

**Laporan Tugas Besar IF1210 Dasar Pemrograman  
SEMESTER II 2023/2024**

**Program Pembantu Agent P Melawan Monster Ciptaan Dr. Asep  
Spakbor**

**Kelompok K01-A :**

Afrizal Hendra Pratama	(16523041)
Muhammad Nabil Raihan	(16523191)
Nadia Azwa Wibowo	(16523211)
Ardell Aghna Mahendra	(19623091)
Azfa Radhiyya Hakim	(19623261)



**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,  
Afrizal Hendra Pratama (16523041)  
Muhammad Nabil Raihan (16523191)  
Nadia Azwa Wibowo (16523211)  
Ardell Aghna Mahendra (19623091)  
Azfa Radhiyya Hakim (19623261)

## Daftar Isi

Pernyataan Kelompok.....	2
Daftar Isi.....	3
Daftar Tabel.....	4
Daftar Gambar.....	5
Deskripsi Persoalan.....	7
Daftar Pembagian Kerja Kelompok.....	8
Checklist Hasil Pengerjaan Tugas Besar.....	13
Desain Command.....	14
Desain Kamus Data.....	35
Desain Dekomposisi Algoritmik dan Fungsional Program.....	56
Spesifikasi Fungsi-Fungsi dalam Program.....	64
Hasil Pengujian Program.....	118
Lampiran.....	147

## **Daftar Tabel**

Tabel 01 : Daftar Pembagian Kerja Kelompok.....	8
Tabel 02 : Daftar Pembagian Pembuatan Laporan.....	11
Tabel 03 : Checklist Hasil Pengerjaan Tugas Besar.....	13

## Daftar Gambar

Gambar 01. Tampilan awal register user.....	118
Gambar 02. Tampilan awal login user terdaftar.....	118
Gambar 03. Tampilan logout.....	118
Gambar 04. Tampilan detail Menu & Help bagi user tanpa role.....	119
Gambar 05. Tampilan detail Menu & Help bagi Admin.....	119
Gambar 06. Tampilan detail Menu & Help bagi Agent.....	119
Gambar 07. Tampilan utama owcadex.....	119
Gambar 08. Tampilan detail monster awal.....	120
Gambar 09. Tampilan input tidak valid pada owcadex.....	120
Gambar 10. Tampilan detail monster sesuai level pilihan.....	120
Gambar 11. Tampilan awal penggunaan potion.....	121
Gambar 12. Tampilan efek peningkatan.....	121
Gambar 13. Tampilan input tidak valid saat penggunaan potion.....	121
Gambar 14. Tampilan penggunaan potion lain.....	122
Gambar 15. Tampilan healing potion jika hp masih penuh.....	122
Gambar 16. Tampilan resilience potion jika def sudah 50.....	122
Gambar 17. Tampilan utama inventory.....	123
Gambar 18. Tampilan input tidak valid dalam inventory.....	123
Gambar 19. Tampilan detail monster.....	124
Gambar 20. Tampilan detail potion.....	124
Gambar 21. Tampilan awal battle.....	125
Gambar 22. Tampilan kemunculan lawan dan pilih monster.....	125
Gambar 23. Tampilan kemunculan monster dan turn 1.....	126
Gambar 24. Tampilan attack dan detailnya.....	127
Gambar 25. Tampilan input tidak valid dalam battle.....	127
Gambar 26. Tampilan pilih potion.....	127
Gambar 27. Tampilan pilih cancel di penggunaan potion.....	128
Gambar 28. Tampilan tidak dapat memilih potion.....	128
Gambar 29. Tampilan skip battle dan kemenangan.....	128
Gambar 30. Tampilan quit battle.....	128
Gambar 31. Tampilan awal arena.....	129
Gambar 32. Tampilan pilih monster dan stage awal.....	130
Gambar 33. Tampilan muncul lawan dan battle arena.....	130
Gambar 34. Tampilan skip battle dan stage lanjutan.....	130
Gambar 35. Tampilan kalah battle dan akhir arena.....	130
Gambar 36. Tampilan awal shop.....	131
Gambar 37. Tampilan lihat monster.....	131
Gambar 38. Tampilan membeli monster yang sudah dimiliki.....	132
Gambar 39. Tampilan berhasil membeli monster.....	132

Gambar 40. Tampilan lihat potion.....	132
Gambar 41. Tampilan input tidak valid di shop.....	132
Gambar 42. Tampilan out of stock.....	133
Gambar 43. Tampilan berhasil membeli potion.....	133
Gambar 44. Tampilan out of coin.....	134
Gambar 45. Tampilan keluar shop.....	134
Gambar 46. Tampilan masuk lab.....	135
Gambar 47. Tampilan out of coin.....	136
Gambar 48. Tampilan input tidak valid di lab.....	136
Gambar 49. Tampilan upgrade berhasil.....	137
Gambar 50. Tampilan upgrade maksimum.....	138
Gambar 51. Tampilan keluar dari lab.....	138
Gambar 52. Tampilan awal Shop Management.....	139
Gambar 53. Tampilan Jika menambah monster.....	139
Gambar 54. Tampilan jika ingin mengubah potion.....	140
Gambar 55. Tampilan ketika selesai mengedit.....	140
Gambar 56. Tampilan membuat monster baru.....	141
Gambar 57. Tampilan melihat kembali monster yang telah ditambah.....	141
Gambar 58. Tampilan ketika selesai mengedit.....	142
Gambar 59. Tampilan tidak menerima inputan folder.....	142
Gambar 60. Tampilan inputan folder yang tidak ada.....	142
Gambar 61. Tampilan inputan folder yang valid (sudah ada).....	142
Gambar 62. Tampilan inputan folder baru.....	143
Gambar 63. Tampilan inputan nama folder yang sudah ada.....	143
Gambar 67. Tampilan awal Permainan Jackpot.....	144
Gambar 68. Tampilan ketika User menggerakkan Player.....	145
Gambar 69. Tampilan ketika mendekati jackpot dan akan bermain.....	146
Gambar 70. Form Asistensi 1.....	147
Gambar 71. Form Asistensi 2.....	148
Gambar 72. Penggunaan Git Best Practice (B01).....	149

## **Deskripsi Persoalan**

Tugas besar ini meminta untuk membuatkan program untuk membantu Purry si Platypus, alias Agent P, mengalahkan monster-monster jahat yang diciptakan oleh Dr. Asep Spakbor, yang mengancam keamanan kota Danville. Ada fungsionalitas untuk mengkoordinasikan bantuan dari agen-agen lain di markas rahasia O.W.C.A., termasuk perencanaan strategi untuk melawan monster-monster kuat tersebut. Purry membutuhkan bantuan karena monster terbaru Dr. Asep Spakbor terlalu kuat untuk dihadapi sendirian. Agen-agen tambahan, harus berkolaborasi dengan Purry untuk menyusun rencana yang matang.

Program ini juga memerlukan fungsionalitas untuk mencari dan melatih monster-monster sendiri yang akan digunakan dalam *battle*. Pencarian dilakukan di hutan terpencil, yang dikenal sebagai tempat tinggal banyak jenis monster. Petualangan di hutan ini akan menguji keberanian dan ketangkasan para Agent. Program menekankan validasi setiap aksi yang dilakukan untuk memastikan semua langkah berjalan dengan benar. Desain tampilan atau interface sistem dibebaskan untuk berkreasi, dan kreativitas dalam desain akan menjadi salah satu pertimbangan dalam penilaian.

### **Spesifikasi Program**

Terdapat kebutuhan fungsional wajib yang harus dipenuhi oleh Agent. Desain tampilan atau interface sistem dibebaskan, sehingga dapat berkreasi sesuai keinginan, output tidak harus persis seperti contoh, yang penting spesifikasi terpenuhi. Kreativitas interface akan menjadi salah satu pertimbangan penilaian.

Penekanan diberikan pada alur program dan validasi setiap aksi yang dilakukan. Banyaknya fungsionalitas wajib tidak berarti program harus panjang dan kompleks. Dengan program modular dan fungsi yang jelas, program dapat dibuat singkat dan sederhana.

## Daftar Pembagian Kerja Kelompok

**Tabel 01 : Daftar Pembagian Kerja Kelompok**

<b>Fitur</b>	<b>Implementasi</b>	<b>NIM Desainer</b>	<b>NIM Coder</b>	<b>NIM Tester</b>
F00-Random Number Generator	function LCG  Function RandomNumber	19623261	19623261	19623261
F01-Register	function valid  function register	19623261	19623261	19623261
F02-Login	function login	19623261	19623261	19623261
F03-Logout	function logout	16523041	16523041	19623261
F04-Menu & Help	function help_sudah_login_admin  procedure list_help_admin  function help_sudah_login_agent  Procedure list_help_agent	19623261	19623261	19623261
F05-Monster	function lvlStat  function TampilMonsterDetail  function TampilanOwcaDex  Procedure owcaDex	16523191	16523191	16523191 19623261
F06-Potion	function PeningkatanEfek  function	16523191	16523191	16523191 19623261



	GunakanPotionLain function TampilanPotionUse function PotionUse			
F07-Inventory	function ItemInventory  function UserMonsters  function UserPotions  function KembaliKeInventory  function TampilPotionDetail  function TampilanInv  procedure TampilInventoryUser	16523191	16523191	16523191 19623261
F08-Battle	function RandomMonster  procedure KemunculanMonster  function PilihUserMonster  function Skip  function Attack  function CanUsePotion  function BattleTurns  function Battle	16523191	16523191	16523191 19623261

F09-Arena	function CekBonus function Arena	16523191	16523191	16523191 19623261
F10-Shop & Currency	procedure shop	16523041	16523041	16523041 19623261
F11-Laboratory	procedure laboratory function upgrade_cost	16523211	16523211	16523211 19623261
F12-Shop Management	function is_integer Procedure shop_management	19623261	19623261	19623261
F13-Monster Management	function monster_management	16523041	16523041	16523041 19623261
F14-Load	procedure load function parse_csv	19623091	19623091	19623091 19623261
F15-Save	procedure save function save_data	19623091	19623091	19623091 19623261
F16-Exit	procedure exit	19623091	19623091	19623091 19623261
B01-Git Best Practice	commit message	16523041 16523191 16523211 19623091 19623261	16523041 16523191 16523211 19623091 19623261	16523041 16523191 16523211 19623091 19623261
B02-Typing	-	16523041 16523191 16523211 19623091 19623261	16523041 16523191 16523211 19623091 19623261	16523041 16523191 16523211 19623091 19623261
B03-Monster Ball	-	-	-	-

B04-JACKPOT!	function jackpot	19623261	19623261	19623261
B05-Peta Kota Danville	procedure peta_danville function bush function komando	19623261	19623261	19623261

**Tabel 02 : Daftar Pembagian Pembuatan Laporan**

No.	Bagian Laporan	NIM
1	Halaman Cover	16523041 16523191 16523211 19623091 19623261
2	Daftar Isi	16523041 16523191 16523211 19623091 19623261
3	Daftar Tabel	16523041 16523191 16523211 19623091 19623261
4	Daftar Gambar	16523041 16523191 16523211 19623091 19623261
5	Deskripsi Persoalan	16523041 16523191 16523211 19623091 19623261
6	Daftar Pembagian Kerja Kelompok	16523041 16523191 16523211 19623091 19623261

7	Checklist Hasil Pengerjaan Tugas Besar	16523041 16523191 16523211 19623091 19623261
8	Desain Command	16523041 16523191 16523211 19623091 19623261
9	Desain Kamus Data	16523041 16523191 16523211 19623091 19623261
10	Desain Dekomposisi Algoritmik dan Fungsional Program	16523041 16523191 16523211 19623091 19623261
11	Spesifikasi Fungsi-Fungsi dalam Program	16523041 16523191 16523211 19623091 19623261
12	Hasil Pengujian Program	16523041 16523191 16523211 19623091 19623261
13	Lampiran	16523041 16523191 16523211 19623091 19623261

## ***Checklist Hasil Pengerjaan Tugas Besar***

**Tabel 03 : *Checklist Hasil Pengerjaan Tugas Besar***

<b>Fitur</b>	<b>Desain</b>	<b>Implementasi</b>	<b>Testing</b>
F00-Random Number Generator	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
B01-Git Best Practice	V	V	V
B02-Typing	V	V	V
B03-Monster Ball	-	-	-
B04-JACKPOT!	V	V	V
B05-Peta Kota Danville	V	V	V

## Desain Command

### 1. F01-Register

```
Masukkan command 'help' untuk melihat list command
>>> 1
Silahkan mengisi username baru dan password yang aman :D
Username hanya boleh berisi alfabet, angka, underscore, dan strip!
Masukkan username: namaakuitb?!<>
Masukkan password: password
Username hanya boleh berisi alfabet, angka, underscore, dan strip!
```

```
Masukkan command 'help' untuk melihat list command
>>> 1
Masukkan username: AGENT_1
Masukkan password: akuanakitb
Username AGENT_1 sudah terpakai, silahkan gunakan username lain!
```

```
Masukkan command 'help' untuk melihat list command
>>> 1
Masukkan username: daspro
Masukkan password: akucintadaspro
Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachu
2. Bulbu
3. Zeze
Masukkan pilihan Anda: 4
Masukkan tidak valid, ayo yang benar!
```

```
Masukkan command 'help' untuk melihat list command
>>> 1
Masukkan username: daspro
Masukkan password: akucintadaspro
Silahkan pilih salah satu monster sebagai monster awalmu.
1. Pikachu
2. Bulbu
3. Zeze
Masukkan pilihan Anda: 1
Selamat datang Agent daspro. Mari kita mengalahkan Dr. Asep Spakbor dengan Pikachu!
Masukkan command "help" untuk daftar command yang dapat kamu panggil.
```

### 2. F02-Login

```
LOGIN
```

```
Masukkan command 'help' untuk melihat list command
>>> 2
Halo! Silahkan login dengan username dan password anda! (ketik 'cancel' pada username
untuk kembali ke menu awal)
Username: dasprooooo
Password: 123456
Username tidak terdaftar!
```

```
Masukkan command 'help' untuk melihat list command
```

```
>>> 2
Halo! Silahkan login dengan username dan password anda! (ketik 'cancel' pada username
untuk kembali ke menu awal)
Username: cancel
Password: 123456 #bebas diisi dengan apa saja
Silahkan masukkan command kembali
```

```
Masukkan command 'help' untuk melihat list command
>>> 2
Halo! Silahkan login dengan username dan password anda! (ketik 'cancel' pada username
untuk kembali ke menu awal)
Username: AGENT_1
Password: akuanakitb
Selamat datang, agent AGENT_1!
Masukkan command "help" untuk daftar command yang dapat kamu panggil
```

### 3. F03-Logout

```
Command:
>>> 1
Apakah Anda yakin ingin logout? Semua file yang tidak di save akan terhapus! (y/n): n
Kembali ke program.
Command:
>>> 1
Apakah Anda yakin ingin logout? Semua file yang tidak di save akan terhapus! (y/n): y
SAMPAI JUMPA KING!
```

### 4. F04-Menu & Help

```
Masukkan command 'help' untuk melihat list command
>>> help
=====HELP=====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.
1. Register: Membuat akun baru
2. Login: Masuk ke dalam akun yang sudah terdaftar
Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
=====
```

```
Masukkan command "help" untuk daftar command yang dapat kamu panggil.
Command:
>>> help
=====HELP=====
Halo Agent!. Kamu memanggil command Help. Kamu memilih jalan yang benar, semoga kamu
tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
1. Logout: Keluar dari akun yang sedang digunakan
2. Monster: Melihat owca-dex yang dimiliki oleh Agent
3. Potion: Melihat Potion yang dimiliki oleh Agent
4. Mulai bermain!
5. Save data
6. Selesai bermain
Footnote:
1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
2. Jangan lupa untuk memasukkan input yang valid
=====
```

## 5. F05-Monster

```
Masukkan command: owcadex
# menampilkan daftar monster yang ada di dalam game

===== OWCA DEX =====
1. Pikachu
2. Bulbu
3. Zeze
4. Zuko
5. Chacha
6. ...

Ketik monster id untuk melihat detail.
Ketik 0 untuk keluar

>>> Pilih:
```

```
>>> Pilih: 2
# menampilkan monster dengan index id = 2, lvl di set = 1

===== BULBU LVL 1 =====
Monster id : 2
ATK Power  : 50
DEF Power  : 50
HP          : 1200

Ketik 1-5 untuk melihat status monster pada level tersebut.
Ketik 0 untuk kembali ke OWCA DEX.

>>> Pilih:
```

```
>>> Pilih: k
# input yang tidak sesuai akan diminta input kembali, tampilan tetap

[Peringatan] Pilihan tidak sesuai!

===== BULBU LVL 1 =====
Monster id : 2
ATK Power  : 50
DEF Power  : 50
HP          : 1200

Ketik 1-5 untuk melihat status monster pada level tersebut.
Ketik 0 untuk kembali ke OWCA DEX.

>>> Pilih:
```

```
>>> Pilih: 3
# menampilkan status Bulbu pada level 3, setiap stat meningkat 10% setiap kenaikan
level, ditampilkan dalam bentuk integer, def power max = 50

===== BULBU LVL 3 =====
Monster id : 2
ATK Power  : 60
DEF Power  : 50
HP          : 1440

Ketik 1-5 untuk melihat status monster pada level tersebut.
Ketik 0 untuk kembali ke OWCA DEX.

>>> Pilih:
```

```
>>> Pilih: 0
# kembali ke tampilan utama, ketik 0 lagi untuk keluar

===== OWCA DEX =====
```



```
1. Pikachu
2. Bulbu
3. Zeze
4. Zuko
5. Chacha

Ketik monster id untuk melihat detail.
Ketik 0 untuk keluar

>>> Pilih:
```

## 6. F06-Potion

```
>>> Pilih: 2
# dari battle, jika potion akan digunakan, ditampilkan semua potion yang tersedia,
setiap potion hanya digunakan sekali dalam battle
```

```
Pilih potion:
1. Resilience Potion    (Qty: 1| Uses: 1)
2. Strength Potion      (Qty: 1| Uses: 1)
3. Cancel
```

```
>>> Pilih:
```

```
>>> Pilih: 1
# jika def = 50 atau hp = hp max, tidak dapat menggunakan potion yang bersangkutan

[Peringatan] DEF monster sudah mencapai batas maksimum.
Tidak bisa menggunakan Resilience potion.
```

```
>>> Mau gunakan potion lain? (y/n)
```

```
>>> Mau gunakan potion lain? (y/n) y
# jika memilih y atau Y, kembali memilih potion, jika memilih n atau N, kembali ke
battle untuk memilih pilihan lainnya. Pilihan selain yang disebut, user akan diminta
input ulang
```

```
Pilih potion:
1. Resilience Potion    (Qty: 1| Uses: 1)
2. Strength Potion      (Qty: 1| Uses: 1)
3. Cancel
```

```
>>> Pilih:
```

```
>>> Pilih: 2
# menampilkan efek yang ditingkatkan dari stat yang bersangkutan dengan potion yang
dipilih.
```

```
Efek:
ATK Power = 70 --> 73
```

```
>>> Yakin ingin menggunakan potion ini? (y/n)
# pilihan y atau Y akan menerapkan efek lalu kembali ke battle, n atau N memberikan
kesempatan user untuk memilih kembali potion atau tidak, input lainnya meminta user
untuk input ulang
```

## 7. F07-Inventory

```
Masukkan command: inventory
# menampilkan inventory user dengan coin yang dimiliki, monster letaknya di atas
item, secara spesifik seperti di bawah:
```

```

===== LEBAH'S INVENTORY (User ID: 6) =====
Jumlah O.W.C.A Coin mu sekarang adalah 7047
1. Monster      (Name: Zeze, Lvl: 5, HP: 140)
2. Monster      (Name: Bulbu, Lvl: 5, HP: 1680)
3. Potion       (Type: DEF, Qty: 1)
4. Potion       (Type: ATK, Qty: 1)

Ketik id untuk melihat detail
Ketik 0 untuk keluar

>>> Pilih:

>>> Pilih: 2
# menampilkan detail monster dengan stat sesuai levelnya

===== BULBU LVL 5 =====
Monster id   : 2
ATK Power    : 70
DEF Power    : 50
HP           : 1680

Ketik 1 untuk kembali ke inventory
Ketik 0 untuk keluar

>>> Pilih: 1
# kembali ke tampilan utama inventory

>>> Pilih: 3
# menampilkan detail item(potion), jumlah yang dimiliki, kegunaan, dan batasannya

===== RESILIENCE POTION =====
(Dimiliki: 1)

Potion ini memberikan efek peningkatan DEF Power sebesar 5%. Item ini
hanya dapat digunakan sekali dalam battle. Tidak dapat digunakan jika
DEF Power monster sudah mencapai maksimum (50).

Ketik 1 untuk kembali ke inventory
Ketik 0 untuk keluar

>>> Pilih:

```

## 8. F08-Battle

```

Masukkan command: battle
# masuk ke dalam battle, menampilkan intro lawan, monster lawan di-generate secara
random dengan level yang random pula, user diminta untuk memilih monsternya untuk
bertarung.

RAAWRRRR!!! Monster Bulbu telah muncul!!!

Name       : Bulbu
ATK Power  : 50
Def Power  : 50
HP         : 1200
Level      : 1

Pilih Monster-mu untuk bertarung!

===== MONSTER LIST =====
1. Zeze
2. Bulbu

>>> Pilih:

```

```
>>> Pilih: 2
# menampilkan intro monster yang dipilih, serta giliran secara bergantian, turn
ganjil = giliran user. User dapat memilih untuk menyerang, menggunakan potion, skip
battle, dan meninggalkan battle.
```

```
RAAWRRRR!!! Lebah mengeluarkan monster Bulbu!!!
```

```
Name      : Bulbu
ATK Power  : 70
Def Power  : 50
HP         : 1680
Level      : 5
```

```
===== TURN 1 (BULBU - LEBAH) =====
```

```
1. Attack
2. Use Potion
3. Skip Battle
4. Quit
```

```
>>> Pilih:
```

```
>>> Pilih: 1
# setiap turn akan ditampilkan siapa penyerang dan sisa HP defender, serta
menampilkan detail attack.
```

```
(LEBAH) Bulbu menyerang!!!
```

```
(ENEMY) Bulbu's HP = 1158
```

```
===== Attack Detail =====
```

```
Attack Power : 83 (+19%)
Reduced by    : 41 (-50%)
Result        : 42
```

```
===== TURN 2 (BULBU - ENEMY) =====
```

```
(ENEMY) Bulbu menyerang!!!
```

```
(LEBAH) Bulbu's HP = 1657
```

```
===== Attack Detail =====
```

```
Attack Power : 45 (-10%)
Reduced by    : 22 (-50%)
Result        : 23
```

```
===== TURN 3 (BULBU - LEBAH) =====
```

```
1. Attack
2. Use Potion
3. Skip Battle
4. Quit
```

```
>>> Pilih:
```

```
>>> Pilih: 2
# penggunaan potion dijelaskan pada bagian F06
```

```
>>> Pilih: 3
# skip battle = battle secara otomatis dikalkulasikan kondisi kemenangannya dengan
monster selalu dalam keadaan menyerang.
```

```
Dengan memilih skip battle, monster dengan otomatis dalam keadaan menyerang
```

```
>>> Yakin ingin skip battle? (y/n) y
# user diberi kesempatan untuk meyakinkan pilihannya
```

```
Selamat, Anda berhasil mengalahkan monster Bulbu!!!
```

```
===== STATS =====
```

```
Hadiah      : 35 OC
```

```
Damage diberikan      : 1214
Damage diterima       : 821
# jika menang diberi hadiah coin dalam rentang 30-50 coin. Damage ditampilkan dalam
# bentuk integer, damage bisa melebihi hp max
```

```
>>> Pilih: 4
# pilihan quit

Anda melarikan diri dari pertarungan

===== STATS =====
Hadiah                : 0 OC
Damage diberikan      : 0
Damage diterima       : 0
# kabur tidak akan mendapat bonus apapun
```

```
>>> Yakin ingin skip battle? (y/n) y

Yahhh, Anda dikalahkan monster Chacha. Jangan menyerah, coba lain kali!!!

===== STATS =====
Hadiah                : 0 OC
Damage diberikan      : 420
Damage diterima       : 144
# kekalahan tidak akan mendapat bonus apapun
```

## 9. F09-Arena

```
Masukkan command: arena
# masuk ke dalam arena, user memilih monster terlebih dahulu
```

Selamat datang di Arena!

Pilih Monster-mu untuk bertarung!

```
===== MONSTER LIST =====
1. Bulbu
2. Zeze
```

>>> Pilih:

```
>>> Pilih: 1
# setelah memilih monster, ditampilkan detail monster yang dipilih
```

RAAWRRRR!!! Agen\_P mengeluarkan monster Bulbu!!!

```
Name      : Bulbu
ATK Power  : 55
Def Power  : 50
HP         : 1320
Level      : 2
```

```
===== STAGE 1 =====
# stage 1 dimulai, spesifikasi sama dengan battle
```

RAAWRRRR!!! Monster Zuko telah muncul!!!

```
Name      : Zuko
ATK Power  : 100
Def Power  : 25
HP         : 800
Level      : 1
```

```
===== TURN 1 (BULBU - AGEN_P) =====
1. Attack
```

```
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih:

Selamat, Anda berhasil mengalahkan monster Zuko!!!
# tampilan jika menang, lanjut ke stage berikutnya, hp monster penuh kembali

STAGE CLEARED! Anda mendapatkan 31 OC pada stage ini

Memulai stage berikutnya . . .

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

RAAWRRRR!!! Monster Zeze telah muncul!!!

Name      : Zeze
ATK Power  : 330
Def Power  : 11
HP         : 110
Level      : 2

===== TURN 1 (BULBU - AGEN_P) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih:

Yahhh, Anda dikalahkan monster Bulbu. Jangan menyerah, coba lain kali!!!
# tampilan jika kalah, ditampilkan total hadiah, stage yang diselesaikan, dan jumlah
damage yang diberikan dan diterima

GAME OVER! Sesi latihan berakhir pada stage 3

===== STATS =====
Total hadiah      : 64
Jumlah stage      : 2
Damage diberikan  : 2164
Damage diterima   : 2558
```

## 10. F10-Shop & Currency

```
=====

          SHOP

=====

Selamat datang di Shop!
Silahkan pilih aksi (lihat/beli/keluar) : lihat
Mau lihat apa? (monster/potion): monster

ID      |Type      |ATK Power|DEF Power|HP      |Stok     |Harga    |
-----|-----|-----|-----|-----|-----|-----|
1       |Pikachow  |125     |10       |600     |10       |30       |
2       |Bulbu     |50      |50       |1200    |4        |700      |
3       |Zeze      |300     |10       |100     |3        |1000     |
4819127 |Zuko      |100     |25       |800     |10       |10       |
1725431 |Chacha    |80      |30       |700     |10       |20       |
6132743 |Luffy     |437     |20       |800     |100      |100      |
```

Mau lihat apa? (monster/potion): **potion**

ID	Type	Stock	Harga
1	strength	10	50
2	resilience	5	30
3	healing	3	20

Mau beli apa? (monster/potion) : **monster**

Masukkan id monster : 12345

ID tidak ditemukan dalam data.

Mau beli apa? (monster/potion) : **monster**

Masukkan id monster : 6132743

Koin tersisa: 5000 OC

Mau beli apa? (monster/potion) : **potion**

Masukkan ID item potion : **1**

Pembelian Gagal :( , Stock tidak mencukupi

Mau beli apa? (monster/potion) : **potion**

Masukkan ID item potion : 2

Selamat!!! Anda Telah membeli potion resilience sebanyak 4

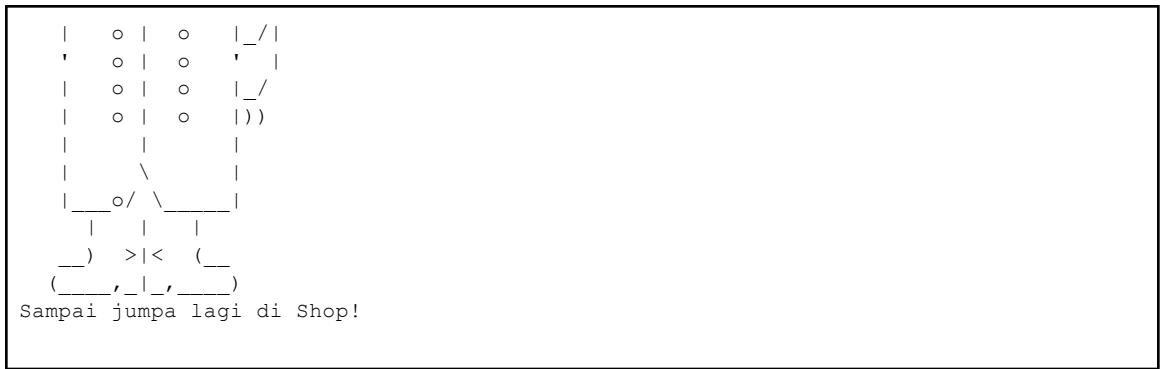
Mau beli apa? (monster/potion) : **potion**

Masukkan ID item potion : 2

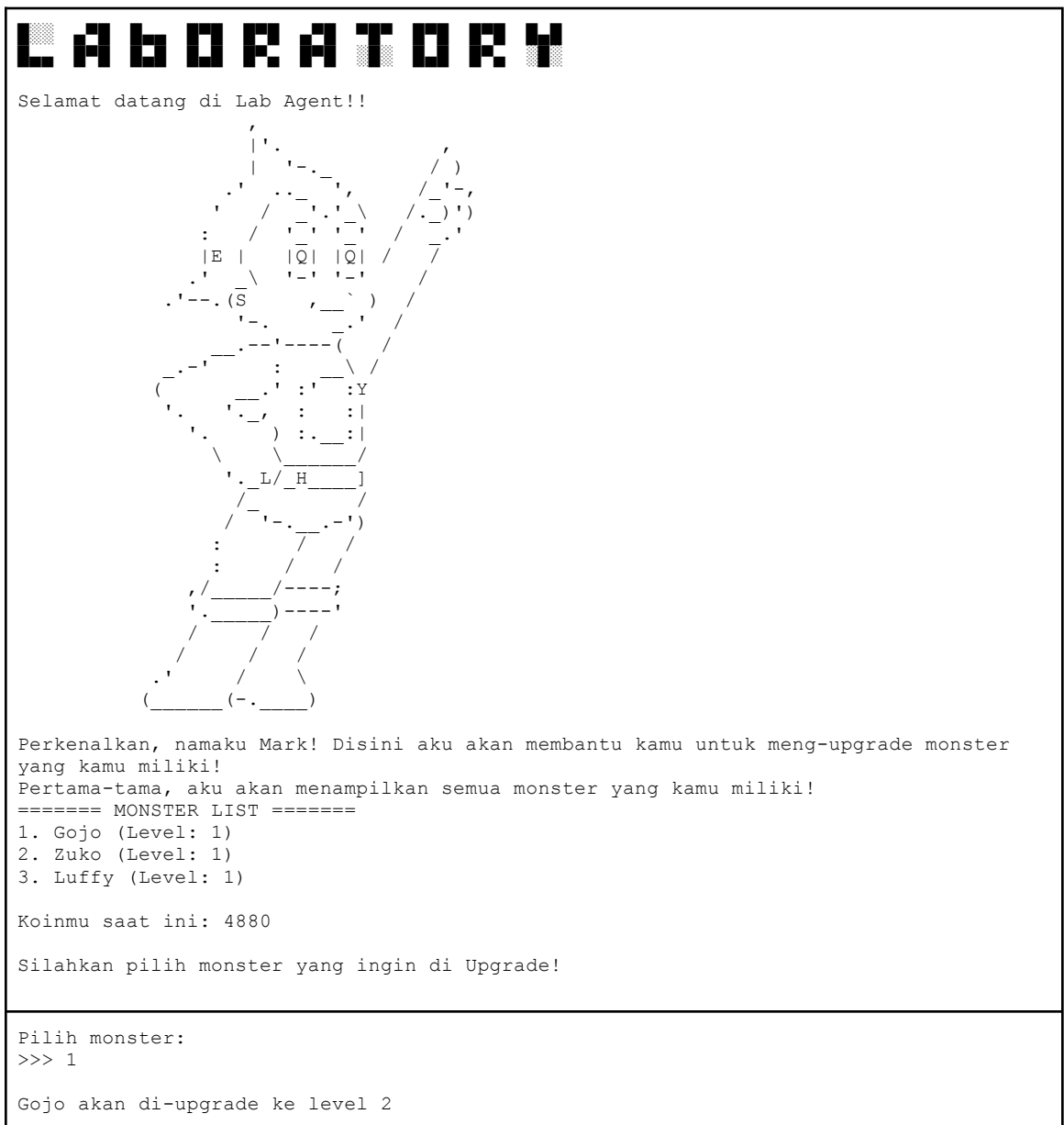
Pembelian Gagal :( , Stock tidak mencukupi

Silahkan pilih aksi (lihat/beli/keluar) : keluar

Thankyou...



