

Penerapan Algoritma Greedy dan Algoritma Uniform Cost Search dalam Permainan Age of Empires: The Age of Kings

Rafael Sean Putra / 13518119
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13518119@std.stei.itb.ac.id

Abstract—Permainan bergenre *turn-based strategy* memerlukan perencanaan yang matang dalam mengambil keputusan. Salah satu cara perencanaan yang bisa digunakan adalah dengan menggunakan algoritma-algoritma yang telah tersedia, seperti algoritma UCS dan algoritma greedy.

Keywords—*Turn-Based; Uniform Cost Search; Greedy; Age of Empires*;

I. PENDAHULUAN

Di jaman sekarang ini, *video game* telah menjadi sebuah komoditas yang umum kita temukan. Bisa dibilang, *video game* telah menjadi bagian dari kehidupan banyak orang saat ini. *Video game* telah menjadi sarana bagi kita menghilangkan stres, sebagai sarana komunikasi dan berhubungan dengan orang banyak, dan bahkan sebagai sarana belajar.

Video game memiliki banyak genre. Tiap-tiap genre dari *video game* tersebut memiliki keunikannya sendiri. Mulai dari *puzzle*, *racing*, simulasi, aksi, penjelajahan, hingga strategi, tiap genre memiliki bermacam-macam *video game*-nya masing-masing dan berbagai macam penggemar.

Salah satu dari berbagai macam genre tersebut adalah genre strategi. *Video game* bergenre strategi memiliki banyak penggemar karena *video game* bergenre strategi dapat digunakan oleh orang banyak untuk melatih kemampuan pengambilan keputusan serta merencanakan strategi untuk mencapai kemenangan pada permainan. Beberapa *video game* bergenre strategi seperti *Civilization*, *Romance of the Three Kingdoms*, *Battleship*, *Total War*, dan masih banyak lagi juga turut mendorong popularitas dari *video game* bergenre strategi.

Salah satu *video game* bergenre strategi yang meninggalkan bekas bagi penulis adalah *video game* bertajuk *Age of Empires: The Age of Kings*. *Age of Empires: The Age of Kings* merupakan sebuah *video game* bergenre *turn-based strategy* yang memiliki komponen-komponen permainan yang cukup solid dan dapat memberikan sensasi bermain strategi yang cukup menantang.

Namun seiring berjalannya waktu, mulai muncul pertanyaan. Apakah saat memainkan *video game* tersebut, penulis telah melakukan aksi-aksi yang optimal? Lalu,

bagaimana caranya untuk menentukan apakah aksi yang dilakukan adalah aksi yang optimal?

Dalam makalah ini, penulis mencoba menjawab pertanyaan tersebut. Penulis menggunakan dua buah algoritma, yaitu algoritma *greedy* dan algoritma *uniform cost search* dalam menentukan pergerakan unit yang paling optimal pada permainan *Age of Empires: The Age of Kings*.

II. DASAR TEORI

A. Algoritma Greedy

Algoritma Greedy merupakan sebuah metode yang populer digunakan untuk memecahkan persoalan yang berhubungan dengan pencarian solusi yang paling optimum. Persoalan tersebut dapat dibedakan menjadi dua, yaitu persoalan minimasi dan persoalan maksimasi. Persoalan minimasi biasa berhubungan dengan mengurangi biaya atau cost yang harus diberikan, seperti persoalan menentukan rute terpendek dari satu titik ke titik lainnya. Persoalan maksimasi berhubungan dengan memperbanyak hal yang bisa didapatkan dari persoalan, seperti persoalan mengenai barang-barang yang dapat dimasukkan ke dalam sebuah container (*knapsack problem*).

Algoritma greedy memiliki tiga buah karakteristik, yaitu:

1. Pemilihan sebuah pilihan yang paling optimum pada sebuah momen lokal. Pilihan ini kita sebut pilihan optimum lokal,
2. Tidak ada *backtracking* atau pemilihan ulang pilihan optimum dari suatu momen lokal yang telah memiliki pilihan optimum lokal,
3. Penggabungan pilihan optimum lokal dari semua momen lokal dengan harapan bahwa hasil penggabungan merupakan pilihan optimum global.

