

Penerapan Algoritma Greedy Pada Permainan Othello

Rian Permata Putra¹, Dadan Ramdan Mangunpraja², Satrio Ajie Wijaya³

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if14083@students.if.itb.ac.id¹,
if14087@students.if.itb.ac.id², if14092@students.if.itb.ac.id³

Abstrak

Salah satu aplikasi *software* yang paling populer adalah *game*. Beragam jenis *game* dengan tampilan yang menarik dan tingkat kesulitan yang menantang membuat *game* menjadi *software* yang digemari oleh hampir seluruh kalangan. Kemampuan komputer dalam menyelesaikan *game* (mencapai *win state*) terkait dengan *Artificial Intelligent*, yang terkait pula dengan strategi pemilihan algoritmanya. Semakin bagus *Artificial Intelligent*-nya, *game* menjadi semakin menantang sehingga membuatnya menjadi sangat menarik.

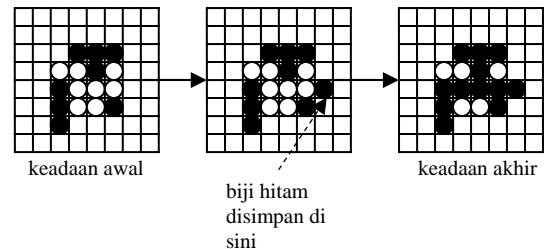
Salah satu *game* yang menuntut *Artificial Intelligent* yang bagus adalah “Othello”. Pada makalah ini, penulis membahas analisis penggunaan algoritma *greedy* sebagai salah satu cara membangun *Artificial Intelligent* yang cukup bagus untuk menjalankan *game* tersebut.

Kata kunci: *Artificial Intelligent, Othello, Algoritma Greedy,*

1. Pendahuluan

Game/permainan merupakan hal yang mengiringi peradaban manusia di suatu daerah. Hampir setiap daerah memiliki jenis permainan tradisionalnya masing-masing (dikenal dengan *game*/permainan tradisional). Permainan tersebut berkembang karena adanya unsur kreativitas manusia itu sendiri.

Salah satu *game* tradisional yang cukup terkenal adalah *game Othello* yang berasal dari daerah Jepang. *Game* ini dimainkan oleh dua orang. Tempat permainannya berupa suatu papan dengan kotak berjumlah 8x8. Masing-masing pemain memiliki biji dengan warna yang berbeda, satu pemain berwarna hitam dan yang lainnya berwarna putih. Dalam *game* tersebut, tiap pemain berusaha untuk mengganti warna biji lawan menjadi warna biji miliknya dengan menjepit/memblok biji-biji musuh baik itu secara vertikal, diagonal, maupun horizontal (contoh, lihat gambar 1.1).



Gambar 1.1

Seiring dengan perkembangan zaman dan kemajuan teknologi, kini permainan tradisional banyak yang sudah diimplementasikan dalam bentuk *game* komputer, sehingga dibutuhkan tiruan kecerdasan (*Artificial Intelligent*) agar komputer mampu menjalankan *game-game* tersebut sesuai aturan aslinya. Dalam makalah ini, penulis ingin menjelaskan mengenai penerapan algoritma *greedy* dalam permainan *Othello*, yaitu penjelasan mengenai cara bagaimana komputer berusaha untuk mencari posisi terbaik agar daerah yang dikuasainya aman dan agar melakukan langkah ke posisi terbaik sehingga komputer memiliki kemungkinan besar untuk memenangkan permainan.

2. Algoritma Greedy

Algoritma *greedy* merupakan algoritma sederhana dimana penyelesaian dilakukan dengan cara yang *straight forward* dan membentuk solusi langkah per langkah (*step by step*). Algoritma ini biasanya digunakan untuk perolehan nilai optimum dari suatu kasus. Dalam penyelesaian masalahnya, algoritma *greedy* mengambil langkah yang memberikan pilihan paling optimum pada saat itu (prinsip *take what you can get now*), walaupun mungkin pilihan tersebut bukan pilihan optimum pada saat penyelesaian akhir kasus. Setelah solusi dipilih diharapkan solusi yang diperoleh merupakan solusi optimum dari penyelesaian seluruh kasus.

