

LAPORAN TUGAS BESAR
IF1210 DASAR PEMROGRAMAN
SEMESTER II 2023/2024
K10-G



Anggota Kelompok:
Daniel Pedrosa Wu (19623099)
Devon Wiraditya Tanumihardja (19623039)
Darryl Rayhananta Adenan (19623049)
Muhammad Aymar Barkhaya (19623189)
Fathur Rahman Arahkhal (16523199)

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,

Daniel Pedrosa Wu (19623099)
Devon Wiraditya Tanumihardja (19623039)
Darryl Rayhananta Adenan (19623049)
Muhammad Aymar Barkhaya (19623189)
Fathur Rahman Arahkhal (16523199)

Daftar Isi

Daftar Isi.....	2
Daftar Gambar.....	4
Daftar Tabel.....	5
Deskripsi Persoalan.....	6
Daftar Pembagian Kerja Kelompok.....	7
Pembagian Laporan.....	8
Checklist Hasil Pengerjaan Tugas Besar.....	9
Desain Command.....	10
F00 - RNG.....	10
F01 - Register.....	10
F02 - Login.....	11
F03 - Logout.....	12
F04 - Menu & Help.....	12
F05 - Monster.....	12
F06 - Potion.....	13
F07 - Inventory.....	13
F08 - Battle.....	14
F09 - Arena.....	17
F10 - Shop & Currency.....	21
F11 - Laboratory.....	22
F12 - Shop Management.....	22
F13 - Monster Management.....	24
F14 - Load.....	25
F15 - Save.....	25
F16 - Exit.....	26
Desain Kamus Data.....	26
Program Utama.....	26
F00 - RNG.....	26
F01 - Register.....	27
F02 - Login.....	27
F03 - Logout.....	27
F04 - Menu & Help.....	27
F05 - Monster.....	27
F06 - Potion.....	28
F07 - Inventory.....	28
F08 - Battle.....	28
F09 - Arena.....	28
F10 - Shop & Currency.....	29

F11 - Laboratory.....	29
F12 - Shop Management.....	29
F13 - Monster Management.....	29
F14 - Load.....	30
F15 - Save.....	30
F16 - Exit.....	30
B03 - Monster Ball.....	31
Desain Dekomposisi Algoritmik.....	31
F00 - RNG.....	31
F01 - Register.....	31
F02 - Login.....	31
F03 - Logout.....	31
F04 - Menu & help.....	31
F05 - Monster.....	32
F06 - Potion.....	32
F07 - Inventory.....	32
F08 - Battle.....	32
F09 - Arena.....	32
F10 - Shop & currency.....	33
F11 - Laboratory.....	33
F12 - Shop management.....	33
F13 - Monster management.....	33
F14 - Load.....	33
F15 - Save.....	33
F16 - Exit.....	34
B03 - Monster Ball.....	34
Spesifikasi.....	34
F00 - RNG.....	34
F01 - Register.....	35
F02 - Login.....	37
F03 - Logout.....	39
F04 - Menu & Help.....	39
F05 - Monster.....	41
F06 - Potion.....	41
F07 - Inventory.....	41
F08 - Battle.....	42
F09 - Arena.....	55
F10 - Shop & Currency.....	57
F11 - Laboratory.....	57
F12 - Shop Management.....	60
F13 - Monster Management.....	62

F14 - Load.....	62
F15 - Save.....	63
F16 - Exit.....	64
B03 - Monster Ball.....	65
Lampiran Hasil Pengujian Program.....	67
Lampiran Form Asistensi.....	80
Daftar Referensi.....	84

Daftar Gambar

Gambar 1.1.....	38
Gambar 1.2.....	38
Gambar 1.3.....	38
Gambar 1.4.....	38
Gambar 1.5.....	39
Gambar 1.6.....	40
Gambar 1.7.....	40
Gambar 1.8.....	41
Gambar 1.9.....	41
Gambar 1.10.....	41
Gambar 1.11.....	42
Gambar 1.12.....	42
Gambar 1.13.....	42
Gambar 1.14.....	42
Gambar 1.15.....	43
Gambar 1.16.....	43
Gambar 1.17.....	44
Gambar 1.18.....	45
Gambar 1.19.....	45
Gambar 1.20.....	45
Gambar 1.21.....	47
Gambar 1.22.....	47
Gambar 1.23.....	48
Gambar 1.24.....	48
Gambar 1.25.....	48
Gambar 1.26.....	49
Gambar 1.27.....	49
Gambar 2.1 : Form Asistensi 1.....	50
Gambar 2.2 : Form Asistensi 2.....	52

Daftar Tabel

Tabel 1. Pembagian Kerja Kelompok.....	8
Tabel 2. Pembagian Laporan.....	9
Tabel 3. Checklist hasil pengerjaan.....	10
Tabel 4. Lampiran Hasil Pengujian Program.....	50

Deskripsi Persoalan

Tugas besar ini meminta untuk membuatkan sebuah program game dari sebuah cerita di pinggiran kota Danville, di mana Purry si platypus sedang meghadapi tantangan besar. Dr. Asep Spakbor, seorang ilmuwan jahat, telah menciptakan monster - monster mengerikan yang mengancam keamanan kota Danville. Purry tidak dapat mengalahkan monster - monster itu sendirian, untuk itu kami diminta untuk membantu Purry mengalahkan Dr. Asep Spakbor.

Untuk menjalankan program tersebut, ada beberapa fungsi yang harus ada yaitu

1. Fungsi Random Number Generator untuk menggenerasi angka secara acak dari batas bawah hingga batas atas.
2. Fungsi user melakukan Register akun dengan membuat username dan password, lalu user memilih monster awal, mendapatkan 0 coin, dan role agent.
3. Fungsi user melakukan Login akun yang sudah diregister untuk masuk ke dalam sistem dan menggunakan perintah dalam sistem.
4. Fungsi user melakukan Logout akun untuk keluar dari akun setelah selesai menggunakan sistem.
5. Fungsi Menu & Help digunakan untuk menampilkan semua command yang dapat digunakan sesuai dengan akses yang dimiliki pemain.
6. Fungsi Monster akan disimpan dalam suatu database monster. Monster player akan memiliki data tipe monster, atk power, defense power, dan HP.
7. Fungsi Potion akan memberi buff kepada monster yang dimiliki player dengan durasi 1 battle dan hanya bisa digunakan sekali.
8. Fungsi Inventory akan menunjukkan item dan monster yang dimiliki player.
9. Fungsi Battle, player dan monster akan bergiliran melakukan aksi. Terdapat 3 opsi yakni attack, quit, dan use potion.
10. Fungsi Arena adalah tempat permainan terjadi, player akan melewati stage untuk mendapatkan reward.
11. Fungsi Shop & Currency digunakan untuk membeli item atau monster dan menunjukkan currency.
12. Fungsi Laboratory digunakan untuk meng-upgrade level dari monster yang dimiliki player.
13. Fungsi Shop Management adalah fungsi khusus admin. Digunakan untuk mengubah item - item yang ada di shop.
14. Fungsi Monster Management untuk menambahkan monster ke database.
15. Fungsi Load dimana prosedur ini digunakan untuk memuat data yang sesuai dengan struktur data eksternal. Prosedur ini akan dijalankan sekali saat pertama kali program dijalankan.
16. Fungsi Save dimana command ini digunakan untuk menjalankan prosedur penyimpanan data yang berada di program sesuai dengan struktur data eksternal.
17. Fungsi Exit dimana prosedur ini digunakan untuk keluar dari permainan, dan sebelum keluar dari program, pemain akan diberikan opsi untuk melakukan prosedur save atau tidak.

Untuk mengoperasikan fungsi-fungsi tersebut, diperlukan akses ke file-file csv yang berisi data yang disimpan oleh program. Selain itu, Bonus 01 (Git Best Practice), B02 (Typing), B03 (Monster Ball), B04 (Jackpot) juga dapat dikerjakan.

Daftar Pembagian Kerja Kelompok

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
CSV Parser	*	19623099	19623099	19623099
F00 - RNG	*	19623099	19623099	19623099
F01 - Register	*	19623189	19623189	19623189
F02 - Login	*	16523199	16523199	16523199
F03 - Logout	*	16523199	16523199	16523199
F04 - Menu & Help	*	19623189	19623189	19623189
F05 - Monster	*	19623049	19623049	19623049
F06 - Potion	*	19623039	19623039	19623039
F07 - Inventory	*	19623049	19623049	19623049
F08 - Battle	*	19623099	19623099	19623099
F09 - Arena	*	19623099	19623099	19623099
F10 - Shop & Currency	*	19623099	19623099	19623099
F11 - Laboratory	*	19623189	19623189	19623189
F12 - Shop Management	*	19623099	19623099	19623099
F13 - Monster Management	*	19623099	19623099	19623099

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
CSV Parser	*	19623099	19623099	19623099
F00 - RNG	*	19623099	19623099	19623099
F01 - Register	*	19623189	19623189	19623189
F02 - Login	*	16523199	16523199	16523199
F03 - Logout	*	16523199	16523199	16523199
F04 - Menu & Help	*	19623189	19623189	19623189
F14 - Load	*	16523199	16523199	16523199
F15 - Save	*	16523199	16523199	16523199
F16 - Exit	*	19623039	19623039	19623039

Tabel 1. Pembagian Kerja Kelompok

Pembagian Laporan

No	Bagian Laporan	NIM Pembuat
1	Cover	19623049
2	Pernyataan Kelompok	19623049
3	Daftar isi, Daftar tabel, Daftar gambar, & Daftar referensi	19623049
4	Deskripsi Persoalan	19623049
5	Desain Command	19623039 19623049
6	Desain Kamus Data	19623189
7	Desain Dekomposisi Algoritmik dan Fungsional Program	19623049
8	Spesifikasi Fungsional Data	19623039 19623049 19623099 19623189

9	Pengujian Program	19623049
10	Lampiran	19623049 19623099 19623189 16523199

Tabel 2. Pembagian Laporan

Checklist Hasil Pengerjaan Tugas Besar

Fitur	Desain	Implementasi	Testing
CSV Parser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F00 - RNG	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F01 - Register	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F02 - Login	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F03 - Logout	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F04 - Menu & Help	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F05 - Monster	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F06 - Potion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F07 - Inventory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F08 - Battle	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F09 - Arena	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F10 - Shop & Currency	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F11 - Laboratory	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F12 - Shop Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

F13 - Monster Management	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F14 - Load	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F15 - Save	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
F16 - Exit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CSV Parser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tabel 3. Checklist hasil pengerjaan

Desain Command

F00 - RNG

I.S. fungsi ini tidak dapat kita panggil.
F.S. fungsi ini berfungsi di dalam fungsi yang lain.

>>> Random Number Generator
Pada fungsi ini, kita menggunakannya pada beberapa fitur seperti register untuk mendapatkan ID agent, di Battle untuk menampilkan monster musuh yang akan bertarung, dan lain-lain.

F01 - Register

I.S. User melakukan registrasi akun
F.S. akun tersimpan ke dalam csv

```

>>> Register
Username :
Password :

# ketika username atau password kosong, maka akan muncul pesan
Username atau password tidak boleh kosong!

# ketika password salah, maka akan muncul pesan
Password salah!

# ketika username telah dipakai, maka akan muncul pesan
Username {username} telah terpakai, silahkan gunakan username lain!

# ketika penulisan username tidak valid, maka akan muncul pesan
Username hanya boleh berisi alfabet, angka, underscore, dan strip!

# ketika username dan password yang dimasukkan valid, maka akan muncul
pesan
Anda telah mendaftarkan user {username}
1 Pikachu ATK Power: 55 DEF Power: 40 HP: 35 Speed: 90
2 Bulbasaur ATK Power: 49 DEF Power: 49 HP: 45 Speed: 45
3 Charmander ATK Power: 52 DEF Power: 43 HP: 39 Speed: 65
Masukkan ID:

# ketika id yang dimasukkan tidak valid
Masukkan ID yang valid

```

F02 - Login

I.S. User memasukkan username yang terdaftar di user.csv, user tidak bisa melakukan login lagi setelah melakukan login, user harus logout terlebih dahulu

F.S. User login

```

>>> login
Username :
Password :

# ketika username salah, maka akan muncul pesan
Username tidak terdaftar!

# ketika password salah, maka akan muncul pesan
Password salah!

# ketika username dan password yang dimasukkan benar, maka akan muncul
pesan

```

Anda telah login dengan username {current_user[1]}, silahkan lakukan 'LOGOUT' sebelum melakukan register.")

