

Penerapan Algoritma *Greedy* dalam Strategi Bermain The Sims Medieval

Ruth Nattassha (13510096)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
salvaterarug@gmail.com

Abstract—Algoritma *greedy* merupakan salah satu cara yang dapat digunakan untuk memenuhi berbagai persoalan optimasi. Salah satu penggunaannya yang paling umum adalah dalam menyusun strategi dalam bermain game. Makalah ini membahas penerapan algoritma tersebut dalam strategi bermain The Sims Medieval, baik dalam pemilihan gedung-gedung yang akan dibangun dan juga pemilihan *quest* yang tepat.

Index Terms—*greedy*, optimasi, *quest*, solusi

I. PENGANTAR

Game komputer sudah tidak lagi menjadi sesuatu yang asing dalam kehidupan kita. Tujuan utama keberadaan *game* ini jelas untuk menjadi suatu bentuk hiburan bagi mereka yang sedang bosan atau jenuh dengan beban kehidupan sehari-hari ini. Walaupun memang yang diinginkan adalah dapat memainkan *game* tanpa perlu memikirkan strategi dan sejenisnya, dalam beberapa kondisi justru cara bermain yang seperti itu memperpanjang waktu penyelesaian persoalan dalam permainan dan membuat pemainnya semakin frustrasi. Dan jelas, hal tersebut sudah menyalahi fungsi utama *game* tersebut dibuat.

Oleh karena itu, tidak ada salahnya memperhitungkan beberapa metode yang dapat mempermudah menjalani permainan dan menyelesaikan persoalan-persoalan yang ada. Metode-metode ini terangkum dalam strategi-strategi algoritma yang bahkan dapat dibuat menjadi suatu program pembangkit tersendiri. Dengan adanya strategi algoritma tersebut, tidak lagi diperlukan adanya sistem coba dan gagal dalam memainkan *game* yang ada, karena analisa terhadap langkah yang harus diambil memang sudah terlebih dahulu dilakukan bersama dengan perkiraan terhadap kemungkinan hasilnya.

Dalam makalah ini, strategi algoritma yang menjadi pilihan adalah algoritma *greedy* (penjelasan lebih lengkap akan dilakukan tersendiri). Adapun *game* yang dipilih untuk penerapan algoritma *greedy* ini adalah The Sims Medieval, dan akan dibahas persoalan-persoalan yang ada dalam *game* tersebut beserta penyelesaiannya

dengan menggunakan algoritma *greedy*.

II. ALGORITMA *GREEDY*

Di antara banyak algoritma yang dapat digunakan dalam masalah optimasi, algoritma *greedy* adalah salah satu yang paling banyak digunakan. Nama algoritma ini sendiri berarti ‘tamak’ atau ‘rakus’. Penamaan tersebut dikarenakan prinsip utama algoritma ini adalah mengambil solusi yang paling baik pada saat itu juga. Solusi yang baik ini adalah solusi yang memberikan sumbangan paling berarti (memiliki bobot paling kecil ataupun keuntungan yang paling besar) dan masih memenuhi syarat-syarat yang diberlakukan dalam pengambilan.

Perlu diperhatikan bahwa dalam menyelesaikan persoalan, algoritma *greedy* akan terdiri dari beberapa tahap. Untuk setiap tahap, terdapat sebuah solusi yang juga disebut sebagai optimum lokal. (karena dinyatakan optimum hanya untuk kondisi saat itu juga). Nilai-nilai optimum lokal inilah yang, setelah digabungkan, diharapkan akan menjadi optimum global (alias nilai optimum yang diicipai untuk persoalan tersebut).

Untuk menggunakan algoritma *greedy* dalam menyelesaikan suatu persoalan, terlebih dahulu haruslah didefinisikan elemen-elemen algoritma *greedy* yang berlaku dalam persoalan tersebut. Adapun elemen-elemen yang dimiliki algoritma *greedy* antara lain:

- Himpunan kandidat (C)
- Himpunan solusi (S)
- Fungsi seleksi
- Fungsi kelayakan
- Fungsi obyektif

Pembahasan mengenai elemen-elemen tersebut secara lebih mendetail antara lain:

A. Himpunan Kandidat

Himpunan kandidat merupakan himpunan yang mengandung elemen-elemen yang dapat menjadi bagian dari solusi. Contoh untuk himpunan kandidat adalah koin-koin dalam persoalan optimasi penukaran uang dengan sejumlah koin.