Algoritma greedy terdiri dari 5 buah komponen, yaitu:

1. Himpunan kandidat

Himpunan kandidat merupakan himpunan yang berisi pilihan-pilihan yang dapat diambil sebagai sebuah solusi.

2. Himpunan solusi

Himpunan solusi merupakan himpunan yang berisi semua pilihan yang dipilih dari himpunan kandidat sebagai pilihan yang optimum oleh algoritma greedy.

3. Fungsi seleksi

Fungsi seleksi merupakan fungsi yang digunakan untuk memilih pilihan pada himpunan kandidat yang akan dimasukkan ke dalam himpunan solusi.

4. Fungsi kelayakan

Fungsi kelayakan merupakan fungsi yang digunakan untuk mengecek apakah pilihan yang terdapat di dalam himpunan solusi memenuhi syarat dari persoalan.

5. Fungsi obyektif

Fungsi obyektif adalah fungsi lain yang diperlukan untuk mengoptimalkan hasil algoritma greedy.

Meskipun algoritma greedy akan menghasilkan sebuah solusi yang dianggap optimum pada tiap momen lokalnya, namun algoritma greedy belum tentu menghasilkan solusi yang paling optimum untuk sebuah persoalan secara keseluruhan.

B. Graf

Graf adalah suatu permodelan yang digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf terdiri dari dua buah komponen, yaitu himpunan tidak kosong dari simpul-simpul (*vertices*) dan himpunan sisi (*edge*) yang menghubungkan sepasang simpul.

Berdasarkan orientasi arah, graf dibedakan menjadi 2 jenis. Yaitu:

1. Graf tak berarah (*undirected graph*)

Graf yang sisi-sisinya tidak memiliki arah. Dalam graf tak berarah, sebuah sisi antara simpul A dan simpul B menyatakan hubungan dari simpul A ke simpul B dan juga hubungan dari simpul B ke simpul A.

2. Graf berarah (*directed graph*)

Graf yang setiap sisinya memiliki arah. Arah pada sisi graf biasa digambarkan menggunakan tanda panah. Dalam graf berarah, sebuah sisi antara simpul A dan simpul B hanya menyatakan satu buah hubungan, entah itu hubungan dari simpul A ke simpul B atau hubungan dari simpul B ke simpul A. Hal tersebut ditentukan oleh arah dari panah pada sisi yang menghubungkan kedua simpul tersebut.

C. Algoritma Uniform Cost Search

Uniform Cost Search, atau yang biasa disingkat sebagai UCS, adalah sebuah algoritma yang digunakan untuk menyelesaikan persoalan mengenai perencanaan rute. Ketika menggunakan algoritma UCS, sama seperti algoritma

pencarian rute lainnya, peta daerah yang akan dicari rutenya akan diubah ke dalam bentuk graf terlebih dahulu.

Algoritma UCS terdiri dari beberapa komponen. Komponen pertama adalah antrian pemanggilan. Setelah sebuah simpul dibangkitkan, simpul tersebut akan dimasukkan ke dalam antrian pemanggilan. Di dalam antrian pemanggilan, simpul-simpul diurutkan berdasarkan jarak simpul tersebut dari simpul awal (*root*).

Komponen kedua adalah *list* simpul yang telah dipanggil. Ketika sebuah simpul telah diperiksa, maka simpul tersebut akan dimasukkan ke dalam *list* simpul yang telah diperiksa. *List* ini digunakan untuk mencegah sebuah simpul yang telah diperiksa dibangkitkan kembali dan dimasukkan ke dalam antrian pemanggilan.

Algoritma UCS berguna dalam menangani graf yang tidak memiliki ujung maupun graf yang berukuran terlalu besar untuk direpresentasikan di dalam memori.

D. Turn-Based Strategy

Turn-Based Strategy (TBS) adalah salah satu genre pada *video game* yang merupakan sub-genre dari genre strategi. Genre TBS merupakan sebuah genre pada *video game* dimana beberapa pihak bermain secara bergiliran. Pada giliran mereka, pihak yang sedang bermain akan melakukan tindakan yang diinginkan. Ketika pihak yang sedang bermain sudah melakukan semua tindakan yang diinginkan atau pihak tersebut tidak dapat melakukan tindakan apapun lagi, maka giliran dari pihak tersebut akan berakhir dan giliran dipindah ke pihak yang lain untuk bermain. Permainan akan dinyatakan berakhir bila tersisa hanya satu pihak saja pada permainan atau salah satu pihak telah mencapai suatu tujuan yang telah ditentukan.