F03 - Logout

I.S. User sudah melakukan login
F.S. User logout

```
>>> logout
# Kasus ketika belum login, namun melakukan logout
Logout gagal!
Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout
# Kasus ketika sudah login
Terima kasih sudah bermain, sampai jumpa di lain hari!
```

F04 - Menu & Help

I.S. User membutuhkan bantuan apa yang sebaiknya dilakukan
F.S. User mendapatkan bantuan

```
>>> menu & help
# Kasus sebelum login
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

# Kasus setelah login sebagai agent
Halo Agent Purry. Kamu memanggil command help. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:
  1. Logout      : Keluar dari akun yang sedang digunakan
  2. Inventory: Melihat owca-dex yang dimiliki oleh agent.
  3. ...
```

F05 - Monster

I.S. Database monster kosong
F.S. Monster memiliki atribut dan level

```
# Monster - monster yang ada disatukan pada suatu database bernama "owca-dex".
Type
```

ATK Power
DEF Power
HP
Selain atribut tersebut, setiap monster memiliki data level sendiri
Level 1-5

F06 - Potion

I.S. Atribut monster sama seperti database
F.S. Monster mendapatkan buff ketika battle

Terdapat beberapa jenis potion
1. Strength potion
2. Resilience potion
3. Healing potion
4. Speed potion

Masing - masing potion hanya dapat digunakan sekali dalam 1 battle

F07 - Inventory

I.S. User ingin melihat item yang dimiliki
F.S. User dapat melihat item yang dimiliki

Menampilkan jumlah O.W.C.A. coin user, potion, monster ball, dan monster yang dimiliki user

Kasus memilih salah satu monster user
Ketikkan id untuk menampilkan detail item:
...
Name : ...
ATK Power : ...
DEF Power : ...
HP : ...
Level : ...

Kasus memilih salah satu potion user
Ketikkan id untuk menampilkan detail item:
...
Potion
Type : ...
Quantity : ...

Kasus memilih salah satu monster ball

```
...
Monster Ball
Quantity : ...
```

```
I.S. User mulai bertarung
F.S. User selesai bertarung
```

Pilih monstermu

1. _____
2. _____
3. ...

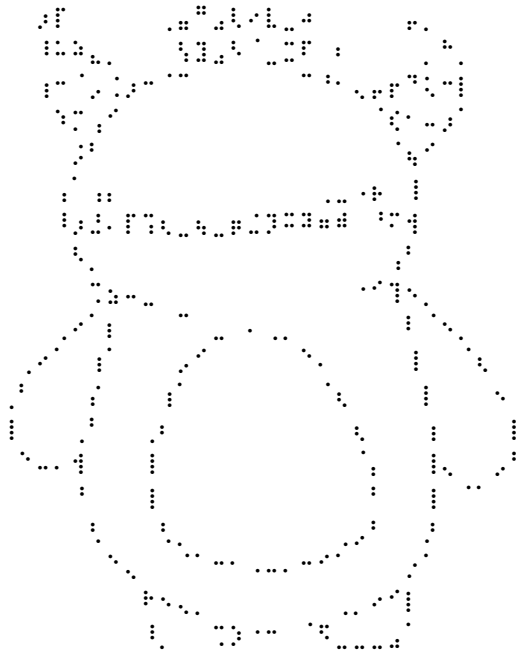
#Kasus ketika input yang dimasukkan valid

Apakah kamu yakin? (Y/N):

#Jika N

Pilih monster

#Jika Y



____, Aku memilih kamu

Player

Name : ____

Type : ____

Level : ____

Attack Power : ____

Defense Power : ____

Speed : ____

HP : ____

Select Action:

1. Attack
2. Switch
3. Use Potion
4. Run
5. Monster Ball

#Jika memasukkan nomor 1

Attack: ____, Reduced by: ____

____ attack ____ for ____ damage

#Serangan bisa tidak kena jika nilai RNG lebih kecil dari perbedaan speed

attacker dengan defender
___ attacks ___ but missed.

#Jika kalah
___ fainted
Anda dikalahkan monster ___

#Jika menang
Selamat anda telah berhasil mengalahkan monster ___
Total OC = ___

#Jika memasukkan nomor 2 dan ada monster lain
Pilih monsternya
1. ___
2. ___
3. ...

#Jika memasukkan nomor 2 dan tidak ada monster lain
Kamu tidak memiliki monster lain

#Jika memasukkan nomor 3 dan memiliki potion
Select potion:
1. Strength (Quantity:)
2. Speed (Quantity:)
3. Resilience (Quantity:)
4. Healing (Quantity:)
5. Cancel

#Jika memasukkan nomor 1 pertama kalinya
___ minum Strength Potion

#Jika memasukkan nomor 1 bukan pertama kalinya
___ sudah minum Strength Potion

#Jika memasukkan nomor 1 dan tidak memiliki strength potion
Anda tidak memiliki Strength Potion

#Jika memasukkan nomor 2 pertama kalinya
___ minum Speed Potion

#Jika memasukkan nomor 2 bukan pertama kalinya
___ sudah minum Speed Potion

#Jika memasukkan nomor 2 dan tidak memiliki Speed Potion
Anda tidak memiliki Speed Potion

#Jika memasukkan nomor 3 pertama kalinya

___ minum Resilience Potion

#Jika memasukkan nomor 3 bukan pertama kalinya

___ sudah minum Resilience Potion

#Jika memasukkan nomor 3 dan tidak memiliki Resilience Potion
Anda tidak memiliki Resilience Potion

#Jika memasukkan nomor 4 pertama kalinya

___ minum Healing Potion

#Jika memasukkan nomor 4 bukan pertama kalinya

___ sudah minum Healing Potion

#Jika memasukkan nomor 4 dan tidak memiliki strength potion
Anda tidak memiliki Healing Potion

#Jika memasukkan nomor 3 dan tidak memiliki potion
Anda tidak memiliki potion

#Jika memasukkan nomor 4 dan berhasil
Anda berhasil kabur dari battle

#Jika memasukkan nomor 4 dan gagal
Anda tidak bisa kabur

#Jika memasukkan nomor 5 dan memiliki monster ball
Catch? (Y/N)

#Jika Y

Kamu melempar Monster Ball

#Jika monster berhasil tertangkap

Monster berhasil ditangkap

Nama monster:

Selamat anda telah berhasil mengalahkan monster ___

Total OC = ___

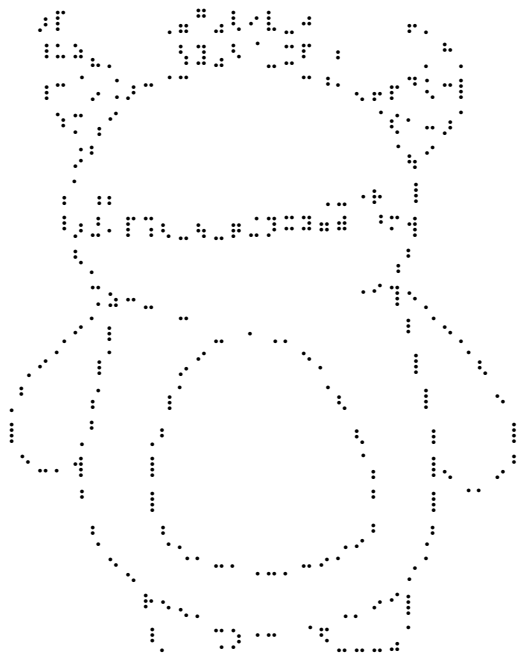
#Jika monster gagal tertangkap

Monster lepas dari Monster Ball

#Jika memasukkan nomor 5 dan tidak memiliki monster ball

Anda tidak memiliki monster ball

[illegible]



____, Aku memilih kamu

Player

Name : ____
Type : ____
Level : ____
Attack Power : ____
Defense Power : ____
Speed : ____
HP : ____

Select Action:

1. Attack
2. Switch
3. Use Potion
4. Run

#Jika memasukkan nomor 1

Attack: ____, Reduced by: ____
____ attack ____ for ____ damage

#Jika kalah pada stage 1

__ fainted

Anda dikalahkan monster ____

Total damage dealt : ____

Total damage taken : ____

Stage cleared: 0

Anda tidak mendapat OWCA Coins

Enemy

Name : ____
Type : ____
Level : ____
Attack Power : ____
Defense Power : ____
Speed : ____
HP : ____

```

#Jika kalah pada stage selain stage 1
Total damage dealt : ____
Total damage taken : ____
Stage cleared: ____
Anda mendapat reward ____ OWCA Coins

#Jika menang
Selamat anda telah berhasil mengalahkan monster ____
#User akan masuk ke stage selanjutnya dan memulai pertarungan baru

#Jika memasukkan nomor 2 dan ada monster lain
Pilih monstermu
1. ____
2. ____
3. ...

#Jika memasukkan nomor 2 dan tidak ada monster lain
Kamu tidak memiliki monster lain

#Jika memasukkan nomor 3 dan memiliki potion
Select potion:
1. Strength (Quantity:)
2. Speed (Quantity:)
3. Resilience (Quantity:)
4. Healing (Quantity:)
5. Cancel

#Jika memasukkan nomor 1 pertama kalinya
____ minum Strength Potion

#Jika memasukkan nomor 1 bukan pertama kalinya
____ sudah minum Strength Potion

#Jika memasukkan nomor 1 dan tidak memiliki strength potion
Anda tidak memiliki Strength Potion

#Jika memasukkan nomor 2 pertama kalinya
____ minum Speed Potion

#Jika memasukkan nomor 2 bukan pertama kalinya
____ sudah minum Speed Potion

#Jika memasukkan nomor 2 dan tidak memiliki Speed Potion
Anda tidak memiliki Speed Potion

#Jika memasukkan nomor 3 pertama kalinya

```

___ minum Resilience Potion

#Jika memasukkan nomor 3 bukan pertama kalinya

___ sudah minum Resilience Potion

#Jika memasukkan nomor 3 dan tidak memiliki Resilience Potion
Anda tidak memiliki Resilience Potion

#Jika memasukkan nomor 4 pertama kalinya

___ minum Healing Potion

#Jika memasukkan nomor 4 bukan pertama kalinya

___ sudah minum Healing Potion

#Jika memasukkan nomor 4 dan tidak memiliki strength potion
Anda tidak memiliki Healing Potion

#Jika memasukkan nomor 3 dan tidak memiliki potion
Anda tidak memiliki potion

#Jika memasukkan nomor 4 dan berhasil
Anda berhasil kabur dari battle

#Jika memasukkan nomor 4 dan gagal
Anda tidak bisa kabur

F10 - Shop & Currency

I.S. agent dapat membeli item yang dipilih sebanyak apapun kecuali monster, jika agent sudah memiliki monster maka agent tidak dapat membeli monster itu lagi.

F.S. OC yang dimiliki agent harus mencukupi dengan apa yang agent beli jika tidak transaksi akan gagal.

>>>shop

Pilih aksi (lihat/beli/keluar)

Jika agent memilih melihat

Agent dapat melihat berbagai item yang ada di shop

Jika agent memilih beli

Jumlah OWCA Coins mu sekarang adalah ___

Mau beli apa? (item/monster) : ____

Jika agent memilih keluar
Agent akan keluar dari fungsi shop

F11 - Laboratory

I.S. User dapat meng-upgrade monster dengan level maksimal level 5.
F.S. Setiap upgrade level memiliki harga yang berbeda yang dibayarkan dengan OC

>>> laboratory

Jika bukan agent
Anda bukan Agent

Jika agent
Menampilkan OWCA Coins dan monster yang dimiliki
Upgrade Price
Level 1 -> Level 2: 300 OC
Level 2 -> Level 3: 500 OC
Level 3 -> Level 4: 700 OC
Level 4 -> Level 5: 900 OC
Pilih monster atau x untuk keluar:

Jika memilih monster yang sudah level 5
Maaf, monster yang Anda pilih sudah memiliki level maksimum

Jika memilih monster untuk diupgrade
Monster akan diupgrade ke level 2
Harga untuk melakukan upgrade monster adalah 300 OC
Lanjutkan upgrade? (y/n)

y tapi tidak cukup OWCA Coins
OC Anda tidak cukup

y
Selamat, monster berhasil diupgrade ke level 2

F12 - Shop Management

I.S. Admin dapat mengatur item apa saja yang diperjualkan dalam Shop & Currency
F.S. Setelah admin melakukan pembaruan, Shop & Currency akan terupdate.

```
>>> shop
# Akses ditolak
Anda bukan Admin

# Akses diterima
Pilih aksi (lihat/tambah/ubah/hapus/keluar):

# lihat
Mau lihat apa? (item/monster): ____
# Jika item
    ID          Type          Stock      Price

    Pilih aksi (lihat/tambah/ubah/hapus/keluar):____
# Jika monster
    ID          Type          ATK Power DEF Power HP          Speed
Stock      Price
    Pilih aksi (lihat/tambah/ubah/hapus/keluar):

# tambah
Mau menambah apa? (item/monster):

    # tambah item tapi shop sudah lengkap
    Semua item sudah ada di shop

    # tambah item
    # menampilkan item yang kosong
    Tambahkan Item ke shop? (y/n):

        #y
        Item telah berhasil ditambahkan ke dalam shop

    # tambah monster tapi shop sudah lengkap
    Semua monster sudah ada di shop

    # tambah monster
    Tambahkan Monster ke shop? (y/n):

        #y
        Monster telah berhasil ditambahkan ke dalam shop

# ubah
Mau ubah apa? (Item/Monster):

    # ubah item/monster tapi shop kosong
    Tidak ada item/monster di shop
```



```

# hapus
Mau hapus apa? (Item/Monster):

# hapus item/monster tapi shop kosong
Tidak ada item/monster yang bisa dihapus

# hapus item/monster
Masukkan ID:

# ID tidak valid
Masukkan input yang valid

# ID valid
Apakah anda yakin? (y/n)?

# y
item/monster telah dihapus dari shop!

