

Spesifikasi Tugas Besar

IF1210 Dasar Pemrograman 2024

Tim Asisten Lab Pemrograman 2021

Versi : **5**

Tgl. Revisi Terakhir : **16 Mei 2024**

Deadline : **20 Mei 2024, 12.10 WIB**

Revisi 1: 26/04/2024 – Ketentuan paradigma pemrograman yang digunakan

Revisi 2: 27/04/2024 – Menampilkan O.W.C.A Coin pada Inventory dan *type checking*

Revisi 3: 28/04/2024 – Pengunduran deadline Asistensi 1 & 2

Revisi 4: 30/04/2024

- Pemberian penjelasan mengenai mekanisme Battle
- Perbaikan penjelasan spesifikasi pada Monster
- Penambahan video tutorial menggunakan *git*

Revisi 5: 02/05/2024 – Penambahan penjelasan mengenai fungsi Load dan Save

Revisi 6: 06/05/2024 – Penyesuaian B05 terkait implementasi F04

Revisi 7: 16/05/2024 – Pengunduran Deadline Tugas Besar

Daftar Isi

Daftar Isi	2
Deskripsi Persoalan	4
Spesifikasi Program	6
Spesifikasi Program Utama	7
F00 – Random Number Generator	7
F01 – Register	7
F02 – Login	8
F03 – Logout	8
F04 – Menu & Help	9
F05 – Monster	10
F06 – Potion	11
F07 – Inventory	11
F08 – Battle	12
F09 – Arena	17
F10 – Shop & Currency	19
F11 – Laboratory	22
F12 – Shop Management	23
F13 – Monster Management	26
F14 – Load	28
F15 – Save	29
F16 – Exit	30
Spesifikasi Bonus	31
B01 – Git Best Practice	31
B02 – Typing	33
B03 – Monster Ball	34
B04 – JACKPOT!	36
B05 – Peta Kota Danville	38
BXX – Kreativitas	44
Struktur Data File	46
File User (user.csv)	47
File Monster (monster.csv)	47
File Inventory – Item (item_inventory.csv)	48
File Inventory – Monster (monster_inventory.csv)	49

File Shop - Item (item_shop.csv)	49
File Shop - Monster (monster_shop.csv)	50
Ketentuan dan Batasan	51
A. Ketentuan Umum	51
B. Batasan Penggerjaan	51
C. Jadwal Pelaksanaan Kegiatan	52
D. Penggerjaan Tugas Besar	52
Pengumpulan dan Deliverables	54
Referensi	55
Extras	56

Deskripsi Persoalan



"Purry si Platypus atau yang biasa disebut sebagai Agent P membutuhkan bantuan agent lainnya untuk menaklukan monster milik Dr. Asep Spakbor"

Di pinggiran kota Danville, Purry si Platypus, yang juga dikenal sebagai Agent P, sedang menghadapi tantangan yang besar. Dr. Asep Spakbor, seorang ilmuwan jahat, telah menciptakan monster-monster mengerikan yang mengancam keamanan kota Danville. Tugas Purry adalah untuk menghentikan ancaman ini, tetapi monster terbaru Dr. Asep Spakbor terlalu kuat bahkan bagi Purry.



Dr. Asep Spakbor sedang menciptakan monster

Purry menyadari bahwa dia membutuhkan bantuan dari agen-agen lainnya untuk mengalahkan monster itu. Dia bergegas ke markas rahasia O.W.C.A. (Organisasi Warga Cool Abiez) di mana dia bertemu dengan agen-agen lainnya, yaitu kalian.



O.W.C.A

Purry mengatakan bahwa untuk mengalahkan Dr. Asep Spakbor, kalian harus bekerja sama untuk mengalahkan monster-monster kuat. Kalian harus merencanakan strategi yang matang agar dapat menyelesaikan misi ini dengan baik. Setelah mendengarkan perkataan Purry, kalian bersiap-siap untuk mengeksekusi rencana dengan teliti.

Purry pun meminta bantuan kalian untuk mencari dan melatih monster-monster sendiri untuk digunakan dalam pertempuran melawan Dr. Asep Spakbor. Kalian merasa tertantang oleh misi ini, namun kalian juga yakin bahwa dengan kerja tim dan tekad yang kuat, pasti bisa berhasil.

Kalian memutuskan untuk memulai misi pencarian monster di hutan terpencil yang diyakini menjadi tempat tinggal bagi banyak jenis monster. Petualangan kalian di hutan yang gelap dan misterius ini akan menguji keberanian dan ketangkasan kalian. Kalian harus siap menghadapi segala tantangan yang mungkin muncul di perjalanan ini demi keselamatan kota Danville.

Spesifikasi Program

Terdapat kebutuhan fungsional wajib yang diperlukan oleh kalian sebagai Agent, seperti yang tertera di bawah. Tampilan atau interface dari sistem dibebaskan, silahkan berkreasi; output tidak harus persis seperti contoh, yang penting spesifikasi terpenuhi. Kreativitas interface akan menjadi pertimbangan penilaian. Perlu diperhatikan bahwa yang menjadi penekanan bukan hanya pada **alur jalannya program**, tetapi juga pada **validasi aksi** yang dilakukan.

Banyaknya fungsionalitas yang wajib diimplementasikan bukan berarti program yang akan dibuat juga akan panjang dan kompleks. Dengan pembuatan program yang **modular** dengan **fungsi/prosedur yang jelas**, program dapat dibuat secara relatif lebih **singkat dan sederhana**. Silakan pahami spesifikasi Tugas Besar ini dengan baik sebelum melakukan pengeraaan. Ingat, ini merupakan Tugas Besar berkelompok, sehingga aturlah sedemikian mungkin agar tugas ini tidak dibebankan kepada beberapa orang saja.

WARNING: CICIL PENERJAAN DARI JAUH-JAUH HARI DAN KERJAKAN DENGAN KONSISTEN. JIKA MENGERJAKAN DEKAT DENGAN DEADLINE, SANGAT BERESIKO TUGAS BESARINI TIDAK SELESAI DENGAN BAIK!

Spesifikasi Program Utama

F00 – Random Number Generator

Random Number Generator (RNG) akan digunakan untuk menghasilkan bilangan acak dari sebuah Range angka. Fitur ini akan digunakan untuk berbagai fitur, terutama pada Monster dan Battle. Implementasi RNG dilarang menggunakan library random python dan **harus** menggunakan algoritma [Linear Congruential Generator \(LCG\)](#).

F01 – Register

Sebelum memasuki sistem, agen harus mendaftarkan akunnya terlebih dahulu. Seorang pengguna baru harus memasukan *username* dan *password* yang akan digunakan. Pengguna yang mendaftar akan otomatis mendapatkan role *agent* dan **tanpa** O.W.C.A. Coin (atau berjumlah 0). Jika belum ada akun yang terdaftar dengan *username* yang diberikan, maka pengguna akan diharuskan memilih salah satu monster dari [File Monster \(monster.csv\)](#) sebagai monster awal. Username hanya dapat mengandung alfabet A-Za-z, underscore "_", strip "-", dan angka 0-9.

```
# DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja
>>> REGISTER
Masukan username: bangkitganteng
Masukan password: inipassword124

Silahkan pilih salah satu monster sebagai monster awalmu.
1. Charizard
2. Bulbasaur
3. Aspal

Monster pilihanmu: 1

Selamat datang Agent bangkitganteng. Mari kita mengalahkan Dr. Asep Spakbor dengan Charizard!
```

```
>>> REGISTER
Masukan username: Purry
Masukan password: inipassword124

Username Purry sudah terpakai, silahkan gunakan username lain!
```

```
>>> REGISTER
Masukan username: b@ngkitganteng
Masukan password: inipassword124
```

Username hanya boleh berisi alfabet, angka, underscore, dan strip!

>>> REGISTER

Register gagal!

Anda telah login dengan username Purry, silahkan lakukan “LOGOUT” sebelum melakukan register.

F02 – Login

Agent dan Admin bisa melakukan *login* ke dalam sistem. Terdapat 3 jenis kesalahan yang mungkin terjadi ketika memanggil *command* ini: username tidak terdaftar, password salah, dan pemanggilan *command* ketika sudah *logged in*.

DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja

>>> LOGIN

Username: Purry

Password: pl@tYPu5

Selamat datang, Agent Purry!

Masukkan command “help” untuk daftar command yang dapat kamu panggil.

>>> LOGIN

Username: Purri

Password: pl@tYPu5

Username tidak terdaftar!

>>> LOGIN

Username: Purry

Password: platypus

Password salah!

>>> LOGIN

Login gagal!

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

F03 – Logout

Setiap pengguna yang sedang *login* dapat melakukan prosedur ini untuk keluar dari akun yang sedang dipakai. Setelah melakukan *logout*, pengguna kehilangan akses dari akun sebelumnya dan dapat melakukan *login* lagi menggunakan akun yang berbeda. Prosedur ini hanya dapat dipanggil ketika sedang ada akun yang *logged in*.

DISCLAIMER: Tampilan/ interface dibebaskan, berikut hanya contoh saja

>>> LOGOUT

```
# Keluar dari akun
>>>
>>> LOGOUT
Logout gagal!
Anda belum login, silahkan login terlebih dahulu sebelum melakukan logout
```

F04 – Menu & Help

Setelah login, Agent dan Admin bingung apa yang sebaiknya dilakukan dan membutuhkan bantuan dari The Mighty God. The Mighty God menjawab permintaan mereka dengan menurunkan sebuah *command* yang dapat membantu perjalanan mereka, yaitu **HELP**. *Command* **HELP** adalah *command* yang sangat sakti hingga dapat mengingatkan pengguna untuk menuntun pengguna dan mengingatkan untuk melakukan validasi input dalam bentuk *footnote*. Deskripsi dari penggunaan *command* **HELP** dapat dilihat pada tabel dibawah.

