

# Implementasi Algoritma Greedy untuk Intelegensia Buatan pada MMORPG Berdasarkan Jumlah Kemunculan Status di Himpunan Solusi

Mukhammad Ifanto / 13508110  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
if18110@if.itb.ac.id

**Abstrak**—MMORPG (Massive Multiplayer Online Role Playing Game) merupakan jenis game yang termasuk dalam kategori Multiplayer Online dengan tema RPG (Role Playing Game). Multiplayer Online berarti game ini dapat dimainkan oleh lebih dari 2 orang. RPG berarti pemain memiliki peran sendiri dalam game. Di dunia MMORPG terdapat musuh yang disebut monster. Monster ini dapat diserang dan dapat menyerang balik. Sehingga di dalam monster tersebut diberikan intelegensia buatan yang akan merespon suatu aksi yang berhubungan dengan dirinya. Sebagai contoh, ketika satu monster diserang oleh beberapa pemain pada saat yang sama, monster akan mengambil keputusan pemain mana yang akan diserang karena monster tersebut hanya dapat melakukan serangan biasa ke 1 pemain. Bentuk pengambilan keputusan tersebut didasarkan pada algoritma greedy, di mana monster akan menghitung pemain mana yang berpeluang besar sebagai ancaman yang berbahaya bagi dirinya sekaligus pemain yang mudah dikalahkan sehingga jumlah ancaman terhadap monster tersebut dapat diminimalisir oleh monster dalam waktu yang singkat. Implementasi algoritma greedy ini menggunakan fungsi seleksi lebih dari satu kemudian hasil dari himpunan solusi tiap fungsi seleksi digabungkan untuk dilihat pemain mana yang paling sering dipilih oleh fungsi-fungsi seleksi tersebut sebagai solusi dari algoritma greedy (dengan kata lain sebagai keputusan yang diambil oleh monster untuk menyerang pemain tersebut). Tujuan penggunaan fungsi seleksi lebih dari satu adalah agar peluang solusi yang dihasilkan dari generalisasi fungsi-fungsi seleksi sebagai hasil optimum (optimum global sebenarnya) menjadi lebih besar.

**Kata Kunci**—MMORPG, intelegensia buatan, greedy, optimum global.

## I. PENDAHULUAN

Disaat kita berinteraksi dengan suatu tampilan visual elektronik dengan menggunakan alat untuk memasukkan perintah kita yang di timpali kembali oleh alat dan visual tersebut, dan didalamnya ada kegiatan dalam aturan tertentu dan hasilnya dapat membuat pemainnya terhibur, saat itulah kita bermain video game. Game itu sendiri terdiri atas bermacam-macam dalam berbagai kategori kelompok. Kategori pengelompokan game tersebut antara lain:

- Jenis platform atau alat yang digunakan.
- Genre permainan

- Serta kategori lainnya yang merupakan jenis-jenis game yang baru dan berkembang akhir-akhir ini.

### Berdasarkan jenis platform atau alat yang digunakan, game dibedakan atas:

- Arcade games**  
Game yang dimainkan di suatu box atau mesin yang khusus di desain untuk jenis video games tertentu. Selain itu, biasanya disediakan pula fitur tambahan yang dapat membuat pemainnya merasa lebih masuk ke game seperti pistol, kursi khusus, sensor gerak, sensor injak dan stir mobil.
- PC games**  
Video game yang dimainkan menggunakan Personal Computers.
- Console games**  
Video games yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
- Handheld games**  
Vide game yang dimainkan di console khusus video game yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
- Mobile games**  
Video game yang dimainkan atau khusus untuk mobile phone atau PDA.

### Berdasarkan genre permainannya, game terdiri atas:

- Aksi tembak menembak (shooting)
- Pertarungan (fighting)
- Aksi petualangan
- Simulasi konstruksi dan manajemen
- Role Playing
- Strategi
- Puzzle
- Simulasi kendaraan
- Olah raga

**Jenis game lainnya yang belum masuk dalam dua kategori tersebut antara lain:**

a. Multiplayer online

Game yang sedang populer akhir-akhir ini. Game ini dapat dimainkan secara bersamaan oleh lebih dari 2 orang (bahkan dapat mencapai puluhan ribu orang dalam satu waktu) sehingga membuat pemain dapat bermain bersama dalam satu dunia virtual mulai dari sekedar chatting hingga membunuh naga bersama teman yang entah bermain di mana. Umumnya permainan tipe ini dimainkan di PC dan bertema RPG, walau ada juga yang bertema music atau action.

b. Casual games

Game ini tidak kompleks, dimainkan dengan rileks dan sangat mudah untuk dipelajari (bahkan cenderung langsung bisa dimainkan). Genre permainannya biasanya puzzle atau action sederhana dan umumnya dapat dimainkan hanya menggunakan mouse.

c. Edugames

Video Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan, entah untuk belajar mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing.