```

F13 - Monster Management

I.S. Admin diberi opsi untuk menampilkan semua monster, menambahkan monster baru, dan keluar
 F.S. penambahan monster baru harus menginput data type, nama, ATK power, DEF power, dan HP.

```

>>> monster

# Akses ditolak
Anda bukan Admin

# Akses diterima
1. Tampilkan semua Monster
2. Tambah Monster baru
3. Hapus Monster
4. Keluar

# Option 1
Enak copas dari output programnya tapi aku blm tau cara ngerun main

# Option 2
Menambahkan monster dan statsnya

# Option 3 dengan database kosong

```

Tidak ada monster di database

Option 3

Masukkan ID monster:

Gagal

Tolong masukkan ID yang valid!

Sukses

Hapus Monster dari database? (y/n)

y

Monster telah dihapuskan dari database

Option 4

Kembali ke halaman utama

F14 - Load

I.S. File belum ter-load dan belum mengubah csv menjadi array

F.S. File ter-load dan csv menjadi array

>>> Load

Jika tidak ada nama folder yang diberikan, maka akan muncul pesan

Tidak ada nama folder yang diberikan

Jika folder tidak ditemukan, maka akan muncul pesan

Folder {directory} tidak ditemukan

Jika file berhasil dimuat, maka akan muncul pesan

Selamat datang di Program OWCA

F15 - Save

I.S. Array belum ditulis ke dalam csv

F.S. Permainan akan disave dan array akan ditulis ke dalam bentuk csv

```
>>>save
Masukkan nama folder penyimpanan:
Membuat parent folder save...
Membuat folder ____ ...
Save berhasil dilakukan di folder ____

# jika tidak ditemukan
Folder tidak ditemukan sehingga dibuat folder pada ____
```

F16 - Exit

I.S. Diberikan opsi apakah ingin melakukan penyimpanan file
F.S. Keluar dari program

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

# Jika y
True

# Selain y
False
```

Desain Kamus Data

Program Utama

```
user_id : string

current_user : list

user_data, monster_data, monster_shop_data, monster_inventory_data,
item_shop_data, item_inventory_data : Matrix

type Dictionary : Dictionaries of string and union of integer and string
type DictList : list of Dictionary
type Matrix : list of list of string
type Array : list of string
```

F00 - RNG

KAMUS

a, x0, c, m, x : integer

F01 - Register

KAMUS

username, password, monster_dict, id, x, prompt : string

starter_range, monster_index: integer

starter_ids, new_user_data : list

user : boolean

F02 - Login

KAMUS

username, password, user_index : string

F03 - Logout

KAMUS

username, role_type : string

F04 - Menu & Help

KAMUS

user : boolean

role : string

F05 - Monster

KAMUS

monster, mons_id : string

level, attack, defense, hp : integer

```
monster_list : list
```

F06 - Potion

KAMUS

```
potion : string
```

```
attack, defense, hp, maxhp : integer
```

F07 - Inventory

KAMUS

```
name, char, prompt : string
```

```
int_data, user_monster, monster_inventory, monster_dict, user_items,  
item_inventory, int_data, used_name, used_monsters, options : list
```

```
contains_invalid_char, battle = boolean
```

```
index : integer
```

```
user_items : Matrix
```

F08 - Battle

KAMUS

```
odds, monster_index, current_monster_index, increase, new_stat_value,  
monster_level, max_value, reward : integer
```

```
monster_ids, monster_indices, monster_dict : list
```

```
enemy_dict, monster, monster1, monster2, attacker, defender, enemy,  
first, second : dict
```

```
x, y, status : string
```

```
arena : boolean
```

```
status_effect, user_items : Matrix
```

F09 - Arena

KAMUS

i, monster_level, stage_cleared, reward, user_index : integer
element : string

F10 - Shop & Currency**KAMUS**

Item_id, prompt : string

monster_types, monster_shop_dict, item_shop_dict, item_ids,
amt_bought, monster_ids : list

Index, monster_index, stock, price, i, user_index, item_index :
integer

monster : dict

found : boolean

F11 - Laboratory**KAMUS**

user_index, index, price, max_hp : integer

A, prompt, user_oc = string

monster_dict : list

options : array of string

monster : dict

F12 - Shop Management**KAMUS**

item_shop_dict, monster_shop_dict, unadded_item_dict, monster_ids,
item_list, ids : list

prompt, item_id, stock, price, id, shop_type : string

index : integer

F13 - Monster Management

```
KAMUS
atk_power, def_power, hp, speed : integer

type, prompt, stat: string

min, max : Optional integer
```

F14 - Load

```
KAMUS
args.nama_folder : string

user_data, monster_data, monster_shop_data, monster_inventory_data,
item_shop_data, item_inventory_data : list

argparser : class
```

F15 - Save

```
KAMUS
directory : string

userfile :
    SEQFILE of
        (*) directory +"/user.csv" : data
        (1) Mark : string

monsterfile :
    SEQFILE of
        (*) directory +"/monster.csv" : data
        (1) Mark : string

monstershopfile:
    SEQFILE of
        (*) directory +"/monster_shop.csv" : data
        (1) Mark : string

itemshopfile :
    SEQFILE of
        (*) directory +"/item_shop.csv" : data
        (1) Mark : string
```

F16 - Exit

KAMUS
save_prompt : string

B03 - Monster Ball

KAMUS
enemy : dict

Desain Dekomposisi Algoritmik

F00 - RNG

function rng
{I.S. fungsi ini tidak dapat dipanggil}
{F.S. fungsi ini berfungsi untuk fungsi lain}

F01 - Register

function register
{I.S. user yang belum membuat akun, hanya dapat menggunakan fungsi menu & help.}
{F.S. user yang sudah register harus memilih salah satu monster dan dapat melakukan login}

F02 - Login

function login
{I.S. User tidak bisa menggunakan fungsi apapun {kecuali menu & help, login, dan exit}
{F.S. User berhasil login atau user tidak dapat login}

F03 - Logout

function logout ()
{I.S. User di dalam program}
{F.S. User keluar dari program}

F04 - Menu & help

procedure menu & help ()

{I.S. User yang belum login diberikan dua opsi, yaitu melakukan login atau register.}

{F.S. User yang sudah login dapat memanggil semua fungsi yang terdaftar}

F05 - Monster

function monster ()

{I.S. informasi dasar disimpan dalam owca-dex, sementara data level monster disimpan dalam inventory}

{F.S. atribut monster dikalkulasikan sesuai dengan level monster saat battle}

F06 - Potion

function potion ()

{I.S. Masing-masing jenis potion hanya dapat digunakan satu kali dalam satu battle}

{F.S. Atribut tambahan dari potion akan menyesuaikan dengan level monster}

F07 - Inventory

procedure inventory ()

{I.S. Fungsi inventory dapat memanggil semua item yang berada di inventory}

{F.S. menampilkan detail item dan monster yang kita panggil}

F08 - Battle

function battle ()

{I.S. Monster dipanggil oleh fungsi akan menampilkan ATK power, DEF power, HP, dan level}

{F.S. User akan mendapatkan opsi untuk Attack, switch, use potion, run, dan monster ball.}

F09 - Arena

function arena ()

{I.S. monster yang muncul pada arena random dari database}

{F.S. jika user memilih quit atau kalah, akan ditampilkan data coin,damage yang diberikan, dan damage yang diterima.}

F10 - Shop & currency

function shop & currency ()
{I.S. User membeli item, khusus monster tidak bisa membeli lebih dari satu}
{F.S. currency yang dimiliki user harus cukup, jika tidak maka transaksi gagal.}

F11 - Laboratory

function laboratory ()
{I.S. User dapat meng-upgrade monster dengan level maksimal level 5.}
{F.S. Setiap upgrade level memiliki harga yang berbeda yang dibayarkan dengan OC}

F12 - Shop management

function shop management ()
{I.S. Admin dapat mengatur item apa saja yang diperjualkan dalam Shop & Currency}
{F.S. Setelah admin melakukan pembaruan, Shop & Currency akan terupdate.}

F13 - Monster management

function monster management ()
{I.S. Admin diberi opsi untuk menampilkan semua monster, menambahkan monster baru, dan keluar}
{F.S. penambahan monster baru harus menginput data type, nama, ATK power, DEF power, dan HP.}

F14 - Load

function load ()
{I.S. Ketika data dipanggil dan fungsi ditemukan, akan langsung dijalankan di sistem}
{F.S. Jika fungsi tidak ditemukan, maka akan menampilkan data tidak ditemukan}

F15 - Save

```
procedure save ()  
{I.S. Jika nama data sudah ada data akan di-overwrite}  
{F.S. Jika nama data belum terdaftar, akan membuat data save baru}
```

F16 - Exit

```
procedure exit ()  
{I.S. User yang tidak melakukan save akan kehilangan data}  
{F.S. User yang telah melakukan save data-nya akan tersimpan}
```

B03 - Monster Ball

Spesifikasi F00 - RNG

```
function rng(lower : Optional integer, upper : Optional integer, x0 :  
Array, a : integer, c : integer, m : integer) -> int  
{fungsi untuk menggenerasi angka secara acak dari batas bawah lower ke  
batas atas upper}
```

ALGORITMA

```
x <- (a * x0[0] + c) mod m  
x0[0] <- x  
  
if lower = None and upper = None then  
    -> x  
else
```

```
-> int((x / (m - 1)) * abs(upper - lower + 1) +  
utils.min(lower,upper))
```

F01 - Register

```
function register(user_data : Matrix, current_user : Matrix, monster_data :  
Matrix, monster_inventory_data : Matrix) -> Tuple[Matrix, Matrix]:
```

```
{fungsi untuk membuat akun user baru dengan mengecek user_data yang telah  
ada}  
{akses : guest}
```

ALGORITMA

```
    if not utils.is_empty(current_user) then  
        output("Anda telah login dengan username {current_user1}, silahkan  
lakukan 'LOGOUT' sebelum melakukan register.")  
        time.sleep(1)  
        utils.clear_terminal()  
    else  
        while True do  
            username <- utils.strip(input("Masukkan username: "))  
            if utils.is_space(username) then  
                output("Username tidak boleh kosong")  
                time.sleep(1)  
                utils.remove_x_line_above(2)  
            else  
                break  
        while True do  
            password <- utils.strip(input("Masukkan password: "))  
            if utils.is_space(password) then  
                output("Password tidak boleh kosong")  
                time.sleep(1)  
                utils.remove_x_line_above(2)  
            else  
                break  
        # Jika sudah ada username yang terpakai  
        if utils.is_in_column(utils.slice_matrix(user_data, row_start = 1),  
1, username) then  
            output("Username {username} telah terpakai, silahkan gunakan  
username lain!")  
            time.sleep(1)  
            utils.clear_terminal()  
        elif not check_valid_characters(username) then  
            output("Username hanya boleh berisi alfabet, angka, underscore,  
dan strip!")  
            time.sleep(1)
```

```

        utils.clear_terminal()
    else
        # Jika tidak, maka digenerate hal yang diperlukan
        id <- generate_user_id(user_data)
        new_user_data <- [id, username, password, 'agent', '0']
        user_data.append(new_user_data)
        output("Anda telah mendaftarkan user {username}")
        time.sleep(1)
        utils.clear_terminal()
        monster_inventory_data <- choose_starter(id, monster_data,
monster_inventory_data)

