

Penerapan Algoritma Greedy dalam Pembuatan Artificial Intelligence Permainan Reversi

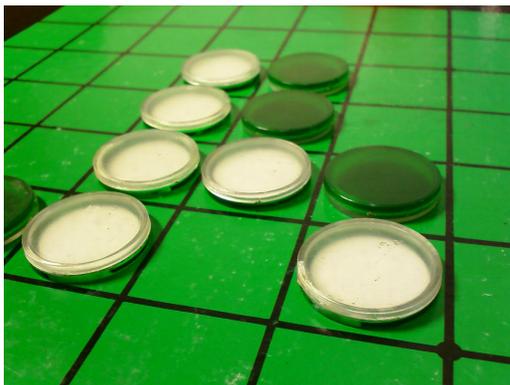
Zacki Zulfikar Fauzi / 13515147
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515147@std.stei.itb.ac.id

Abstract — Dengan semakin berkembangnya teknologi saat ini, banyak sekali permainan yang mengimplementasikan artificial intelligence sebagai lawan bermain. Artificial intelligence tersebut dapat berperan sebagai simulasi/latihan melawan musuh manusia sesungguhnya. Permainan yang dulunya harus dimainkan oleh dua orang pun akan bisa dimainkan oleh satu orang dengan adanya artificial intelligence ini. Dalam makalah ini akan dibahas tentang implementasi algoritma greedy dalam perancangan artificial intelligence untuk permainan reversi.

Keywords—Reversi, Board game, Artificial intelligence, Greedy.

I. PENDAHULUAN

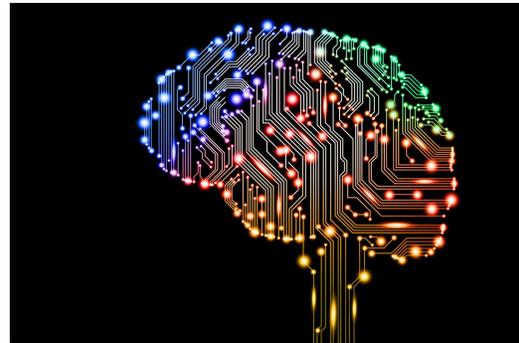
Seiring berjalannya waktu, manusia senantiasa mengembangkan permainan untuk mengisi waktu senggang. Berbagai jenis permainan telah diciptakan oleh manusia, berbagai permainan tersebut berbeda-beda dari cara bermainnya, alat bermainnya hingga jumlah pemainnya. Salah satu jenis permainan yang cukup terkenal adalah permainan papan atau yang biasa disebut *board game*.



Gambar 1.1 Permainan reversi

Salah satu contoh dari *board game* adalah reversi. Sama seperti kebanyakan *board game* lainnya, reversi sudah mulai diadopsi menjadi permainan komputer atau *smartphone*. Reversi memerlukan 2 orang sebagai

pemain, oleh karena itu maka dibutuhkan seseorang sebagai lawan bermain. Hal tersebut mengakibatkan seseorang harus mengajak pemain lain jika ingin memainkan reversi. Walaupun dengan berkembangnya teknologi dan hampir setiap orang saat ini memiliki *smartphone* masing-masing, ada kalanya seseorang ingin memainkan reversi tetapi tidak memiliki lawan bermain. Bagaimana masalah tersebut dapat diselesaikan, tentunya dengan menggunakan kecerdasan buatan atau yang biasa disebut AI (Artificial Intelligence).



Gambar 1.2 Ilustrasi artificial intelligence

Artificial intelligence dapat memberikan keputusan layaknya seperti manusia. Dalam kasus ini, artificial intelligence dapat menggantikan peran lawan bermain reversi. Dengan algoritma yang diberikan pada artificial intelligence tersebut, ia akan bisa menentukan gerakan yang akan dilakukan untuk melawan pemain. Semakin canggih algoritma yang digunakan, maka perlawanan dari AI tersebut akan semakin sengit.

Salah satu algoritma yang dapat digunakan untuk merancang AI ini adalah algoritma greedy. Algoritma greedy cocok untuk digunakan dalam kasus ini karena permasalahan permainan reversi merupakan persoalan dinamis dimana pergerakan lawan selanjutnya hampir tidak bisa diprediksi. Algoritma ini akan memberikan solusi optimal untuk giliran itu sehingga diharapkan akan mencapai hasil akhir yang optimal juga. Dalam makalah ini akan dibahas bagaimana algoritma greedy dapat digunakan untuk merancang artificial intelligence untuk permainan reversi.