MMORPG (Massive Multiplayer Online Role Playing Game) merupakan kategori game Multiplayer Online dengan tema RPG (Role Playing Game). Contoh MMORPG antara lain:

- a. Ragnarok Online
- b. Seal Online
- c. World of Warcraft
- d. Perfect World Online

Role Playing Game itu sendiri merupakan tipe permainan di mana pemain diharuskan untuk memainkan karakter dengan peran tertentu melalui permainan peran. Biasanya untuk PC Games bertema RPG, terdapat jalan cerita mulai dari awal cerita hingga cerita berakhir atau tamat. Akan tetapi, untuk MMORPG, pemain disediakan suatu jalan cerita yang sepertinya tidak mempunyai ujung. Misi utama pada MMORPG adalah membuat karakter tersebut menjadi semakin kuat daripada karakter lainnya.

Salah satu cara untuk menjadi kuat adalah dengan menambah level melalui berburu dan membunuh musuh. Musuh dalam MMORPG biasanya berupa monster-monster dalam berbagai bentuk, rupa dan ukuran. Monster tersebut akan menyerang pemainnya juga ketika mereka merasa terancam. Seperti mendapat serangan dari pemain atau merasakan ancaman hanya dengan kehadiran pemain didekatnya.

Ketika seorang pemain menyerang monster, monster tersebut juga akan balik menyerang. Ketika seorang pemain lari dari monster yang diserangnya, monster tersebut akan berlari mengejar pemain selama masih dalam jarak pandangannya. Ketika monster merasakan sebuah ancaman melalui kehadiran pemain di dekatnya, monster tersebut akan langsung menyerang pemain tanpa

peringatan terlebih dahulu. Ketika monster diserang oleh beberapa pemain, monster akan mengambil keputusan untuk melakukan sebuah aksi seperti menyerang pemain yang di anggap berbahaya atau mengeluarkan kemampuan khususnya. Kecerdasan-kecerdasan tentang pertahanan dan serangan diberikan ke monster tersebut untuk memberikan suasana game yang lebih menarik.

Salah satu teori yang digunakan untuk mengaplikasikan kecerdasan tersebut adalah algoritma greedy. Melalui algoritma ini, monster dapat mengambil keputusan hendak menyerang pemain yang mana ketika sedang diserang oleh banyak pemain. Monster melakukan perhitungan dengan faktor-faktor tertentu untuk mengetahui pemain mana yang layak diserang, kemudian mengambil keputusan berdasarkan hitungan tersebut untuk menyerang pemain bersangkutan.

Berikut ini merupakan gambar-gambar yang merepresentasikan monster dan pemain yang ada dalam permainan MMORPG.



**Gambar 1 Pertarungan dengan boss monster di Granado Espada Online**



**Gambar 2 Pertarungan dengan boss monster Barrones Retribution di Ragnarok Online**



**Gambar 3 Pertarungan dengan boss monster Valkyrie Randgriz di Ragnarok Online**

## II. ALGORITMA GREEDY

Pada kasus kecerdasan monster tersebut, monster diharapkan memperoleh keputusan yang merupakan solusi optimum yang merupakan solusi terbaik. Solusi terbaik adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Pada pencarian solusi optimum tersebut, diberikan sejumlah kendala (constraint) dan fungsi optimasi. Solusi yang memenuhi semua kendala disebut sebagai solusi layak (feasible solution). Sedangkan solusi layak yang mengoptimalkan fungsi optimasi disebut sebagai solusi optimum.

Algoritma greedy merupakan salah satu metode untuk memecahkan persoalan optimasi tersebut. Algoritma ini bersifat sederhana dan lempang (straightforward). Secara harfiah, greedy berarti rakus atau tamak. Prinsip greedy adalah: "take what you can get now". Ambil apa yang dapat diperoleh sekarang. Prinsip tersebut diadopsi dalam pemecahan masalah optimasi.

