

Penerapan Algoritma *Greedy* pada *Game Brigandine* *The legend of Forsena*

Bobby Indra Nainggolan / 13515102

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
Jalan Ganesha No. 10 Bandung 40132, Indonesia
13515102@std.stei.itb.ac.id

Abstract—*Brigandine : The Legend of Forsena* merupakan sebuah *game Role-Playing Game(RPG)* dengan konsol *playstation(PS 1)*. *Game* ini memiliki tujuan untuk menguasai seluruh wilayah/kerajaan yang ada pada peta *game* dengan cara melakukan invasi pada kerajaan-kerajaan yang berdekatan dengan kerajaannya sendiri serta mampu bertahan dari serangan dari kerajaan lain.

Pada permainan ini, pertahanan/penyerangan tersebut tidak dapat lepas dari analisis keuangan dan juga kuota kerajaan dalam pembangkitan monster yang dibutuhkan. Makalah ini kemudian dirancang dengan tujuan untuk menunjukkan pemanfaatan algoritma *greedy* dalam menyelesaikan persoalan yang sering terjadi pada *game* ini, mulai dari manajemen kuota kerajaan hingga mode pertarungan pada *game*.

Keywords— *Algoritma greedy, pertarungan, manajemen kuota kerajaan, pertarungan.*

I. PENDAHULUAN

A. Latar Belakang Game

Brigandine : The Legend of Forsena karya dari Hearty Robin merupakan sebuah *game* yang dirilis pada tahun 1998. *Game* ini merupakan *game* yang sangat terkenal dari konsol *Playstation 1*. Pada tahun 2000, *game* ini kemudian di-*remake* oleh Robin menjadi *game* dengan fitur yang lebih banyak dan menarik, salah satunya adalah fitur bermain secara *multiplayer*.

Game ini memiliki genre permainan *peran taktik*. *Game* ini diawali dengan *story* singkat mengenai keadaan sebuah benua fiksi bernama Forsena pada masa tersebut. Setelah *story* selesai, pemain akan diberikan pilihan negara yang akan digunakan saat bermain. Tujuan dari *game* ini adalah untuk menguasai semua negara/kerajaan yang ada di benua fiksi Forsena.

Penaklukan negara/kerajaan tersebut dilakukan dengan cara penyerangan terhadap negara yang akan diserang. Penyerangan tersebut akan dilakukan oleh pasukan yang

merupakan campuran dari manusia dan makhluk fiksi pada *game*.

B. Karakter Player dalam Game

Dalam *Brigandine*, terdapat 6 negara bagian. Pada wilayah Barat Terdapat kerajaan New Almekia yang dipimpin oleh Prince Lance, wilayah kerajaan ini ditandai dengan kerajaan-kerajaan berwarna merah muda. Pada wilayah timur terdapat kerajaan Leonia yang dipimpin oleh Queen Lynosse, wilayah kerajaan ini ditandai dengan kerajaan kerajaan warna biru muda pada benderanya. Selanjutnya pada bagian Selatan dibagi atas 2 kerajaan yaitu kerajaan Iscalio yang merupakan kerajaan warna kuning, kerajaan ini dipimpin oleh Super Tyrant Dryst dan kerajaan Carleon yang merupakan kerajaan berwarna hijau, kerajaan ini dipimpin oleh Warlock Cai.

Kerajaan pada wilayah Utara yaitu Kerajaan Norgard dipimpin oleh Lord Vaynard (yang merupakan sampul gambar video *game* ini), kerajaan ini ditandai dengan kerajaan warna biru tua pada benderanya. Yang terakhir adalah kekaisaran Esgares yang dipimpin oleh Emperor Zameckis, kerajaan ini ditandai dengan kerajaan kerajaan berwarna ungu pada benderanya.

Sebenarnya, pada *game* ini kerajaan Emperor Zameckis yaitu Esgares, tidak dapat dipilih untuk digunakan sebagai karakter yang akan dimainkan didalam *game*. Emperor Zameckis merupakan karakter jahat /protagonist dalam *game* ini.