II. LANDASAN TEORI

1. Algoritma Greedy

Algoritma greedy membentuk solusi secara langkah demi langkah (step by step). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Algoritma greedy akan mengambil langkah optimum pada tiap step dengan harapan dengan mengambil setiap langkah optimum pada setiap step akan menghasilkan langkah optimum global nantinya. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah berikutnya. Algoritma greedy terkadang tidak berhasil menghasilkan solusi optimum. Karena banyak kasus dimana solusi optimum suatu langkah tidak menghasilkan solusi optimum global.



Gambar 2.1 Ilustrasi greedy

Contoh (Masalah Penukaran uang):

Persoalan: Diberikan uang senilai A . Tukar A dengan koin-koin uang yang ada. Berapa jumlah minimum koin yang diperlukan untuk penukaran tersebut?

Contoh: tersedia koin-koin 1, 5, 10, dan 25

Uang senilai 32 dapat ditukar dengan cara berikut:

$$32 = 1 + 1 + \dots + 1 \quad (32 \text{ koin})$$

$$32 = 5 + 5 + 5 + 5 + 10 + 1 + 1 \quad (7 \text{ koin})$$

$$32 = 10 + 10 + 10 + 1 + 1 \quad (5 \text{ koin})$$

... dan seterusnya

Minimum: $32 = 25 + 5 + 1 + 1$ (hanya 4 koin)

Strategi *greedy* yang digunakan adalah:

Pada setiap langkah, pilihlah koin dengan nilai sebesar mungkin dari himpunan koin yang tersisa dengan syarat (kendala) tidak melebihi nilai uang yang ditukarkan.

Tinjau masalah menukarkan uang 32 dengan koin 1, 5, 10, dan 25:

Langkah 1: pilih 1 buah koin 25 (Total = 25)

Langkah 2: pilih 1 buah koin 5 (Total = 25 + 5 = 30)

Langkah 3: pilih 2 buah koin 1 (Total = 25+5+1+1= 32)

Solusi: Jumlah koin minimum = 4 (solusi optimal!)

Pada setiap langkah di atas kita memperoleh optimum lokal, dan pada akhir algoritma kita memperoleh optimum global (yang pada contoh ini merupakan solusi optimum). Algoritma greedy memiliki 5 buah elemen umum yaitu :

1. Himpunan Kandidat, C

Himpunan kandidat berisi elemen-elemen yang dapat membentuk solusi.

2. Himpunan Solusi, S

Himpunan yang berisi elemen dari himpunan kandidat yang telah dipilih sebagai solusi dari persoalan.

3. Fungsi seleksi

Fungsi yang pada setiap langkah akan memilih elemen dari himpunan kandidat yang paling memungkinkan mencapai solusi optimal.

4. Fungsi kelayakan

Fungsi yang memeriksa apakah elemen kandidat yang telah dipilih jika dimasukkan ke dalam himpunan solusi tidak melanggar constraint yang ada.

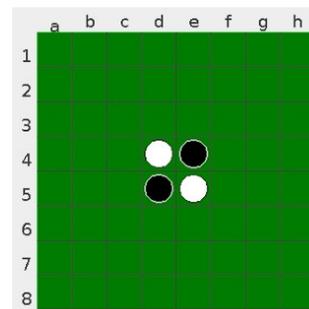
5. Fungsi objektif

Fungsi untuk meminimumkan atau memaksimalkan solusi yang telah dipilih.

Walau algoritma greedy terkadang tidak memberikan solusi optimal, namun untuk banyak kasus algoritma greedy berhasil memberikan solusi optimal dari suatu persoalan. Algoritma greedy banyak dipakai untuk menyelesaikan masalah penukaran uang, minimasi waktu dalam sistem, knapsack, TSP, dll.

2. Reversi

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

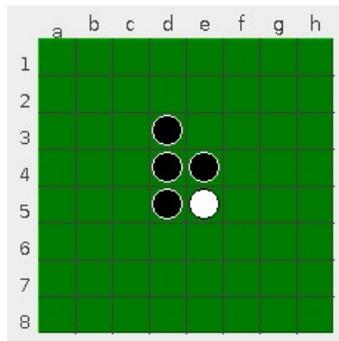


Gambar 2.2 Papan reversi

Tujuan dari permainan ini adalah kedua pemain saling berusaha memiliki jumlah keping terbanyak di akhir permainan untuk jadi pemenang.

