

Algoritma *Greedy* dalam Strategi Permainan Othello

Juan Felix Parsaoran Tarigan, 13516143

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

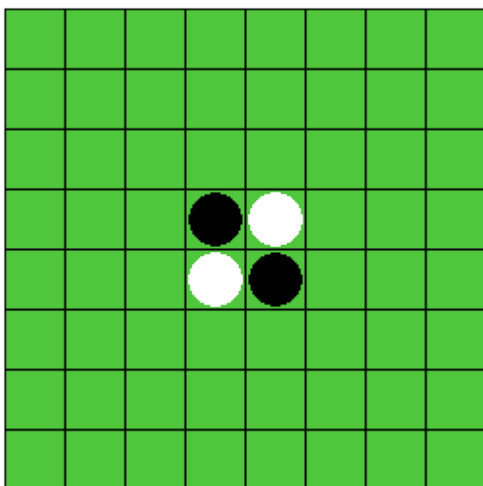
13516143@std.stei.itb.ac.id

Abstract—Permainan othello merupakan permainan papan yang sangat terkenal di dunia. Permainan ini memakai papan 8x8 dan dimainkan oleh 2 masing-masing memegang keping putih dan keping hitam. Permainan ini dapat dimainkan oleh semua kalangan. Algoritma *greedy* merupakan algoritma yang sering dipakai dalam masalah optimasi. Algoritma *greedy* meninjau setiap langkah persoalan dan mencari solusi optimum setiap langkah sesuai fungsi objektifnya yang diharapkan bisa membawa kepada solusi optimum global. Pada permainan othello, tujuan akhir permainan adalah untuk mencapai keping sebanyak-banyaknya sampai papan permainan sudah penuh atau sudah tidak ada langkah yang mungkin lagi antara kedua pemain. Hal tersebut tentu berhubungan dengan masalah optimasi. Oleh karena itu, permainan othello dapat dimainkan dengan menggunakan algoritma *greedy*.

Keywords—Othello; Greedy; Keping; Optimasi;

I. PENDAHULUAN

Permainan othello adalah permainan papan yang sangat terkenal di dunia. Permainan ini menggunakan papan 8x8 dan dimainkan oleh 2 pemain. Tujuan permainan ini adalah untuk mendapatkan keping sebanyak-banyaknya sampai papan penuh. Permainan dimulai dengan 2 keping putih dan 2 keping hitam di tengah seperti pada gambar berikut:



Gambar 1.1 Keadaan awal permainan othello

(Sumber: www2.cs.duke.edu)

Othello muncul pertama kali di Jepang pada tahun 1945, setelah bom atom dijatuhkan di Hiroshima dan Nagasaki. Hasegawa Goro menciptakan permainan ini saat Jepang sedang dalam keadaan mencekam. Lalu pada tahun 1971, Jepang mempatenkan Othello. Nama Othello sendiri diambil dari salah satu karya Shakespeare dengan judul yang sama.

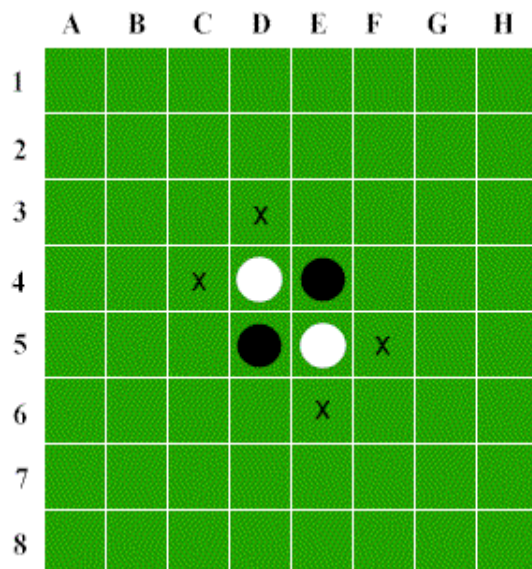
Sesuai tujuan permainan othello, maka masalah yang dihadapi dalam permainan ini adalah masalah optimasi. Banyak algoritma yang dapat menyelesaikan masalah optimasi seperti *greedy*, *dynamic programming*, dan masih banyak lagi. Dalam permainan ini, salah satu algoritma yang paling sangkil dan mangkus adalah *greedy* karena sesuai dengan tujuan permainan ini yaitu mencapai keping semaksimal mungkin di akhir permainan, baik papan permainan sudah penuh atau sudah tidak ada langkah yang memungkinkan diantara kedua pemain.