## 11. F11-Laboratory



Harga untuk melakukan upgrade adalah 200 OC.

>>> Lanjutkan upgrade (Y/N): y

Selamat, Gojo berhasil di-upgrade ke level 2!

Apakah kamu ingin melanjutkan upgrade?

Ketik nomor dari pilihan berikut

1. Lanjut upgrade
2. Keluar

>>> 1

Koinmu saat ini: 4680

Silahkan pilih monster yang ingin di Upgrade!

Pilih monster:

>>> 1

Gojo akan di-upgrade ke level 3

Harga untuk melakukan upgrade adalah 300 OC.

>>> Lanjutkan upgrade (Y/N): y

Selamat, Gojo berhasil di-upgrade ke level 3!

Apakah kamu ingin melanjutkan upgrade?

Ketik nomor dari pilihan berikut

1. Lanjut upgrade
2. Keluar

>>> 1

Koinmu saat ini: 4380

Silahkan pilih monster yang ingin di Upgrade!

Pilih monster

>>> 1

Gojo akan di-upgrade ke level 4

Harga untuk melakukan upgrade adalah 600 OC.

>>> Lanjutkan upgrade (Y/N): y

Selamat, Gojo berhasil di-upgrade ke level 4!

Apakah kamu ingin melanjutkan upgrade?

Ketik nomor dari pilihan berikut

1. Lanjut upgrade
2. Keluar

>>> 1

Koinmu saat ini: 3780

Silahkan pilih monster yang ingin di Upgrade!

Pilih monster

>>> 1

Gojo akan di-upgrade ke level 5

Harga untuk melakukan upgrade adalah 800 OC.

>>> Lanjutkan upgrade (Y/N): y

Selamat, Gojo berhasil di-upgrade ke level 5!

Apakah kamu ingin melanjutkan upgrade?

Ketik nomor dari pilihan berikut

1. Lanjut upgrade
2. Keluar

>>> 1

Koinmu saat ini: 2980



```
Silahkan pilih monster yang ingin di Upgrade!
Pilih monster:
>>> 1
Maaf, monster yang Anda pilih sudah memiliki level maksimum.

Apakah kamu ingin melanjutkan upgrade?
Ketik nomor dari pilihan berikut
1. Lanjut upgrade
2. Keluar

>>> 45
Perintah tidak dikenal!

Perintah tidak dikenal!
1. Lanjut upgrade
2. Keluar

>>> 1
Koinmu saat ini: 2980

Silahkan pilih monster yang ingin di Upgrade!
Pilih monster:
>>> 1
Pilihan monster tidak valid. Silakan pilih lagi.
```

## 12. F12-Shop Management

```
Command:
>>> 2
=====

SHOP MANAGEMENT
=====
Konichiwa :D, Selamat datang kembali Admin!
Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
Mau lihat apa? (monster/potion/back): monster
ID      |Type      |ATK Power |DEF Power |HP      |Stok      |Harga      |
-----|-----|-----|-----|-----|-----|-----|
1      |Pikachow  |125       |10        |600     |10        |30         |
2      |Bulbu     |50        |50        |1200    |4         |700        |
3      |Zeze      |300       |10        |100     |3         |1000       |
4819127|Zuko      |100       |25        |800     |10        |10         |
1725431|Chacha    |80        |30        |700     |10        |20         |
6132743|Luffy     |437       |20        |800     |100       |100        |

Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
Mau lihat apa? (monster/potion/back): potion
ID |Type      |Stok      |Harga      |
---|-----|-----|-----|
1  |strength  |10        |50         |
2  |resilience|5         |30         |
3  |healing   |3         |20         |

Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
Mau tambah apa? (monster/potion/back): monster
ID      |Type      |ATK Power |DEF Power |HP      |
-----|-----|-----|-----|
```

```
-----|-----|-----|-----|-----|
8760271 |Gojo      |526      |47      |700      |
9441521 |Sasuke    |333      |18      |767      |
9573653 |JOKER     |1000     |49      |800      |
```

Jika tidak jadi menambahkan, silahkan ketik 'cancel'  
Masukkan id monster: 767676  
ID yang kamu masukan tidak valid! ayo masukan id lagi!  
Masukkan id monster: 9441521  
Masukkan stok: 20  
Masukkan harga: 500  
Monster berhasil ditambah!

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

Jika tidak jadi menambahkan, silahkan ketik 'cancel'  
Masukkan id potion: cancel  
Pilih aksi (lihat/tambah/ubah/hapus/keluar):

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

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	10	30
2	Bulbu	50	50	1200	4	700
3	Zeze	300	10	100	3	1000
4819127	Zuko	100	25	800	10	10
1725431	Chacha	80	30	700	10	20
6132743	Luffy	437	20	800	100	100
9441521	Sasuke	333	18	767	20	500

Jika tidak jadi mengubah, silahkan ketik -1  
Masukkan id monster: 9441521  
Masukkan stok baru: 15  
Masukkan harga baru: 600  
Monster berhasil diubah!

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

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	10	30
2	Bulbu	50	50	1200	4	700
3	Zeze	300	10	100	3	1000
4819127	Zuko	100	25	800	10	10
1725431	Chacha	80	30	700	10	20
6132743	Luffy	437	20	800	100	100
9441521	Sasuke	333	18	767	15	600

Jika tidak jadi mengubah, silahkan ketik -1  
Masukkan id monster: -1  
Pilih aksi (lihat/tambah/ubah/hapus/keluar):

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

ID	Type	Stok	Harga
1	strength	10	50
2	resilience	5	30
3	healing	3	20

Jika tidak jadi menambahkan, silahkan ketik -1  
Masukkan id potion: 1  
Masukkan stok baru: 9  
Masukkan harga baru: 60  
Item telah berhasil diubah!

```
Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
Mau hapus apa? (monster/potion/back): monster
ID      |Type      |ATK Power |DEF Power |HP      |Stok      |Harga      |
-----|-----|-----|-----|-----|-----|-----|
1      |Pikachow  |125       |10        |600     |10        |30         |
2      |Bulbu     |50        |50        |1200    |4         |700        |
3      |Zeze      |300       |10        |100     |3         |1000       |
4819127|Zuko      |100       |25        |800     |10        |10         |
1725431|Chacha    |80        |30        |700     |10        |20         |
6132743|Luffy     |437       |20        |800     |100       |100        |
9441521|Sasuke    |333       |18        |767     |15        |600        |
```

Jika tidak jadi menghapus, silahkan ketik -1

Masukkan id monster: 12345

ID yang kamu masukan tidak valid! ayo masukan id lagi!

Masukkan id monster: 6132743

Apakah anda yakin (y/n)? y

Monster berhasil dihapus!

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

Mau lihat apa? (monster/potion/back): monster

```
ID      |Type      |ATK Power |DEF Power |HP      |Stok      |Harga      |
-----|-----|-----|-----|-----|-----|-----|
1      |Pikachow  |125       |10        |600     |10        |30         |
2      |Bulbu     |50        |50        |1200    |4         |700        |
3      |Zeze      |300       |10        |100     |3         |1000       |
4819127|Zuko      |100       |25        |800     |10        |10         |
1725431|Chacha    |80        |30        |700     |10        |20         |
9441521|Sasuke    |333       |18        |767     |15        |600        |
```

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

Mau hapus apa? (monster/potion/back): potion

```
ID |Type      |Stok      |Harga      |
---|-----|-----|-----|
1  |strength  |9         |60         |
2  |resilience|5         |30         |
3  |healing   |3         |20         |
```

Jika tidak jadi menambahkan, silahkan ketik -1

Masukkan id potion: 3

Apakah anda yakin (y/n)? y

Berhasil dihapus!

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

Mau lihat apa? (monster/potion/back): potion

```
ID |Type      |Stok      |Harga      |
---|-----|-----|-----|
1  |strength  |9         |60         |
2  |resilience|5         |30         |
```

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

Mau tambah apa? (monster/potion/back): monster

```
ID      |Type      |ATK Power |DEF Power |HP      |
-----|-----|-----|-----|-----|
8760271 |Gojo      |526       |47        |700     |
6132743 |Luffy     |437       |20        |800     |
9573653 |JOKER     |1000      |49        |800     |
```

Jika tidak jadi menambahkan, silahkan ketik 'cancel'

Masukkan id monster: foienfocnp

Input yang dimasukkan tidak valid!

Masukkan id monster: cancel

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

### 13. F13-Monster Management

MONSTER MANAGEMENT					
Selamat datang di Monster Management! ----SELAMAT DATANG DI MENU DATA MONSTER MANAGEMENT---- ----SILAHKAN PILIH FITUR---- 1. Tampilkan Monster yang anda miliki 2. Tambahkan Monster Baru 3. Keluar					
Pilih fitur diatas (1/2/3) :1					
1	Pikachow	125	10	600	
2	Bulbu	50	50	1200	
3	Zeze	300	10	100	
4819127	Zuko	100	25	800	
1725431	Chacha	80	30	700	
8760271	Gojo	526	47	700	
6132743	Luffy	437	20	800	
9441521	Sasuke	333	18	767	
9573653	JOKER	1000	49	800	
Pilih fitur diatas (1/2/3) :2					
Masukkan Nama Monster barumu : dasprogogogo Masukkan ATK Power : 100 Masukkan DEF Power (0-50) : 90 INPUT SALAH!! DEF Power harus bernilai 0-50, coba lagi! Masukkan DEF Power Baru (0-50): 47 Masukkan HP Monster : 100 BERHASIL, Monster Baru Telah Dibuat!! Type : dasprogogogo ATK Power : 100 DEF Power : 47 HP : 100 Apakah Monster baru akan dimasukkan ke database? (Y/N) >>> y Monster berhasil dimasukkan ke database!					
Pilih fitur diatas (1/2/3) :2					
Masukkan Nama Monster barumu : dasprogogogo Masukkan ATK Power : 400 Masukkan DEF Power (0-50) :8 Masukkan HP Monster : 89 BERHASIL, Monster Baru Telah Dibuat!! Type : dasprogogogo ATK Power : 400 DEF Power : 8 HP : 89 Apakah Monster baru akan dimasukkan ke database? (Y/N) >>> n Penambahan Monster Dibatalkan!!					
Pilih fitur diatas (1/2/3) :3					
<div>Thankyou...</div> <div>/</div>					



## 14. F14-Load

```
python Load.py test
Loading...
# program akan dilanjutkan dengan data yang diload
```

```
python Load.py
Tidak ada nama folder yang diberikan! Harap masukkan nama folder!
# keluar dari program
```

```
python Load.py owca
Yah, folder "owca" tidak dapat ditemukan.
# keluar dari program
```

## 15. F15-Save

```
# folder belum pernah dibuat
>>> Save
Masukkan nama folder untuk penyimpanan: user1

Membuat folder savedata\user1...

Saving....
Berhasil menyimpan data di folder savedata\user1
```

```
# folder sudah pernah dibuat
>>> Save
Masukkan nama folder untuk penyimpanan: user1

Folder savedata\user1 sudah ada.

Saving....
Berhasil menyimpan data di folder savedata\user1
```

## 16. F16-Exit

```
Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan
terhapus! (y/n): n
```

```
Kembali ke program.
```

```
Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan
terhapus! (y/n): d
```

```
# input yang dimasukkan tidak valid
```

```
Pilihan tidak valid!
```

```
Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan
terhapus! (y/n): y
```

```
# menjalankan prosedur save (F15) dan keluar dari program
```

## 17. B04-JACKPOT!

# JACKPOT

```
===== Daftar Item =====
```

1. Topi: 50 OC
2. Pedang: 100 OC
3. Koin: 200 OC
4. Potion: 300 OC
5. Monster: 500 OC

```
Mulai bermain? (y/n): y
```

```
Koin kamu saat ini adalah 5100 OC
$$$$$$$$$      Topi | Monster | Pedang      $$$$$$$$$$
650 OC telah ditambahkan ke akun Anda!
```

```
Lanjut bermain (y/n): y
Koin kamu saat ini adalah 5350 OC
$$$$$$$$$      Topi | Topi | Topi      $$$$$$$$$$
```

```
JACKPOTTT!!! Monster Luffy telah dimasukkan dalam inventory anda!
```

```
Lanjut bermain (y/n): y
Koin kamu saat ini adalah 4950 OC
```

```
Lanjut bermain (y/n): n
Kamu mendapat koin sebesar 650 OC
Koin kamu saat ini adalah 4550 OC!
Selamat tinggal!
```

## 18. B05-Peta Kota Danville

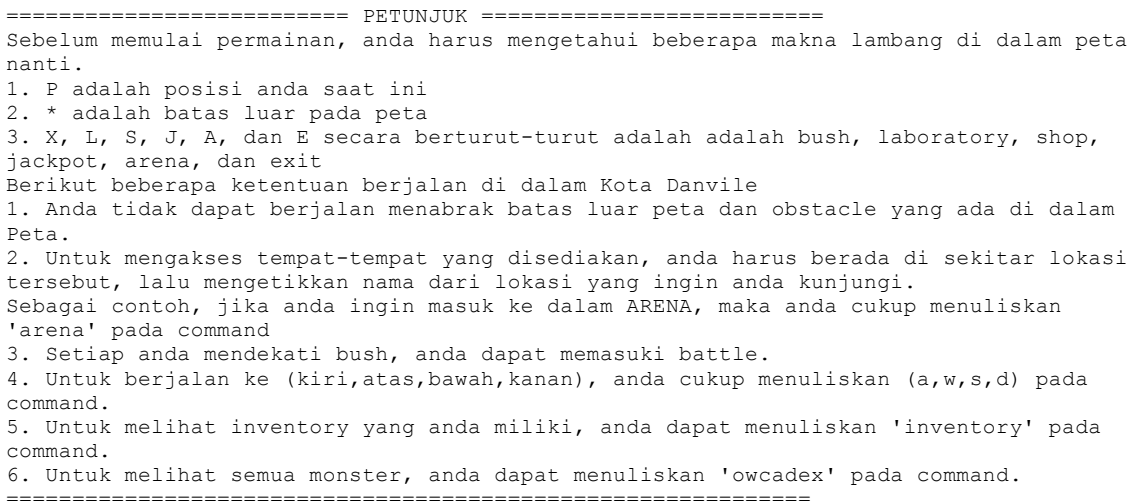
```
Command:
```

```
>>> 4
```

```
Loading...
```

# GAME DIMULAI

```
  .--..--..--..--..--..
.' \  (`. _  ( _  _  /
```



```
Masukkan command: d
Agent akan pindah ke kanan!

* * * * *
*   P   X       X   *
*       X   S   X   E *
*               X   *
*           X       X   *
*       X           *
*       X           X   *
*               X       *
```

```

*   L   X X X X   A   *
*           X           *
*           J   X       *
* * * * * * * * * *
P saat ini berada di posisi (0,1)

```

Masukkan command: s

Agent akan pindah ke bawah!

```

* * * * * * * * * *
*           X           X           *
*   P   X   S   X   E   *
*           X           *
*           X           X           *
*           X           *
*           X           X           *
*           X           *
*   L   X X X X   A   *
*           X           *
*           J   X       *
* * * * * * * * * *
P saat ini berada di posisi (1,1)

```

Masukkan command: a

Agent akan pindah ke kiri!

```

* * * * * * * * * *
*           X           X           *
*   P   X   S   X   E   *
*           X           *
*           X           X           *
*           X           *
*           X           X           *
*           X           *
*   L   X X X X   A   *
*           X           *
*           J   X       *
* * * * * * * * * *
P saat ini berada di posisi (1,0)

```

Masukkan command: w

Agent akan pindah ke atas!

```

* * * * * * * * * *
*   P   X           X           *
*           X   S   X   E   *
*           X           *
*           X           X           *
*           X           *
*           X           X           *
*           X           *
*   L   X X X X   A   *
*           X           *
*           J   X       *
* * * * * * * * * *
P saat ini berada di posisi (0,0)

```

Masukkan command: d

Agent tidak bisa pindah karena terdapat Obstacle!

```

* * * * * * * * * *
*   P X           X           *
*           X   S   X   E   *

```



```

*           X      *
*      X      X      *
*      X      *      *
*      X      X      *
*           X      *
*      L  X X X X  A  *
*           X      *
*           J  X      *
* * * * *
P saat ini berada di posisi (0,2)

```

Masukkan command: w

Agent tidak bisa pindah karena terdapat Obstacle!

```

* * * * *
*      P X      X      *
*           X  S  X  E  *
*           X      *
*      X      X      *
*      X      *      *
*      X      X      *
*           X      *
*      L  X X X X  A  *
*           X      *
*           J  X      *
* * * * *
P saat ini berada di posisi (0,2)

```

```

* * * * *
*           X      X      *
*           X  S  X  E  *
*           X      *
*      X      X  P  *
*      X      *      *
*      X      X      *
*           X      *
*      L  X X X X  A  *
*           X      *
*           J  X      *
* * * * *
P saat ini berada di posisi (3,9)
Masukkan command: exit

```

Agent tidak berada di area Exit!

```

* * * * *
*           X      X      *
*           X  S  X  E  *
*           X      *
*      X      X  P  *
*      X      *      *
*      X      X      *
*           X      *
*      L  X X X X  A  *
*           X      *
*           J  X      *
* * * * *
P saat ini berada di posisi (3,9)
Masukkan command: w

```

Agent akan pindah ke atas!

```

* * * * *
*           X      X      *
*           X  S  X  E  *
*           X      P  *

```

```
*      X      X      *
*      X      *
*      X      X      *
*      X      *
*      L      X X X X      A      *
*      X      *
*      J      X      *
* * * * * * * * * *
P saat ini berada di posisi (2,9)
Masukkan command: exit
Apakah benar anda akan keluar? (y/n):y
```

## Desain Kamus Data

### 1. F00-Random Number Generator

```
function LCG(seed : integer) → integer

KAMUS LOKAL

{variables}
    a : integer
    c : integer
    m : integer

function RandomNumber(range : array of integer) → integer

KAMUS LOKAL

{variables}
    a : integer
    c : integer
    m : integer
    x_n : integer
```

### 2. F01-Register

```
function valid
    (input username: string) → (output cek: bool)

KAMUS LOKAL

{variables}
    username: string
    kapital: array [1..26] of integer
    kecil: array [1..26] of integer
    angka: array [1..10] of integer
    underscore: int
    strip : int
    cek: bool

function register
    (input user: arr of User, input monster: arr of Monster, input
    monster_inventory: arr of Monster_Inventory) → (output about_user: User)

KAMUS LOKAL

{types}
    type User : < id: integer;
                  username: string;
                  password: string;
                  role: string;
                  oc: integer>
    type Monster: < id: integer;
                   type: string;
                   atk_power: int;
                   def_power: integer;
                   hp: integer>
    type Monster_Inventory : < user_id: integer;
                              monster_id: integer;
                              level: integer>

{variables}
    username: string
    password: string
```

```

cek: bool
i: integer
user: array of User
monster: array of Monster
monster_inventory: array of Monster_Inventory
monster_pilihan: integer
id_baru: integer
masuk_database_user: User
masuk_database_monster: Monster
about_user: User

```

### 3. F02-Login

```

function login (input user: array of User) -> return_login

```

**KAMUS LOKAL**

**{types}**

```

    type User : < id: integer;
                  username: string;
                  password: string;
                  role: string;
                  oc: integer>

    type Return_Login: < berhasil_login: bool;
                       id: integer;
                       username: string;
                       password: string;
                       role: string;
                       oc: integer >

```

**{variables}**

```

username: string
password: string
user: array of User
berhasil_login: bool
cek: bool
i: int
idx_row: int
return_login: Return_Login

```

### 4. F03-Logout

```

function logout() -> (output:boolean)

```

**KAMUS LOKAL**

**{variables}**

```

Pilihan_user : character

```

### 5. F04-Menu & Help

```

function help_sudah_login_admin (input user: array of User, input monster: array of
Monster, input monster_shop: array of Monster_Shop, input monster_inventory: array of
Monster_Inventory, input item_shop: array of Item_Shop, input item_inventory: array
of Item_Inventory) -> (output: exit: bool)

```

**KAMUS LOKAL**

{types}

```
type User : < id: integer;
               username: string;
               password: string;
               role: string;
               oc: integer>
type Monster: < id: integer;
                 type: string;
                 atk_power: int;
                 def_power: integer;
                 hp: integer>
type Monster_Shop: < monster_id: integer;
                    stock: integer;
                    price: integer>
type Monster_Inventory : < user_id: integer;
                           monster_id: integer;
                           level: integer>
type Item_Shop: < type: string;
                 stock: integer;
                 price: integer >
type Item_Inventory: < user_id: integer;
                       type: string;
                       quantity: integer >
type Hasil_Fungsi: < monster_shop: array of Monster_Shop;
                     item_shop: array of Item_Shop;
                     monster: array of Monster;
                     item_inventory : array of Item_Inventory>
```

{variables}

```
user: array of User
monster: array of Monster
monster_shop: array of Monster_shop
monster_inventory: array of Monster_Inventory
item_shop: array of Item_Shop
item_inventory: array of Item_Inventory
hasil_fungsi: Hasil_Fungsi
exit: bool
input_user: integer
validasi_logout: bool
validasi_exit: bool
```

{functions}

```
procedure list_help_admin()

function logout() -> (output:boolean)

function shop_management (input monster_shop: array of Monster_Shop, input
item_shop: array of Item_Shop, input monster: array of Monster, input item_inventory:
array of Item_Inventory) -> (output: functions_value: Hasil_Fungsi)

function monster_management (input monster: array of Monster) -> (output:
monster: array of Monster)

function save_data(nama_folder : string , user : array of User, userMonster :
array of any , monster_shop : array of Monster Shop, monster_inventory : array of
Monster_Inventory, item_shop : array of Item_Shop, item_inventory : array of
Item_Inventory) -> string

procedure exit()
```

```
function help_sudah_login_agent(input user: array of User, input user_id: int, input
username : string, input peta : array of string, input monster_shop: array of
Monster_Shop, input item_shop: array of Item_Shop, input monster: array of Monster,
```

```

input item_inventory: array of
Item_Inventory, input coin: integer) -> (output exit -> bool)

KAMUS LOKAL

{types}
    type User : < id: integer;
                    username: string;
                    password: string;
                    role: string;
                    oc: integer>
    type Monster: < id: integer;
                    type: string;
                    atk_power: int;
                    def_power: integer;
                    hp: integer>
    type Monster_Shop: < monster_id: integer;
                        stock: integer;
                        price: integer>
    type Item_Shop: < type: string;
                    stock: integer;
                    price: integer >
    type Monster_Inventory : < user_id: integer;
                              monster_id: integer;
                              level: integer>

    type Item_Inventory: < user_id: integer;
                          type: string;
                          quantity: integer >
    type Hasil_Fungsi: < monster_shop: array of Monster_Shop;
                        item_shop: array of Item_Shop;
                        monster: array of Monster;
                        item_inventory : array of Item_Inventory>

{variables}
    user: array of User
    monster: array of Monster
    monster_shop: array of Monster_shop
    item_shop: array of Item_Shop
    monster_inventory: array of Monster_Inventory
    item_inventory: array of Item_Inventory
    hasil_fungsi: Hasil_Fungsi
    exit: bool
    input_user: integer
    validasi_logout: bool
    validasi_exit: bool

{functions}

    procedure list_help_agent()

    function logout() -> (output:boolean)

    function shop_management (input monster_shop: array of Monster_Shop, input
item_shop: array of Item_Shop, input monster: array of Monster, input item_inventory:
array of Item_Inventory) -> (output: functions_value: Hasil_Fungsi)

    function monster_management (input monster: array of Monster) -> (output:
monster: array of Monster)

    function save_data(nama_folder : string , user : array of User, userMonster :
array of any , monster_shop : array of Monster_Shop, monster_inventory : array of
Monster_Inventory, item_shop : array of Item_Shop, item_inventory : array of
Item_Inventory) -> string

    procedure exit()

```

## 6. F05-Monster

**function** lvlStat  
(input baseStat: BasicStat, lvl: integer) → (output stat: BattleStat)

**KAMUS LOKAL**

{types}  
    type BasicStat : < atk : integer ;  
                            def : integer ;  
                            hp : integer >  
    type BattleStat : < atk : integer ;  
                            def : integer ;  
                            hp : integer ;  
                            hpmax : integer >

{variables}  
    baseStat: BasicStat  
    lvl: integer  
    stat: BattleStat

{functions/procedures}  
    function RealToInt (x: real) → integer  
    {mengubah input bertipe real menjadi integer}

**function** TampilMonsterDetail(input monster: Monster) → (output tampilan: string)

**KAMUS LOKAL**

{types}  
    type BasicStat : < atk : integer ;  
                            def : integer ;  
                            hp : integer >  
    type BattleStat : < atk : integer ;  
                            def : integer ;  
                            hp : integer ;  
                            hpmax : integer >  
    type Monster : < id : integer ;  
                            name : string ;  
                            lvl : integer ;  
                            baseStat : BasicStat >

{variables}  
    monster: Monster  
    tampilan, nama: string  
    id, lvl: integer  
    stat: BattleStat

{functions/procedures}  
    function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat  
    {menghasilkan status monster berdasarkan status base dan levelnya}

**function** TampilanOwcaDex  
(input monsters: array of array of string) → (output tampilan: string)

**KAMUS LOKAL**

{variables}  
    monsters: array of array of string  
    tampilan: string  
    i: integer

{functions/procedures}  
    function len (x: list) → integer  
    {menghitung panjang list}  
    function IntToStr (x: integer) → string  
    {mengubah input bertipe integer menjadi string}

**procedure** owcaDex(input monsters: array of array of string)

#### KAMUS LOKAL

##### {types}

```
type BasicStat : < atk : integer ;  
                  def : integer ;  
                  hp : integer >  
type Monster : < id : integer ;  
                  name : string ;  
                  lvl : integer ;  
                  baseStat : BasicStat >
```

##### {variables}

```
monsters: array of array of string  
tampilan, tampilMonster, name: string  
idValid, lvlValid: array of string  
id, lvl: integer  
baseStat: BasicStat  
monster: Monster
```

##### {functions/procedures}

```
function TampilanOwcaDex (monsters: array of array of string) → string  
{menghasilkan tampilan utama owcadex}  
function IntToStr (x: integer) → string  
{mengubah input bertipe integer menjadi string}  
function StrToInt (x: string) → integer  
{mengubah input bertipe string menjadi integer}  
function len (x: list) → integer  
{menghitung panjang list}  
function TampilMonsterDetail (monster: Monster) → string  
{menghasilkan tampilan detail mengenai monster yang dipilih}  
function inputValid (text: string, line: string, pilihanValid: array of  
string,  
                    typeValid: string) → (integer / string)  
{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
```

## 7. F06-Potion

#### function PeningkatanEfek

```
(input opsi: integer, potions: array of BattlePotion, stat: BattleStat) →  
(output i, efek: integer, tampilan, potions[i-1].type: string)
```

#### KAMUS LOKAL

##### {types}

```
type BattlePotion : < type : string ;  
                    qty : integer ;  
                    uses : integer >  
type BattleStat : < atk : integer ;  
                   def : integer ;  
                   hp : integer ;  
                   hpmax : integer >
```

##### {variables}

```
opsi, i, NPotion, statAsal, efek: integer  
potions: array of BattlePotion  
stat: BattleStat  
tampilan: string
```

##### {functions/procedures}

```
function len (x: list) → integer  
{menghitung panjang list}  
function RealToInt (x: real) → integer  
{mengubah input bertipe real menjadi integer}
```

```
function GunakanPotionLain() → boolean
```



#### KAMUS LOKAL

##### {variables}

pilihanValid: array of character  
lineInput, pilih: string

##### {functions/procedures}

function inputValid (text: string, line: string, pilihanValid: array of  
string,  
typeValid: string) → (integer / string)

##### function TampilanPotionUse

(input potions: array of BattlePotion) →  
(output tampilan: string, number: integer)

#### KAMUS LOKAL

##### {types}

type BattlePotion : < type : string ;  
qty : integer ;  
uses : integer >

##### {variables}

potions: array of BattlePotion  
tampilan, namaPotion: string  
number, NPotion: integer

##### {functions/procedures}

function len (x: list) → integer  
{menghitung panjang list}  
function IntToStr (x: integer) → string  
{mengubah input bertipe integer menjadi string}

##### function PotionUse

(input stat: BattleStat, potions: array of BattlePotion) →  
(output stat: BattleStat, potions: array of BattlePotion, used: integer)

#### KAMUS LOKAL

##### {types}

type BattleStat : < atk : integer ;  
def : integer ;  
hp : integer ;  
hpmax : integer >  
type BattlePotion : < type : string ;  
qty : integer ;  
uses : integer >

##### {variables}

stat: BattleStat  
potions: array of BattlePotion  
used, cancel, opsi: integer  
tampilan, textInput, lineInput, pilih: string  
optionsValid, pilihanValid: array of string

##### {functions/procedures}

##### function TampilanPotionUse

(input potions: array of BattlePotion) →  
(output tampilan: string, number: integer)

{menghasilkan tampilan utama potion use, berisi pilihan potion yang dimiliki,  
serta mengembalikan nomor pilihan cancel}

##### function PeningkatanEfek

(input opsi: integer, potions: array of BattlePotion, stat: BattleStat)

→

(output i, statID, efek: integer, tampilan: string)

{menghasilkan letak pilihan yang sebenarnya, karena potion dengan qty = 0

```

tidak      ditampilkan, besar peningkatan efek, tampilan efek jika potion digunakan,
serta      type potion yang ditingkatkan}
          function GunakanPotionLain() → boolean
          {menghasilkan statement apakah user ingin menggunakan potion lain atau tidak}
          function IntToStr (x: integer) → string
          {mengubah input bertipe integer menjadi string}

```

## 8. F07-Inventory

```

function ItemInventory
  (input user_id: string, items_inv: array of array of string,
   userPotions: array of Potion) →
  (output items_inv: array of array of string)

```

### KAMUS LOKAL

```

{types}
  type Potion : < type : string ;
                  qty : integer >

{variables}
  user_id, qty: string
  items_inv: array of array of string
  userPotions: array of Potion
  m, i, j: integer

```

```

{functions/procedures}
  function len (x: list) → integer
  {menghitung panjang list}
  function IntToStr (x: integer) → string
  {mengubah input bertipe integer menjadi string}

```

```

function UserMonsters
  (input user_id: string, Monsters: array of array of string,
   monsters_inv: array of array of string) →
  (output userMonsters: array of Monster)

```

### KAMUS LOKAL

```

{types}
  type Monster : < id : integer ;
                  name : string ;
                  lvl : integer ;
                  baseStat : BasicStat >
  type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >

```

```

{variables}
  user_id: string
  monsters, monsters_inv: array of array of string
  userMonsters: array of Monster
  monster: Monster
  i: integer

```

```

{functions/procedures}
  function len (x: list) → integer
  {menghitung panjang list}
  function StrToInt (x: string) → integer
  {mengubah input bertipe string menjadi integer}

```

```

function UserPotions

```

(input user\_id: string, items\_inv: array of array of string) →  
(output userPotions: array of Potion)

