

Penerapan Algoritma Greedy dalam Permainan Seven Seas Deluxe

Samuel Sandi Kristianto Lim - 13516069

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
samuelsandi1703@gmail.com

Abstrak—Banyak permainan teka-teki berbasis giliran yang memerlukan strategi-strategi khusus untuk memenangkannya. Salahsatu contoh permainan teka-teki pada komputer yang berbasis giliran adalah Seven Seas Deluxe. Pada makalah ini, akan dijelaskan mengenai penerapan salahsatu jenis strategi algoritma yaitu *greedy* dalam usaha untuk memenangkan sebanyak mungkin level dalam permainan Seven Seas Deluxe tersebut.

Kata kunci—Seven Seas Deluxe, greedy, strategi, algoritma.

I. PENDAHULUAN

Seven Seas Deluxe merupakan sebuah permainan yang didesain untuk dijalankan pada *Personal Computer (PC)* dan dirilis pada bulan Maret 2006 oleh PopCap Games. PopCap Games merupakan sebuah perusahaan pembuat permainan komputer yang berdiri di Seattle pada tahun 2000, dan telah membuat banyak permainan komputer terkenal yang bersifat klasik seperti *Insaniquarium*, *Mummy Maze Deluxe*, *Plants vs Zombies*, *Bejeweled*, dan sebagainya.

Permainan Seven Seas Deluxe ini memiliki tema pertempuran kapal melawan bajak laut yang terjadi di tujuh daerah di lautan bebas yaitu *Crimson Sea*, *Gold Coast*, *Blackwater Isle*, *Gulf of Ghosts*, *Sargasso Sea*, *Dire Straits*, dan *Bermuda Triangle*.

II. LANDASAN TEORI

A. Seven Seas Deluxe

Pemain memulai permainan dari *level* satu dengan sebuah kapal yang berada di tengah lautan dan dikelilingi oleh beberapa kapal musuh. Lautan tersebut dibatasi oleh kotak-kotak kecil yang menjadi satuan terkecil dari gerakan kapal-kapal, baik kapal pemain maupun kapal musuh. Pada keempat pojok dari lautan tersebut, terdapat empat buah pusaran air yang apabila kapal pemain masuk ke dalamnya maka kapal tersebut kemudian akan muncul kembali di lautan tersebut pada lokasi yang acak. Pusaran air ini dapat digunakan untuk meloloskan diri dari kejaran kapal-kapal bajak laut ketika kapal pemain telah berada dalam kondisi terpojok.



Gambar 1.1 Permainan Seven Seas Deluxe

Dalam satu giliran, setiap kapal dapat bergerak satu kali sejauh satu kotak baik secara horizontal, vertikal, maupun diagonal. Kapal pemain memiliki pilihan untuk bergerak atau menembak. Sekali menembak, kapal pemain akan mengeluarkan dua buah peluru meriam ke arah kanan dan kiri kapal yang masing-masing akan menghancurkan sebuah kapal musuh yang berada dalam ruang tembak kapal pemain. Apabila tidak ada kapal musuh yang berada pada ruang tembak, maka peluru akan jatuh ke air.



Gambar 1.2 Kapal pemain

Kapal-kapal bajak laut musuh bertujuan untuk menabrak dan menghancurkan kapal pemain, sementara itu tugas pemain adalah menghancurkan semua kapal musuh yang ada di level tersebut. Sebuah kapal dapat hancur ketika ditembak, menabrak pulau, menabrak reruntuhan kapal, atau menabrak kapal lain. Kapal pemain tidak dapat menabrak pulau ataupun

reruntuhan kapal, tetapi dapat hancur jika ditembak atau ditabrak oleh kapal musuh. Setiap kapal yang hancur akan meninggalkan reruntuhan kapal yang ditandai dengan lambang tengkorak di kotak hancurnya kapal tersebut. Pemain akan memenangkan sebuah *level* ketika semua kapal musuh di *level* tersebut telah hancur. Sebaliknya, pemain akan kalah apabila kapalnya hancur dan tidak ada lagi kapal cadangan yang tersisa. Apabila kapal pemain hancur namun masih terdapat kapal cadangan yang tersisa, maka indikator jumlah kapal cadangan akan berkurang disertai muncul kembalinya kapal pemain di lautan pada lokasi yang acak.