C. Storyline Game

Kisah awal permainan ini bermula saat terjadi perang besar antara Almekia dan Norgard. Perang ini dimenangkan oleh Almekia. Pada saat pertempuran perang Raja Doremit meninggal. King Henguist yang merupakan pemimpin Almekia akhirnya memberikan gencatan senjata sampai Norgard pulih dan mendapatkan Pemimpin baru.

Beberapa bulan setelah perang itu, Zameckis yang merupakan sang jendral besar Almekia kemudian melakukan pemberontakan kepada King Henguist, dengan cepat dia dan pasukannya berhasil menghancurkan Ibukota Logres. Semua petinggi dan keluarga besar Almekia Terbunuh, tetapi sang putra Raja, pangeran Lance berhasil diselamatkan oleh Gereint. Mereka melarikan diri ke tepi barat, tepatnya kerajaan Padstow yang akhirnya menjadi Negara New Almekia dan dipimpin oleh Pangeran Lance sendiri.

Sejarah panjang Kerajaan terkuat Almekia akhirnya runtuh di tangan Zameckis dan pasukan pemberontaknya. Akhirnya Zameckis membuat sebuah kerajaan baru yang disebut dengan kekaisaran Esgares yang dipimpin sendiri olehnya. Dengan cepat Zameckis membuka perang ke semua negara. Disamping itu, Norgard telah memiliki pemimpin baru bernama White Wolf Vaynard. Begitulah game ini bermula.

D. Ruler Game

1. Lance (Prince) akan berubah menjadi (King) jika menaklukan Esgares Empire. Catatan: Lance adalah yang terlemah dari yang lainnya, tetapi akan menjadi paling kuat dari yang lainnya jika telah menjadi King.
2. Vaynard (Lord) Catatan: Seorang pemimpin yang Ambisius ingin menaklukan Forsena. Ia Menjadi seorang pemimpin Norgard setelah kematian Doremit.
3. Lyonesse (Queen) Catatan: Pemimpin dari negara yang religius, dalam petempuran ia sering menggunakan sihir putih.
4. Dryst (Tyrant) akan berubah menjadi (Super Tyrant) jika telah level 30. Catatan: Seorang Raja gila dan kuat, tetapi ia dikenal baik hati oleh para gadis di Iscalio.
5. Cai (Warlock) Catatan: Dapat disebut Penyihir terbaik di Forsena. Ia menggunakan sihir Merah, Biru dan Hijau.
6. Zameckis (Emperor) Catatan: Adalah mantan jendral almekia yang memberontak. Setelah mengalahkan Almekia, ia berhasrat menaklukan seluruh Forsena.

E. Gameplay

Pemilihan kerajaan / raja sangat penting pada awal permainan, karena dalam game ini setiap wilayah memiliki jenis *summon monster* yang berbeda, jenis *monster* akan mempengaruhi gaya bermain player, sehingga sebaiknya player harus mampu memilih negara sesuai gaya bermain yang tepat.



Gambar 1. Pemilihan Raja/Kerajaan

Setelah melakukan pemilihan raja, *player* akan melakukan penataan negara (dilakukan melalui *menu organize* pada *game*) dengan cara melakukan *summon* sesuai keuangan negara dan juga pemilihan *monster* yang cocok dan sesuai kuota untuk meng-invasi negara lain.



Gambar 2. Penataan Leader dan Monster

Setiap Negara akan memiliki beberapa *leader* (manusia) yang dapat mengontrol monster. Setiap *leader* memiliki kapasitas masing-masing dalam menampung monster. Setiap monster memiliki nilai berat tertentu dan ini tergantung kepada jenis monsternya. kapasitas terpakai/kapasitas maksimum *leader* ditandai oleh tulisan biru di sebelah kanan bar *leader* (lihat gambar 2) .

Setelah selesai melakukan penataan negara, kita dapat mengubah status permainan ke mode invasi, kemudian melakukan penyerangan ke negara lain. Pada mode pertarungan, permainan akan dilakukan dalam beberapa turn. Keunikan dari game ini adalah game akan dimenangkan ketika lawan kabur (*retreat*) atau semua *leader* lawan mati. Hal ini berarti kemenangan dapat dilakukan tidak harus dengan membunuh semua unit monster dari lawan.