Sebelum *login*, pengguna juga bingung apa yang dapat dilakukan. The Mighty God yang melihat kejadian tersebut juga turun tangan dan memberikan *command* **HELP**. Sebelum *login*, pengguna perlu melakukan *login* atau *register* terlebih dahulu.

```
# sebelum login
>>> HELP
=====
Kamu belum login sebagai role apapun. Silahkan login terlebih dahulu.

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

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

```
# setelah login sebagai Agent
>>> HELP
=====
Halo Agent Purry. Kamu memanggil command HELP. Kamu memilih jalan yang benar, semoga kamu tidak sesat kemudian. Berikut adalah hal-hal yang dapat kamu lakukan sekarang:

1. Logout: Keluar dari akun yang sedang digunakan
2. Inventory: Melihat owca-dex yang dimiliki oleh Agent
3.
# ...dan seterusnya
```

Footnote:

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

>>>

```
# setelah login sebagai Admin
```

```
>>> HELP
```

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

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

1. Logout: Keluar dari akun yang sedang digunakan
2. Shop: Melakukan manajemen pada SHOP sebagai tempat jual beli peralatan Agent
- 3.

```
# ... dan seterusnya
```

Footnote:

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

>>>

F05 – Monster

Dr. Agus Heisenberg sudah meneliti monster-monster yang ada dan menyatukannya pada suatu database bernama “owca-dex”. Berikut adalah atribut-atribut yang dicatat Dr. Agus pada database “owca-dex”. Contoh format csv-nya dapat dilihat pada [File Monster \(monster.csv\)](#).

- **Type:** Tipe atau nama dari monster yang bersifat unik.
- **ATK Power:** Kekuatan serangan setiap monster. Ketika monster melakukan serangan, nilai kekuatan serangannya diambil secara acak dengan rentang $\pm 30\%$ dari ATK Power. Contoh untuk ATK Power = 500, rentang nilai serangannya adalah 350 – 850.
- **DEF Power:** Kekuatan pertahanan setiap monster dari serangan. DEF digunakan sebagai faktor pengali untuk mengurangi serangan musuh. Rentang nilainya adalah 0 – 50. Contoh untuk DEF Power = 50, maka monster dapat menurunkan serangan musuh sebesar 50%.
- **HP:** Jumlah darah yang dimiliki oleh monster

Selain atribut tersebut, setiap monster juga memiliki data **Level** sendiri. Data level disimpan pada [F07 – Inventory](#), sehingga dalam “owca-dex”, hanya informasi dasar saja yang disimpan. Rentang level adalah 1 – 5.

- **Level 1:** Monster menggunakan Base Attribute

- **Level 2 - 5:** Ketika sedang Battle, Base Attribute monster ditambah dengan Base Attribute dikalikan dengan $((\text{Level} - 1) * 10)\%$. Ketika monster naik level, Base Attribute tetap disimpan, kalkulasi atribut sesuai level dilakukan saat Battle.

F06 - Potion

Dengan bantuan Dr. Agus Heisenberg, agent dapat memiliki dan menggunakan *potion* yang dapat membantu mereka dalam *battle*. Berikut adalah jenis *potion* yang tersedia beserta efeknya.

1. Strength Potion: Meningkatkan ATK Power sebanyak 5% dari ATK Power.
2. Resilience Potion: Meningkatkan DEF Power sebanyak 5% dari DEF Power.
3. Healing Potion: Mengisi darah sebanyak 25% dari Base HP. Pastikan HP tidak melebihi dari maksimal HP.

ATK Power, DEF Power, dan HP mengikuti level Monster saat ini, lakukanlah perhitungan terlebih dahulu.

Masing-masing *potion* hanya dapat digunakan sekali dalam 1 battle dan efeknya hanya berlaku hingga battle itu selesai.

F07 - Inventory

Akses: Agent

Setiap agent seperti agent Purry, dilengkapi dengan sebuah teknologi inovatif "Invento-Stock". Teknologi ini adalah sebuah sistem penyimpanan inventory yang dinamis. Isi dari invento-Shock dapat berupa Potion dan Monster yang dapat disimpan dalam file *item_inventory.csv* dan *monster_inventory.csv*. Setiap item yang disimpan pada inventory dapat dimunculkan detailsnya. Untuk potion, akan ditampilkan kegunaan (type) dan kuantitasnya. Untuk monster, akan ditampilkan atribut-atribut yang dimiliki oleh monster tersebut (type/name, atk power, def power, hp, level). Tampilkan juga O.W.C.A. coin yang dimiliki user saat ini.

Contoh format csv-nya dapat dilihat pada :

[File Inventory - Item \(item_inventory.csv\)](#)

[File Inventory - Monster \(monster_inventory.csv\)](#)

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

```
===== INVENTORY LIST (User ID: 1) =====
Jumlah O.W.C.A. Coin-mu sekarang 900.
1. Monster      (Name: Zuko, Lvl: 2, HP: 100)
2. Monster      (Name: Pikachu, Lvl: 5, HP: 200)
3. Potion       (Type: Heal, Qty: 10)
4. Potion       (Type: ATK, Qty: 7)
5. Monster Ball (Qty: 5)
```

DISCLAIMER: Tampilan pada inventory list wajib menampilkan komponen-komponen seperti berikut (Ini bukan detail item! Hanya tampilan singkat/ringkasan saja):

1. User ID
2. Monster: Name, Lvl, HP
3. Potion: Type, Qty
4. Monsterball: Qty

Ketikkan id untuk menampilkan detail item:

```
>>> 1
Monster
Name      : Zuko
ATK Power : 20
DEF Power : 20
HP        : 100
Level     : 2
```

DISCLAIMER: Mekanisme untuk menampilkan detail item dibebaskan:

1. Pengguna bisa meminta detail item hingga mengetik command tertentu untuk berhenti, atau
2. Pengguna mengetik ulang command “INVENTORY” baru kemudian mengambil detail item lagi.

Ketikkan id untuk menampilkan detail item:

```
>>> 3
Potion
Type      : Heal
Quantity  : 10
```

Ketikkan id untuk menampilkan detail item:

```
>>> 5
Monster Ball
Quantity  : 10
```

Buatlah mekanisme untuk keluar dari Inventory

Ketikkan id untuk menampilkan detail item:

```
>>> KELUAR
# ...Kembali ke menu utama
```

Jika mengerjakan bagian [B03 - Monster Ball](#), ketika monster ditaruh di MonsterBall dan penangkapan berhasil, maka monster tersebut akan didaftarkan pada *inventory* juga.

F08 – Battle

Akses: Agent

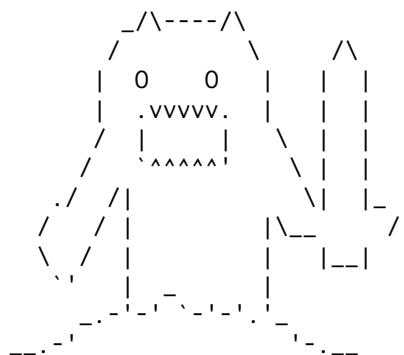
Agent dapat memilih untuk bertarung melawan monster secara random. Prosedur ini bersifat *turn-based 1v1 game*. Setiap pertempuran akan terdiri dari serangkaian putaran

dimana Agent dan monster bergantian melakukan tindakan. Berikut adalah alur pertarungan.

1. Muncul monster (musuh) secara random (RNG) database Monster; Agent memilih monster (agent) yang ingin dipertarungkan
2. Setiap putaran Agent memiliki pilihan untuk "Attack", "Use Potion", atau "Quit"; Monster (musuh) hanya bisa "Attack"
3. Kondisi kemenangan adalah saat nyawa monster (musuh) habis; Agent mendapatkan **OC** (OC yang diterima random (RNG), misal 5-30)
4. Kondisi kekalahan adalah saat nyawa monster (agent) habis; Agent keluar dari pertempuran, nyawa monster yang dipertarungkan kembali penuh (tidak hilang dari inventory)

Penjelasan mengenai mekanisme Battle mohon ditampilkan dengan format bebas.

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



```
RAWRRR, Monster Zuko telah muncul !!!
```

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

```
===== MONSTER LIST =====
```

1. Chacha
2. Pikachuow
3. Zeze

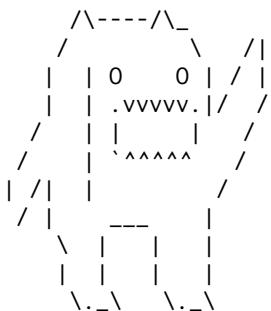
```
# Apabila monster yang dipilih tidak ada
Pilih monster untuk bertarung: 4
```

```
Pilihan nomor tidak tersedia!
```

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

```
# Apabila monster yang dipilih ada
```

```
Pilih monster untuk bertarung: 2
```



```
RAWRRR, Agent X mengeluarkan monster Pikachuow !!!
```

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

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

- 1. Attack
- 2. Use Potion
- 3. Quit

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

```
Anda berhasil kabur dari BATTLE!
```

```
>>>
```

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

```
Anda tidak memiliki Potion dalam inventory!
```

```
Pilih perintah: _
```

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

```
# Mengambil Potion dari inventory
```

```
===== POTION LIST =====
```

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

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

```
# Apabila memilih Strength Potion
Pilih perintah: 1
```

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

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

```
...
```

```
# Apabila memilih Resilience Potion
```

```
Pilih perintah: 2
```

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

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

```
...
```

```
# Apabila memilih Healing Potion
```

```
Pilih perintah: 3
```

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

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

```
...
```

```
# Apabila memilih Potion yang telah diminum sebelumnya
```

```
Pilih perintah: 1
```

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

```
===== POTION LIST =====
```

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

```
# Apabila memilih Potion yang tidak dimiliki
```

```
===== POTION LIST =====
```

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

```
Pilih perintah: 1
```

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

```
===== POTION LIST =====
```

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

```
# Apabila memilih Attack
```

```
Pilih perintah: 1
```

SCHWINKKK, Pikachu menyerang Zuko !!!

Name : Zuko

ATK Power : 20

DEF Power : 20

```

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

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

SCHWINNKKK, Zuko menyerang Pikachu !!!