**KAMUS LOKAL**

**{types}**

type Potion : < type : string ;  
                  qty : integer >

**{variables}**

user\_id: string  
items\_inv: array of array of string  
userPotions: array of Potion  
potion: Potion  
i: integer

**{functions/procedures}**

function len (x: list) → integer  
{menghitung panjang list}  
function StrToInt (x: string) → integer  
{mengubah input bertipe string menjadi integer}

function KembaliKeInventory() → boolean

**KAMUS LOKAL**

**{variables}**

pilihanValid: array of character  
textInput: string  
pilih: integer

**{functions/procedures}**

function inputValid (text: string, line: string, pilihanValid: array of  
string,  
  typeValid: string) → (integer / string)

function TampilPotionDetail

(input userPotions: array of Potion, pilih: integer) → (output tampilan:  
string)

**KAMUS LOKAL**

**{types}**

type Potion : < type : string ;  
                  qty : integer >

**{variables}**

userPotions: array of Potion  
tampilan, type, nama: string  
potion: Potion  
pilih, i, qty: integer

**{functions/procedures}**

function len (x: list) → integer  
{menghitung panjang list}  
function IntToStr (x: integer) → string  
{mengubah input bertipe integer menjadi string}

function TampilanInv

(input userMonsters: array of Monster, userPotions: array of Potion,  
          user\_id, username: string, userOC: integer) →  
(output tampilan: string, number: integer)

**KAMUS LOKAL**

**{types}**

```

type Monster : < id : integer ;
                name : string ;
                lvl : integer ;
                baseStat : BasicStat >
type Potion : < type : string ;
                qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type BattleStat : < atk : integer ;
                   def : integer ;
                   hp : integer ;
                   hpmax : integer >

```

#### {variables}

```

userMonsters: array of Monster
userPotions: array of Potion
stat: BattleStat
user_id, username, name, tampilan: string
userOC, NMonster, Npotion, i, number, lvl: integer

```

#### {functions/procedures}

```

function len (x: list) → integer
{menghitung panjang list}
function IntToStr (x: integer) → string
{mengubah input bertipe integer menjadi string}
function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
{menghasilkan status monster berdasarkan status base dan levelnya}

```

#### procedure TampilInventoryUser

```

(input userMonsters: array of Monster, userPotions: array of Potion, user_id:
string, username: string, userOC: integer)

```

#### KAMUS LOKAL

#### {types}

```

type Monster : < id : integer ;
                name : string ;
                lvl : integer ;
                baseStat : BasicStat >
type Potion : < type : string ;
                qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >

```

#### {variables}

```

userMonsters : array of Monster
userPotions : array of Potion
userOC, pilih, number, NMonster : integer
rangeMonster, rangePotion : range
pilihanValid : array of string
tampilan : string

```

#### {functions/procedures}

```

function TampilanInv
(input userMonsters: array of monster, userPotions: array of Potion,
User_id, username: string, userOC: integer) →
(output tampilan: string, number: integer)
{menghasilkan tampilan utama inventory user dan batas nomor pilihan}
function TampilMonsterDetail (monster: Monster) → string
{menghasilkan tampilan detail mengenai monster yang dipilih}
function TampilPotionDetail (userPotions: array of Potion, pilih: integer) →
string
{menghasilkan tampilan detail mengenai potion yang dipilih, userPotion dengan
qty = 0 tidak akan ditampilkan}
function inputValid (text: string, line: string, pilihanValid: array of
string,
typeValid: string) → (integer / string)

```

```

{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
function len (x: list) → integer
{menghitung panjang list}
function KembaliKeInventory () → boolean
{menghasilkan statement apakah user ingin kembali ke inventory atau keluar}
function IntToStr (x: integer) → string
{mengubah input bertipe integer menjadi string}

```

## 9. F08-Battle

```

function RandomMonster
  (input monsters: array of array of string, lvl: integer) →
  (output monster: Monster)

```

### KAMUS LOKAL

```

{types}
  type BasicStat : < atk : integer ;
                    def : integer ;
                    hp : integer >
  type Monster : < id : integer ;
                  name : string ;
                  lvl : integer ;
                  baseStat : BasicStat >

```

```

{variables}
  monsters : array of array of string
  rndmonster : array of string
  monster : Monster

```

```

{functions/procedures}
  function StrToInt (x: string) → integer
  {mengubah masukan bertipe string menjadi integer}

```

```

procedure KemunculanMonster
  (input monster: Monster, character: string)

```

### KAMUS LOKAL

```

{types}
  type BasicStat : < atk : integer ;
                    def : integer ;
                    hp : integer >
  type BattleStat : < atk : integer ;
                    def : integer ;
                    hp : integer ;
                    hpmax : integer >
  type Monster : < id : integer ;
                  name : string ;
                  lvl : integer ;
                  baseStat : BasicStat >

```

```

{variables}
  monster : Monster
  character : string {username atau "enemy"}
  stat : BattleStat

```

```

{functions/procedures}
  function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
  {menghasilkan status monster berdasarkan status base dan levelnya}

```

```

function PilihUserMonster
  (input userMonsters: array of Monster) → (output monster: Monster)

```

#### KAMUS LOKAL

##### {types}

```
type BasicStat : < atk : integer ;  
                  def : integer ;  
                  hp : integer >  
type Monster : < id : integer ;  
                 name : string ;  
                 lvl : integer ;  
                 baseStat : BasicStat >
```

##### {variables}

```
userMonsters : array of Monster  
monster : Monster  
Nmonster : integer  
pilihanValid : array of string  
tampilan : string
```

##### {functions/procedures}

```
function IntToStr (x: integer) → string  
{mengubah input bertipe integer menjadi string}  
function inputValid (text: string, line: string, pilihanValid: array of  
string,  
                    typeValid: string) → (integer / string)  
{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}  
function len (x: list) → integer  
{menghitung panjang list}
```

#### function Skip

```
(input statMonster, statEnemy: BattleStat, turn: integer) →  
(output endedStatus: string, damageGiven, damageReceived: integer)
```

#### KAMUS LOKAL

##### {types}

```
type BattleStat : < atk : integer ;  
                   def : integer ;  
                   hp : integer ;  
                   hpmax : integer >
```

##### {variables}

```
statMonster, statEnemy : BattleStat  
turn, damageGiven, damageReceived, ATKPower, DEFPower : integer  
percent, ATK, reduced, damage, HPTarget : integer  
endedStatus : string
```

##### {functions/procedures}

```
function RealToInt (x: real) → integer  
{mengubah input bertipe real menjadi integer}  
function random_number ((x,y): range) → integer  
{menghasilkan bilangan integer random yang berada pada range (x,y)}
```

#### function Attack

```
(input statAtk, statDef: BattleStat, nameAtk, nameDef, chrAtk, chrDef:  
string) →  
(output Hptarget, damage: integer)
```

#### KAMUS LOKAL

##### {types}

```
type BattleStat : < atk : integer ;  
                   def : integer ;  
                   hp : integer ;  
                   hpmax : integer >
```

##### {variables}

```

{Atk = penyerang, Def = bertahan}
statAtk, statDef : BattleStat
nameAtk, nameDef, chrAtk, chrDef : string
ATKPower, DEFPower, percent, ATK, reduced, result, HPtarget : integer
percentage : string

```

#### **{functions/procedures}**

```

function random number ((x,y): range) → integer
{menghasilkan bilangan integer random yang berada pada range (x,y)}
function RealToInt (x: real) → integer
{mengubah input bertipe real menjadi integer}

```

#### **function CanUsePotion**

```

(input battlePotions: array of BattlePotion, monsterStat: BattleStat) →
boolean

```

#### **KAMUS LOKAL**

##### **{types}**

```

type BattlePotion : < type : string ;
                    qty : integer ;
                    uses : integer >
type BattleStat : < atk : integer ;
                   def : integer ;
                   hp : integer ;
                   hpmax : integer >

```

##### **{variables}**

```

battlePotions : array of BattlePotion
monsterStat : BattleStat
can : boolean
i : integer

```

##### **{functions/procedures}**

```

function len (x: list) → integer
{menghitung panjang list}

```

#### **function BattleTurns**

```

(input monster,enemy: Monster,userPotions: array of Potion, username: string)
→
(output userPotions: array of Potion, totalDamageReceived, totalDamageGiven:
integer, endedStatus: string)

```

#### **KAMUS LOKAL**

##### **{types}**

```

type Monster : < id : integer ;
                 name : string ;
                 lvl : integer ;
                 baseStat : BasicStat >
type Potion : < type : string ;
               qty : integer >
type BattlePotion : < type : string ;
                    qty : integer ;
                    uses : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type BattleStat : < atk : integer ;
                   def : integer ;
                   hp : integer ;
                   hpmax : integer >

```

##### **{variables}**

```

monster, enemy : Monster
monsterStat, enemyStat : BattleStat
userPotions : array of Potion
battlePotions : array of BattlePotion

```

```

totalDamageReceived, totalDamageGiven, damageReceived, damageGiven : integer
pilih, turn : integer
endedStatus, username, textPilihan, textInput, lineInput : string
opsi : character
pilihanValid, opsiValid : array of integer
ended : boolean

```

#### **{functions/procedures}**

```

function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
{menghasilkan status monster berdasarkan status base dan levelnya}
function inputValid (text: string, line: string, pilihanValid: array of
string,
                                typeValid: string) →(integer / string)
{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
function Attack
    (input statAtk, statDef: BattleStat,
     nameAtk, nameDef, chrAtk, chrDef: string) →
    (output HPTarget, damage: integer)
{menghitung damage dan menampilkan detail serangan monster}
function CanUsePotion (battlePotions: array of BattlePotion,
                        monsterStat: BattleStat) → boolean
{menentukan apakah user dapat menggunakan potion saat battle atau tidak}
function len (x: list) → integer
{menghitung panjang list}

```

#### **function Battle**

```

(input monsters: array of array of string, userMonsters: array of Monster,
 userPotions: array of Potion, username: string) →
(output userPotions: array of Potion, bonus: integer)

```

#### **KAMUS LOKAL**

#### **{types}**

```

type Monster : < id : integer ;
                  name : string ;
                  lvl : integer ;
                  baseStat : basicStat >
type Potion : < type : string ;
                qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >

```

#### **{variables}**

```

monsters : array of array of string
userMonsters : array of Monster
monster, enemy : Monster
userPotions : array of Potion
username, endedStatus : string
bonus, enemyLvl, damageReceived, damageGiven: integer

```

#### **{functions/procedures}**

```

function RandomMonster (monsters: array of array of string, lvl: integer) →
    Monster
{men-generate random monster dengan level yang sudah ditentukan}
function random number ((x,y): range) → integer
{menghasilkan bilangan integer random yang berada pada range (x,y)}
procedure KemunculanMonster (input monster: Monster, character: string)
{menampilkan intro dan detail monster saat di awal battle}
function PilihUserMonster (userMonsters: array of Monster) → Monster
{menghasilkan data monster pilihan user yang telah dimiliki user}
function BattleTurns
    (input Monster, enemy: Monster, userPotions: array of Potion,
 username:
                                string) →
    (output userPotions: array of Potion, totalDamageReceived,
     totalDamageGiven: integer, endedStatus: string)
{menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan
potion, kabur, skip battle, dan menampilkan detail status hingga battle}

```



berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta diterima, dan status battle}

## 10. F09-Arena

### function CekBonus

(input endedStatus: string, stage: integer) →  
(output bonus : integer)

#### KAMUS LOKAL

##### {variables}

endedStatus : string  
stage : integer  
bonus : integer

##### {functions/procedures}

function RealToInt (x: real) ← integer  
{mengubah input bertipe integer menjadi string}

### function Arena

(input userMonsters: array of Monster, userPotions: array of Potion,  
username: string, monsters: array of array of string) →  
(output userPotions: array of Potion, totalBonus: integer)

#### KAMUS LOKAL

##### {types}

type Monster : < id : integer ;  
name : string ;  
lvl : integer ;  
baseStat : BasicStat >  
type Potion : < type : string ;  
qty : integer >  
type BasicStat : < atk : integer ;  
def : integer ;  
hp : integer >

##### {variables}

userMonsters : array of Monster  
userPotions : array of Potion  
monsters : array of array of string {read\_csv("monster.csv")}  
bonus, totalBonus, stage, clear : integer  
damageReceived, damageGiven, totalDamageReceived, totalDamageGiven : integer  
monster, enemy : Monster

##### {functions/procedures}

function CekBonus (endedStatus: string, stage: integer) → integer  
{menghitung bonus berdasarkan status kemenangan dan stage arena}  
function RandomMonster (monsters: array of array of string, lvl: integer) →  
Monster  
{men-generate random monster dengan level yang sudah ditentukan}  
procedure KemunculanMonster (input monster: Monster, character: string)  
{menampilkan intro dan detail monster saat di awal battle}  
function PilihUserMonster (userMonsters: array of Monster) → Monster  
{menghasilkan data monster pilihan user yang telah dimiliki user}  
function BattleTurns  
(input monster, enemy: Monster, userPotions: array of Potion,  
username: string) →  
(output userPotions: array of Potion, totalDamageReceived,  
totalDamageGiven: integer, endedStatus: string)  
{menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan  
potion, kabur, skip battle, dan menampilkan detail status hingga battle  
berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta  
diterima, dan status battle}

## 11. F10-Shop & Currency

```
procedure shop
  (input userid:String , monster_shop:array of integer , item_shop:list,
   monster:array of any, monster_inventory:array of any , coin : integer) →
  (output monster_shop:array of any ,item_shop : array of any,monster ; array
   of any ,monster_inventory : array of any ,item_inventory : array of any ,
   coin : Integer)
```

### KAMUS LOKAL

#### {types}

```
type monster_shop : < monster_id : integer ;
                      stock : Integer ;
                      Price : integer ; >
type item_shop : < Item_type : string ;
                  qty : integer;
                  Price : integer ; >
type monster : < id : integer ;
                 name : string ;
                 lvl : integer ;
                 baseStat : basicStat >
```

#### {variables}

```
pilih : String
lihat_apa : String
item : String
id_monster_input : Any
cek_inventory : Boolean
found : Boolean
harga_item : Integer
harga_total : Integer
Monster_tambah : List of String
id_item_input : Integer
jumlah_item : Integer
item_type : String
```

## 12. F11-Laboratory

```
procedure laboratory
  ( input user_id : , user_coin: integer, monster: array of string,
    monster_inventory: array of any ) → ( output function_value : array )
```

### KAMUS LOKAL

#### {types}

```
type monsterStat : < id : integer ; type : string ; atk_power : integer ;
                    def_power : integer ; hp : integer >
type userMonster : < user_id : integer ; monster_id : integer ; level :
                    integer >
```

#### {variables}

```
list_index : array of integer
list_id_monster : array of string
enumerate : integer
selected_monster_index : string
cek_valid : boolean
selected_monster : string
idx_monster_di_inventory : integer
idx_monster_di_database : integer
next_level : integer
```

```

level : integer
cost : integer
upgrade : string
opt : string
function_value : array

```

```

function upgrade_cost
  ( input level : integer ) → ( output cost : integer )

```

**KAMUS LOKAL**

```

(variables)
  cost : integer

```

### 13. F12-Shop Management

```

function is_integer
  ( input n : string ) → ( output cek: bool)

```

**KAMUS LOKAL**

```

(variables)
  cek : bool
  char : str

```

```

function validasi_int ( input n: string, tipe: string) -> ( output n: string)

```

**KAMUS LOKAL**

```

(variables)

  n: string
  tipe: string

```

```

procedure shop_management
  ( input monster_shop : array of array of integer, item_shop : list, monster : array of
    any, item_inventory : array of array of (integer, string, integer)) →
  ( output monster_shop : array of array of integer, item_shop : array of array of
    (string,
      integer, integer), monster : array of array of (integer, string, integer,
      integer, integer), item_inventory : array of array of (integer, string,
      integer )

```

**KAMUS LOKAL**

```

(types)
  type battleStat : < atk : integer ;
                     def : integer ;
                     hp : integer ;
                     hpmax : integer >

```

```

(variables)
  list_item : list of string
  id_monster : list of integer
  aksi_user : string
  tipe_dilihat : string
  input_id_ditambah, stok_awal_ditambah, harga_ditambah : string
  data_ditambah : list of integer
  cek_valid : bool

```

**(functions)**

```

function is_integer (input n: string) -> (output cek: bool)

function validasi_int (input n: string, input tipe: string) -> (output n:
string)

```

## 14. F13-Monster Management

```

function is_integer
( input n : string ) → (cek: bool)

```

**KAMUS LOKAL**

```

{variables}
    cek : bool
    char : str

```

```

function validasi_int (input n: string, input tipe: string) -> (output n: string)

```

**KAMUS LOKAL**

```

{variables}

    n: string
    tipe: string

```

```

function monster_management
    (input monster : Array of string) →
    (output monster : Array of String)

```

**KAMUS LOKAL**

```

{types}
    type monster : < id : integer ;
                    name : string ;
                    lvl : integer ;
                    baseStat : basicStat >

```

```

{variables}
    input_fitur: string
    char: string
    input_nama_monster: string
    cek_file: bool
    found: bool
    input_atk_power: string
    Cek_Integer: bool
    input_def_power: string
    Cek_Rentang: bool
    input_HP: string
    id_monster: integer
    validasi_database: string
    monster_baru: list of any

```

```

{functions}
    function is_integer (input n: string) -> (output cek: bool)

    function validasi_int (input n: string, input tipe: string) -> (output n:
string)

```

## 15. F14-Load

```
procedure load ()  
  
KAMUS LOKAL  
  
{variables}  
    parser : argparse.ArgumentParser  
    nilai_fungsi : array of any  
    User, userMonster, monster_shop, monster_inventory, item_shop, item_inventory  
:    array of any
```

```
function parse_csv(file_path : string) -> array of array of string  
  
KAMUS LOKAL  
  
{variables}  
    file : SEQFILE of string
```

## 16. F15-Save

```
procedure save()  
  
KAMUS LOKAL  
  
{variables}  
    nama_folder : string
```

```
function save_data(nama_folder : string , user : array of User, userMonster : array of any , monster_shop : array of Monster_Shop, monster_inventory : array of Monster_Inventory, item_shop : array of Item_Shop, item_inventory : array of Item_Inventory) -> string  
  
KAMUS LOKAL  
  
{variables}  
    Parent_save, folder, data_default : string  
    files_di_folder : array of string  
    file : SEQFILE of string
```

## 17. F16-Exit

```
procedure exit()  
  
KAMUS LOKAL  
  
{variables}  
    pilihan_user : string
```

## 18. B04-JACKPOT!

```
function jackpot (input user_id: string, input monster_inventory: Monster_Inventory,
```

```
input koin: integer) -> (output functions_value: Functions_Value)
```

**KAMUS LOKAL**

**{type}**

```
    type Monster_Inventory : <  user_id: integer;
                                monster_id: integer;
                                level: integer >
```

**{variables}**

```
    user_id: string
    monster_inventory: Monster_Inventory
    functions_value: Functions_Value
    input_user: string
    item_satu,item_dua,item_tiga: integer
    koin,koin_tambahan,koin_didapat: integer
```

**{functions}**

```
    function LCG(seed : integer) → integer
    function RandomNumber(range : array of integer) → integer
```

## 19. B05-Peta Kota Danville

```
procedure peta_danville (input peta: array of string)
```

**KAMUS LOKAL**

**{variables}**

```
    peta: array of string
    i,j: integer
```

```
function bush (input peta: array of string) -> (output bush: array of string)
```

**KAMUS LOKAL**

**{variables}**

```
    peta: array of string
    bush: array of string
    i,j: integer
```

```
function komando (input user_id: string, input username: string, input monster_shop:
Monster_Shop, input item_shop: Item_Shop, input monster: Monster, input
monster_inventory: Monster_Inventory, input item_inventory: Item_Inventory, input
coin: integer) -> (output: function_value: Funtion_Value)
```

**KAMUS LOKAL**

**{types}**

```
    type Monster: < id: integer;
                    type: string;
                    atk_power: int;
                    def_power: integer;
                    hp: integer>
    type Monster_Shop: < monster_id: integer;
                       stock: integer;
                       price: integer>
    type Item_Shop: < type: string;
                    stock: integer;
                    price: integer >
    type Monster: < id: integer;
                    type: string;
                    atk_power: int;
                    def_power: integer;
                    hp: integer>
```

```

type Monster_Inventory : < user_id: integer;
                           monster_id: integer;
                           level: integer>

type Item_Inventory: <      user_id: integer;
                       type: string;
                       quantity: integer >
type Function_Value: < monster_shop: array of Monster_Shop;
                       item_shop: array of Item_Shop;
                       monster: array of Monster;
                       monster_inventory: array of Monster_Inventory;
                       item_inventory : array of Item_Inventory;
                       coin: integer >

{variables}
monster: Monster
monster_shop: Monster_Shop
item_shop: Item_Shop
monster: Monster
monster_inventory: Monster_Inventory
item_inventory: Item_Inventory
function_value: Function_Value
obstacle: array of string
command: string
idx_baris,idx_kolom: integer
function_value: Function_Value

{functions}

procedure peta_danville (input peta: array of string)

function bush (input peta: array of string) -> (output bush: array of string)

procedure shop
(input userid:String , monster_shop:array of integer , item_shop:list, monster:array
of any, monster_inventory:array of any , coin : integer) →
(output monster_shop:array of any ,item_shop : array of any,monster ; array of any
,monster_inventory : array of any ,item_inventory : array of any , coin : Integer)

function Battle
(input monsters: array of array of string, userMonsters: array of Monster,
userPotions: array of Potion, username: string) →
(output userPotions: array of Potion, bonus: integer)

procedure laboratory
( input user_id : , user_coin: integer, monster: array of string,
monster_inventory: array of any ) → ( output function_value : array )

function jackpot (input user_id: string, input monster_inventory:
Monster_Inventory, input koin: integer) -> (output functions_value: Functions_Value)

function Arena
(input userMonsters: array of Monster, userPotions: array of Potion,
username: string, monsters: array of array of string) →
(output userPotions: array of Potion, totalBonus: integer)

procedure exit()

```

## Desain Dekomposisi Algoritmik dan Fungsional Program

### 1. F00-Random Number Generator

```
function LCG(seed : integer) → integer
{ program akan menghasilkan bilangan acak
I.S. seed terdefinisi
F.S. menghasilkan bilangan acak }
```

```
function RandomNumber(range : array of integer) → integer
{ program akan menghasilkan bilangan acak dalam suatu rentang
I.S. range terdefinisi
F.S. Menghasilkan bilangan acak dalam rentang tertentu atau tanpa rentang }
```

### 2. F01-Register

```
function valid (input username: string) -> bool

{ Program akan mengecek apakah username yang diberikan sesuai spesifikasi yang
diharuskan, yaitu hanya terdiri dari huruf alfabet, angka, underscore, dan strip}
{I.S input variable username yang bertipe string}
{F.S validasi berupa boolean}
```

```
function register
(input user: arr of User, input monster: arr of Monster, input
monster_inventory: arr of Monster_Inventory) → (output about_user: User)

{ Program akan menghasilkan data pengguna baru setelah pengguna berhasil register}
{ I.S input database yang dibutuhkan}
{ F.S data tentang pengguna berupa type User}
```

### 3. F02-Login

```
procedure login
(input user : array of array) →
(output about_user : User)
{ Program dibuat untuk mengakses data pengguna yang akan digunakan untuk bermain.
Program meminta pengguna untuk memasukkan username dan password. Jika username dan
password cocok dengan data yang tersimpan, pengguna dianggap berhasil login.
I.S. Data pengguna dan input username serta password terdefinisi.
F.S. Status keberhasilan login dan informasi pengguna yang berhasil login. }
```



#### 4. F03-Logout

**Prosedure** Logout() → Boolean

{Program akan menjalankan fungsi logout yang hasilnya yaitu keluar dari program  
I.S. status login  
F.S. keluar dari program}

#### 5. F04-Menu & Help

**function** help\_sudah\_login\_admin (input user: array of User, input monster: array of Monster, input monster\_shop: array of Monster\_Shop, input monster\_inventory: array of Monster\_Inventory, input item\_shop: array of Item\_Shop, input item\_inventory: array of Item\_Inventory) → (output: exit: bool)

{ Program akan menampilkan menu yang dapat di akses oleh admin setelah login}  
{ I.S menginput semua database}  
{ F.S menghasilkan exit bertipe bool, menandakan jika user memilih untuk keluar}

**function** help\_sudah\_login\_agent(input user: array of User, input user\_id: int, input username : string, input peta : array of string, input monster\_shop: array of Monster\_Shop, input item\_shop: array of Item\_Shop, input monster: array of Monster, input monster\_inventory: array of Monster\_Inventory, input item\_inventory: array of Item\_Inventory, input coin: integer) → (output exit → bool)

{ Program akan menampilkan menu yang dapat di akses oleh agent setelah login}  
{ I.S menginput semua database}  
{ F.S menghasilkan exit bertipe bool, menandakan jika user memilih untuk keluar}

#### 6. F05-Monster

**function** lvlStat

(input baseStat: BasicStat, lvl: integer) → (output stat: BattleStat)

{menghasilkan status monster berdasarkan status base dan levelnya.  
I.S. status dasar dan level monster terdefinisi  
F.S. dihasilkan status monster berdasarkan levelnya}

**function** TampilMonsterDetail(input monster: Monster) → (output tampilan: string)

{menghasilkan tampilan detail mengenai monster yang dipilih.  
I.S. data monster terdefinisi  
F.S. dihasilkan tampilan/keterangan detail monster yang dipilih}

**function** TampilanOwcaDex

(input monsters: array of array of string) → (output tampilan: string)

{menghasilkan tampilan utama owcadex.  
I.S. seluruh data monster-monster terdefinisi  
F.S. dihasilkan tampilan utama untuk owcadex}

**procedure** owcaDex(input monsters: array of array of string)

{menampilkan owcadex, library dimana user dapat melihat monster apa saja yang ada dalam game, serta dapat melihat statusnya pada level 1-5.  
I.S. seluruh data monster-monster terdefinisi  
F.S. ditampilkan owcadex sesuai spesifikasi pada layar}

#### 7. F06-Potion

<p><b><u>function</u></b> PeningkatanEfek  <input <u="" integer,="" opsi:="" potions:=""/>array of BattlePotion, stat: BattleStat) →  (output i, efek: <u>integer</u>, tampilan, potions[i-1].type: string)  {menghasilkan letak pilihan yang sebenarnya, karena potion dengan qty = 0 tidak ditampilkan, besar peningkatan efek, tampilan efek jika potion digunakan, serta type potion yang ditingkatkan.  I.S. status sebelum peningkatan dan potion yang dipilih terdefinisi  F.S. status ditingkatkan berdasarkan potion yang dipilih}</p>
<p><b><u>function</u></b> GunakanPotionLain() → <u>boolean</u>  {menghasilkan statement apakah user akan menggunakan potion lain atau tidak.  I.S. user tidak jadi atau terkena batasan saat memilih potion  F.S. user memilih untuk menggunakan potion lain atau tidak}</p>
<p><b><u>function</u></b> TampilanPotionUse  <input <u="" potions:=""/>array of BattlePotion) →  (output tampilan: string, number: <u>integer</u>)  {menghasilkan tampilan utama potion use, berisi pilihan potion yang dimiliki, serta mengembalikan nomor pilihan cancel.  I.S. user memiliki potion untuk digunakan  F.S. dihasilkan tampilan utama potion use, pilihan terakhir adalah cancel}</p>
<p><b><u>function</u></b> PotionUse  <input <u="" battlestat,="" potions:="" stat:=""/>array of BattlePotion) →  (output stat: BattleStat, potions: <u>array of</u> BattlePotion, used: <u>integer</u>)  {ditampilkan jika user memilih untuk menggunakan potion dalam battle, menampilkan efek yang terjadi ketika menggunakan potion.  I.S. user memiliki potion yang dapat digunakan  F.S. status monster ditingkatkan jika memilih potion, status monster tetap jika tidak jadi menggunakan potion}</p>

## 8. F07-Inventory

<p><b><u>function</u></b> ItemInventory  <input <u="" items_inv:="" string,="" user_id:=""/>array of array of string,  userPotions: <u>array of</u> Potion) →  (output items_inv: <u>array of array of</u> string)  {mengembalikan nilai-nilai yang berubah pada userPotions ke dalam item_inventory.  I.S. seluruh parameter input terdefinisi  F.S. isi item_inventory diisi dengan userPotion yang berubah setelah battle, hanya quantity yg berubah}</p>
<p><b><u>function</u></b> UserMonsters  <input <u="" monsters:="" string,="" user_id:=""/>array of array of string,  monsters_inv: <u>array of array of</u> string) →  (output userMonsters: <u>array of</u> Monster)  {mengumpulkan data monster yang dimiliki user dan disimpan di userMonsters.  I.S. seluruh parameter input terdefinisi  F.S. data seluruh monster yang dimiliki user disimpan di userMonsters}</p>
<p><b><u>function</u></b> UserPotions  <input <u="" items_inv:="" string,="" user_id:=""/>array of array of string) →  (output userPotions: <u>array of</u> Potion)  {mengumpulkan data potion yang dimiliki user dan disimpan pada userPotions.  I.S. seluruh parameter input terdefinisi  F.S. data seluruh potion yang dimiliki user disimpan di userPotions}</p>
<p><b><u>function</u></b> KembaliKeInventory() → <u>boolean</u>  {menghasilkan statement apakah user ingin kembali ke inventory atau keluar.  I.S. user berada pada tampilan status monster/item  F.S. user memilih untuk kembali ke inventory atau langsung keluar, true jika kembali, false jika keluar}</p>
<p><b><u>function</u></b> TampilPotionDetail</p>

<p>(<u>input</u> userPotions: <u>array of</u> Potion, pilih: <u>integer</u>) → (<u>output</u> tampilan: string)</p> <p>{menghasilkan tampilan detail mengenai potion yang dipilih, userPotion dengan qty = 0 tidak akan ditampilkan.</p> <p>I.S. user memilih untuk melihat detail potion yang dimiliki</p> <p>F.S. dihasilkan tampilan detail potion}</p>
<p><b><u>function</u></b> TampilanInv</p> <p>(<u>input</u> userMonsters: <u>array of</u> Monster, userPotions: <u>array of</u> Potion, user_id, username: string, userOC: <u>integer</u>) →</p> <p>(<u>output</u> tampilan: string, number: <u>integer</u>)</p> <p>{menghasilkan tampilan utama inventory user dan batas nomor pilihan.</p> <p>I.S. seluruh parameter input terdefinisi</p> <p>F.S. dihasilkan tampilan utama inventory user berdasarkan monster dan item yang dimiliki, mengeluarkan nomor pilihan terakhir}</p>
<p><b><u>procedure</u></b> TampilInventoryUser</p> <p>(<u>input</u> userMonsters: <u>array of</u> Monster, userPotions: <u>array of</u> Potion, user_id: string, username: string, userOC: <u>integer</u>)</p> <p>{menampilkan isi inventory user.</p> <p>I.S. user memiliki monster dan item(potion)</p> <p>F.S. ditampilkan setiap monster dan item yang dimiliki user}</p>

