

Pokémon Moonstone

1. Deskripsi Tugas Besar



Gambar 1. Tampilan permainan Pokémon Emerald

Pada tugas besar ini, peserta kuliah IF2111 Algoritma dan Struktur Data diminta untuk mengimplementasikan permainan Pokémon Moonstone dengan antarmuka berbasis *command line* (Command Line Interface/CLI). Pada permainan ini, pemain dapat memiliki pokémon, menjelajahi dunia pokémon, bertarung melawan pokémon, maupun membeli barang-barang. Referensi permainan lebih lanjut dapat mengacu pada permainan Pokémon Emerald yang dapat dilihat pada pranala:

https://en.wikipedia.org/wiki/Pok%C3%A9mon_Emerald.

2. Spesifikasi Tugas Besar

Pada permainan Pokémon Moonstone ini, terdapat beberapa fungsionalitas yang harus diimplementasikan, yakni sebagai berikut:

2.1 Main Menu

Pada awal permainan, pengguna dapat melihat menu utama yang berisi perintah:

- a. *Mulai*: memulai permainan baru, pengguna dapat mengisi nama dan pokémon pertama yang ingin digunakan.

- b. *Petunjuk*: berisi petunjuk dalam memainkan permainan.
- c. *Tentang*: berisi identitas pembuat program.
- d. *Keluar* : keluar dari permainan.

2.2 Permainan Baru

Ketika memulai permainan baru, pengguna dapat mengisi nama dan pokémon pertama yang ingin digunakan. Daftar pokémon yang dapat digunakan ketika awal permainan baru harus ditampilkan ke layar. Daftar pokémon tersebut dapat dilihat pada *file* `pokemon-permainan-baru.txt`. Selain itu, ketika pemain baru memulai permainan baru, pemain memiliki *experience* (EXP) sebesar 0 serta uang awal sebesar 1000P.

2.3 Deskripsi Permainan

1. Pemain memiliki informasi sebagai berikut:
 - a. Nama
 - b. Pokémon aktif dan HP-nya
 - c. Daftar pokémon beserta HP-nya.
 - d. Uang
 - e. *Experience* (EXP)
 - f. *Level* (pemetaan berdasarkan EXP dapat dilihat pada *file* `pemetaan-level.txt`)
 - g. Posisi pada peta (X,Y)
2. Pemain memiliki tas yang dapat diisi *potion* atau Poké Ball. *Potion* dapat digunakan untuk menambah *health point* (HP) dari pokémon sedangkan Poké Ball dapat digunakan untuk menangkap dan menyimpan pokémon.
3. Pemain memiliki uang (Pokédollar) yang dapat digunakan untuk membeli barang di toko.
4. Pemain dapat menjelajahi dunia Pokémon yang didefinisikan melalui sebuah (beberapa peta yang saling terhubung jika mengimplementasikan bonus). Setiap koordinat pada peta dapat merepresentasikan jalan setapak, toko, padang rumput, atau semak-semak. Peta dapat dilihat pada *file* `peta.txt`.

