

Algoritma *Divide and Conquer* untuk Optimasi Pemberian Efek *Aura* pada Permainan *DotA (Defense of the Ancient)*

Arifin Luthfi Putranto

NIM 13508050

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

Email : arifin.luthfi@gmail.com

ABSTRAK

Defense of the Ancient (DotA) adalah sebuah *custom map* untuk permainan komputer buatan Blizzard yang berjudul *Warcraft III: Frozen Throne*. Tujuan utama permainan ini adalah untuk menghancurkan “Ancient” musuh, sebuah struktur yang dijaga ketat di pojok kiri bawah atau kanan atas (yang satu untuk sisi sendiri dan yang satu lagi untuk sisi musuh).

Dalam *DotA*, terdapat banyak karakter (*hero*) yang dapat kita pilih untuk dimainkan. Beberapa karakter memiliki kemampuan yang dapat mempengaruhi status musuh yang ada di sekitarnya dengan jarak (*AoE – Area of Effect*) tertentu. Kemampuan ini biasanya disebut sebagai “*Aura*”. Untuk mengetahui unit mana saja yang terkena efek dari aura tersebut, perlu dihitung jarak dari unit ke sumber aura.

Makalah ini akan membahas tentang pengalokasian algoritma *Divide and Conquer* untuk mendapatkan unit mana saja yang terkena efek dari aura yang dikeluarkan. *Divide and Conquer* digunakan sebagai optimasi pencarian unit-unit yang dimaksud dikarenakan luas *map* (peta arena permainan) pada *DotA* cukup luas.

Kata kunci : *AoE, Aura, Combine, Divide and Conquer, DotA, map, sub-masalah.*

1. PENDAHULUAN

1.1. Aura dalam *DotA*

DotA adalah permainan tim, dalam bermain *DotA* kita tidak hanya fokus untuk mengembangkan karakter milik kita sendiri, kita juga harus dapat mendukung pemain satu tim agar dapat bersama-sama menghancurkan *ancient* milik musuh. Musuh yang kita hadapi dalam permainan juga tidak hanya satu, banyak musuh yang harus dihadapi secara bersamaan.

Aura dalam *DotA* sangat membantu dalam tim. Dalam *DotA*, aura dapat membantu teman satu tim ataupun membuat lemah tim lawan. Aura biasanya mengenai tidak hanya satu unit, melainkan seluruh unit yang ada di dalam *AoE* tertentu. Oleh karena itu, aura akan sangat membantu tim untuk memenangkan permainan.

Di bawah ini merupakan contoh dari penggunaan aura dalam permainan *DotA* :



Gambar 1: *AoE* dari *Degen Aura*

Pada gambar diatas, dapat terlihat daerah mana saja yang terkena efek dari kemampuan *Degen Aura* yang dimiliki oleh karakter *Omniknight*. *Degen Aura* dapat memperlambat gerakan musuh yang termasuk kedalam *AoE* aura tersebut. Dapat dilihat monster-monster musuh yang masuk kedalam *AoE* tersebut terkena efek aura (ditandai dengan adanya sinar biru dibawah monster).

Dalam pengimplementasian efek aura, untuk mengecek apakah suatu unit dalam efek aura atau tidak, perlu dihitung jarak antara unit dengan sumber aura. Tentunya komputer akan melakukan komputasi untuk menghitung jarak antara unit dengan sumber aura.

1.2. Algoritma *Divide and Conquer*

Divide and Conquer merupakan sebuah algoritma yang bekerja dengan cara memecah masalah yang besar menjadi beberapa buah sub-masalah yang kemudian akan diselesaikan satu-persatu. Algoritma ini dapat digunakan untuk mengoptimasi masalah-masalah mana saja yang perlu dikerjakan untuk menyelesaikan suatu masalah besar, tanpa harus mengerjakan bagian masalah lain yang tidak ada hubungannya dengan tujuan yang ingin kita capai.

2. LANDASAN TEORI

2.1. *Divide and Conquer*

Divide and Conquer adalah metode pemecahan masalah yang bekerja dengan membagi masalah besar menjadi beberapa sub-masalah yang lebih kecil, kemudian menyelesaikan masing-masing sub-masalah tersebut

secara independen, dan akhirnya menggabungkan solusi masing-masing sub-masalah sehingga menjadi solusi dari masalah semula.