II. DASAR TEORI

A. Peraturan dan Cara Bermain Othello

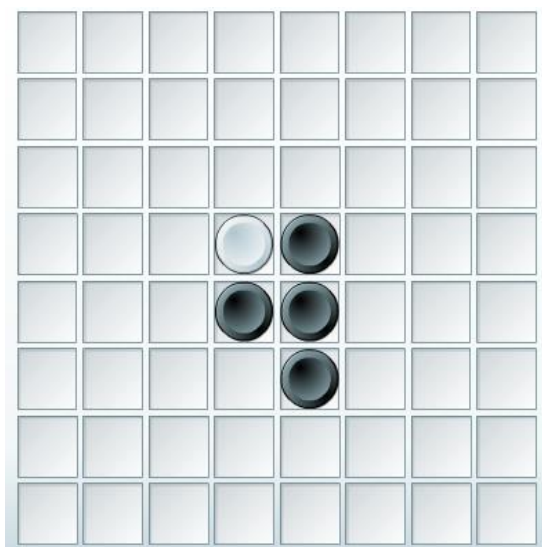
Permainan othello memiliki beberapa aturan dan cara memainkannya. Berikut adalah peraturan dalam permainan othello:

1. Permainan diawali dengan 2 keping berwarna hitam dan 2 keping berwarna putih. Posisi keping berada di pertengahan papan dengan posisi keping saling bersilangan.
2. Pemain dengan keping berwarna hitam mulai terlebih dahulu. Pemain meletakkan kepingnya sehingga keping yang dimiliki lawan terpit. Lalu keping lawan tersebut berubah menjadi keping pemain. Kondisi terpit yang ada adalah dalam kondisi horizontal, vertikal, dan diagonal.



Gambar 2.1 Langkah yang dapat dilakukan oleh pemain berkeping hitam

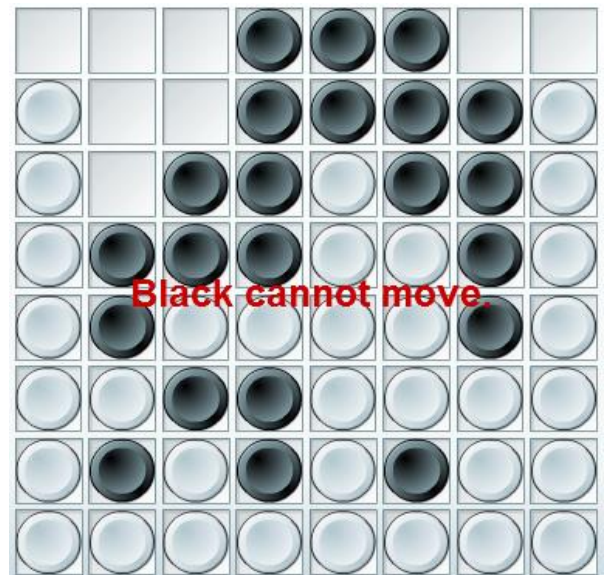
(Sumber: britishothello.org.uk)



Gambar 2.2 Keadaan saat hitam meletakkan keping pada koordinat (5,6)

(Sumber : othelloonline.org dengan olahan sendiri)

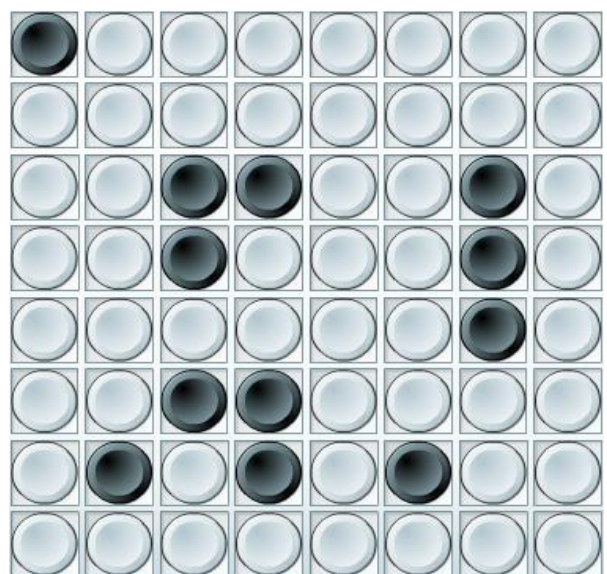
3. Apabila pemain tidak memiliki tempat lagi untuk mengapit keping yang dimiliki lawan, maka giliran pemain tersebut di *pass*.