Permainan bergenre *turn-based strategy*, seperti permainan strategi pada umumnya, menitikberatkan aspek strategi dan perencanaan aksi-aksi yang akan dilakukan oleh seorang pemain. Seorang pemain permainan TBS diharapkan telah memiliki sebuah "rencana besar" mengenai apa yang akan dilakukan oleh dirinya pada sebuah permainan sehingga aksi-aksi yang akan dilakukannya di tiap giliran akan dibuat agar dapat mendukung rencana besar yang telah dia rencanakan.

Dalam permainan bergenre TBS, terdapat dua hal yang menjadi fokus permainan, yaitu manajemen sumber daya (*resource management*) dan manajemen aksi (*action management*). Manajemen sumber daya pada game *turn-based strategy* berfokus ke sumber daya yang disediakan oleh permainan secara terbatas dan bagaimana tiap pemain mengatur agar sumber daya tersebut digunakan dengan sebaik mungkin. Manajemen aksi pada game *turn-based strategy* berfokus kepada aksi atau Tindakan yang dilakukan oleh pemain tiap gilirannya dan pemilihan aksi yang dilakukan untuk menghasilkan hasil yang baik selama permainan tersebut.

E. Permainan Age of Empires: The Age of Kings

Age of Empires: The Age of Kings adalah sebuah *video game* bergenre *turn-based strategy* buatan Backbone Entertainment yang dirilis pada tahun 2006 pada platform Nintendo DS. Dalam game Age of Empires: The Age of Kings,

pemain mengendalikan salah satu dari lima kerajaan yang disediakan dan harus menyelesaikan tujuan-tujuan yang telah ditentukan. Tujuan pada *video game* ini bervariasi, mulai dari memindahkan sebuah unit tertentu ke suatu posisi tertentu, mengalahkan kerajaan lain yang dikendalikan oleh pemain lain atau oleh komputer (AI), mengalahkan sebuah unit spesifik milik pihak lawan, dan berbagai macam tujuan lainnya.



Gambar 2.1 *Interface* dari Permainan Age of Empires: The Age of Kings, Menampilkan Data dari Militia, Salah Satu Unit dalam Permainan
(Sumber: Dokumen Pribadi)

Dalam *video game* ini, terdapat beberapa aspek-aspek dalam permainan yang harus diperhatikan, yaitu:

1. Daerah permainan

Pada Age of Empires: The Age of Kings, daerah permainan berbentuk persegi atau persegi panjang dengan ukuran bervariasi. Tiap daerah permainan dibagi menjadi beberapa petak yang tiap petaknya memiliki medannya masing-masing dan memiliki biaya jalan (*walking cost*) untuk berpindah ke petak tersebut. Terdapat beberapa macam medan di dalam permainan Age of Empires: The Age of Kings, yaitu di antaranya: Jalan, Rerumputan, Rawa-rawa, Gurun, Hutan, Bukit, Pegunungan, Jembatan, Sungai, dan Laut. Dari semua medan tersebut, medan Sungai dan medan Laut adalah medan yang tidak dapat dilalui oleh semua unit pada permainan.



Gambar 2.2 Permodelan Daerah Permainan Beserta Biaya Jalan dan Medan Setiap Petak
(Sumber: Dokumen Pribadi)

2. Unit

Tiap pemain dapat memproduksi dan menggerakkan unit pada tiap gilirannya. Unit digunakan pemain untuk meningkatkan laju perkembangan pihaknya atau

menghambat laju perkembangan musuh. Hal tersebut dapat dilakukan dengan cara menggunakan unit di pihak kita untuk menyerang unit milik musuh. Beberapa unit juga memiliki kemampuan khusus yang hanya dimiliki oleh unit itu sendiri, seperti membuat bangunan, serangan jarak jauh, menyembuhkan unit lain, dan kemampuan-kemampuan lainnya.