Algoritma *Divide and Conquer* memiliki tiga proses utama yaitu:

- Divide, membagi masalah menjadi beberapa sub-masalah yang memiliki kemiripan dengan masalah semula namun berukuran lebih kecil.
- Conquer, memecahkan masing-masing sub-masalah
- Combine, menggabungkan solusi masing-masing dari sub-masalah sehingga membentuk solusi dari masalah semula.

Pada algoritma *Divide and Conquer*, tiap-tiap sub-masalah mempunyai karakteristik yang sama dengan karakteristik masalah yang akan diselesaikan, sehingga metode *Divide and Conquer* lebih natural diungkapkan dalam skema rekursif. Secara umum, algoritma Divide and Conquer memiliki skema sebagai berikut :

```

procedure DIVIDE_and_CONQUER (input n
: integer)
{
Menyelesaikan masalah dengan algoritma
Divide and Conquer.
Masukan : masukan yang berukuran n.
Keluaran : solusi dari masalah semula.
}
Deklarasi
r, k : integer
Algoritma
if n ≤ n0 then
{ukuran masalah sudah cukup kecil}
SOLVE sub-masalah;
else
{masalah masih perlu dibagi}
Bagi menjadi r sub-masalah, masing
masing berukuran n/k;
for masing-masing dari r sub-
masalah do
DIVIDE_and_CONQUER(n/k);
endfor
COMBINE solusi dari r sub-masalah;
{gabung solusi dari sub-masalah}
endif

```

3. METODE PEMECAHAN MASALAH

Perhitungan unit mana saja yang termasuk dalam efek dari aura dalam permainan DotA dilakukan dengan mengecek setiap unit yang ada di dalam *map*. Setiap unit di cek apakah jaraknya dengan sumber aura lebih kecil atau sama dengan AoE dari kemampuan aura yang dimiliki oleh sumber. Perhitungan ini akan memakan waktu cukup lama apabila semua unit yang ada pada *map* dalam permainan harus di cek. Ini dikarenakan ukuran *map* pada permainan cukup besar dan juga banyak terdapat unit-unit (monster, bangunan, *hero*, dan sebagainya) yang terlibat.

Berikut adalah *mini-map* (peta dengan ukuran yang dikecilkan) dari permainan DotA :



Gambar 2 : *Mini-Map* DotA

Dapat dilihat dari *mini-map* diatas bahwa cukup banyak terdapat unit-unit (ditunjukkan dengan titik-titik berwarna) didalam permainan. Dirasa kurang efektif jika seluruh unit yang ada di dalam *map* harus dicek, karena biasanya radius aura dari hanya berdiameter 1/8 ukuran *map* dari sumber. Untuk melakukan optimasi, perlu dibatasi unit-unit mana saja yang harus di cek.

3.1. Aplikasi Algoritma *Divide and Conquer*

Untuk mengoptimasi pencarian mana saja unit yang akan mendapatkan efek dari suatu aura tertentu, kita dapat menggunakan algoritma *Divide and Conquer*. Algoritma ini dapat membatasi wilayah pencarian unit-unit mana saja yang jaraknya harus dihitung dengan sumber dari aura yang dimaksud. Dengan algoritma ini, kita dapat membagi-bagi *map* menjadi bagian-bagian kecil dimana bagian-bagian tersebut dapat mengelompokkan unit apa saja yang terdapat didalamnya. Dengan begitu, kita tidak perlu menghitung jarak seluruh unit yang ada dalam *map* terhadap sumber aura. Kita hanya perlu menghitung jarak unit-unit yang ada pada bagian peta yang letaknya berdekatan dengan unit sumber dari aura, sesuai dengan AoE yang dimaksud.

Pembagian *map* dalam DotA dapat dilakukan seperti pada gambar dibawah ini :



Gambar 3 : Pembagian map

Map yang telah dibagi-bagi lalu kemudian diberi nomor sesuai dengan urutan untuk mengetahui posisinya. Setelah dilakukan pembagian map menjadi bagian-bagian yang lebih kecil, selanjutnya kita cukup menghitung jarak unit-unit yang ada pada bagian map yang berdekatan dengan bagian map sumber aura tersebut berasal. Kita tidak perlu lagi menghitung jarak seluruh unit yang ada di dalam map.