5. Pemain dapat *bertemu* dengan pokémon dan bertanding dengan pokémon tersebut ketika berada di semak-semak. Kemunculan pokémon di suatu koordinat semak-semak dihitung berdasarkan nilai *random* dengan probabilitas sebesar 50%.
6. Ketika melawan pokémon, permainan dilaksanakan selayaknya pada permainan Pokémon (*turn-based*) dengan pemain mendapat giliran terlebih dahulu. Dalam pertarungan, pemain dapat memilih opsi untuk:
 - a. Menyerang
 - b. Membuka tas
 - c. Kabur
7. Ketika menyerang dengan kemampuan yang dimiliki, besar *attack point* yang dapat diberikan bergantung pada tipe pokémon yang digunakan serta tipe pokémon lawan. Perhitungan kekuatan pokémon dapat dilihat pada *file ability.txt*. Permainan harus menampilkan informasi ketika pokémon pemain mendapatkan nilai *combo* maupun jika lawan mendapat nilai *combo* untuk *attack point*.
8. Dengan membuka tas ketika melawan pokémon, pemain dapat menggunakan barang yang dimilikinya. *Potion* dapat digunakan untuk meningkatkan HP pokémon sedangkan Poké Ball untuk menangkap pokémon. Keberhasilan penangkapan pokémon dihitung berdasarkan nilai *catch rate* dari masing-masing pokémon.
9. Ketika memilih untuk kabur dari suatu pertandingan melawan pokémon, keberhasilan untuk kabur ditentukan secara *random* dengan probabilitas sebesar 50%.
10. Pemain bisa mendapatkan *experience* (EXP) serta uang (Pokédollar) apabila menang melawan pokémon. Besar EXP maupun Pokédollar yang didapatkan bergantung pada jenis pokémon yang dilawan.
11. Pemain dapat membeli barang di toko. Daftar barang yang dijual di toko dapat dilihat pada *file toko.txt*.
12. Permainan akan berakhir apabila *health point* (HP) dari seluruh pokémon milik pemain bernilai 0.

3. File Eksternal

3.1 File `pokemon-permainan-baru.txt` dan `pokemon.txt`

Format <i>file</i>	Contoh
nama pokémon tipe pokémon health point (HP) attack range exp range money range catch rate	Torchic FIRE 45 60-80 65-100 130-262 6

Keterangan:

- nama pokémon: nama dari pokémon.
- tipe pokémon: tipe pokémon berpengaruh terhadap *attack rate* ketika melawan pokémon lain.
- health point: atribut yang mengindikasikan kondisi pokémon dalam permainan.
- attack range: rentang kuat serangan yang mampu diberikan suatu pokémon kepada lawannya. Perhitungan dilakukan secara *random* berdasarkan rentang tersebut.
- exp range: rentang *experience* (EXP) yang didapatkan pemain apabila berhasil mengalahkan pokémon tersebut. Perhitungan dilakukan secara *random* berdasarkan rentang tersebut.
- money range: rentang uang (Pokédollar) yang didapatkan pemain apabila berhasil mengalahkan pokémon tersebut. Perhitungan dilakukan secara *random* berdasarkan rentang tersebut.
- catch rate: probabilitas pemain mendapatkan pokémon tersebut dengan Poké Ball.

3.2 File `peta.txt`

File eksternal	Peta asli
<pre> R R R R S S S S S S R R M R S S S S S S J J J J S S S S S S S S R J R S S S S R S S R J R R R R R R S S R J R P R R M R J J J J J J J J J X S S R R R J R R S R S S S S R J R S S S S S S S S X S S S S </pre>	

Keterangan:

P		Poin awal permainan
J		Jalan setapak
R		Padang rumput
S		Semak-semak
M		Toko

Tanda X menunjukkan pintu ke peta lain jika mengimplementasikan bonus. Jika tidak mengimplementasikan bonus, tanda X dapat dianggap sebagai jalan setapak (J).

3.3 File `toko.txt`

Format <i>file</i>	Contoh
tipe_barang nama_barang harga efek	<pre> pokeball Poke_Ball 200 0 pokeball Great_Ball 600 10 potion Potion 300 20 potion Super_Potion 700 50 potion Hyper_Potion 1200 200 </pre>

Keterangan:

- Tipe barang hanya ada dua jenis, yakni `pokeball` atau `potion`.
- Efek barang bergantung pada tipe barang. Efek barang bertipe `pokeball` akan berpengaruh pada pertambahan nilai *catch rate* dalam persen (%) sedangkan efek

barang bertipe *poison* akan berpengaruh pada nilai *health point* yang didapatkan ketika *poison* tersebut digunakan.

Contoh:

pokeball Great_Ball 600 10 ketika digunakan untuk melawan *Torchic*, nilai *catch rate*-nya menjadi: $45\% + 10\% = 65\%$

poison Super_Poison 700 50 ketika digunakan untuk suatu Pokémon akan menambah *health point*-nya (HP) sebanyak 50 poin.