Algoritma greedy akan membentuk solusi langkah per langkah (step by step). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Pendekatan yang dilakukan di dalam algoritma greedy adalah membuat pilihan yang "tampaknya" memberikan perolehan terbaik, yaitu dengan membuat pilihan optimum lokal (optimum local) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi optimum global (global optimum).

Secara ringkas, algoritma greedy adalah algoritma yang memecahkan masalah langkah per langkah, di mana pada setiap langkah:

Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip "take what you can get now!"). Asumsi bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Dalam pemecahan masalah optimasi, algoritma greedy disusun oleh elemen-elemen sebagai berikut:

- Himpunan kandidat,  $C$   
Himpunan ini berisi elemen-elemen pembentuk solusi. Contohnya adalah himpunan koin, himpunan simpul di dalam graf.
- Himpunan solusi,  $S$ .  
Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi merupakan himpunan bagian dari himpunan kandidat.
- Fungsi seleksi, dinyatakan dengan predikat SELEKSI  
Fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Biasanya setiap kandidat,  $x$ , di-assign sebagai sebuah nilai numerik, dan fungsi seleksi memilih  $x$  yang mempunyai nilai terbesar atau memilih  $x$  yang mempunyai nilai terkecil.
- Fungsi kelayakan (feasible), dinyatakan dengan predikat LAYAK  
Fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (constraint) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
- Fungsi obyektif  
Fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Persoalan optimasi yang diselesaikan dengan algoritma greedy melibatkan pencarian sebuah himpunan bagian,  $S$ , dari himpunan kandidat,  $C$ ; di mana  $S$  harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan  $S$  dioptimasi oleh fungsi obyektif. Optimasi tersebut dapat berarti minimisasi atau maksimasi, bergantung pada persoalan yang dipecahkan.

Optimum global yang dihasilkan dari algoritma greedy belum tentu merupakan solusi optimum (terbaik), tetapi dapat merupakan solusi sub-optimum atau pseudo-optimum. Hal ini dikarenakan oleh dua faktor berikut:

Algoritma greedy tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada (seperti pada metode exhaustive search)

Pemilihan fungsi SELEKSI: fungsi SELEKSI biasanya didasarkan pada fungsi obyektif.

Dengan kata lain, algoritma greedy tidak selalu menghasilkan solusi yang benar-benar optimum. Algoritma greedy sering menjadi basis untuk pendekatan heuristik.

Jika persoalan dapat dipecahkan secara eksak (memberikan solusi yang optimum) dengan algoritma greedy, maka pembuktian kebenaran dari algoritma greedy tersebut dapat dianggap sebagai sebuah proses yang non-trivial.

### III. IMPLEMENTASI ALGORITMA GREEDY DALAM PERMAINAN MMORPG

#### A. Atribut-atribut pada Permainan

Algoritma greedy tersebut dapat diimplementasikan pada permainan MMORPG untuk memberikan monster suatu kecerdasan buatan sehingga monster dapat mengambil sebuah keputusan terhadap suatu kasus. Kasus yang mengimplementasikan algoritma greedy tersebut adalah ketika satu monster di serang oleh banyak pemain sekaligus. Monster hanya dapat memberikan serangan biasa ke satu orang pemain. Oleh karena itu, monster harus menentukan pilihan pemain mana yang hendak diserang. Tujuan dari serangan balasan tersebut adalah sebagai bentuk pertahanan dari monster. Setidaknya monster dapat bertahan lebih lama dari serangan-serangan yang diberikan atau bahkan monster tersebut dapat mengungguli pemain-pemain yang menyerangnya dengan membuat pemain tersebut KO (Knock Out) semua.

Dalam pengambilan keputusan tersebut, dibutuhkan atribut-atribut yang berkaitan dengan pemain dan monster sebagai faktor penentu dalam elemen-elemen fungsi algoritma greedy. Faktor-faktor tersebut antara lain:

##### a. HP (Health Point)

Merupakan nyawa dari pemain atau monster yang direpresentasikan dalam angka. Bilamana HP pemain mencapai 0, berarti pemain tersebut dianggap KO (Knock Out)

##### b. Defense (Pertahanan)

Merupakan tingkat pertahanan dari pemain atau monster terhadap suatu serangan. Semakin besar tingkat pertahanannya semakin kecil serangan yang diterimanya, dan sebaliknya.