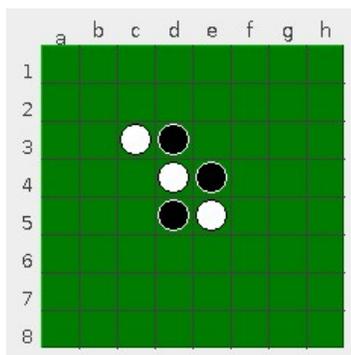
Aturan permainannya adalah sebagai berikut:

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 menggapit keping putih di antara dua keping hitam, yaitu di antara keping hitam yang sudah ada di papan dan keping hitam yang baru diletakkan, boleh menggapit secara horizontal, vertikal maupun diagonal, misalnya di kotak "d3".
3. Keping putih yang terjepit pada no. 2 di atas, dibalik semua menjadi keping hitam.



Gambar 2.3 Papan Reversi 2

4. Berikutnya giliran pemain putih melangkah dengan meletakkan keping putih di kotak kosong di mana dia bisa menggapit keping hitam di antara dua keping putih, yaitu di antara keping putih yang sudah ada di papan dan keping putih yang baru diletakkan, boleh menggapit secara horizontal, vertikal maupun diagonal, misalnya di kotak "c3".
5. Keping hitam yang terjepit pada no. 4 di atas, dibalik semua menjadi keping putih.



Gambar 2.4 Papan Reversi 3

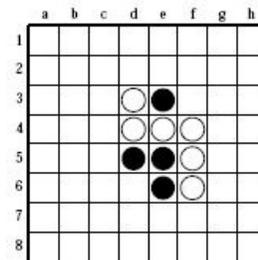
6. Demikian kedua pemain bergantian saling meletakkan kepingnya di kotak kosong di mana dia bisa menggapit keping lawannya, dan membalik keping lawannya menjadi kepingnya.
7. Kedua pemain tidak boleh meletakkan kepingnya di kotak yang sudah terisi, atau di kotak kosong di mana dia tidak menggapit keping lawannya.
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.
10. Pemain dengan jumlah keping terbanyak adalah pemenang.

III. IMPLEMENTASI ALGORITMA GREEDY

1. Artificial intelligence greedy dalam *strategy board game reversi*

Artificial intelligence sudah umum digunakan dalam berbagai jenis permainan, begitu pula dalam *strategy board game*. Dalam *strategy board game*, AI akan mengambil keputusan ketika gilirannya sedang berlangsung, dan mengambil keputusan yang dianggapnya paling baik. Permasalahan pada setiap giliran akan selalu berbeda karena keadaan papan akan selalu berubah pada setiap putaran. Karena permasalahannya selalu berubah, maka ini termasuk permasalahan dinamis dimana gerakan lawan tidak bisa diprediksi. Oleh karena itu, AI dalam permainan reversi akan cocok bila menggunakan algoritma greedy. Hal tersebut dikarenakan pemilihan langkah optimal pada giliran itu (optimal lokal) akan menghasilkan solusi optimal global juga.

Kriteria memenangkan permainan reversi adalah memiliki kepingan paling banyak pada akhir permainan. Oleh karena itu. Oleh karena itu, akan lebih baik jika setiap giliran seorang pemain membalikkan kepingan lawannya sebanyak mungkin dengan harapan pada akhir permainan sang player akan memiliki kepingan lebih banyak dari lawannya. Prinsip tadi sangatlah cocok dengan prinsip greedy yaitu *take what you can* yaitu ambil sebanyak-banyaknya atau semaksimal mungkin.

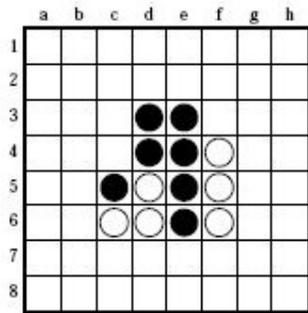


Gambar 3.1 Contoh kasus reversi 1

Gambar diatas merupakan salah satu kasus dalam permainan reversi. Bila AI merupakan pemain yang menggerakkan kepingan hitam, maka ada beberapa posisi tempat menaruh kepingan selanjutnya yaitu : d1, c3, c5, f3, g5, g6, g3, g7. Jika menggunakan prinsip greedy, maka AI akan memilih posisi c3 atau d1 karena kedua posisi tersebut memberikan hasil paling maksimal yaitu membalikkan 2 kepingan lawan sedangkan posisi lain hanya membalikkan 1 kepingan lawan. Proses itu akan diulang hingga permainan usai dengan harapan pada akhir permainan sang AI akan memiliki kepingan yang lebih banyak dari lawannya.