## 9. F08-Battle

<p><b><u>function</u></b> RandomMonster</p> <p>(<u>input</u> monsters: <u>array of array of</u> string, lvl: <u>integer</u>) →</p> <p>(<u>output</u> monster: Monster)</p> <p>{men-generate random monster dengan level yang sudah ditentukan.</p> <p>I.S. monsters, data mentah dari csv yang diread, dan level yang sudah ditentukan terdefinisi</p> <p>F.S. dihasilkan monster random, digunakan sebagai lawan dalam pertarungan}</p>
<p><b><u>procedure</u></b> KemunculanMonster</p> <p>(<u>input</u> monster: Monster, character: string)</p> <p>{menampilkan intro dan detail monster saat di awal battle.</p> <p>I.S. monster dan karakter(user/enemy) terdefinisi</p> <p>F.S. ditampilkan intro dan status monster pada layar}</p>
<p><b><u>function</u></b> PilihUserMonster</p> <p>(<u>input</u> userMonsters: <u>array of</u> Monster) → (<u>output</u> monster: Monster)</p> <p>{menghasilkan data monster pilihan user yang telah dimiliki user.</p> <p>I.S. data monster milik user telah dipisahkan dari data utama</p> <p>F.S. dikeluarkan monster pilihan user untuk battle}</p>
<p><b><u>function</u></b> Skip</p> <p>(<u>input</u> statMonster, statEnemy: BattleStat, turn: integer) →</p> <p>(<u>output</u> endedStatus: string, damageGiven, damageReceived: <u>integer</u>)</p> <p>{menentukan apakah user akan menang atau kalah dengan kondisi monster selalu saling menyerang, tanpa menggunakan potion, serta menghitung total damage yang diberikan dan diterima.</p> <p>I.S. user memilih skip saat battle, seluruh parameter input terdefinisi</p> <p>F.S. dikeluarkan hasil battle secara otomatis}</p>
<p><b><u>function</u></b> Attack</p> <p>(<u>input</u> statAtk, statDef: BattleStat, nameAtk, nameDef, chrAtk, chrDef: string) →</p> <p>(<u>output</u> HPtarget, damage: <u>integer</u>)</p> <p>{menghitung damage dan menampilkan detail serangan monster.</p> <p>I.S. seluruh parameter input terdefinisi</p> <p>F.S. hp defender berkurang sesuai kalkulasi attack}</p>
<p><b><u>function</u></b> CanUsePotion</p> <p>(<u>input</u> battlePotions: <u>array of</u> BattlePotion, monsterStat: BattleStat) →</p>

<p><u>boolean</u></p> <p>{menentukan apakah user dapat menggunakan potion saat battle atau tidak. I.S. seluruh parameter input terdefinisi F.S. dihasilkan statement apakah user dapat memakai potion}</p>
<p><u>function</u> BattleTurns</p> <p>(<u>input</u> monster,enemy: Monster,userPotions: <u>array of</u> Potion, username: string) →</p> <p>→</p> <p>(<u>output</u> userPotions: <u>array of</u> Potion, totalDamageReceived, totalDamageGiven: <u>integer</u>, endedStatus: string)</p> <p>{menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan potion, kabur, skip battle, dan menampilkan detail status hingga battle berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta diterima, dan status battle. I.S. seluruh parameter input terdefinisi F.S. dikeluarkan damage serta status kemenangan hasil battle}</p>
<p><u>function</u> Battle</p> <p>(<u>input</u> monsters: <u>array of array of</u> string, userMonsters: <u>array of</u> Monster, userPotions: <u>array of</u> Potion, username: string) →</p> <p>(<u>output</u> userPotions: <u>array of</u> Potion, bonus: <u>integer</u>)</p> <p>{men-generate lawan, memilih monster, dan melakukan battle. I.S. user dengan role agent memasukkan command battle, seluruh parameter input terdefinisi F.S. dikeluarkan userPotion yang berubah karena penggunaan serta bonus berdasarkan status kemenangan battle}</p>

## 10. F09-Arena

<p><u>function</u> CekBonus</p> <p>(<u>input</u> endedStatus: string, stage: <u>integer</u>) →</p> <p>(<u>output</u> bonus : <u>integer</u>)</p> <p>{menghasilkan coin berdasarkan status kemenangan dan stage dalam arena. I.S. user telah menyelesaikan suatu stage F.S. dihasilkan bonus berdasarkan status kemenangan dan stage}</p>
<p><u>function</u> Arena</p> <p>(<u>input</u> userMonsters: <u>array of</u> Monster, userPotions: <u>array of</u> Potion, username: string, monsters: <u>array of array of</u> string) →</p> <p>(<u>output</u> userPotions: <u>array of</u> Potion, totalBonus: <u>integer</u>)</p> <p>{menjalankan arena: 5 stage, di setiap stage terdapat battle dengan lvl enemy = stage. I.S. user dengan role agent memilih command arena, seluruh parameter input terdefinisi F.S. mengembalikan userPotion yang berubah karena penggunaan serta bonus coin yang diperoleh}</p>

## 11. F10-Shop & Currency

<p><u>prosedure</u> shop</p> <p>(<u>input</u> userid:String , monster_shop:array of any , item_shop:list, monster:array of any, monster_inventory:array of any , coin : integer) →</p> <p>(<u>output</u> monster_shop:array of any ,item_shop : array of any,monster ; array of any ,monster_inventory : array of any ,item_inventory : array of any , coin : Integer)</p> <p>{ program akan menerima inputan data berupa array kemudian , program akan meminta user untuk memilih jenis yang ingin dibeli dan jumlah yang akan dibeli. Selanjutnya program akan melanjutkan dengan menampilkan koin dan akan mengembalikan value dari data yang telah diubah I.S.seluruh parameter input terdefinisi F.S.mengembalikan value data yang telah dibeli user }</p>
---

## 12. F11-Laboratory

```
function upgrade_cost
    (input level : integer) →
    (output cost : integer)
{ program ini menghitung biaya peningkatan level monster berdasarkan level saat ini.
  I.S. level monster yang ingin ditingkatkan terdefinisi
  F.S. mengembalikan biaya koin OC yang sesuai dengan level monster saat ini }
```

```
procedure laboratory
    ( input user_id : integer, user_coin : integer, monster : array of array,
      monster_inventory : array of array ) →
    ( output function_value : array of array )
{ program akan meng-upgrade level dari monster yang dimiliki oleh pengguna. Program
  ini akan melibatkan pemilihan monster, membayar biaya upgrade, dan meningkatkan level
  monster jika pengguna memiliki cukup koin.
  I.S. seluruh parameter input terdefinisi
  F.S. mengembalikan array yang berisi koin pengguna dan monster_inventory yang telah
  diperbarui }
```

## 13. F12-Shop Management

```
function is_integer(n:String)-> Boolean
{ program memvalidasi apakah bilangan integer
  I.S. input bilangan n
  F.S. validasi berupa boolean }
```

```
function validasi_int(n: string, tipe: string) -> string:
{ program memvalidasi apakah bilangan integer kemudian meminta input hingga input
  user berupa integer
  I.S. input bilangan n dan cek
  F.S. validasi berupa boolean dan nilai n }
```

```
procedure shop_management
    (input monster_shop: array of Monster_Shop, input item_shop: array of
      Item_Shop, input monster: array of Monster, input item_inventory: array of
      Item_Inventory)→ (output functions_value: Hasil_Fungsi)
{ Program untuk mengelola stok dan harga barang di toko. User dapat melakukan
  berbagai tindakan seperti menambah atau mengurangi stok monster dan item, serta
  mengubah harga jualnya. Program ini membantu dalam pengaturan inventaris toko agar
  sesuai dengan kebutuhan dan permintaan pengguna.
  I.S. Data stok dan harga barang di toko, serta inventaris item pengguna terdefinisi.
  F.S. Data stok dan harga barang di toko, serta inventaris item pengguna diperbarui
  sesuai dengan tindakan penambahan yang dilakukan dalam fungsi ini. }
```

## 14. F13-Monster Management

```
procedure monster_management
    (input monster : Array of string) →
    (output monster : Array of String)
{ Program mengubah data kumpulan monster sesuai dengan yang diubah oleh admin )
  I.S. file data monster yang terdefinisi
  F.S. Data monster yang telah diubah }
```

```
function is_integer(n:String)-> Boolean
```

```
{ program memvalidasi apakah bilangan integer  
I.S. input bilangan n  
F.S. validasi berupa boolean }
```

```
function validasi_int(n: string, tipe: string) -> string:  
{ program memvalidasi apakah bilangan integer kemudian meminta input hingga input  
user berupa integer  
I.S. input bilangan n dan cek  
F.S. validasi berupa boolean dan nilai n }
```

## 15. F14-Load

```
procedure load()  
{ program membaca file csv sesuai folder yang diinput oleh user dan mengeluarkan data  
dalam bentuk matriks  
I.S. file terdefinisi  
F.S. data di csv dikeluarkan }
```

```
function parse_csv(file_path : string) -> array of array of string  
{ membaca file csv dan mengubah data menjadi array of array of string }  
I.S. file terdefinisi  
F.S. data di dikeluarkan }
```

## 16. F15-Save

```
procedure save()  
{ program menyimpan file csv ke dalam folder yang diinput oleh user  
I.S. nama_folder, user, userMonster, monster_shop, monster_inventory, item_shop,  
item_inventory terdefinisi  
F.S. seluruh file berhasil tersimpan }
```

```
function save_data(nama_folder : string , user : array of any , userMonster : array  
of any , monster_shop : array of any , monster_inventory : array of any , item_shop :  
array of any , item_inventory : array of any) -> string  
{ menyimpan data yang diterima ke csv dalam sebuah folder yang telah/baru  
dibuat  
I.S. nama_folder, user, userMonster, monster_shop, monster_inventory, item_shop,  
item_inventory terdefinisi  
F.S. seluruh file tersimpan di dalam nama_folder }
```

## 17. F16-Exit

```
procedure exit()  
{ program untuk keluar dari permainan setelah menerima inputan apakah user ingin save  
file atau tidak  
I.S. program berjalan  
F.S. keluar dari program }
```

## 18. B04-JACKPOT!

```

function jackpot (input user_id: string, input monster_inventory: Monster_Inventory,
 koin: integer) -> (output functions_value: Functions_Value)
{ Menghasilkan hadiah yang dapat diterima oleh pengguna dengan mengundi}
{ I.S menginput user_id, database monster_inventory, dan banyak koin yang dimiliki
user}
{F.S menghasilkan database baru yang telah terupdate dengan hadiah yang diperoleh
user}

```

## 19. B05-Peta Kota Danville

```

procedure peta_danville (input peta: array of string)
{ Menampilkan peta kota danville}
{ I.S array peta yang secara berkala ter-update}
[ F.S. menampilkan peta kota danville pada layar}

```

```

function bush (input peta: array of string) -> (output bush: array of string)
{ Mendeteksi koordinat mana saja pada peta yang merupakan bush}
{ I.S menginput array peta}
{ F.S menghasilkan array yang berisi bush pada peta}

```

```

function komando (input user_id: string, input username: string, input monster_shop:
Monster_Shop, input item_shop: Item_Shop, input monster: Monster, input
monster_inventory: Monster_Inventory, input item_inventory: Item_Inventory, input
coin: integer) -> (output: function_value: Funtion_Value)
{ Menghubungkan pilihan user pada peta dengan fungsi-fungsi yang dapat diakses}
{ I.S menginput semua database yang dibutuhkan}
{ F.S menghasilkan type Function_Value yang merupakan hasil dari perjalanan user}

```

## Spesifikasi Fungsi-Fungsi dalam Program

### 1. F00-Random Number Generator

<b>function</b> LCG(seed : <u>integer</u> ) → <u>integer</u> { program akan menghasilkan bilangan acak I.S. seed terdefinisi F.S. menghasilkan bilangan acak }
<b>KAMUS LOKAL</b> a : <u>integer</u> c : <u>integer</u> m : <u>integer</u>
<b>ALGORITMA</b> a ← 1103515245 c ← 12345 m ← 2 <sup>31</sup> seed ← integer(os.getpid() + time.time() * 231) → (a * seed + c) <u>mod</u> m

<b>function</b> RandomNumber(range : <u>array of integer</u> ) → <u>integer</u> { program akan menghasilkan bilangan acak I.S. range terdefinisi F.S. Menghasilkan bilangan acak dalam rentang tertentu atau tanpa rentang }
<b>KAMUS LOKAL</b> a : <u>integer</u> c : <u>integer</u> m : <u>integer</u> x_n : <u>integer</u>
<b>ALGORITMA</b> a ← 1103515245 c ← 12345 m ← 2 <sup>31</sup> x_n ← (a * LCG() + c) <u>mod</u> m  <b>if</b> range = None <b>then</b> → x_n <b>else</b> → <u>integer</u> ((x_n / m) * (range[1] - range[0]) + range[0])

### 2. F01-Register

<b>function</b> valid ( <u>input</u> username: string) → bool  { Program akan mengecek apakah username yang diberikan sesuai spesifikasi yang diharuskan, yaitu hanya terdiri dari huruf alfabet, angka, underscore, dan strip} {I.S input variable username yang bertipe string} {F.S validasi berupa boolean}
<b>KAMUS LOKAL</b>  {types} type User : < id: integer; username: string; password: string; role: string; oc: integer>



```

type Monster: < id: integer;
                type: string;
                atk_power: int;
                def_power: integer;
                hp: integer>
type Monster_Inventory : < user_id: integer;
                           monster_id: integer;
                           level: integer>

{variables}
username: string
password: string
cek: bool
i: integer
user: array of User
monster: array of Monster
monster_inventory: array of Monster_Inventory
monster_pilihan: integer
id_baru: integer
masuk_database_user: User
masuk_database_monster: Monster
about_user: User

```

#### ALGORITMA

```

output("Silahkan mengisi username baru dan password yang aman :D")
output("Username hanya boleh berisi alfabet, angka, underscore, dan strip!")
repeat
    input(username)
    input(password)
    if valid(username) then
        cek <- false
        repeat
            for i <- 1 to length(user) do
                if username = user[i][2] then
                    output("Username ", username, " sudah terpakai, silahkan
                        gunakan username lain!")
                    input(username)
                    input(password)
                else
                    cek <- true
            until cek
        else
            output("Username hanya boleh berisi alfabet, angka, underscore, dan
                strip!")
    until cek

output("Silahkan pilih salah satu monster sebagai monster awalmu.")
for i <- 1 to 3 do
    output(i, ". ", monster[i][2])
input(monster_pilihan)
indeks_pilihan_monster <- [1, 2, 3]
repeat
    if monster_pilihan not in indeks_pilihan_monster then
        output("Masukkan tidak valid, ayo yang benar!")
        input(monster_pilihan)
until monster_pilihan in indeks_pilihan_monster

output("Selamat datang Agent ", username, ". Mari kita mengalahkan Dr. Asep
    Spakbor dengan ", monster[monster_pilihan][2], "!")

id_baru <- F00.random_number([100000, 1000000])
masuk_database_user <- [id_baru, username, password, "agent", 0]
User <- (masuk_database_user)
masuk_database_monster <- [id_baru, monster_pilihan, 1]
Monster_inventory <- (masuk_database_monster)
about_user <- [id_baru, username, password, "agent", 0]
return about_user

```

```
function RandomNumber(range : array of integer) → integer
{ program akan menghasilkan bilangan acak dalam suatu rentang
I.S. range terdefinisi
F.S. Menghasilkan bilangan acak dalam rentang tertentu atau tanpa rentang }
```

#### KAMUS LOKAL

```
user: string
kapital: array[1..26] of integer
kecil: array[1..26] of integer
angka: array[1..10] of integer
underscore: integer
strip: integer
cek: boolean
```

#### ALGORITMA

```
kapital <- [65, 66, 67, ..., 90] {ASCII untuk A-Z}
kecil <- [97, 98, 99, ..., 122] {ASCII untuk a-z}
angka <- [48, 49, 50, ..., 57] {ASCII untuk 0-9}
underscore <- 95 {ASCII untuk _}
strip <- 45 {ASCII untuk -}
cek <- true

for each i in user do
  if (ord(i) >< kapital) and (ord(i) >< kecil) and (ord(i) >< angka) and (ord(i)
    != underscore) and (ord(i) != strip) then
    cek <- false
    break
```

### 3. F02-Login

```
procedure login
  (input user : array of array) →
  (output about_user : User)
{ Program dibuat untuk mengakses data pengguna yang akan digunakan untuk bermain.
Program meminta pengguna untuk memasukkan username dan password. Jika username dan
password cocok dengan data yang tersimpan, pengguna dianggap berhasil login.
I.S. Data pengguna dan input username serta password terdefinisi.
F.S. Status keberhasilan login dan informasi pengguna yang berhasil login. }
```

#### KAMUS LOKAL

##### {types}

```
type User : < id: integer;
  username: string;
  password: string;
  role: string;
  oc: integer>

type Return_Login: < berhasil_login: bool;
  id: integer;
  username: string;
  password: string;
  role: string;
  oc: integer >
```

##### {variables}

```
username: string
password: string
user: array of User
berhasil_login: bool
cek: bool
i: int
idx_row: int
return_login: Return_Login
```

**ALGORITMA**

```
input(username)
input(password)

while (True) do
  if (username = "cancel") then
    berhasil_login <- False
    break
  else
    i <- 0
    cek <- False
    i traversal [0..len(user)-1]
    if (user[i][1] = username) then
      cek <- True
      idx_row <- i
      continue
    if (not cek) then
      output("Username tidak terdaftar!")
      input(username)
      input(password)
    else
      if (password != user[idx_row][2]) then
        output("Password salah!")
        input(username)
        input(password)
      else
        output("Selamat datang, ",user[idx_row][3]," ",username,"!")
        berhasil_login <- True
        break

if (berhasil_login = True) then
  -> (True, user[idx_row])
else
  -> (False, user)
```

#### 4. F03-Logout

**Prosedure** Logout() → **Boolean**

{Program akan menjalankan fungsi logout yang hasilnya yaitu keluar dari program  
I.S. status login  
F.S. keluar dari program}

**KAMUS LOKAL**

{variables}  
pilihan\_user : character

**ALGORITMA**

```
input(pilihan_user)
while (true) do
  depend on (pilihan_user)
  y: output(true)
  n: output("Kembali ke program.")
  output(false)
else:
  output("Pilihan tidak valid!")
  input(pilihan_user)
```

#### 5. F04-Menu & Help



```

type Monster: < id: integer;
                type: string;
                atk_power: int;
                def_power: integer;
                hp: integer>
type Monster_Shop: < monster_id: integer;
                    stock: integer;
                    price: integer>
type Item_Shop: < type: string;
                 stock: integer;
                 price: integer >
type Monster_Inventory : < user_id: integer;
                           monster_id: integer;
                           level: integer>

type Item_Inventory: < user_id: integer;
                      type: string;
                      quantity: integer >
type Hasil_Fungsi: < monster_shop: array of Monster_Shop;
                     item_shop: array of Item_Shop;
                     monster: array of Monster;
                     item_inventory : array of Item_Inventory>

{variables}
user: array of User
monster: array of Monster
monster_shop: array of Monster_shop
item_shop: array of Item_Shop
monster_inventory: array of Monster_Inventory
item_inventory: array of Item_Inventory
hasil_fungsi: Hasil_Fungsi
exit: bool
input_user: integer
validasi_logout: bool
validasi_exit: bool

{functions}

procedure list_help_agent()

function logout() -> (output:boolean)

function shop_management (input monster_shop: array of Monster_Shop, input
item_shop: array of Item_Shop, input monster: array of Monster, input
item_inventory: array of Item_Inventory) -> (output: functions_value:
Hasil_Fungsi)

function monster_management (input monster: array of Monster) -> (output:
monster: array of Monster)

function save_data(nama_folder : string , user : array of User, userMonster :
array of any , monster_shop : array of Monster_Shop, monster_inventory : array
of Monster_Inventory, item_shop : array of Item_Shop, item_inventory : array
of Item_Inventory) -> string

procedure exit()

```

#### ALGORITMA

```

output("Command: ")
input(input_user)

while True do
    if input_user = "1" then
        validasi_logout <- F03.logout()
        if validasi_logout = true then
            exit <- false

```

```

        break
    else
        {pass}

    elif input_user = "2" then
        output("Berikut adalah monster yang anda miliki")
        output("ID          |Type          |ATK Power |DEF Power |HP          |")
        output("-----|-----|-----|-----|-----|")
        for each row in 0 to length(monster_inventory)-1 do
            if str(monster_inventory[row][0]) = str(user_id) then
                for each type in 0 to length(monster)-1 do
                    if str(monster[type][0]) = str(monster_inventory[row][1])
                then
                    for each char in monster[type] do
                        lenchar <- length(char)
                        output(str(char) + " "*(10-lenchar), " |",
                    newline=false)
                    output("")

    elif input_user = "3" then
        output("No |Type          |Kuantitas      |")
        output("---|-----|-----|")
        i <- 1
        for each row in 0 to length(item_inventory)-1 do
            if str(item_inventory[row][0]) = str(user_id) then
                output(i, " |", newline=false)
                lenchar1 <- length(item_inventory[row][1])
                lenchar2 <- length(item_inventory[row][2])
                output(str(item_inventory[row][1]) + " "*(15-lenchar1) + "|" +
            item_inventory[row][2] + " "*(15-lenchar2) + "|")
                i <- i + 1

    elif input_user = "4" then
        output("Loading...")
        delay(1)
        output(""" Game Dimulai """)
        output("")

        function_value <- B05.komando(user_id, username, peta, monster_shop,
    item_shop, monster, monster_inventory, item_inventory, coin)
        monster_shop <- function_value[0]
        item_shop <- function_value[1]
        monster <- function_value[2]
        monster_inventory <- function

```

**function** help\_sudah\_login\_admin (input user: array of User, input monster: array of Monster, input monster\_shop: array of Monster\_Shop, input monster\_inventory: array of Monster\_Inventory, input item\_shop: array of Item\_Shop, input item\_inventory: array of Item\_Inventory) -> (output: exit: bool)  
 { Program akan menampilkan menu yang dapat di akses oleh admin setelah login}  
 { I.S menginput semua database}  
 { F.S menghasilkan exit bertipe bool, menandakan jika user memilih untuk keluar}

#### KAMUS LOKAL

{types}

```

type User : < id: integer;
              username: string;
              password: string;
              role: string;
              oc: integer>
type Monster: < id: integer;
               type: string;
               atk_power: int;
               def_power: integer;

```

```

        hp: integer>
type Monster_Shop: < monster_id: integer;
                    stock: integer;
                    price: integer>
type Monster_Inventory : < user_id: integer;
                           monster id: integer;
                           level: integer>
type Item_Shop: < type: string;
                 stock: integer;
                 price: integer >
type Item_Inventory: < user_id: integer;
                      type: string;
                      quantity: integer >
type Hasil_Fungsi: < monster_shop: array of Monster_Shop;
                    item_shop: array of Item_Shop;
                    monster: array of Monster;
                    item_inventory : array of Item_Inventory>

```

{variables}

```

user: array of User
monster: array of Monster
monster_shop: array of Monster_shop
monster_inventory: array of Monster_Inventory
item_shop: array of Item_Shop
item_inventory: array of Item_Inventory
hasil_fungsi: Hasil_Fungsi
exit: bool
input_user: integer
validasi_logout: bool
validasi_exit: bool

```

{functions}

```

procedure list_help_admin()

```

```

function logout() -> (output:boolean)

```

```

function shop_management (input monster_shop: array of Monster_Shop, input
item_shop: array of Item_Shop, input monster: array of Monster, input
item_inventory: array of Item_Inventory) -> (output: functions_value:
Hasil_Fungsi)

```

```

function monster_management (input monster: array of Monster) -> (output:
monster: array of Monster)

```

```

function save_data(nama_folder : string , user : array of User, userMonster :
array of any , monster_shop : array of Monster_Shop, monster_inventory : array
of Monster_Inventory, item_shop : array of Item_Shop, item_inventory : array
of Item_Inventory) -> string

```

```

procedure exit()

```

#### ALGORITMA

```

output("Command: ")
input(input_user)

while True do
    if input_user = "1" then
        validasi_logout <- F03.logout()
        if validasi_logout = true then
            exit <- false
            break
        else
            {pass}

```

```

        elif input_user = "2" then
            hasil_fungsi <- F12.shop_management(monster_shop, item_shop, monster,
item_inventory)
            monster_shop <- hasil_fungsi[0]
            item_shop <- hasil_fungsi[1]
            monster <- hasil_fungsi[2]
            item_inventory <- hasil_fungsi[3]

        elif input_user = "3" then
            monster <- F13.monster_management(monster)

        elif input_user = "4" then
            F15.save(user, monster, monster_shop, monster_inventory, item_shop,
item_inventory)

        elif input_user = "5" then
            validasi_exit <- F16.exit()
            if validasi_exit = true then
                exit <- true
                break
            else
                {pass}

        elif input_user = "help" then
            list_help_admin()

        else
            {pass}

        output("Command: ")
        input(input_user)

```

## 6. F05-Monster

**function** lvlStat  
 (input baseStat: BasicStat, lvl: integer) → (output stat: BattleStat)  
 {menghasilkan status monster berdasarkan status base dan levelnya.  
 I.S. status dasar dan level monster terdefinisi  
 F.S. dihasilkan status monster berdasarkan levelnya}

### KAMUS LOKAL

#### {types}

```

type BasicStat : < atk : integer ;
                    def : integer ;
                    hp : integer >
type BattleStat : < atk : integer ;
                    def : integer ;
                    hp : integer ;
                    hpmax : integer >

```

#### {variables}

```

baseStat: BasicStat
lvl: integer
stat: BattleStat

```

#### {functions/procedures}

```

function RealToInt (x: real) → integer
{mengubah input bertipe real menjadi integer}

```

### ALGORITMA

```

stat.atk ← RealToInt(baseStat.atk * (1 + 0.1*(lvl-1)))

```



```

stat.def ← RealToInt(baseStat.def * (1 + 0.1*(lvl-1)))
stat.hp ← RealToInt(baseStat.hp * (1 + 0.1*(lvl-1)))

if (stat[1] > 50) then:
    stat[1] ← 50
stat.hpmax ← stat.hp
→(stat)

```

**function** TampilMonsterDetail(input monster: Monster) → (output tampilan: string)  
 {menghasilkan tampilan detail mengenai monster yang dipilih.  
 I.S. data monster terdefinisi  
 F.S. dihasilkan tampilan/keterangan detail monster yang dipilih}

#### KAMUS LOKAL

```

{types}
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type BattleStat : < atk : integer ;
                  def : integer ;
                  hp : integer ;
                  hpmax : integer >
type Monster : < id : integer ;
                 name : string ;
                 lvl : integer ;
                 baseStat : BasicStat >

```

```

{variables}
monster: Monster
tampilan, nama: string
id, lvl: integer
stat: BattleStat

```

```

{functions/procedures}
function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
{menghasilkan status monster berdasarkan status base dan levelnya}

```

#### ALGORITMA

```

id ← monster.id
nama ← monster.name
lvl ← monster.lvl
stat ← lvlStat(monster.baseStat, lvl)
tampilan ← "
===== " , (nama) , " LVL " , (lvl) , " =====
Monster id   : " , (id) , "
ATK Power   : " , (stat.atk) , "
DEF Power   : " , (stat.def) , "
HP           : " , (stat.hp)
→ (tampilan)

```

**function** TampilanOwcaDex  
 (input monsters: array of array of string) → (output tampilan: string)  
 {menghasilkan tampilan utama owcadex.  
 I.S. seluruh data monster-monster terdefinisi  
 F.S. dihasilkan tampilan utama untuk owcadex}

#### KAMUS LOKAL

```

{variables}
monsters: array of array of string
tampilan: string
i: integer

```

```

{functions/procedures}
  function len (x: list) → integer
  {menghitung panjang list}
  function IntToStr (x: integer) → string
  {mengubah input bertipe integer menjadi string}

```

#### ALGORITMA

```

    tampilan ← "
===== OWCA DEX ====="
    i traversal [0..(len(monsters))]
        tampilan ← tampilan + "
, IntToStr(i+1) , ". " , (monsters[i][1]) , "

    tampilan ← tampilan + "

Ketik monster id untuk melihat detail.
Ketik 0 untuk keluar"
    → (tampilan)

```

**procedure** owcaDex(input monsters: array of array of string)  
 {menampilkan owcadex, library dimana user dapat melihat monster apa saja yang ada dalam game, serta dapat melihat statusnya pada level 1-5.  
 I.S. seluruh data monster-monster terdefinisi  
 F.S. ditampilkan owcadex sesuai spesifikasi pada layar}

#### KAMUS LOKAL

```

{types}
  type BasicStat : < atk : integer ;
                    def : integer ;
                    hp : integer >
  type Monster : < id : integer ;
                   name : string ;
                   lvl : integer ;
                   baseStat : BasicStat >

{variables}
  monsters: array of array of string
  tampilan, tampilMonster, name: string
  idValid, lvlValid: array of string
  id, lvl: integer
  baseStat: BasicStat
  monster: Monster

{functions/procedures}
  function TampilanOwcaDex (monsters: array of array of string) → string
  {menghasilkan tampilan utama owcadex}
  function IntToStr (x: integer) → string
  {mengubah input bertipe integer menjadi string}
  function StrToInt (x: string) → integer
  {mengubah input bertipe string menjadi integer}
  function len (x: list) → integer
  {menghitung panjang list}
  function TampilMonsterDetail (monster: Monster) → string
  {menghasilkan tampilan detail mengenai monster yang dipilih}
  function inputValid (text: string, line: string, pilihanValid: array of
string,
                        typeValid: string) → (integer / string)
  {menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}

```

#### ALGORITMA

```

    tampilan ← TampilanOwcaDex(monsters)
    i traversal [0..len(monsters)]:

```

```

        idValid ← IntToStr(i)
    i traversal [0..5]:
        lvlValid ← IntToStr(i)

        id ← 1
        while True:
            id ← inputValid(tampilan,"",idValid,"int")
            if (id = 0) then:
                break
            else:
                monster.name ← monsters[id-1][1]
                i traversal [0..2]:
                    monster.baseStat[i] ← StrToInt(monsters[id-1][i+2])

                lvl ← 1
                while (lvl ≠ 0):
                    monster.lvl ← lvl
                    tampilMonster ← TampilMonsterDetail(monster)
                    tampilMonster ← tampilMonster + "

Ketik 1-5 untuk melihat status monster pada level tersebut.
Ketik 0 untuk kembali ke OWCA DEX."
                    lvl ←
                        inputValid(tampilMonster,"",lvlValid,"int")

```

## 7. F06-Potion

**function** PeningkatanEfek  
 (input opsi: integer, potions: array of BattlePotion, stat: BattleStat) →  
 (output i, efek: integer, tampilan, potions[i-1].type: string)  
 {menghasilkan letak pilihan yang sebenarnya, karena potion dengan qty = 0 tidak  
 ditampilkan, besar peningkatan efek, tampilan efek jika potion digunakan, serta type  
 potion yang ditingkatkan.  
 I.S. status sebelum peningkatan dan potion yang dipilih terdefinisi  
 F.S. status ditingkatkan berdasarkan potion yang dipilih}

### KAMUS LOKAL

**{types}**  
type BattlePotion : < type : string ;  
                                 qty : integer ;  
   uses : integer >  
type BattleStat : < atk : integer ;  
                                 def : integer ;  
                                 hp : integer ;  
                                 hpmax : integer >  
**{variables}**  
 opsi, i, NPotion, statAsal, efek: integer  
 potions: array of BattlePotion  
 stat: BattleStat  
 tampilan: string  
**{functions/procedures}**  
function len (x: list) → integer  
 {menghitung panjang list}  
function RealToInt (x: real) → integer  
 {mengubah input bertipe real menjadi integer}

### ALGORITMA

```

NPotion ← len(potions)
i ← 0

{Cek letak opsi sebenarnya}

```

```

while (i < NPotion) and (opsi ≠ 0) do:
    if (potions[i][1] ≠ 0) then:
        opsi ← opsi - 1
        i ← i + 1

{Cek stat yang ditingkatkan}
depend on potions[i-1].type
    (potions[i-1].type = "Heal"):
        jenisStat ← "HP"
        statAsal ← stat.hp
        efek ← RealToInt(stat.hp * 1.25)
        if (efek > stat.hpmax):
            efek ← stat[3]
    (potions[i-1].type = "ATK"):
        jenisStat ← "ATK Power"
        statAsal ← stat.atk
        efek ← RealToInt(stat.atk * 1.05)
    (potions[i-1].type = "DEF"):
        jenisStat ← "DEF Power"
        statAsal ← stat.def
        efek ← RealToInt(stat.def * 1.05)
        if (efek > 50):
            efek ← 50

{Generate tampilan}
tampilan ← "
Efek:
" , (jenisStat) , " = " , (statAsal) , " --> " , (efek)

→ (i,efek,tampilan,potions[i-1].type)

```

**function** GunakanPotionLain() → boolean

{menghasilkan statement apakah user akan menggunakan potion lain atau tidak.  
I.S. user tidak jadi atau terkena batasan saat memilih potion  
F.S. user memilih untuk menggunakan potion lain atau tidak}

#### KAMUS LOKAL

##### {variables}

pilihanValid: array of character  
lineInput, pilih: string

##### {functions/procedures}

function inputValid (text: string, line: string, pilihanValid: array of  
string,  
typeValid: string) → (integer / string)

#### ALGORITMA

```

pilihanValid ← ["y","Y","n","N"]
lineInput ← "Mau gunakan potion lain? (y/n) "

Pilih ← inputValid("",lineInput,pilihanValid,"str")
if (pilih = "y") or (pilih = "Y") then:
    → (True)
else:
    → (False)

```

**function** TampilanPotionUse

(input potions: array of BattlePotion) →  
(output tampilan: string, number: integer)

{menghasilkan tampilan utama potion use, berisi pilihan potion yang dimiliki, serta mengembalikan nomor pilihan cancel.  
I.S. user memiliki potion untuk digunakan  
F.S. dihasilkan tampilan utama potion use, pilihan terakhir adalah cancel}

#### KAMUS LOKAL

##### {types}

```
type BattlePotion : < type : string ;  
               qty : integer ;  
               uses : integer >
```

##### {variables}

```
potions: array of BattlePotion  
tampilan, namaPotion: string  
number, NPotion: integer
```

##### {functions/procedures}

```
function len (x: list) → integer  
{menghitung panjang list}  
function IntToStr (x: integer) → string  
{mengubah input bertipe integer menjadi string}
```

#### ALGORITMA

```
NPotion ← len(potions)  
tampilan ← "  
Pilih potion:"  
  
{Generate tampilan}  
number ← 1  
i traversal [0..NPotion-1]:  
    depend on potions[i].type  
        (potions[i].type = "ATK"): namaPotion ← "Strength  
Potion"  
        (potions[i].type = "DEF"): namaPotion ← "Resilience  
Potion"  
        (potions[i].type = "Heal"): namaPotion ← "Healing  
Potion"  
  
    if (potions[i].qty ≠ 0) then:  
        tampilan ← tampilan + "  
    " , IntToStr(number) , ". " , (namaPotion) , " + " *(20-len(namaPotion)) +  
    "(Qty: " , " IntToStr(potions[i].qty) , "| Uses: " ,  
    IntToStr(potions[i].uses) , ")"  
        number ← number + 1  
  
    tampilan ← tampilan + "  
    " , IntToStr(number) , ". Cancel"  
  
→ (tampilan,number)
```

#### function PotionUse

```
(input stat: BattleStat, potions: array of BattlePotion) →
```

```
(output stat: BattleStat, potions: array of BattlePotion, used: integer)
```

{ditampilkan jika user memilih untuk menggunakan potion dalam battle, menampilkan efek yang terjadi ketika menggunakan potion.