B. *Himpunan Solusi*

Himpunan solusi terdiri dari elemen-elemen yang sesungguhnya merupakan bagian dari himpunan kandidat dan apabila dikumpulkan bersama dapat menjadi solusi global dari permasalahan yang ada. Contoh himpunan solusi adalah kombinasi uang logam berjumlah paling sedikit namun memiliki nilai yang diinginkan dalam persoalan optimasi penukaran uang.

C. *Fungsi Seleksi*

Fungsi inilah yang berperan dalam pemilihan solusi lokal setiap tahap dalam algoritma *greedy*. Solusi lokal pada tahap tersebut adalah nilai yang dihasilkan oleh fungsi seleksi setelah digunakan pada himpunan kandidat. Contoh fungsi seleksi ini adalah memilih koin dengan nilai paling tinggi di antara himpunan kandidat yang disediakan.

D. *Fungsi Kelayakan*

Fungsi yang digunakan untuk mengetahui apakah solusi yang optimal itu masih sesuai dengan syarat yang diminta dalam persoalan. Fungsi kelayakan merupakan pembatas dalam pemanfaatan fungsi seleksi, karena hasil fungsi seleksi haruslah dinyatakan layak apabila diperiksa dengan fungsi kelayakan. Contoh fungsi kelayakan adalah fungsi yang memeriksa nilai total dari koin-koin yang ada, agar tidak lebih daripada nilai yang diharapkan pada masalah optimasi penukaran uang.

E. *Fungsi Obyektif*

Optimasi yang diharapkan dari pemecahan masalah dengan menggunakan algoritma *greedy*. Biasanya bersifat mencari yang sekecil-kecilnya atau mencari yang sebesar-besarnya. Untuk masalah penukaran uang, contoh fungsi obyektifnya adalah jumlah koin yang digunakan minimum.

Karena algoritma *greedy* hanya memperhitungkan solusi-solusi yang elemennya merupakan solusi terbaik dalam tahapan tertentu, optimum global belum tentu menjadi solusi paling baik yang mungkin didapatkan. Masih ada kemungkinan bahwa solusi yang tidak diperhitungkan dengan menggunakan algoritma *greedy* justru merupakan solusi yang paling optimum untuk permasalahan yang ada. Oleh karena itu, algoritma *greedy* lebih banyak digunakan untuk mencari hampiran terhadap sebuah solusi. Untuk mendapatkan solusi yang paling mendekati solusi sebenarnya, pendefinisian fungsi seleksi harus dilakukan dengan hati-hati.

III. THE SIMS MEDIEVAL

The Sims Medieval merupakan salah satu *game* dalam seri The Sims milik Electronic Arts. *Game* ini dibangun berdasarkan *engine* milik saudaranya, yaitu The Sims 3, dengan beberapa konsep yang berbeda. Seri The Sims pada umumnya menitikberatkan pada kehidupan sehari-hari karakter-karakter yang dapat kita desain sendiri, dengan latar yang didesain sedemikian rupa sehingga mendekati kehidupan modern abad ini yang sesungguhnya. Pemain dibebaskan untuk mendesain rupa karakter, usianya, kepribadian karakter yang bersangkutan, pekerjaannya, hubungannya dengan karakter-karakter lain, bahkan sampai desain rumah tempat karakter itu tinggal dan juga perabotan yang digunakannya. Dapat dikatakan, haluan serial The Sims pada umumnya adalah *real-life simulator*, di mana kita dapat mensimulasikan kehidupan sehari-hari melalui permainan itu.