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

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

Pilih perintah: 1

SCHWINNKKK, Pikachu menyerang Zuko !!!

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

Selamat, Anda berhasil mengalahkan monster Zuko !!!

Total OC yang diperoleh: 30

>>>

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

SCHWINNKKK, Zuko menyerang Pikachu !!!

Name      : Pikachu
ATK Power : 25
DEF Power : 5
HP        : 0
Level     : 1
# Misal: ATT Results: 19.2, HP terakhir 20, hasil akhir: FLOOR(20 - 19.2) = 0
# Penjelasan battle mohon untuk ditampilkan, format bebas

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

>>>

```

F09 – Arena

Akses: Agent

Purry menyadari bahwa monster-monster yang ada harus sering dilatih secara intensif demi bisa mengalahkan Dr. Asep Spakbor. Maka dari itu, O.W.C.A memutuskan untuk membuat Arena untuk meningkatkan kemampuan agen dan para monster.

Terdapat beberapa ketentuan dalam arena ini, yaitu:

1. Terdapat 5 stage dalam sekali sesi latihan di Arena. Masing-masing stage merepresentasikan level monster yang dilawan. Artinya, agent akan melawan monster level 1 di stage 1, monster level 2 di stage 2, dst.
2. Monster yang muncul pada tiap stage terpilih secara acak dari monster-monster yang terdapat pada database. Monster dengan jenis yang sama dapat muncul lebih dari sekali pada satu sesi latihan sehingga jumlah monster pada database dapat kurang dari 5.
3. Mekanisme pertarungan pada tiap stage sama seperti battle biasa. Pada setiap akhir stage, health dari monster agent akan dipulihkan sepenuhnya.
4. Apabila agent kalah pada suatu stage (monster miliknya mati), maka sesi latihan berakhir dan game over. Agent dapat keluar dari pertarungan dan akan mendapatkan status game over juga.
5. Setelah sesi latihan berakhir yaitu agent berhasil memenangkan kelima stage atau terjadi game over, hasil stats sesi latihan akan ditampilkan pada scoreboard yang **setidaknya** terdiri atas informasi total hadiah yang diterima, berhasil menang hingga stage berapa, total damage yang diberikan, dan total damage yang diterima. Informasi lain dapat ditambahkan (**opsional**).
6. Jumlah hadiah pada setiap stage dibebaskan (hardcode pada program, tidak perlu disimpan dalam konfigurasi atau data store). Misalnya:
 - Stage 1: 30 OC
 - Stage 2: 50 OC
 - Stage 3: 100 OC
 - dst.

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

Selamat datang di Arena!!

```
===== MONSTER LIST =====
1. Chacha
```

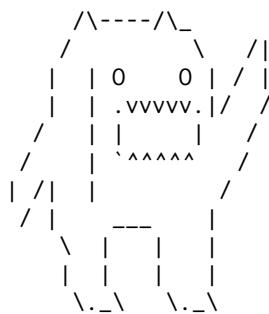
2. Pikachow
3. Zeze

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

Pilihan nomor tidak tersedia!

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

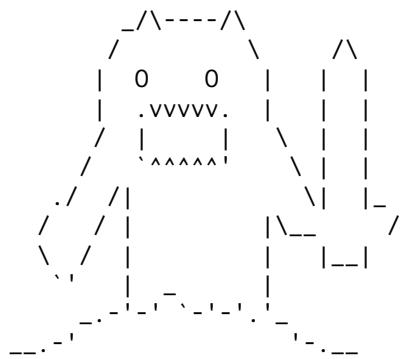
```
# Apabila monster yang dipilih ada  
Pilih monster untuk bertarung: 2
```



RAWRRR, Agent X mengeluarkan monster Pikachu !!!

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

```
# Memulai stage, ambil acak monster dari database. Level monster diset 1  
===== STAGE 1 =====
```



RAWRRR, Monster Zuko telah muncul !!!

```
# Lanjut seperti battle biasa
```

```
# Apabila stage berhasil dimenangkan dan ada stage berikutnya
```

Selamat, Anda berhasil mengalahkan monster Zuko !!!

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

```
Memulai stage berikutnya...
```

```
===== STAGE 2 =====
# Lanjut stage berikutnya
```

```
# Apabila agent kalah
```

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

```
GAME OVER! Sesi latihan berakhir pada stage 2!
```

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

```
Total hadiah : 30 OC
Jumlah stage : 1
Damage diberikan : 180
Damage diterima : 250
```

```
# Apabila stage 5 berhasil diselesaikan
```

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

```
STAGE CLEARED! Anda akan mendapatkan 200 OC pada sesi ini!
```

```
Selamat, Anda berhasil menyelesaikan seluruh stage Arena !!!
```

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

```
Total hadiah : 200 OC
Jumlah stage : 5
Damage diberikan : 780
Damage diterima : 400
```

```
# Apabila memilih Quit
```

```
Pilih perintah: 3
```

```
GAME OVER! Anda mengakhiri sesi latihan!
```

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

```
Total hadiah : 0 OC
Jumlah stage : 0
Damage diberikan : 100
Damage diterima : 50
```

F10 - Shop & Currency

Akses: Agent

Kepala O.W.C.A., Mr. Monogram, punya *bestie* bernama Mr. Yanto Williams. Mr. Yanto punya sebuah shop, yang menjual monster dan potion, yang terletak di lantai bawah tanah Superduper Mega Superstore. Shop inilah yang menjadi tempat Agent membeli monster dan potion.



Superduper Mega Superstore

Mr. Yanto bersedia memberikan *harga teman* dan memberikan akses manajemen toko pada Mr. Monogram, dengan syarat pembeliannya menggunakan OC (O.W.C.A. Coin).



O.W.C.A. Coin (OC)

OC dapat diperoleh Agent dengan cara memenangkan Battle, Arena, atau mini game [B04 – JACKPOT!](#). Sebagai catatan, saat akun Agent pertama kali dibuat, jumlah OC yang dimilikinya adalah 0.

OC ini akan digunakan untuk membeli potion atau monster di shop yang ada di Superduper Mega Superstore. Berikut ketentuannya.

1. Item-item pada shop (monster dan potion) disimpan di database. Detail terkait manajemen item di shop dari sisi Mr. Monogram dapat dilihat pada [F12 – Shop Management](#).
2. Agent dapat mengunjungi shop dengan command SHOP. Setelah itu, agent dapat memilih untuk melihat item-item, membeli item, dan keluar dari shop.
3. Saat Agent ingin melihat item-item, Agent dapat memilih untuk melihat potion atau monster.
 - o Jika Agent memilih monster, tampilkan id, type, ATK power, DEF power, HP, stok, dan harganya.
 - o Jika Agent memilih potion, tampilkan id, type, stok, dan harganya.

Pastikan data item yang dilihat oleh Agent hanyalah item yang dijual di shop (meskipun stoknya 0), bukan seluruh item yang ada di data monster.

4. Saat Agent ingin membeli item, tampilkan juga currency yang dimilikinya. Selanjutnya Agent dapat memasukkan tipe (monster/potion). Agent akan memasukkan id monster/potion yang ingin dibeli. Khusus untuk potion, Agent juga harus memasukkan jumlahnya.
5. Jika item berhasil dibeli, item akan masuk ke inventory pengguna. Jika tipe item yang berhasil dibeli adalah monster, monster tersebut berlevel 1 saat masuk ke inventory.

Lakukan validasi dalam proses pembelian monster dan/atau potion di shop. **Contoh** validasi yang perlu dilakukan yaitu jumlah OC (O.W.C.A Coin) yang dimiliki dan ketersediaan item di shop. Tambahkan validasi lain yang diperlukan. Khusus untuk pembelian monster, lakukan validasi juga bahwa **monster yang sudah dimiliki di inventory tidak dapat dibeli lagi**.

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

Irasshaimase! Selamat datang di SHOP!!

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

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

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

# Melihat potion
>>> Pilih aksi (lihat/beli/keluar): lihat
>>> Mau lihat apa? (monster/potion): potion
ID | Type      | Stok | Harga
1  | Strength Potion | 1    | 20
2  | Resilience Potion | 5    | 300

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

# Membeli monster
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 900.

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 1
Berhasil membeli item: Pokemon Air. Item sudah masuk ke inventory-mu!

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

# Membeli potion
>>> Pilih aksi (lihat/beli/keluar): beli
```

```

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

>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 2
>>> Masukkan jumlah: 2
Berhasil membeli item: 2 Potion of Defense. Item sudah masuk ke inventory-mu!

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

# Membeli: gagal dalam validasi
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 200.

>>> Mau beli apa? (monster/potion): potion
>>> Masukkan id potion: 2
>>> Masukkan jumlah: 1
OC-mu tidak cukup.

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

# Membeli monster: gagal dalam validasi inventory
>>> Pilih aksi (lihat/beli/keluar): beli
Jumlah O.W.C.A. Coin-mu sekarang 200.

>>> Mau beli apa? (monster/potion): monster
>>> Masukkan id monster: 1
Monster Pokemon Air sudah ada dalam inventory-mu! Pembelian dibatalkan.

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

>>> Pilih aksi (lihat/beli/keluar): keluar
Mr. Yanto bilang makasih, belanja lagi ya nanti :)

>>>