I.S. user memiliki potion yang dapat digunakan

F.S. status monster ditingkatkan jika memilih potion, status monster tetap jika tidak jadi menggunakan potion}

#### KAMUS LOKAL

##### {types}

```
type BattleStat : < atk : integer ;  
               def : integer ;
```



```

                                (type = "DEF") : stat.def ← efek
                                (type = "Heal") : stat.hp ← efek
                                potions[i-1].qty ← potions[i-1].qty - 1
                                potions[i-1].uses ← 0
                                used ← 1
                                → (stat,potions,used)
                                if not(GunakanPotionLain()) then:
                                    → (stat,potions,used)
                                →(stat,potions,used)

```

## 8. F07-Inventory

**function** ItemInventory  
 (input user\_id: string, items\_inv: array of array of string,  
 userPotions: array of Potion) →  
 (output items\_inv: array of array of string)  
 {mengembalikan nilai-nilai yang berubah pada userPotions ke dalam item\_inventory.  
 I.S. seluruh parameter input terdefinisi  
 F.S. isi item\_inventory diisi dengan userPotion yang berubah setelah battle, hanya  
 quantity yg berubah}

### KAMUS LOKAL

#### {types}

type Potion : < type : string ;  
 qty : integer >

#### {variables}

user\_id, qty: string  
 items\_inv: array of array of string  
 userPotions: array of Potion  
 m, i, j: integer

#### {functions/procedures}

function len (x: list) → integer  
 {menghitung panjang list}  
function IntToStr (x: integer) → string  
 {mengubah input bertipe integer menjadi string}

### ALGORITMA

```

m ← len(items_inv)
j ← 0
i traversal [0..m-1]:
    if (items_inv[i][0] = user_id) then:
        qty ← IntToStr(userPotions[j].qty)
        items_inv[i][2] ← qty
        j ← j + 1
→ (items_inv)

```

**function** UserMonsters  
 (input user\_id: string, monsters: array of array of string,  
 monsters\_inv: array of array of string) →  
 (output userMonsters: array of Monster)  
 {mengumpulkan data monster yang dimiliki user dan disimpan di userMonsters.  
 I.S. seluruh parameter input terdefinisi  
 F.S. data seluruh monster yang dimiliki user disimpan di userMonsters}

### KAMUS LOKAL

#### {types}

```

type Monster : < id : integer ;
                    name : string ;
                    lvl : integer ;
                    baseStat : basicStat >
type BasicStat : < atk : integer ;
                    def : integer ;
                    hp : integer >

{variables}
user_id: string
monsters, monsters_inv: array of array of string
userMonsters: array of Monster
monster: Monster
i: integer

{functions/procedures}
function len (x: list) → integer
{menghitung panjang list}
function StrToInt (x: string) → integer
{mengubah input bertipe string menjadi integer}

```

#### ALGORITMA

```

i traversal [0..len(monsters_inv)-1]:
    if (monsters_inv[i][0]) = user_id) then:
        {data dari monster_inventory.csv}
        monster.id ← StrToInt(monsters_inv[i][1])
        monster.lvl ← StrToInt(monsters_inv[i][2])

        {data dari monster.csv}
        monster.name ← monsters[id-1][1]
        monster.baseStat.atk ← StrToInt(monsters[id-1][2])
        monster.baseStat.def ← StrToInt(monsters[id-1][3])
        monster,baseStat.hp ← StrToInt(monsters[id-1][4])
        userMonsters[i] ← monster
    → (userMonsters)

```

```

function UserPotions
    (input user_id: string, items_inv: array of array of string) →
    (output userPotions: array of Potion)
{mengumpulkan data potion yang dimiliki user dan disimpan pada userPotions.
I.S. seluruh parameter input terdefinisi
F.S. data seluruh potion yang dimiliki user disimpan di userPotions}

```

#### KAMUS LOKAL

```

{types}
type Potion : < type : string ;
                    qty : integer >

{variables}
user_id: string
items_inv: array of array of string
userPotions: array of Potion
potion: Potion
i: integer

{functions/procedures}
function len (x: list) → integer
{menghitung panjang list}
function StrToInt (x: string) → integer
{mengubah input bertipe string menjadi integer}

```

#### ALGORITMA



```

i traversal [0..(len(items_inv)-1):
    if (items_inv[i][0] = user_id) then:
        potion.type ← items_inv[i][1]
        depend on type
            (type = "strength") : type ← "ATK"
            (type = "resilience") : type ← "DEF"
            (type = "healing") : type ← "Heal"
        potion.qty ← StrToInt(items_inv[i][2])
        userPotions[i] ← potion
→ (userPotions)

```

**function** KembaliKeInventory() → boolean  
 {menghasilkan statement apakah user ingin kembali ke inventory atau keluar.  
 I.S. user berada pada tampilan status monster/item  
 F.S. user memilih untuk kembali ke inventory atau langsung keluar, true jika kembali,  
 false jika keluar}

#### KAMUS LOKAL

**{variables}**  
 pilihanValid: array of character  
 textInput: string  
 pilih: integer

**{functions/procedures}**  
function inputValid (text: string, line: string, pilihanValid: array of  
 string,  
 typeValid: string) → (integer / string)

#### ALGORITMA

```

    pilihanValid ← ["0","1"]
    textInput ← "
Ketik 1 untuk kembali ke inventory
Ketik 0 untuk keluar"
    pilih ← inputValid(textInput,"",pilihanValid,"int")
    if (pilih = 0) then:
        → (False)
    else:
        → (True)

```

**function** TampilPotionDetail  
 (input userPotions: array of Potion, pilih: integer) → (output tampilan:  
 string)  
 {menghasilkan tampilan detail mengenai potion yang dipilih, userPotion dengan qty = 0  
 tidak akan ditampilkan.  
 I.S. user memilih untuk melihat detail potion yang dimiliki  
 F.S. dihasilkan tampilan detail potion}

#### KAMUS LOKAL

**{types}**  
type Potion : < type : string ;  
 qty : integer >

**{variables}**  
 userPotions: array of Potion  
 tampilan, type, nama: string  
 potion: Potion  
 pilih, i, qty: integer

**{functions/procedures}**

```

function len (x: list) → integer
{menghitung panjang list}
function IntToStr (x: integer) → string
{mengubah input bertipe integer menjadi string}

```

#### ALGORITMA

```

    {cek letak pilihan sebenarnya, qty = 0 di skip}
    i ← 0
    while (i < len(userPotions)) and (pilih >= 1) do:
        if (userPotions[i].qty ≠ 0) then:
            pilih ← pilih - 1
            i ← i + 1

    potion ← userPotions[i-1]

    type ← potion.type
    qty ← potion.qty
    depend on type
        (type = "ATK") : nama ← "strength"
        (type = "DEF") : nama ← "resilience"
        (type = "Heal") : nama ← "healing"
    tampilan ← "
===== " , (nama) , " POTION =====
(Dimiliki: " , IntToStr(qty) , ")
"
    depend on type
        (type = "Heal") :
            tampilan ← tampilan + "
Potion ini memberikan efek peningkatan HP monster sebesar 25% dari base HP.
Item ini hanya dapat digunakan sekali dalam battle. Tidak dapat digunakan
jika HP monster masih penuh."
        (type = "DEF") :
            tampilan ← tampilan + "
Potion ini memberikan efek peningkatan DEF Power sebesar 5%. Item ini
hanya dapat digunakan sekali dalam battle. Tidak dapat digunakan jika
DEF Power monster sudah mencapai maksimum (50)."
        (type = "ATK") :
            tampilan ← tampilan + "
Potion ini memberikan efek peningkatan ATK Power sebesar 5%. Item ini
hanya dapat digunakan sekali dalam battle."
    → (tampilan)

```

```

function TampilanInv
    (input userMonsters: array of Monster, userPotions: array of Potion,
     user_id, username: string, userOC: integer) →
    (output tampilan: string, number: integer)
{menghasilkan tampilan utama inventory user dan batas nomor pilihan.
I.S. seluruh parameter input terdefinisi
F.S. dihasilkan tampilan utama inventory user berdasarkan monster dan item yang
dimiliki, mengeluarkan nomor pilihan terakhir}

```

#### KAMUS LOKAL

```

{types}
type Monster : < id : integer ;
                  name : string ;
                  lvl : integer ;
                  baseStat : basicStat >
type Potion : < type : string ;
                qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type BattleStat : < atk : integer ;
                  def : integer ;

```

```

        hp : integer ;
        hpmax : integer >

{variables}
    userMonsters: array of Monster
    userPotions: array of Potion
    stat: BattleStat
    user_id, username, name, tampilan: string
    userOC, NMonster, Npotion, i, number, lvl: integer

{functions/procedures}
    function len (x: list) → integer
    {menghitung panjang list}
    function IntToStr (x: integer) → string
    {mengubah input bertipe integer menjadi string}
    function lvlStat (baseStat: basicStat , lvl: integer) → BattleStat
    {menghasilkan status monster berdasarkan status base dan levelnya}

```

#### ALGORITMA

```

        NMonster ← len(userMonsters)
        NPotion ← len(userPotions)
        tampilan ← "
===== " , (username) , "'S INVENTORY (User ID: " , (user_id) , ")
=====
Jumlah O.W.C.A Coin mu sekarang adalah " , (userOC)
        number ← 0
        i traversal [0..Nmonster-1]:
            number ← number + 1
            name ← userMonsters[i].name
            lvl ← userMonsters[i].lvl
            stat ← lvlStat(userMonsters[i].baseStat,lvl)
            tampilan ← tampilan + "
            " , IntToStr(number) , ". Monster (Name: " , (name) , ", Lvl: " ,
(lvl) , ", HP: " , IntToStr(stat.hp)

            i traversal [0..Npotion-1]:
                if (userPotions[i].qty ≠ 0) then:
                    number ← number + 1
                    tampilan ← tampilan + "
                    " , (number) , ". Potion (Type: " , (userPotions[i].type) , ",
Qty: " , IntToStr(userPotions[i].qty)

            tampilan ← tampilan + "

        Ketik id untuk melihat detail
        Ketik 0 untuk keluar"
        → (tampilan,number)

```

```

procedure TampilInventoryUser
    (input userMonsters: array of Monster, userPotions: array of Potion, user_id:
    string, username: string, userOC: integer)
    {menampilkan isi inventory user.
    I.S. user memiliki monster dan item(potion)
    F.S. ditampilkan setiap monster dan item yang dimiliki user}

```

#### KAMUS LOKAL

```

{types}
    type Monster : < id : integer ;
                    name : string ;
                    lvl : integer ;
                    baseStat : basicStat >
    type Potion : < type : string ;

```

```

        qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >

{variables}
userMonsters : array of Monster
userPotions : array of Potion
userOC, pilih, number, NMonster : integer
rangeMonster, rangePotion : range
pilihanValid : array of string
tampilan : string

{functions/procedures}
function TampilanInv
    (input userMonsters: array of Monster, userPotions: array of Potion,
     User_id, username: string, userOC: integer) →
    (output tampilan: string, number: integer)
    {menghasilkan tampilan utama inventory user dan batas nomor pilihan}
    function TampilMonsterDetail (monster: Monster) → string
    {menghasilkan tampilan detail mengenai monster yang dipilih}
    function TampilPotionDetail (userPotions: array of Potion, pilih: integer) →
        string
    {menghasilkan tampilan detail mengenai potion yang dipilih, userPotion dengan
     qty = 0 tidak akan ditampilkan}
    function inputValid (text: string, line: string, pilihanValid: array of
string,
                        typeValid: string) → (integer / string)
    {menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
    function len (x: list) → integer
    {menghitung panjang list}
    function KembaliKeInventory () → boolean
    {menghasilkan statement apakah user ingin kembali ke inventory atau keluar}
    function IntToStr (x: integer) → string
    {mengubah input bertipe integer menjadi string}

```

#### ALGORITMA

```

NMonster ← len(userMonsters)
(tampilan,number) ← TampilanInv(userMonsters, userPotions, user_id, username,
                                userOC)

i traversal [0..number]
    pilihanValid[i] ← IntToStr(i)
rangeMonster ← range(1,NMonster+1)
rangePotion ← range(NMonster+1, number+1)

pilih ← 1
while True:
    pilih ← inputValid(tampilan,"",pilihanValid,"int")
    depend on pilih
        (pilih = 0):
            break
        (pilih in rangeMonster):
            monster ← userMonsters[pilih-1]
            output(TampilMonsterDetail(monster))
        (pilih in rangePotion):
            pilih ← pilih - NMonster
            output(TampilPotionDetail(userPotions,pilih))

    if not(KembaliKeInventory()) then:
        break

```

## 9. F08-Battle

```
function RandomMonster
    (input monsters: array of array of string, lvl: integer) →
    (output monster: Monster)
{men-generate random monster dengan level yang sudah ditentukan.
I.S. monsters, data mentah dari csv yang diread, dan level yang sudah ditentukan
terdefinisi
F.S. dihasilkan monster random, digunakan sebagai lawan dalam pertarungan}
```

#### KAMUS LOKAL

```
{types}
    type BasicStat : < atk : integer ;
                        def : integer ;
                        hp : integer >
    type Monster : < id : integer ;
                    name : string ;
                    lvl : integer ;
                    baseStat : BasicStat >

{variables}
    monsters : array of array of string
    rndmonster : array of string
    monster : Monster

{functions/procedures}
    function StrToInt (x: string) → integer
    {mengubah masukan bertipe string menjadi integer}
```

#### ALGORITMA

```
    rndmonster ← monsters[random_number(len(monsters))]
    monster.id ← StrToInt(rndmonster[0])
    monster.lvl ← lvl
    monster.name ← rndmonster[1]
    i traversal [0..2]
        monster.baseStat[i] ← StrToInt(rndmonster[i+2])
    → (monster)
```

```
procedure KemunculanMonster
    (input monster: Monster, character: string)
{menampilkan intro dan detail monster saat di awal battle.
I.S. monster dan karakter(user/enemy) terdefinisi
F.S. ditampilkan intro dan status monster pada layar}
```

#### KAMUS LOKAL

```
{types}
    type BasicStat : < atk : integer ;
                        def : integer ;
                        hp : integer >
    type BattleStat : < atk : integer ;
                        def : integer ;
                        hp : integer ;
                        hpmax : integer >
    type Monster : < id : integer ;
                    name : string ;
                    lvl : integer ;
                    baseStat : BasicStat >

{variables}
    monster : Monster
    character : string {username atau "enemy"}
    stat : BattleStat

{functions/procedures}
```

```

function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
{menghasilkan status monster berdasarkan status base dan levelnya}

```

#### ALGORITMA

```

stat ← lvlStat(monster.baseStat,monster.lvl)
if (character = "enemy") then:
    output("RAAWRRRR!!! Monster " , (monster.name) , " telah muncul!!!")
else:
    output("RAAWRRRR!!! " , (character) , " mengeluarkan monster " ,
        (monster.name) , " !!!")
output("
Name       : " , (monster.name) , "
ATK Power  : " , (stat.atk,2) , "
Def Power  : " , (stat.def,2) , "
HP         : " , (stat.hp,2) , "
Level      : " , (monster.lvl))

```

```

function PilihUserMonster
(input userMonsters: array of Monster) → (output monster: Monster)
{menghasilkan data monster pilihan user yang telah dimiliki user.
I.S. data monster milik user telah dipisahkan dari data utama
F.S. dikeluarkan monster pilihan user untuk battle}

```

#### KAMUS LOKAL

##### {types}

```

type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type Monster : < id : integer ;
                 name : string ;
                 lvl : integer ;
                 baseStat : basicStat >

```

##### {variables}

```

userMonsters : array of Monster
monster : Monster
Nmonster : integer
pilihanValid : array of string
tampilan : string

```

##### {functions/procedures}

```

function IntToStr (x: integer) → string
{mengubah input bertipe integer menjadi string}
function inputValid (text: string, line: string, pilihanValid: array of
string,
                    typeValid: string) → (integer / string)
{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
function len (x: list) → integer
{menghitung panjang list}

```

#### ALGORITMA

```

NMonster ← len(userMonsters)
i traversal [0..NMonster-1]
    pilihanValid[i] ← IntToStr(i+1)
tampilan ← "
Pilih Monster-mu untuk bertarung!

===== MONSTER LIST ====="
i traversal [0..NMonster-1]
    tampilan ← tampilan +
(i+1), ". " , (userMonsters[i][1])"
    pilih ← inputValid(tampilan, "", pilihanValid, "int")
    monster ← userMonsters[pilih-1]

```

→ (monster)

**function** Skip  
(input statMonster, statEnemy: BattleStat, turn: integer) →  
(output endedStatus: string, damageGiven, damageReceived: integer)  
{menentukan apakah user akan menang atau kalah dengan kondisi monster selalu saling menyerang, tanpa menggunakan potion, serta menghitung total damage yang diberikan dan diterima.  
I.S. user memilih skip saat battle, seluruh parameter input terdefinisi  
F.S. dikeluarkan hasil battle secara otomatis}

#### KAMUS LOKAL

##### {types}

```
type BattleStat : < atk : integer ;  
                    def : integer ;  
                    hp : integer ;  
                    hpmax : integer >
```

##### {variables}

```
statMonster, statEnemy : BattleStat  
turn, damageGiven, damageReceived, ATKPower, DEFPower : integer  
percent, ATK, reduced, damage, HPtarget : integer  
endedStatus : string
```

##### {functions/procedures}

```
function RealToInt (x: real) → integer  
{mengubah input bertipe real menjadi integer}  
function random_number ((x,y): range) → integer  
{menghasilkan bilangan integer random yang berada pada range (x,y)}
```

#### ALGORITMA

```
damageReceived ← 0  
damageGiven ← 0  
while (statMonster[2] > 0) and (statEnemy[2] > 0) do:  
    if (turn mod 2 = 1) then:  
        ATKPower ← statMonster.atk  
        DEFPower ← statEnemy.def  
        percent ← random_number((-30, 30))  
        ATK ← RealToInt(ATKPower * (1 + percent/100))  
        reduced ← RealToInt(ATK * DEFPower/100)  
        damage ← ATK - reduced  
        HPtarget ← statEnemy.hp - damage  
        if (HPtarget < 0) then:  
            HPtarget ← 0  
        statEnemy.hp ← HPtarget  
        damageGiven ← damageGiven + damage  
        turn ← turn + 1  
    else:  
        ATKPower ← statEnemy.atk  
        DEFPower ← statMonster.def  
        percent ← random_number((-30, 30))  
        ATK ← RealToInt(ATKPower * (1 + percent/100))  
        reduced ← RealToInt(ATK * DEFPower/100)  
        damage ← ATK - reduced  
        HPtarget ← statMonster.hp - damage  
        if (HPtarget < 0) then:  
            HPtarget ← 0  
        statMonster.hp ← HPtarget  
        damageReceived ← damageReceived + damage  
        turn ← turn + 1  
if (statMonster.hp = 0) then:  
    endedStatus ← "kalah"  
else:  
    endedStatus ← "menang"
```

```
→ (endedStatus,damageReceived,damageGiven)
```

#### function Attack

```
(input statAtk, statDef: BattleStat, nameAtk, nameDef, chrAtk, chrDef:
string) →
(output Hptarget, damage: integer)
{menghitung damage dan menampilkan detail serangan monster.
I.S. seluruh parameter input terdefinisi
F.S. hp defender berkurang sesuai kalkulasi attack}
```

#### KAMUS LOKAL

##### {types}

```
type BattleStat : < atk : integer ;
                    def : integer ;
                    hp : integer ;
                    hpmax : integer >
```

##### {variables}

```
{Atk = penyerang, Def = bertahan}
statAtk, statDef : BattleStat
nameAtk, nameDef, chrAtk, chrDef : string
ATKPower, DEFPower, percent, ATK, reduced, result, HPTarget : integer
percentage : string
```

##### {functions/procedures}

```
function random_number ((x,y): range) → integer
{menghasilkan bilangan integer random yang berada pada range (x,y)}
function RealToInt (x: real) → integer
{mengubah input bertipe real menjadi integer}
```

#### ALGORITMA

```
ATKPower ← statAtk.atk
DEFPower ← statDef.def

percent ← random_number((-30,30))
if (percent >= 0) then:
    percentage ← "+" , (percent) , " %"
else:
    percentage ← "" , (percent) , " %"

ATK ← RealToInt(ATKPower * (1 + percent/100))
reduced ← RealToInt(ATK * DEFPower/100)
result ← ATK - reduced

HPTarget ← statDef[2] - result
if (HPTarget < 0) then:
    HPTarget ← 0

    output("
(" , (chrAtk) , ") " , (nameAtk) , " menyerang!!!

(" , (chrDef) , ") " , (nameDef) , "'s HP = " , (HPTarget) , "

==== Attack Detail ====
Attack Power   : " , (ATK) , " (" , (percentage) , ")
Reduced by     : " , (reduced) , " (" , (-DEFPower) , "%
Result         : " , (result))
→ (HPTarget,damage)
```

#### function CanUsePotion

```
(input battlePotions: array of BattlePotion, monsterStat: BattleStat) →
```



<pre>         <u>boolean</u> {menentukan apakah user dapat menggunakan potion saat battle atau tidak. I.S. seluruh parameter input terdefinisi F.S. dihasilkan statement apakah user dapat memakai potion} </pre>	
<p><b>KAMUS LOKAL</b></p> <pre> {types}     <u>type</u> BattlePotion : &lt; type : string ;                         qty : <u>integer</u> ;                         uses : <u>integer</u> &gt;     <u>type</u> BattleStat : &lt; atk : <u>integer</u> ;                       def : <u>integer</u> ;                       hp : <u>integer</u> ;                       hpmax : <u>integer</u> &gt;  {variables}     battlePotions : <u>array of</u> BattlePotion     monsterStat : BattleStat     can : <u>boolean</u>     i : <u>integer</u>  {functions/procedures}     <u>function</u> len (x: list) → <u>integer</u>     {menghitung panjang list} </pre>	
<p><b>ALGORITMA</b></p> <pre>     can ← False     i ← 0     <u>while</u> (i &lt; len(battlePotion)) <u>and</u> (<u>not</u>(can)) <u>do</u>:         <u>if</u> (battlePotion[i].qty ≠ 0) <u>and</u> (battlePotion[i].uses ≠ 0) <u>then</u>:             <u>depend on</u> (battlePotion[i].type,monsterStat)                 (battlePotion[i].type = "ATK"): can ← True                 (battlePotion[i].type = "DEF") <u>and</u> (monsterStat.def &lt; 50)                                 : can ← True                 (battlePotion[i].type = "Heal") <u>and</u> (monsterStat.hp &lt;                                 monsterStat.hpmax) : can ← True             i ← i + 1     → (can) </pre>	

<pre> <u>function</u> BattleTurns     (<u>input</u> monster,enemy: Monster,userPotions: <u>array of</u> Potion, username: string)     →     (<u>output</u> userPotions: <u>array of</u> Potion, totalDamageReceived, totalDamageGiven:         <u>integer</u>, endedStatus: string) {menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan potion, kabur, skip battle, dan menampilkan detail status hingga battle berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta diterima, dan status battle. I.S. seluruh parameter input terdefinisi F.S. dikeluarkan damage serta status kemenangan hasil battle} </pre>	
<p><b>KAMUS LOKAL</b></p> <pre> {types}     <u>type</u> Monster : &lt; id : <u>integer</u> ;                     name : string ;                     lvl : <u>integer</u> ;                     baseStat : basicStat &gt;     <u>type</u> Potion : &lt; type : string ;                   qty : <u>integer</u> &gt;     <u>type</u> BattlePotion : &lt; type : string ;                        qty : <u>integer</u> ;                        uses : <u>integer</u> &gt; </pre>	

```

type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >
type BattleStat : < atk : integer ;
                   def : integer ;
                   hp : integer ;
                   hpmax : integer >

{variables}
monster, enemy : Monster
monsterStat, enemyStat : BattleStat
userPotions : array of Potion
battlePotions : array of BattlePotion
totalDamageReceived, totalDamageGiven, damageReceived, damageGiven : integer
pilih, turn : integer
endedStatus, username, textPilihan, textInput, lineInput : string
opsi : character
pilihanValid, opsiValid : array of integer
ended : boolean

{functions/procedures}
function lvlStat (baseStat: BasicStat , lvl: integer) → BattleStat
{menghasilkan status monster berdasarkan status base dan levelnya}
function inputValid (text: string, line: string, pilihanValid: array of
string,
                    typeValid: string) →(integer / string)
{menghasilkan input pilihan yang valid sesuai typeValid dan pilihanValid}
function Attack
    (input statAtk, statDef: BattleStat,
     nameAtk, nameDef, chrAtk, chrDef: string) →
    (output HPtarget, damage: integer)
{menghitung damage dan menampilkan detail serangan monster}
function CanUsePotion (battlePotions: array of BattlePotion,
                      monsterStat: BattleStat) → boolean
{menentukan apakah user dapat menggunakan potion saat battle atau tidak}
function len (x: list) → integer
{menghitung panjang list}

```

#### ALGORITMA

```

ended ← False
turn ← 1
enemyStat ← lvlStat(enemy.baseStat, enemy.lvl)
monsterStat ← lvlStat(monster.baseStat, monster.lvl)

i traversal [0..len(battlePotions)-1]
    battlePotions[i].type ← userPotions[i].type
    battlePotions[i].qty ← userPotions[i].qty
    battlePotions[i].uses ← 1

textPilihan ← "
1. Attack
2. Use Potion
3. Skip Battle
4. Quit"
pilihanValid ← ["1", "2", "3", "4"]
opsiValid ← ["y", "Y", "n", "N"]

totalDamageGiven ← 0
totalDamageReceived ← 0

while not(ended) do:
    if (turn mod 2 = 1) then:
        textInput ← "
===== TURN " , (turn) , " (" , (monster.name) , " - " , (username) , "
===== " + textPilihan
        pilih ← inputValid(textInput, "", pilihanValid, "int")
        depend on pilih
            (pilih = 1) :

```

```

Attack(monsterStat,
enemyStat,monster.name,
enemy.name,username,"enemy")
totalDamageGiven ← totalDamageGiven +
damageGiven
turn ← turn + 1
(pilih = 2) :
    if
    not(CanUsePotion(battlePotion,monsterStat))
    then:
        output("
Tidak ada potion yang dapat digunakan.
Tidak dapat memilih potion.")
    else:
        (monsterStat,battlePotions,used) ←
        PotionUse(monsterStat,battlePotions)
        if (used = 1) then:
            turn ← turn + 1
        (pilih = 3) :
            textInput ← "
Dengan memilih skip battle, monster dengan otomatis dalam keadaan
menyerang."
            lineInput ← "Yakin ingin skip battle?
(y/n)"
            opsi ← inputValid(textInput,lineInput,
opsiValid,"str")
            if (opsi = "y") or (opsi = "Y") then:
                (endedStatus, damageReceived, damageGiven)
                ← Skip(monsterStat, enemyStat, turn)
                totalDamageReceived ←
                totalDamageReceived
                +
                damageReceived
                totalDamageGiven ←
                totalDamageGiven +
                damageGiven
            (pilih = 4) :
                output("Anda melarikan diri dari
pertarungan")
                ended ← True
                endedStatus ← "kabur"
        else:
            output("
===== TURN " , (turn) , " (" , (enemy.name) , " - enemy =====")
            (monsterStat.hp, damageReceived) ←
            Attack(enemyStat,monsterStat,
enemy.name,
monster.name,
"enemy",username)
            totalDamageReceived ← totalDamageReceived +
            damageReceived
            turn ← turn + 1

        if not(ended) then:
            depend on (monsterStat.hp, enemyStat.hp)
            (monsterStat.hp = 0) :
                ended ← True
                endedStatus ← "kalah"
                output("
Yahhh, Anda dikalahkan monster " , (enemy.name) , ". Jangan menyerah, coba
lain kali!!!")
            (enemyStat.hp = 0) :

```

```

ended ← True
endedStatus ← "menang"
output("
Selamat, Anda berhasil mengalahkan monster " , (enemyName) , "!!!")

i traversal [0..(len(userPotions)-1)]
    userPotions[i].type ← battlePotion[i].type
    userPotions[i].qty ← battlePotion[i].qty
→ (userPotions, totalDamageReceived, totalDamageGiven, endedStatus)

```

**function** Battle  
 (input monsters: array of array of string, userMonsters: array of Monster,  
 userPotions: array of Potion, username: string) →  
 (output userPotions: array of Potion, bonus: integer)  
 {men-generate lawan, memilih monster, dan melakukan battle.  
 I.S. user dengan role agent memasukkan command battle, seluruh parameter input  
 terdefinisi  
 F.S. dikeluarkan userPotion yang berubah karena penggunaan serta bonus berdasarkan  
 status kemenangan battle}

#### KAMUS LOKAL

##### {types}

```

type Monster : < id : integer ;
                name : string ;
                lvl : integer ;
                baseStat : basicStat >
type Potion : < type : string ;
                qty : integer >
type BasicStat : < atk : integer ;
                  def : integer ;
                  hp : integer >

```

##### {variables}

```

monsters : array of array of string
userMonsters : array of Monster
monster, enemy : Monster
userPotions : array of Potion
username, endedStatus : string
bonus, enemyLvl, damageReceived, damageGiven: integer

```

##### {functions/procedures}

```

function RandomMonster (monsters: array of array of string, lvl: integer) →
    Monster
{men-generate random monster dengan level yang sudah ditentukan}
function random_number ((x,y): range) → integer
{menghasilkan bilangan integer random yang berada pada range (x,y)}
procedure KemunculanMonster (input monster: Monster, character: string)
{menampilkan intro dan detail monster saat di awal battle}
function PilihUserMonster (userMonsters: array of Monster) → Monster
{menghasilkan data monster pilihan user yang telah dimiliki user}
function BattleTurns
    (input monster, enemy: Monster, userPotions: array of Potion,
    username:
        string) →
        (output userPotions: array of Potion, totalDamageReceived,
        totalDamageGiven: integer, endedStatus: string)
{menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan
potion, kabur, skip battle, dan menampilkan detail status hingga battle
berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta
diterima, dan status battle}

```

#### ALGORITMA

```

enemyLvl ← random_number((1,5))
enemy ← RandomMonster(monsters, enemyLvl)

```

```

KemunculanMonster(enemy, "enemy")

monster ← PilihUserMonster(userMonsters)
KemunculanMonster(monster, username)

functionvalue ← BattleTurns(monster, enemy, userPotions, username)
userPotions ← functionvalue[0]
damageReceived ← functionvalue[1]
damageGiven ← functionvalue[2]
endedStatus ← functionvalue[3]

if (endedStatus ≠ "menang") then:
    bonus ← 0
else:
    bonus ← random_number((30,50))
output("
===== STATS =====
Hadiah                : " , (bonus) , " OC
Damage diberikan      : " , (damageGiven) , "
Damage diterima       : " , (damageReceived)
→ (userPotions, bonus)

```

## 10. F09-Arena

```

function CekBonus
    (input endedStatus: string, stage: integer) →
    (output bonus : integer)
{menghasilkan coin berdasarkan status kemenangan dan stage dalam arena.
I.S. user telah menyelesaikan suatu stage
F.S. dihasilkan bonus berdasarkan status kemenangan dan stage}

```

### KAMUS LOKAL

```

{variables}
    endedStatus : string
    stage : integer
    bonus : integer

{functions/procedures}
    function RealToInt (x: real) ← integer
    {mengubah input bertipe integer menjadi string}

```

### ALGORITMA

```

if (endedStatus ≠ "menang") then:
    bonus ← 0
    output("GAME OVER! Sesi latihan berakhir pada stage " , (stage))
else:
    bonus ← RealToInt(random_number((30,35))*(0.9 + 0.1*stage))
    output("STAGE CLEARED! Anda mendapatkan " , (bonus) , "OC pada stage
        ini")
    if (stage = 5) then:
        output("Selamat, Anda berhasil menyelesaikan seluruh stage
            arena!")
    →(bonus)

```

```

function Arena
    input userMonsters: array of Monster, userPotions: array of Potion,
        username: string, monsters: array of array of string) →
    (output userPotions: array of potion, totalBonus: integer)
{menjalankan arena: 5 st(age, di setiap stage terdapat battle dengan lvl enemy =
stage.