Untuk garis besar konsep haluannya, The Sims Medieval masihlah belum menyimpang dari serialnya (karena itu nama yang diberikan pun masih kurang lebih sama). Namun, The Sims Medieval memiliki beberapa batasan dan juga fitur yang tidak dimiliki oleh anggota serial The Sims lainnya. Batasan dan fitur tersebut (hanya yang relevan dengan strategi yang akan dibahas) antara lain:

A. *Latar Medieval*

Memainkan *game* dari serial The Sims mungkin jadi membuat jenuh orang-orang yang menginginkan suasana baru dalam permainannya. Untuk mereka yang menginginkan latar yang lebih mengarah ke fantasi, The Sims Medieval merupakan pilihan yang paling tepat di antara rangkaian *game* yang terdapat dalam serial The Sims. Alih-alih kehidupan zaman modern dengan pernak-perniknya, The Sims Medieval mengambil latar medieval atau disebut juga sebagai abad pertengahan.

Dalam *game* ini, pemain berperan sebagai *Watcher* yang dianggap sebagai dewa oleh karakter-karakter yang ada. Adapun tugas pemain adalah membangun kerajaan yang dimilikinya dan memainkan karakter-karakter yang ada dalam kerajaan itu. Perlu diingat di sinilah poin menarik The Sims Medieval: karena latarnya abad pertengahan, kehidupan yang dijalani karakter-karakternya pun masih khas kehidupan masa itu. Karakter masih harus berburu, mencari ikan di sungai, dan bahkan memungut lintah untuk melakukan suatu operasi pengobatan sederhana,

B. *Quest*

Sudah disebutkan sebelumnya bahwa dalam

serial The Sims umumnya, pemain diberikan kebebasan dalam menentukan karakternya dan bahkan dunia tempat karakter itu dimainkan, dengan tujuan yang juga didefinisikan oleh pemain itu sendiri. The Sims Medieval memberikan tujuan yang tidak ada dalam *game* The Sims lainnya, yaitu *quest*. *Quest* adalah suatu tugas yang harus dilaksanakan oleh penghuni kerajaan, biasanya oleh orang-orang yang memiliki posisi tertentu. Menyelesaikan sebuah *quest* akan mendapatkan Resource Point (RP) untuk membangun kerajaannya, Aspects (dijelaskan di bagian *Kingdom*), Renown (juga dijelaskan di bagian *Kingdom*), dan juga Experience Points (EXP) untuk karakter yang terlibat dalam *quest* tersebut.

Pengambilan *quest* tidak dapat dilakukan dengan sesuka hati. Harus diperhatikan terlebih dahulu apakah terdapat Quest Point (QP) yang cukup dan apakah karakter yang harus terlibat dalam *quest* itu sudah tersedia.

C. *Kingdom*

Membangun kerajaan adalah perhatian utama dalam permainan The Sims Medieval. Pembangunan ini dilakukan terhadap banyak elemen, yang pada umumnya dapat dicapai dengan menyelesaikan suatu *quest*. Adapun elemen dari sebuah kerajaan dalam The Sims Medieval sebagai berikut:

- Bangunan, gedung-gedung yang berada dalam kerajaan. Gedung tidak akan dibangun dan didesain sendiri oleh pemain, karena pemain hanya akan memilih gedung apa yang dibangun. Gedung tertentu akan menambahkan sejumlah poin Aspects dan karakter baru yang dapat dimainkan. Untuk menambah gedung baru, pemain harus membayar dengan Resource Point.
- Renown, merupakan tingkat popularitas kerajaan dalam dunia itu. Renown bisa didapatkan melalui penyelesaian *quest* (jumlah Renown yang didapat sama dengan jumlah Resource Point yang didapat) dan juga melalui beberapa *event* yang mungkin terjadi selama game dimainkan.
- Aspects, kondisi yang dimiliki oleh kerajaan pada saat itu dan dapat menentukan *quest* yang tersedia. Terdapat empat jenis aspek: *Well-Being*, *Security*, *Culture*, dan *Knowledge*. *Well-Being* merupakan gambaran garis besar kehidupan

penduduk kerajaan dalam hal kesehatan, kesejahteraan, dan juga kenyamanan mereka. *Security* adalah tingkat keamanan kerajaan tersebut dan juga stabilitasnya, dengan tingkat yang rendah dapat mengundang banyak bandit. *Culture* menggambarkan perlakuan kerajaan terhadap seni dan agama. *Knowledge* menandakan sejauh apa tereduksinya penduduk di kerajaan tersebut.