```

F11 - Laboratory

Akses: Agent

Agent dapat melakukan upgrade monster yang dimiliki di *inventory* dengan command LABORATORY. Berikut adalah mekanisme melakukan upgrade level monster:

1. Menampilkan pilihan monster pada *inventory* beserta level-nya.
2. Memilih monster yang ingin di-upgrade.
3. Jika level monster < 5, tampilkan jumlah OC (O.W.C.A. Coin) yang dibutuhkan untuk melakukan *upgrade*. Jika tidak, tampilkan bahwa monster telah memiliki level maksimum. Pengaturan jumlah OC (O.W.C.A. Coin) dibebaskan. Misalnya:
 - Level 1 -> Level 2 : 300 OC
 - Level 2 -> Level 3 : 500 OC
 - dst.
4. Konfirmasi apakah agent yakin melakukan *upgrade*.

5. Jika agent memilih untuk melanjutkan *upgrade* dan memiliki *currency* cukup, level monster pada *inventory* akan ter-*upgrade*. Jika tidak, tunjukkan pesan kegagalan.

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

```
Selamat datang di Lab Dokter Asep !!!
```

```
===== MONSTER LIST =====
```

1. Chacha (Level: 1)
2. Pikachu (Level: 2)
3. Zeze (Level: 5)

```
===== UPGRADE PRICE =====
```

1. Level 1 -> Level 2: 300 OC
2. Level 2 -> Level 3: 500 OC
3. Level 3 -> Level 4: 800 OC
4. Level 4 -> Level 5: 1000

```
>>> Pilih monster: 1
```

```
# Level yang dapat dipilih hanya 2 - 5 karena level Chaca saat ini 1
```

```
Chaca akan di-upgrade ke level 2.
```

```
Harga untuk melakukan upgrade Chaca adalah 300 OC.
```

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

```
Selamat, Chaca berhasil di-upgrade ke level 2 !
```

```
>>> Pilih monster: 3
```

```
Maaf, monster yang Anda pilih sudah memiliki level maksimum
```

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

```
# Level yang dapat dipilih hanya 2 - 5 karena level Chaca saat ini 1
```

```
Pikachu akan di-upgrade ke level 3.
```

```
Harga untuk melakukan upgrade Pikachu adalah 500 OC.
```

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

```
Selamat, Pikachu berhasil di-upgrade ke level 3 !
```

F12 – Shop Management

Akses: Admin

Sebagai kepala departemen Logistik O.W.C.A., Mr. Yanto Williams dapat mengatur barang-barang yang dijual untuk para Agent yang akan berguna dalam menjalankan misinya menaklukkan monster-monster. Berikut adalah mekanisme yang dapat dilakukan:

- I. Monster dan Potion yang ditampilkan di dalam Shop berasal dari Database

2. Terdapat Database sendiri untuk Item yang dijual di Shop, sehingga bisa dilakukan penambahan, perubahan, dan penghapusan Item yang dijual dari Database.
3. Terdapat harga dan jumlah dari tiap Item yang dapat diatur

Bentuk database dari Shop dapat dilihat pada:

[File Shop - Item \(item_shop.csv\)](#)

[File Shop - Monster \(monster_shop.csv\)](#)

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

Irasshaimase! Selamat datang kembali, Mr. Monogram!

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

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

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

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

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

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

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

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

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

```

3 | Healing Potion |

# Jangan lupa lakukan validasi input
>>> Masukkan id potion: 3
>>> Masukkan stok awal: 10
>>> Masukkan harga: 140
Healing Potion telah berhasil ditambahkan ke dalam shop!

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


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

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

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

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


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

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

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

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


# Menghapus monster dari Shop
>>> Pilih aksi (lihat/tambah/ubah/hapus/keluar): hapus
>>> Mau hapus apa? (monster/potion): monster
# Menampilkan seluruh monster yang ada di shop

```

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

```

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

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

```


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

```

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

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

```



```

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

>>>

```

F13 – Monster Management

Akses: Admin

Dr. Agus Heisenberg ingin agar monster yang ada di database dapat ditambahkan oleh role Admin. Oleh karena itu, ~~untuk menambah pekerjaan kalian~~ untuk menambah keseruan dalam game ini lakukan implementasi fitur monster management yang berfungsi untuk mengatur monster dalam database. Jangan lupa validasi input sesuai dengan ketentuan yang ada di database.

Bentuk database dari Monster dapat dilihat pada:

[File Monster \(monster.csv\)](#)

```

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

```

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

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

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

>>> Masukkan Type / Nama : Pokemon Udara

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

>>> Masukkan ATK Power : 20

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

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

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

# Cek Database
>>> Pilih Aksi : 1
ID | Type | ATK Power | DEF Power | HP
1 | Pokemon Air | 10 | 20 | 200
2 | Pokemon Api | 20 | 10 | 200
3 | Pokemon Tanah | 30 | 30 | 100
4 | Pokemon Udara | 20 | 50 | 200
```

```
# Opsi N pada konfirmasi penambahan monster
>>> Tambahkan Monster ke database (Y/N) : N
Monster gagal ditambahkan!
>>> Pilih Aksi : 1
ID | Type | ATK Power | DEF Power | HP
1 | Pokemon Air | 10 | 20 | 200
2 | Pokemon Api | 20 | 10 | 200
3 | Pokemon Tanah | 30 | 30 | 100
```

F14 - Load

Setelah mengambil waktu istirahat untuk membeli ayam geprek takol, agent purry ingin melanjutkan pekerjaan yang dia tinggalkan sebelum istirahat.

Prosedur ini akan memuat data yang sesuai dengan **struktur data eksternal**. Prosedur ini akan dijalankan sekali ketika pengguna memulai program. Cara menjalankan prosedur ini adalah dengan memberikan **nama folder** yang berisi file penyimpanan.

Untuk prosedur ini, asumsikan bahwa seluruh file penyimpanan dalam suatu folder :

1. Terjamin ada
2. Memiliki nama yang *fixed*
3. Memiliki format yang sesuai dengan struktur data eksternal. Akan, tetapi bentuk struktur data pada program dibebaskan, silakan definisikan sendiri struktur data terbaik yang dapat digunakan pada program.

Namun, prosedur tetap harus melakukan validasi apakah folder tersebut **ada atau tidak**.

Program hanya akan melakukan read pada file CSV apabila pengguna menjalankan fungsi **load**

Hint: Gunakan argparse untuk menerima argumen saat menjalankan file python.

[Tutorial argparse.](#)

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

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

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

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

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

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

F15 – Save

Prosedur ini digunakan untuk menyimpan data ke dalam file yang sesuai dengan **struktur data eksternal**. Dalam prosedur ini, program akan meminta **nama folder** yang akan digunakan sebagai tempat penyimpanan file. Berikut adalah ketentuannya:

1. Jika nama folder tidak ditemukan, program akan membuat folder sesuai dengan masukan dan memberikan pesan bahwa ia **membuat folder baru** dan **berhasil menyimpan data** di folder tersebut.
2. Jika nama folder ditemukan namun belum ada file pada folder tersebut maka program akan **menaruh file baru** pada folder tersebut
3. Jika nama folder dan file sudah ada, program akan mengganti file pada folder tersebut dengan yang lebih baru dan tidak perlu memberikan pesan tambahan. (*overwrite/replace*)

Note :

1. Tidak perlu meng-handle path ‘..’/ atau ‘~/’
2. Boleh menentukan folder *parent save*, misalnya untuk contoh di bawah “./data”
3. Manipulasi yang dilakukan terhadap data program akan disimpan terlebih dahulu dalam suatu variabel (in memory). Program hanya akan melakukan write terhadap file csv ketika pengguna menjalankan fungsi save.

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

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

Saving...

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

# nama folder ditemukan
>>> SAVE

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

Saving...

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

>>> SAVE

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

Saving...

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

F16 - Exit

Agent dan Admin tidak dapat terus melakukam perjalanan mereka. Mereka merasa lelah. Pada suatu saat, Agent memutuskan untuk menyerah saja. The Mighty God mendengarkan permintaan mereka dan menyediakan suatu *command* lagi, yaitu *exit*. Namun, The Mighty God masih memberikan kesempatan bagi para Agent dan Admin jika masih ingin melanjutkan petualangan mereka. Agent dan Admin diarahkan untuk melakukan *save* terlebih dahulu sebelum keluar dari petualangan mereka.

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

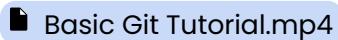
```
>>> EXIT
# Contoh input tidak valid
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) a
Apakah Anda mau melakukan penyimpanan file yang sudah diubah? (y/n) Y
# Menjalankan prosedur save (F15) dan keluar program
```

Spesifikasi Bonus

Karena beberapa Agent ~~merasa Tugas Besar ini terlalu gampang~~ menyukai tantangan, terdapat beberapa Spesifikasi Bonus yang dapat dikerjakan untuk menambah kreativitas dari program. Perlu dicatat untuk mengutamakan penggerjaan Spesifikasi Utama terlebih dahulu, karena nilai sempurna juga dapat diraih dengan program yang dapat memenuhi seluruh Spesifikasi Utama dengan sempurna. Kerjakan Spesifikasi Bonus jika merasa Spesifikasi Utama sudah selesai dengan baik serta masih memiliki waktu lebih.

B01 - Git Best Practice

Penggunaan Git secara spesifikasi wajib, hanyalah sebagai wadah pengumpulan. Satu kali upload file / commit yang berisi seluruh program, lalu membuat Release Tag sudah dianggap cukup. Pada bonus ini, kalian harus menerapkan penggunaan Git secara bersamaan dengan anggota kalian dan menerapkan *best practice* yang ada, yaitu melingkupi Commit Message dan Branching. Video tutorial Git dapat dilihat pada tautan berikut:



Commit Message

Commit message pada git digunakan sebagai log untuk menjelaskan perubahan yang terjadi pada repository. Commit message yang baik harus jelas, ringkas, dan informatif. Karakteristik commit message yang baik:

Concise: Singkat dan deskriptif

Relevance: Fokus terhadap kenapa perubahan tersebut dilakukan, bukan bagaimana

Clarity: Hindari deskripsi yang ambigu, pesan harus spesifik

Consistency: Pilih dan ikuti format standar (bisa dicari di internet) sehingga mudah untuk dibaca

Commit message yang baik

