

Aplikasi Algoritma Greedy pada Permainan Zuma

Auliya Unnisa Fitri S. / 13509067
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509067@std.stei.itb.ac.id

Abstrak—Zuma merupakan salah satu permainan *arcade puzzle visual matching* yang cukup dikenal. Permainan ini sangat sederhana dan mudah dimainkan. Pemain berupaya memenuhi Zuma meter secepat-cepatnya agar gundu yang harus dihabiskan berhenti dikeluarkan sembari harus menghabiskan gundu sebelum ujung rangkaian gundu masuk ke dalam lubang. Pemenuhan Zuma meter merupakan persoalan optimisasi yang dalam pemecahannya dapat menggunakan algoritma greedy. Dalam studi ini dibahas beberapa strategi berdasarkan pendekatan *greedy* yaitu *greedy by time* dan *greedy by stealing point and coin*. *Greedy by time* berprinsip mengambil rangkaian gundu yang berjumlah paling banyak untuk dijadikan sasaran tembakan gundu pada setiap langkahnya. *Greedy by stealing point and coin* berusaha memenuhi Zuma meter dengan mengincar gundu yang mampu menciptakan pemusnahan beruntun dan menembak koin yang akan memberikan poin bonus. Kedua strategi ini memberikan hasil yang mendekati hasil paling optimal.

Kata kunci—*greedy*, optimisasi, poin, Zuma

I. PENDAHULUAN

Zuma adalah sebuah permainan *arcade puzzle visual-matching* yang dikembangkan oleh PopCap Games dan cukup dikenal. Yang dimaksud dengan *arcade games* adalah semua permainan yang mudah dimengerti, menyenangkan dan grafiknya berwarna-warni yang bertemakan hal-hal yang jenaka dan menarik. Kalimat *arcade* berasal dari *amusement arcades*, dimana permainan-permainan jenis ini biasa dimainkan di jaman dulu. Zuma terdiri dari gundu-gundu berwarna-warni yang datang berturut-turut ke dalam suatu jalur dan harus dimusnahkan. Dua atau lebih gundu berwarna sama yang berurutan dapat dimusnahkan dengan menembakkan gundu yang berwarna sama pula ke arahnya atau menabrak gundu yang berwarna sama. Karena itulah, zuma disebut permainan *puzzle visual-matching*. Permainan Zuma terinspirasi oleh permainan *Puzz Loop* milik Mitchell Corporation. Permainan lain yang sejenis adalah permainan *Luxor*.

Objektif dari permainan Zuma adalah memusnahkan seluruh gundu sebelum pangkal rantai gundu memasuki lubang. Beberapa tampilan permainan Zuma :



Gambar 1. Gundu Berwarna Biru Akan Ditembakkan ke Rangkaian Gundu Berwarna Sama



Gambar 2. Contoh Jalur Gundu yang Memiliki Daerah yang Tidak Bisa Dijangkau Tembakan

Elemen dari permainan Zuma antara lain :

1. Zuma meter. Meteran ini dapat dipenuhi dengan menghasilkan poin.
2. Zuma memiliki katak penembak yang selalu memegang dua gundu. Gundu pertama adalah gundu yang akan ditembakkan dan yang kedua adalah gundu selanjutnya. Kedua gundu ini dapat dipertukarkan. Warna gundu yang akan ditembak akan menyesuaikan dengan warna gundu yang masih tersisa.

3. Jalur yang berisi gundu. Jalur ini berbeda-beda tingkat kesulitannya. Dalam suatu waktu terdapat bagian pada jalur yang sulit atau tidak bisa dicapai tembakan. Oleh karena itu, pemain harus jeli dalam melihat kesempatan menembak. Pada gambar (2), terdapat jalur yang tidak terjangkau tembakan, yaitu bagian menyerupai gua di sisi kanan kiri bagian atas. Bentuk jalur yang menaik memberikan efek berbeda saat pemusnahan gundu. Jika gundu yang musnah berada di bagian bawah rangkaian, maka rangkaian gundu akan merapat ke belakang.
4. Gundu berwarna-warni. Warna gundu dibangkitkan secara acak. Gundu-gundu baru akan terus bertambah sampai Zuma meter penuh. Gundu ini harus dimusnahkan. Pemusnahan gundu akan menghasilkan poin. Selain gundu biasa, terdapat 4 jenis gundu yang memiliki efek istimewa jika ditembak.
 1. gundu yang memiliki ikon lingkaran akan menghanguskan gundu yang berada di sekitarnya. Kita sebut saja gundu bom.
 2. gundu yang memiliki ikon *pause* akan membuat pergerakan gundu melambat untuk sementara. Kita sebut saja gundu *pause*.
 3. gundu yang memiliki ikon tanda panah akan membuat gundu mundur beberapa saat. Kita sebut saja gundu mundur.
 4. gundu terakhir akan memperjelas arah tembakan dengan memancarkan sinar fokus dari mulut katak penembak. Kita sebut saja gundu arah.