Aspek didapat dengan menjalankan *quest* dan juga menambahkan gedung baru.

D. *Playable Characters*

Karakter yang dapat dimainkan dalam The Sims Medieval terbatas oleh peran-peran yang ada. Hal itu diartikan bahwa memang pemain masih dapat mendesain karakter yang ingin mereka mainkan (ciri fisik, kepribadian, dan sejenisnya) dan memainkannya, akan tetapi hanya untuk peran-peran tertentu saja. Bahkan, untuk kasus memainkan, pemain hanya dapat menggerakkan karakter yang berhubungan dengan *quest* yang sedang dilaksanakan.

Terdapat sepuluh jenis karakter (disebut juga Hero Sims) yang dapat didesain dan dimainkan: *Bard*, *Blacksmith*, *Jacobsen Priest*, *Knight*, *Merchant*, *Monarch*, *Peteran Priest*, *Physician*, *Spy*, dan *Wizard*. Opsi untuk mendesain dan memainkan karakter ini akan muncul apabila gedung-gedung yang menjadi syarat keberadaan mereka sudah ada di kerajaan. Untuk mengetahui karakter yang bisa didapatkan melalui suatu gedung, cukup dengan melihat karakter apa yang akan *diunlock* pada keterangan yang ada di tiap gedung.

Karakter-karakter ini memiliki level-level yang menentukan keahliannya. Karakter dapat bertambah levelnya apabila sudah memiliki Experience Point yang cukup. Experience Point sendiri bergantung pada aspek *Knowledge* yang dimiliki oleh kerajaan tersebut.

IV. GREEDY DALAM SIMS MEDIEVAL

Algoritma greedy akan digunakan dalam penyusunan strategi pemilihan *quest* dan gedung. Pembahasan akan dilakukan satu-per-satu.

A. *Pemilihan Quest*

Sudah dijabarkan sebelumnya bahwa untuk memilih *quest* dibatasi oleh Quest Point dan menyelesaikan *quest*

dapat memberikan Resource Point, Aspect, Renown, dan juga Experience Point. Yang dapat dinilai secara eksplisit adalah Resource Point (dan Renown) dan Aspect. Algoritma greedy akan dibuat berdasarkan kedua jenis penilaian.

Elemen algoritma greedy untuk permasalahan *quest* adalah sebagai berikut:

- Himpunan kandidat (C): *quest* yang tersedia pada saat itu juga.
- Himpunan solusi (S): *quest* yang dipilih untuk dilaksanakan.
- Fungsi seleksi: *quest* dengan Resource Point terbanyak atau *quest* dengan Aspect (yang diinginkan) yang terbanyak.
- Fungsi kelayakan: Quest Point masih mencukupi dan karakter yang dibutuhkan sudah tersedia.
- Fungsi obyektif: mendapatkan list *quest* yang memberikan jumlah Resource Point terbanyak ataupun Aspect (yang diinginkan) yang terbanyak.

Contoh penerapannya bisa dilihat dalam kasus berikut:

Diketahui properti setiap quest adalah QP (Quest Point yang dibutuhkan), RP (resource point yang dihasilkan), AS-X (Aspect yang dihasilkan, X = W untuk Well-Being, S untuk Security, C untuk Culture dan K untuk Knowledge), dan HS (Hero Sims, karakter yang dibutuhkan untuk quest tersebut). Quest yang menjadi kasus dinyatakan melalui Tabel 1.

i	RP _i	QP _i	AS-W _i	AS-S _i	AS-C _i	AS-K _i	HS _i
1.	50	3	2	0	1	0	Any
2.	20	1	1	0	0	0	Any
3.	50	4	1	1	1	1	Physician + any
4.	30	2	0	1	1	0	Monarch

Tabel 1. Contoh Kasus Pemilihan Quest

Diketahui juga bahwa karakter yang dimiliki saat ini adalah *Monarch*, *Physician*, *Merchant*, *Wizard*, dan *Knight*. Quest Point yang tersedia adalah 8 QP. Untuk kasus dengan poin yang sama, akan dipilih *quest* dengan i lebih kecil.