Gambar 3. Mode Pertarungan Game

Sehingga pada mode pertarungan ini, tujuan utama *player* adalah membunuh semua *leader* lawan, dan melindungi semua *leader player*. Hal ini menggambarkan bahwa strategi *greedy* yang paling cocok dilakukan untuk melindungi *leader* sendiri dan membunuh *leader* lawan.

Setelah seluruh *leader* lawan telah mati / *retreat*, wilayah peta akan berubah warna (terlihat seperti gambar 4).



Gambar 4. Perubahan Warna Bendera Kerajaan

II. TEORI

A. Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang sangat erat kaitannya dengan persoalan optimasi. Persoalan optimasi merupakan persoalan yang menuntut pencarian solusi optimum. Optimum bukan berarti hanya maksimum, tetapi bisa juga berarti minimum. Solusi optimum merupakan solusi maksimum/minimum dari sekumpulan alternatif solusi yang mungkin. Sehingga algoritma optimasi sering digolongkan atas 2 masalah umum yaitu maksimum *problem* dan minimum *problem*.

Prinsip utama dari strategi *greedy* adalah “*take the best that you can get now*”, hal ini berarti algoritma akan memilih

pilihan terbaik yang tersedia pada kondisi saat itu. Prinsip ini sebenarnya berharap agar solusi-solusi optimum lokal yang diperoleh akan mengantarkan kita menuju solusi maksimum yang diharapkan pada akhirnya.

Pada kasus optimasi ini terdapat 2 elemen umum persoalan yaitu :

1. Kendala
2. Fungsi objektif / fungsi optimasi

Solusi yang dapat memenuhi semua kendala merupakan *solusi layak* dan Solusi layak yang mampu mengoptimalkan fungsi optimasi adalah solusi optimum.

Algoritma Greedy tersusun atas elemen elemen berikut :

1. Himpunan kandidat,
Himpunan ini merupakan himpunan yang berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi,
Himpunan ini merupakan himpunan yang berisi kandidat-kandidat terpilih sebagai solusi persoalan.
3. Fungsi Seleksi,
Fungsi ini merupakan fungsi yang memilih kandidat yang paling memungkinkan untuk mencapai solusi optimal. Sebagai keterangan, kandidat yang telah dipilih pada suatu langkah, tidak dipertimbangkan lagi pada langkah selanjutnya.
4. Fungsi Kelayakan,
Fungsi ini merupakan fungsi yang memeriksa apakah suatu kandidat yang telah diperoleh mampu memberikan solusi yang layak, yaitu kandidat tersebut bersama-sama dengan himpunan solusi yang sudah dibentuk tidak melanggar kendala yang ada. Kandidat yang layak kemudian dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak dipertimbangkan lagi.
5. Fungsi Obyektif,
Fungsi ini merupakan fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya berat, panjang lintasan, keuntungan, dan lainnya).

Setelah melihat penjelasan sebelumnya, dapat dijelaskan bahwa algoritma *greedy*:

1. Membuat himpunan kosong sebagai set solusi awal.
2. Melakukan pengujian pada salah satu anggota kandidat, kemudian melakukan penghapusan anggota kandidat dari himpunan kandidat jika tidak lolos melalui fungsi seleksi.

- Melakukan uji menggunakan fungsi kelayakan dan menambahkan ke himpunan solusi jika lolos dari fungsi kelayakan.