```
fix: fix property value display behaviour when editing and selected tool is cursor
d ditramadja committed 2 weeks ago

fix: set width and height rectangle
b bernarduswillson committed 2 weeks ago
```

Commit message yang jahat



fix sama gasssss ini maksudnya apaaa????

Contoh standar commit message:

<type>: <description>

type:

feat - Penambahan fitur baru

fix - Bug fix

docs - Perubahan pada dokumentasi (README, laporan, dsb)

style - Formatting, penambahan type safe, dsb

refactor - Refactor kode, ganti nama variable, dsb

test - Penambahan/perubahan script untuk testing

chore - Perubahan pada config file

style: add type safe for RNG algorithm

feat: add RNG algorithm

feat: add database for monsters

refactor: refactor command parser

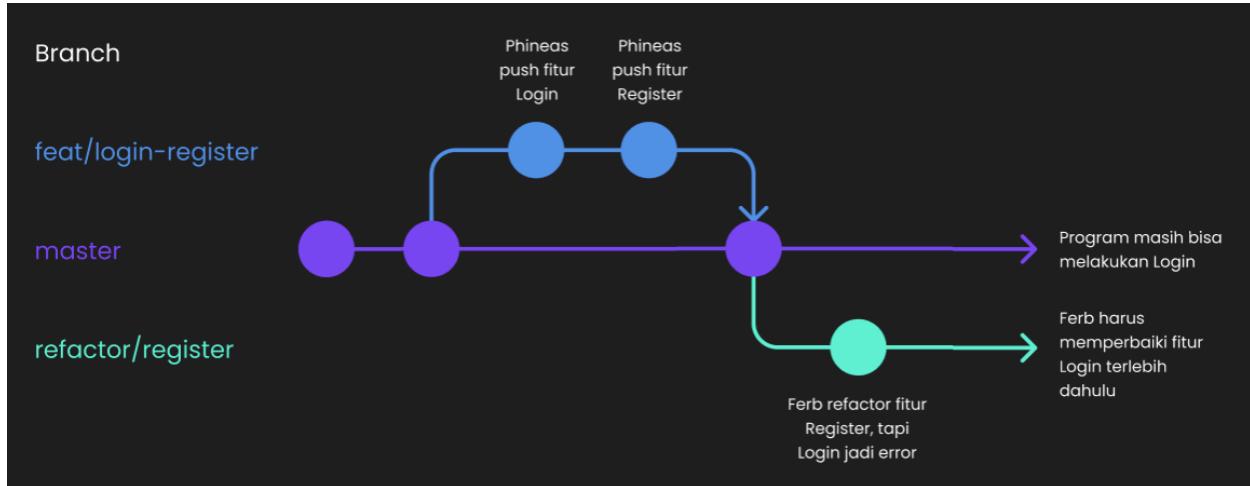
feat: add command parser

Branching

Branching adalah pembagian cabang dari repository kalian. Branching berfungsi untuk memisahkan suatu pekerjaan dengan pekerjaan lainnya (isolasi). Dengan branching, kalian bisa bekerja secara paralel pada branch yang berbeda untuk memitigasi *conflict*.

Misal Phineas dan Ferb tidak menerapkan branching dan hanya bekerja pada branch utama (branch **master** atau **main**). Phineas membuat fitur Login dan Register lalu di-push. Kemudian, Ferb melakukan refactor fitur Register lalu di-push. Walaupun fitur Register tetap bekerja, ternyata fitur Login jadi error. Pada akhirnya Phineas tidak bisa melakukan login dan tidak bisa melanjutkan pekerjaannya sebelum fitur Login diperbaiki.

Alternatifnya, Phineas dan Ferb menerapkan branching. Phineas telah membuat fitur Login dan Register di merge terpisah, lalu di-merge ke branch utama. Ferb membuat branch baru terlebih dahulu, sebelum mengubah fitur Register dan melakukan push. Saat fungsi Register telah di-refactor, ternyata fitur Login jadi error. Tetapi, karena Ferb bekerja pada branch terpisah, kode Ferb tidak berpengaruh pada kode Phineas. Ferb harus memperbaiki fitur Login terlebih dahulu, dalam waktu yang bersamaan, Phineas tetap dapat melakukan Login dan melanjutkan pekerjaannya.



B02 – Typing

Dalam O.W.C.A terdapat beberapa aturan penulisan program yang wajib dipatuhi oleh semua Agent. Salah satu aturan yang paling penting adalah menulis kode program yang *type safe*. Akan tetapi, Python bukan merupakan bahasa yang *type safe*, sehingga sebagai Programmer harus memperhatikan typing yang digunakan pada fungsi atau variabel yang digunakan. Maka dari itu, terapkanlah Typing yang meliputi pendefinisan tipe pada setiap atribut masukan fungsi dan tipe hasil return fungsi. Perhatikan bahwa penggunaan fitur typing pada python hanya dapat digunakan untuk python dengan versi di atas 3.9. Kalian diperbolehkan untuk melakukan *import typing* jika mengerjakan bonus ini dan jika menggunakan lambda atau fungsi sebagai parameter, diperbolehkan untuk melakukan *import Callable from collections.abc*. Diperboleh juga menggunakan *from typeguard import typechecked* untuk melakukan *type checking* saat *runtime*.

Berikut adalah contoh fungsi yang sudah menerapkan Typing:

```
type Vector = list[float]
```

```

@typechecked
def scale(scalar: float, vector: Vector) -> Vector:
    return [scalar * num for num in vector]

# passes type checking; a list of floats qualifies as a Vector.
new_vector = scale(2.0, [1.0, -4.2, 5.4])

```

B03 – Monster Ball



Monster Ball untuk menangkap monster

Monster Ball adalah fitur dalam permainan yang memungkinkan pemain untuk menangkap monster yang dihadapi saat pertarungan (**BATTLE**). Monster Ball **tidak** dapat digunakan saat pertarungan di **Arena**. Monster Ball ini disimpan dalam inventory dengan atribut *item_type* monsterball, sehingga jika mengerjakan bonus ini **WAJIB** menyesuaikan database (konfigurasi database dapat mengikuti Potion). Monster Ball ini dapat dibeli dari **Shop** dengan harga OC yang ditentukan.

Setiap menggunakan Monster Ball, terdapat peluang/ kemungkinan untuk mendapatkan monster yang ingin ditangkap. Hal ini bergantung pada level monster yang hendak ditangkap. Berikut peluangnya.

- Level 1 : 75%
- Level 2 : 50%
- Level 3 : 25%
- Level 4 : 10%
- Level 5 : 5%

Gunakan RNG untuk perhitungan ini.

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

- 1. Attack
- 2. Use Potion
- 3. Use Monster Ball
- 4. Quit

```
# Apabila memilih Use Monster Ball, namun tidak memiliki Monster Ball  
Pilih perintah: 3
```

```
Anda tidak memiliki Monster Ball dalam inventory!
```

```
Pilih perintah: _
```

```
# Apabila gagal mendapatkan monster
```

```
Pilih perintah: 3
```

```
Swoosshhhh, Anda mengeluarkan Monster Ball !!!
```

```
Yahhh, Anda belum berhasil mendapatkan monster Zuko !!!
```

```
Sisa Monster Ball Anda: 4
```

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

- 1. Attack
- 2. Use Potion
- 3. Use Monster Ball
- 4. Quit

```
Pilih perintah: _
```

```
# Apabila sudah memiliki monster di inventory
```

```
Pilih perintah: 3
```

```
Anda sudah memiliki monster Zuko dalam inventory!
```

```
Pilih perintah: _
```

```
# Apabila berhasil mendapatkan monster
```

```
Pilih perintah: 3
```

```
Swoosshhhh, Anda mengeluarkan Monster Ball !!!
```

```
Selamat, Anda berhasil mendapatkan monster Zuko !!!
```

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

