

Penerapan Algoritma Greedy dalam Permainan Othello

Fabrian Oktavino H - 13510053
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
fabrian.oktavino@students.itb.ac.id

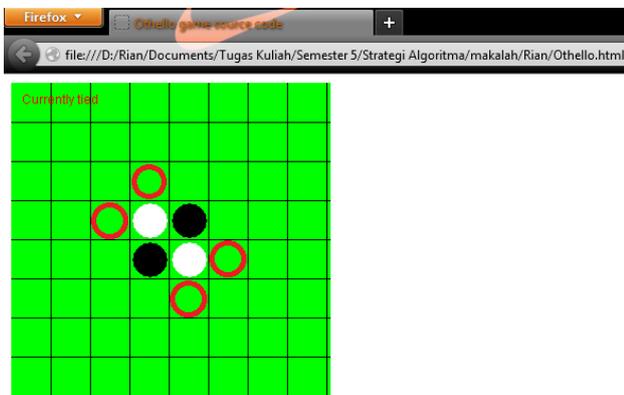
Abstract—Algoritma *Greedy* adalah algoritma pembentuk solusi langkah per langkah. Pada setiap langkah itu akan dipilih keputusan yang paling optimal. Keputusan yang diambil tersebut tidak memperdulikan keputusan selanjutnya yang akan diambil, dan keputusan tidak dapat diubah lagi pada langkah selanjutnya.

Othello adalah permainan menggunakan papan berisi kotak sebanyak 8x8 yang dimainkan oleh dua orang menggunakan keping hitam dan putih. Permainan ini bertujuan untuk memiliki keping paling banyak untuk menjadi pemenang.

Index Terms—Greedy, Othello

I. PENDAHULUAN

Othello adalah permainan yang menggunakan papan berisi kotak sebanyak 8x8, antara dua orang pemain dengan keping hitam dan putih seperti di bawah.

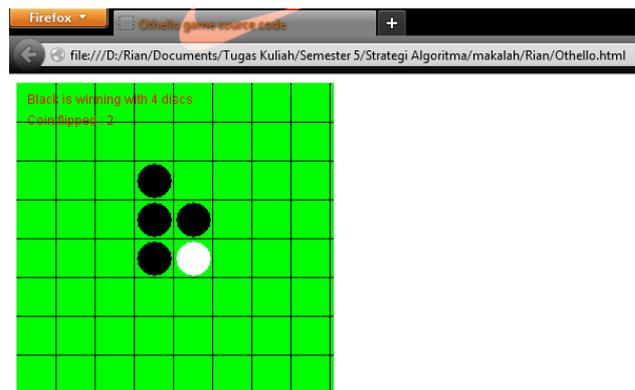


Gambar 1 Othello Board Game

Aturan permainannya adalah sebagai berikut:

1. Permainan dimulai dari posisi papan dengan susunan keping hitam dan putih seperti pada gambar di atas, yaitu dua keping hitam dan dua keping putih tepat di tengah-tengah papan dengan posisi saling memotong secara diagonal, dengan keping hitam miring ke kanan-atas dan putih miring ke kiri-atas.

2. Pemain hitam melangkah pertama kali dengan meletakkan keping hitam di kotak kosong di mana dia bisa mengagap keping putih di antara dua keping hitam, yaitu di antara keping hitam yang sudah ada di papan dan keping hitam yang baru diletakkan, boleh mengagap secara horizontal, vertikal maupun diagonal. Pada Gambar 1 diatas keping hitam bisa melangkah di kotak yang berisikan lingkaran merah. Semisal hitam melangkah seperti gambar berikut :



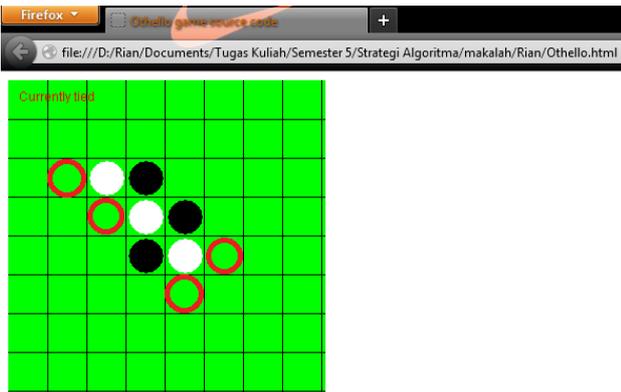
Gambar 2 Othello Board Game

3. Keping putih yang terjepit pada no. 2 di atas, dibalik semua menjadi keping hitam.
4. Berikutnya giliran pemain putih melangkah dengan meletakkan keping putih di kotak kosong di mana dia bisa mengagap keping hitam di antara dua keping putih, yaitu di antara keping putih yang sudah ada di papan dan keping putih yang baru diletakkan, boleh mengagap secara horizontal, vertikal maupun diagonal. Contoh seperti gambar berikut :