    -> user_data, monster_inventory_data

```

```

function check_valid_characters(x : string) -> boolean
{fungsi untuk mengecek apakah karakter valid}

```

```

ALGORITMA
    valid_character <-
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_- "
    if len(x) = 0 then
        -> False
    else:
        for char in x
            if char not in valid_character then
                -> False
        -> True

```

```

function generate_user_id(user_data : Matrix) -> string
{Fungsi untuk membuat user ID baru}

```

```

ALGORITMA
existing_id <- utils.ascending_sort([user_data[i].0 for i in range(1,
len(user_data)) if not utils.is_space(user_data[i].0)])
    num <- 1
    while str(num) in existing_id do
        num <- num + 1
    -> str(num)

```

```

procedure choose_starter(user_id : string, monster_data : Matrix,
monster_inventory_data : Matrix) -> None
{Prosedur untuk memberi user monster di awal game}

```

```

ALGORITMA
monster_list <- utils.slice_matrix(monster_data,row_start = 1)

```

```

# Jika tidak ada monster di database, maka program diterminasi
if len(monster_list) = 0 then
    output("Tidak ada monster di database")
    time.sleep(1)
    utils.remove_xth_line_above(1)
    -> monster_inventory_data
else:
    # Ambil maksimal 3 monster untuk pilihan pertama
    starter_range <- utils.min(len(monster_list), 3)
    starter_ids <- []
    output("{ID':<10}{Type':<20}{ATK Power':<10}{DEF
Power':<10}{HP':<10}{Speed':<10}")
    for i in range(starter_range)

output("{monster_list[i][0]:<10}{monster_list[i][1]:<20}{monster_list[i][2]
:<10}{monster_list[i][3]:<10}{monster_list[i][4]:<10}{monster_list[i][5]:<1
0}")

    starter_ids.append(monster_list[i][0])
    while True do
        while True do
            prompt <- utils.strip(input("Masukkan ID: "))
            if utils.strip(prompt) not in starter_ids then
                output("Masukkan ID yang valid")
                time.sleep(1)
                utils.remove_x_line_above(2)
            else
                break
        id <- prompt

        while True do
            prompt = utils.strip(input("Apakah Anda yakin? (Y/N): "))
            if prompt.lower() = 'y' then
                monster_index <- utils.find_row(monster_list ,0, id)
                monster_inventory_data.append([user_id,
monster_listmonster_index.0,
'1',inventory.name_monster(user_id,monster_inventory_data),monster_listmons
ter_index.4])

                utils.remove_x_line_above(2)
                output("Selamat! Anda telah mendapatkan monster pertama
Anda")

                time.sleep(1.5)
                utils.clear_terminal()
                -> monster_inventory_data
            elif prompt.lower() = 'n' then
                utils.remove_x_line_above(3)
                break

```

F02 - Login

```
function login(user_data : Matrix, current_user : Array) -> Tuple[Array,  
Optional string]  
{fungsi untuk masuk ke dalam akun yang telah ada}  
{akses : guest}
```

ALGORITMA

```
# Cek apakah tidak ada user yang sedang menggunakan program  
if not utils.is_empty(current_user) then  
    output(f"Anda telah login dengan username {current_user.1},  
silahkan lakukan 'LOGOUT' sebelum melakukan register.")  
    time.sleep(1)  
    utils.remove_x_line_above(2)  
    -> current_user, None  
else  
    while True do  
        # Validasi username  
        username <- utils.strip(input("Masukkan username: "))  
        if utils.is_space(username) then  
            output("Username tidak boleh kosong")  
            time.sleep(1)  
            utils.remove_x_line_above(2)  
        else  
            break  
    while True do  
        # Validasi password  
        password <- utils.strip(input("Masukkan password: "))  
        if utils.is_space(password) then  
            output("Password tidak boleh kosong")  
            time.sleep(1)  
            utils.remove_x_line_above(2)  
        else  
            break  
    # Cari user  
    user_index <- utils.find_row(utils.slice_matrix(user_data,  
row_start = 1), 1, username) + 1  
    user_id <- user_datauser_index.0  
  
    if not utils.is_in_column(utils.slice_matrix(user_data, row_start =  
1), 1, username) then  
        output("Username tidak terdaftar!")  
        time.sleep(1)  
        utils.remove_x_line_above(4)  
        -> current_user, None  
    elif not password = user_datauser_index.2:  
        output("Password salah!")
```

```

        time.sleep(1)
        utils.remove_x_line_above(4)
        -> current_user, None
    else:
        current_user <- utils.copy_array(user_data[user_index])
        output("Anda telah login dengan user {current_user[1]}")
        time.sleep(1)
        utils.remove_x_line_above(4)
        -> current_user, user_id

```

F03 - Logout

```

function logout(current_user : Array) -> Tuple[Array, string]
{fungsi untuk keluar dari akun yang sedang digunakan}
{akses : agent, admin}

```

ALGORITMA

```

    # Jika kosong, maka tidak bisa logout
    if utils.is_empty(current_user) then
        output("Logout gagal!")
        output("Anda belum login, silahkan login terlebih dahulu sebelum
melakukan logout")
        time.sleep(1)
        utils.remove_x_line_above(3)
    else
        output("Anda keluar dari akun {current_user[1]}")
        time.sleep(1)
        utils.remove_xth_line_above(1)
        current_user <- []

    -> current_user, ''

```

F04 - Menu & Help

```

procedure help(current_user : Array)
{fungsi untuk menunjukkan tampilan bantuan bagi user}
{akses : agent, admin, guest}

```

ALGORITMA

```

    if utils.is_empty(current_user) then
        message <- f"""
=====
=====
===== HELP
=====

```



```
=====
=====
```

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

Login : Masuk ke dalam akun yang sudah terdaftar
Register : Membuat akun baru
Exit : Keluar dari aplikasi

Footnote:

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

```
=====
=====
```

```
    """
    elif current_user[3] == 'agent' then
        message <- f"""
```

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

Halo Agen {current_user[1]}. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

Logout : Keluar dari akun yang sedang digunakan
Inventory : Melihat owca-dex yang dimiliki oleh agen
Battle : Memulai pertarungan dengan monster
Arena : Bertanding dalam arena
Heal : Mengunjungi Healing Fountain
Shop : Membeli barang di Superduper Mega Superstore
Laboratory : Melakukan upgrade kepada monster
Save : Menyimpan data ke database
Exit : Keluar dari aplikasi

Footnote:

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

```
=====
=====
    """
    elif current_user[3] == 'admin' then
```

```

message <- f"""
=====
=====
===== HELP
=====
=====
=====

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

Logout : Keluar dari akun yang sedang digunakan
Shop   : Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan
Agent
Monster: Melakukan manajemen pada database monster
Save   : Menyimpan data ke database
Exit   : Keluar dari aplikasi

Footnote:
Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
Jangan lupa untuk memasukkan input yang valid

=====
=====
    """
    output(message)

```

F05 - Monster

Bukan fungsi/prosedur

F06 - Potion

Bukan fungsi/prosedur

F07 - Inventory

```

procedure inventory(current_user : Array, user_id : string, user_data :
Matrix, item_inventory_data : Matrix, monster_inventory_data : Matrix,
monster_data : Matrix) -> None
{prosedur untuk menunjukkan inventory dari user}

```




```
procedure print_enemy_monster() -> None
{prosedur untuk memprint monster musuh}
```

ALGORITMA

```
output("""
```

[illegible]

```
function randomize_enemy_level() -> integer
{fungsi untuk mencari level musuh secara acak}
```

ALGORITMA

```

odds <- rng.rng(1,15)
if odds = 1 then
  -> 5
elif odds <= 3 then
  -> 4
elif odds <= 6 then
  -> 3
elif odds <= 10 then
  -> 2
elif odds <= 15 then
  -> 1

```

```

function randomize_enemy(monster_data : integer) -> integer
{fungsi untuk mengacak musuh}

```

ALGORITMA

```

monster_ids <- []

for i in range(1, len(monster_data))
  monster_ids.append(monster_data[i,0])
monster_id <- monster_ids[rng.rng(0, len(monster_ids) - 1)]

-> monster_id

```

```

function load_enemy( monster_data: Matrix, monster_level: integer) ->
Dictionary
{fungsi untuk mengubah musuh menjadi dictionary}

```

ALGORITMA

```

monster_id <- randomize_enemy(monster_data)
monster_index <- utils.find_row(monster_data, index=0,
element=monster_id)

type <- monster_data[monster_index,1]
name <- 'Enemy' + type
atk_power <- int(int(monster_data[monster_index,2]) * (1 + (monster_level -
1) * 0.1))
def_power <- utils.min(int(int(monster_data[monster_index,3]) * (1 +
(monster_level - 1) * 0.1)), 50)
hp <- int(int(monster_data[monster_index,4]) * (1 + (monster_level - 1) *
0.1))
speed <- utils.min(int(int(monster_data[monster_index,5]) * (1 +
(monster_level - 1) * 0.1)), 50)
enemy_dict <- {'name' : name, 'type' : type, 'id': monster_id, 'level':
monster_level, 'atk_power': atk_power, 'def_power': def_power, 'hp': hp,
'max_hp' : hp, 'speed' : speed}
-> enemy_dict

```

```
function def switch_monster(monster_dict: DictList, current_monster_index:
integer = None) -> Tuple[int, Dictionary, bool]
{fungsi untuk mengganti monster dalam pertarungan}
```

```

    monster_indices <- []
    for i in range(len(monster_dict))
        if i != current_monster_index then
            monster_indices.append(str(i + 1))
    output("Pilih monstermu")
    for i, monster in enumerate(monster_dict)
        output("{i + 1}. {monster['name']}")

    while True do
        x <- utils.strip(input(">>> "))
        if x in monster_indices then
            while True do
                y <- utils.strip(input("Apakah kamu yakin? (Y/N): "))
                if y.lower() = 'y':
                    utils.remove_xth_line_above(1)
                    break
                elif y.lower() = 'n':
                    if current_monster_index = None then
                        utils.remove_x_line_above(2)
                        output("Pilih monster")
                        while True do
                            x <- utils.strip(input(">>> "))
                            if x not in monster_indices then
                                print("Masukkan input yang valid")
                                time.sleep(1)
                                utils.remove_x_line_above(2)
                            else
                                break
                        else
                            utils.remove_x_line_above(2)
                    ->
                    current_monster_index, monster_dict[current_monster_index], False
                else
                    output("Masukkan input yang valid")
                    time.sleep(1)
                    utils.remove_x_line_above(2)

                utils.clear_terminal()
                print_user_monster()
                current_monster_index <- int(x) - 1
                output("{monster_dict[current_monster_index]['name']}, Aku
memilih kamu")
                break
            else
                output("Masukkan input yang valid")

```

<pre> time.sleep(1) utils.remove_x_line_above(2) time.sleep(1.5) utils.clear_terminal() -> current_monster_index,monster_dict[current_monster_index],True </pre>
<pre> function calculate_increase(stat_name: str,monster: Dictionary,max_value: int = None) -> str {fungsi untuk menghitung pertambahan stat} </pre>
<pre> ALGORITMA if stat_name in status_effect[current_monster_index] then increase <- int(0.05 / 1.05 * monster[stat_name]) if increase = 0 then increase <- 1 if max_value is not None then new_stat_value <- utils.min(monster[stat_name] + increase, max_value) increase <- new_stat_value - monster[stat_name] -> f" ({increase})" else: -> "" </pre>
<pre> function select_action(arena: bool = False) -> string {fungsi untuk memilih aksi} </pre>
<pre> ALGORITMA options <- ['1','2','3','4'] if not arena then options.append('5') while True do output("Select Action:") output("1. Attack") output("2. Switch") output("3. Use Potion") output("4. Run") if not arena then print("5. Monster Ball") x <- utils.strip(input(">>> ")) if x in options then if arena then utils.remove_x_line_above(6) else utils.remove_x_line_above(7) break else then output("Masukkan input yang valid") time.sleep(1) if arena then </pre>

```

        utils.remove_x_line_above(7)
    else
        utils.remove_x_line_above(8)
-> x

```

```

function select_potion(user_items: Matrix,monster:
Dictionary,current_monster_index: int,status_effect: Matrix,monster_data:
Matrix) -> bool
{fungsi untuk memilih potion}

```

ALGORITMA

```

    # Mencari potion terdapat di index mana saja
    strength_index = utils.find_row(user_items, 1, 'strength')
    speed_index = utils.find_row(user_items, 1, 'speed')
    resilience_index = utils.find_row(user_items, 1, 'resilience')
    healing_index = utils.find_row(user_items, 1, 'healing')