3.4 File *ability.txt*

	BUG	ELECTR	FIRE	GRASS	POISON	PSYCHIC	WATER
BUG	1	1	2	0.5	2	1	1
ELECTR	1	0.5	1	1	1	1	1
FIRE	0.5	1	0.5	0.5	1	1	2
GRASS	2	0.5	2	0.5	2	1	0.5
POISON	2	1	1	0.5	1	2	1
PSYCHIC	2	1	1	1	1	0.5	1
WATER	1	2	0.5	2	1	1	0.5

Perhitungan untuk serangan Pokémon X terhadap Pokémon Y dapat dihitung berdasarkan nilai pada perpotongan kolom 1 (Pokémon X) dengan baris 1 (Pokémon Y).

Contoh perhitungan:

- Pokémon A bertipe FIRE dan Pokémon B bertipe GRASS maka serangan Pokémon A menjadi 2 kali lipat dari sebelumnya sedangkan serangan Pokémon B menjadi $\frac{1}{2}$ kali lipat dari sebelumnya.
- Pokémon C bertipe WATER dan Pokémon D bertipe ELECTR maka serangan Pokémon C menjadi $\frac{1}{2}$ kali lipat dari sebelumnya sedangkan serangan Pokémon D tetap (1 kali lipat dari sebelumnya).

4. Bonus

- Tampilan menarik.
- Terdapat fungsi *save* dan *load* permainan.

3. Ada narasi dan cerita seiring perjalanan.
4. Mengimplementasikan top 10 *highscore* berdasarkan total *experience* yang dimiliki pemain hingga permainan berakhir dengan informasi tambahan berupa waktu permainan berakhir. Top 10 *highscore* diurutkan dari nilai tertinggi hingga nilai terendah
5. Mengimplementasikan peta 2, peta 3, dan peta 4.

5. Deskripsi Pengerjaan Tugas

5.1 Batasan

Berikut ini adalah batasan yang harus diperhatikan:

- a. Implementasi tugas besar harus menggunakan bahasa C. Tidak diperkenankan untuk menggunakan bahasa pemrograman lainnya.
- b. Masukan dari pengguna harus selalu divalidasi. Apabila masukan pengguna tidak sesuai, maka harus ada peringatan yang menginformasikan bahwa masukan pengguna tersebut salah dan harus memberitahukan pengguna contoh masukan yang benar.
- c. Berikan komentar pada kode yang dibuat. Usahakan agar kode yang dibuat benar-benar *well-commented*.
- d. Bagian bonus dikerjakan apabila semua spesifikasi utama telah diimplementasikan. Apabila bonus telah dikerjakan tetapi terdapat spesifikasi utama yang belum dikerjakan, maka bonus tersebut akan gugur.

5.2 Penggunaan ADT

Pengerjaan tugas ini didasarkan pada materi kuliah IF2111 yang diajarkan di kelas. Implementasi ADT dalam tugas besar ini adalah sebagai berikut.

1. ADT Point

Pemain dapat berkeliling kota untuk mencari pokemon maupun membeli barang. Posisi pemain pada kota harus direpresentasikan dalam ADT Point yang menyatakan koordinat X dan Y pada peta.

2. ADT Matriks

Kota tempat permainan berlangsung dapat diimplementasi dalam ADT Matriks.

3. Mesin Kata

ADT mesin kata dimanfaatkan untuk memuat *file-file* eksternal yang digunakan dalam permainan seperti file peta, *ability*, toko, dll.

4. ADT Stack

Penyimpanan dan pengambilan pokémon yang berhasil ditangkap harus menggunakan ADT Stack. Sehingga jika pada saat pertarungan *health point* pokémon mencapai nilai 0 maka Pokémon pengganti diambil dari *stack*.