```
Sisa Monster Ball Anda: 4
```

```
>>>
```

B04 – JACKPOT!



Agar Agent tidak bosan dan dapat memuaskan *passion-gacha* menguji keberuntungan mereka, dibuatlah permainan **JACKPOT!**. Mesin memiliki 3 “slot” yang akan menunjukkan *item* secara *random*. Setiap *item* akan memiliki *value* tertentu. Mesin **minimal** memiliki **5** *item* berbeda.

Pengguna perlu membayar sejumlah OC (O.W.C.A Coin) untuk memainkan permainan ini. Jumlah OC untuk bermain dibebaskan. Setelah membayar, sistem akan men-generate 3 *item* secara *random* (gunakan fungsi *random* yang telah diimplementasikan).

- Jika agent mendapatkan JACKPOT (3 *item* yang sama), maka agent akan mendapatkan monster random.
- Jika tidak, agent akan mendapatkan *sum* dari *value* masing-masing *item*.

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

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$ Apakah Anda siap untuk menguji keberuntungan?$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$ Menangkan Snorleks dengan 400 OC saja !!!$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

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

>> Mulai bermain (Y/N): Y
```

Anda Mendapatkan:

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ TOPI | TOPI | PEDANG \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

200 OC telah ditambahkan ke akun Anda! # Topi: 50 OC, Pedang: 100 OC. Total OC: 50 x 2 + 100 = 200 OC

Misalnya saat awal bermain pengguna memiliki 1000 OC. Digunakan 400 OC untuk bermain dan pengguna mendapatkan 200 OC hasil permainan. Jadi, jumlah OC pemain: 1000 - 400 + 200 = 800 OC. RUGI DONG!

>> Mulai bermain (Y/N): Y

Anda Mendapatkan:

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ TOPI | TOPI | TOPI \$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

JACKPOT!!! Selamat, Anda mendapatkan monster TORCIK.
Monster telah ditambahkan ke inventory Anda.

Jika uang agent tidak cukup

>> Mulai bermain (Y/N): Y

Maaf, anda tidak memiliki cukup OC untuk bermain JACKPOT.

B05 – Peta Kota Danville

Karena banyaknya Monster yang tersebar luas di seluruh penjuru kota Danville. Sebagai Agent yang gigih, kamu harus dapat menjelajahi kota untuk menemukan monster-monster tersebut. Maka dari itu, buatlah sebuah tampilan peta dari kota Danville yang dapat kamu jelajahi.

Ketentuan:

- a. Peta bertujuan untuk menunjukkan navigasi Agent dari satu tempat ke tempat lainnya untuk tujuan-tujuan tertentu.
 - b. Peta ini akan digunakan sebagai Menu utama dari program, sehingga fitur F-04 tetap diimplementasikan dengan tambahan legenda dengan fitur bonus ini.
 - c. Tampilan peta hanya berlaku untuk dari sisi Agent, sehingga **tidak perlu dibuat tampilan peta dari sisi Admin.**
 - d. Peta wajib direpresentasikan dalam sebuah **Matriks** berukuran $N \times M$ dengan titik (i, j) merupakan sebuah petak pada baris ke- i dan kolom ke- j .
 - e. Ukuran peta akan didefinisikan melalui file .txt yang dibaca.
 - f. Terdapat command navigasi untuk dapat menggerakkan Agent, misal: **MOVE UP / MOVE DOWN / MOVE LEFT / MOVE RIGHT**
 - g. Pastikan saat melakukan navigasi pada peta, Agent tidak akan bisa menabrak Elemen yang ada pada peta.
 - h. Berikut adalah ilustrasi peta 10×10 yang ditampilkan:

Simbol * menyatakan batas pada peta, sehingga Agent tidak dapat melewati batas tersebut.

- i. Elemen dalam peta yang valid adalah:
 - Agent/Player (P)
 - Shop (S)
 - Arena (A)
 - Laboratory (L)

- Bush (X)
 - JACKPOT! (J), jika mengerjakan B04 - JACKPOT!

j. Untuk melakukan aksi-aksi tertentu, Agent harus berada di tempat yang *adjacent* terhadapnya, yaitu berada di **atas, bawah, kiri, dan kanan** (diagonal tidak termasuk).

k. Untuk fitur Battle, akan digantikan dengan Bush, sehingga Agent harus berada di tempat yang *adjacent* dengan Bush agar dapat melakukan Battle.

l. Berikut adalah konfigurasi yang digunakan untuk melakukan *Load* pada peta:

```

10 # Jumlah baris Matriks
10 # Jumlah kolom Matriks
P#####
#####S###X##
#M#####X##
#####X##L
#####XXXX##
#X#####X##
#X#####X##
#XXX##A###
#####X##
#####XXXXX#
# Akan Menghasilkan Peta:
# * * * * * * * * * * * * *
# * P S X * *
# * * X * *
# * * X L *
# * X X X X * *
# * X * *
# * X * *
# * X X X A * *
# * * *
# * * * * * * * * * * * * *

```

Berikut contoh skenario tampilan pada peta:

a. Tampilan *Idle*

```

* * * * * * * * * * * * * *
* P * * * * * * * * * * * *
*   S   X   * * * * * *
*           X   * * * *
*           X   L   * * *
*   X X X X   * * * *
*   X           * * * *
*   X X X       A   * * *
*           X X X X X   * * *
* * * * * * * * * * * * * *
>>> _

```

b. Tampilan saat melakukan navigasi

```
Agent Purry di posisi: (0,0)
```

```
* * * * * * * * * * * *  
* P *  
* S X *  
* X *  
* X L *  
* X X X *  
* X *  
* X *  
* X X X A *  
* *  
* X X X X X *  
* * * * * * * * * *
```

```
>>> MOVE LEFT
```

```
Agent Purry tidak bisa pindah karena terdapat Obstacle!
```

```
Agent Purry di posisi: (0,0)
```

```
* * * * * * * * * * * *  
* P *  
* S X *  
* X *  
* X L *  
* X X X *  
* X *  
* X *  
* X X X A *  
* *  
* X X X X X *  
* * * * * * * * * *
```

```
>>> MOVE RIGHT
```

```
Agent Purry akan pindah ke kanan!
```

```
Agent Purry di posisi: (1,0)
```

```
* * * * * * * * * * * *  
* P *  
* S X *  
* X *  
* X L *  
* X X X *  
* X *  
* X *  
* X X X A *  
* *  
* X X X X X *  
* * * * * * * * * *
```

```
>>> _
```

c. Tampilan Shop

Agent Purry di posisi: (2,1)

>>> **SHOP**

Agent Purry tidak berada di area Shop!

Agent Purry di posisi: (1,3)

>>> **SHOP**

Akan mengakses Shop, karena berada pada posisi yang *adjacent* dengan Shop (S)

d. Tampilan Arena

Agent Purry di posisi: (6,3)

>>> ARENA

Agent Purry tidak berada di area Arena!

Agent Purry di posisi: (6,6)

>>> ARENA

Akan mengakses Arena, karena berada pada posisi yang *adjacent* dengan Arena (A)

e. Tampilan Laboratory

Agent Purry di posisi: (0,7)

>>> LABORATORY

Agent Purry tidak berada di area Laboratory!

Agent Purry di posisi: (2,9)

>>> LABORATORY

Akan mengakses Laboratory, karena berada pada posisi yang *adjacent* dengan Laboratory (L)

f. Tampilan Battle

Agent Purry di posisi: (2,1)

>>> BATTLE

Agent Purry tidak berada di area Bush untuk melakukan Battle!

Agent Purry di posisi: (6,9)

>>> BATTLE

Akan melakukan Battle, karena berada pada posisi yang *adjacent* dengan Bush (X)

BXX – Kreativitas



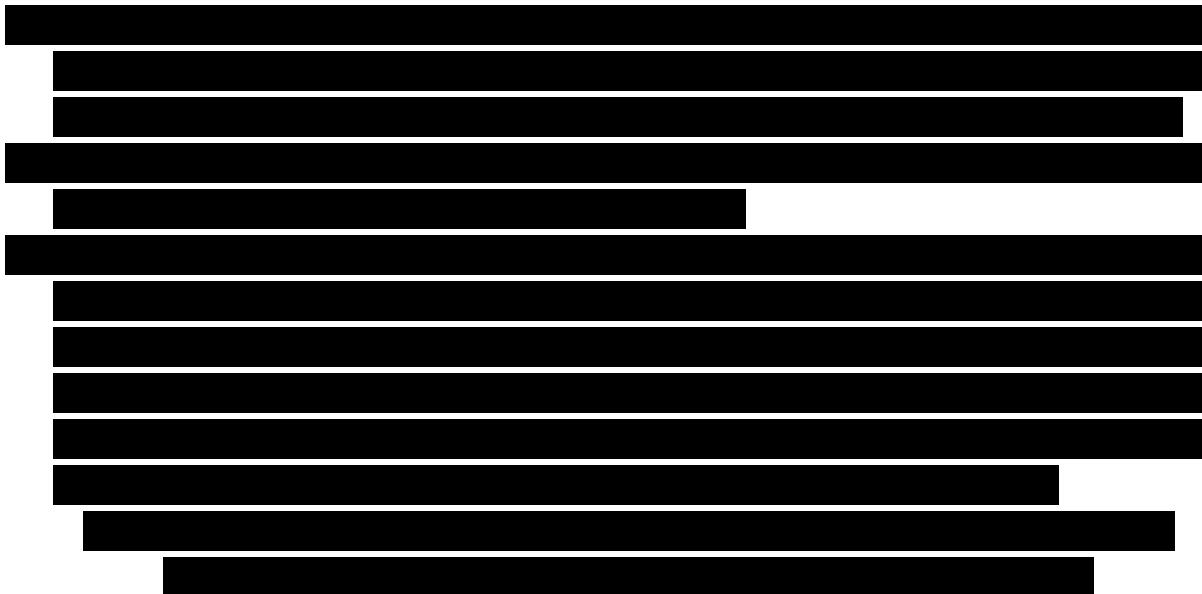
Mr. Monogram sangat menyukai Agent yang kreatif dan dapat berpikir *out of the box*. Oleh karena itu, kalian dapat mengimplementasikan **fitur di luar yang telah disebutkan** di spesifikasi. Sebelum itu, silakan mengajukan fitur tersebut kepada Asisten, atau via Sheets QNA. Perlu dicatat bahwa fitur tambahan tersebut jangan sampai merubah alur dari spesifikasi utama. Jika argumen dan fitur yang kalian ajukan cukup mengesankan, kalian akan diberi **nilai bonus** dari Mr. Monogram.