    # Pilih potion
    if user_items and (user_items[strength_index][2] != 0 or
user_items[speed_index][2] !=0 or user_items[resilience_index][2] != 0 or
user_items[healing_index][2] != 0):
        while True:
            print("Select Potion:")
            print(f"1. Strength (Quantity: {user_items[strength_index][2]
if strength_index != -1 else 0})")
            print(f"2. Speed (Quantity: {user_items[speed_index][2] if
speed_index != -1 else 0})")
            print(f"3. Resilience (Quantity:
{user_items[resilience_index][2] if resilience_index != -1 else 0})")
            print(f"4. Healing (Quantity: {user_items[healing_index][2] if
healing_index != -1 else 0})")
            print("5. Cancel")
            x = utils.strip(input(">>> "))
            if x in ['1','2','3','4','5']:
                if x == '1':
                    if strength_index != -1 and
int(user_items[strength_index][2]) > 0:
                        if not 'atk_power' in
status_effect[current_monster_index]:
                            user_items[strength_index][2] =
user_items[strength_index][2] - 1
potion('atk_power',monster,current_monster_index,status_effect)
                            print(f"{monster['name']} meminum Strength
Potion")

                            time.sleep(1)
                            utils.remove_x_line_above(8)
                            return True

```



```

        else:
            print(f"{monster['name']} sudah meminum
Strength Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return False

        else:
            print("Anda tidak memiliki Strength Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return False

        elif x == '2':
            if speed_index != -1 and
int(user_items[speed_index][2]) > 0:
                if not 'speed' in
status_effect[current_monster_index]:
                    user_items[speed_index][2] =
user_items[speed_index][2] - 1
            potion('speed',monster,current_monster_index,status_effect,50)
            print(f"{monster['name']} meminum Speed
Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return True
        else:
            print(f"{monster['name']} sudah meminum Speed
Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return False
        else:
            print("Anda tidak memiliki Speed Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return False

        elif x == '3':
            if resilience_index != -1 and
int(user_items[resilience_index][2]) > 0:
                if not 'def_power' in
status_effect[current_monster_index]:
                    user_items[resilience_index][2] =
user_items[resilience_index][2] - 1
            potion('def_power',monster,current_monster_index,status_effect,50)

```

```

        print(f"{monster['name']} meminum Resilience
Potion")
        time.sleep(1)
        utils.remove_x_line_above(8)
        return True
    else:
        print(f"{monster['name']} sudah meminum
Resilience Potion")
        time.sleep(1)
        utils.remove_x_line_above(8)
        return False
    else:
        print("Anda tidak memiliki Resilience Potion")
        time.sleep(1)
        utils.remove_x_line_above(8)
        return False

    elif x == '4':
        if healing_index != -1 and
int(user_items[healing_index][2]) > 0:
            if not 'healing' in
status_effect[current_monster_index]:
                user_items[healing_index][2] =
user_items[healing_index][2] - 1
                monster['hp'] = heal(monster,monster_data)

            status_effect[current_monster_index].append('healing')
            print(f"{monster['name']} meminum Healing
Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return True
        else:
            print(f"{monster['name']} sudah meminum Healing
Potion")
            time.sleep(1)
            utils.remove_x_line_above(8)
            return False
    else:
        print("Anda tidak memiliki Healing Potion")
        time.sleep(1)
        utils.remove_x_line_above(8)
        return False

    elif x == '5':
        time.sleep(1)
        utils.remove_x_line_above(7)
        return False

    else:

```

<pre> print("Masukkan input yang valid") time.sleep(1) utils.remove_x_line_above(8) else: print("Anda tidak memiliki potion") time.sleep(1) utils.remove_xth_line_above(1) return False </pre>
<pre> (function) def execute_turn(first: Dictionary, second: Dictionary, monster: Dictionary, enemy: Dictionary, status_effect: Matrix, current_monster_index: int) -> Tuple[int, int] </pre>
<pre> first_damage = attack(first,second, monster, enemy,status_effect, current_monster_index) if second['hp'] <= 0: print(f"{second['name']} fainted") return first_damage, 0 # Jika lawan mati, maka tidak terkena damage dari serangan yang akan datang itu second_damage = attack(second,first, monster, enemy,status_effect, current_monster_index) if first['hp'] <= 0: print(f"{first['name']} fainted") return first_damage, second_damage return first_damage, second_damage </pre>
<pre> (function) def run(monster: Dictionary, enemy: Dictionary, escape_attempt: int) -> bool </pre>
<pre> if monster['speed'] >= enemy['speed']: return True else: escape_odds = ((monster['speed'] * 128) // enemy['speed'] + 30 * escape_attempt) % 256 if rng.rng(1,255) <= escape_odds: return True else: </pre>

return False
<pre>(function) def heal(monster: Dictionary, monster_data: Matrix) -> int</pre>
<pre> monster_index = utils.find_row(monster_data, index = 0, element = monster['id']) # Mencari index monster max_hp = int(int(monster_data[monster_index][4]) * (1 + (monster['level'] - 1) * 0.1)) # Mencari darah maksimal monster return int(utils.min(monster['hp'] + 0.25 * max_hp, max_hp)) # Mengubah darah monster sekarang</pre>
<pre>(function) def potion(status: str, monster: Dictionary, current_monster_index: int, status_effect: Matrix, max_value: int = None) -> None</pre>
<pre> status_effect[current_monster_index].append(status) if max_value is not None: if monster[status] <= 20: # Jika boost sangat kecil, maka dibulatkan ke 11 monster[status] = monster[status] + 1 else: monster[status] = utils.min(int(1.05 * monster[status]),max_value) else: monster[status] = int(1.05 * monster[status])</pre>
<pre>(function) def battle(current_user: Array, monster_level: int, user_data: Matrix, user_id: str, monster_inventory_data: Matrix, item_inventory_data: Matrix, monster_data: Matrix, arena: bool, reward: int = None) -> Tuple[int, int, bool, Matrix, Matrix]</pre>
<pre> total_damage_taken = 0 total_damage_dealt = 0 victory = False</pre>

```

# Cek apakah user sudah login atau merupakan user
if utils.is_empty(current_user):
    print("Anda belum login")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return total_damage_dealt, total_damage_taken, victory,
item_inventory_data,monster_inventory_data
elif utils.strip(current_user[3]) != 'agent':
    print("Anda bukan Agent")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return total_damage_dealt, total_damage_taken, victory,
item_inventory_data,monster_inventory_data
else:
    utils.clear_terminal()

    # Inisialisasi
    monster_dict =
inventory.load_user_monsters(user_id,monster_inventory_data,monster_data,
battle = True)
    enemy = load_enemy(monster_data,monster_level)
    user_items =
inventory.get_user_inventory(user_id,item_inventory_data)

    # Jika tidak ada monster, maka tidak bisa bertarung
    if utils.is_empty(monster_dict):
        print("Anda tidak memiliki monster yang bisa bertarung")
        time.sleep(1)
        utils.remove_x_line_above(2)
        return 0, 0, False, item_inventory_data,monster_inventory_data
    else:
        status_effect = [['']] for i in range(len(monster_dict)) #
Matrix tempat mengisi status effect monster
        escape_attempt = 0
        print_enemy_monster()
        print(f"RAWRRR, Monster {enemy['type']} telah muncul")
        time.sleep(1.5)
        current_monster_index, monster, action_executed =
switch_monster(monster_dict) # Pilih monster pertama

        while monster['hp'] > 0 and enemy['hp'] > 0:
            utils.clear_terminal()
            action_executed = False
            show_both_stat(monster,
enemy,status_effect,current_monster_index)
            action = select_action(arena)
            if action == '1':
                if monster['speed'] > enemy['speed']:

```

```

        damage_dealt, damage_taken =
execute_turn(monster,enemy,monster, enemy,status_effect,
current_monster_index)
        total_damage_dealt = total_damage_dealt +
damage_dealt
        total_damage_taken = total_damage_taken +
damage_taken

        elif monster['speed'] < enemy['speed']:
            damage_taken, damage_dealt =
execute_turn(enemy,monster,monster, enemy,status_effect,
current_monster_index)
            total_damage_dealt = total_damage_dealt +
damage_dealt
            total_damage_taken = total_damage_taken +
damage_taken

        else:
            random_priority = rng.rng(1,2)

            if random_priority == 1:
                damage_taken, damage_dealt =
execute_turn(enemy,monster,monster, enemy,status_effect,
current_monster_index)
                total_damage_dealt = total_damage_dealt +
damage_dealt
                total_damage_taken = total_damage_taken +
damage_taken

            elif random_priority == 2:
                damage_dealt, damage_taken =
execute_turn(monster,enemy,monster, enemy,status_effect,
current_monster_index)
                total_damage_dealt = total_damage_dealt +
damage_dealt
                total_damage_taken = total_damage_taken +
damage_taken

            time.sleep(1.5)

        elif action == '2':
            if len(monster_dict) > 1:
                current_monster_index,monster,action_executed =
switch_monster(monster_dict, current_monster_index)

                if action_executed:
                    damage_taken = attack(enemy,monster,monster,
enemy,status_effect, current_monster_index)
                    total_damage_taken = total_damage_taken +
damage_taken

```

```

        else:
            print("Kamu tidak memiliki monster lain")
            time.sleep(1.5)

        elif action == '3':
            action_executed =
select_potion(user_items,monster,current_monster_index,status_effect,monster_data)

            if action_executed:
                damage_taken = attack(enemy,monster,monster,
enemy,status_effect, current_monster_index)
                total_damage_taken = total_damage_taken +
damage_taken
                time.sleep(1.5)

        elif action == '4':
            if run(monster,enemy,escape_attempt):
                break
            else:
                print("Anda tidak bisa kabur")
                time.sleep(1)
                damage_taken = attack(enemy,monster,monster,
enemy,status_effect, current_monster_index)
                total_damage_taken = total_damage_taken +
damage_taken
                time.sleep(1.5)

        elif action == '5':
            action_executed =
monster_ball.monster_ball(user_id,enemy,user_items,monster_inventory_data)
            if enemy['hp'] != 0 and action_executed:
                damage_taken = attack(enemy,monster,monster,
enemy,status_effect, current_monster_index)
                total_damage_taken = total_damage_taken +
damage_taken
                time.sleep(1.5)

        # Reward dibagikan berdasarkan menang atau tidak
        if enemy['hp'] <= 0:
            print(f"Selamat Anda telah berhasil mengalahkan monster
{enemy['type']}")
            victory = True
            if reward:
                print(f"Total OC = {reward}")
                user_data[utils.find_row(user_data, index = 0, element
= user_id)][4] = str(int(user_data[utils.find_row(user_data, index = 0,
element = user_id)][4]) + reward)
            elif monster['hp'] <= 0:
                print(f"Anda dikalahkan monster {enemy['type']}")

```

```

        victory = False
    else:
        print("Anda berhasil kabur dari battle")
        victory = False

    # Update data
    item_inventory_data = update_item_inventory_data(user_id,
user_items, item_inventory_data)
    monster_inventory_data = update_monster_inventory_data(user_id,
monster_dict, monster_inventory_data)
    monster_dict = []
    user_items = []
    enemy = {}
    time.sleep(2)
    utils.clear_terminal()

    return total_damage_dealt, total_damage_taken, victory,
item_inventory_data,monster_inventory_data

```

```

(function) def update_item_inventory_data(
    user_id: str,
    user_items: Matrix,
    item_inventory_data: Matrix
) -> Matrix

```

```

    for item in item_inventory_data:
        if item[0] == user_id:
            for user_item in user_items:
                if item[1] == user_item[1]:
                    item[2] = str(user_item[2])
    return utils.remove_row(item_inventory_data, 2,'0')

```

```

(function) def update_monster_inventory_data(
    user_id: str,
    monster_dict: DictList,
    monster_inventory_data: Matrix
) -> Matrix

```

```

    for monster in monster_inventory_data:
        if monster[0] == user_id:
            for user_monster in monster_dict:
                if user_monster['name'] == monster[3]:
                    monster[4] = str(user_monster['hp'])
    return monster_inventory_data

```


F09 - Arena

```
(function) def arena(  
    current_user: Array,  
    user_data: Matrix,  
    user_id: str,  
    monster_inventory_data: Matrix,  
    item_inventory_data: Matrix,  
    monster_data: Matrix  
) -> Tuple[Matrix, Matrix]
```

ALGORITMA

```
    total_damage_dealt = 0  
    total_damage_taken = 0  
    stage_cleared = 0  
  
