketentuan mengenai library apa yang boleh digunakan • Pembuatan laporan jangan terlalu dekat dengan deadline
Tindak Lanjut
<ul style="list-style-type: none"> • Membuat file csv parser • Melanjutkan pekerjaan sesuai dengan pembagian tugas masing-masing
Dokumentasi


Asistensi 2

Gambar Lembar Asistensi 2

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

Nomor Asistensi : 2
No. Kelompok/Kelas : H/K-02
Tanggal asistensi : Minggu, 12/05/2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	19623032/Jovandra Otniel P. S.
	2	19623242/Muhammad Iqbal Haidar
	3	16523142/Teresa Dillen Tshia
	4	16523012/Muthia Nabilla Azzahra
	5	16523272/Syauqi Muhammad Fatih
Asisten pembimbing	6	
	NIM / Nama 13521153 / Made Debby Almadea Putri	

Catatan Asistensi:

Rangkuman Diskusi
<ol style="list-style-type: none">1. read() dan rstrip() tidak boleh digunakan2. pace pengerjaan harus ditingkatkan3. validasi input
Tindak Lanjut
<ol style="list-style-type: none">1. cari alternatif lain, definisikan sendiri2. memberikan validasi untuk setiap input pada program
Dokumentasi