Gambar di bawah ini akan menunjukkan bagian map mana saja yang unit-unit didalamnya harus dihitung jaraknya dengan sumber aura :



Gambar 4: Contoh Wilayah Pengecekan Aura

Sumber aura ditandai dengan sebuah titik kuning pada bagian map yang memiliki nomor 5.4. Pada gambar diatas dapat dilihat juga bahwa jari-jari dari AoE tidak lebih besar dari panjang/lebar map yang sudah dibagi-bagi. Dengan demikian, kita hanya perlu menghitung jarak unit-unit yang terdapat pada bagian map yang bertetangga langsung dengan bagian map yang didalamnya terdapat sumber aura (bagian map nomor 5.4), juga bagian map tempat sumber aura itu berasal. Dalam hal ini, yang perlu kita hitung adalah unit-unit yang terdapat dalam bagian map nomor 4.3, 4.4, 4.5, 5.3, 5.4, 5.5, 5.6, 6.3, 6.4 dan 6.5. Unit yang berada di bagian-bagian map yang lain tidak perlu dihitung karena sudah dapat dipastikan unit-

unit yang berada di dalamnya mempunyai jarak dengan sumber yang lebih besar dibandingkan dengan AoE dari aura tersebut. Dengan begitu, unit-unit tersebut sudah pasti tidak akan mendapatkan pengaruh dari aura.

Setelah kita melakukan perhitungan jarak unit-unit yang terdapat dalam bagian map yang masuk dalam hitungan, kita lalu menyatukan hasil unit mana saja yang masuk ke dalam AoE dari aura tersebut. Dengan dilakukannya penyatuan ini, kita dapat mengetahui mana saja unit yang kemungkinan mendapatkan efek dari aura yang ada.

Tidak semua unit yang masuk kedalam AoE aura terkena efek dari aura yang dimaksud. Terdapat beberapa jenis aura yang ada didalam permainan DotA. Ada aura yang berguna bagi tim untuk menambah daya serang (seperti pada *True Shoot Aura* yang dimiliki *Drow Ranger*), ada pula aura yang merugikan yang akan dirasakan oleh tim lawan (seperti *Degen Aura* yang dimiliki *Omniknight*). Hanya unit-unit milik tim sendiri lah yang akan terkena aura dari *True Shoot Aura*, begitu pula dengan *Degen Aura*, hanya unit milik tim lawan saja yang akan terkena efeknya.

Pada intinya, pengaplikasian algoritma *Divide and Conquer* untuk mencari unit yang masuk ke dalam AoE dapat dijabarkan sebagai berikut :

1. Bagi map menjadi bagian-bagian yang lebih kecil. (Divide)
2. Tentukan bagian map tempat sumber aura yang dimaksud berasal.
3. Hitung jarak-jarak unit yang berada pada bagian-bagian map yang berdekatan dengan bagian map tempat aura berasal, apakah masuk ke dalam AoE. (Conquer)
4. Tentukan apakah unit yang masuk ke dalam AoE terkena efek dari aura atau tidak.
5. Gabungkan semua unit yang terkena efek dari aura. (Combine)

Jika ditulis dalam notasi algoritma, aplikasi algoritma *Divide and Conquer* dalam pencarian unit yang terkena efek dari aura adalah sebagai berikut :

```

Function AURAUNIT (input m : map, s :
unit, output lo : list of unit)
{
Mencari unit mana saja dalam map m
yang terkena aura yang berasal dari
unit s.
Masukan : map (m), sumber aura (s).
Keluaran : list dari unit yang terkena
efek dari aura (lo).
}
Algoritma
if m ≤ ukuran_minimum_map then
{ukuran peta sudah cukup kecil}
for setiap unit yang ada di m do
hitung_jarak_thd_sumber();
if jarak ≤ AoE then
if unit masuk kategori then

```

```

        masukan_unit_ke_list();
    endif
endif
endfor
else
{map masih perlu dibagi}
bagi_map_menjadi_bagian_kecil();
for masing-masing bagian map do
    AURAUNIT(m, s);
endfor
COMBINE_list_unit_dari_map();
{menggabungkan list unit-unit yang
mendapatkan efek dari aura}
endif