Tiap unit terdiri dari beberapa komponen, yaitu nyawa (*health*) yang dimiliki oleh unit tersebut, kekuatan serangan (*attack*), kekuatan bertahan (*defense*), kemampuan berjalan, serta karakteristik spesial.

Terdapat empat tipe unit pada permainan Age of Empires: The Age of Kings. Keempat tipe tersebut adalah *infantry*, *cavalry*, *ranged*, dan *siege*. Tiap unit memiliki karakteristiknya masing-masing, seperti unit *cavalry* tidak boleh bergerak melalui medan rawa-rawa, unit *ranged* dapat menyerang 2-3 petak dari lokasi unit lawan, dan unit *siege* tidak bisa bergerak melalui rawa-rawa dan pegunungan.

Beberapa unit memiliki kemampuan-kemampuan unik yang dapat digunakan bila memenuhi beberapa syarat. Contoh dari kemampuan unik ini adalah unit *pikemen* yang merupakan unit bertipe *infantry* mendapat peningkatan kekuatan serangan sebesar 33% dari kekuatan serangan normal bila unit tersebut menyerang unit bertipe *cavalry*.

3. Pertempuran

Ketika sebuah unit berada di sebelah (atau dalam kasus unit *ranged*, 2-3 petak dari) unit musuh, pemain dapat memilih opsi untuk melakukan pertempuran antara unit miliknya dengan unit milik lawannya. Dalam permainan ini, pertempuran dilakukan secara otomatis dan hasil akhirnya ditentukan oleh kekuatan serangan, kekuatan bertahan, serta nyawa dari kedua belah pihak.

III. PENERAPAN ALGORITMA

Dalam makalah ini, algoritma digunakan untuk mengurus aspek manajemen aksi dari permainan Age of Empires: The Age of Kings. Aspek manajemen sumber daya tidak akan dibahas di dalam makalah ini untuk mensimplifikasi penerapan algoritma.

Seperti yang telah disebutkan sebelumnya, pada permainan Age of Empires: The Age of Kings, terdapat berbagai macam tujuan yang perlu kita capai. Salah satu aspek dasar yang perlu dilakukan untuk mencapai tujuan tersebut adalah mengenai unit apa yang harus digerakkan pada tiap putaran/giliran. Hal ini dapat ditentukan menggunakan dua algoritma, yaitu algoritma *uniform cost search* untuk mencari tahu jarak antara sebuah unit dengan target dan berapa banyak giliran yang perlu dilakukan. Lalu begitu data jarak didapatkan, data jarak tersebut dan informasi-informasi lainnya mengenai unit yang kita miliki akan diolah menggunakan algoritma *greedy* untuk menentukan unit mana saja yang akan kita gerakkan pada giliran tersebut.

2. Apakah unit memiliki perbandingan kekuatan serangan yang lebih besar atau sama bila dibandingkan dengan kekuatan bertahan milik unit target?
3. Apakah unit memiliki jumlah langkah minimum untuk mencapai petak tujuan?

Setelah semua unit dicek dan didapatkan nilai pengambilan milik semua unit. Kita akan melakukan pengambilan dari unit-unit yang tersedia untuk digerakkan pada giliran tersebut. Pengambilan ini dilakukan terhadap unit yang memiliki nilai pengambilan melebihi batas minimal nilai pengambilan yang telah kita tentukan.

Bila berpacu kepada skema komponen algoritma *greedy*, maka komponen-komponen dari penerapan algoritma *greedy* ini adalah sebagai berikut:

1. Himpunan kandidat: Seluruh unit milik pemain yang ada di daerah permainan pada giliran tersebut
2. Himpunan solusi: Unit yang akan digerakkan oleh pemain pada giliran tersebut
3. Fungsi seleksi: Unit yang pada giliran tersebut memiliki nilai pengambilan lebih tinggi dari batas nilai pengambilan yang telah ditentukan sebelumnya
4. Fungsi kelayakan: -
5. Fungsi obyektif: fungsi penentuan jarak optimal dari unit dengan target menggunakan algoritma uniform cost search dan fungsi penentuan nilai pengambilan