Terdapat beberapa jenis musuh yang memiliki penampilan dan karakteristik berbeda-beda sebagai berikut:

1. Kapal bajak laut hitam

Gerakan kapal berjenis ini memiliki pola yang sangat sederhana yaitu selalu mencari rute tercepat untuk menabrak kapal pemain tanpa mempedulikan halangan di depannya. Akibatnya, kapal jenis ini sangat sering menabrak reruntuhan kapal, pulau, maupun sesama kapal bajak laut. Kapal berjenis ini hanya memiliki satu kemungkinan gerakan di setiap langkah.



Gambar 1.3 Kapal bajak laut hitam

2. Kapal bajak laut merah

Gerakan kapal berjenis ini dapat lebih beragam daripada kapal bajak laut hitam. Pada hampir setiap langkah, kapal jenis ini memiliki lebih dari satu kemungkinan gerakan yang dipilih secara acak. Biasanya kapal jenis ini cenderung menghindari ruang yang dapat ditembak oleh kapal pemain serta tidak dapat menabrak pulau ataupun reruntuhan kapal kecuali dalam kondisi tidak dapat bergerak mendekati pemain lagi.



Gambar 1.4 Kapal bajak laut merah

3. Kapal bajak laut biru

Kapal berjenis ini merupakan jenis kapal paling berbahaya karena menjadi satu-satunya jenis kapal bajak laut yang memiliki meriam sehingga dapat menembak kapal lain. Berbeda dengan kapal pemain yang tidak dapat menembak dan bergerak bersamaan dalam satu giliran, kapal bajak laut biru dapat melakukan hal tersebut. Karakteristik meriam kapal ini sama dengan karakteristik meriam kapal pemain, dan dapat menghancurkan baik kapal pemain maupun kapal bajak laut lainnya. Sementara itu gerakan kapal

jenis ini seringkali diarahkan untuk sebisa mungkin menembak kapal pemain sehingga dengan keberadaan kapal jenis ini pemain harus lebih berhati-hati memilih langkah.



Gambar 1.5 Kapal bajak laut biru

4. *The Flying Dutchman*

Kapal berjenis ini merupakan jenis kapal yang paling sulit dibunuh. Gerakan dari kapal jenis ini sangat menghindari ruang tembak kapal pemain, namun masih berusaha untuk menabrak dan menghancurkan kapal pemain di saat yang bersamaan. Diperlukan taktik khusus untuk menjebak kapal jenis ini agar dapat masuk ke ruang tembak kapal pemain. Selain itu, kapal jenis ini juga tidak dapat menabrak pulau, reruntuhan kapal, ataupun kapal lain kecuali dalam kondisi terperangkap dan tak dapat lagi mendekati kapal pemain.



Gambar 1.6 *The Flying Dutchman*

5. *Sea Monster*

Berbeda dengan jenis-jenis musuh sebelumnya, *Sea Monster* tidak termasuk kapal bajak laut. Gerakan hewan ini tidak dapat ditebak sama sekali karena tidak mengejar kapal pemain. Selain itu, hewan ini tidak dapat dibunuh baik dengan cara ditabrak oleh kapal maupun dengan cara ditembak.



Gambar 1.7 *Sea Monster*

Pada setiap level kelipatan sepuluh, tidak ada pertarungan melawan bajak laut melainkan terdapat sebuah perburuan peti harta karun. Namun hal ini tidak akan dibahas lebih lanjut pada makalah ini karena tidak berkaitan dengan strategi *greedy* yang telah disebutkan sebelumnya.