Sayangnya keistimewaan gundu ini terbatas waktu. Jika waktunya habis, dia akan kembali menjadi gundu biasa.

5. Lubang. Lubang ini bergambar tengkorak. Jika gundu mencapai lubang ini, permainan akan berakhir.
6. Koin emas yang jika ditembak akan menghasilkan poin tambahan. Koin ini muncul sesekali.

Selain dari penembakan koin, poin tambahan juga bisa didapatkan dengan cara *combo* serta menciptakan celah antar rangkaian dan menembak menembus celah tersebut. Pencarian poin merupakan hal yang penting karena hanya dengan koin lah Zuma meter dapat penuh.

Tujuan dari studi adalah untuk pencarian cara optimal dalam menghasilkan poin untuk memenuhi Zuma meter dengan pendekatan algoritma *greedy*.

II. DASAR TEORI

Dalam permainan Zuma, pemain dikatakan menang apabila semua gundu telah dimusnahkan sebelum pangkal rantai gundu mencapai lubang. Jika dilihat sekilas, yang terpenting dalam permainan Zuma adalah kecepatan dalam memusnahkan gundu. Anggapan ini tidak salah. Akan tetapi, hal yang patut jadi perhatian sebenarnya adalah keadaan Zuma meter. Mengapa?

Alasan utamanya adalah karena Zuma meter lah yang menjadi penentu kapan gundu berhenti dikeluarkan. Sebelum Zuma meter penuh, gundu akan terus dikeluarkan yang warnanya dibangkitkan secara acak. Saat Zuma meter penuh terisi poin, barulah gundu berhenti dikeluarkan dan pemain tinggal menghabiskan sisa gundu yang masih berada di dalam jalur. Jadi, sebagai pemain yang cerdas yang perlu kita lakukan adalah memenuhi Zuma meter. Pemenuhan Zuma meter adalah masalah optimisasi. Masalah optimisasi cukup efektif dipecahkan dengan algoritma *greedy*. Persoalan optimisasi pada pemenuhan Zuma meter tergolong ke masalah maksimasi.

Greedy dalam bahasa Indonesia berarti rakus, tamak, loba. Prinsip *greedy* ialah “*take what you can get now!*”. Algoritma *greedy* membentuk solusi langkah per langkah (*step by step*). Pada setiap langkah, kita membuat pilihan optimum lokal (*local optimum*) dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global (*global optimum*).

Elemen-elemen algoritma *greedy*:

1. Himpunan kandidat, C
Himpunan ini berisi elemen-elemen pembentuk solusi.
2. Himpunan solusi, S
Himpunan bagian dari himunan kandidat yang merupakan solusi atas permasalahan.
3. Fungsi seleksi (*selection function*)
Fungsi yang memakai prinsip *greedy* dalam memilih kandidat.
4. Fungsi kelayakan (*feasible*)
Fungsi yang memeriksa apakah kandidat yang telah dipilih dapat memberikan solusi yang layak atau tidak.
5. Fungsi obyektif
Fungsi yang mengoptimisasi nilai solusi.

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.

Pada kenyataannya, solusi akhir yang dibentuk algoritma *greedy* tidak selalu merupakan solusi optimal. Hal ini terjadi karena algoritma *greedy* tidak beroperasi secara menyeluruh terhadap semua alternatif solusi yang ada, tetapi hanya pada solusi yang terbaik relatif terhadap fungsi seleksi.

Walaupun demikian, algoritma *greedy* sangat cocok digunakan dalam menghasilkan solusi hampiran (*approximation*). Strategi penggunaan algoritma *greedy* tidak hanya satu, namun bisa sangat banyak. Fungsi seleksi yang digunakan pun berbeda-beda. Karena terdapat beberapa fungsi seleksi yang berbeda, kita harus jeli dalam memilih fungsi yang tepat jika kita ingin algoritma menghasilkan solusi yang lebih mendekati optimal dibanding fungsi lainnya.