Dengan menggunakan algoritma greedy dan fungsi seleksi yang didefinisikan sebelumnya, maka *quest* yang dipilih akan menjadi sebagai berikut:

– *Greedy by Resource Point*

Fungsi ini akan memilih *quest* berdasarkan bonus Resource Pointnya. Oleh karena itu, pada tahapan pertama yang akan dipilih adalah *quest* 1, dengan QP 3, mengirimkan *Merchant/Wizard/Knight*. Pada tahap kedua, yang dipilih adalah *quest* 3, dengan QP 4,

mengirimkan *Physician* dan *Merchant/Wizard/Knight*. Pada tahap ketiga, seharusnya yang dipilih adalah *quest* 4, karena $RP_4 > RP_2$. Akan tetapi, QP yang tersedia hanya 1 QP, yang berarti *quest* 4 tidak bisa diambil. Oleh karena itu, yang diambil adalah *quest* 2, mengirimkan *Merchant/Wizard/Knight*. Dengan kata lain, solusinya adalah *quest* 1, 2, dan 3. Pada kasus ini, algoritma *greedy* mencapai solusi yang paling optimal (total RP 120).

– *Greedy by Well-Being Aspect*

Quest dipilih berdasarkan aspek Well-Being yang disumbangkan. Tahap pertama yang dipilih adalah *quest* 1, dengan nilai AS-W 2 dan mengirimkan *Merchant/Wizard/Knight*. Tahap kedua yang dipilih adalah *quest* 2, dengan AS-W 1 dan mengirimkan *Merchant/Wizard/Knight*. Tahap ketiga yang dipilih adalah *quest* 3, dengan AS-W 1 dan mengirimkan *Physician* dan *Merchant/Wizard/Knight*. Solusi ini sama dengan yang didapatkan pada algoritma *greedy by Resource Point*, dengan sumbangan AS-W adalah 4 aspek.

– *Greedy by Security Aspect*

Akan dipilih terlebih dahulu *quest* yang menyumbang AS-S paling banyak. Pada tahap pertama, akan diambil *quest* 2 dengan AS-S 1 mengirimkan *Merchant/Wizard/Knight*. Pada tahap kedua, diambil *quest* 4 dengan AS-S 1, mengirimkan *Monarch*. Tahap ketiga seharusnya mengambil *quest* 1 (dimulai dari *quest* dengan i paling kecil), tapi QP yang tersedia sudah tidak mencukupi. Oleh karena itu, yang akan diambil adalah *quest* 2 dengan AS-S 0, mengirimkan *Merchant/Wizard/Knight*. Dengan demikian, *quest* yang diambil adalah *quest* 2, 3, 4 dengan total AS-S 2.

– *Greedy by Culture Aspect*

Pemilihan dilakukan berdasarkan AS-C. Tahap pertama yang dipilih adalah *quest* 1 dengan AS-C 1, mengirimkan *Merchant/Wizard/Knight*. Tahap kedua memilih *quest* 3 dengan AS-C 1, mengirimkan *Physician* dan *Merchant/Wizard/Knight*. Tahap ketiga seharusnya memilih *quest* 4 dengan AS-C 1, akan tetapi QP yang tersedia tidak lagi cukup untuk mengambil *quest* itu. Oleh karena itu, yang dipilih adalah *quest* 2. Solusi sama dengan algoritma *greedy by Resource Point* dengan nilai total AS-C sebanyak 2.

– *Greedy by Knowledge Aspect*

Dengan melakukan pemilihan berdasarkan AS-K, tahap pertama akan memilih *quest* 3 yang memiliki AS-K bernilai 1, mengirimkan *Physician* dan *Merchant/Wizard/Knight*.

Selanjutnya, karena ketiga *quest* memiliki nilai AS-K yang sama, yaitu 0, pemilihan akan dilakukan secara berurutan. Dengan kata lain, yang dipilih selanjutnya adalah *quest* 1 dan 2, masing-masing mengirimkan *Merchant/Wizard/Knight*. Sumbangan AS-K yang diberikan adalah 1 poin dengan kombinasi solusi sama dengan *greedy by Resource Point*.