```

4. ANALISIS

Aplikasi algoritma *Divide and Conquer* sangat membantu dalam pencarian unit yang terkena efek dari aura. Dengan pengaplikasian algoritma ini, proses komputasi untuk melakukan pencarian akan jauh berkurang jika dibandingkan pencarian dengan cara menghitung semua unit yang ada di dalam *map*. Kita dapat melihat perbandingan komputasi yang dilakukan dengan *Divide and Conquer* dengan perhitungan langsung melalui contoh dibawah :



Gambar 5: Daerah Pencarian Unit

Dapat dilihat pada gambar diatas, dengan pengaplikasian algoritma *Divide and Conquer*, kita hanya perlu menghitung jarak sumber terhadap unit-unit yang terdapat pada wilayah yang ditandai oleh kotak biru. Kita misalkan saja terdapat 100 buah unit pada keseluruhan map dalam permainan. Tanpa menggunakan algoritma *Divide and Conquer*, kita perlu menghitung jarak keseluruhan 100 unit tersebut dengan sumber aura. Unit-unit dengan posisi yang masuk ke dalam AoE dari aura lalu kemudian diberi efek aura. Dengan begini kita harus menjalankan fungsi untuk menghitung jarak 100 kali. Hal ini tidak efisien karena DotA adalah permainan realtime, dimana proses pencarian unit mana yang terkena efek dari aura akan

selalu diupdate secara berkala.

Sekarang anggap saja ada 20 unit yang berada pada daerah kotak biru pada gambar diatas. Ini berarti kita cukup melakukan perhitungan jarak antara unit dengan sumber aura sebanyak 20 kali saja, proses komputasi menjadi jauh lebih efektif. Kita tidak perlu menghitung jarak unit yang ada di luar daerah biru tersebut karena sudah dapat dipastikan unit yang berada di luar daerah tersebut tidak akan terkena efek dari aura yang dihasilkan sumber.

Pengaplikasian algoritma *Divide and Conquer* terbukti memiliki kelebihan dibandingkan pencarian dengan cara biasa, yaitu waktu komputasi yang jauh lebih cepat. Kecepatan komputasi sangatlah diperlukan dalam sebuah permainan realtime seperti DotA. Namun, pengaplikasian algoritma *Divide and Conquer* sedikit sulit sehingga kemungkinan terdapat *bug* dalam permainan menjadi bertambah.

5. SIMPULAN

Kecepatan komputasi dalam permainan realtime seperti DotA sangatlah penting. Untuk meunjang kecepatan proses komputasi dalam permainan, diperlukan algoritma yang tepat untuk menyelesaikan setiap permasalahan yang dihadapi.

Divide and Conquer merupakan salah satu paradigma desain algoritma yang penting dan bekerja secara rekursif dengan memecah masalah yang besar menjadi beberapa sub-masalah sehingga mudah diselesaikan. Setelah itu, hasil penyelesaian sub-masalah tersebut kemudian digabungkan kembali untuk menghasilkan solusi dari permasalahan awal.

Pengaplikasian algoritma *Divide and Conquer* pada permainan DotA ini sendiri terdapat pada proses pencarian unit mana yang akan terkena efek dari kemampuan aura yang dimiliki oleh sebuah karakter. Untuk mencari tahu unit mana saja yang terkena kemampuan aura dari suatu karakter, algoritma *Divide and Conquer* sangatlah baik digunakan karena dengan algoritma ini proses komputasi akan berjalan jauh lebih cepat. Dengan algoritma ini, wilayah pencarian unit yang terkena efek dari aura dapat dibatasi.

Algoritma *Divide and Conquer* membagi *map* dalam permainan DotA dimana bagian-bagian tersebut menyimpan data unit apa saja yang ada di dalamnya. Dengan begitu, untuk mencari unit yang kemungkinan mendapatkan efek dari aura, proses penghitungan jarak unit yang kemungkinan akan mendapatkan efek dari aura akan menjadi lebih sedikit.

6. REFERENSI

[1] Munir, Rinaldi,. Strategi Algoritmik, Teknik Informatika ITB. Bandung. 2007.

- [2] http://en.wikipedia.org/wiki/Divide_and_conquer_algorithm -
Tanggal Akses 7/12/2010
- [3] http://id.wikipedia.org/wiki/Defense_of_the_Ancients - Tanggal
Akses 7/12/2010

7. 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, 7 Desember 2010



Arifin Luthfi Putranto
13508050