3. Penerapan Algoritma

Ide penerapan algoritma *greedy* dalam *game* ini adalah dengan membuat prioritas daerah yang harus dikuasai terlebih dahulu. Untuk setiap prioritas, jika langkah-langkah yang diperiksa mungkin untuk dilakukan, maka masukkan informasi koordinat (langkah) beserta informasi jumlah biji hitam yang berubah menjadi putih jika langkah tersebut dijalankan ke dalam sebuah media penyimpanan (misalnya *array*), kemudian cari mana langkah yang menghasilkan daerah “kekuasaan” terbanyak di antara semua langkah pada *array* tersebut dan jadikan itu langkah solusi pada saat itu. Jika tidak ada satupun langkah yang mungkin dilakukan, maka lanjut ke prioritas berikutnya (kecuali untuk prioritas kelima, karena jika langkah prioritas kelima tidak ada yang mungkin, berarti *game* sudah berakhir atau mengalami “langkah mati”).

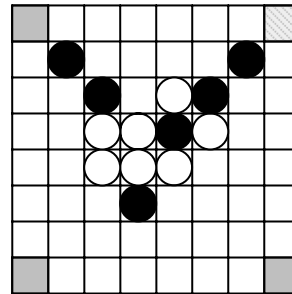
Tetapan-tetapan algoritma *greedy*-nya yaitu :

- Himpunan Kandidat = semua daerah /koordinat.
- Himpunan Solusi = Penguasaan daerah yang “aman” dan terbanyak.
- Fungsi Seleksi = prioritas daerah lebih tinggi, lalu penguasaan daerah terbanyak.
- Fungsi Kelayakan = daerah yang bisa ditempati (yang bisa memblok biji lawan secara horizontal, vertikal, atau diagonal).
- Fungsi Objektif = Koordinat yang menghasilkan daerah aman dan blokade terbanyak

Berikut adalah prioritas daerah yang dimaksud di atas (kotak yang dihitamkan adalah daerah prioritas yang dimaksud, anggap komputer memainkan biji putih) :

- Prioritas pertama

Periksa apakah bisa menaruh biji putih di setiap daerah sudut, yaitu koordinat-koordinat $\{(1,1), (8,8), (1,8), \text{ dan } (8,1)\}$. Daerah-daerah ini menjadi prioritas utama karena jika daerah ini bisa dikuasai, maka tidak mungkin lagi dikuasai lawan. Selain itu akan membuat daerah-daerah lain relatif lebih mudah dikuasai. (lihat contoh pada gambar 3.1)

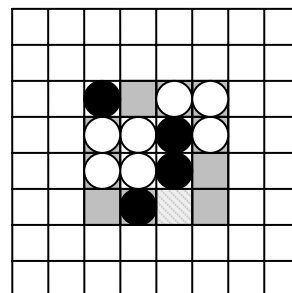


Ada 2 koordinat yang bisa ditempati, yaitu (1,1) dan (8,1). Ambil koordinat (8,1) (yang diarsir) karena memblok biji hitam lebih banyak.

Gambar 3.1

- Prioritas kedua

Periksa apakah bisa menaruh biji putih di daerah sekitar tengah, yaitu yang dibatasi oleh koordinat-koordinat $\{(3,3), (6,3), (3,6), \text{ dan } (6,6)\}$ (lihat contoh pada gambar 3.2).

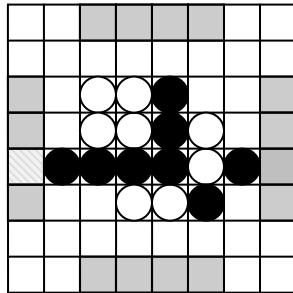


Hanya ada 1 koordinat yang bisa ditempati, yaitu (5,6) (yang diarsir), jadi pilih koordinat tersebut.

Gambar 3.2

- Prioritas ketiga

Periksa apakah bisa menaruh biji putih di daerah-daerah tepi selain sekitar sudut-sudut.