##### c. Attack (serangan)

Merupakan tingkat kerusakan yang diberikan oleh pemain ke target. Serangan ini akan mengurangi HP target. Atribut yang dapat mempengaruhi serangan adalah defense.

##### d. Distance (Jarak)

Merupakan jarak antara pemain dengan target

Monster akan mencatat setiap atribut-atribut tersebut beserta ID dari pemain yang bersangkutan.

#### B. Penentuan Elemen-elemen

Ketika monster diserang oleh seorang pemain, monster akan mencatat ID dari penyerang tersebut kemudian mengakses informasi yang dibutuhkan untuk mengambil suatu keputusan. Sehingga, bila pemain ada dalam jarak pandang monster tetapi tidak melakukan serangan ke monster, keberadaan pemain tersebut tidak dianggap oleh monster. Bila semua pemain yang menyerang monster

KO, monster akan mengubah target ke setiap pemain yang ada di jarak pandangnya karena merupakan sebuah ancaman bagi monster tersebut.

Elemen-elemen pada kasus pertahanan dari multi-attack tersebut dalam konteks algoritma greedy terdiri atas:

- Himpunan kandidat, C.
- Himpunan ID pemain yang menyerang monster.
- Himpunan bagian dari C, S, adalah solusi berdasarkan fungsi SELEKSI yang digunakan.
- S disebut layak jika memenuhi atau tidak melanggar dari kendala (constraint) fungsi SELEKSI
- Fungsi obyektif akan mencari nilai minimal atau nilai maksimal bergantung dari atribut fungsi SELEKSI yang digunakan untuk perhitungan.
- Fungsi SELEKSI memilih kandidat (yaitu ID pemain) yang nilainya maksimal/minimal bergantung dari atribut yang digunakan dari perhitungan. Keseluruhan fungsi seleksi digunakan dalam perhitungan kemudian dilihat ID yang paling sering menjadi target untuk setiap fungsi seleksi akan dipilih sebagai target utama.

#### C. Fungsi Seleksi

Fungsi seleksi digunakan untuk memilih kandidat mana yang memungkinkan untuk mencapai solusi optimal. Pada kasus ini (monster yang diserang oleh banyak pemain), dibuat beberapa fungsi seleksi untuk berbagai macam kondisi sehingga kecerdasan yang dimiliki oleh monster menjadi semakin lebih baik. Fungsi seleksi tersebut di tetapkan berdasarkan informasi atribut yang dicatat oleh monster. Fungsi seleksi tersebut antara lain:

##### 1. Fungsi berdasarkan 1 atribut.

###### a. Berdasarkan jumlah HP penyerang.

Dengan persamaan:  $\%HP = \frac{HP_{max} - HP_{current}}{HP_{max}} \times 100\%$

Monster akan mencari pemain yang memiliki jumlah HP paling sedikit. Berdasarkan hal tersebut, monster menganggap, pemain dengan HP paling sedikit dapat segera di bereskan sehingga jumlah ancaman ke monster tersebut berkurang. (dicari nilai yang paling kecil dari persamaan, minimasi)

###### b. Berdasarkan tingkat pertahanan penyerang

Dengan persamaan:  $\frac{1}{Defense}$

Monster akan mencari pemain yang memiliki tingkat pertahanan paling kecil. Berdasarkan hal tersebut, monster menganggap, pemain dengan tingkat pertahanan paling kecil akan mendapat serangan dalam jumlah yang cukup besar dari monster tersebut. Sehingga, pemain tersebut dapat segera di bereskan dan jumlah ancaman ke monster tersebut berkurang. (dicari nilai yang paling besar dari persamaan, maksimasi atau nilai defense yang paling kecil, minimasi)

- c. Berdasarkan besarnya serangan yang diterima monster dari penyerang.

Dengan persamaan: Serangan

Monster akan mencari pemain yang memberikan tingkat serangan paling besar ke monster tersebut. Monster menganggap pemain dengan tingkat kerusakan yang ditimbulkan paling besar merupakan sebuah ancaman berbahaya bagi dirinya sehingga harus segera dibersihkan. (dicari nilai yang paling besar dari persamaan, maksimasi)

## 2. Fungsi berdasarkan kombinasi 2 atribut.

- a. Berdasarkan jumlah HP penyerang dan tingkat pertahanan penyerang.