III. IMPLEMENTASI

Seperti yang disebutkan sebelumnya, pemenuhan Zuma meter dalam permainan Zuma dapat disimulasikan dengan algoritma *greedy*. Di bagian ini akan dibahas bagaimana mencari cara tercepat dalam pencarian poin untuk memenuhi Zuma meter dengan pendekatan *greedy*. Sebelumnya perlu menjadi catatan bahwa pencarian poin dengan cara combo dan penembakan melalui celah dianggap hal yang mudah terjadi sehingga tidak dipentingkan. Alasannya adalah dengan pemanfaatan algoritma *greedy* ini yang bekerja adalah sistem dan sistem mampu menembak dengan tepat asal sasaran tembakan berada dalam jangkauan tembakan.

Terdapat banyak cara dalam mengimplementasikan prinsip *greedy*. Kali ini akan diberikan dua contoh penggunaan algoritma *greedy* yaitu :

1. *Greedy by Time*
2. *Greedy by Stealing Point and Coin*

A. *Greedy By Time*

Strateginya adalah bagaimana memusnahkan gundu secepat-cepatnya. Pemusnahan gundu akan memberikan poin standar. Logikanya walaupun poin yang didapatkan tidak terlalu banyak tetapi jika cepat maka Zuma meter akan cepat penuh.

Pada setiap langkah, sistem akan mencatat jumlah setiap rangkaian gundu berwarna dan letaknya berurutan serta letak gundu-gundu istimewa. Rangkaian di sini berarti dua atau lebih gundu. Rangkaian berwarna gundu tembakan yang memiliki gundu istimewa memiliki prioritas yang lebih tinggi dibanding yang berjumlah paling banyak. Sedang gundu tunggal, gundu yang tidak berdekatan dengan gundu yang berwarna dengannya, memiliki prioritas terendah.

Gundu istimewa juga memiliki prioritas. Gundu bom memiliki prioritas dibanding ketiga gundu istimewa lainnya karena mampu memusnahkan gundu dalam jumlah besar. Gundu *pause* dan gundu mundur memiliki prioritas yang sama karena sama-sama memberikan waktu ekstra untuk sistem. Gundu arah tidak dipedulikan karena tidak memiliki keuntungan bagi sistem (sistem mampu mencatat posisi gundu dengan tepat).

Sistem akan menembakkan gundu ke rangkaian yang memiliki prioritas paling tinggi.

Elemen-elemen *greedy* :

1. Himpunan kandidat
Seluruh rangkaian gundu dan gundu tunggal yang berada di dalam jalur. Setiap rangkaian gundu dan gundu tunggal memiliki nilai berdasar prioritasnya.
2. Himpunan solusi
Kumpulan dari kandidat yang memiliki nilai paling besar.
3. Fungsi seleksi
Memilih salah satu dari himpunan kandidat yang berwarna sama dan meng-assign nilainya berdasar prioritas.

4. Fungsi kelayakan

Memeriksa apakah kandidat bisa ditembak atau tidak. Artinya kandidat tidak berada di jalur yang sulit atau tidak bisa dicapai tembakan.

5. Fungsi obyektif

Memilih kandidat yang memiliki nilai terbesar. Jika terdapat lebih dari satu kandidat, maka kandidat dipilih secara acak.

B. *Greedy By Stealing Point and Coin*

Strateginya adalah bagaimana memusnahkan gundu dengan memperhatikan letak koin emas. Penembakan koin emas akan memberikan poin bonus yang cukup besar. Dengan banyak menembak koin emas, Zuma meter akan penuh dengan cepat. *Stealing point* di sini maksudnya mencuri poin dari menciptakan pemusnahan beruntun yang terjadi langsung setelah pemusnahan dari gundu tembakan (*consecutive*). Contohnya dalam suatu bagian terdapat 2 gundu merah, 2 gundu hijau, 1 gundu merah, berurutan. Jika gundu hijau dimusnahkan, gundu-gundu merah di kedua sisinya akan turut musnah karena terjadi tabrakan 3 atau lebih gundu berwarna sama. Pemusnahan karena tabrakan juga menyumbang poin.