Solusi secara garis besar untuk kasus ini adalah *quest* 1, 2, dan 3. Secara naluri, dapat dilihat pula bahwa solusi optimum untuk penyelesaian masalah pemilihan *quest* dalam The Sims Medieval tidak berbeda dengan hasil algoritma *greedy*. Oleh karena itu, untuk contoh kasus ini, dapat dikatakan algoritma *greedy* berhasil mencapai solusi optimum.

B. Pemilihan Gedung

Setiap gedung dalam kerajaan The Sims Medieval memberikan sumbangan aspek yang berbeda-beda. Untuk membangun mereka, pun dibutuhkan sejumlah Resource Point. Sementara, jumlah Resource Point yang disediakan untuk pemain terbatas. Oleh karena itu, untuk masalah ini, definisi elemen algoritma *greedy* yang ada antara lain:

- Himpunan kandidat (C): gedung-gedung yang ada dalam kerajaan.
- Himpunan solusi (S): gedung-gedung pilihan yang dapat dibangun.
- Fungsi seleksi: gedung dengan sumbangan Aspect (AS-X) paling besar.
- Fungsi kelayakan: jumlah Resource Point (RP) masih cukup untuk membangun gedung.
- Fungsi obyektif: mendapatkan list gedung yang bisa dibangun dengan memberikan keuntungan AS-X paling besar.

Contoh penerapan algoritma dapat dilihat dengan kasus ini:

Diketahui properti setiap gedung adalah RP (resource point yang dibutuhkan) dan AS-X (Aspect yang dihasilkan, X = W untuk Well-Being, S untuk Security, C untuk Culture dan K untuk Knowledge). Gedung dan atributnya dapat dilihat di Tabel 2. Apabila dimiliki sebanyak 70 Resource Point di awal, gedung apa saja yang akan dibangun? Catatan: jika nilai aspek sama, maka akan dipilih gedung dengan RP lebih kecil.

Gedung	RP	AS-W	AS-S	AS-C	AS-K
Castle	0	2	2	2	2
Barracks	10	0	4	0	0
Tavern	10	2	0	1	1
Market	10	2	0	0	2
Smithy	10	0	3	1	0

Clinic	10	4	0	0	0
Wizard's Tower	10	0	0	0	4
Spy Quarters	10	0	2	0	2
Cathedral	10	1	0	3	0
Monastery	10	0	0	3	1
Lighthouse	30	0	2	1	1
Kingball Court	30	1	1	2	0
Mill	30	3	0	0	1
Watcher Pavilion	30	2	0	1	1

Tabel 2. Contoh Kasus Pemilihan Gedung

Penyelesaian dari masalah tersebut dengan algoritma *greedy* akan dilakukan sebagai berikut:

- Greedy by Well-Being Aspect

Program akan mencari gedung yang memberikan AS-W paling besar. Pada tahap pertama, diambil Clinic dengan AS-W 4, RP tersisa 60. Tahap kedua, diambil Mill dengan AS-W 3 dan RP tersisa 30. Tahap ketiga akan memilih Castle dengan AS-W 2 dan RP tersisa 30 (karena cost RP untuk Castle 0). Selanjutnya, akan diambil Tavern dan Market dengan AS-W masing-masing 2 dan sisa RP 10. Setelah ini, yang seharusnya diambil berdasarkan nilai AS-W adalah Watcher Pavilion, akan tetapi sisa RP tidak memenuhi. Oleh karena itu, yang diambil adalah Cathedral dengan AS-W 1. Jadi, solusi yang didapatkan adalah Clinic, Mill, Castle, Tavern, Market, dan Cathedral dengan total AS-W 14.