Dengan persamaan:  $\text{Defense} \times \% \text{HP}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki defense dan HP yang rendah. (dicari nilai yang paling kecil dari persamaan, minimasi)

- b. Berdasarkan jumlah HP penyerang dan besarnya serangan yang diberikan penyerang

Dengan persamaan:  $\frac{\text{Serangan}}{\% \text{HP}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan HP yang rendah. (dicari nilai yang paling besar dari persamaan, maksimasi)

- c. Berdasarkan tingkat pertahanan penyerang dan besarnya serangan yang diberikan.

Dengan persamaan:  $\frac{\text{Serangan}}{\text{Defense}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan Defense yang rendah. (dicari nilai yang paling besar dari persamaan, maksimasi)

- d. Berdasarkan tingkat serangan/kerusakan yang ditimbulkan dan jarak penyerang.

Dengan persamaan:  $\frac{\text{Serangan}}{\text{Jarak}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan berada pada jarak yang lebih dekat ke monster. (dicari nilai yang paling besar dari persamaan, maksimasi)

## 3. Fungsi berdasarkan kombinasi 3 atribut.

- a. Berdasarkan HP penyerang, tingkat pertahanan penyerang dan besarnya serangan yang diberikan.

Dengan persamaan:  $\frac{\text{Serangan}}{\% \text{HP} * \text{Defense}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan memiliki jumlah HP sedikit dan tingkat pertahanan yang kecil. (dicari nilai yang paling besar dari persamaan, maksimasi)

- b. Berdasarkan tingkat pertahanan penyerang, besarnya serangan yang diberikan pemain dan

jarak penyerang dari monster.

Dengan persamaan:  $\frac{\text{Serangan}}{\text{Jarak} * \text{Defense}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan memiliki tingkat pertahanan yang kecil dan jarak yang lebih dekat ke monster. (dicari nilai yang paling besar dari persamaan, maksimasi)

## 4. Fungsi berdasarkan kombinasi 4 atribut

- a. Berdasarkan HP penyerang, tingkat pertahanan penyerang, besarnya serangan yang diberikan pemain dan jarak penyerang dari monster.

Dengan persamaan:  $\frac{\text{Serangan}}{\text{Jarak} * \text{Defense} * \% \text{HP}}$

Semakin besar nilai tersebut, menandakan bahwa kemungkinan pemain memiliki serangan yang tinggi dan memiliki jumlah HP sedikit, tingkat pertahanan yang kecil dan jarak yang lebih dekat ke monster. (dicari nilai yang paling besar dari persamaan, maksimasi)

## 5. Fungsi yang tidak berdasarkan atribut

- a. Berdasarkan pemain yang memberikan serangan pertama ke monster.

Fungsi seleksi ini khusus untuk menangani kondisi ketika monster mendapat serangan pertama kali. Monster akan menyerang pemain yang menyerangnya pertama kali. Prioritas dari pemain tersebut diutamakan terlebih dahulu.

Pada awal/inisiasi penyerangan, monster akan menyerang pemain yang menyerangnya pertama kali. Setiap periode tertentu, monster akan melakukan kalkulasi ulang untuk melihat apakah ada target baru yang lebih optimal berdasarkan fungsi yang digunakan tersebut. Setelah monster menetapkan targetnya, monster akan menyerang target tersebut selama beberapa satuan waktu, hingga diperoleh target baru. Fungsi seleksi yang digunakan dikombinasikan satu sama lain pada suatu periode berdasarkan kondisi monster tersebut. Kemudian dihitung kemunculan ID pemain dalam himpunan solusi untuk setiap fungsi seleksi. Pemain yang paling sering muncul akan dijadikan sebagai target oleh monster.

## IV. CONTOH IMPLEMENTASI (STUDI KASUS)

Suatu ketika monster 123, diserang oleh 5 orang pemain A, B, C, dan D. Pada inisialisasi penyerangan, urutan serangan yang masuk dari yang pertama kali hingga yang terakhir secara berurutan adalah C – D – A – B. Setiap periode 3 s, monster melakukan update atribut yang merupakan informasi dari pemain-pemain tersebut. Sebagai contoh, ketika waktu : 3 s, monster memperoleh informasi:

ID	%HP	Defense	Serangan Total	Jarak
A	50	200	1000	1
B	20	100	1300	3
C	80	150	600	2
D	40	80	900	1

(ID C yang menyerang pertama kali masih bertahan setelah 3 detik berjalan)

Himpunan Kandidat = {A, B, C, D}.

