

Penerapan Algoritma *Greedy* pada Artificial Intelligence dalam Permainan Defence of the Ancient

Gurun Nevada Dharan/ 13509076
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia
13509076@std.stei.ac.id

Abstract— Algoritma *Greedy* adalah algoritma yang mengikuti pemecahan masalah heuristik dengan cara membuat pilihan optimal lokal dengan mengharapkan menemukan optimum global.

Pada pengaplikasiannya, algoritma *Greedy* digunakan di banyak cabang masalah kehidupan. Salah satunya adalah digunakan di dalam permainan DotA sebagai AI.

Penggunaan algoritma *Greedy* pada AI DotA ini memiliki dampak yang lumayan baik, walaupun masih terdapat kekurangan. Dengan penggunaannya yang dibarengi dengan algoritma lain, menyebabkan tingkat kepintaran AI yang semakin tinggi dan semakin mempercepat komputasi pada sistem permainan.

Index Terms—Algoritma, *Greedy*, AI, DotA.

I. PENDAHULUAN

I.1 Latar Belakang

Strategi Algoritma digunakan dalam berbagai permasalahan dalam kehidupan. Tentu saja tujuan utama digunakan strategi algoritma adalah untuk memberikan solusi terbaik agar mempermudah kehidupan. Dalam hal ini permainan komputer merupakan salah satu bagian yang tidak mungkin dapat lepas dari penggunaan strategi algoritma. Karena dengan menggunakan strategi algoritma inilah dapat dibuatnya AI atau lawan komputer yang cerdas yang menyebabkan permainan semakin asyik untuk dimainkan.

I.2 Tujuan

Makalah Penerapan Algoritma *Greedy* pada Artificial Intelligence dalam Permainan Defence of the Ancient memiliki tujuan :

1. Bagi penulis :
 - a. Memenuhi tugas makalah IF3051 tahun ajaran 2011/2012.
 - b. Menambah pemahaman penulis secara pribadi tentang pentingnya mempelajari Strategi Algoritma.
 - c. Memahami pentingnya algoritma *Greedy* dalam mencari solusi maksimum.
2. Bagi pembaca :
 - a. Membuka mata pembaca tentang begitu banyaknya penerapan Strategi Algoritma di

dalam kehidupan sehari-hari dan di sekitar kita.

- b. Menunjukkan bahwa teori dasar Strategi Algoritma banyak digunakan dalam permainan Defence of the Ancients, apalagi pengembangannya.
- c. Menunjukkan bahwa Strategi Algoritma sangat membantu di dalam kehidupan nyata, dengan menggunakan permainan sebagai perwakilan dari kehidupan nyata.

II. LANDASAN TEORI

II.1 Defence of the Ancients

Defence of the Ancients (atau disingkat DotA) adalah sebuah peta buatan untuk permainan komputer buatan Blizzard berjudul *Warcraft III : Frozen Throne*, yang dibuat berdasarkan map “*Aeon of Strife*” dari permainan Blizzard lainnya, *Starcraft*.

Tujuan utama permainan ini adalah untuk menghancurkan “*Ancient*” musuh, sebuah struktur yang dijaga ketat di pojok kiri bawah atau (yang satu untuk sisi sendiri dan yang satu lagi untuk sisi musuh). Kedua sisi yang bertarung adalah *The Sentinel* dan *The Scourge*.

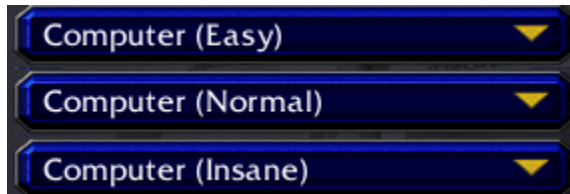
Permainan dapat menggunakan tokoh kuat yang disebut pahlawan dan dibantu oleh pasukan sekutu dan anak buah petarung yang dikontrol oleh *Artificial Intelligence*. Seperti dalam permainan peran lainnya, pemain dapat meningkatkan *level* tokoh pahlawan mereka dan menggunakan emas untuk membeli perlengkapan selama permainan,

DotA dapat dimainkan dengan sistem pemain lawan pemain dan pemain lawan *Artificial Intelligence*.

II.2 Artificial Intelligence

Artificial Intelligence (atau kecerdasan buatan) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini dianggap pada umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer, dan robotika. Pada makalah ini, merupakan penggunaan kecerdasan buatan dalam bidang permainan komputer

yaitu permainan *Warcraft III : Frozen Throne, Defence of the Ancient*.