    # Cek apakah user sudah login atau merupakan agen  
    if utils.is_empty(current_user):  
        print("Anda belum login")  
        time.sleep(1)  
        utils.remove_x_line_above(2)  
        return item_inventory_data, monster_inventory_data  
    elif utils.strip(current_user[3]) != 'agent':  
        print("Anda bukan Agent")  
        time.sleep(1)  
        utils.remove_x_line_above(2)  
        return item_inventory_data, monster_inventory_data  
    else:  
        utils.clear_terminal()  
        for i in range(5):  
            print(f"Stage {i + 1}") # Print nomor stage  
            time.sleep(2)  
            monster_level = i + 1  
            damage_dealt, damage_taken,  
victory, item_inventory_data, monster_inventory_data =  
battle.battle(current_user, monster_level, user_data, user_id,  
monster_inventory_data, item_inventory_data, monster_data, True)  
            total_damage_dealt = total_damage_dealt + damage_dealt  
            total_damage_taken = total_damage_taken + damage_taken  
            if not victory:  
                break  
            else:  
                stage_cleared = stage_cleared + 1  
  
        # Hitung reward  
        if stage_cleared == 0:  
            reward = 0  
        else:
```

```

        reward = 50 * (2 ** (stage_cleared - 1))

        # Update OC user
        user_index = utils.find_row(user_data, index = 0, element =
user_id)
        user_data[user_index][4] = str(int(user_data[user_index][4]) +
int(reward))

        # Print hasil
        print(f"Total damage dealt : {total_damage_dealt}")
        print(f"Total damage taken : {total_damage_taken}")
        print(f"Stage cleared: {stage_cleared}")
        if reward > 0:
            print(f"Anda mendapat reward {reward} OWCA Coins")
        else:
            print("Anda tidak mendapat OWCA Coins")
        time.sleep(3)
        utils.clear_terminal()
        return item_inventory_data,monster_inventory_data
    
```

F10 - Shop & Currency

Bukan fungsi/prosedur

F11 - Laboratory

```
(function) def print_laboratory() -> None
```

Algoritma

```
print("""
```

```
(function) def laboratory(
    current_user: Array,
    user_id: str,
    monster_inventory_data: Matrix,
    monster_data: Matrix,
    user_data: Matrix
) -> Tuple[Matrix, Matrix]
```

Algoritma

```
# Cek apakah user sudah login atau merupakan agent
if utils.is_empty(current_user):
    print("Anda belum login")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return monster_inventory_data, user_data
elif utils.strip(current_user[3]) != 'agent':
    print("Anda bukan Agent")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return monster_inventory_data, user_data
else:
    utils.clear_terminal()
    print_laboratory()

    while True:
        user_index = utils.find_row(utils.slice_matrix(user_data,
row_start = 1), 0, user_id) + 1 # Cari index user
        user_oc = user_data[user_index][4] # Cari jumlah OC user
        print(f"Jumlah OWCA Coins Anda sekarang: {user_oc}")
        monster_dict =
inventory.load_user_monsters(user_id,monster_inventory_data,monster_data) #
Load dictionary list

print(f"{'No.':<4}{'Name':<20}{'Type':<20}{'Level':<10}{'HP':<10}")
    for i, monster in enumerate(monster_dict):
        print(f"{i +
1:<4}{utils.title(monster['name']):<20}{monster['type']:<20}{monster['level
']:<10}{monster['hp']}/{monster['max_hp']}")
        options = [str(i + 1) for i in range(len(monster_dict))]
        print("Upgrade Price")
        print("Level 1 -> Level 2: 300 OC")
        print("Level 2 -> Level 3: 500 OC")
        print("Level 3 -> Level 4: 700 OC")
        print("Level 4 -> Level 5: 900 OC")
        while True:
            a = utils.strip(input("Pilih monster atau x untuk
keluar:"))
            if a in options:
```

```

        monster_inventory_data, user_data =
level_up(user_oc, user_index, user_id, user_data, monster_dict, monster_inventor
y_data, monster_data, a)
        break
    elif a == 'x':
        utils.clear_terminal()
        return monster_inventory_data, user_data
    else:
        print("Masukkan input yang valid")
        time.sleep(1)
        utils.remove_x_line_above(2)

```

```

(function) def level_up(
    user_oc: str,
    user_index: int,
    user_id: str,
    user_data: Matrix,
    monster_dict: DictList,
    monster_inventory_data: Matrix,
    monster_data: Matrix,
    index: int
) -> Tuple[Matrix, Matrix]

```

Algoritma

```

    index = int(index) - 1
    monster = monster_dict[index]

    # Cek level monster
    if int(monster['level']) >= 5:
        print("Maaf, monster yang Anda pilih sudah memiliki level
maksimum")
        time.sleep(1)
        utils.clear_terminal()
        print_laboratory()
    else:
        price = 300 + 200 * (int(monster['level'])-1)
        print(f"{monster['name']} akan diupgrade ke level {monster['level']
+ 1}.")
        print(f"Harga untuk melakukan upgrade {monster['name']} adalah
{price}")
        while True:
            prompt = utils.strip(input("Lanjutkan upgrade? (Y/N) "))
            if prompt.lower() == 'y':
                utils.remove_xth_line_above(1)
                if int(user_oc) >= price: # Cek OC user
                    user_data[user_index][4] =
str(int(user_data[user_index][4]) - price)
                    monster['level'] = str(int(monster['level']) + 1)

```

```

        monster_inventory_data =
change_level(user_id,monster,monster_inventory_data,monster_data)
        print(f"Selamat, {monster['name']} berhasil diupgrade
ke level ke level {monster['level']}")
        break
    else:
        print("OC Anda tidak cukup")
        break

    elif prompt.lower() == 'n':
        break
    else:
        print("Masukkan input yang valid")
        time.sleep(1)
        utils.remove_x_line_above(2)
time.sleep(1)
utils.clear_terminal()
print_laboratory()
return monster_inventory_data,user_data

```

```

(function) def change_level(
    user_id: str,
    monster: Dictionary,
    monster_inventory_data: Matrix,
    monster_data: Matrix
) -> Matrix

```

Algoritma

```

    max_hp = int((1 + 0.1 *(int(monster['level']) - 1)) *
int(monster_data[utils.find_row(monster_data,index = 0, element =
monster['id'])][4]))
    for i in monster_inventory_data:
        if i[0] == user_id and i[3] == monster['name']:
            i[2] = monster['level']
            i[4] = str(max_hp)
            return monster_inventory_data
    return monster_inventory_data

```

F12 - Shop Management

```

(function) def shop_management(current_user : Array, item_shop_data :
Matrix, monster_shop_data : Matrix, monster_data : Matrix) -> Tuple[Matrix,
Matrix]:

```

```

    # Cek apakah user sudah login atau merupakan Admin

```

```

if utils.is_empty(current_user) then
    print("Anda belum login")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return item_shop_data, monster_shop_data
elif utils.strip(current_user[3]) != 'admin' then
    print("Anda bukan Admin")
    time.sleep(1)
    utils.remove_x_line_above(2)
    return item_shop_data, monster_shop_data
else
    utils.clear_terminal()
    print_shop_management()

    # Load dictionary list
    item_shop_dict ← shop.load_item_shop(item_shop_data)
    monster_shop_dict ←
shop.load_monster_shop(monster_shop_data,monster_data)

    while True:
        prompt ←utils.strip(input("Pilih aksi
(lihat/tambah/ubah/hapus/keluar): "))
        if prompt = 'lihat' then
            shop.look(item_shop_dict,monster_shop_dict)
        elif prompt = 'tambah' then
            item_shop_data, item_shop_dict, monster_shop_data,
monster_shop_dict ← add(item_shop_data, monster_shop_data, monster_data,
item_shop_dict,monster_shop_dict)
        elif prompt = 'ubah' then
            item_shop_data, item_shop_dict, monster_shop_data,
monster_shop_dict ← change(item_shop_data, monster_shop_data,
item_shop_dict, monster_shop_dict)
        elif prompt = 'hapus' then
            item_shop_data, item_shop_dict, monster_shop_data,
monster_shop_dict ←
remove(item_shop_data,item_shop_dict,monster_shop_data,monster_shop_dict,mo
nster_data)
        elif prompt = 'keluar' then
            utils.clear_terminal()
            return item_shop_data, monster_shop_data
        else
            print("Masukkan input yang valid")
            time.sleep(1)
            utils.remove_x_line_above(2)

(function) def add(item_shop_data : Matrix, monster_shop_data : Matrix,
monster_data : Matrix, item_shop_dict : DictList, monster_shop_dict : DictList) ->
Tuple[Matrix, Dictionary, Matrix, Dictionary]:

```

```

while True do
  prompt ← utils.strip(input("Mau menambah apa? (item/monster): "))
  if prompt = "item" then
    item_shop_data, item_shop_dict = add_item(item_shop_data,
item_shop_dict)
    break
  elif prompt = "monster" then
    monster_shop_data, monster_shop_dict = add_monster(monster_shop_data,
monster_data, monster_shop_dict)
    break
  else
    Output ("Masukkan input yang valid")
    time.sleep(1)
    utils.remove_x_line_above(2)
  return item_shop_data, item_shop_dict, monster_shop_data, monster_shop_dict

```

F13 - Monster Management

```

(function) def monster_management(current_user : Array, monster_data : Matrix,
monster_inventory_data : Matrix, monster_shop_data : Matrix) -> Tuple[Matrix,
Matrix, Matrix]:

```

```

# Cek apakah user sudah login atau merupakan Admin
if utils.is_empty(current_user)then
  output("Anda belum login")
  time.sleep(1)
  utils.remove_x_line_above(2)
  return monster_data, monster_inventory_data, monster_shop_data
elif utils.strip(current_user[3]) != 'admin' then
  output("Anda bukan Admin")
  time.sleep(1)
  utils.remove_x_line_above(2)
  return monster_data, monster_inventory_data, monster_shop_data
else
  utils.clear_terminal()
  print_monster_management()

# Pilihan opsi
while True do
  output("1. Tampilkan semua Monster")
  output("2. Tambah Monster baru")
  output("3. Hapus Monster")
  output("4. Keluar")
  prompt ← utils.strip(input("Pilih aksi: "))
  if prompt = '1' then
    show_monster(monster_data)
  elif prompt = '2' then
    monster_data ← add_monster(monster_data)
  elif prompt = '3' then
    monster_data, monster_inventory_data, monster_shop_data =
remove_monster(monster_data, monster_inventory_data, monster_shop_data)
  elif prompt = '4' then

```

```

        time.sleep(1)
        utils.clear_terminal()
        return monster_data, monster_inventory_data, monster_shop_data
    else
        output("Masukkan input yang valid")
        time.sleep(1)
        utils.remove_x_line_above(6)

```

F14 - Load

```

(function) def load() -> Tuple[Optional[Matrix], Optional[Matrix],
Optional[Matrix], Optional[Matrix], Optional[Matrix]]:

    argparser ← argparse.ArgumentParser()
    argparser.add_argument('nama_folder', nargs ← '?', help←'Nama folder
tempat data tersimpan')
    args ← argparser.parse_args()
    directory ← args.nama_folder
    if not directory then
        output("Tidak ada nama folder yang diberikan")
        output("Usage : python main.py <nama_folder>")
        return None, None, None, None, None, None
    else
        if os.path.isdir(directory) then
            user_data = csv_parser.parse_csv(directory + '/user.csv')
            monster_data = csv_parser.parse_csv(directory + '/monster.csv')
            monster_shop_data = csv_parser.parse_csv(directory +
'/monster_shop.csv')
            monster_inventory_data = csv_parser.parse_csv(directory +
'/monster_inventory.csv')
            item_shop_data = csv_parser.parse_csv(directory +
'/item_shop.csv')
            item_inventory_data = csv_parser.parse_csv(directory +
'/item_inventory.csv')
            if None not in (user_data, monster_data, monster_shop_data,
monster_inventory_data, item_shop_data, item_inventory_data):
                print_welcome_to_owca()
                time.sleep(1.5)
                utils.clear_terminal()
                return user_data, monster_data, monster_shop_data,
monster_inventory_data, item_shop_data, item_inventory_data
            else
                output(f"Folder {directory} tidak ditemukan")
                return None, None, None, None, None, None

```

F15 - Save