II. DASAR TEORI

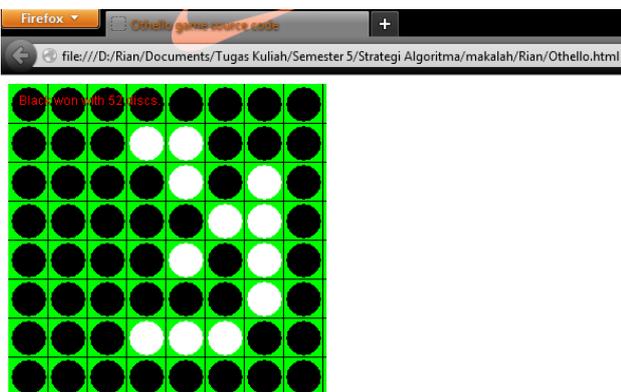
Algoritma *Greedy* adalah salah satu algoritma yang dapat digunakan untuk mendapatkan solusi terbaik dan merupakan algoritma yang paling populer dalam hal ini. Secara Harafiah *Greedy* artinya rakus atau tamak, sifat yang berkonotasi negatif. Orang yang memiliki sifat ini akan mengambil sebanyak mungkin atau mengambil yang paling bagus atau yang paling mahal. Sesuai dengan arti tersebut, Prinsip Greedy adalah take what you can get now. Dalam kehidupan sehari-hari *Greedy* dapat digunakan dalam masalah seperti :

- Memilih beberapa jenis investasi
- Mencari jalur tersingkat



Gambar 3 Othello Game Board

5. Keping hitam yang terjepit pada no. 4 di atas, dibalik semua menjadi keping putih.
6. Demikian kedua pemain bergantian saling meletakkan kepingnya di kotak kosong di mana dia bisa mengagip keping lawannya, dan membalik keping lawannya menjadi kepingnya.
7. Kedua pemain tidak boleh meletakkan kepingnya di kotak yang sudah terisi, atau di kotak kosong di mana dia tidak mengagip keping lawannya. Pada contoh gambar di atas, pemain keping hitam tidak boleh menaruh kepingnya di atas keping hitam, keping putih, dan di luar kotak yang tidak berisi lingkaran merah.
8. Apabila pemain tidak mempunyai kotak di mana dia bisa melangkah, maka dia harus 'pass' yaitu memberikan gilirannya melangkah kepada lawannya.
9. Apabila kedua pemain sama-sama tidak mempunyai kotak di mana dia bisa melangkah, biasanya ketika papan sudah penuh, maka permainan selesai.



Gambar 4 Othello Board Game

10. Pemain dengan jumlah keping terbanyak adalah pemenang.

1.1 Definisi Algoritma Greedy

Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah demi langkah. Pada setiap langkah :

1. Mengambil pilihan terbaik yang dapat diperoleh pada saat itu.
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah maka akan mendapatkan optimum global.

Algoritma *greedy* mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global. Prinsip dari algoritma *greedy* adalah "take what you can get now". Prinsip ini juga digunakan dalam pemecahan masalah optimasi. Algoritma *greedy* merupakan metode yang paling populer untuk memecahkan persoalan optimasi.

Persoalan optimasi (*optimization problems*) adalah persoalan yang menuntut pencarian solusi optimum. Persoalan optimasi sendiri ada dua macam, yaitu : Maksimasi (*maximization*) dan Minimasi (*minimization*). Solusi optimum (terbaik) adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin, sesuai yang dibutuhkan.

1.2 Skema umum Algoritma Greedy

Algoritma *greedy* disusun oleh elemen-elemen berikut:

a. Himpunan kandidat

Berisi elemen-elemen pembentuk solusi. Contoh di dalam kasus permainan Othello ialah dimana kita akan menaruh keping milik kita.

b. Himpunan solusi

Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Contoh di dalam kasus permainan Othello ialah kotak mana saja yang diperbolehkan untuk menaruh keping milik kita. Dari contoh barusan, bisa dilihat bahwa himpunan solusi merupakan himpunan bagian dari himpunan kandidat.

c. Fungsi seleksi (selection function)

Memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Dalam kasus yang akan dibahas pada makalah ini, fungsi seleksinya ialah menentukan keping akan ditaruh dimana agar keping lawan yang terbalik memberikan jumlah yang paling besar (aturan permainan pada bab II).