Kemudian di lakukan analisa untuk semua fungsi seleksi.

A. Fungsi berdasarkan 1 atribut

Tipe yang dipilih	ID yang menjadi target (Himpunan Solusi)
%HP	{B}
Defense	{D}
Serangan Total	{B}

B. Fungsi berdasarkan 2 atribut

- a. Berdasarkan jumlah HP penyerang dan tingkat pertahanan penyerang.

Persamaan: Defense x %HP (minimasi)

ID	Proses Kalkulasi	Hasil
A	200 x 50	10,000
B	20 x 100	2,000
C	80 x 150	12,000
D	40 x 80	3,200

Himpunan Solusi : {B}

- b. Berdasarkan jumlah HP penyerang dan besarnya serangan yang diberikan penyerang

Persamaan:  $\frac{\text{Serangan}}{\%HP}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{50}$	20
B	$\frac{1300}{20}$	65
C	$\frac{600}{80}$	7.5
D	$\frac{900}{40}$	22.5

Himpunan Solusi : {B}

- c. Berdasarkan tingkat pertahanan penyerang dan besarnya serangan yang diberikan.

Persamaan:  $\frac{\text{Serangan}}{\text{Defense}}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{200}$	5
B	$\frac{1300}{100}$	13
C	$\frac{600}{150}$	4
D	$\frac{900}{80}$	11.25

Himpunan Solusi : {B}

- d. Berdasarkan tingkat serangan/kerusakan yang ditimbulkan dan jarak penyerang.

Persamaan:  $\frac{\text{Serangan}}{\text{Jarak}}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{1}$	1000
B	$\frac{1300}{3}$	433.3
C	$\frac{600}{2}$	300
D	$\frac{900}{1}$	900

Himpunan Solusi : {A}

C. Fungsi berdasarkan 3 atribut

- a. Berdasarkan HP penyerang, tingkat pertahanan penyerang dan besarnya serangan yang diberikan.

Persamaan:  $\frac{\text{Serangan}}{\%HP * \text{Defense}}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{50 * 200}$	0.1
B	$\frac{1300}{20 * 100}$	0.65
C	$\frac{600}{80 * 150}$	0.05
D	$\frac{900}{40 * 80}$	0.28

Himpunan Solusi : {B}

- b. Berdasarkan tingkat pertahanan penyerang, besarnya serangan yang diberikan pemain dan jarak penyerang dari monster.

Persamaan:  $\frac{\text{Serangan}}{\text{Jarak} * \text{Defense}}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{1 \times 200}$	5
B	$\frac{1300}{3 \times 100}$	4.33
C	$\frac{600}{2 \times 150}$	2
D	$\frac{900}{1 \times 80}$	11.25

Himpunan Solusi : {D}

#### D. Fungsi berdasarkan 4 atribut

- Berdasarkan HP penyerang, tingkat pertahanan penyerang, besarnya serangan yang diberikan pemain dan jarak penyerang dari monster.

Persamaan:  $\frac{\text{Serangan}}{\text{Jarak} * \text{Defense} * \%HP}$  (maksimasi)

ID	Proses Kalkulasi	Hasil
A	$\frac{1000}{1 \times 200 \times 50}$	0.1
B	$\frac{1300}{3 \times 100 \times 20}$	0.22
C	$\frac{600}{2 \times 150 \times 80}$	0.025
D	$\frac{900}{1 \times 80 \times 40}$	0.28

Himpunan Solusi : {D}

#### E. Fungsi berdasarkan penyerang pertama

Setelah periode pertama (inisiasi) selesai, fungsi ini tidak digunakan lagi karena fungsi ini digunakan saat monster mendapat serangan pertama kali dari pemain. Kemudian setelah  $t > 0$  s dan jumlah pemain lebih dari 1, maka dilakukan pengambilan greedy dengan fungsi seleksi tertentu.