Setiap kapal bajak laut yang berhasil dihancurkan baik secara aktif maupun pasif oleh pemain akan menghasilkan skor bagi pemain. Skor untuk kapal musuh yang dihancurkan dengan cara ditembak dengan menggunakan meriam pemain akan lebih besar sepuluh dibandingkan kapal-kapal yang hancur oleh meriam kapal bajak laut, menabrak pulau, menabrak reruntuhan kapal, atau saling bertabrakan satu sama

lain. Skor memiliki pengaruh yang besar terhadap seberapa lama pemain dapat bertahan. Hal ini karena setiap skor mencapai angka kelipatan seribu, maka sebuah nyawa cadangan akan ditambahkan bagi pemain. Oleh karena itu maka tembakan meriam harus lebih diprioritaskan untuk membasmi musuh dalam permainan ini.

B. Algoritma Greedy

Algoritma *greedy* merupakan salahsatu strategi penyelesaian masalah yang sering digunakan untuk menangani kasus-kasus optimasi yang mencari solusi maksimum ataupun minimum dari sebuah permasalahan. Pada kasus ini, algoritma *greedy* akan digunakan untuk mencari sebuah solusi terbaik yang akan membawa pemain ke *level* setinggi-tingginya.

Terdapat dua buah terminologi dasar yang umum digunakan dalam penggunaan algoritma *greedy* yakni solusi layak dan solusi optimum. Solusi layak adalah solusi yang memenuhi syarat tertentu yang diberikan oleh sebuah persoalan optimasi, sementara solusi optimum adalah solusi yang memberikan nilai terbaik (paling tinggi atau paling rendah) dari semua solusi layak. Algoritma ini memiliki prinsip mengambil langkah yang terlihat paling menarik pada suatu waktu tertentu, tanpa mempertimbangkan konsekuensi pengambilan langkah tersebut di kemudian waktu.

Algoritma *greedy* merupakan salahsatu algoritma yang dalam pembentukan solusinya menggunakan teknik penguraian masalah ke dalam langkah-langkah yang diselesaikan satu demi satu. Setiap langkah harus merupakan keputusan yang terbaik dan tidak dapat diubah ketika langkah selanjutnya telah ditentukan. Namun karena algoritma ini hanya mengambil langkah yang “terlihat” merupakan langkah terbaik dalam suatu tahap tertentu tanpa melihat langkah-langkah yang dapat diambil berikutnya, maka algoritma ini seringkali tidak dapat menentukan dengan pasti mana yang akan menjadi solusi global dari sebuah permasalahan. Namun dengan mengambil solusi-solusi lokal yang terbaik, diharapkan bahwa solusi global yang diambil merupakan solusi yang terbaik pula.

Persoalan optimasi pada algoritma *greedy* memiliki beberapa elemen utama yaitu:

1. Himpunan kandidat

Himpunan ini berisi semua pilihan keputusan-keputusan yang dapat membangun solusi global. Pada setiap langkah, sebuah kandidat diambil dari himpunan ini untuk dipindahkan ke dalam himpunan solusi yang akan benar-benar membangun solusi global dari permasalahan yang sedang dibahas.

2. Himpunan solusi

Himpunan ini berisi hasil-hasil penerapan strategi *greedy* pada setiap langkah. Setiap elemen himpunan ini dapat dikatakan sebagai pilihan keputusan yang “paling menarik” pada saat pilihan tersebut diambil dan dimasukkan dari himpunan kandidat ke dalam himpunan solusi ini.

3. Fungsi seleksi

Fungsi seleksi merupakan fungsi yang digunakan pada setiap langkah pencarian solusi untuk memilih elemen dari himpunan kandidat yang paling berpotensi untuk membentuk solusi global yang terbaik.

4. Fungsi kelayakan

Fungsi kelayakan adalah fungsi yang berguna untuk memeriksa kelayakan dari elemen himpunan kandidat yang telah dipilih oleh fungsi seleksi. Hal ini diperlukan karena dalam algoritma *greedy*, selalu terdapat syarat yang tidak boleh dilanggar dalam pengambilan langkah solusi. Fungsi kelayakan akan memeriksa kelayakan kandidat berdasarkan syarat yang telah ditentukan tersebut.

5. Fungsi obyektif

Fungsi obyektif merupakan fungsi yang berguna untuk memaksimumkan atau meminimumkan nilai solusi yang dicari.