Pada setiap langkah, sistem akan mencatat jumlah setiap rangkaian gundu berwarna dan letaknya berurutan dan mana yang akan menciptakan pemusnahan beruntun dan mana yang mengandung gundu bom. Gundu istimewa selain gundu bom dianggap gundu biasa karena tidak memberikan keuntungan yang besar bagi sistem. Selain itu, sistem juga akan mencatat letak koin emas. Perlu menjadi catatan bahwa koin ini muncul sesekali sehingga letak koin emas bisa *null*.

Koin emas memiliki prioritas tertinggi, disusul oleh rangkaian yang mengandung gundu bom. Prioritas selanjutnya dihitung berdasar jumlah gundu dalam rangkaian. Jika suatu rangkaian bisa menciptakan pemusnahan beruntun, jumlah rangkaiannya dihitung dari jumlah rangkaian sebenarnya ditambah jumlah gundu di sisi kanan dan kiri yang berwarna sama yang bisa musnah setelahnya. Gundu tunggal memiliki prioritas terendah.

Sistem akan menembakkan gundu ke rangkaian yang memiliki prioritas paling tinggi.

Elemen-elemen *greedy* :

1. Himpunan kandidat
Koin, seluruh rangkaian gundu dan gundu tunggal yang berada di dalam jalur. Masing-masing koin, rangkaian gundu dan gundu tunggal memiliki nilai berdasar prioritasnya.
2. Himpunan solusi
Kumpulan dari kandidat yang memiliki nilai paling besar.
3. Fungsi seleksi
Memilih salah satu dari himpunan kandidat baik itu koin maupun rangkaian gundu atau gundu tunggal yang berwarna sama dan meng-assign nilainya berdasar prioritas.

4. Fungsi kelayakan
Memeriksa apakah kandidat bisa ditembak atau tidak. Artinya kandidat tidak berada di jalur yang sulit atau tidak bisa dicapai tembakan.
5. Fungsi obyektif
Memilih kandidat yang memiliki nilai terbesar. Jika terdapat lebih dari satu kandidat, maka kandidat dipilih secara acak.



Gambar 3. Efek dari Penembakan Gundu Bom

IV. STUDI KASUS

Kedua contoh strategi pemenuhan Zuma meter dengan pendekatan greedy memberikan hasil yang berbeda. Mari kita lihat dalam studi kasus di bawah.

Pada awal permainan :

Dengan memakai *greedy by time*, sistem akan langsung memuntahkan tembakan ke rangkaian gundu yang memiliki jumlah paling banyak karena gundu-gundu istimewa biasanya belum keluar. Zuma meter masih terisi sedikit.

Dengan memakai *greedy by stealing point and coin*, sistem bereaksi hampir sama seperti *greedy by time* karena gundu bom dan koin belum muncul. Perbedaannya hanyalah jika ada rangkaian yang terjepit antara dua rangkaian panjang gundu yang berwarna di sisi kanan dan kirinya maka rangkaian yang terjepit itu akan ditembak terlebih dahulu.

Pada tengah permainan :

Dengan *greedy by time*, tujuan tembakan yang utama adalah rangkaian yang memiliki gundu-gundu istimewa, disusul kemudian rangkaian yang panjang. Gundu-gundu tunggal terpinggirkan bahkan yang berada di dekat lubang dibiarkan saja. Hal ini memungkinkan sistem kalah karena ada gundu yang tidak langsung ditangani dan berakhir di lubang. Zuma meter terisi sedikit demi sedikit.



Gambar 4. (kiri atas) Contoh Koin yang Berada di luar Jangkauan Tembakan. (kanan bawah) Contoh Koin yang Berada di dalam Jangkauan Tembakan

Dengan *greedy by stealing point and coin*, sistem mengutamakan penembakan koin. Jika koin berada dalam jangkauan tembakan, koin tersebut langsung ditembak. Dengan begini, isi dari Zuma meter bertambah lumayan banyak. Jika koin tidak berada dalam jangkauan tembakan, sistem akan mencari rangkaian yang mengandung gundu bom. Ilustrasi letak koin dapat dilihat pada gambar (4) di atas. Jika tidak ada juga, sistem akan menembak rangkaian yang mampu menciptakan pemusnahan beruntun. Contoh rangkaian yang mampu menciptakan pemusnahan beruntun jika ditembak terdapat pada gambar (5). Pemusnahan beruntun mengisi Zuma meter lebih cepat dari pemusnahan biasa atau tunggal. Kelemahannya sama dengan *greedy by time*, yaitu gundu-gundu tunggal yang berada di dekat lubang terpinggirkan dan berpotensi membuat sistem kalah karena jatuh ke dalam lubang tanpa sempat ditangani.