d. Fungsi kelayakan (feasible)

Memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (constraints) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi. Dalam contoh permainan Othello ialah apabila suatu keping ditaruh di sebuah kotak, apakah ada keping lawan yang terbalik.

e. Fungsi obyektif

fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain). Contoh fungsi obyektif pada kasus yang akan dibahas di makalah ini ialah jumlah keping maksimum yang terbalik.

```
final int BLACK = 0
final int WHITE = 1

procedure AIbyGreedy(color : integer, othercolor :
integer)
{
  int xtemp = 0
  int ytemp = 0
  int flippedtemp = 0

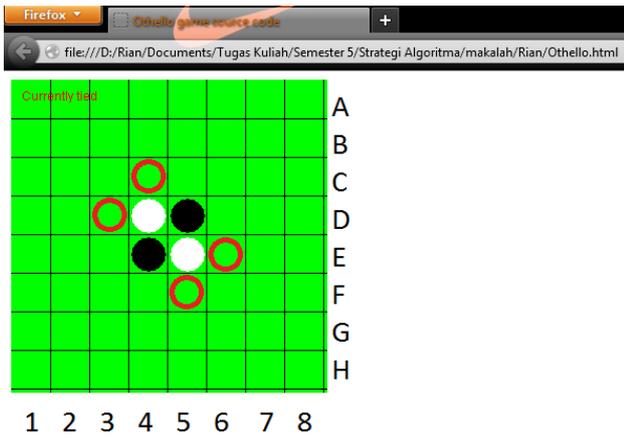
  for (int j = 1; j <= 8; j++)
  {
    for(int i = 1; i <=8; i++)
    {
      if (legalMove(i,j,color,othercolor))
      {
        if (flipped(i,j,color,othercolor) > flippedtemp)
        {
          flippedtemp := flipped(i,j,color,othercolor)
          xtemp := i
          ytemp := j
        }
      }
    }
    j := j+1
  }
}
```

Prosedur AIbyGreedy merupakan prosedur agar AI menjalankan gilirannya menggunakan algoritma *greedy*. *color* merupakan warna keping yang digunakan AI, dan *othercolor* merupakan warna keping yang digunakan oleh lawannya. Apabila AI menggunakan keping warna hitam, maka *color* = BLACK, dan *othercolor* adalah WHITE. *xtemp* merupakan penampung sementara posisi x dari kotak (x,y), *ytemp* merupakan penampung sementara posisi y, dan *flippedtemp* merupakan penampung sementara jumlah keping yang dapat dibalikkan pada kotak (x,y) tersebut. Fungsi *legalMove* mengembalikan boolean dan berfungsi untuk mengecek apakah pada posisi kotak (i,j) keping berwarna *color* boleh ditaruh pada posisi tersebut. Apabila dapat ditaruh pada posisi tersebut, maka fungsi *legalMove* akan mengembalikan nilai *true*. Sedangkan fungsi *flipped* akan mengembalikan jumlah berapa banyak keping lawan (*othercolor*) yang dibalikkan.

Berikut adalah implementasi algoritma *greedy* untuk AI dalam bentuk program :

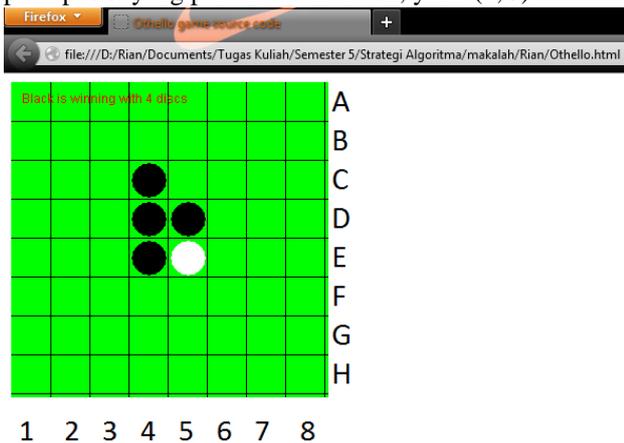
III. IMPLEMENTASI DAN ANALISIS PENGUJIAN