2. Implementasi greedy dalam perancangan AI

Langkah-langkah dalam pemilihan gerakan AI akan diimplementasikan dengan algoritma greedy. Oleh karena itu, setiap kemungkinan dari gerakan akan diberi suatu nilai. Nilai akan diberikan berdasarkan jumlah kepingan lawan yang dapat dibalikkan dengan gerakan tersebut. Gerakan yang dipilih adalah gerakan yang memiliki nilai paling tinggi yaitu yang dapat membalikkan kepingan lawan paling banyak.



Gambar 3.2 Contoh kasus reversi 2

Gambar diatas merupakan salah satu kasus dalam permainan reversi. Bila AI merupakan pemain yang menggerakkan kepingan putih, maka ada beberapa posisi tempat menaruh kepingan selanjutnya yaitu : b5, b4, c2, c3, c4, d2, d7, f3, f7. Posisi b5 akan diberi poin 1, b4 akan diberi poin 1, c2 akan diberi poin 2, c3 akan diberi poin 2, c4 akan diberi poin 3, d2 akan diberi poin 3, d7 akan diberi poin 1, f3 akan diberi poin 1, dan f7 akan diberi poin 1.

Jika menggunakan prinsip greedy, maka AI akan memilih posisi c4 atau d2 karena kedua posisi tersebut memberikan nilai paling besar yaitu 3 poin. Oleh karena itu posisi c4 atau d2 merupakan solusi optimum lokal. Proses itu akan diulang hingga permainan usai dengan harapan pada akhir permainan sang AI akan memiliki kepingan yang lebih banyak dari lawannya.

Algoritma greedy memiliki 5 buah elemen, berikut juga AI reversi ini. 5 buah elemen tersebut adalah:

1. Himpunan Kandidat

Himpunan kandidat berisi kotak-kotak yang dapat dipilih oleh AI. Pada game reversi ini, berarti adalah kotak-kotak yang belum diisi oleh pemain maupun AI itu sendiri (kotak kosong) dan menjepit kepingan lawan.

2. Himpunan Solusi

Himpunan solusi berisi kotak-kotak yang sudah diisi baik oleh pemain maupun oleh AI.

3. Fungsi Seleksi

Memilih langkah yang memiliki nilai paling tinggi / paling menguntungkan.

4. Fungsi Kelayakan

Jumlah langkah yang dipilih oleh AI hanya boleh 1 langkah per giliran, tidak boleh lebih dan tidak boleh kurang.

5. Fungsi Objektif

Jumlah kepingan yang dimiliki oleh AI maksimum.

Pada AI yang dibuat, fungsi seleksi akan mengembalikan posisi yang memiliki nilai paling besar dengan algoritma :

```

procedure Select(input MatriksPoin[8][8], output p : Point)
  i, j, max : int
  i ← 0
  j ← 0
  p.x ← 0
  p.y ← 0
  max ← MatriksPoin[i][j]
  while (i < 8) do
    while (j < 8) do
      if (MatriksPoin[i][j] > max) then
        p.x ← i
        p.y ← j
        max ← MatriksPoin[i][j]
      j ← j + 1
    endwhile
    i ← i + 1
  endwhile

```

Gambar 3.3 Algoritma prosedur Select

Prosedur Select berperan sebagai fungsi seleksi akan mencari nilai terbesar dari matriks integer yang bernama Select. Matriks tersebut akan berisi poin dari setiap posisi pada papan reversi. Poin dari setiap posisi akan ditentukan oleh jumlah kepingan milik musuh yang dapat dibalikkan. Posisi yang tidak bisa dipilih akan memiliki poin 0. Semakin tinggi poinnya maka posisi itu semakin menguntungkan untuk dipilih oleh AI. Misalnya MatriksPoin[0][0] bernilai 3 dan Array[5][3] bernilai 2. Maka AI akan menaruh piecenya di baris pertama kolom pertama (Papan[0][0]) karena jika dilihat dari poin pada posisi tersebut, disimpulkan posisi itu lebih menguntungkan. Jika ada 2 buah posisi yang memiliki skor yang sama, maka program akan memilih posisi yang pertama kali ditemukan. Sehingga program tidak

melanggar constraint (1 langkah per giliran) dan memenuhi fungsi kelayakan. Untuk perhitungan poin setiap posisi digunakan algoritma sebagai berikut :

```

procedure hitungPoin(input Papan : int[8][8], -output
MatriksPoin : int[8][8])
  poin,i,j : int
  poin <- 0
  i <- 0
  j <- 0
  while (i<8) do
    while (j<8) do
      poin <- 0
      poin <- poin + poinHorizontal (Papan,i,j)
      poin <- poin + poinVertikal (Papan,i,j)
      poin <- poin + poinDiagonal (Papan,i,j)
      MatriksPoin[i][j] <- poin
      j <- j+1
    endwhile
    i <- i+1
  endwhile
endwhile