Kelebihan dari algoritma *greedy* antara lain kesederhanaan dan efisiensinya. Algoritma ini dikatakan sederhana karena mudah dijelaskan serta mudah untuk dilakukan penerapannya daripada kebanyakan jenis algoritma yang ada untuk menyelesaikan sebuah persoalan yang sama. Algoritma ini dikatakan efisien karena seringkali algoritma ini, karena kesederhanaannya, dapat diimplementasikan dengan lebih efisien daripada algoritma lain.

Namun algoritma *greedy* juga memiliki kekurangan diantaranya sulit didesain dan sulit untuk diverifikasi solusinya. Algoritma ini dikatakan sulit untuk didesain karena terdapat tantangan dalam menemukan faktor yang harus dipandang sebagai tolak ukur optimasi untuk penerapan algoritma ini. Misalnya dalam kasus permainan ini ada beberapa tolak ukur *greedy* yang dapat dipilih antara lain jumlah skor, jumlah musuh yang mati dalam setiap giliran, jumlah kotak yang tersisa untuk bergerak, dan sebagainya. Algoritma ini juga dikatakan sulit untuk diverifikasi karena seringkali perlu pembuktian-pembuktian dari sisi lain untuk membuktikan bahwa algoritma *greedy* yang telah menghasilkan solusi tertentu memang menghasilkan solusi optimasi yang diharapkan dan telah menghasilkan solusi optimasi yang benar-benar optimal.

III. METODE PENGUNAAN ALGORITMA GREEDY DALAM PERMAINAN SEVEN SEAS DELUXE

Pada permainan ini, penulis telah mengelompokkan kondisi kapal pemain dalam tiga kategori:

1. Kondisi jalan buntu

Pada kondisi ini, langkah apapun yang diambil akan menyebabkan pengurangan nyawa atau kematian dari kapal pemain. Kondisi ini terjadi apabila apapun langkah yang diambil, tetap akan ada kapal musuh yang menabrak kapal pemain.



Gambar 3.1

Pada gambar 3.1, terjadi kondisi jalan buntu. Pilihan aksi yang dapat dilakukan oleh pemain adalah: bergerak ke kiri, bergerak ke arah kiri-atas, bergerak ke kanan, dan menembak. Namun pada kondisi diatas, apapun langkah yang dilakukan pasti akan berakibat pada pengurangan nyawa dari pemain.

2. Kondisi melarikan diri

Pada kondisi ini, ada kapal musuh yang telah berjarak hanya 1 kotak dengan kapal pemain. Dengan kata lain, pemain tidak dapat melakukan tembakan tanpa mengurangi nyawa atau mengalami kematian. Dalam kondisi ini, hal yang harus dilakukan adalah secepat mungkin bergerak menuju pusaran air.



Gambar 3.2

Pada gambar 3.2, terjadi kondisi melarikan diri karena kapal pemain tidak memiliki kemungkinan untuk mengalahkan dua kapal bajak laut merah, dimana salahsatunya hanya memiliki jarak satu kotak dengan kapal pemain, tanpa terlebih dahulu masuk ke pusaran air dan mendapatkan jarak yang cukup untuk melakukan persiapan posisi untuk menembak kapal bajak laut merah tersebut.

3. Kondisi bebas

Pada kondisi ini, jarak kapal terdekat lebih dari satu kotak dan kapal pemain tidak sedang berada dalam kepungan kapal-kapal musuh. Pada keadaan seperti ini, hal terbaik yang dapat dilakukan adalah menyerang atau mempersiapkan serangan untuk mengantisipasi kedatangan kapal-kapal musuh ke kotak yang lebih dekat.



Gambar 3.3

Pada gambar 3.3, kapal pemain sedang berada dalam kondisi bebas sehingga mampu mengatur posisi untuk persiapan menembak.