Berikut adalah beberapa contoh fitur bonus lainnya yang dapat diimplementasikan:

- Kreativitas penggunaan ASCII art ataupun memberikan warna text output untuk mempercantik tampilan program.
- Melakukan *Hashing* terhadap penyimpanan password dalam database menggunakan algoritma Caesar Cipher yang dibuat sendiri. Buatlah fungsi untuk melakukan *encrypt* serta *decrypt*. (ini bonus daspro 2021)
- Selain Jackpot, dapat mengimplementasikan Mini Game lainnya untuk mendapat reward, seperti Tic Tac Toe (ini bonus daspro 2021), Wordle, dsb.
- Arena yang dapat diatur monster serta hadiah dari tiap stage nya. (Dapat menambahkan database Arena dan mengimplementasikan Arena Management).
- Fitur PVP (Player Versus Player), yaitu melakukan Battle antar user. Terdapat menu tambahan untuk melakukan PVP sehingga bisa terdapat 2 user yang melakukan log-in.
- Healing Fountain, pada spek utama, walaupun HP monster berkurang selama Battle, saat masuk ke Battle selanjutnya, HP langsung penuh kembali, jika

mengimplementasikan ini, maka setelah Battle, harus disimpan State HP monster yang berkurang, lalu Monster dapat disembuhkan melewati *Healing Fountain*.

- Fitur Quest, misalkan quest untuk menaklukkan N monster atau menyelesaikan Arena sebanyak N stage, bisa mendapat reward tertentu. Quest dapat di-track progress nya dan disimpan.
- Fitur Statistik, misalnya dapat menampilkan total jumlah monster yang dikalahkan, total damage dari tiap monster, dll. (Dapat dibuat database tersendiri).
- Dan lain-lain....



Struktur Data File

Program perlu membaca beberapa data dari file eksternal untuk mengoperasikan sistem ini. Format file yang diminta adalah file dengan ekstensi .csv yang dipisahkan dengan semicolon/titik koma(;)!

Karena *Dr. Asep Spakbor* melakukan *cyber attack* kepada markas rahasia O.W.C.A., bahasa Python tidak memiliki fungsi **split()** dan **import csv**, maka buatlah CSV parser sendiri untuk membantu Agen P. Fungsi ini WAJIB diimplementasikan dan tidak boleh menggunakan library.

Berikut adalah contoh isi file-csv:

```
id;username;password;role;oc
1;Asep_Spakbor;asepwow123;admin;9999
2;Agen_P;platypus123;agent;0
3;B4ngk1dd0ssss;bangkitganteng;agent;1337
4;Kenny_agen_rahasia;kribogeming55;agent;6699
```

Silakan membuat CSV untuk file lainnya sendiri. CSV akan disimpan pada sebuah folder yang sudah didefinisikan pada *command save*, dan nama file bersifat sudah pasti/*fixed* untuk memudahkan penggerjaan. Untuk template CSV dapat diakses pada tautan berikut:

[IF1210 Sample Database Tugas Besar](#)

Sangat direkomendasikan untuk membuat *monster stats* serta *currency* yang *balance* untuk mempermudah penggerjaan dan *testing* tugas besar ini.

File User (user.csv)

Untuk file user, sudah terdapat data default yang dimasukkan yang bertipe role “admin”, karena tidak dapat dilakukan registrasi untuk role tersebut.

Nama Atribut	Deskripsi	Tipe Data
id	ID dari user, bersifat unik dan fixed, di-generate secara otomatis dari program. (Dapat berupa urutan: 1, 2, 3, dst.)	INTEGER
username	Username masing-masing Agent. Bersifat unik dan harus diverifikasi dalam program. Username akan digunakan untuk login.	STRING
password	Password dari masing-masing pengguna.	STRING
role	Role dari pengguna, dapat bertipe: - Admin - Agent	STRING
oc	O.W.C.A. Coin merupakan saldo setiap pengguna untuk melakukan transaksi.	INTEGER

File Monster (monster.csv)

Untuk Monster, berikan data default minimal 1, agar saat proses registrasi, Agent dapat memilih monster default nya.

Nama Atribut	Deskripsi	Tipe Data
id	ID dari monster, bersifat unik dan fixed, di-generate	INTEGER

	secara otomatis dari program. (Dapat berupa urutan: 1, 2, 3, dst.)	
type	Type dari masing-masing Monster. Bersifat unik dan harus diverifikasi dalam program.	STRING
atk_power	Kekuatan penyerangan dari tiap monster.	INTEGER
def_power	Kekuatan bertahan dari tiap monster	INTEGER
hp	Banyaknya nyawa (Hit Points) dari tiap monster	INTEGER

File Inventory - Item (item_inventory.csv)

File ini akan menyimpan kepemilikan dari Potion (dan Monster Ball jika mengerjakan B03).

Nama Atribut	Deskripsi	Tipe Data
user_id	Merepresentasikan kepemilikan dari inventory. Didapat dari data user.	INTEGER
type	Merepresentasikan tipe dari item tersebut, yaitu: <ul style="list-style-type: none"> - strength - resilience - healing - monster_ball (jika mengerjakan B03) 	STRING
quantity	Merepresentasikan jumlah dari item tersebut	INTEGER

File Inventory - Monster (monster_inventory.csv)

File ini akan menyimpan kepemilikan dari tiap Monster.

Nama Atribut	Deskripsi	Tipe Data
user_id	Merepresentasikan kepemilikan dari inventory. Didapat dari data user.	INTEGER
monster_id	Merepresentasikan ID dari Monster.	INTEGER
level	Merepresentasikan level dari Monster yang disimpan oleh pengguna, secara default bernilai 1.	INTEGER

File Shop - Item (item_shop.csv)

File ini akan menyimpan Item apa saja yang tampil di Shop.

Nama Atribut	Deskripsi	Tipe Data
type	Merepresentasikan tipe dari item tersebut, yaitu: <ul style="list-style-type: none">- strength- resilience- healing- monster_ball (jika mengerjakan B03)	STRING
stock	Merepresentasikan jumlah stok yang ada untuk tiap Item yang dijual.	INTEGER
price	Merepresentasikan harga dari tiap Item yang dijual.	INTEGER

File Shop - Monster (monster_shop.csv)

File ini akan menyimpan Monster apa saja yang tampil di Shop.

Nama Atribut	Deskripsi	Tipe Data
monster_id	Merepresentasikan ID dari Monster.	INTEGER
stock	Merepresentasikan jumlah stok yang ada untuk tiap Item yang dijual.	INTEGER
price	Merepresentasikan harga dari tiap Item yang dijual.	INTEGER

Ketentuan dan Batasan

A. Ketentuan Umum

1. Tugas dikerjakan berkelompok dengan anggota kelompok sebanyak 5–6 orang per kelompok dengan anggota kelompok yang telah ditentukan oleh asisten. Link **pembagian kelompok** serta **pemilihan asisten** dapat dilihat pada tautan berikut:
 [IF1210 Pembagian Kelompok dan Asisten](#)
2. **Template MoM Asistensi** dan **Laporan Tugas Besar** dapat dilihat pada tautan berikut:
 [\[TEMPLATE\] IF1210_FormAsistensiTB_\[1/2\]_K\[XX-Y\].docx](#)
 [\[TEMPLATE\] IF1210_LaporanTB_K\[XX-Y\].docx](#)
3. Pertanyaan terkait Tugas Besar, dapat diajukan dalam **Spreadsheets QNA** yang ada pada tautan berikut:
 [IF1210_QNA](#)
4. Mekanisme penggerjaan silakan perhatikan bagian [Pengumpulan dan Deliverables](#)

B. Batasan Pengerjaan

1. Menggunakan bahasa **Python 3.9+**
2. Hanya boleh `import os, sys, math, time, argparse, datetime` dan `modul-modul` serta fungsi yang telah dibuat sendiri.
Tidak Boleh memakai pandas dan sejenisnya!!!
3. Tidak boleh menggunakan library eksternal dalam bentuk apapun
4. Tidak boleh menggunakan `.split()`, `.sort()`
Silakan mengimplementasikan sendiri.
Pengecualian untuk `lower()` dan `upper()`.
5. Hanya boleh memakai fungsi-fungsi bawaan dari `python` yang diajarkan di kelas.
Silakan mengimplementasikan sendiri fungsi tambahan yang dibutuhkan.
6. **Tidak boleh membuat temporary CSV sebagai tempat penyimpanan sementara.**
Gunakan List dan operasi list yang dibuat sendiri. **CSV hanya boleh digunakan sebagai penyimpanan ketika dipanggil load dan save.**
7. Program berjalan sebagai satu kesatuan. Main program hanya boleh satu. Main program ini akan memanggil fungsi/prosedur dari modul lain yang berada di file lain.
8. Tugas besar harus diimplementasikan dalam **paradigma prosedural dan fungsional** seperti yang diajarkan di kelas. Tidak boleh menggunakan paradigma lain seperti OOP, misalnya menggunakan class.