Pada permainan Othello, salah satu algoritma *greedy* yang dapat diterapkan ialah *greedy* dengan pembalikan keping terbanyak. Jadi, algoritma yang kita gunakan mengecek untuk setiap kotak yang bisa ditaruh keping kita, apakah memiliki jumlah pembalikan keping terbanyak dibandingkan dengan kotak yang lainnya. Kita akan membandingkan jumlah pembalikan antara satu kotak dengan kotak lainnya. Kotak dengan jumlah terbanyak akan kita pilih. Pencarian dimulai dari kotak (1,1) yaitu kotak paling ujung kiri atas, kemudian dicari ke kotak (1,2), (1,3) ... hingga (1,8), lalu mengecek dari kotak (2,1), (2,2) ... dst. Apabila sudah menemukan satu kotak pertama yang dapat kita taruh keping kita, program akan mengecek apakah jumlah keping yang dibalik apabila keping ditaruh di kotak tersebut memiliki nilai yang lebih besar daripada nilai di temporary. Apabila *true*, maka jumlah keping yang dibalik akan dimasukkan ke *temporary* menggantikan nilai yang lama, kemudian mencatat posisi dari kotak tersebut. Kemudian program akan mengecek lagi kotak berikutnya yang mungkin ditaruh keping kita, lalu melakukan pengecekan seperti cara di atas. Berikut *pseudo-code* dari algoritma yang diimplementasikan :



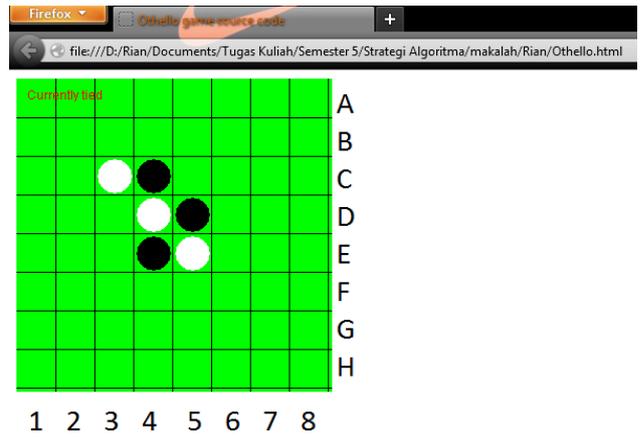
Gambar 5 Awal Game

Pertama, AI (keping hitam) akan mengecek dari (1,A) apakah dapat menaruh keping disana, dikarenakan keping tidak dapat ditaruh disana, maka AI akan mengecek lagi pada posisi (2,A). Seterusnya hingga AI akan mengecek pada posisi (4,C). Karena pada posisi (4,C) keping hitam dapat ditaruh, maka AI akan mengecek apakah jumlah keping putih yang bisa di balik melebihi jumlah yang disimpan dalam *flippedtemp*. Karena ini baru pengecekan pertama kali, secara pasti *flippedtemp* berisi 0, dan karena jumlah keping putih yang dapat dibalik yaitu 1 lebih besar dari 0, maka posisi angka 1 akan dimasukkan ke *flippedtemp*, dan posisi (4,C) akan disimpan juga. Kemudian AI akan mengecek lagi hingga ke posisi (3,D), dan mengecek jumlah keping putih yang dapat dibalik apakah lebih besar dari *flippedtemp*. Karena ternyata tidak lebih besar, maka pengecekan dilanjutkan. Begitu seterusnya hingga sampai di posisi (8,H). Karena tidak ditemukan jumlah pembalikkan yang lebih besar, maka keping akan ditaruh pada posisi yang pertama kali di cek, yaitu (4,C).



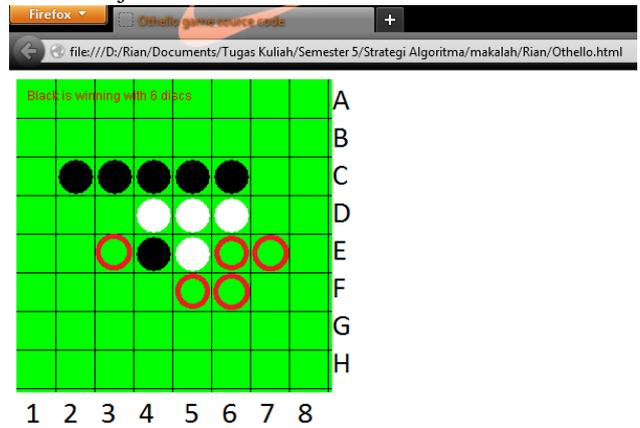
Gambar 6 Keping Hitam pada posisi (4,C)

Kemudian putih (user) berjalan pada posisi (3,C).



Gambar 7 Keping putih pada posisi (3,C)

Pada kejadian di bawah ini :



Gambar 8 Keping Hitam ditaruh dimana?