Karena terdapat tiga macam kondisi yang berbeda, maka diperlukan pula tiga macam strategi *greedy* yang berbeda untuk menangani masing-masing kasus karena pada masing-masing kondisi langkah yang akan diambil saling berbeda:

1. Strategi *greedy* untuk kondisi jalan buntu

Pada kondisi jalan buntu, kapal pemain dipastikan hancur. Untuk mencapai solusi global yakni level tertinggi yang dapat dicapai, solusi lokal yang dicari adalah solusi lokal yang menghasilkan jumlah kehancuran kapal musuh maksimal bersama dengan kehancuran kapal pemain. Hal ini bertujuan agar peluang terjadinya kondisi jalan buntu untuk kedua kalinya di level yang sama menurun karena jumlah kapal musuh telah menjadi lebih sedikit. Oleh karena ini, strategi *greedy* pada kondisi jalan buntu dapat disebut sebagai strategi "*greedy by death*". Berikut elemen-elemen *greedy* dari teknik ini:

a. Himpunan kandidat aksi C

Himpunan kandidat ini selalu berisi pilihan arah gerak dan pilihan menembak

$C = \{\text{atas, kanan_atas, kanan, kanan_bawah, bawah, kiri_bawah, kiri, kiri_atas, tembak}\}$

b. Himpunan solusi S

Himpunan ini berisi solusi langkah yang akan menghasilkan kematian kapal musuh terbanyak.

c. Fungsi seleksi

Fungsi seleksi pada kondisi ini mencari gerakan yang akan menghasilkan kematian kapal musuh terbanyak.

d. Fungsi kelayakan

Fungsi kelayakan pada kondisi ini akan mencari solusi yang tidak bergerak ke tempat dimana sudah ada kapal ataupun objek lain di lautan. Hal ini karena kapal pemain tidak dapat secara aktif menabrak objek lain ataupun kapal musuh kecuali kedua kapal tersebut bertabrakan ketika sedang menuju ke sebuah kotak yang sama.

e. Fungsi obyektif

Fungsi obyektif pada kondisi ini akan menghasilkan sebuah langkah yang akan menghancurkan paling banyak kapal musuh.

2. Strategi *greedy* pada kondisi melarikan diri

Pada kondisi melarikan diri, kapal pemain berada dalam kondisi telah terjepit namun tidak terkepung. Oleh karena itu seharusnya kapal pemain dalam kondisi ini akan mampu mencapai pusaran air tanpa mengalami kerugian nyawa. Namun apabila langkah yang dilakukan salah, kondisi melarikan diri pun dapat berubah menjadi kondisi jalan buntu. Oleh karena itu, pada kondisi ini tujuan utama dari strategi *greedy* adalah untuk mengambil langkah yang paling cepat membawa kapal pemain ke pusaran air. Karena itu maka strategi *greedy* yang digunakan dapat juga disebut sebagai "*greedy by distance to whirlpool*". Berikut elemen-elemen *greedy* dari teknik ini:

a. Himpunan kandidat aksi C

Himpunan kandidat ini selalu berisi pilihan arah gerak dan pilihan menembak

$C = \{\text{atas, kanan_atas, kanan, kanan_bawah, bawah, kiri_bawah, kiri, kiri_atas, tembak}\}$

b. Himpunan solusi S

Himpunan ini berisi solusi langkah yang akan menghasilkan jarak terdekat menuju ke pusaran air.

c. Fungsi seleksi

Fungsi seleksi pada kondisi ini mencari gerakan yang akan mendukung masuknya kapal ke dalam pusaran air dengan jumlah langkah paling sedikit.

d. Fungsi kelayakan

Fungsi seleksi pada kondisi ini mencari gerakan yang tidak menuju pada kondisi jalan buntu serta tidak dapat ditabrak oleh kapal bajak laut.

e. Fungsi obyektif

Fungsi obyektif pada kondisi ini akan menghasilkan sebuah langkah yang akan meminimumkan jumlah langkah yang diperlukan

untuk mendekati dan masuk ke dalam pusaran air.