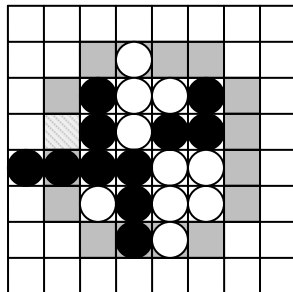


Ada 2 koordinat di daerah prioritas yang bisa ditempati, yaitu (1,5) dan (8,5).
Ambil koordinat (1,5) (yang diarsir) karena memblok biji hitam lebih banyak

Gambar 3.3

- Prioritas keempat

Periksa apakah bisa menaruh biji putih di daerah dalam, sisa daerah selain daerah-daerah sekeliling sudut-sudut. (contoh, lihat gambar 3.4).

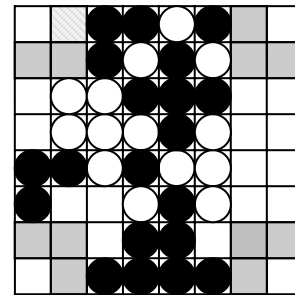


Ada 9 koordinat di daerah prioritas yang bisa ditempati, yaitu (3,2), (2,3), (2,4), (2,6), (3,7), (6,7), (7,5), (7,4), (7,3), dan (6,2). Ambil koordinat (2,4) (yang diarsir) karena memblok biji hitam lebih banyak.

Gambar 3.4

- Prioritas kelima (terakhir)

Periksa daerah-daerah sekeliling sudut-sudut. Daerah-daerah ini memiliki prioritas paling rendah karena jika daerah ini diisi, maka lawan akan lebih mudah menguasai daerah yang berprioritas tinggi (contoh, lihat gambar 3.5).



Ada 3 koordinat di daerah prioritas yang bisa ditempati, yaitu (2,1), (2,2), dan (7,1).
Ambil koordinat (2,1) (yang diarsir) karena memblok biji hitam lebih banyak.

Gambar 3.5

4. Hasil Analisis

Ada beberapa catatan yang harus diperhatikan, yaitu mengenai beberapa kelemahan algoritma ini. Kelemahan yang paling terlihat antara lain :

- Algoritma tidak dipersiapkan untuk “berpikir menyerang”, yaitu mendesak lawan agar membuat langkah yang salah (dalam artian mengisi daerah berprioritas rendah), sedangkan lawan (manusia) mungkin bisa berpikir untuk mendesak.
- Seperti kelemahan umum algoritma *greedy*, yaitu hanya mencari solusi terbaik saat itu, padahal belum tentu terbaik untuk langkah berikutnya.
- Algoritma masih terlalu sederhana, yakni masih bisa dibayangkan dengan relatif mudah oleh lawan (manusia) sehingga masih mungkin “dipatahkan” oleh orang yang menjalankan *game* dengan algoritma/cara serupa.

Di sisi lain, algoritma ini pun mempunyai kelebihan yakni kompleksitas waktu dan ruangnya rendah ($O(n)$), sehingga komputer sangat kecil kemungkinan mengalami masalah dalam lamanya waktu pencarian langkah atau masalah ketersediaan memori. Selain itu, secara umum alur permainan memang menggunakan algoritma ini, jadi solusi terbaik saat itu kemungkinan besar akan menghasilkan solusi yang baik pula untuk langkah berikutnya.

5. Kesimpulan

Algoritma *Greedy* adalah algoritma yang relatif cukup mangkus dalam menyelesaikan suatu kasus karena algoritma ini hanya menemukan solusi terbaik pada saat itu saja walaupun solusi tersebut belum tentu merupakan suatu solusi

yang optimum untuk penyelesaian kasus secara keseluruhan.

Hasil analisis yang diperoleh dari penggunaan algoritma *greedy* dalam permainan *othello* menunjukkan bahwa algoritma ini masih memiliki beberapa kelemahan yaitu pergerakan komputer lebih cenderung pada gerakan bertahan, solusi untuk pergerakan komputer yang diperoleh pada saat itu belum tentu merupakan solusi terbaik untuk keseluruhan permainan, dan pemain lawan cenderung mudah untuk mencari celah agar memperoleh kemenangan. Namun pada umumnya memang bagus untuk melawan pemain yang tidak sangat cerdas.

6. Pustaka

- [1] Munir, Rinaldi. 2005. Strategi Algoritmik. Teknik Informatika ITB : Bandung
- [2] www.freegames.ws/games/boardgames/othello/othello.htm
- [3] boardgamecentral.com/games/reversi_online.html