Gambar 2.3 Keadaan saat keping hitam tidak memiliki langkah, maka gilirannya di *pass*, pemain putih melangkah kembali

(Sumber : othelloonline.org dengan olahan sendiri)

4. Sekali keping digerakkan, tidak boleh diangkat kembali.
5. Permainan selesai saat papan sudah penuh atau kedua pemain sama-sama tidak memiliki langkah lagi.
6. Pemain yang memiliki keping terbanyak di akhir permainan adalah pemenangnya.



Gambar 2.4 Keadaan akhir permainan, putih menang

(Sumber : othelloonline.org dengan olahan sendiri)

B. Algoritma Greedy

Algoritma *greedy* merupakan algoritma yang paling populer digunakan untuk memecahkan persoalan optimasi. Persoalan optimasi adalah persoalan untuk mencari solusi optimum. Ada 2 macam persoalan optimasi yaitu maksimasi dan minimasi. Algoritma *greedy* memegang prinsip optimasi pada setiap langkah, dalam arti mencari nilai optimum lokal pada setiap langkah dengan harapan akan mencapai nilai optimum global. Misalnya ada permasalahan mencari jarak terpendek dari Institut Teknologi Bandung menuju Bandung Indah Plaza. Pada algoritma *greedy* akan ditentukan skema umumnya yaitu himpunan kandidat, himpunan solusi, fungsi seleksi, fungsi kelayakan, dan fungsi objektif.

C. Himpunan Kandidat

Himpunan kandidat dalam algoritma *greedy* merupakan kumpulan elemen yang akan dipilih untuk membentuk himpunan solusi. Elemen-elemen tersebut merupakan elemen yang dapat dipilih untuk mencapai solusi di akhir. Pada persoalan jarak terpendek dari ITB ke BIP, himpunan kandidatnya adalah Taman Sari, Ir.H. Djuanda, Cihampelas, Tubagus Ismail dan Ciumbeluit.

D. Himpunan Solusi

Himpunan solusi dalam algoritma *greedy* merupakan kumpulan elemen yang dipilih dari langkah-langkah dalam mencari solusi dari himpunan kandidat, dalam arti himpunan solusi merupakan himpunan bagian dari himpunan kandidat.

E. Fungsi Seleksi

Fungsi seleksi dalam algoritma *greedy* merupakan fungsi yang akan menyeleksi himpunan kandidat dan akan dimasukkan ke himpunan solusi. Fungsi ini akan memilih optimum lokal dari setiap langkah dan fungsi ini independen untuk setiap langkah. Untuk persoalan jarak terpendek dari ITB ke BIP, maka fungsi seleksinya adalah jarak terpendek dari ITB ke himpunan kandidat, maka elemen itulah yang dimasukkan ke himpunan solusi.

F. Fungsi Kelayakan

Fungsi kelayakan dalam algoritma *greedy* merupakan fungsi yang akan memeriksa apakah langkah yang diambil masih memenuhi batasan persoalan yang ada. Setelah dipilih elemen himpunan kandidat berdasarkan fungsi seleksi, harus divalidasi dulu menggunakan fungsi kelayakan sebelum masuk ke himpunan solusi. Untuk persoalan jarak terpendek dari ITB ke BIP, misalnya fungsi kelayakannya adalah jaraknya harus kurang dari 1km.

G. Fungsi Objektif

Fungsi objektif dalam algoritma *greedy* merupakan fungsi yang menentukan optimasi dalam persoalan tersebut misalnya dalam persoalan jarak terpendek dari ITB ke BIP, maka fungsi objektifnya adalah jarak terpendek dari suatu titik ke titik selanjutnya.

III. ALGORITMA GREEDY DALAM PERMAINAN OTHELLO

Dalam penggunaan algoritma *greedy* untuk permainan othello, maka harus ditentukan terlebih dahulu skema umum algoritma *greedy* dalam permainan othello

A. Himpunan Kandidat

Himpunan kandidat dalam permainan othello adalah semua kotak yang ada di papan.

$$C = \{k1, k2, k3, k4, \dots\}$$

$$C = \{(1,1), (2,1), (3,1), \dots, (6,8), (7,8), (8,8)\}$$

B. Himpunan Solusi

Himpunan solusi pada permainan othello adalah koordinat yang telah memenuhi fungsi objektif, fungsi seleksi dan fungsi kelayakan. Sebagai contoh dalam keadaan awal permainan, jika fungsi objektifnya adalah meminimasi keping, maka himpunan solusinya adalah:

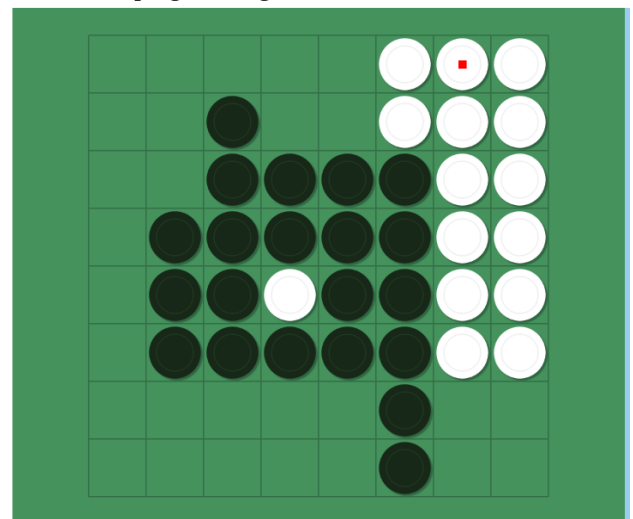
$$S = \{(4,2), (3,4), (5,6), (6,5)\}$$

C. Fungsi Seleksi

Fungsi seleksi pada permainan othello adalah fungsi yang akan menyeleksi koordinat mana yang memenuhi fungsi objektif. Misalkan fungsi objektifnya adalah memaksimalkan keping yang didapat, maka fungsi seleksi akan menyeleksi koordinat yang mana pada himpunan kandidat yang membuat jumlah keping saat ini bertambah lebih besar.

D. Fungsi Objektif

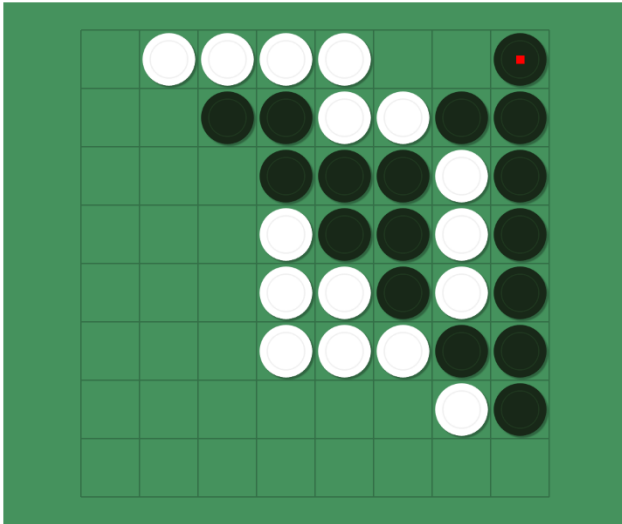
Meskipun tujuan dari permainan othello adalah mendapat keping sebanyak-banyaknya di akhir permainan, bukan berarti fungsi objektif dalam permainan ini selalu memaksimalkan. Pada awal permainan, fungsi objektif yang lebih tepat adalah meminimasi keping. Lihat gambar berikut ini:



Gambar 3.1 Permainan othello
(Sumber : playok.com dengan olahan sendiri)

Terlihat dari gambar tersebut keping hitam lebih banyak dari keping putih, tetapi ruang gerak hitam lebih sedikit dari

putih. Hal ini membuat putih dengan mudahnya menggiring gerakan hitam dan pada akhirnya putih akan menang. Maka dari itu, fungsi objektif yang lebih tepat di awal adalah meminimasi keping. Lihat gambar dibawah ini:



Gambar 3.2 Permainan othello
(Sumber : playok.com dengan olahan sendiri)

Terlihat jumlah hitam lebih sedikit, dan hitam sudah mendapatkan 1 tempat di koordinat sudut. Setelah mendapatkan sudut, fungsi objektif berubah menjadi memaksimalkan keping. Hal ini dikarenakan setelah mendapatkan ujung, keping akan lebih leluasa dalam menggapit keping lawan karena keping tersebut tidak dapat diapit lagi.

```
while moveAvaliable()

if searchUjung()

    {Fungsi Objektif Maksimasi}

else

    {Fungsi Objektif Minimasi}
```

Gambar 3.3 Pseudocode fungsi objektif othello

E. Fungsi Kelayakan

Fungsi kelayakan pada permainan othello adalah apakah penempatan keping tersebut menggapit keping lawan minimal 1 keping atau tidak. Fungsi kelayakan ini sejalan dengan fungsi seleksi pada permainan othello.