Gambar 1 : Tingkat kesukaran AI

Dalam membuat makalah ini digunakan tingkat kesukaran AI Insane atau tingkat kesukaran AI paling sukar (paling pintar). Dan AI pada permainan DotA ini merupakan buatan fans, bukan buatan pengembang. Sehingga masih terdapat banyak kekurangan. Akan tetapi pengembangan AI dari hari ke hari semakin mendekati cara permainan manusia, walaupun unsur kerja sama yang ada pada manusia belum dapat diterapkan di AI pada permainan ini.

II.3 Algoritma Greedy

Algoritma *Greedy* adalah algoritma yang mengikuti pemecahan masalah heuristik dengan cara membuat pilihan optimal lokal dengan mengharapkan menemukan optimum global.

Elemen-elemen algoritma Greedy:

1. Himpunan kandidat, C .
2. Himpunan solusi, S
3. Fungsi seleksi (*selection function*)
4. Fungsi kelayakan (*feasible*)
5. Fungsi obyektif

Dengan kata lain, algoritma *Greedy* melibatkan pencarian sebuah himpunan bagian, S , dari himpunan kandidat, C , yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif.

Secara umum skema algoritma *Greedy* adalah 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
```

Pada algoritma *Greedy*, optimum global yang ditemukan belum tentu merupakan solusi terbaik karena :

1. Algoritma *Greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada (sebagaimana pada metode *exhaustive search*).
2. Terdapat beberapa fungsi seleksi yang berbeda, sehingga kita harus memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi optimal.

Jadi, pada sebagian masalah, algoritma *Greedy* tidak selalu berhasil memberikan solusi yang optimal.

III. PENGGUNAAN ALGORITMA GREEDY PADA ARTIFICIAL INTELLIGENCE DALAM PERMAINAN DEFENCE OF THE ANCIENTS

III.1 Membeli Senjata

Pahlawan dapat membeli senjata untuk digunakan dalam pertempuran. AI dalam melakukan pembelian barang menggunakan algoritma *Greedy*. Dengan uang yang dimiliki oleh AI, AI akan membeli senjata termurah yang dapat dibeli dengan uang yang dimilikinya, akan tetapi pembelian barang dan senjata tidak sembarangan karena setiap AI sudah memiliki daftar senjata yang harus dibeli untuk pahlawan yang digunakannya.

Dalam melakukan pembelian senjata AI memiliki perhitungan prioritas sebagai berikut :

- Senjata yang sudah diprogram untuk dibeli oleh AI, dapat terdiri dari :
 - Senjata bahan
 - Resep
- Senjata penambah status sementara
- Obat-obatan

Kelemahan dari algoritma ini dalam penggunaannya untuk melakukan pembelian senjata, AI terkadang membeli resep (resep digunakan untuk menyatukan beberapa senjata menjadi senjata yang lebih baik) terlebih dahulu dibandingkan dengan membeli bahannya karena harga resep yang lebih murah dibandingkan dengan harga bahan senjata.

Hal ini tentu merugikan bagi AI karena resep tidak memberikan tambahan status kepada pahlawan. Berbeda jika AI membeli bahan senjata terlebih dahulu dibandingkan dengan membeli resep maka pahlawan AI akan mendapatkan tambahan status untuk peperangan selama mengumpulkan uang untuk membeli resep.

Kelemahan lain dari algoritma *Greedy* dalam penggunaannya untuk pembelian senjata adalah ketika AI tidak memiliki uang yang cukup untuk membeli bahan senjata untuk membuat senjata, AI akan membeli barang-barang yang tidak terlalu diperlukan sehingga pembuatan senjata berlangsung lama.

Oleh karena itu kemungkinan senjata yang akan dibeli oleh AI tidak dapat terbeli karena uang yang dimiliki oleh AI habis terpakai untuk membeli obat-obatan dan barang lainnya yang tidak terlalu dibutuhkan ketika permainan sudah memasuki masa pertengahan.