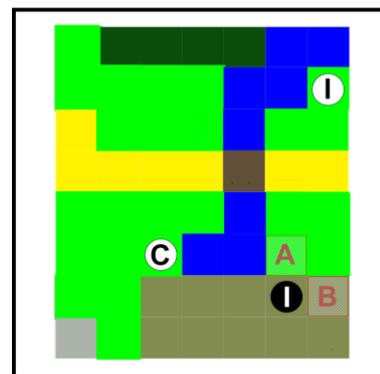
C. Contoh Kasus

Setelah menjabarkan penerapan kedua algoritma tersebut, kita akan menguji penerapan tersebut melalui sebuah contoh kasus. Contoh kasus akan menggunakan daerah permainan yang telah dimodelkan pada gambar 2.2. Pada contoh kasus ini, akan terdapat 3 buah unit, yang spesifikasinya adalah sebagai berikut:

TABEL I. SPESIFIKASI TIAP UNIT

No	Tipe Unit	Pemilik	Kekuatan Serangan	Kekuatan Bertahan	Kemampuan Berjalan
1	Infantry	Putih	100	100	7
2	Cavalry	Putih	150	80	8
3	Invantry	Hitam	100	100	7

Posisi tiap unit pada peta adalah sebagai berikut:



Gambar 3.1 Peta Permainan dengan Ketiga Unit Di Dalamnya
(Sumber: Dokumen Pribadi)

Pada contoh kasus kali ini, kita akan menjadi pihak putih dan memiliki misi untuk mengalakan unit milik pihak hitam, yaitu unit *Infantry* yang dilambangkan dengan lingkaran hitam dengan tulisan "I" putih di gambar 3.1. Dari gambar 3.1, diketahui bahwa untuk melakukan pertempuran dengan unit *Infantry* milik hitam, kita harus memindahkan salah satu unit kita ke petak yang dilabeli A atau B di daerah permainan.

Setelah mengetahui tujuan dari unit kita, kita akan mencari jarak terpendek menggunakan algoritma UCS menuju kedua petak tersebut. Hasil dari penelusuran UCS adalah sebagai berikut:

TABEL II. HASIL PENELUSURAN ALGORITMA UCS PADA GILIRAN PERTAMA

No	Tipe Unit	Cost ke Petak A	Jumlah Langkah yang Dibutuhkan	Cost ke Petak B	Jumlah Langkah yang Dibutuhkan
1	Infantry	8	2	9	2
2	Cavalry	11	2	-	-

Perlu diketahui bahwa pencarian *cost* menuju ke petak B untuk unit *cavalry* tidak akan dilakukan karena petak B memiliki medan berupa rawa-rawa yang tidak dapat dimasuki oleh unit *cavalry*.

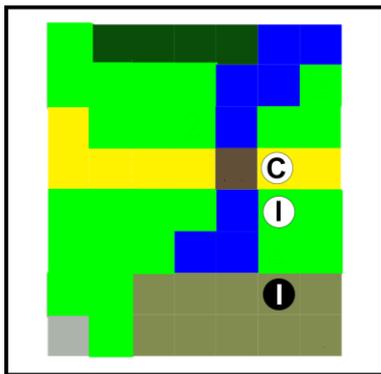
Setelah mengetahui jarak optimal menggunakan algoritma UCS, kita akan menggunakan algoritma *greedy* untuk menentukan unit mana saja yang akan kita gerakkan pada giliran kali ini. Pada contoh kasus kali ini, kita akan menggunakan batas nilai pengambilan bernilai dua. Kita akan mengevaluasi semua unit yang kita miliki berdasarkan kondisi yang telah kita tentukan sebelumnya.