Pertama, AI akan melakukan pengecekan jumlah keping putih yang dibalik pada (3,E), karena ini adalah pengecekan pertama, maka tentu posisi ini akan disimpan, dengan *flippedtemp* sekarang adalah 1. Kemudian AI akan melakukan pengecekan berikutnya pada posisi (6,E). Dari hasil perhitungan didapat bahwa jumlah keping yang dibalik adalah 3. Jumlah keping yang dibalik pada posisi (6,E) lebih banyak daripada jumlah keping yang dibalik pada posisi (3,E) sehingga *flippedtemp* sekarang akan di-assign dengan angka 3, dan posisi (6,E) disimpan dalam *xtemp* dan *ytemp*. Kemudian AI akan melanjutkan pengecekan pada (7,E), (5,F), dan (6,F) yang masing – masing membalik keping sebanyak 1, 2, dan 2. Tidak ada jumlah keping yang dibalik lebih besar daripada posisi (6,E), sehingga keping hitam akan menaruh keping di (6,E).

IV. KESIMPULAN

Algoritma *greedy* untuk membalikkan keping paling banyak dapat diterapkan dalam permainan othello. Namun penggunaan algoritma ini tidak menjamin kemenangan yang pasti, karena tidak memikirkan konsekuensi ke depannya. Apabila lawan cukup pandai, maka bisa saja AI masuk ke dalam jebakan lawan, karena kita hanya melihat jumlah keping yang dibalik pada langkah tersebut, tanpa memikirkan langkah ke depannya.

REFERENCES

- [1] <http://zafnatpanyah.blogspot.com/2010/07/algoritma-greedy.html>
- [2] <http://ngelawak-gan.blogspot.com/2011/12/algoritma-greedy.html>
- [3] Munir, Rinaldi.2009.*Strategi Algoritma*.Bandung: ITB.

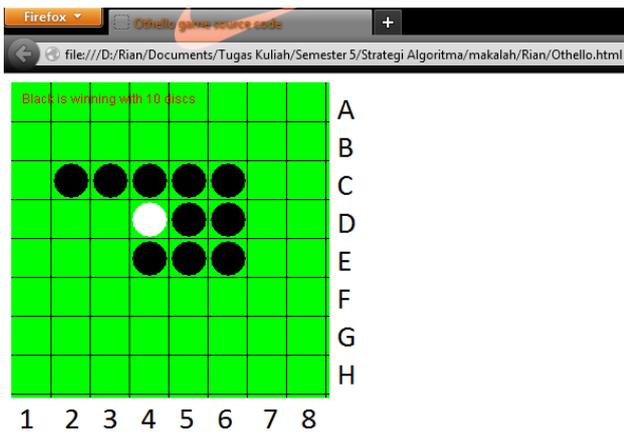
PERNYATAAN

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, 20 Desember 2012

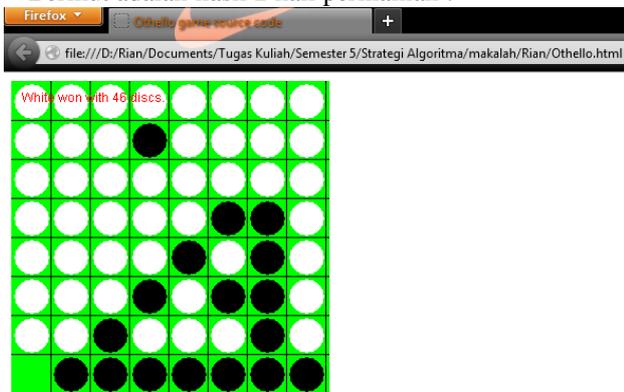


Fabrian Oktavino H
13510053



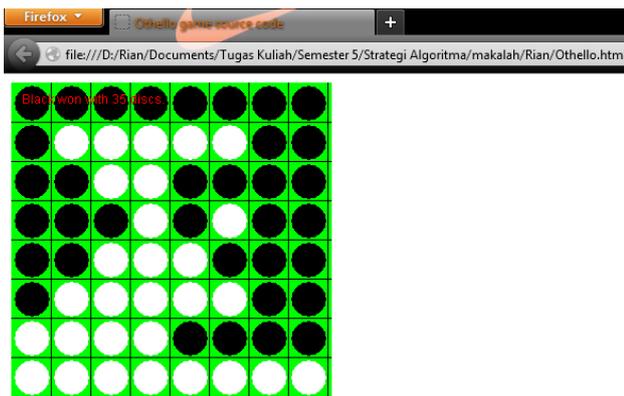
Gambar 9 Keping hitam ditaruh pada posisi (6,E)

Berikut adalah hasil 2 kali permainan :



Gambar 10 Putih menang dengan 46 keping

Kotak terakhir tidak dapat diisi karena keping hitam maupun keping putih tidak dapat mengisi posisi tersebut. Permainan berakhir sampai disini.



Gambar 11 Hitam menang dengan 35 keping