C. Jadwal Pelaksanaan Kegiatan

Kegiatan	Deadline
Launching Tugas Besar	26 Apr 2024
Batas Pengisian Asisten Kelompok	28 Apr 2024, 12.10 WIB
Pengerjaan Tugas Besar:	
• Deadline Asistensi 1	5 Mei 2024
• Deadline Asistensi 2	12 Mei 2024
Pengumpulan Deliverables (laporan dan kode program)	18 Mei 2024, 12.10 WIB
Demo Tugas Besar	20-24 Mei 2024

D. Pengerjaan Tugas Besar

1. Asistensi:
 - a. Setiap kelompok akan didampingi oleh seorang asisten pembimbing dari asisten IF1210. Akan disediakan mekanisme dan informasi untuk mengontak asisten pembimbing.
 - b. Setiap kelompok wajib mengontak asisten pembimbing untuk melakukan asistensi **minimum 2 kali** sepanjang pengerjaan Tugas Besar:
 - i. **Asistensi 1:** paling lambat **Minggu, 5 Mei 2024**
 - ii. **Asistensi 2:** paling lambat **Minggu, 12 Mei 2024**
 - c. Hal-hal yang harus diperhatikan:
 - i. Asisten **berhak menolak** jadwal yang diajukan oleh mahasiswa apabila mengontak terlalu dekat dengan deadline asistensi.
 - ii. Maksimal mengontak asisten **paling lambat H-1** dari jadwal asistensi yang diajukan
 - iii. Asistensi dapat dilakukan secara **daring ataupun luring** tergantung dengan ketersediaan asisten, sangat disarankan untuk berdiskusi dengan asisten secara lebih lanjut.
 - iv. Setelah menyepakati jadwal dengan asisten, silakan membuat **Google Calendar** (disertakan link Google Meet / Zoom jika pelaksanaan secara daring) dan mengundang asisten. Format penamaan Google Calendar adalah:
IF1210_Asistensi_[1/2]_K[XX-Y]
Contoh: *IF1210_Asistensi_1_K99-Z*

- d. Asistensi dapat digunakan untuk melakukan **klarifikasi dan diskusi spesifikasi** tugas besar. Asistensi **tidak bertujuan** untuk meminta *debugging* / mencari Error, karena *obstacle* tersebut adalah wadah kalian dapat belajar memahami pemrograman.
 - e. Setiap kali asistensi, praktikan diminta **membuat MoM** (Minutes of Meeting) asistensi dan mengisinya ke dalam form asistensi dengan format yang ditetapkan. MoM asistensi akan dilampirkan ke dalam laporan.
2. Demo:
- a. Setelah masa deadline peng交aan Tugas Besar, akan dilaksanakan Demo Tugas Besar secara sinkron dengan asisten masing-masing.
 - b. Demo bertujuan untuk mempresentasikan hasil peng交aan tugas besar.
 - c. Program akan dijalankan di depan asisten dan akan diuji fungsionalitas, validasi, dan juga pemahaman kalian terhadap program.
 - d. Mekanisme demo akan dijelaskan lebih lanjut di kemudian hari.

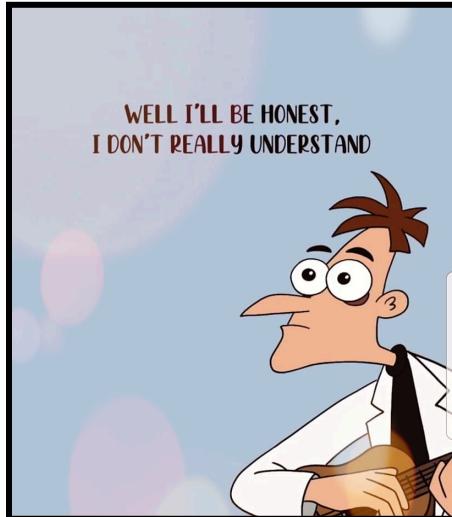
Pengumpulan dan Deliverables

1. Untuk tugas ini Anda diwajibkan menggunakan *git version control system* dengan menggunakan sebuah repository **private** di Github Classroom "Labpro-21" (gunakan email @stei.itb.ac.id agar gratis). Silakan akses Github Classroom pada: [Github Classroom - Labpro 21](#)
2. Silakan membuat Group dengan format nama "KXX-Y", dengan penjelasan:
XX: Nomor kelas (ditulis dalam 2 digit, misal: K03)
Y: Kode kelompok (ditulis dalam 1 huruf, misal: A)
Contoh: K03-A
3. Pembuatan Group dilakukan **satu kali** untuk tiap kelompok, setelah dibuat, anggota kelompok lain **tidak perlu membuat Group lagi** dan dapat langsung bergabung dengan Group yang sudah dibuat.
4. Akan terdapat *repository private* yang dibuat secara **otomatis** untuk tiap Group, silakan gunakan *repository* tersebut untuk pengeroaan Tugas Besar. Repository tersebut akan memiliki nama: if1210-2024-tubes-kxx-y
5. Sumber kode program yang dibuat sesuai standar yang diajarkan di kuliah, yaitu:
 - a. **Menggunakan nama variabel dan file yang berarti.**
Contoh yang salah: aaa, bbb, akusayangkamu, akucintatugasbesar dsb.
 - b. **Bersih**
Hanya mengandung bagian-bagian yang diperlukan
 - c. **Well-Commented**
6. Repository berisi:
 - Folder **src**: Berisi kode program / fungsi-fungsi.
 - Folder **data**: Contoh data yang dapat di load
 - Folder **doc**: Berisi laporan dan MoM dalam **.pdf**
 - **main.py**: Berisi kode program utama sebagai *driver* dari fungsi-fungsi yang dibuat.
 - **README.md**: Deskripsi singkat cara menjalankan program
7. Laporan dan MoM Tugas Besar dikumpulkan pada GitHub Repository yang telah dibuat pada folder **doc**. Pastikan memiliki format **.pdf**
8. Pengumpulan dilakukan dengan membuat **release** dengan major version **1** seperti **v1.x**. Versi patch dapat digunakan jika ada perbaikan. Contoh: Jika ada revisi pada versi **v1.0**, selama tidak melewati *deadline*, dipersilahkan melakukan revisi kode dan melakukan *release* lagi dengan versi **v1.1**. Penilaian akan dilakukan terhadap versi rilis terakhir.

Referensi

- [Wikipedia - Linear Congruential Generator \(LCG\)](#)
- [Python Argument Parser Tutorial \(Argparse\)](#)
- [Git Best Practices](#)
- [Git Semantic Commit Message](#)
- [Typing in Python](#)
- [Caesar Cipher](#)

Extras



- ssssteh -

"EZ kok dulu aku ngerjain cuman 20 jam (x 3 minggu)"

- bangkid -

"Smgt  "

- chi -



- ditra -



- aul -



- Nik -



"Potret anak STEI 23 melihat tubes daspro"

- Ep -

 **You**
buatlah kata-kata kepada mahasiswa STEI 23 untuk menyemangati mereka menghadapi tugas besar mata kuliah IF1210 Dasar Pemrograman
< 3/3 > 

 **ChatGPT**
Tentu, berikut adalah pesan semangat untuk mahasiswa STEI 23 yang menghadapi tugas besar mata kuliah IF1210 Dasar Pemrograman:

Halo Mahasiswa STEI 23,

Kalian sedang mengarungi lautan pengetahuan dengan mata kuliah IF1210 Dasar Pemrograman, sebuah tantangan yang memang tak mudah namun juga tak takluk. Ketekunan dan semangat kalian adalah kunci untuk menaklukkan setiap tantangan yang ada.

Ingatlah, setiap baris kode yang kalian tulis adalah langkah maju dalam perjalanan kalian sebagai programmer yang handal. Tiap kesalahan adalah pelajaran berharga yang membentuk kalian menjadi lebih baik. Jangan pernah ragu untuk bertanya, belajar dari teman-teman seangkatan, dan eksplorasi lebih jauh.

Ketika merasa lelah, ingatlah mengapa kalian memilih jalur ini. Impian, ambisi, dan tujuan yang menginspirasi kalian. Setiap kali merasa sulit, renungkan pada hasil akhir yang akan menjadi bukti betapa kuatnya kalian.

Kalian bukan hanya mahasiswa biasa. Kalian adalah bagian dari komunitas yang saling mendukung dan menginspirasi. Bersama-sama, kalian akan melampaui batas-batas yang tampaknya tak terjangkau.

Teruslah berjuang, teruslah belajar, dan percayalah pada kemampuan diri sendiri. Tugas besar ini hanyalah salah satu puncak dari banyak pencapaian yang menanti kalian di masa depan.

Semangat, kejar mimpi-mimpi besar kalian!

Salam,

[Your Name]

– kibo –