#### F. Total kemunculan ID tertentu berdasarkan semua fungsi seleksi yang ada

Hasil dari himpunan solusi untuk setiap fungsi seleksi digabungkan sehingga diperoleh

Himpunan solusi global : {{B}, {D}, {B}, {B}, {B}, {B}, {A}, {B}, {D}, {D}}

ID	Jumlah kemunculan
A	1
B	6
C	0
D	3

#### G. Pengambilan Keputusan

Pada inisiasi karena yang menyerang terlebih dahulu adalah ID C, maka pemain C akan diserang terlebih dahulu selama 3 detik. Kemudian setelah 3 detik, monster akan mengakses informasi dari pemain yang

menyerangnya yaitu pemain A, B, C dan D (pemain C masih bertahan). Kemudian dilakukan perhitungan untuk semua fungsi seleksi yang ada berdasarkan atribut saat itu. Secara berurutan dari yang paling besar, diperoleh bahwa pemain ID B dipilih oleh fungsi-fungsi seleksi sebanyak 6 kali, D sebanyak 3 kali, A sebanyak 1 kali dan C tidak dipilih. Oleh karena itu, monster akan mengubah targetnya ke pemain B karena dianggap yang paling mudah dikalahkan dan berbahaya bagi monster tersebut. Selanjutnya setelah 3 detik berikutnya, dilakukan perhitungan lagi. Bilamana sudah ada pemain yang KO (terbunuh) oleh monster tersebut, ID pemain dihapus dari tabel dan tidak dimasukkan dalam perhitungan selanjutnya.

Tujuan dilakukan perhitungan dengan banyak fungsi seleksi kemudian dihitung jumlahnya dijadikan target atau pilihan adalah untuk mengoptimalkan hasil dari optimum global. Bilamana hanya menggunakan 1 fungsi seleksi saja, hasil optimum global tersebut masih belum tentu merupakan solusi terbaik.

## V. KESIMPULAN

Hasil dari optimum global pada algoritma greedy dapat dioptimalkan dengan menggunakan fungsi seleksi yang banyak. Kemudian dihitung kemunculan setiap status dari seluruh fungsi seleksi tersebut. Maksud dari kemunculan tersebut adalah ketika status tersebut dipilih sebagai nilai optimum untuk fungsi seleksi tertentu. Nilai kemunculan untuk status tersebut di-increment (di tambah). Status yang memiliki nilai kemunculan paling besar menandakan bahwa status tersebut berpeluang besar untuk menjadi solusi optimum global.

Meskipun menggunakan banyak fungsi seleksi, hasil optimum global dari keseluruhan fungsi tersebut juga belum tentu merupakan optimum global yang sebenarnya. Akan tetapi, dengan pendekatan seperti ini peluang sebuah status menjadi optimum global yang sebenarnya lebih besar daripada hanya dengan menggunakan satu fungsi seleksi.

## REFERENSI

- Munir, Rinaldi, *Diklat Kuliah IF3051 Strategi Algoritma*. Bandung: STEI ITB.  
<http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/Greedy/greedyIntro.htm>, waktu akses: Rabu, 8 Desember 2010, Pukul 20.00  
<http://www.wordiq.com/definition/MMORPG>, waktu akses: Rabu, 8 Desember 2010, Pukul 20.06  
[http://www.wordiq.com/definition/Role-playing\\_game](http://www.wordiq.com/definition/Role-playing_game), waktu akses: Rabu, 8 Desember 2010, Pukul 20.07  
<http://www.gamexeon.com/forum/console-gaming/57020-jenis-jenis-game-sekitar-kita.html>, waktu akses: Rabu, 8 Desember 2010, Pukul 20.13  
<http://andretobb.wordpress.com/teknologi-game-mmorpg/>, waktu akses: Rabu, 8 Desember 2010, Pukul 20.19  
<http://www.lintasberita.com/go/996128>, waktu akses: Rabu, 8 Desember 2010, Pukul 20.23  
<http://img.youtube.com/vi/SH-fOdy6Fuw/0.jpg>, waktu akses: Rabu, 9 Desember 2010, Pukul 02.17

<http://biniku.files.wordpress.com/2007/03/earthq.jpg>, waktu akses:  
Rabu, 9 Desember 2010, Pukul 02.23  
<http://forum.lytogame.com/showthread.php?t=5079>, waktu akses: Rabu,  
9 Desember 2010, Pukul 02.29

#### 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, 29 April 2010



Mukhammad Ifanto / 13508110