```

I.S. user dengan role agent memilih command arena, seluruh parameter input terdefinisi  
F.S. mengembalikan userPotion yang berubah karena penggunaan serta bonus coin yang diperoleh}

#### KAMUS LOKAL

##### {types}

```
type Monster : < id : integer ;  
                  name : string ;  
                  lvl : integer ;  
                  baseStat : BasicStat >  
type Potion : < type : string ;  
                qty : integer >  
type BasicStat : < atk : integer ;  
                  def : integer ;  
                  hp : integer >
```

##### {variables}

```
userMonsters : array of Monster  
userPotions : array of Potion  
monsters : array of array of string {read_csv("monster.csv")}  
bonus, totalBonus, stage, clear : integer  
damageReceived, damageGiven, totalDamageReceived, totalDamageGiven : integer  
monster, enemy : Monster
```

##### {functions/procedures}

```
function CekBonus (endedStatus: string, stage: integer) → integer  
{menghitung bonus berdasarkan status kemenangan dan stage arena}  
function RandomMonster (monsters: array of array of string, lvl: integer) →  
    Monster  
{men-generate random monster dengan level yang sudah ditentukan}  
procedure KemunculanMonster (input monster: Monster, character: string)  
{menampilkan intro dan detail monster saat di awal battle}  
function PilihUserMonster (userMonsters: array of Monster) → Monster  
{menghasilkan data monster pilihan user yang telah dimiliki user}  
function BattleTurns  
    (input monster, enemy: Monster, userPotions: array of Potion,  
     username: string) →  
    (output userPotions: array of Potion, totalDamageReceived,  
     totalDamageGiven: integer, endedStatus: string)  
{menampilkan giliran-giliran pada battle dengan opsi menyerang, menggunakan  
potion, kabur, skip battle, dan menampilkan detail status hingga battle  
berakhir, mengembalikan potion yang tidak digunakan, damage diberikan serta  
diterima, dan status battle}
```

#### ALGORITMA

```
output("Selamat datang di Arena!")  
monster ← PilihUserMonster(userMonsters)  
KemunculanMonster(monster, username)  
  
stage ← 1  
clear ← 0  
endedStatus ← "menang"  
totalBonus ← 0  
totalDamageReceived ← 0  
totalDamageGiven ← 0  
  
while (stage < 6) and (endedStatus = "menang") do:  
    if (stage ≠ 1) then:  
        output("Memulai stage berikutnya . . .")  
  
        output("===== STAGE" , (stage) , " =====")  
        enemy ← RandomMonster(monsters,stage)  
        KemunculanMonster(enemy, "enemy")  
        functionvalue ← BattleTurns(monster, enemy, userPotions, username)  
        userPotions ← functionvalue[0]
```

```

endedStatus ← functionvalue[3]

totalBonus ← totalBonus + CekBonus(endedStatus,stage)
totalDamageReceived ← totalDamageReceived + functionvalue[1]
totalDamageGiven ← totalDamageGiven + functionvalue[2]

if (endedStatus = "menang") then:
    clear ← clear + 1
    stage ← stage + 1

output("
===== STATS =====
Total hadiah          : " , (totalBonus) "
Jumlah stage          : " , (clear) "
Damage diberikan      : " , (round(totalDamageGiven,2)) "
Damage diterima       : " , (round(totalDamageReceived,2))
→(userPotions,totalBonus)

```

## 11. F10-Shop & Currency

**prosedure** shop  
 (input userid:String , monster\_shop:array of any , item\_shop:list,  
 monster:array of any, monster\_inventory:array of any , coin : integer) →  
 (output monster\_shop:array of any ,item\_shop : array of any,monster ; array  
 of any ,monster\_inventory : array of any ,item\_inventory : array of any ,  
 coin : Integer)  
 { program akan menerima inputan data berupa array kemudian , program akan meminta  
 user untuk memilih jenis yang ingin dibeli dan jumlah yang akan dibeli. Selanjutnya  
 program akan melanjutkan dengan menampilkan koin dan akan mengembalikan value dari  
 data yang telah diubahh  
 I.S.seluruh parameter input terdefinisi  
 F.S.mengembalikan value data yang telah dibeli user }

### KAMUS LOKAL

```

pilih : String
lihat_apa : String
item : String
id_monster_input : Any
cek_inventory : Boolean
found : Boolean
harga_item : Integer
harga_total : Integer
Monster_tambah : List of String
id_item_input : Integer
jumlah_item : Integer
item_type : String

```

### ALGORITMA

```

output("Selamat datang di Shop!")
sleep(1)

switch <- 0
while (switch = 0) do
    linePilih <- "Pilih aksi (lihat/beli/keluar)"
    pilihValid[1] <- "lihat"
    pilihValid[2] <- "beli"
    pilihValid[3] <- "keluar"
    inputValid("", linePilih, pilihValid, "str") -> pilih

    if (pilih = "keluar") then

        output('\nSampai jumpa lagi di Shop!\n')
        switch <- 1
    else if (pilih = "lihat") then

```

```

lineLihat <- "Mau lihat apa? (monster/potion)"
lihatValid[1] <- "monster"
lihatValid[2] <- "potion"
inputValid("", lineLihat, lihatValid, "str") -> lihat_apa
if (lihat_apa = "monster") then
  output("\nID |Type |ATK Power |DEF Power |HP |Stok |Harga |")
  output("----|-----|-----|-----|-----|-----|")
  for each row in monster do
    if (str(row[0]) in str(monster_shop)) then
      for each char in row do
        length_char <- len(char)
        output(str(char) + " "*(10-length_char), end="|")
      for each row2 in monster_shop do
        if (row2[0] = row[0]) then
          length_row2 <- len(str(row2[1]))
          length_row3 <- len(str(row2[2]))
          output(str(row2[1]) + " "*(10-length_row2) + "|" +
            str(row2[2]) + " "*(10-length_row3) + "|", end="")
          output("")
    else if (lihat_apa = "potion") then
      output("\nID |Type |Stock |Harga |")
      output("----|-----|-----|-----|")
      i <- 1
      for each row in item_shop do
        output(i, end=" |")
        i <- i + 1
        for each char in row do
          length_char <- len(char)
          output(str(char) + " "*(15-length_char), end="|")
        output("")

else if (pilih = "beli") then
  lineInput <- "Mau beli apa? (monster/potion)"
  itemValid[1] <- "monster"
  itemValid[2] <- "potion"
  inputValid("", lineInput, itemValid, "str") -> item

  if (item = "monster") then
    idValid[1] <- '0'
    for i traversal [1..len(monster_shop)] do
      idValid[i+1] <- str(monster_shop[i][0])

    textInput <- f"Koin Anda saat ini: {coin} OC\nKetik id monster untuk
      membeli\nKetik 0 untuk kembali"
    inputValid(textInput, "", idValid, "int") -> id_monster_input

    cek_inventory <- true
    while (cek_inventory) do
      found <- false
      for each row in monster_inventory do
        if (row[0] = user_id) then
          if (str(id_monster_input) = str(row[1])) then
            found <- true
            break
      if (found) then
        output(f"\nAnda telah memiliki
          {monster[id_monster_input-1][1]}")
        inputValid(textInput, "", idValid, "int") -> id_monster_input
      else
        break

    for i, row in enumerate(monster_shop) do
      if (str(row[0]) = str(id_monster_input)) then
        baris <- i
        break
    if (0 >= int(monster_shop[baris][1])) then
      output("\nPembelian Gagal :( , Stock tidak mencukupi")
    else
      harga_item <- int(monster_shop[baris][2])
      if (harga_item > coin) then

```



```

        output("\nPembelian Gagal :( , Koin tidak mencukupi")
    else
        monster_tambah[1] <- str(user_id)
        monster_tambah[2] <- str(id_monster_input)
        monster_tambah[3] <- str(1)
        monster_inventory <- monster_tambah

        monster_shop[i][1] <- int(monster_shop[i][1]) - 1

        coin <- coin - harga_item
        output(f"\nSelamat, monster {monster[id_monster_input-1][1]}
telah berhasil dibeli!")
        output(f"\nKoin tersisa: {coin} OC")

    else if (item == "potion") then
        for i traversal [1..len(item_shop)+1] do
            idValid[i] <- str(i-1)
            textInput <- f"Koin Anda saat ini: {coin} OC\nKetik id potion untuk
membeli\nKetik 0 untuk kembali"
            inputValid(textInput, "", idValid, "int") -> id_item_input

            for i, row in enumerate(monster_shop) do
                if (str(row[0]) == str(id_item_input)) then
                    baris <- i
                    break
            input("\nMasukkan Jumlah Item yang ingin dibeli: ") -> jumlah_item
            if (0 >= int(item_shop[int(id_item_input)-1][1]) or int(jumlah_item) >
int(item_shop[int(id_item_input)-1][1])) then
                output("\nPembelian Gagal :( , Stock tidak mencukupi")
            else
                harga_item <- int(item_shop[int(id_item_input)-1][2])
                harga_total <- harga_item * jumlah_item
                item_type <- item_shop[int(id_item_input)-1][0]
                if (harga_total > coin) then
                    output("\nPembelian Gagal :( , Koin tidak mencukupi")
                else
                    found <- false
                    i <- 0
                    while (i < len(item_inventory) and not(found)) do
                        if (str(item_inventory[i][0]) == str(user_id)) then
                            if (str(item_inventory[i][1]) == str(item_type)) then
                                item_inventory[i][2] <-
str(int(item_inventory[i][2]) + jumlah_item)
                                found <- true
                                i <- i + 1

                            if (not(found)) then
                                item_tambah[1] <- str(user_id)
                                item_tambah[2] <- str(item_type)
                                item_tambah[3] <- str(jumlah_item)
                                item_inventory <- item_tambah

                                item_shop[int(id_item_input)-1][1] <-
str(int(item_shop[int(id_item_input)-1][1]) - jumlah_item)
                                output(f"\nSelamat!!! Anda Telah membeli potion {item_type}
sebanyak {jumlah_item}")

                                coin <- coin - harga_total
                                output(f"\nKoin tersisa: {coin} OC")

function_value[1] <- monster_shop
function_value[2] <- item_shop
function_value[3] <- monster
function_value[4] <- monster_inventory
function_value[5] <- item_inventory
function_value[6] <- coin

```

## 12. F11-Laboratory

```
function upgrade_cost(level: integer) -> integer
{ Menghasilkan banyak koin OC yang dibutuhkan untuk meng-upgrade sesuai dengan levelnya }
{ I.S level user}
{ F.S biaya upgrade}
```

### KAMUS LOKAL

cost : integer

### ALGORITMA

```
if level = 1 then
    cost ← 200
else If level = 2 then
    cost ← 300
else If level = 3 then
    cost ← 600
else If level = 4 then
    cost ← 800
output(cost)
```

```
procedure laboratory
    ( input user_id : integer, user_coin : integer, monster : array of string,
      monster_inventory : array of array ) →
    ( output function_value : array of array )
{ program akan meng-upgrade level dari monster yang dimiliki oleh pengguna. Program ini akan melibatkan pemilihan monster, membayar biaya upgrade, dan meningkatkan level monster jika pengguna memiliki cukup koin.
  I.S. seluruh parameter input terdefinisi
  F.S. mengembalikan array yang berisi koin pengguna dan monster_inventory yang telah diperbarui }
```

### KAMUS LOKAL

#### (types)

```
type monsterStat : < id : integer ; type : string ; atk_power : integer ;
                    def_power : integer ; hp : integer >
type userMonster : < user_id : integer ; monster_id : integer ; level :
                    integer >
```

#### (variables)

```
list_index : array of integer
list_id_monster : array of string
enumerate : integer
selected_monster_index : string
cek_valid : boolean
selected_monster : string
idx_monster_di_inventory : integer
idx_monster_di_database : integer
next_level : integer
level : integer
cost : integer
upgrade : string
opt : string
function_value : array
```

### ALGORITMA

```

list_index ← []
list_id_monster ← []
enumerate ← 1
row ← 0
(anggap len sudah terdefinisi sebagai panjang dari array)
while (row < len-1) do
  if (monster_inventory[row][0] = user_id) then
    idx_monster ← monster_inventory[row][1]
    for i from 0 to length(monster) - 1 Do
      If monster[i][0] = idx_monster Then
        list_id_monster ← list_id_monster + idx_monster
        list_index ← list_index + enumerate
        output(enumerate, monster[i][1], monster_inventory[row][2])
        enumerate ← enumerate + 1
    row ← row + 1

bool ← True
while (bool = True) do
  output(user_coin)
  output("Silahkan pilih monster yang ingin di Upgrade!")
  input(selected_monster_index)

  cek_valid ← False
  while cek_valid = False do
    if selected_monster_index not in list_index then
      output("Pilihan monster tidak valid. Silakan pilih lagi.")
      input(selected_monster_index)
    else
      cek_valid ← True
      selected_monster ← list_id_monster[int(selected_monster_index) -
1]
      index traversal [0..len(monster_inventory)]
        if (monster_inventory[index][0] = user_id) and
(monster_inventory[index][1] = selected_monster) then
          idx_monster_di_inventory ← index
          index2 traversal [0..len(monster_inventory)]
            if (monster[index2][0] = selected_monster) then
              idx_monster_di_database ← index2

          if monster_inventory[idx_monster_di_inventory][2] < 5 then
            next_level ← monster_inventory[idx_monster_di_inventory][2] + 1
            output(monster[idx_monster_di_database][1], next_level)
            level ← monster_inventory[idx_monster_di_inventory][2]
            cost ← upgrade_cost(level)
            output(cost)

            bool2 ← True
            while bool2 = True do
              input(upgrade)
              if upgrade = 'y' then
                if user_coin >= cost then
                  monster_inventory[idx_monster_di_inventory][2] ←
next_level
                  user_coin ← user_coin - cost
                  output(monster[idx_monster_di_database][1], next_level)
                  bool2 ← False
                else
                  output("Maaf, OC Anda tidak mencukupi untuk melakukan
upgrade.")
                  bool2 ← False
              elif upgrade = 'n' then
                output("Upgrade dibatalkan.")
                bool2 ← False
              else
                output("Input tidak valid. Upgrade dibatalkan.")
            else
              output("Maaf, monster yang Anda pilih sudah memiliki level
maksimum.")

```

```

        output("Apakah kamu ingin melanjutkan upgrade?")
        output("1. Lanjut upgrade")
        output("2. Keluar")

        input(opt)
        while opt ≠ '1' and opt ≠ '2' do
            output("Perintah tidak dikenal!")
            output("1. Lanjut upgrade")
            output("2. Keluar")
            input(opt)

        If opt = '2' then
            output(user_coin)
            output("Sampai jumpa lagi di Lab!")
            bool <- False

        function_value <- [user_coin, monster_inventory]
        Return function_value

```

### 13. F12-Shop Management

**function** is\_integer(n:String)-> Boolean  
 { program memvalidasi apakah bilangan integer  
 I.S. input bilangan n  
 F.S. validasi berupa boolean }

#### KAMUS LOKAL

**{variables}**  
 cek : Boolean

#### ALGORITMA

```

    cek <- True

    Repeat for char in str(n):
        if ord(char) not in range(48 until 58) then
            cek <- False

```

**function** validasi\_int(n: string, tipe: string) -> string:  
 { program memvalidasi apakah bilangan integer kemudian meminta input hingga input  
 user berupa integer  
 I.S. input bilangan n dan cek  
 F.S. validasi berupa boolean dan nilai n }

#### KAMUS LOKAL

**{variables}**  
 cek : Boolean

#### ALGORITMA

```

    while (True) do
        if (is_integer(n) = True) then
            break
        Else
            output("Input yang dimasukkan tidak valid!")
            input(n)

```

```

procedure shop_management
    (input monster_shop: array of Monster_Shop, input item_shop: array of
    Item_Shop, input monster: array of Monster, input item_inventory: array of
    Item_Inventory)→ (output functions value: Hasil_Fungsi)
{ Program untuk mengelola stok dan harga barang di toko. User dapat melakukan
berbagai tindakan seperti menambah atau mengurangi stok monster dan item, serta
mengubah harga jualnya. Program ini membantu dalam pengaturan inventaris toko agar
sesuai dengan kebutuhan dan permintaan pengguna.
I.S. Data stok dan harga barang di toko, serta inventaris item pengguna terdefinisi.
F.S. Data stok dan harga barang di toko, serta inventaris item pengguna diperbarui
sesuai dengan tindakan penambahan yang dilakukan dalam fungsi ini. }

```

#### KAMUS LOKAL

```

{variables}
    list_item : list of string
    id_monster : list of integer
    aksi_user : string
    tipe_dilihat : string
    input_id_ditambah, stok_awal_ditambah, harga_ditambah : string
    data_ditambah : list of integer
    cek_valid : bool

```

#### ALGORITMA

```

input(AksiUser)
while (true) do
    ListItem <- ["strength", "resilience", "healing"]
    IDMonster <- []
    for I traversal [1..length(MonsterShop)] do
        append(IDMonster, MonsterShop[I,1])

    if (AksiUser = "lihat") then
        input(TipeDilihat)
        if (TipeDilihat = "monster") then
            output("ID      |Type      |ATK Power |DEF Power |HP      |Stok
            |Harga      |")

            output("-----|-----|-----|-----|-----|---
            -----|")
            for Row traversal [1..length(Monster)] do
                if (str(Monster[Row,1]) in str(IDMonster)) then
                    for Char traversal [1..5] do
                        length_char <- length(str(Monster[Row,Char]))
                        output(str(Monster[Row,Char]) + " "*(10-length_char),
                        terminator="|")
                    for Row2 traversal [1..length(MonsterShop)] do
                        if (str(MonsterShop[Row2,1]) = str(Monster[Row,1])) then
                            length_row2 <- length(str(MonsterShop[Row2,2]))
                            length_row3 <- length(str(MonsterShop[Row2,3]))
                            output(str(MonsterShop[Row2,2]) + " "*(10-length_row2)
                            + "|" +
                                str(MonsterShop[Row2,3]) + " "*(10-length_row3) +
                                "|", terminator="")
                            output("")
                        elif (TipeDilihat = "potion") then
                            output("ID |Type      |Stok      |Harga      |")
                            output("---|-----|-----|-----|")
                            I <- 1
                            for Row traversal [1..length(ItemShop)] do
                                output(I, terminator=" |")
                                I <- I + 1
                                for Char traversal [1..3] do
                                    length_char <- length(str(ItemShop[Row,Char]))
                                    output(str(ItemShop[Row,Char]) + " "*(15-length_char),
                                    terminator="|")
                                output("")
                            elif (TipeDilihat = "back") then
                                pass
                            else

```

```

        output("Input yang dimasukkan tidak valid!")

    elif (AksiUser = "tambah") then
        input(TipeDilihat)
        if (TipeDilihat = "monster") then
            output("ID      |Type      |ATK Power |DEF Power |HP      |")
            output("-----|-----|-----|-----|-----|")
            for Row traversal [1..length(Monster)] do
                if (str(Monster[Row,1]) not in str(IDMonster)) then
                    for Char traversal [1..5] do
                        length_char <- length(str(Monster[Row,Char]))
                        output(str(Monster[Row,Char]) + " "*(10-length_char),
                            terminator="|")
                    output("")
                output("Jika tidak jadi menambahkan, silahkan ketik 'cancel'")
                input(InputIDDitambah)
                CekValid <- false
                while (CekValid = false) do
                    if (is_integer(InputIDDitambah)) then
                        for Row traversal [1..length(Monster)] do
                            if (int(Monster[Row,1]) = int(InputIDDitambah)) and
                                (str(InputIDDitambah) not in str(IDMonster)) then
                                CekValid <- true
                            if (CekValid = false) then
                                output("ID yang kamu masukan tidak valid! ayo masukan id
                                lagi!")
                                input(InputIDDitambah)
                            elif (CekValid = true) then
                                input(StokAwalDitambah)
                                StokAwalDitambah <- validasi_int(StokAwalDitambah, "stok")
                                input(HargaDitambah)
                                HargaDitambah <- validasi_int(HargaDitambah, "harga")
                                data_ditambah <- [int(InputIDDitambah),
                                    int(StokAwalDitambah), int(HargaDitambah)]
                                append(MonsterShop, data_ditambah)
                                append(IDMonster, int(InputIDDitambah))
                                output("Monster berhasil ditambah!")
                            else
                                if (InputIDDitambah = "cancel") then
                                    break
                                else
                                    output("Input yang dimasukkan tidak valid!")
                                    input(InputIDDitambah)
                                end if
                            end if
                        end for
                    end if
                end while
            end if
        elif (TipeDilihat = "potion") then
            idx_i <- []
            output("ID |Type      ")
            output("---|-----")
            for I traversal [1..length(ListItem)] do
                if (str(ListItem[I]) not in str(ItemShop)) then
                    output(f"{I} | {ListItem[I]}")
                    append(idx_i, I)
                end if
            end for
            output("Jika tidak jadi menambahkan, silahkan ketik 'cancel'")
            input(InputIDTambahItem)
            CekValid <- false
            while (CekValid = false) do
                if (is_integer(InputIDTambahItem)) then
                    if (int(InputIDTambahItem) in idx_i) then
                        CekValid <- true
                        input(StokAwalDitambah)
                        StokAwalDitambah <- validasi_int(StokAwalDitambah, "stok")
                        input(HargaDitambah)
                        HargaDitambah <- validasi_int(HargaDitambah, "harga")
                        data_ditambah <- [ListItem[int(InputIDTambahItem)-1],
                            int(StokAwalDitambah), int(HargaDitambah)]
                        append(ItemShop, data_ditambah)
                        output("Item berhasil ditambahkan!")
                    else
                        output("ID yang kamu masukan tidak valid! ayo masukan id
                        lagi!")
                        input(InputIDTambahItem)
                    end if
                end if
            end while
        end if
    end if
end if

```

```

        InputIDTambahItem <- validasi_int(InputIDTambahItem, "id
        potion")
    else
        if (InputIDTambahItem = "cancel") then
            break
        else
            output("Input yang dimasukkan tidak valid")
            input(InputIDTambahItem)
            InputIDTambahItem <- validasi_int(InputIDTambahItem, "id
            potion")
    elif (TipeDilihat = "back") then
        pass
    else
        output("Input yang dimasukkan tidak valid!")

elif (AksiUser = "ubah") then
    input(TipeDiubah)
    if (TipeDiubah = "monster") then
        output("ID          |Type          |ATK Power |DEF Power |HP          |Stok
        |Harga          |")

        output("-----|-----|-----|-----|-----|---
        -----|-----|")
        for Row traversal [1..length(Monster)] do
            if (str(Monster[Row,1]) in str(IDMonster)) then
                for Char traversal [1..5] do
                    length_char <- length(str(Monster[Row,Char]))
                    output(str(Monster[Row,Char]) + " "*(10-length_char),
                    terminator="|")
                for Row2 traversal [1..length(MonsterShop)] do
                    if (str(MonsterShop[Row2,1]) = str(Monster[Row,1])) then
                        length_row2 <- length(str(MonsterShop[Row2,2]))
                        length_row3 <- length(str(MonsterShop[Row2,3]))
                        output(str(MonsterShop[Row2,2]) + " "*(10-length_row2)
                        + "|" +
                        str(MonsterShop[Row2,3]) + " "*(10-length_row3) +
                        "|", terminator="")
                    output("")
                output("Jika tidak jadi mengubah, silahkan ketik -1")
                input(InputIDDiubah)
                InputIDDiubah <- validasi_int(InputIDDiubah, "id monster")
                CekValid <- false
                while (CekValid = false) do
                    if (InputIDDiubah = "-1") then
                        break
                    else
                        for Row traversal [1..length(MonsterShop)] do
                            if (str(MonsterShop[Row,1]) = str(InputIDDiubah)) then
                                CekValid <- true
                            if (CekValid = false) then
                                output("ID yang kamu masukan tidak valid! ayo masukan id
                                lagi!")
                                input(InputIDDiubah)
                            elif (CekValid = true) then
                                input(StokBaruDiubah)
                                StokBaruDiubah <- validasi_int(StokBaruDiubah, "stok baru")
                                input(HargaBaruDiubah)
                                HargaBaruDiubah <- validasi_int(HargaBaruDiubah, "harga
                                baru")
                                for Row traversal [1..length(MonsterShop)] do
                                    if (str(MonsterShop[Row,1]) = str(InputIDDiubah)) then
                                        MonsterShop[Row,2] <- int(StokBaruDiubah)
                                        MonsterShop[Row,3] <- int(HargaBaruDiubah)
                                        output("Monster berhasil diubah!")
                        elif (TipeDiubah = "potion") then
                            idx_i <- []
                            output("ID |Type          |Stok          |Harga          |")
                            output("---|-----|-----|-----|")
                            I <- 1
                            for Row traversal [1..length(ItemShop)] do

```

```

output(I, terminator="|")
append(idx_i, I)
for Char traversal [1..3] do
    length_char <- length(str(ItemShop[Row,Char]))
    output(str(ItemShop[Row,Char]) + " "*(15-length_char),
        terminator="|")
    I <- I + 1
    output("")
output("Jika tidak jadi mengubah, silahkan ketik -1")
input(InputIDDiubah)
InputIDDiubah <- validasi_int(InputIDDiubah, "id potion")
CekValid <- false
while (CekValid = false) do
    if (InputIDDiubah = "-1") then
        break
    else
        if (int(InputIDDiubah) in idx_i) then
            CekValid <- true
            input(StokBaruDiubah)
            StokBaruDiubah <- validasi_int(StokBaruDiubah, "stok baru")
            input(HargaBaruDiubah)
            HargaBaruDiubah <- validasi_int(HargaBaruDiubah, "harga
            baru")
            ItemShop[int(InputIDDiubah)-1,2] <- int(StokBaruDiubah)
            ItemShop[int(InputIDDiubah)-1,3] <- int(HargaBaruDiubah)
            output("Item telah berhasil diubah!")
        else
            output("ID yang kamu masukan tidak valid! ayo masukan id
            lagi!")
            input(InputIDDiubah)
            InputIDDiubah <- validasi_int(InputIDDiubah, "id potion")
        elif (TipeDilihat = "back") then
            pass
        else
            output("Input yang dimasukkan tidak valid!")
    end if
end while

elif (AksiUser = "hapus") then
    input(TipeDihapus)
    if (TipeDihapus = "monster") then
        output("ID          |Type          |ATK Power |DEF Power |HP          |Stok
        |Harga          |")

        output("-----|-----|-----|-----|-----|---
        -----|-----|")
        for Row traversal [1..length(Monster)] do
            if (str(Monster[Row,1]) in str(IDMonster)) then
                for Char traversal [1..5] do
                    length_char <- length(str(Monster[Row,Char]))
                    output(str(Monster[Row,Char]) + " "*(10-length_char),
                        terminator="|")
                end for
                for Row2 traversal [1..length(MonsterShop)] do
                    if (str(MonsterShop[Row2,1]) = str(Monster[Row,1])) then
                        length_row2 <- length(str(MonsterShop[Row2,2]))
                        length_row3 <- length(str(MonsterShop[Row2,3]))
                        output(str(MonsterShop[Row2,2]) + " "*(10-length_row2)
                            + "|" +
                                str(MonsterShop[Row2,3]) + " "*(10-length_row3) +
                                "|", terminator="")
                    end if
                end for
                output("")
            end if
        end for
        output("Jika tidak jadi menghapus, silahkan ketik -1")
        input(InputIDDihapus)
        InputIDDihapus <- validasi_int(InputIDDihapus, "id monster")
        CekValid <- false
        while (CekValid = false) do
            if (InputIDDihapus = "-1") then
                break
            else
                for Row traversal [1..length(MonsterShop)] do
                    if (str(InputIDDihapus) = str(MonsterShop[Row,1])) then
                        CekValid <- true
                    end if
                end for
            end if
        end while
    end if
end if