```

Gambar 3.4 Algoritma penghitungan poin

Prosedur hitungPoin adalah prosedur yang memiliki parameter input berupa matriks integer 8x8 bernama Papan yang merupakan representasi papan reversi dalam program. Prosedur hitungPoin memiliki parameter output berupa matriks integer 8x8 bernama MatriksPoin yang berisi poin dari setiap posisi pada papan.

Jika Papan[i][j] memiliki nilai 0 maka posisi tersebut belum diisi oleh kedua pemain. Jika Papan[i][j] memiliki nilai 1 maka posisi tersebut telah diisi dengan kepingan milik pemain pertama. Jika Papan[i][j] memiliki nilai 2 maka posisi tersebut sudah diisi dengan kepingan milik pemain kedua yaitu sang AI.

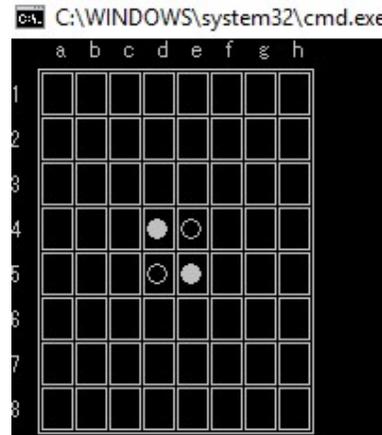
Prosedur hitungPoin akan menghitung poin yang dimiliki oleh setiap posisi dengan cara menghitung jumlah kepingan lawan yang dapat dibalikkan jika kepingan milik AI akan diletakkan di posisi tersebut. Penghitungan dilakukan dengan cara memanggil fungsi poinHorizontal, poinVertikal, dan poinDiagonal. Sesuai namanya, fungsi poinHorizontal akan menghitung jumlah kepingan lawan yang dapat dibalikkan pada baris/horizontal dari posisi tersebut. Fungsi poinVertikal akan menghitung dari kolom/vertikal dari posisi tersebut, dan poinDiagonal akan menghitung poin secara diagonal dari posisi tersebut. Setiap kepingan lawan yang dapat dibalikkan akan bernilai 1 poin, sehingga semakin banyak kepingan yang bisa dibalikkan maka poinnya akan semakin besar pula.

Pada program utama, setiap gilirannya akan diambil posisi yang memiliki poin terbesar. Jika ada lebih dari 1 posisi yang memiliki poin terbesar, maka akan dipilih posisi yang lebih awal ditemukan.

Proses pengamilan langkah akan terus dilakukan hingga tidak ada elemen dari matriks Papan yang bernilai 0, yang artinya seluruh posisi dari papan reversi sudah terisi, atau ketika kedua pemain tidak bisa bergerak lagi.

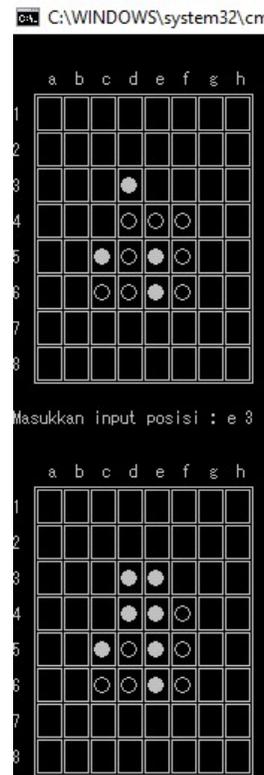
IV. HASIL IMPLEMENTASI PROGRAM

Untuk eksperimen dari AI yang dibuat, saya membuat simulasi dari permainan reversi dan AI dalam bahasa C++.