Gambar 5. Rangkaian Gundu yang Jika Ditembak Akan Menghasilkan Pemusnahan Beruntun

Setelah Zuma meter terisi penuh :

Dengan algoritma *greedy by time*, tugas sistem tinggal memusnahkan gundu-gundu yang tersisa. Seharusnya tidak ada masalah yang berarti selain adanya gundu tunggal yang berada di dekat lubang yang masih jadi prioritas terakhir dan belum ditangani.

Dengan algoritma *greedy by stealing point and coin*, sistem akan lebih cepat menyelesaikan permainan dari *greedy by time* karena berpikir selangkah ke depan. Gundu tunggal juga menjadi masalah. Namun jika terdapat dua atau lebih gundu sewarna gundu tunggal yang hanya dipisahkan oleh suatu rangkaian gundu, hal ini tidak lagi menjadi masalah karena sistem akan menembak rangkaian gundu yang menjadi pemisah keduanya. Lalu gundu tunggal akan bertabrakan dengan rangkaian gundu sewarna dan akhirnya ikut musnah.

V. ANALISIS

Kedua contoh strategi pemenuhan Zuma meter dengan pendekatan *greedy* memberikan hasil yang berbeda. Dari studi kasus diatas terlihat algoritma mana yang memberikan hasil lebih optimal. Algoritma *greedy by stealing point and coin* memberikan hasil yang lebih optimal dibanding algoritma *greedy by time*.

Walaupun permainan Zuma mengandalkan strategi yang jitu, penggunaan kedua algoritma *greedy* di atas terkadang tidak memberikan hasil yang diinginkan. Contohnya saja sistem yang bisa saja kalah karena tidak mempedulikan gundu tunggal yang berada di dekat lubang. Seharusnya gundu yang berada di dekat lubang juga turut mendapat penanganan khusus.



Gambar 6. Gundu-gundu Tunggal yang Belum Tertangani Berpotensi Menyebabkan Kekalahan Sistem

Agar permainan berjalan lebih mulus, diperlukan penanganan gundu berdasar kedekatan posisinya dengan lubang serta mekanisme setelah Zuma meter terisi penuh. Perlakuan sebelum Zuma meter terisi penuh dan sesudah Zuma meter terisi penuh seharusnya sedikit berbeda karena obyektifnya bukanlah pemenuhan Zuma meter lagi, melainkan pemusnahan untuk menghabiskan gundu yang tersisa.

VI. KESIMPULAN

Pemecahan masalah optimisasi dapat diselesaikan dengan cukup baik melalui pendekatan *greedy*. Hasil solusi yang didapatkan cenderung mendekati hasil optimal. Keampuhan algoritma *greedy* terletak pada pemilihan fungsi seleksi yang tepat, dari sekian banyak fungsi seleksi yang bisa digunakan.

Pengimplementasian algoritma *greedy* sangat sederhana. Jika sudah memiliki gambaran akan suatu penyelesaian, penentuan elemen pun sangat mudah. Agar mendapat hasil yang lebih baik, sebelumnya pikirkan dahulu segala kondisi yang mungkin dihadapi dengan matang. Hal ini bertujuan untuk menghindari kecerobohan-kecerobohan kecil yang bisa membuat sistem kecolongan. Meskipun demikian, kita tidak bisa memutuskan suatu cara benar-benar akan memberikan hasil paling optimal tanpa melalui perhitungan secara matematis. Untuk kasus sederhana seperti pemenuhan Zuma meter pada permainan Zuma yang cukup diberi hasil aproksimasi, penggunaan algoritma *greedy* sudah cukup memenuhi target.

REFERENCES

- [1] http://en.wikipedia.org/wiki/Zuma_%28video_game%29. Waktu akses 4 Desember 2011
- [2] <http://zone.msn.com/en/zuma/article/zumatips.htm>. Waktu akses 4 Desember 2011
- [3] Munir, Rinaldi. 2009. *Strategi Algoritma*. Bandung : Penerbit ITB.

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

Auliya Unnisa Fitri S.
13509067