Akan tetapi, semuanya kembali tergantung pada kondisi AI saat itu, ketika berada di kondisi kalah, dimana biasanya uang yang dimiliki sedikit, AI tidak akan mampu membeli senjata yang diinginkannya, sehingga AI hanya membeli obat-obatan. Yang mana hal ini akan menyebabkan AI tidak dapat menabung untuk senjata yang seharusnya AI beli. Akan tetapi dari segi lain, dengan menghabiskan uang untuk pembelian senjata dan obat-obatan, AI memanfaatkan uangnya dengan baik tanpa kehilangan uang ketika pahlawannya dihancurkan oleh musuh.



Gambar 2: AI membeli barang dan senjata dengan algoritma *Greedy*

Dari gambar 2, yang merupakan perbandingan senjata yang dimiliki oleh AI pada awal dan pertengahan permainan saja dapat dianalisis bahwa algoritma *Greedy* pada AI menyebabkan efektifitas dalam penggunaan uang, tetapi tidak berani mengambil resiko untuk menabung sampai senjata yang benar-benar diinginkan dapat dibeli tanpa harus membeli obat-obatan dan senjata penambah status kecil-kecilan.

III.2 Menghancurkan Anak Buah Petarung

Dalam menghancurkan anak buah petarung, AI menggunakan algoritma *Greedy*. AI akan menghancurkan anak buah petarung yang memiliki darah tersedikit dan berada di sekitarnya. Dalam melakukan penghancuran anak buah petarung ini, AI memiliki tiga perhitungan prioritas yaitu:

- Anak buah petarung teman.
- Anak buah petarung musuh.
- Anak buah petarung terdekat.

Untuk jenis pahlawan bertipe kecepatan biasanya prioritas menghancurkan anak buah petarung musuh berada di atas prioritas menghancurkan anak buah petarung teman. Dan untuk pahlawan bertipe kepintaran dan kekuatan biasanya prioritas menghancurkan anak buah petarung teman berada di atas prioritas menghancurkan anak buah petarung musuh.

Jadi jika terdapat anak buah petarung yang sekarat di sekitarnya, AI akan menghancurkannya. Algoritma ini cukup efektif digunakan untuk hal ini karena AI akan bergerak efektif yaitu bergerak hanya ketika terdapat anak buah petarung yang sekarat. Sehingga di waktu selain itu AI dapat melakukan hal lain. Sebelum menggunakan algoritma ini, pahlawan AI akan menyerang anak buah petarung secara acak satu-persatu yang menyebabkan

pahlawan AI terpecah fokusnya antara pahlawan musuh atau anak buah petarung, tetapi sekarang pahlawan AI menjadi lebih pintar dengan penghancuran sekali serang yang diterapkan menggunakan algoritma *Greedy* tersebut.

Walau begitu penerapan algoritma *Greedy* dalam hal ini tetap ada kekurangannya yaitu menyebabkan AI tidak menghiraukan serangan terhadap pahlawannya ketika berusaha menghancurkan anak buah petarung.

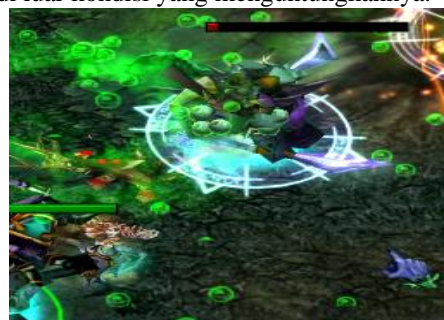
Dalam penggunaan algoritma *Greedy* dalam hal ini akan tepat sekali jika digunakan bersama algoritma *Divide and Conquer*. Dengan menggunakan algoritma *Divide and Conquer* akan sangat memudahkan AI untuk mendeteksi anak buah petarung yang sekarat. Sehingga kebutuhan akan komputasi sistem permainan yang cepat pun akan tercapai.

III.3 Menyerang Pahlawan Musuh dan Kabur dari Pertarungan

Dalam memperhitungkan kapan menyerang pahlawan musuh, AI juga akan menggunakan algoritma *Greedy*. Selama AI berada dalam posisi menguntungkan, yaitu:

1. Pahlawan AI memiliki jarak serang yang lebih jauh dibandingkan dengan jarak serang pahlawan musuh.
2. Pahlawan AI memiliki darah yang lebih banyak dibandingkan pahlawan musuh.
3. Pahlawan AI memiliki tingkatan yang lebih tinggi dibanding pahlawan musuh.
4. Pahlawan musuh berada di wilayah kekuasaan AI.
5. Pahlawan musuh menyerang pahlawan AI.
6. Pahlawan AI memiliki persediaan obat-obatan.
7. Dan kondisi lainnya yang menguntungkan bagi AI.