```



```

        if (CekValid = false) then
            output("ID yang kamu masukan tidak valid! ayo masukan id
            lagi!")
            input(InputIDDihapus)
            InputIDDihapus <- validasi_int(InputIDDihapus, "id
            monster")
        elif (CekValid = true) then
            input(ValidasiInputHapus)
            if (ValidasiInputHapus = "y") then
                for Row traversal [1..length(MonsterShop)] do
                    if (str(MonsterShop[Row,1]) = str(InputIDDihapus))
                then
                    remove(MonsterShop, Row)
                    output("Monster berhasil dihapus!")
                    break
                elif (ValidasiInputHapus = "n") then
                    output("Baik!")
                    pass
            elif (TipeDihapus = "potion") then
                idx_i <- []
                output("ID |Type          |Stok          |Harga          |")
                output("----|-----|-----|-----|")
                I <- 1
                for Row traversal [1..length(ItemShop)] do
                    output(I, terminator=" |")
                    append(idx_i, I)
                    for Char traversal [1..3] do
                        length_char <- length(str(ItemShop[Row,Char]))
                        output(str(ItemShop[Row,Char]) + " "*(15-length_char),
                        terminator="|")
                    I <- I + 1
                    output("")
                output("Jika tidak jadi menghapus, silahkan ketik -1")
                input(InputIDDihapus)
                InputIDDihapus <- validasi_int(InputIDDihapus, "id potion")
                CekValid <- false
                while (CekValid = false) do
                    if (InputIDDihapus = "-1") then
                        break
                    else
                        if (int(InputIDDihapus) in idx_i) then
                            CekValid <- true
                            input(ValidasiInputHapus)
                            if (ValidasiInputHapus = "y") then
                                remove(ItemShop, int(InputIDDihapus)-1)
                                output("Berhasil dihapus!")
                            elif (ValidasiInputHapus = "n") then
                                pass
                        else
                            output("ID yang kamu masukan tidak valid! Ayo masukan id
                            lagi!")
                            input(InputIDDihapus)
                            InputIDDihapus <- validasi_int(InputIDDihapus, "id potion")
                elif (TipeDilihat = "back") then
                    pass
                else
                    output("Input yang dimasukkan tidak valid!")

            elif (AksiUser = "keluar") then
                sleep(1)
                output('Sampai jumpa lagi, Admin!\n')
                break
            else
                output("Input yang dimasukkan salah :( Ayo input yang benar!")

            input(AksiUser)

```

```
FunctionValues <- [MonsterShop, ItemShop, Monster, ItemInventory]
```

## 14. F13-Monster Management

```
procedure monster_management
    (input monster : Array of string) →
    (output monster : Array of String)
{ Program mengubah data kumpulan monster sesuai dengan yang diubah oleh admin }
I.S. file data monster yang terdefinisi
F.S. Data monster yang telah diubah }
```

### KAMUS LOKAL

```
switch: integer
input_fitur: string
row: list
char: string
length_char: integer
input_nama_monster: string
cek_file: bool
found: bool
input_atk_power: string
Cek_Integer: bool
input_def_power: string
Cek_Rentang: bool
input_HP: string
id_monster: integer
validasi_database: string
monster_baru: list
```

### ALGORITMA

```
output ( "----SELAMAT DATANG DI MENU DATA MONSTER MANAGEMENT----" )
While (True) do
    output("""-----SILAHKAN PILIH FITUR----- \n)
        1. Tampilkan Monster yang anda miliki \n
        2. Tambahkan Monster Baru \n
        3. Keluar""")
    Input (input_fitur) {1/2/3}
    If (input_fitur = "1") then { Kondisi Apabila memilih fitur lihat }
        Iterate for row in monster
            Iterate for char in monster
                Output( (char) + " "(15-length_char), end="|")
            Output()

        Depends on (input_fitur = "2") then { Kondisi Apabila memilih fitur
    beli }
        {input Nama Monster}
        Input (input_nama_monster)
        { Program Mengecek apakah terdapat nama file yang sama dalam
    file }
        Cek_file <- True
        While (Cek_file = True) do
            Found <- False
            for row in monster:
                if (str(input_nama_monster) == str(row[1])) then
                    found <- True
                    cek_file <- False
                    output(break)
            if (found) then
                output("Nama Sudah Terdaftar, Silahkan Coba Lagi!!")
                input (input_nama_monster)
            else : output (break)

            {input ATK Monster}
            input(input_atk_power)
            Cek_Integer <- True
            while (Cek_Integer == True) , do
                if (function is_integer(input_atk_power)) == False:
```

```

        output("Masukkan input bertipe Integer, coba lagi!")
        input (input_atk_power)
    else :
        Cek_Integer <- False

    {input DEF Monster}
    Input ( input_def_power )
    Cek_Integer <- True
    while (Cek_Integer == True) , do
    if (is_integer(input_def_power) == False) then
        output("Masukkan input bertipe Integer, coba lagi!")
        input (input_def_power)
    else :
        Cek_Integer <- False
    Break
    Cek_Rentang <- True
    while (Cek_Rentang == True) do
    if (not(0 <= int(input_def_power) <= 50)) then
        output("INPUT SALAH!! DEF Power harus bernilai 0-50, coba
lagi!")
        input (input_def_power)
    else :
        Cek_Rentang <- False
    {input HP Monster}
    input (input_HP)
    Cek_Integer <- True
    while (Cek_Integer == True) do
    if (is_integer(input_HP) == False) then
        output("Masukkan input bertipe Integer, coba lagi!")
        input (input_HP)
    else :
        Cek_Integer <- False
    Output(monster_baru <- [input_nama_monster , input_atk_power ,
        input_def_power , input_HP])
    Input(validasi_database) {Y/N}
    if (validasi_database == "Y") then
        monster <- [monster_baru]
    else if (validasi_database == "N") then
        output("Penambahan Monster Dibatalkan!!\n")
    else :
        output("Input tidak valid , Penambahan dibatalkan\n")
    Depends on (input_fitur = "3") then          { Kondisi Apabila memilih fitur
keluar }
        Break
    Else {Return to Loop}
    Output(Return value of monster)

```

**function** is\_integer(n:String)-> Boolean  
 { program memvalidasi apakah bilangan integer  
 I.S. input bilangan n  
 F.S. validasi berupa boolean }

#### KAMUS LOKAL

**{variables}**  
 cek : Boolean

#### ALGORITMA

```

cek <- True

Repeat for char in str(n):
    if ord(char) not in range(48 until 58) then
        cek <- False

```

```
function validasi_int(n: string, tipe: string) -> string:
{ program memvalidasi apakah bilangan integer kemudian meminta input hingga input
user berupa integer
I.S. input bilangan n dan cek
F.S. validasi berupa boolean dan nilai n }
```

#### KAMUS LOKAL

```
{variables}
cek : Boolean
```

#### ALGORITMA

```
while (True) do
    if (is_integer(n) = True) then
        break
    Else
        output("Input yang dimasukkan tidak valid!")
        input(n)
```

## 15. F14-Load

```
procedure load()
{ program membaca file csv sesuai folder yang diinput oleh user dan mengeluarkan data
dalam bentuk matriks
I.S. file terdefinisi
F.S. data di csv dikeluarkan }
```

#### KAMUS LOKAL

```
{variables}
parser : argparse.ArgumentParser
nilai_fungsi : array of any
User, userMonster, monster_shop, monster_inventory, item_shop, item_inventory
: array of any
```

#### ALGORITMA

```
user ← []
monster ← []
monster_shop ← []
monster_inventory ← []
item_shop ← []
item_inventory ← []
peta ← []

if os.path.exists(nama_folder) then
    time.sleep(0.5)
    output("Loading...")
    time.sleep(1)
    files_di_folder ← os.listdir(nama_folder)
    file traversal files_di_folder do
        nama_file ← nama_folder + "/" + file
        depend on (nama_file)
            nama_folder + "/user.csv" :
                user ← ParseCSV(nama_file)
            nama_folder + "/monster.csv" :
                monster ← ParseCSV(nama_file)
            nama_folder + "/monster_shop.csv" :
                monster_shop ← ParseCSV(nama_file)
            nama_folder + "/monster_inventory.csv" :
                monster_inventory ← ParseCSV(nama_file)
```

```

        nama_folder + "/item_shop.csv" :
            item_shop ← ParseCSV(nama_file)
        nama_folder + "/item_inventory.csv" :
            item_inventory ← ParseCSV(nama_file)
        nama_folder + "/peta.csv" :
            peta ← ParseCSV(nama_file)

    → True, user, monster, monster_shop, monster_inventory, item_shop,
       item_inventory, peta
else
    output("Yah, folder \"", nama_folder, "\"" tidak dapat ditemukan.")
    → False

```

#### ALGORITMA

```

{algoritma utama}
    sudah_bisa_login ← False
    while True do
        args ← argparse.ArgumentParser(description = "OWCA Program - Load")
        args.add_argument("nama_folder", nargs = "?", help = "Nama folder
untuk          memuat data")
        args ← args.parse_args()

        if args.nama_folder = "" then
            output("Tidak ada nama folder yang diberikan! Harap masukkan nama
            folder!")
            break
        else
            nilai_fungsi ← load(args.nama_folder)
            if nilai_fungsi[0] = True then
                user ← nilai_fungsi[1]
                monster ← nilai_fungsi[2]
                monster_shop ← nilai_fungsi[3]
                monster_inventory ← nilai_fungsi[4]
                item_shop ← nilai_fungsi[5]
                item_inventory ← nilai_fungsi[6]
                peta ← nilai_fungsi[7]
                sudah_bisa_login ← True
                break
            else
                break

```

```

function parse_csv(file_path : string) -> array of array of string
{ membaca file csv dan mengubah data menjadi array of array of string }
I.S. file terdefinisi
F.S. data di dikeluarkan }

```

#### KAMUS LOKAL

```

{variables}
    file : SEQFILE of string

```

#### ALGORITMA

```

    data ← []
    assign(file, file_path)
    reset(file)

    line traversal file do
        baris ← []
        temp_add ← ""
        readln(file, line)

        char traversal [1..length(line)] do
            if line[char] = ";" or line[char] = '\n' then
                baris ← baris + [temp_add]

```

```

        temp_add ← ""
        else
            temp_add ← temp_add + line[char]

        if temp_add ≠ "" then
            baris ← baris + [temp_add]
            data ← data + [baris]

    data.pop(0)
    → data

```

## 16. F15-Save

**procedure** save()  
 { program menyimpan file csv ke dalam folder yang diinput oleh user  
 I.S. user, userMonster, monster\_shop, monster\_inventory, item\_shop, item\_inventory  
 terdefinisi  
 F.S. seluruh file berhasil tersimpan }

### KAMUS LOKAL

**{variables}**  
 nama\_folder : string

### ALGORITMA

```

    output(">>> Save")
    output("Masukkan nama folder untuk penyimpanan: ")
    input(nama_folder)
    save_data (nama_folder, user, monster, monster_shop, monster_inventory,
    item_shop, item_inventory)

```

**function** save\_data(nama\_folder : string , user : array of any , userMonster : array of any , monster\_shop : array of any , monster\_inventory : array of any , item\_shop : array of any , item\_inventory : array of any) → string  
 { menyimpan data yang diterima ke csv dalam sebuah folder yang telah/baru dibuat  
 I.S. nama\_folder, user, userMonster, monster\_shop, monster\_inventory, item\_shop, item\_inventory terdefinisi  
 F.S. seluruh file tersimpan di dalam nama\_folder }

### KAMUS LOKAL

**{variables}**  
 Parent\_save, folder, data\_default : string  
 files\_di\_folder : array of string  
 file : SEQFILE of string

### ALGORITMA

```

    parent_save ← "savedata"
    folder ← os.path.join(parent_save, nama_folder)

    if not os.path.exists(folder) then
        output("")
        time.sleep(1)
        output("Membuat folder ", folder, "...")
        os.makedirs(folder)
    else
        output("")
        time.sleep(1)
        output("Folder ", folder, " sudah ada.")

```

```

files_di_folder ← os.listdir(folder)
if_files_di_folder ≠ [] then
  output("")
  time.sleep(1)
  output("Saving...")
  time.sleep(1)
  output("Berhasil menyimpan data di folder ", folder)
else
  output("")
  time.sleep(1)
  output("Saving...")
  time.sleep(1)
  output("Berhasil menyimpan data di folder ", folder)

assign(file, os.path.join(folder, "user.csv"))
rewrite(file)
data_default ← "id;username;password;role;oc"
write(file, data_default)
data_traversal user do
  write(file, "\n" )
  write(";" .join(map(str, data)))

assign(file, os.path.join(folder, "monster.csv"))
rewrite(file)
data_default ← "id;type;atk_power;def_power;hp"
write(file, data_default)
data_traversal monster do
  write(file, "\n" )
  write(";" .join(map(str, data)))

assign(file, os.path.join(folder, "monster_shop.csv"))
rewrite(file)
data_default ← "monster_id;stock;price"
write(file, data_default)
data_traversal monster_shop do
  write(file, "\n" )
  write(";" .join(map(str, data)))

assign(file, os.path.join(folder, "monster_inventory.csv"))
rewrite(file)
data_default ← "user_id;monster_id;level"
write(file, data_default)
data_traversal monster_inventory do
  write(file, "\n" )
  write(";" .join(map(str, data)))

assign(file, os.path.join(folder, "item_shop.csv"))
rewrite(file)
data_default ← "type;stock;price"
write(file, data_default)
data_traversal item_shop do
  write(file, "\n" )
  write(";" .join(map(str, data)))

assign(file, os.path.join(folder, "item_inventory.csv"))
rewrite(file)
data_default ← "user_id;type;quantity"
write(file, data_default)
data_traversal item_inventory do
  write(file, "\n" )
  write(";" .join(map(str, data)))

```

## 17. F16-Exit

```

procedure exit()
{ program untuk keluar dari permainan setelah menerima inputan apakah user ingin save
file atau tidak

```

I.S. program berjalan F.S. keluar dari program }
<b>KAMUS LOKAL</b>  <b>{variables}</b> pilihan_user : string
<b>ALGORITMA</b> <u>output</u> ("Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): "), <u>input</u> (pilihan_user)  <u>repeat</u> <u>depend on</u> (pilihan_user) 'y' : -> True 'n' : <u>output</u> ("Kembali ke program.") -> False else: <u>output</u> ("Pilihan tidak valid!") <u>output</u> ("Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): "), <u>input</u> (pilihan_user)

## 18. B04-JACKPOT!

<b><u>function</u></b> jackpot ( <u>input</u> user_id: string, <u>input</u> monster_inventory: Monster_Inventory, <u>input</u> koin: integer) -> ( <u>output</u> functions_value: Functions_Value) { Menghasilkan hadiah yang dapat diterima oleh pengguna dengan mengundi} { I.S menginput user_id, database monster_inventory, dan banyak koin yang dimiliki user} {F.S menghasilkan database baru yang telah terupdate dengan hadiah yang diperoleh user}
<b>KAMUS LOKAL</b>  <b>{type}</b> <u>type</u> Monster_Inventory : < user_id: integer; monster_id: integer; level: integer >  <b>{variables}</b> user_id: string monster_inventory: Monster_Inventory functions_value: Functions_Value input_user: string item_satu,item_dua,item_tiga: integer koin,koin_tambahan,koin_didapat: integer  <b>{functions}</b> <u>function</u> LCG(seed : integer) → integer <u>function</u> RandomNumber(range : array of integer) → integer
<b>ALGORITMA</b>  input(UserID) input(MonsterInventory) input(Koin) KoinDidapat <- 0 item_jackpot <- [{"Topi",50}, {"Pedang",100}, {"Koin",200}, {"Potion",300}, {"Monster",500}] <u>output</u> ("") input(InputUser) while (InputUser != "n") do if (InputUser = "y") then



```

output("")
output("Koin kamu saat ini adalah ", Koin, " OC")
if (Koin < 400) then
    output("Maaf, anda tidak memiliki cukup OC untuk bermain JACKPOT.")
    break
else
    Koin <- Koin - 400
    ItemSatu <- ((F00.random_number())^2 + 1) mod 5
    ItemDua <- ((F00.random_number())^3 + 2) mod 5
    ItemTiga <- ((F00.random_number())^5 + 3) mod 5
    output("$$$$$$$$$ ", ItemJackpot[ItemSatu,1], " | ",
ItemJackpot[ItemDua,1], " | ", ItemJackpot[ItemTiga,1], " $$$$$$$$$$")
    if (ItemSatu = ItemDua) and (ItemDua = ItemTiga) then
        output("JACKPOTTT!!! Monster Luffy telah dimasukkan dalam
inventory anda!")
        DataDitambah[1] <- UserID
        DataDitambah[2] <- 6132743
        DataDitambah[3] <- 1
        append(MonsterInventory, DataDitambah)
    else
        KoinTambahan <- ItemJackpot[ItemSatu,2] + ItemJackpot[ItemDua,2] +
ItemJackpot[ItemTiga,2]
        KoinDidapat <- KoinDidapat + KoinTambahan
        Koin <- Koin + KoinTambahan
        output(KoinTambahan, " OC telah ditambahkan ke akun Anda!")
        input(InputUser)
    else
        output("Kamu salah input!")
        input(InputUser)
output("Kamu mendapat koin sebesar ", KoinDidapat, " OC")
output("Koin kamu saat ini adalah ", Koin, " OC!")
output("Selamat tinggal!")
FunctionValues[1] <- MonsterInventory
FunctionValues[2] <- Koin

```

## 19. B05-Peta Kota Danville

**procedure** peta\_danville (input peta: array of string)

**KAMUS LOKAL**

**{variables}**  
peta: array of string  
i,j: integer

**ALGORITMA**

```

input(Peta)
output("")
for I traversal [1..12] do
    for J traversal [1..12] do
        if (Peta[I,J] = "#") then
            output(" ", terminator=" ")
        else
            output(Peta[I,J], terminator=" ")
    output("")

```

**function** bush (input peta: array of string) -> (output bush: array of string)

**KAMUS LOKAL**

**{variables}**  
peta: array of string

```
bush: array of string  
i,j: integer
```

#### **ALGORITMA**

```
input(Peta)  
Bush <- []  
for I traversal [1..12] do  
  for J traversal [1..12] do  
    if (Peta[I,J] = "X") and (Peta[I,J] not in Bush) then  
      for I2 traversal [I-1..I+1] do  
        for J2 traversal [J-1..J+1] do  
          append(Bush, [I2,J2])
```

**function** komando (input user\_id: string, input username: string, input monster\_shop: Monster\_Shop, input item\_shop: Item\_Shop, input monster: Monster, input monster\_inventory: Monster\_Inventory, input item\_inventory: Item\_Inventory, input coin: integer) -> (output: function\_value: Funtion\_Value)

#### **KAMUS LOKAL**

##### **{types}**

```
type Monster: < id: integer;  
                type: string;  
                atk_power: int;  
                def_power: integer;  
                hp: integer>  
type Monster_Shop: < monster_id: integer;  
                    stock: integer;  
                    price: integer>  
type Item_Shop: < type: string;  
                stock: integer;  
                price: integer >  
type Monster: < id: integer;  
                type: string;  
                atk_power: int;  
                def_power: integer;  
                hp: integer>  
type Monster_Inventory : < user_id: integer;  
                           monster_id: integer;  
                           level: integer>  
  
type Item_Inventory: < user_id: integer;  
                      type: string;  
                      quantity: integer >  
type Function_Value: < monster_shop: array of Monster_Shop;  
                      item_shop: array of Item_Shop;  
                      monster: array of Monster;  
                      monster_inventory: array of Monster_Inventory;  
                      item_inventory : array of Item_Inventory;  
                      coin: integer >
```

##### **{variables}**

```
monster: Monster  
monster_shop: Monster_Shop  
item_shop: Item_Shop  
monster: Monster  
monster_inventory: Monster_Inventory  
item_inventory: Item_Inventory  
function_value: Function_Value  
obstacle: array of string  
command: string  
idx_baris,idx_kolom: integer  
function_value: Function_Value
```

##### **{functions}**



```

        IdxBaris <- IdxBaris - 1
        output("\nAgent akan pindah ke atas!")
    elif (Command = "s") then
        if (Peta[IdxBaris+1,IdxKolom] in Obstacle) then
            output("\nAgent tidak bisa pindah karena terdapat Obstacle!")
        else
            Peta[IdxBaris,IdxKolom] <- "#"
            Peta[IdxBaris+1,IdxKolom] <- "P"
            IdxBaris <- IdxBaris + 1
            output("\nAgent akan pindah ke bawah!")
        elif (Command = "shop") then
            if (IdxBaris in range(1..4)) and (IdxKolom in range(5..8)) then
                output("\nLoading...")
                sleep(1)
                FunctionValue <- Shop(UserID, MonsterShop, ItemShop, Monster,
MonsterInventory, ItemInventory, Coin)
                MonsterShop <- FunctionValue[1]
                ItemShop <- FunctionValue[2]
                Monster <- FunctionValue[3]
                MonsterInventory <- FunctionValue[4]
                ItemInventory <- FunctionValue[5]
                Coin <- FunctionValue[6]
                sleep(2)
                output("\nLoading...")
                sleep(2)
            else
                output("\nAgent tidak berada di area Shop!")
            elif (Command = "battle") then
                if ([IdxBaris,IdxKolom] in BattleBush) then
                    UserMonsters <- UserMonsters(UserID, Monster, MonsterInventory)
                    UserPotions <- UserPotions(UserID, ItemInventory)
                    output("\nLoading...")
                    sleep(1)
                    (UserPotions, Bonus) <- Battle(Monster, UserMonsters, UserPotions,
Username)
                    Coin <- Coin + Bonus
                    ItemInventory <- ItemInventory(UserID, ItemInventory, UserPotions)
                    sleep(2)
                    output("\nLoading...")
                    sleep(2)
                else
                    output("\nAgent tidak berada di area Battle!")
                elif (Command = "laboratory") then
                    if (IdxBaris in range(7..10)) and (IdxKolom in range(1..3)) then
                        output("\nLoading...")
                        sleep(1)
                        FunctionValue <- Laboratory(UserID, Coin, Monster, MonsterInventory)
                        Coin <- FunctionValue[1]
                        MonsterInventory <- FunctionValue[2]
                        sleep(2)
                        output("\nLoading...")
                        sleep(2)
                    else
                        output("\nAgent tidak berada di area Laboratory!")
                    elif (Command = "jackpot") then
                        if (IdxBaris in range(9..12)) and (IdxKolom in range(4..7)) then
                            output("\nLoading...")
                            sleep(1)
                            FunctionValue <- Jackpot(UserID, MonsterInventory, Coin)
                            MonsterInventory <- FunctionValue[1]
                            Coin <- FunctionValue[2]
                            sleep(2)
                            output("\nLoading...")
                            sleep(2)
                        else
                            output("\nAgent tidak berada di area JACKPOT!")
                    elif (Command = "arena") then
                        if (IdxBaris in range(7..10)) and (IdxKolom in range(8..11)) then
                            UserMonsters <- UserMonsters(UserID, Monster, MonsterInventory)
                            UserPotions <- UserPotions(UserID, ItemInventory)

```

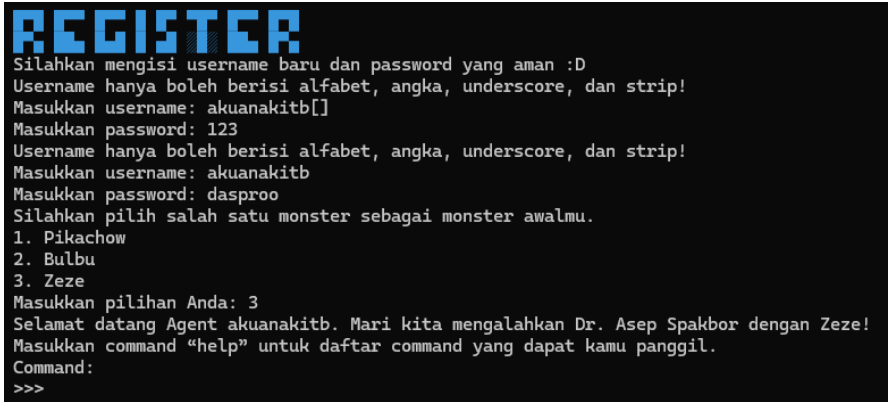
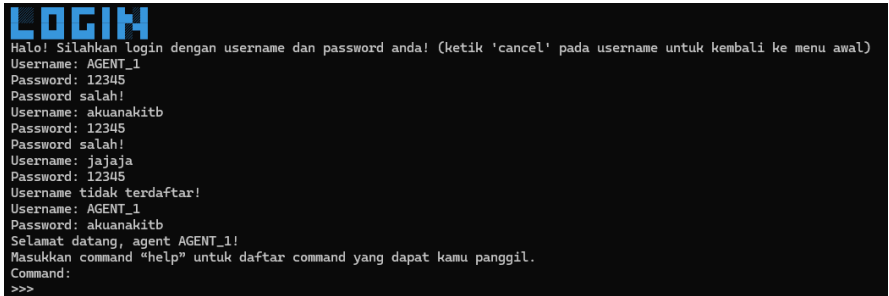
```

        output("\nLoading...")
        sleep(1)
        (UserPotions, Bonus) <- Arena(UserMonsters, UserPotions, Username,
Monster)
        Coin <- Coin + Bonus
        ItemInventory <- ItemInventory(UserID, ItemInventory, UserPotions)
        sleep(2)
        output("\nLoading...")
        sleep(2)
    else
        output("\nAgent tidak berada di area Arena!")
    elif (Command = "inventory") then
        UserMonsters <- UserMonsters(UserID, Monster, MonsterInventory)
        UserPotions <- UserPotions(UserID, ItemInventory)
        TampilInventoryUser(UserMonsters, UserPotions, UserID, Username, Coin)
        sleep(2)
        output("\nLoading...")
        sleep(2)
    elif (Command = "owcadex") then
        OwcaDex(Monster)
        sleep(2)
        output("\nLoading...")
        sleep(2)
    elif (Command = "exit") then
        if (IdxBaris in range(1..4)) and (IdxKolom in range(9..12)) then
            input(ValidasiKeluar)
            if (ValidasiKeluar = "y") then
                break
            else
                output("\nAgent tidak berada di area Exit!")
        else
            output("Command yang kamu masukkan tidak valid!")
            sleep(1)

FunctionValue <- [MonsterShop, ItemShop, Monster, MonsterInventory,
ItemInventory, Coin]

```

## Hasil Pengujian Program

Fitur	Pengujian Program
F01-Register	 <p>Gambar 01. Tampilan awal register user</p>
F02-Login	 <p>Gambar 02. Tampilan awal login user terdaftar</p>

F03-Logout	<div data-bbox="548 212 1422 296"><pre>Command: &gt;&gt;&gt; 1 Apakah Anda yakin ingin logout? Semua file yang tidak di save akan terhapus! (y/n): y SAMPAI JUMPA KING!</pre></div> <div data-bbox="792 306 1169 342">Gambar 03. Tampilan logout</div>
F04-Menu & Help	<div data-bbox="548 373 1422 583"><pre>&gt;&gt;&gt; help ===== HELP ===== Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu. 1. Register: Membuat akun baru 2. Login: Masuk ke dalam akun yang sudah terdaftar Footnote: 1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar 2. Jangan lupa untuk memasukkan input yang valid =====</pre></div> <div data-bbox="574 594 1383 630">Gambar 04. Tampilan detail Menu &amp; Help bagi user tanpa role</div> <div data-bbox="548 632 1422 930"><pre>&gt;&gt;&gt; help ===== HELP ===== Halo Admin! Berikut adalah hal-hal yang dapat kamu lakukan: 1. Logout: Keluar dari akun yang sedang digunakan 2. Shop Management: Mengatur ketersediaan monster dan potion pada shop 3. Monster Management: Mengatur monster dalam database 4. Save data 5. Selesai Footnote: 1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar 2. Jangan lupa untuk memasukkan input yang valid ===== Command:</pre></div> <div data-bbox="623 940 1334 976">Gambar 05. Tampilan detail Menu &amp; Help bagi Admin</div> <div data-bbox="548 978 1422 1136"><pre>===== HELP ===== Halo Agent!. Kamu memanggil command Help. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang: 1. Logout: Keluar dari akun yang sedang digunakan 2. Monster: Melihat monster yang dimiliki oleh Agent 3. Potion: Melihat potion yang dimiliki oleh Agent 4. Mulai bermain! 5. Save data 6. Selesai bermain Footnote: 1. Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar 2. Jangan lupa untuk memasukkan input yang valid =====</pre></div> <div data-bbox="630 1146 1328 1182">Gambar 06. Tampilan detail Menu &amp; Help bagi Agent</div>

F05-Monster

```
Masukkan command: owcadex
```

```
===== OWCA DEX =====
```

1. Pikachow
2. Bulbu
3. Zeze
4. Zuko
5. Chacha

Ketik monster id untuk melihat detail.  
Ketik 0 untuk keluar

```
>>> Pilih: █
```

Gambar 07. Tampilan utama owcadex

```
>>> Pilih: 2
```

```
===== BULBU LVL 1 =====
```

```
Monster id   : 2  
ATK Power    : 50  
DEF Power    : 50  
HP           : 1200
```

Ketik 1-5 untuk melihat status monster pada level tersebut.  
Ketik 0 untuk kembali ke OWCA DEX.

```
>>> Pilih: █
```

Gambar 08. Tampilan detail monster awal

```
>>> Pilih: 6
```

```
[Peringatan] Pilihan tidak sesuai!
```

```
===== BULBU LVL 1 =====
```

```
Monster id   : 2  
ATK Power    : 50  
DEF Power    : 50  
HP           : 1200
```

Ketik 1-5 untuk melihat status monster pada level tersebut.  
Ketik 0 untuk kembali ke OWCA DEX.

```
>>> Pilih: █
```



	<p>Gambar 09. Tampilan input tidak valid pada owcadex</p> <pre> &gt;&gt;&gt; Pilih: 4  ===== BULBU LVL 4 ===== Monster id   : 2 ATK Power    : 65 DEF Power    : 50 HP           : 1560  Ketik 1-5 untuk melihat status monster pada level tersebut. Ketik 0 untuk kembali ke OWCA DEX.  &gt;&gt;&gt; Pilih:  </pre> <p>Gambar 10. Tampilan detail monster sesuai level pilihan</p>
F06-Potion	<pre> &gt;&gt;&gt; Pilih: 2  Pilih potion: 1. Resilience Potion (Qty: 1  Uses: 1) 2. Cancel  &gt;&gt;&gt; Pilih:  </pre> <p>Gambar 11. Tampilan awal penggunaan potion</p> <pre> &gt;&gt;&gt; Pilih: 1  Efek: DEF Power = 35 --&gt; 36  &gt;&gt;&gt; Yakin ingin menggunakan potion ini? (y/n)  </pre>

Gambar 12. Tampilan efek peningkatan

```
>>> Yakin ingin menggunakan potion ini? (y/n) 1  
[Peringatan] Pilihan tidak sesuai!  
Efek:  
DEF Power = 35 --> 36  
  
>>> Yakin ingin menggunakan potion ini? (y/n) 
```

Gambar 13. Tampilan input tidak valid saat penggunaan potion

```
>>> Yakin ingin menggunakan potion ini? (y/n) n  
  
>>> Mau gunakan potion lain? (y/n) y  
  
Pilih potion:  
1. Resilience Potion (Qty: 1| Uses: 1)  
2. Cancel  
  
>>> Pilih: 
```

Gambar 14. Tampilan penggunaan potion lain

```
>>> Pilih: 3  
  
[Peringatan] HP masih penuh.  
Tidak bisa menggunakan Healing potion.  
  
>>> Mau gunakan potion lain? (y/n) 
```

Gambar 15. Tampilan healing potion jika hp masih penuh

```
>>> Pilih: 2  
  
[Peringatan] DEF monster sudah mencapai batas maksimum.  
Tidak bisa menggunakan Resilience potion.  
  
>>> Mau gunakan potion lain? (y/n) 
```

Gambar 16. Tampilan resilience potion jika def sudah 50

F07-Inventory

```
Masukkan command: inventory

===== RADHI'S INVENTORY (User ID: 6) =====
Jumlah O.W.C.A Coin mu sekarang adalah 90889
1. Monster      (Name: Pikachow, Lvl: 3, HP: 720)
2. Monster      (Name: Bulbu, Lvl: 4, HP: 1560)
3. Monster      (Name: Zuko, Lvl: 5, HP: 1120)
4. Potion       (Type: DEF, Qty: 1)
5. Potion       (Type: Heal, Qty: 1)

Ketik id untuk melihat detail
Ketik 0 untuk keluar

>>> Pilih: 0
```

Gambar 17. Tampilan utama inventory

```
>>> Pilih: k

[Peringatan] Pilihan tidak sesuai!

===== RADHI'S INVENTORY (User ID: 6) =====
Jumlah O.W.C.A Coin mu sekarang adalah 90889
1. Monster      (Name: Pikachow, Lvl: 3, HP: 720)
2. Monster      (Name: Bulbu, Lvl: 4, HP: 1560)
3. Monster      (Name: Zuko, Lvl: 5, HP: 1120)
4. Potion       (Type: DEF, Qty: 1)
5. Potion       (Type: Heal, Qty: 1)

Ketik id untuk melihat detail
Ketik 0 untuk keluar

>>> Pilih: 0
```

Gambar 18. Tampilan input tidak valid dalam inventory

```
>>> Pilih: 2

===== BULBU LVL 4 =====
Monster id   : 2
ATK Power    : 65
DEF Power    : 50
HP           : 1560

Ketik 1 untuk kembali ke inventory
Ketik 0 untuk keluar

>>> Pilih: 
```

Gambar 19. Tampilan detail monster

```
>>> Pilih: 4

===== RESILIENCE POTION =====
(Dimiliki: 1)

Potion ini memberikan efek peningkatan DEF Power sebesar 5%. Item ini
hanya dapat digunakan sekali dalam battle. Tidak dapat digunakan jika
DEF Power monster sudah mencapai maksimum (50).

Ketik 1 untuk kembali ke inventory
Ketik 0 untuk keluar

>>> Pilih: 
```

Gambar 20. Tampilan detail potion

F08-Battle

```
Masukkan command: battle

Loading...

BATTLE!

Battle akan dimulai dalam...
3
2
1
```

Gambar 21. Tampilan awal battle

```
RAAWRRRR!!! Monster Bulbu telah muncul!!!

Name      : Bulbu
ATK Power : 50
Def Power : 50
HP        : 1200
Level     : 1

Pilih Monster-mu untuk bertarung!

===== MONSTER LIST =====
1. Pikachow
2. Bulbu
3. Zuko

>>> Pilih: 
```

Gambar 22. Tampilan kemunculan lawan dan pilih monster

```
>>> Pilih: 2

RAAWRRRR!!! Radhi mengeluarkan monster Bulbu!!!

Name      : Bulbu
ATK Power : 65
Def Power : 50
HP        : 1560
Level     : 4

===== TURN 1 (BULBU - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 23. Tampilan kemunculan monster dan turn 1

```
>>> Pilih: 1

(RADHI) Bulbu menyerang!!!