```

directory ← input("Masukkan nama folder penyimpanan: ")

if os.path.isdir(directory) then
  csv_parser.generate_csv(user_data, directory + '/user.csv')
  csv_parser.generate_csv(monster_data, directory + '/monster.csv')
  csv_parser.generate_csv(monster_shop_data, directory
+ '/monster_shop.csv')
  csv_parser.generate_csv(monster_inventory_data, directory +
'/monster_inventory.csv')
  csv_parser.generate_csv(item_shop_data, directory +
'/item_shop.csv')
  csv_parser.generate_csv(item_inventory_data, directory +
'/item_inventory.csv')
else
  os.mkdir(directory)
  output((f"Folder tidak ditemukan sehingga dibuat folder pada
{directory}"))
  csv_parser.generate_csv(user_data, directory + '/user.csv')
  csv_parser.generate_csv(monster_data, directory + '/monster.csv')
  csv_parser.generate_csv(monster_shop_data, directory
+ '/monster_shop.csv')
  csv_parser.generate_csv(monster_inventory_data, directory +
'/monster_inventory.csv')
  csv_parser.generate_csv(item_shop_data, directory +
'/item_shop.csv')
  csv_parser.generate_csv(item_inventory_data, directory +
'/item_inventory.csv')

```

F16 - Exit

```

procedure exit(user_data : Matrix, monster_data : Matrix, monster_shop_data
: Matrix, monster_inventory_data : Matrix, item_shop_data : Matrix,
item_inventory_data : Matrix)

save_prompt <- input("Apakah Anda mau melakukan penyimpanan file yang sudah
diubah? (Y/N): ")
do save_prompt.lower() ≠ 'y' and save_prompt.lower() ≠ 'n':
  output("Masukkan input yang valid")
  save_prompt <- input("Apakah Anda mau melakukan penyimpanan file
yang sudah diubah? (Y/N): ")
if save_prompt.lower() = 'y' then
  save.save(user_data, monster_data, monster_shop_data,
monster_inventory_data, item_shop_data, item_inventory_data)

```

B03 - Monster Ball

```
(function) def catch(user_id : str, enemy : Dictionary,  
monster_inventory_data : Matrix):
```

ALGORITMA

```
odds ← rng.rng(1,100) # Mencari angka dari 1-100  
  
# Makin rendah hp musuh, makin tinggi peluang menangkapnya  
if enemy['level'] = 1 then  
    if odds < int((1-(enemy['hp']/enemy['max_hp'])) * 75) then  
        output("Monster berhasil ditangkap")  
        name ← inventory.name_monster(user_id, monster_inventory_data)  
        monster_inventory_data.append([user_id, enemy['id'],  
enemy['level'], name, enemy['hp']])  
        enemy['hp'] ← 0  
    else  
        output("Monster lepas dari Monster Ball")  
elif enemy['level'] = 2 then  
    if odds < int((1-(enemy['hp']/enemy['max_hp'])) * 50) then  
        output("Monster berhasil ditangkap")  
        name ← inventory.name_monster(user_id, monster_inventory_data)  
        monster_inventory_data.append([user_id, enemy['id'],  
enemy['level'], name, enemy['hp']])  
        enemy['hp'] ← 0  
    else  
        output("Monster lepas dari Monster Ball")  
elif enemy['level'] = 3:  
    if odds < int((1-(enemy['hp']/enemy['max_hp'])) * 25) then  
        output("Monster berhasil ditangkap")  
        name ← inventory.name_monster(user_id, monster_inventory_data)  
        monster_inventory_data.append([user_id, enemy['id'],  
enemy['level'], name, enemy['hp']])  
        enemy['hp'] ← 0  
    else:  
        output("Monster lepas dari Monster Ball")  
elif enemy['level'] = 4 then  
    if odds < int((1-(enemy['hp']/enemy['max_hp'])) * 10):  
        output("Monster berhasil ditangkap")  
        name ← inventory.name_monster(user_id, monster_inventory_data)  
        monster_inventory_data.append([user_id, enemy['id'],  
enemy['level'], name, enemy['hp']])  
        enemy['hp'] ← 0  
    else:  
        output("Monster lepas dari Monster Ball")  
elif enemy['level'] = 5 then  
    if odds < int((1-(enemy['hp']/enemy['max_hp'])) * 5):  
        print("Monster berhasil ditangkap")
```

```

        name ← inventory.name_monster(user_id,monster_inventory_data)
        monster_inventory_data.append([user_id, enemy['id'],
enemy['level'], name, enemy['hp']])
        enemy['hp'] ← 0
    else
        output("Monster lepas dari Monster Ball")

```

```

(function) def monster_ball(user_id : str, enemy : Dictionary, user_items :
Matrix, monster_inventory_data : Matrix) -> bool:

```

```

monster_ball_index ← utils.find_row(user_items, 1, 'monsterball') # Cari
index monster ball di inventory
    if monster_ball_index != -1 and user_items[monster_ball_index][2] > 0
then
    output(f"Monster Ball (Quantity:
{user_items[monster_ball_index][2]})") # Print jumlah monster ball
    while True:
        x ← utils.strip(input("Catch? (Y/N) "))
        if x.lower() = 'y' then
            output("Kamu melempar Monster Ball")
            user_items[monster_ball_index][2] =
user_items[monster_ball_index][2] - 1
            time.sleep(1)
            utils.remove_x_line_above(3)
            time.sleep(3)
            catch(user_id,enemy,monster_inventory_data)
            return True
        elif x.lower() = 'n' then
            utils.remove_x_line_above(2)
            return False
        else
            output("Masukkan input yang valid")
            time.sleep(1)
            utils.remove_x_line_above(2)
    else
        output("Anda tidak memiliki monster ball")
        time.sleep(1)
        utils.remove_x_line_above(1)
        return False

```

Lampiran Hasil Pengujian Program

Tautan Git : <https://github.com/Labpro-21/if1210-2024-tubes-k10-g.git>

Fitur	Hasil Pengujian
--------------	------------------------

F01-Register

Gambar 1.1. Register

```
>>> register
Masukkan username: purry
Masukkan password: spakbor
```

Gambar 1.2. Pilih monster

ID	Type	ATK	Power	DEF	Power	HP	Speed
1	Pikachow	350		10		4500	30
2	Bulbu	450		10		6000	30
3	Zeze	600		10		6500	30

```
Masukkan ID: 3
Apakah Anda yakin? (Y/N): y
```

F02-Login

Gambar 1.3. Login sukses

```
>>> login
Masukkan username: purry
Masukkan password: spakbor
Anda telah login dengan user purry
```

Gambar 1.4. Login gagal

```
>>> login
Masukkan username: purry
Masukkan password: spak
Password salah!
```

F03-Logout

Gambar 1.5. Logout

```
>>> logout  
Anda keluar dari akun purry  
█
```

F04-Menu & Help

Gambar 1.6. Help sebelum login

```
>>> help

=====
=====  HELP  =====
=====

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

Login   : Masuk ke dalam akun yang sudah terdaftar
Register : Membuat akun baru
Exit    : Keluar dari aplikasi

Footnote:
Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
Jangan lupa untuk memasukkan input yang valid

=====

>>> █
```

Gambar 1.7. Help setelah login

```
>>> help

=====
=====  HELP  =====
=====

Halo Agen 1. Kamu memanggil command HELP. Kamu memilih jalan
yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat
kamu lakukan sekarang:

Logout    : Keluar dari akun yang sedang digunakan
Inventory : Melihat owca-dex yang dimiliki oleh agen
Battle    : Memulai pertarungan dengan monster
Arena     : Bertanding dalam arena
Heal      : Mengunjungi Healing Fountain
Shop      : Membeli barang di Superduper Mega Superstore
Laboratory : Melakukan upgrade kepada monster
Save      : Menyimpan data ke database
Exit      : Keluar dari aplikasi

Footnote:
Untuk menggunakan aplikasi, silahkan masukkan nama fungsi yang terdaftar
Jangan lupa untuk memasukkan input yang valid

=====

>>> █
```

F05-Monster

Gambar 1.8. Monster jika bukan admin

```
>>> monster
Anda bukan Admin
█
```

Gambar 1.9. Monster jika admin

```
MONSTER MANAGEMENT

1. Tampilkan semua Monster
2. Tambah Monster baru
3. Hapus Monster
4. Keluar
Pilih aksi: █
```

F06-Potion

Gambar 1.10. Potion

```
Anda tidak memiliki potion
█
```

Gambar 1.11. Potion dalam battle

```
Player                                     Enemy
Name      : skibidi                      Name      : EnemyMoonfang
Type       : Zeze                        Type       : Moonfang
Level      : 1                          Level      : 1
Attack Power : 600                      Attack Power : 400
Defense Power : 10                      Defense Power : 25
Speed       : 30                        Speed       : 15
HP          : 6108/6500                 HP          : 4627/5000
Anda tidak memiliki potion
█
```


F07-Inventory

Gambar 1.12. Inventory



Gambar 1.13. Battle

```

Player                                     Enemy
Name      : skibidi                      Name      : EnemyMoonfang
Type       : Zeze                        Type       : Moonfang
Level      : 1                          Level      : 1
Attack Power : 600                      Attack Power : 400
Defense Power : 10                      Defense Power : 25
Speed      : 30                         Speed      : 15
HP         : 6500/6500                  HP         : 5000/5000
Select Action:
1. Attack
2. Switch
3. Use Potion
4. Run
>>> █

```

Gambar 1.14. Tampilan Attack

```

Player                                     Enemy
Name      : skibidi                      Name      : EnemyMoonfang
Type       : Zeze                        Type       : Moonfang
Level      : 1                          Level      : 1
Attack Power : 600                      Attack Power : 400
Defense Power : 10                      Defense Power : 25
Speed      : 30                         Speed      : 15
HP         : 6500/6500                  HP         : 5000/5000
Attack: 498 (~17%), Reduced by: 125 (25%)
skibidi attacks EnemyMoonfang for 373 damage.
█

```

Gambar 1.15. Tampilan potion

```

Player                                     Enemy
Name      : skibidi                      Name      : EnemyMoonfang
Type       : Zeze                        Type       : Moonfang
Level      : 1                          Level      : 1
Attack Power : 600                      Attack Power : 400
Defense Power : 10                      Defense Power : 25
Speed      : 30                         Speed      : 15
HP         : 6108/6500                  HP         : 4627/5000
Anda tidak memiliki potion
█

```

Gambar 1.16. Tampilan menang battle

Player		Enemy	
Name	: skibidi	Name	: EnemyMoonfang
Type	: Zeze	Type	: Moonfang
Level	: 1	Level	: 1
Attack Power	: 600	Attack Power	: 400
Defense Power	: 10	Defense Power	: 25
Speed	: 30	Speed	: 15
HP	: 2321/6500	HP	: 0/5000

EnemyMoonfang fainted
Selamat Anda telah berhasil mengalahkan monster Moonfang

Gambar 1.17. Tampilan kabur

Player		Enemy	
Name	: skibidi	Name	: EnemyMoonloo
Type	: Zeze	Type	: Moonloo
Level	: 1	Level	: 2
Attack Power	: 600	Attack Power	: 880
Defense Power	: 10	Defense Power	: 27
Speed	: 30	Speed	: 16
HP	: 2321/6500	HP	: 6600/6600

Anda berhasil kabur dari battle

Gambar 1.18. Selesai battle

Total damage dealt : 5328
Total damage taken : 4179
Stage cleared: 1
Anda mendapat reward 50 OWCA Coins

F09-Arena

Gambar 1.19. Arena



Gambar 1.20. Shop



Gambar 1.21. Shop : lihat



Gambar 1.22. Shop : beli

```
>>> shop
```



```
Pilih aksi (lihat/beli/keluar):beli  
Jumlah OWCA Coins mu sekarang adalah 50  
Mau beli apa? (item/monster): item  
Masukkan ID: █
```

Gambar 1.23. Laboratory

```
>>> laboratory

LABORATORY

Jumlah OWCA Coins Anda sekarang: 50
No. Name           Type           Level   HP
1  Skibidi         Zeze           1       2321/6500
Upgrade Price
Level 1 -> Level 2: 300 OC
Level 2 -> Level 3: 500 OC
Level 3 -> Level 4: 700 OC
Level 4 -> Level 5: 900 OC
Pilih monster atau x untuk keluar:█
```

Gambar 1.24. Upgrade monster