Jika berada dalam satu atau lebih kondisi di atas, AI akan otomatis menyerang pahlawan musuh. Pahlawan AI akan terus menyerang, bahkan sampai mengejar pahlawan musuh jika kabur. Penyerangan akan berakhir ketika AI berada di luar kondisi yang menguntungkannya.



Gambar 3 : AI melarikan diri dari pertarungan.

Contoh, ketika sedang bertarung dengan pahlawan musuh, pahlawan AI terkena kerusakan yang berat, maka *Greedy* penyerangan akan berhenti dan berganti dengan *Greedy* lainnya yaitu *Greedy* kabur dari pertarungan. Selama berada dalam fase kabur dari pertarungan, AI akan menggunakan semua upaya untuk dapat pulang ke markas dan memulihkan pahlawannya.



Gambar 4: AI mengejar pahlawan musuh.

Kelebihan dari algoritma *Greedy* dalam hal ini, dengan kondisi awal yang jelas, AI menjadi lebih hati-hati dalam menyerang dan bertahan. Dan kekurangannya adalah AI tidak terlalu memikirkan keadaan setelah aksi yang dilakukannya, misal AI menyerang pahlawan musuh, AI akan terus menyerang walaupun pahlawan musuh dibantu oleh anak buah petarungannya. Sehingga menyebabkan AI akan cepat memasuki kondisi yang tidak menguntungkan baginya.

Kekurangan lainnya adalah serangan yang dilakukan tidak menghancurkan pahlawan musuh. Karena pahlawan AI menyerang ketika berada pada kondisi yang menguntungkannya bukan ketika pahlawan musuh tidak menguntungkan. Tujuan penyerangan yang dilakukan oleh AI untuk menghancurkan tetapi terkadang tidak tercapai karena serangan yang lemah, tidak adanya *skill* yang dapat membantu dalam penyerangan, atau pahlawan musuh berada dalam kondisi yang baik atau tidak sekarat.

Kekurangan algoritma *Greedy* yang digunakan dalam usaha kabur adalah AI tidak memikirkan apa saja yang akan terjadi di masa mendatang, jadi ketika sedang berada di posisi yang tidak menguntungkan bagi pahlawannya maka AI akan kabur walaupun terkadang pahlawannya dapat menghancurkan bangunan atau pahlawan musuh dengan kondisi pahlawannya yang tidak menguntungkan tersebut. Tetapi di kasus lain, algoritma *Greedy* ini memberikan solusi terbaik yaitu menghindari pahlawan AI untuk dihancurkan.

Dari gambar 3 dapat dilihat AI menggunakan skill yang dimilikinya untuk melarikan diri. Semua upaya yang dapat dilakukannya pada saat itu akan dilakukan dengan tujuan mencapai markas untuk memulihkan diri. Analisis yang dapat dilakukan bahwa AI akan berada dalam posisi melarikan diri atau tidak akan ada niat untuk memberikan kerusakan kepada pahlawan musuh, sehingga skill yang digunakan untuk melarikan diri pun tidak ada tujuan untuk melukai pahlawan musuh, hanya digunakan untuk menakuti atau menghentikan pengejaran pahlawan musuh.

Hal ini tentu memiliki keuntungan dan kerugian tersendiri. Selamat sampai ke markas merupakan tujuan lokal yang ingin dicapai, tetapi tetap saja tujuan keseluruhannya adalah memenangkan permainan. Dari segi keuntungan yaitu, tujuan atau maksimum lokal yang ingin dicapai dapat terwujud. Tetapi kerugiannya untuk maksimum secara keseluruhan dapat tidak tercapai.

III.4 Menggunakan Skill

Dalam penggunaan skill pahlawan, AI juga menggunakan algoritma *Greedy*. Jika terdapat banyak anak buah petarung dan pahlawan musuh dalam satu daerah AI akan otomatis mengeluarkan skill area yang dapat memberikan kerusakan ke banyak unit.

Keuntungannya, AI mendapatkan banyak uang dari banyaknya anak buah petarung yang hancur dan sekaligus memberikan kerusakan kepada pahlawan musuh. Kekurangannya karena hanya menginginkan uang, biasanya serangan ini tidak dilanjutkan dengan serangan lainnya yang bersifat serius untuk menghancurkan pahlawan musuh (jika berada pada posisi yang tidak menguntungkan bagi AI).



Gambar : Skill area untuk menghancurkan anak buah petarung.