TABEL III. TABEL KONDISI PADA GILIRAN PERTAMA

No	Kondisi	Unit 1 (Infantry)	Unit 2 (Cavalry)
1	Petak tujuan dapat diakses oleh unit	Ya	Tidak
2	Unit memiliki kekuatan	Ya	Ya

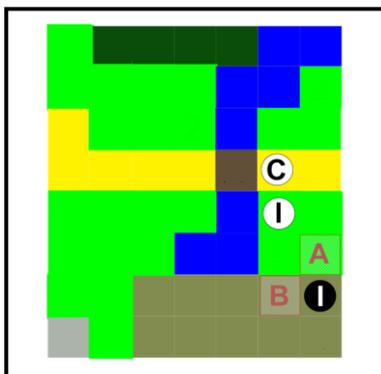
No	Kondisi	Unit 1 (<i>Infantry</i>)	Unit 2 (<i>Cavalry</i>)
	serangan lebih besar atau sama bila dibandingkan dengan kekuatan pertahanan musuh		
3	Unit memiliki jumlah langkah minimum untuk mencapai musuh	Ya	Ya

Dari Tabel III, kita mendapatkan bahwa kedua unit milik kita memiliki nilai pengambilan yang lebih atau sama dengan 2. Unit *Infantry* kita memiliki nilai pengambilan bernilai 3, sementara unit *cavalry* kita memiliki nilai pengambilan bernilai 2. Oleh karena itu, kita akan menggerakkan kedua unit tersebut pada giliran kita kali ini. Hasil dari pergerakan kedua unit tersebut dapat dilihat pada gambar 3.2 berikut.



Gambar 3.2 Peta Permainan Setelah Pihak Putih Menggerakkan Unit Miliknya (Sumber: Dokumen Pribadi)

Setelah menggerakkan kedua unit tersebut, pihak putih akan mengakhiri gilirannya tersebut dan pihak hitam akan bergerak. Kita misalkan pihak hitam menggerakkan unit *Infantry* miliknya satu petak ke kanan.



Gambar 3.3 Peta Permainan Ketika Pihak Putih memulai Giliran Kedua (Sumber: Dokumen Pribadi)

Setelah itu, pihak hitam akan menyudahi gilirannya dan permainan akan masuk ke giliran kedua dari pihak putih. Dilihat dari gambar 3.3, bahwa Ketika unit *Infantry* milik pihak hitam berpindah posisi di peta permainan, maka petak tujuan yang harus dicapai unit milik pihak putih akan berpindah juga.

Pada giliran kedua pihak putih, akan dihitung terlebih dahulu jarak dari setiap unit untuk menuju petak tujuan menggunakan algoritma UCS.

TABEL IV. HASIL PENELUSURAN ALGORITMA UCS PADA GILIRAN KEDUA

No	Tipe Unit	Cost ke Petak A	Jumlah Langkah yang Dibutuhkan	Cost ke Petak B	Jumlah Langkah yang Dibutuhkan
1	<i>Infantry</i>	4	1	4	1
2	<i>Cavalry</i>	5	1	-	-

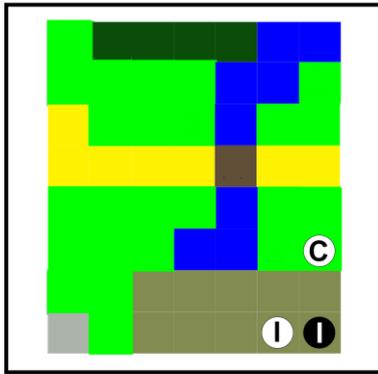
Sama seperti pada giliran pertama, pencarian *cost* untuk pergerakan unit *Cavalry* menuju petak B tidak dilakukan karena petak B memiliki medan berupa rawa-rawa yang tidak dapat dimasuki oleh unit *Cavalry*.

Setelah mengetahui jarak optimal menggunakan algoritma UCS, kita akan menggunakan algoritma *greedy* untuk menentukan unit mana saja yang akan kita gerakkan pada giliran kali ini. Kita akan mengevaluasi semua unit yang kita miliki berdasarkan kondisi yang telah kita tentukan sebelumnya.

TABEL V. TABEL KONDISI PADA GILIRAN KEDUA

No	Kondisi	Unit 1 (<i>Infantry</i>)	Unit 2 (<i>Cavalry</i>)
1	Petak tujuan dapat diakses oleh unit	Ya	Tidak
2	Unit memiliki kekuatan serangan lebih besar atau sama bila dibandingkan dengan kekuatan pertahanan musuh	Ya	Ya
3	Unit memiliki jumlah langkah minimum untuk mencapai musuh	Ya	Ya

Mengacu pada data yang terdapat pada tabel V, kita mendapatkan bahwa unit *Infantry* milik pihak putih memiliki nilai pengambilan bernilai 3 dan unit *Cavalry* milik pihak putih memiliki nilai pengambilan bernilai 2. Karena kedua unit milik pihak putih memiliki nilai pengambilan yang bernilai sama atau melebihi batas nilai pengambilan, maka kedua unit tersebut akan digerakkan pada giliran kedua pihak putih.