```
>>> laboratory

LABORATORY

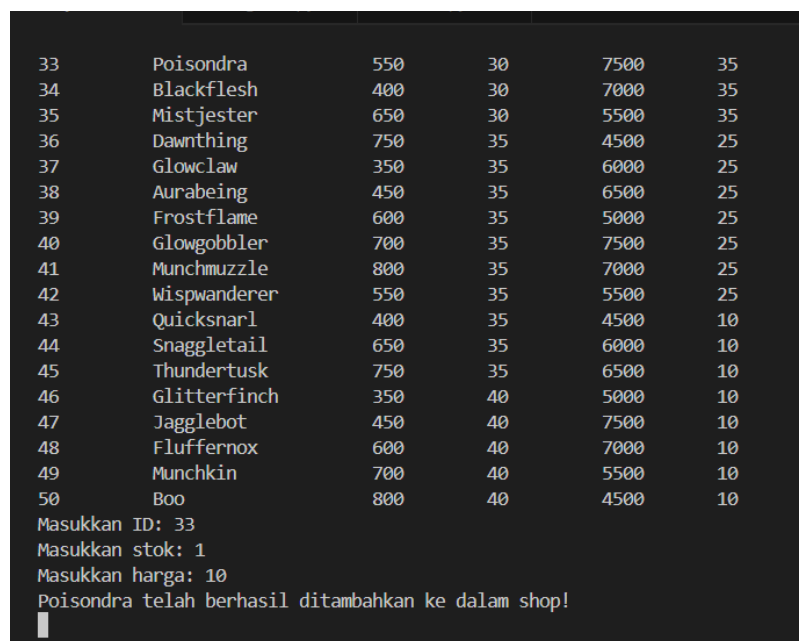
Jumlah OWCA Coins Anda sekarang: 50
No. Name           Type           Level   HP
1  Skibidi         Zeze           1       2321/6500
Upgrade Price
Level 1 -> Level 2: 300 OC
Level 2 -> Level 3: 500 OC
Level 3 -> Level 4: 700 OC
Level 4 -> Level 5: 900 OC
Pilih monster atau x untuk keluar:1
skibidi akan diupgrade ke level 2.
Harga untuk melakukan upgrade skibidi adalah 300
Lanjutkan upgrade? (Y/N) █
```

F12-Shop Management

Gambar 1.25. Shop Management



Gambar 1.24. Menambah item



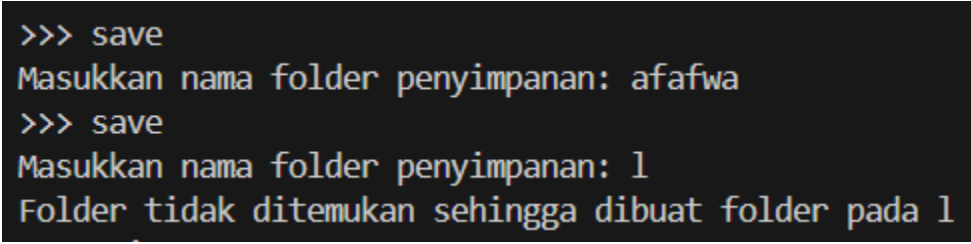
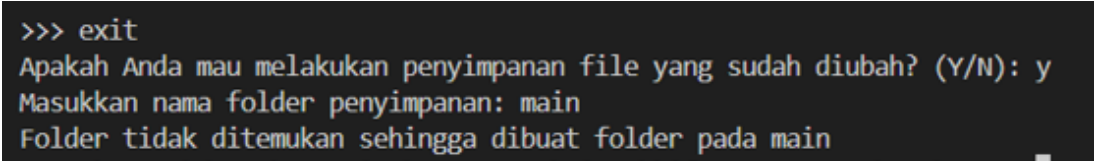
F13-Monster Management

Gambar 1.26. Monster Management



F14-Load

-

F15-Save	<p>Gambar 1.27. Save</p>  <pre> >>> save Masukkan nama folder penyimpanan: afafwa >>> save Masukkan nama folder penyimpanan: l Folder tidak ditemukan sehingga dibuat folder pada l </pre>
F16-Exit	<p>Gambar 1.28. Exit</p>  <pre> >>> exit Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (Y/N): y Masukkan nama folder penyimpanan: main Folder tidak ditemukan sehingga dibuat folder pada main </pre>

Tabel 4. Lampiran Hasil Pengujian Program

Lampiran Form Asistensi

Gambar 2.1 : Form Asistensi 1

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

Nomor Asistensi : 1
No. Kelompok/Kelas : K10/G
Tanggal asistensi : 3 Mei 2024

Anggota kelompok	NIM / Nama (Hanya yang Hadir)
1	19623099/Daniel Pedrosa Wu
2	19623039/Devon Wiraditya Tanumihardja
3	19623049/Darryl Rayhananta Adenan
4	19623189/Muhammad Aymar Barkhaya
5	
6	

Asisten pembimbing	NIM / Nama
	13521019/Ditra Rizqa Amadia

Catatan Asistensi: -

Rangkuman Diskusi

Pada asistensi pertama, kelompok K10-G mengajukan berbagai pertanyaan teknis, terutama mengenai hal-hal yang diperbolehkan dan yang tidak diperbolehkan dalam pengerjaan program tugas besar. Mereka juga menanyakan secara rinci mengenai spesifikasi, baik untuk fungsi utama maupun fungsi bonus. Penggunaan variabel global diklarifikasikan oleh asisten karena terdapat konflik informasi pada spreadsheet QnA, yang menyebabkan kebingungan dalam kelompok.

Asisten menjelaskan bahwa fungsi tambah dan hapus pada fungsi manajemen tidak langsung memodifikasi file CSV. Sebelumnya, kelompok berasumsi bahwa perubahan langsung terjadi pada file CSV. Namun, yang sebenarnya dilakukan adalah menyimpan data dalam suatu matriks terlebih dahulu. Kemudian, saat fungsi simpan dijalankan, data tersebut dikonversi dari matriks ke file CSV.

Terkait dekorasi antarmuka, kelompok diberikan kebebasan untuk mengimpor modul warna (color). Ini memberikan fleksibilitas dalam mendesain tampilan antarmuka yang lebih menarik dan user-friendly. Asisten juga menjelaskan bahwa fungsi typing digunakan untuk implementasi type-safe. Type-safe memastikan agar operasi yang dilakukan sesuai dengan jenis data yang telah ditentukan, sehingga mengurangi risiko kesalahan tipe data selama eksekusi program. Selain itu, asisten menjelaskan bahwa penggunaan fungsi array yang diperbolehkan hanyalah append, sehingga penggunaan fungsi pop tidak diperbolehkan.

Diskusi terakhir menyentuh cara kerja kelompok sejauh ini yang masih menggunakan metode oper-oper file. Asisten memberikan klarifikasi dan saran agar proses kerja kelompok menjadi lebih efisien dan terstruktur. Asisten merekomendasikan penggunaan GitHub untuk mempermudah kolaborasi. Dengan memanfaatkan fitur-fitur Git seperti commit, push, pull, branch, dan merge, kelompok dapat bekerja secara paralel dan terkoordinasi. Ini juga membantu dalam meninjau versi riwayat perubahan, sehingga

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

setiap anggota tim dapat dengan mudah melacak dan memahami kontribusi yang dilakukan oleh anggota lain.

Tindak Lanjut

Asisten menyarankan agar segera memanfaatkan GitHub untuk mempermudah pekerjaan. Meskipun GitHub wajib digunakan sebagai sarana pengumpulan, jika digunakan juga selama proses pengerjaan, pekerjaan akan menjadi lebih mudah berkat fitur-fitur yang ada pada Git, seperti push, pull, commit, branch, dan merge. Fitur-fitur ini memungkinkan anggota tim untuk bekerja secara paralel dan terkoordinasi, menghindari konflik dan redundansi pekerjaan. Penggunaan GitHub memudahkan kolaborasi antara banyak kontributor, memastikan setiap perubahan terdokumentasi dengan baik, dan memberikan riwayat versi yang jelas. Dengan demikian, pembuatan program menjadi lebih teratur, efisien, dan transparan.

Dokumentasi



Gambar 2.2 : Form Asistensi 2

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

Nomor Asistensi : 2
 No. Kelompok/Kelas : K10/G
 Tanggal asistensi : 10 Mei 2024

Anggota kelompok	NIM/ Nama (Hanya yang Hadir)	
	1	19623099/Daniel Pedrosa Wu
	2	19623049/Darryl Rayhananta Adenan
	3	19623189/Muhammad Aymar Barkhaya
	4	16523199/Fathur Rahman Arakhal
	5	
Asisten pembimbing	NIM/ Nama	
	13521019/Ditra Rizqa Amadia	

Catatan Asistensi :

Rangkuman Diskusi
<p>Asisten menanyakan perkembangan terkait program yang sedang dikerjakan oleh kelompok K-10G. Saat ini, kemajuan pengerjaan program tidak terlalu cepat, namun kelompok masih memiliki waktu yang cukup banyak untuk menyelesaikannya. Salah satu hal yang diklarifikasi oleh asisten adalah mengenai penggunaan semantic Git. Asisten menjelaskan konvensi semantic yang umum digunakan dalam Git. Contohnya, penamaan commit yang konsisten seperti feat untuk fitur baru, fix untuk perbaikan bug, refactor untuk penamaan variabel, dan docs untuk perubahan dokumentasi. Asisten juga memberikan contoh template file markdown README untuk diimplementasikan di repository GitHub kelompok. Template ini mencakup bagian-bagian penting seperti deskripsi proyek, cara instalasi, penggunaan, dan kontribusi.</p> <p>Mengenai penyimpanan data, asisten menjelaskan bahwa data dari file CSV diperbolehkan disimpan sebagai string dan dapat dikonversi ke integer saat diperlukan untuk operasi tertentu. Ini dilakukan agar kelompok tidak kesulitan dalam handling filenya. Asisten juga memberikan panduan mengenai penulisan notasi algoritmik dalam laporan Tugas Besar. Notasi algoritmik harus mencakup semua langkah yang dilakukan oleh program, ditulis secara sistematis dan mudah dipahami. Kamus data diisi oleh variabel-variabel beserta tipe-tipe datanya, baik yang built-in maupun buatan.</p> <p>Lebih lanjut, asisten menekankan pentingnya integrasi semua bagian program ke dalam file main.py secepat mungkin. Integrasi yang dilakukan dekat dengan batas waktu pengumpulan dapat mengakibatkan banyak error dan masalah, yang pada akhirnya dapat mempengaruhi kualitas hasil akhir program. Oleh karena itu, disarankan untuk segera menggabungkan semua modul dan fungsi yang telah dibuat, lalu melakukan pengujian secara menyeluruh untuk memastikan program berjalan dengan baik.</p>

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

Tindak Lanjut

Dari hasil asistensi, kelompok K10-G segera mengimplementasikan program main.py sebagai driver utama dari program mereka. Langkah ini dilakukan untuk mengintegrasikan semua modul dan fungsi yang telah dikembangkan oleh anggota kelompok. Dengan adanya main.py, program menjadi lebih terstruktur dan memudahkan proses pengujian serta debugging. Selain itu, kelompok K10-G meningkatkan praktik penggunaan GitHub mereka dengan menerapkan commit message yang lebih terstruktur dan konsisten. Implementasi type-safe juga mulai dilakukan untuk memastikan bahwa semua operasi dan fungsi yang dilakukan sesuai dengan jenis data yang telah ditentukan. Dengan menggunakan anotasi tipe data, kesalahan tipe data yang mungkin terjadi selama eksekusi program dapat diminimalisir.

Dokumentasi



Daftar Referensi

Sumber Online:

1. Wikipedia. [Linear Congruential Generator](#). Diakses pada 1 Mei 2024.
2. Python Documentation. [argparse - Parsing command-line arguments](#). Diakses pada 1 Mei 2024.
3. freeCodeCamp. [How to Use Git: Best Practices for Beginners](#). Diakses pada 1 Mei 2024.
4. Josh Buchea. [Caesar cipher in Python](#). Diakses pada 1 Mei 2024.
5. Python Documentation. [typing - Python Typing](#). Diakses pada 1 Mei 2024.

PPT IF1210.2022:

1. Tim Pengajar IF1210.2022. *Notasi Algoritmik dan Padanan dalam Bahasa Python 3 - Analisis Kasus, Pengulangan, Subprogram*.
2. Tim Pengajar IF1210.2022. *Skema Standar Bag.1) : Skema Validasi, Pengulangan, Pemrosesan Sekuensial*.
3. Tim Pengajar IF1210.2022. *Skema Standar (Bag.2) : Skema Pemrosesan Sekuensial pada Array*.
4. Tim Pengajar IF1210.2022. *Skema Standar (Bag.3) : Skema Sorting pada Array*.
5. Tim Pengajar IF1210.2022. *Skema Standar (Bag.4) : Pemrosesan File Sekuensial*.
6. Tim Pengajar IF1210.2022. *Skema Standar (Bag.4) : Implementasi File Sekuensial di Python*.