AI akan menggunakan skill juga ketika sedang menyerang pahlawan musuh. AI akan mengeluarkan skill jika mana (syarat untuk menjalankan skill) tercukupi. Jadi selama ada mana, AI akan langsung mengeluarkan skill-nya untuk menyerang pahlawan AI.

Tentu ini memiliki keuntungan dan kerugiannya tersendiri dalam permainan. Keuntungannya AI dapat menggunakan skill berkali-kali ketika bertarung dengan pahlawan musuh. Karena skill yang telah digunakan akan memiliki waktu tunggu untuk dapat dijalankan lagi. Oleh karena itu, jika digunakan di awal-awal pertarungan dengan pahlawan musuh, ada kemungkinan mana yang tersedia masih mencukupi dan waktu tunggu skill tersebut selesai, sehingga skill dapat digunakan lagi.



Gambar : Skill yang digunakan ke pahlawan musuh.

Kekurangannya adalah skill terkadang digunakan di waktu yang tidak tepat dan terbuang percuma karena biasanya jika digunakan di saat yang tidak tepat, skill hanya akan memberikan kerusakan kepada lawan tetapi tidak sampai menghancurkannya.

III.5 Melindungi Bangunan

Dalam melindungi bangunannya AI memiliki dua cara, yaitu :

1. Menggunakan *Glyph* yang menyebabkan bangunan menjadi kebal.
2. Mendatangi bangunan dan melindunginya secara langsung.

Jika AI mendapati bahwa sakah satu bangunannya diserang, dan tidak terdapat pahlawan yang menjaganya maka AI akan langsung menjalankan pergerakan perlindungan dengan prioritas langkah seperti di atas.

Jika *Glyph* tersedia dan tidak berada dalam waktu tunggu maka AI akan menggunakan *Glyph*. Akan tetapi jika *Glyph* tidak tersedia, AI akan berpindah tempat menuju bangunan tersebut untuk melakukan perlindungan.



Gambar : Melindungi bangunan menggunakan *Glyph*.

Beberapa keuntungan dalam penggunaan algoritma *Greedy* di dalam perlindungan bangunan ini adalah AI sangat protektif dengan bangunannya. Tetapi tentu saja terdapat kekurangan yang sangat mendasar dari algoritma ini ketika digunakan dalam perlindungan bangunan.

Kelemahannya yaitu AI terlalu berlebihan dalam melindungi bangunannya, karena tidak memikirkan tingkat keberbahayaan serangan yang dilakukan terhadap bangunannya menyebabkan AI akan melindungi bangunannya walaupun serangan terhadap bangunannya tidak terlalu bahaya. Ini tentu akan sangat merugikan AI karena menggunakan *Glyph* di saat yang tidak perlu dan berpindah-pindah tempat secara terus menerus menyebabkan AI tidak dapat fokus melakukan hal yang lain.

IV. KESIMPULAN

1. Algoritma *Greedy* digunakan di beberapa bagian permainan DotA, menunjukkan bahwa strategi algoritma sangat berguna di kehidupan nyata.
2. Penggunaan algoritma *Greedy* di dalam permainan DotA memiliki kelebihan dan kekurangan. Akan tetapi kekurangannya selalu mengalami penurunan karena algoritma yang digunakan selalu diperbaharui oleh pembuat-pembuat AI untuk permainan DotA.
3. Penggunaan algoritma *Greedy* yang dibarengi dengan penggunaan algoritma lainnya akan meningkatkan kepintaran AI dan mempercepat komputasi sistem permainan.

4. Penggunaan algoritma *Greedy* saja tidak akan dapat mencapai solusi maksimum terus menerus, oleh karena itu pengkombinasian dengan algoritma lain merupakan salah satu solusi yang dapat digunakan untuk memaksimalkannya.

V. REFERENSI

- Rinaldi Munir. Slide mata kuliah IF3051 tahun ajaran 2011/2012.
http://id.wikipedia.org/wiki/Defense_of_the_Ancients
Waktu Akses 30 November 2011, Pukul 3.30 WIB.
http://id.wikipedia.org/wiki/Kecerdasan_buatan
Waktu Akses 30 November 2011, Pukul 3.40 WIB.
http://en.wikipedia.org/wiki/Greedy_algorithm
Waktu Akses 30 November 2011, Pukul 3.45 WIB.

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, 8 Desember 2011

Gurun Nevada Dharan/ 13509076