(ENEMY) Bulbu's HP = 1172

===== Attack Detail =====
Attack Power : 55 (-15%)
Reduced by   : 27 (-50%)
Result      : 28

===== TURN 2 (BULBU - ENEMY) =====

(ENEMY) Bulbu menyerang!!!

(RADHI) Bulbu's HP = 1538

===== Attack Detail =====
Attack Power : 43 (-13%)
Reduced by   : 21 (-50%)
Result      : 22

===== TURN 3 (BULBU - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 24. Tampilan attack dan detailnya

```
>>> Pilih: 5

[Peringatan] Pilihan tidak sesuai!

===== TURN 3 (BULBU - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 25. Tampilan input tidak valid dalam battle

```
>>> Pilih: 2

Pilih potion:
1. Resilience Potion (Qty: 1| Uses: 1)
2. Healing Potion (Qty: 1| Uses: 1)
3. Cancel

>>> Pilih: 
```

Gambar 26. Tampilan pilih potion

```
>>> Pilih: 3

===== TURN 3 (BULBU - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 27. Tampilan pilih cancel di penggunaan potion

```
>>> Pilih: 2

Tidak ada potion yang dapat digunakan.
Tidak dapat memilih potion.

===== TURN 5 (BULBU - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 28. Tampilan tidak dapat memilih potion

```
>>> Pilih: 3

Dengan memilih skip battle, monster dengan otomatis dalam keadaan menyerang

>>> Yakin ingin skip battle? (y/n) y

Selamat, Anda berhasil mengalahkan monster Bulbu!!!

===== STATS =====
Hadiah           : 36 OC
Damage diberikan : 1217
Damage diterima  : 950
```

Gambar 29. Tampilan skip battle dan kemenangan

```
>>> Pilih: 4

Anda melarikan diri dari pertarungan

===== STATS =====
Hadiah           : 0 OC
Damage diberikan : 0
Damage diterima  : 0
```

Gambar 30. Tampilan quit battle



F09-Arena

```
Masukkan command: arena

Loading...

Selamat datang di Arena!

Pilih Monster-mu untuk bertarung!

===== MONSTER LIST =====
1. Pikachow
2. Bulbu
3. Zuko

>>> Pilih: 
```

Gambar 31. Tampilan awal arena

```
>>> Pilih: 1

RAAWRRRR!!! Radhi mengeluarkan monster Pikachow!!!

Name      : Pikachow
ATK Power : 150
Def Power : 12
HP        : 720
Level     : 3

===== STAGE 1 =====
```

Gambar 32. Tampilan pilih monster dan stage awal

```
RAAWRRRR!!! Monster Bulbu telah muncul!!!

Name      : Bulbu
ATK Power : 50
Def Power : 50
HP        : 1200
Level     : 1

===== TURN 1 (PIKACHOW - RADHI) =====
1. Attack
2. Use Potion
3. Skip Battle
4. Quit

>>> Pilih: 
```

Gambar 33. Tampilan muncul lawan dan battle arena

```
>>> Pilih: 3

Dengan memilih skip battle, monster dengan otomatis dalam keadaan menyerang

>>> Yakin ingin skip battle? (y/n) y

Selamat, Anda berhasil mengalahkan monster Bulbu!!!

STAGE CLEARED! Anda mendapatkan 30 OC pada stage ini

Memulai stage berikutnya . . .

===== STAGE 2 =====
```

Gambar 34. Tampilan skip battle dan stage lanjutan

```
>>> Yakin ingin skip battle? (y/n) y

Yahhh, Anda dikalahkan monster Zuko. Jangan menyerah, coba lain kali!!!

GAME OVER! Sesi latihan berakhir pada stage 2

===== STATS =====
Total hadiah      : 30
Jumlah stage      : 1
Damage diberikan  : 2106
Damage diterima   : 1441
```

Gambar 35. Tampilan kalah battle dan akhir arena

Masukkan command: shop

Loading...

=====

# SHOP

=====

Selamat datang di Shop!

>>> Pilih aksi (lihat/beli/keluar) █

Gambar 36. Tampilan awal shop

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

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

ID	Type	ATK Power	DEF Power	HP	Stok	Harga
1	Pikachow	125	10	600	10	500
2	Bulbu	50	50	1200	4	700
3	Zeze	300	10	100	3	1000
4	Zuko	100	25	800	10	10
5	Chacha	80	30	700	10	20

>>> Pilih aksi (lihat/beli/keluar) █

Gambar 37. Tampilan lihat monster

>>> Pilih: 1

Anda telah memiliki Pikachow

Koin Anda saat ini: 90889 OC

Ketik id monster untuk membeli

Ketik 0 untuk kembali

>>> Pilih: █

Gambar 38. Tampilan membeli monster yang sudah dimiliki

```
>>> Pilih: 3

Selamat, monster Zeze telah berhasil dibeli!

Koin tersisa: 89889 OC

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

Gambar 39. Tampilan berhasil membeli monster

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

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

ID |Type          |Stock      |Harga      |
---|-----|-----|-----|
1  |strength      |10         |50         |
2  |resilience    |5          |30         |
3  |healing        |3          |20         |

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

Gambar 40. Tampilan lihat potion

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

[Peringatan] Pilihan tidak sesuai!

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

Gambar 41. Tampilan input tidak valid di shop

```
>>> Pilih: 1

Masukkan Jumlah Item yang ingin dibeli : 11

Pembelian Gagal :( , Stock tidak mencukupi

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

Gambar 42. Tampilan out of stock

```
>>> Pilih: 1  
  
Masukkan Jumlah Item yang ingin dibeli : 5  
  
Selamat!!! Anda Telah membeli potion strength sebanyak 5  
  
Koin tersisa: 89639 OC  
  
>>> Pilih aksi (lihat/beli/keluar) █
```

Gambar 43. Tampilan berhasil membeli potion

```
>>> Pilih aksi (lihat/beli/keluar) beli  
  
>>> Mau beli apa? (monster/potion) monster  
  
Koin Anda saat ini: 90 OC  
  
Ketik id monster untuk membeli  
Ketik 0 untuk kembali  
  
>>> Pilih: 3  
  
Pembelian Gagal :( , Koin tidak mencukupi  
  
>>> Pilih aksi (lihat/beli/keluar) █
```

[illegible]

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

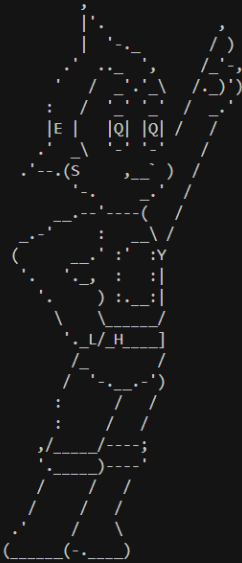
Thankyou...

1

Sampai jumpa lagi di Shop!

Gambar 45. Tampilan keluar shop

Selamat datang di Lab Agent!!



Perkenalkan, namaku Mark! Disini aku akan membantu kamu untuk meng-upgrade monster yang kamu miliki! Pertama-tama, aku akan menampilkan semua monster yang kamu miliki!

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

1. Pikachu (Level: 3)
2. Bulbu (Level: 4)
3. Zuko (Level: 5)

Koinmu saat ini: 90

Silahkan pilih monster yang ingin di Upgrade!

Pilih monster:

>>>

Gambar 46. Tampilan masuk lab

Silahkan pilih monster yang ingin di Upgrade!

Pilih monster:

>>> 1

Pikachow akan di-upgrade ke level 4

Harga untuk melakukan upgrade adalah 600 OC.

>>> Lanjutkan upgrade (Y/N): y

Maaf, OC Anda tidak mencukupi untuk melakukan upgrade.

Apakah kamu ingin melanjutkan upgrade?

Ketik nomor dari pilihan berikut

1. Lanjut upgrade
2. Keluar

>>>

Gambar 47. Tampilan out of coin

```
Apakah kamu ingin melanjutkan upgrade?  
Ketik nomor dari pilihan berikut  
1. Lanjut upgrade  
2. Keluar  
  
>>> 3  
Perintah tidak dikenal!  
1. Lanjut upgrade  
2. Keluar  
  
>>> 
```

Gambar 48. Tampilan input tidak valid di lab

```
Pilih monster:  
>>> 1  
  
Pikachow akan di-upgrade ke level 5  
Harga untuk melakukan upgrade adalah 800 OC.  
  
>>> Lanjutkan upgrade (Y/N): y  
  
Selamat, Pikachow berhasil di-upgrade ke level 5!  
  
Apakah kamu ingin melanjutkan upgrade?  
Ketik nomor dari pilihan berikut  
1. Lanjut upgrade  
2. Keluar  
  
>>> 
```



Gambar 49. Tampilan upgrade berhasil

```
Silahkan pilih monster yang ingin di Upgrade!  
Pilih monster:  
>>> 1  
  
Maaf, monster yang Anda pilih sudah memiliki level maksimum.  
  
Apakah kamu ingin melanjutkan upgrade?  
Ketik nomor dari pilihan berikut  
1. Lanjut upgrade  
2. Keluar  
  
>>> 
```

```

>>> 2

          +-----+
          |         |
          | Thankyou... |
          |         |
          +-----+
          /

      .-.-.-.-.-.
      |             |
      |  +_+  |
      |  D_  |
      |  > \_ / <  |
      |  >_ \o_ | o/ |
      |  |  |  | /  |
      | \,_____/ ) o  |
      |  o  '  o  |  | | |
      |  o  |  o  |  |_/|
      |  o  |  o  |  |_/|
      |  o  |  o  |  |))|
      |  |  |  |  |
      |  \  |  |  |
      |__o/ \_____|
      |  |  |  |
      |__>|<__|
      |_____,|_____|
      Koinmu saat ini: 90
      Sampai jumpa lagi di Lab!

```

>>> 2

Thankyou...

1

The diagram illustrates a hierarchical structure, likely representing a search space or a decision tree. At the top is a dashed line with a central node labeled 'D'. Below 'D' are two nodes labeled '+'. These are connected to a series of nodes labeled 'o' and 'o/' in a branching structure. The diagram includes various symbols like '>', '<', and '>|<' at the bottom, suggesting a comparison or ordering of the paths.

Koinmu saat ini: 90

Sampai jumpa lagi di Lab!

Gambar 51. Tampilan keluar dari lab

# SHOP MANAGEMENT

```

=====
Konichiwa :D, Selamat datang kembali Admin!
Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
Mau lihat apa? (monster/potion/back): monster
ID      |Type      |ATK Power|DEF Power|HP      |Stok      |Harga    |
-----|-----|-----|-----|-----|-----|-----|
1       |Pikachow  |125      |10       |600     |10        |30       |
2       |Bulbu     |50       |50       |1200    |4         |700      |
3       |Zeze      |300      |10       |100     |3         |1000     |
3016525 |Gojo      |526      |47       |700     |10        |10       |
7366714 |Luffy     |437      |20       |800     |10        |20       |
1471882 |Sasuke    |333      |18       |767     |100       |100      |

```

```

Pilih aksi (lihat/tambah/ubah/hapus/keluar): lihat
Mau lihat apa? (monster/potion/back): potion
ID      |Type      |Stok      |Harga    |
-----|-----|-----|-----|
1       |strength  |10        |50       |
2       |resilience|5         |30       |
3       |healing   |3         |20       |

```

Gambar 52. Tampilan awal Shop Management

```

Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
Mau tambah apa? (monster/potion/back): monster
ID      |Type      |ATK Power|DEF Power|HP      |
-----|-----|-----|-----|-----|
8458600 |Zuko      |100      |25       |800     |
3337382 |Chacha    |80       |30       |700     |
3849617 |JOKER     |1000     |49       |800     |
Jika tidak jadi menambahkan, silahkan ketik 'cancel'
Masukkan id monster: 3849617
Masukkan stok: 10
Masukkan harga: 10
Monster berhasil ditambah!
Pilih aksi (lihat/tambah/ubah/hapus/keluar): tambah
Mau tambah apa? (monster/potion/back): potion
ID      |Type      |
-----|-----|
Jika tidak jadi menambahkan, silahkan ketik 'cancel'
Masukkan id potion: cancel

```

Gambar 53. Tampilan Jika menambah monster

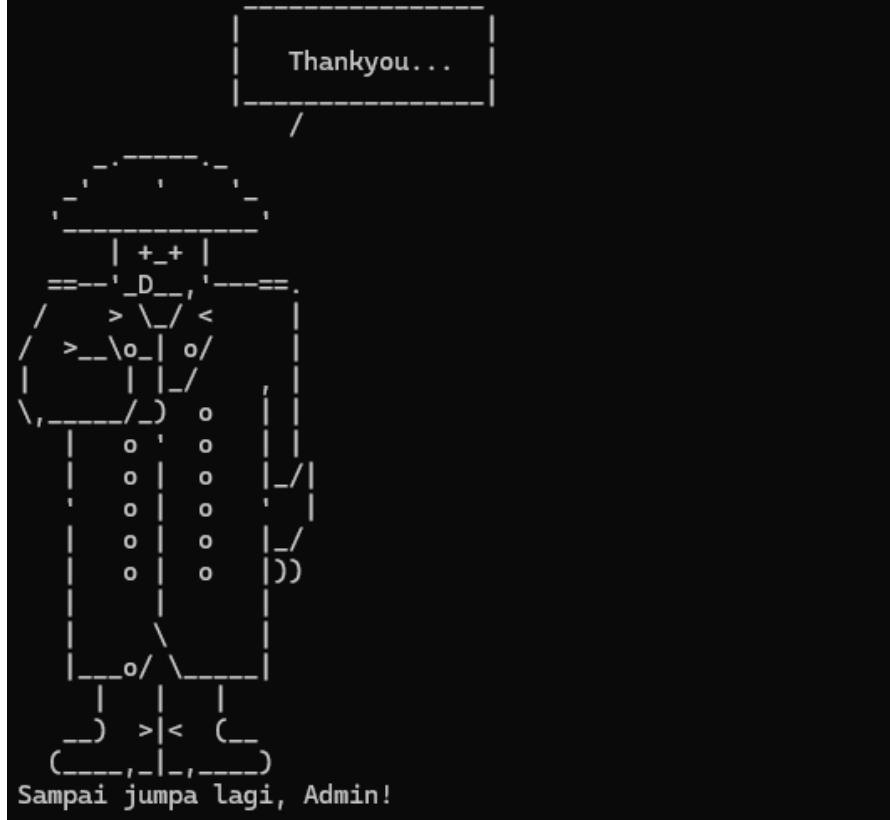
```

Pilih aksi (lihat/tambah/ubah/hapus/keluar): ubah
Mau ubah apa? (monster/potion/back): potion
ID      |Type      |Stok      |Harga    |
-----|-----|-----|-----|
1       |strength  |10        |50       |
2       |resilience|5         |30       |
3       |healing   |3         |20       |
Jika tidak jadi menambahkan, silahkan ketik -1
Masukkan id potion: 2
Masukkan stok baru: 10
Masukkan harga baru: 10
Item telah berhasil diubah!

```

Gambar 54. Tampilan jika ingin mengubah potion

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



Gambar 55. Tampilan ketika selesai mengedit

F13-Monster  
Management



```

===== SILAHKAN PILIH FITUR =====
1. Tampilkan Monster yang anda miliki
2. Tambahkan Monster Baru
3. Keluar

Pilih fitur diatas (1/2/3) :2

Masukkan Nama Monster barumu : dasprohebat
Masukkan ATK Power : 100
Masukkan DEF Power (0-50) : 60
INPUT SALAH!! DEF Power harus bernilai 0-50, coba lagi!
Masukkan DEF Power Baru (0-50): 50
Masukkan HP Monster : 100
BERHASIL, Monster Baru Telah Dibuat!!
Type      : dasprohebat
ATK Power : 100
DEF Power : 50
HP        : 100
Apakah Monster baru akan dimasukkan ke database? (Y/N)
>>> y
Monster berhasil dimasukkan ke database!

```

Gambar 56. Tampilan membuat monster baru

```

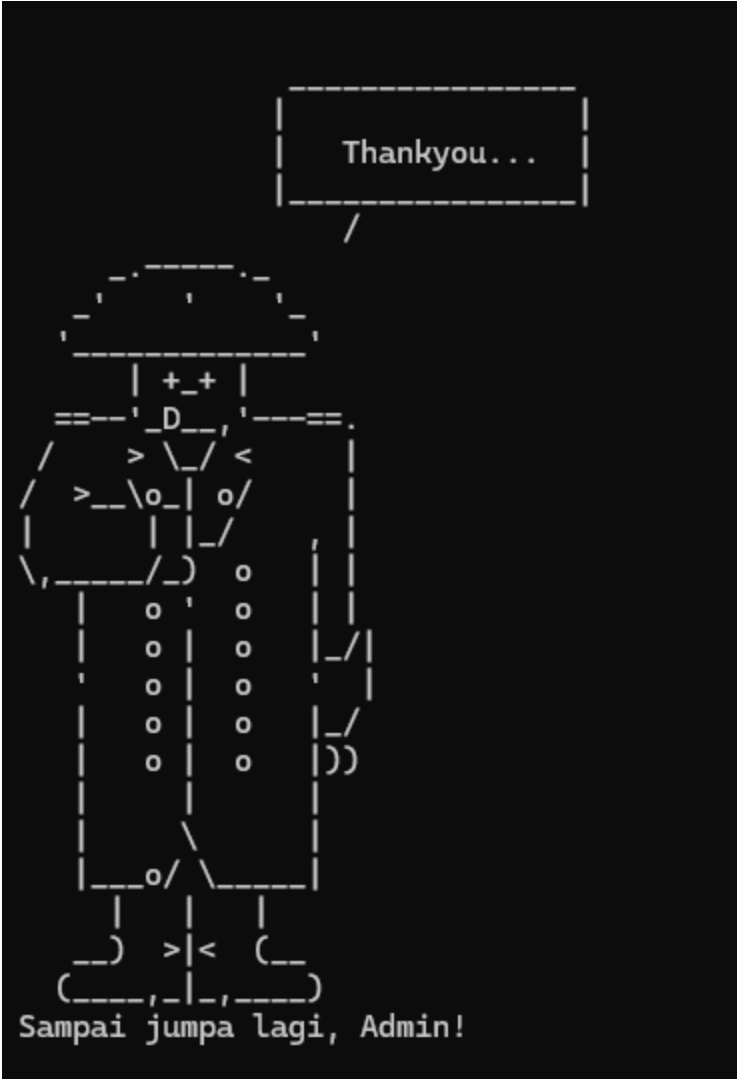
===== SILAHKAN PILIH FITUR =====
1. Tampilkan Monster yang anda miliki
2. Tambahkan Monster Baru
3. Keluar

Pilih fitur diatas (1/2/3) :1

===== Menampilkan semua monster! =====
1      |Pikachow   |125      |10      |600      |
2      |Bulbu      |50        |50       |1200     |
3      |Zeze       |300       |10       |100      |
8458600|Zuko       |100       |25       |800      |
3337382|Chacha     |80        |30       |700      |
3016525|Gojo       |526       |47       |700      |
7366714|Luffy      |437       |20       |800      |
1471882|Sasuke     |333       |18       |767      |
3849617|JOKER      |1000      |49       |800      |
713900 |dasprohebat|100       |50       |100      |

```

Gambar 57. Tampilan melihat kembali monster yang telah ditambah

	 <p>Gambar 58. Tampilan ketika selesai mengedit</p>
F14-Load	<pre>PS C:\daspro\TUBES&gt; python Load.py Tidak ada nama folder yang diberikan! Harap masukkan nama folder!</pre> <p>Gambar 59. Tampilan tidak menerima inputan folder</p> <pre>PS C:\daspro\TUBES&gt; python Load.py Testing Yah, folder "Testing" tidak dapat ditemukan.</pre> <p>Gambar 60. Tampilan inputan folder yang tidak ada</p> <pre>PS C:\daspro\TUBES&gt; python Load.py testing Loading...</pre> <p>Gambar 61. Tampilan inputan folder yang valid (sudah ada)</p>

F15-Save	<pre> &gt;&gt;&gt; Save Masukkan nama folder untuk penyimpanan: testing 1  Folder savedata\testing 1 sudah ada.  Saving.... Berhasil menyimpan data di folder savedata\testing 1 </pre> <p>Gambar 62. Tampilan inputan folder baru</p> <pre> &gt;&gt;&gt; Save Masukkan nama folder untuk penyimpanan: testing 1  Membuat folder savedata\testing 1...  Saving.... Berhasil menyimpan data di folder savedata\testing 1 </pre> <p>Gambar 63. Tampilan inputan nama folder yang sudah ada</p>
F16-Exit	<pre> Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): y </pre> <p>Gambar 64. Tampilan user menginput y (ingin keluar)</p> <pre> Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): n Kembali ke program. </pre> <p>Gambar 65. Tampilan user menginput n (tidak jadi keluar)</p> <pre> Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): a Pilihan tidak valid! Apakah Anda yakin ingin keluar dari program? Semua file yang tidak di save akan terhapus! (y/n): y </pre> <p>Gambar 66. Tampilan inputan user tidak valid</p>

B04-JACKPOT!

```
JACKPOT

===== Daftar Item =====
1. Topi: 50 OC
2. Pedang: 100 OC
3. Koin: 200 OC
4. Potion: 300 OC
5. Monster: 500 OC

Mulai bermain? (y/n): y

Koin kamu saat ini adalah 9999 OC
$$$$$$$$$      Topi | Monster | Pedang      $$$$$$$$$$
650 OC telah ditambahkan ke akun Anda!
Lanjut bermain (y/n): y

Koin kamu saat ini adalah 10249 OC
$$$$$$$$$      Pedang | Koin | Potion      $$$$$$$$$$
600 OC telah ditambahkan ke akun Anda!
Lanjut bermain (y/n): y

Koin kamu saat ini adalah 10449 OC
$$$$$$$$$      Topi | Topi | Topi      $$$$$$$$$$
JACKPOTTT!!! Monster Luffy telah dimasukkan dalam inventory anda!
Lanjut bermain (y/n): n
Kamu mendapat koin sebesar 1250 OC
Koin kamu saat ini adalah 10049 OC!
Selamat tinggal!
```

Gambar 67. Tampilan awal Permainan Jackpot



```

* * * * *
* P   X   X   *
*   X   S   X   E *
*           X   *
*   X       X   *
*   X       *
*   X       X   *
*           X   *
* L   X X X X   A *
*           X   *
*       J   X   *
* * * * *
P saat ini berada di posisi (0,0)
Masukkan command: d

```

Agent akan pindah ke kanan!

```

* * * * *
* P   X   X   *
*   X   S   X   E *
*           X   *
*   X       X   *
*   X       *
*   X       X   *
*           X   *
* L   X X X X   A *
*           X   *
*       J   X   *
* * * * *
P saat ini berada di posisi (0,1)
Masukkan command: d

```

Gambar 68. Tampilan ketika User menggerakan Player

Agent akan pindah ke kanan!

```
* * * * *
*      X      X  *
*      X  S  X  E  *
*              X  *
*      X      X  *
*      X      *
*      X      X  *
*              X  *
*  L  X X X X  A  *
*          P  X  *
*          J  X  *
* * * * *
```

P saat ini berada di posisi (8,4)

Masukkan command: jackpot

Loading...

Gambar 69. Tampilan ketika mendekati jackpot dan akan bermain

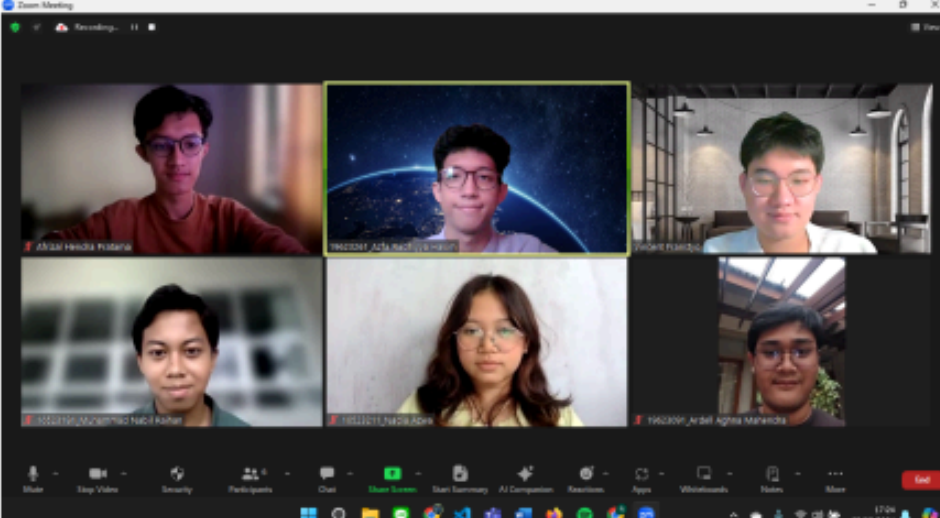
## Lampiran

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

Nomor Asistensi : 1  
 No. Kelompok/Kelas : A/01  
 Tanggal asistensi : 3 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	19623091 / Ardell Aghna Mahendra
2	19623261 / Azfa Radhiyya Hakim
3	16523041 / Afrizal Hendra Pratama
4	16523191 / Muhammad Nabil Raihan
5	16523211 / Nadia Azwa Wibowo
6	
Asisten pembimbing	NIM / Nama
	18221100 / Vincent Franstyo

#### Catatan Asistensi:

<b>Rangkuman Diskusi</b> <ol style="list-style-type: none"> <li>1. Izin mengkonfirmasi, apakah admin akan ikut bermain, atau hanya mengedit-edit shop dan inventory?</li> <li>2. Jika mengerjakan B05, apakah F04 tetap dikerjakan?</li> <li>3. Lebih baik untuk feat langsung nama fungsi</li> <li>4. Laporan jangan lupa dicicil</li> </ol>
<b>Tindak Lanjut</b> <ol style="list-style-type: none"> <li>1. Benar, admin tidak ikut bermain</li> <li>2. F04 tetap dikerjakan</li> </ol>
<b>Dokumentasi</b> 

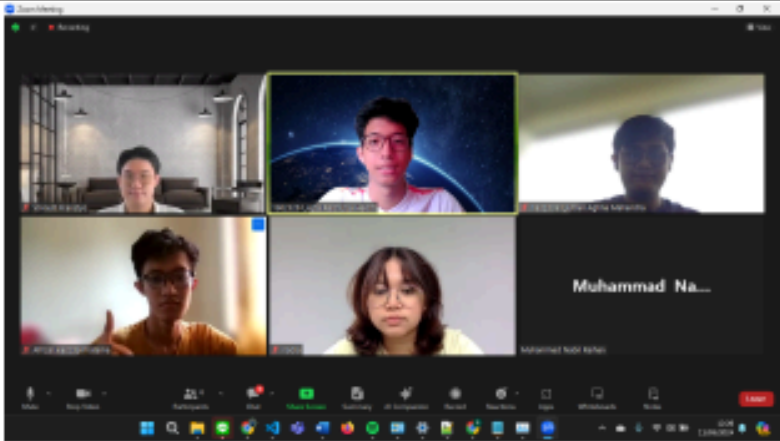
Gambar 70. Form Asistensi 1

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

Nomor Asistensi : 2  
No. Kelompok/Kelas : A/01  
Tanggal asistensi : 11 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)	
	1	19623091 / Ardell Aghna Mahendra
	2	19623261 / Azfa Radhiyya Hakim
	3	16523041 / Afrizal Hendra Pratama
	4	16523191 / Muhammad Nabil Raihan
	5	16523211 / Nadia Azwa Wibowo
Asisten pembimbing	NIM / Nama	
	18221100 / Vincent Franstyo	

**Catatan Asistensi:**

<p><b>Rangkuman Diskusi</b></p> <ol style="list-style-type: none"> <li>1. Apakah notasi algoritmik boleh berbeda dengan program pythonnya?</li> <li>2. Apakah boleh memakai flowchart?</li> <li>3. Commit messages harus diperjelas</li> <li>4. Di dalam fungsi jangan lupa dikasih deskripsi dan kamus seperti di kelas</li> <li>5. Pembuatan release termasuk dalam pengumpulan, bisa dicoba dari sekarang</li> <li>6. Penambahan monster di dalam csv terjadi kalau sudah save</li> <li>7. Apakah boleh menambahkan fitur skip?</li> </ol>
<p><b>Tindak Lanjut</b></p> <ol style="list-style-type: none"> <li>1. Boleh, yang penting logikanya masih sama</li> <li>2. Pakai notasi algoritmik saja</li> <li>3. Boleh, asal tidak merubah program</li> </ol>
<p><b>Dokumentasi</b></p> 

Gambar 71. Form Asistensi 2

Name	Last commit message	Last commit date
gitkeep	Initial commit	3 weeks ago
B04.py	add design tulisan	2 days ago
B05.py	feat: adding color	18 hours ago
F02.py	fix bug	last week
F01.py	feat: adding color	18 hours ago
F02.py	feat: adding color	18 hours ago
F03.py	fix: memperbaiki keseluruhan fungsi	yesterday
F04.py	adding input design and text color	18 hours ago
F05.py	fix: oxcadex bug while choosing monster id	2 days ago
F06.py	fix: rounding bugs	4 days ago
F07.py	resolve conflict	2 days ago
F08.py	feat: adding color	18 hours ago
F09.py	feat: added Arena - F09	last week
F10.py	fix: shop checking id inventory	2 days ago
F11.py	add: sisa koin setelah selesai meng-upgrade	2 days ago
F12.py	fix bug tipe data dan menambahkan desain	2 days ago
F13.py	feat: adding random number generator untuk id monster	2 days ago
F14.py	fix bug	18 hours ago
F15.py	fix variable	2 days ago
F16.py	fix: memperbaiki fungsi exit	2 days ago
Operations.py	fix: function inputValid	2 days ago
Testing_5788.py	testing purpose only for F05 - F09	last week
design_monster.py	resolve conflict	2 days ago
do_csv.py	feat: F01,F02,F03,F04,F12,B05,do_csv/main	2 weeks ago

Gambar 72. Penggunaan Git Best Practice (B01)