- Greedy by Security Aspect

Penentuan dilakukan berdasarkan nilai AS-S paling besar. Pemilihan pertama akan jatuh pada Barracks dengan AS-S 4, sehingga tersisa 60 RP. Pilihan kedua adalah Smithy, dengan AS-S 3 dan menyebabkan tersisanya 50 RP. Pilihan selanjutnya adalah Castle, Spy Quarters dan Lighthouse, masing-masing memiliki AS-S 2 dan menyebabkan tersisa 10 RP. Selanjutnya seharusnya Kingball Court dengan AS-S 1, tapi karena RP tidak mencukupi akan dipilih Tavern dengan AS-S 0 dan RP 10. Jadi, solusi optimumnya adalah Barracks, Smithy, Castle, Spy Quarters, Lighthouse, dan Tavern dengan nilai AS-S 13.

- Greedy by Culture Aspect

Tahap pertama dan kedua pemilihan akan jatuh pada Cathedral dan Monastery, masing-masing memiliki AS-C 3 dan menyebabkan tersisa 50 RP. Selanjutnya dipilih Castle dan Kingball Court, masing-masing memiliki AS-C 2 dan menyebabkan tersisa 20 RP. Kemudian, diambil Tavern dan Smithy dengan masing-masing AS-C 1 dan RP 10. Solusi optimumnya adalah Cathedral, Monastery, Castle, Kingball Court,

Tavern, dan Smithy dengan total AS-C sebesar 12.

– *Greedy by Knowledge Aspect*

Gedung dengan nilai AS-K paling besar adalah Wizard's Tower, dengan nilai AS-K 4, pengambilannya menyebabkan tersisa 60 RP. Selanjutnya akan dipilih Castle, Market, dan Spy Quarters dengan masing-masing memiliki 2 AS-K, menyebabkan tersisanya 40 RP. Kemudian diambil Tavern dan Monastery yang memiliki nilai AS-K 1, sehingga tersisa 20 RP. Pengambilan Lighthouse dan gedung lain yang bernilai 30 RP dengan sumbangan AS-K 1 tidak dapat dilakukan karena sisa RP yang dimiliki tidak mencukupi. Oleh karena itu, yang berikutnya diambil adalah Barracks dengan nilai AS-K 0. Jadi, solusi optimum dengan algoritma ini adalah Wizard's Tower, Castle, Market, Spy Quarters, Tavern, Monastery, dan Barracks dengan total AS-K 12.

Melalui kumpulan solusi di atas, dapat dilihat bahwa tujuan yang berbeda menghasilkan solusi yang benar-benar berbeda, tidak seperti pada contoh kasus pemilihan *quest* yang hasilnya tidak terlalu jauh berbeda. Contoh kasus ini menunjukkan betapa pentingnya menyesuaikan fungsi seleksi dengan tujuan yang diinginkan, karena apabila definisi fungsi seleksi dilakukan dengan tidak tepat, solusi bisa menjadi sangat jauh dari yang diinginkan.

V. KESIMPULAN

Algoritma greedy merupakan salah satu strategi untuk menyelesaikan persoalan optimasi. Penyelesaian masalah dilakukan dengan cara membagi persoalan menjadi beberapa tahap dan menggabungkan solusi untuk setiap tahap sehingga membentuk suatu solusi global bagi permasalahan. Hasil dari algoritma ini dapat mendekati hasil yang benar-benar paling optimal, walaupun terkadang solusi yang diberikannya hanya mendekati optimal. Untuk mendapatkan hasil yang sesuai dengan tujuan penyelesaian permasalahan, fungsi seleksi yang digunakan dalam memilih solusi setiap tahap haruslah didefinisikan dengan hati-hati dan sesuai dengan tujuan penyelesaian masalah.

REFERENSI

- [1] Rinaldi Munir, "Algoritma Greedy" Diktat Kuliah IF3051 Strategi Algoritma, Bandung: Program Studi Teknik Informatika, 2009, hal.41-84.
- [2] The Sims Medieval Wiki (<http://thesimsmedieval.wikia.com/>), Artikel Quest, Buildings, dan Kingdom. Diakses tanggal 20 Desember 2012 pukul 20.00 WIB. Catatan: Tidak ada referensi ke artikel atau sumber lain karena isinya dibuat berdasarkan

pengalaman dan pengetahuan bermain masing-masing penyumbang artikel.

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, 20 Desember 2012



Ruth Natassha (13510096)