5. ADT Queue

Kemunculan pokémon di semak-semak diatur berdasarkan antrian (*queue*) dengan urutan pokémon pada *file* `pokemon.txt` dilanjutkan dengan *file* `pokemon-permainan-baru.txt` (total 7 pokémon). Pokémon yang akan muncul pada pertarungan adalah Pokémon pada antrian pertama, kedua, dst. Misalnya, pokémon memiliki urutan antrian A, D, dan T. Maka, jika pemain bertarung melawan pokémon, maka Pokémon lawan yang muncul adalah A. Selanjutnya jika Pokémon lawan kalah/mati, pokémon yang dilawan selanjutnya adalah D. Begitu juga seterusnya.

6. ADT List Linear

ADT List Linier juga dapat digunakan sebagai *container* potion atau Poké Ball.

5.3 Deliverables

Berikut ini adalah hal-hal yang harus diperhatikan dalam pengumpulan deliverable:

- a. Kode program harus disertai dengan komentar-komentar yang dapat menjelaskan kode yang dibuat.
- b. Penamaan *file* untuk ADT:

`nama-ADT_nomor-kelompok.h` untuk file header

`nama-ADT_nomor-kelompok.c` untuk realisasi ADT

Contoh: `matriks_01.h` untuk file *header* dan `matriks_01.c` untuk realisasi ADT

Proposal Tugas Besar IF2111 Algoritma dan Struktur Data
Semester I – 2016/2017

- c. Penamaan *file* untuk *driver*:

main_nomor-kelompok.c

Contoh: main_01.c

- d. Untuk setiap *file* yang Anda buat, buatlah *header* sebagai berikut:

/* Nomor Kelompok	:	*/
/* Nama file	:	*/
/* Deskripsi	:	*/

- e. Semua file yang dikumpulkan dikompres menjadi .zip dengan format penamaan:

IF2111_TB_nomor-kelompok.zip

File zip berisi tiga *folder*, yaitu:

src : berisi kode program

bin : berisi *executable* program

doc : berisi *softcopy* laporan

- f. Setiap kelompok diwajibkan memasukan *file* tugas besar kedalam CD dan dikumpulkan bersama dengan laporan *hardcopy*. Pada saat pengumpulan *hardcopy*, sertakan pula *form* asistensi.

- Laporan untuk tugas besar ini meliputi hal-hal berikut ini:
- Ringkasan
- Penjelasan tambahan spesifikasi tugas
- Program utama
- Algoritma menarik
- Pembagian kerja dalam kelompok
- Notulen rapat dan log activity setiap anggota kelompok

Template laporan diberikan terpisah dengan *file* ini.

6. Waktu Pengerjaan

Kegiatan	Tanggal	Jam
Asistensi 1	10-18 November 2016	Ditentukan dengan asisten
Asistensi 2	15-25 November 2016	Ditentukan dengan asisten

Proposal Tugas Besar IF2111 Algoritma dan Struktur Data
Semester I – 2016/2017

<u>Deadline</u> softcopy (kumpulkan ke oddyseus.if.itb.ac.id)	2 Desember 2016	18.00 WIB
<u>Deadline</u> hardcopy (kumpulkan ke oddyseus.if.itb.ac.id)	2 Desember 2016	18.00 WIB
Demo tugas besar	12-16 Desember 2016	Ditentukan dengan asisten

Catatan:

- Asistensi WAJIB dilakukan sebanyak minimal 2 kali selama pengerjaan tugas besar ini. Semua anggota kelompok harus hadir pada saat asistensi.
- Setiap kali asistensi, praktikan harus mengisi *form* asistensi dengan lengkap. *Form* asistensi hanya boleh diisi dan diparaf saat asistensi berlangsung (tidak boleh menyusul)
- Pembuatan janji untuk asistensi dan demo dilakukan paling lambat H-1 pelaksanaan asistensi.
- Segala bentuk kecurangan dan plagiarisme akan ditindak sesuai dengan ketentuan yang berlaku.