3. Strategi *greedy* pada kondisi bebas

Pada kondisi bebas, kapal pemain tidak berada dalam ancaman kecuali di lautan saat itu terdapat kapal bajak laut biru. Kondisi ini biasanya dicapai saat pemain baru memulai permainan, saat kapal pemain telah memasuki pusaran air, atau setelah kapal pemain hancur dan mengurangi jumlah nyawa yang ada. Pada kondisi ini, kapal pemain memiliki keleluasaan untuk memilih arah gerak yang akan menghasilkan posisi terbaik untuk menembak kapal yang berada di dekatnya. Oleh karena itu, strategi *greedy* pada langkah ini bertujuan untuk secepat mungkin menuju ke posisi yang dapat menembak kapal musuh terdekat. Oleh karena itu, strategi *greedy* pada langkah ini dapat disebut sebagai "*greedy by distance to firing position*". Berikut elemen-elemen *greedy* dari teknik ini:

a. Himpunan kandidat aksi C

Himpunan kandidat ini selalu berisi pilihan arah gerak dan pilihan menembak

$C = \{\text{atas, kanan_atas, kanan, kanan_bawah, bawah, kiri_bawah, kiri, kiri_atas, tembak}\}$

b. Himpunan solusi S

Himpunan ini berisi solusi langkah yang akan menghasilkan jarak terdekat menuju ke posisi dimana kapal musuh terdekat berada dalam ruang tembak kapal pemain, atau langkah yang langsung membuat kapal pemain berada di posisi yang diinginkan tersebut.

c. Fungsi seleksi

Fungsi seleksi pada kondisi ini mencari gerakan yang akan mendukung masuknya kapal musuh ke dalam ruang tembak kapal pemain dalam jumlah giliran yang sesedikit mungkin.

d. Fungsi kelayakan

Fungsi seleksi pada kondisi ini mencari gerakan yang tidak menuju pada kondisi jalan buntu serta tidak menuju pada kondisi melarikan diri. Apabila tidak dapat ditemukan posisi menembak kapal terdekat sebelum kapal musuh terdekat tersebut berjarak satu kotak dengan kapal pemain, maka kondisi permainan saat itu berubah menjadi kondisi melarikan diri. Selain itu, apabila terdapat kapal bajak laut biru maka fungsi kelayakan juga harus mempertimbangkan kemungkinan kotak yang dituju ditembak oleh kapal bajak laut biru. Hal ini karena apabila terjadi kesalahan pengambilan keputusan maka akan terjadi kematian tanpa melalui kondisi jalan buntu terlebih dahulu.

e. Fungsi obyektif

Fungsi obyektif pada kondisi ini akan menghasilkan sebuah langkah yang akan meminimumkan jumlah langkah yang diperlukan untuk memasuki posisi menembak terhadap kapal musuh terdekat.

IV. SIMPULAN

Algoritma *greedy* terbukti efektif dalam memandu pengambilan keputusan untuk kasus-kasus tertentu. Pada percobaan yang dilakukan oleh penulis dengan menggunakan strategi *greedy* tersebut, penulis telah menyelesaikan seluruhnya seratus tiga puluh delapan *level* dari permainan Seven Seas Deluxe tersebut. Permainan tidak dapat dilanjutkan karena terdapat kesalahan program dari permainan Seven Seas Deluxe tersebut pada *level* seratus tiga puluh sembilan.

UCAPAN TERIMAKASIH

Pertama, penulis ingin mengucapkan syukur kepada Tuhan Yang Maha Esa atas rahmatNya sehingga makalah ini dapat terselesaikan dengan baik. Penulis juga berterimakasih kepada Bapak Rinaldi Munir selaku dosen pengajar mata kuliah IF2211 Strategi Algoritma dan sumber inspirasi untuk penulisan makalah yang membahas mengenai penerapan algoritma *greedy* dalam permainan Seven Seas Deluxe ini.

REFERENCES

- [1] Munir, Rinaldi. 2018. *Strategi Algoritma*. Bandung: Institut Teknologi Bandung.
- [2] <https://web.stanford.edu/class/archive/cs/cs161/cs161.1138/lectures/13/Small13.pdf> diakses tanggal 14 Mei 2018, pukul 01.13
- [3] <https://www.ea.com/studios/popcap/about> diakses tanggal 14 Mei 2018, pukul 01.47

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, 13 Mei 2018



Samuel Sandi Kristianto Lim, 13516069