Hal tersebut dapat dijelaskan menggunakan pseudo-code sebagai berikut :

```
function greedy(input C: himpunan_kandidat) → himpunan_kandidat
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy
Masukan: himpunan_kandidat C
Keluaran: himpunan solusi yang bertipe himpunan_kandidat
}
Deklarasi
x : kandidat
S : himpunan_kandidat
Algoritma:
S ← [] { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {} ) do
x ← SELEKSI(C) { pilih sebuah kandidat dari C }
C ← C - {x} { elemen himpunan_kandidat berkurang satu }
if LAYAK(S ∪ {x}) then
S ← S ∪ {x}
endif
endwhile
{SOLUSI(S) or C = {} }
if SOLUSI(S) then
return S
else
write('tidak ada solusi')
endif
```

Gambar 5. Pseudo-Code Algoritma Greedy

(<http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2011-2012/Algoritma%20Greedy.ppt>)

Sekali lagi harus ditegaskan bahwa algoritma ini belum tentu mencapai solusi paling optimal yang diharapkan pada suatu persoalan. Meskipun hal ini merupakan salah satu kelemahan dari algoritma greedy tetapi algoritma ini masih tetap populer sebagai salah satu algoritma optimasi yang sederhana dan mudah digunakan.

III. ANALISIS

Pada game *Brigandine : The Legend of Forsena*, terdapat banyak sekali permasalahan yang dapat diselesaikan oleh algoritma ini, seperti memenangkan pertarungan dengan cara greedy membunuh *leader* dari lawan, pemilihan monster yang akan dibawa oleh seorang *leader* (mirip dengan *knapsack problem*), melakukan *summon monster* terbaik dengan total uang negara(*mana*) yang tersedia dan juga penataan *monster* sesuai lokasi wilayah yang cocok bagi mereka.



Gambar 6. Pseudo-Code Algoritma Greedy

Pertama, kita akan memiliki beberapa jenis monster dan juga leader seperti pada gambar 6. Status setiap leader dijelaskan sebagai berikut :

Leader :

- Leader1
Rune : 0/198
- Leader2
Rune : 0/180
- Leader3
Rune : 0/172

Monster (level 1) :

- LizardMan
Cost : 40
Str : 70
- Gryphon
Cost : 55
Str : 80
- Dragon
Cost : 85
Str : 90
- Hydra
Cost : 65
Str : 100
- ClayGolem
Cost : 45
Str : 83
- Mermaid
Cost : 35
Str : 46
- Centaur
Cost : 25
Str : 44
- HellHound
Cost : 50
Str : 65

Pada game ini, *rune* merupakan kapasitas yang mampu dibawa oleh seorang *leader*, *Str* adalah keuntungan dari monster, dan *cost* merupakan bobot berat monster yang akan ditanggung oleh seorang *leader*. Hal ini menunjukkan bahwa *leader* mirip sebagai sebuah tas penyimpanan dan monster mirip dengan benda-benda jika dilihat sebagai masalah *knapsack problem*.

Pada kasus ini, kita dapat melakukan greedy atas berbagai jenis. Pada kasus ini, kita akan melakukan greedy dengan memperhatikan *str/cost* dari setiap monster.

Setelah melihat permasalahan ini, kelompokkan semua menjadi elemen yang diperlukan dalam strategi greedy yaitu :

1. Himpunan Kandidat : semua monster yang dimiliki pada suatu kerajaan.
2. Himpunan Solusi : kelompok monster dengan bobot *cost* yang mampu ditampung oleh *leader*.
3. Fungsi Seleksi : memilih monster dengan bobot *str/cost* terbesar dan dapat ditampung oleh *leader*.
4. Fungsi Kelayakan : memastikan apakah monster masih dapat ditampung oleh *leader*, dimulai dengan bobot *str/cost monster* terbesar.
5. Fungsi Obyektif : mendapatkan *rune* paling maksimum dari seorang *leader* (total *cost monster* yang dibawa mendekati maksimum *rune* yang dimiliki oleh *leader* serta memiliki nilai *str* yang maksimum juga).

Setelah dilakukan pengujian, diperoleh himpunan solusi sebagai berikut (untuk ketiga Leader) :

Leader1 :
Dragon (Str : 90/ Cost :85)
Hydra (Str : 100/ Cost : 65)
ClayGolem (Str : 83/Cost: 45)
Total Str : 273
Rune : 195/198

Leader2 :
Dragon (Str : 90/Cost : 85)
Hydra (Str : 100 /Cost : 65)
Centaur (Str : 44/Cost :25)
Total Str : 234
Rune : 175/180

Leader3 :
Dragon (Str : 90/Cost : 85)
Hydra (Str : 100 /Cost : 65)
Total Str : 190
Rune : 150/172

Untuk memperoleh solusi berdasarkan keuntungan juga dapat dilakukan, dan metode yang dilakukan juga hampir sama, hanya saja pertimbangan seleksi dipengaruhi oleh level dari monster yang ada, umumnya monster dengan level lebih tinggi memiliki banyak keunggulan dibandingkan dengan monster dengan level yang lebih rendah.