Gambar 3.6 Peta Permainan setelah Pihak Putih Menggerakkan Unit Miliknya (Sumber: Dokumen Pribadi)

Setelah menggerakkan unit *Infantry* milik pihak putih, dapat dilihat bahwa unit *Infantry* tersebut berada di sebelah unit *Infantry* milik pihak hitam, sehingga pihak putih dapat melakukan pertempuran menggunakan unit *Infantry*-nya tersebut.

Kita asumsikan bahwa unit *Infantry* milik pihak hitam kalah setelah pertempuran tersebut. Oleh karena itu, pihak putih berhasil menyelesaikan permainan pada contoh kasus kali ini.

IV. HASIL

Dalam pemanfaatan algoritma ini di dalam permainan, ditemukan bahwa pemanfaatan algoritma *greedy* dan algoritma UCS masih belum menghasilkan hasil yang optimal. Hal ini dikarenakan oleh beberapa faktor, di antaranya yaitu:

1. Algoritma *greedy* yang digunakan dalam memilih unit yang akan digerakkan belum menggunakan semua aspek. Aspek yang digunakan hanya berupa medan petak tujuan, langkah yang dibutuhkan, serta perbandingan kekuatan serangan. Padahal, masih banyak aspek-aspek lain yang dapat dipertimbangkan, seperti nyawa dari unit milik kita dan milik musuh, jumlah unit musuh yang berada di sekitar unit target, medan di petak-petak di sekitar petak target, dan masih banyak aspek lainnya.
2. Algoritma akan menggerakkan unit menuju target secara langsung. Bila tujuan dari permainan merupakan sebuah petak tertentu, hal ini merupakan hal yang baik. Namun, bila tujuan dari permainan adalah mengalahkan unit lawan, hal ini dapat mengakibatkan beberapa unit mengalami kekalahan yang tidak perlu dialami.

V. KESIMPULAN

Algoritma *greedy* dan algoritma UCS merupakan algoritma yang cukup berbeda. Tetapi, kedua algoritma tersebut memiliki

tujuan yang sama, yaitu untuk menghasilkan solusi yang optimum dari suatu persoalan. Salah satu persoalan yang dapat diselesaikan menggunakan algoritma tersebut adalah persoalan menentukan pergerakan unit dalam *video game* bergenre strategi, terutama dalam *video game* Age of Empires: The Age of Kings.

LINK VIDEO

Untuk memperjelas pemahaman mengenai makalah ini, pembaca dapat mengakses link video ini: <https://youtu.be/HXBvTyPObXA>

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa karena berkat rahmat-Nya, makalah berjudul "Penerapan Algoritma Greedy dan Algoritma Uniform Cost Search dalam Permainan Age of Empires: The Age of Kings" dapat diselesaikan. Penulis mengucapkan terima kasih kepada Dr. Nur Ulfa Maulidevi ST, M.Sc selaku dosen pengajar mata kuliah IF2211 Strategi Algoritma yang telah memberikan ilmu kepada penulis terkait materi Strategi Algoritma. Penulis juga ingin berterima

Penulis juga ingin berterima kasih kepada teman-teman penulis yang telah memberikan bantuan dan dukungan kepada penulis dalam penulisan makalah ini.

REFERENSI

- [1] Slide Algoritma Greedy (2020) oleh Rinaldi Munir diakses pada 26 April 2020
- [2] Slide Graf (2019) oleh Rinaldi Munir diakses pada 26 April 2020
- [3] <https://www.britannica.com/topic/electronic-strategy-game#ref1080973> diakses 26 April 2020
- [4] <https://www.geeksforgeeks.org/uniform-cost-search-dijkstra-for-large-graphs/> diakses 26 April 2020

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, 3 Mei 2020

Rafael Sean Putra 13518119