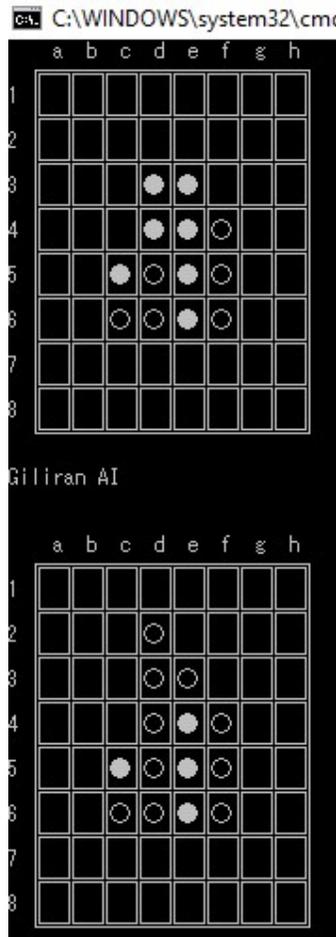
Gambar 4.1 Tampilan dari simulasi reversi

Dalam simulasi ini, kepingan milik pemain pertama dilambangkan dengan lingkaran putih dan kepingan milik AI dilambangkan dengan lingkaran hitam. Program akan menerima input berupa 1 karakter dan 1 integer yang merupakan posisi yang ingin dipilih oleh pemain. Program akan menampilkan keadaan dari papan permainan pada setiap giliran.



Gambar 4.2 Eksekusi simulasi program 1

Pada gambar 4.2 pemain pertama memasukkan input e 3 yang akan membalikkan 2 kepingan milik AI dan setelah itu program akan menampilkan hasil gerakannya.



Gambar 4.3 Eksekusi simulasi program 2

Pada gambar 4.3 AI melakukan proses pemilihan langkah dengan menghitung poin dari setiap posisi. Ada 2 posisi yang memiliki poin tertinggi yaitu d2 dan c3 tapi d2 dipilih karena lebih dahulu ditemukan.

Dari hasil eksekusi tersebut, dapat dibuktikan bahwa algoritma yang dibuat telah berhasil memenuhi prinsip greedy yaitu memilih posisi yang paling banyak membalikkan kepingan milik lawan. Namun menurut saya, hasil tersebut belum begitu maksimal untuk menciptakan AI yang kompetitif karena dengan algoritma greedy yang digunakan, AI tidak memprediksi gerakan lawan selanjutnya yang menyebabkan terkadang dapat menguntungkan lawan bermain jika dia berhasil memanfaatkan keadaan tersebut.

Oleh karena itu, AI harus dikembangkan lebih lanjut dengan pertimbangan memprediksi kemungkinan gerakan lawan selanjutnya, dan memperhatikan batasan lain seperti

tidak membiarkan lawan mendapat posisi pinggiran/pojok. Dengan begitu AI akan bekerja lebih baik dan diharapkan akan menjadi lawan main yang lebih sulit sehingga akan menciptakan permainan yang cukup sengit antara pemain dengan AI.

V. KESIMPULAN

Algoritma greedy dapat digunakan sebagai suatu pendekatan penyelesaian masalah dalam perancangan AI dalam *strategy board game* seperti reversi karena prinsip greedy yang cocok dengan permainan tersebut yaitu *take what you can* alias ambil seanyak-banyaknya. Tetapi, strategi dari AI masih cukup sederhana. Oleh karena itu, AI ini harus dikembangkan lebih lanjut untuk menutupi kekurangan yang dimiliki.

VI. UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan YME karena hanya oleh karena rahmat-Nya penulis dapat menyelesaikan tulisan ini. Penulis juga berterima kasih kepada dosen yang memberikan tugas ini Dr. Ir. Rinaldi Munir atas bimbingan dan jasa beliau yang selama ini telah mengajar dan memberikan ilmu pada mata kuliah strategi algoritma, sehingga penulis mampu membuat tulisan ini. Tak lupa juga penulis berterima kasih atas rekan-rekan yang senantiasa memberikan dorongan dan semangat bagi penulis.

REFERENSI

- [1] Munir, Rinaldi, Strategi Algoritma. Bandung : Penerbit Informatika, Palasari
- [2] Munir, Rinaldi, Slide Perkuliahan IF2211 Algoritma Greedy (revisi. 2017) , 17 Mei 2017. 18:35 WIB.
- [3] <http://genethello.blogspot.co.id/2011/06/apa-itu-othello.html> diakses 17/05/2017, 21:50 WIB

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, 18 Mei 2017

Zacki Zulfikar Fauzi / 13515147