Pada masalah biaya negara dan summon monster juga mengalami hal yang sama, hanya saja pada persoalan ini yang menjadi kapasitas tas nya ada uang negara (mata uangnya adalah *mana*), objeknya tetap monster tetapi bobot bendanya adalah *mana* yang diperlukan untuk *summon monster*. Hal tersebut tinggal diubah pada 5 elemen penting *greedy* pada persoalan sebelumnya, yaitu *rune* sama seperti *uang negara* dan *cost* sama seperti *mana summon monster*.

Strategi greedy bertahan untuk memenangkan pertarungan dilakukan dengan mengupayakan peletakan posisi *leader* yang selalu dikelilingi monsternya, sehingga menghilangkan *path* yang dapat dilalui monster lawan untuk menyerang *leader*. Fungsi Seleksinya adalah mencari *path* yang kosong di sekitar *leader*, jika ditemukan akan diletakkan monster disekitarnya.



Gambar 7. Penataan Monster disekitar Leader

Maka untuk strategi penyerangan dilakukan hal yang berkebalikan dengan strategi bertahan. Jika prioritas utama adalah untuk bertahan, Fungsi yang dipakai adalah ketika total bobot *str* lawan lebih besar dari dari total bobot *str player*, sebaiknya dilakukan *retreat* untuk mengurangi monster yang mati saat bertarung dan sebaliknya bertarung jika lebih besar.

IV. KESIMPULAN

Terdapat begitu banyak persoalan pada game ini yang mampu diselesaikan dengan menggunakan algoritma greedy. Berdasarkan uji coba yang telah dilakukan pada game *Brigandine : The Legend of Forsena*, dapat diperoleh kesimpulan bahwa persoalan *rune Leader* game ini sangat mirip dengan *knapsack problem* yang tentunya mampu diselesaikan dengan menggunakan algoritma *greedy*.

Permasalahan ini semakin mendekati solusi optimum ketika mempertimbangkan keuntungan(*str*) dari monster, sehingga memang sangat optimal untuk dipakai saat berada dalam mode pertarungan.

Penyelesaian persoalan ini mungkin memang optimal dari segi bobot monster, tetapi akan lebih baik jika persoalan diselesaikan dengan memperhatikan atribut lain yang dimiliki oleh monster dan leader terutama dari segi *level* dan *skill*.

Penyelesaian persoalan memenangkan pertarungan dengan cara greedy membunuh leader lawan mungkin bukanlah hal yang mudah dilakukan karena ketika tidak terdapat *path* untuk mendekati leader, kita harus membuka *path* tersebut dengan cara membunuh monster lawan.

Secara tidak lain permainan ini tetap memerlukan feeling dari player untuk memilih membunuh monster mana yang lebih prioritas untuk dibunuh dan tetap saja hal ini merupakan bagian dari persoalan yang dapat diselesaikan secara greedy.

REFERENSI

1. ."Monster". <brigandine.wikia.com/wiki/Monster> Diakses pada tanggal 18 Mei 2017 pukul 17.00.
2. ."Brigandine : The Legend of Forsena Remake". <https://www.kickstarter.com/projects/1799998113/compedium-of-war-brigandine-the-legend-of-forsena?ref=user_menu> Diakses pada tanggal 18 Mei 2017 pukul 18.00.
3. Munir Rinaldi . " IF2211 Strategi Algoritma - Semester II Tahun 2016/2017". <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2016-2017/stima16-17.htm> Diakses pada tanggal 18 Mei 2017 pukul 15.00.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Mei 2017



Bobby Indra Nainggolan