IV. ANALISIS

Algoritma *greedy* akan mengiterasi setiap koordinat kotak yang ada di papan. Lalu setiap kotak akan dicatat menambah

berapa keping saat keping diletakkan pada koordinat tersebut. Saat belum terdapat keping yang berada di ujung, maka fungsi objektif yang meminimasi yang dipakai, artinya koordinat yang menambah keping lebih sedikit yang akan dipakai, tetapi saat sudah mendapat sudut, maka fungsi objektif memaksimalkan yang akan dipakai, artinya koordinat yang menambah keping lebih banyak yang akan dipakai. Iterasi dilakukan terus menerus sampai sudah tidak ada lagi koordinat yang dapat menambah jumlah keping.

```
while moveAvaliable()

if searchUjung()

    max <- -99

    For kotak in papan

        if max <- kepingDiapit(kotak)

            max<-kepingDiapit(kotak)

    letakKeping(max)

else

    min <- 99

    For kotak in papan

        if min > kepingDiapit(kotak)

            min<-kepingDiapit(kotak)

    letakKeping(min)
```

Gambar 4.1 Pseudocode main program

Algoritma ini dapat dioptimasi lagi dengan menambah fungsi seleksinya, yaitu keping tidak diletakkan di koordinat di sekitar diagonal ujung seperti (7,7), (2,7), (7,2), dan (2,2). Hal ini menghindari lawan mendapatkan ujung dengan mudah, dimana dijelaskan sebelumnya, bahwa posisi ujung merupakan kunci kemenangan dalam permainan othello.

```

while moveAvaliable()

if searchUjung()

    max <- -99

    For kotak in papan

        if max <- kepingDiapit(kotak)
and not cekSekitarUjung(kotak) then

            max<-kepingDiapit(kotak)

    letakKeping(max)

else

    min <- 99

    For kotak in papan

        if min > kepingDiapit(kotak)
and not cekSekitarUjung(kotak) then

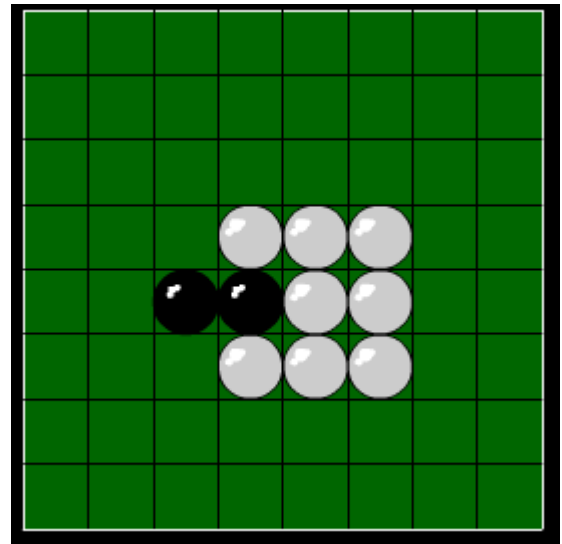
            min<-kepingDiapit(kotak)

    letakKeping(min)

```

Gambar 4.2 Pseudocode main program dengan modifikasi

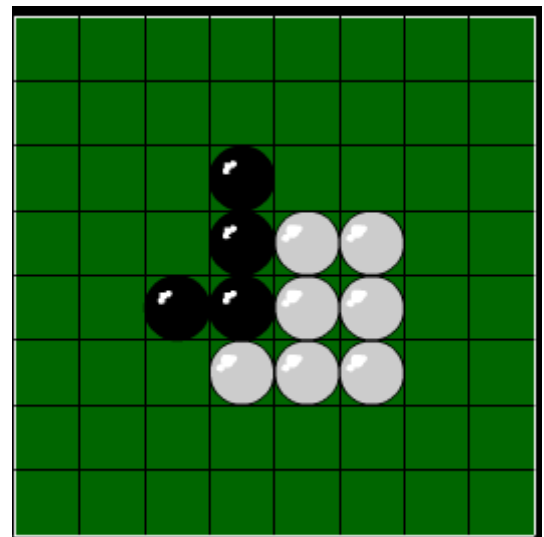
Berikut contoh penerapan algoritma *greedy* dalam permainan othello:



Gambar 4.3 Keadaan permainan awal, sekarang adalah giliran hitam

(Sumber : othelloonline.org dengan olahan sendiri)

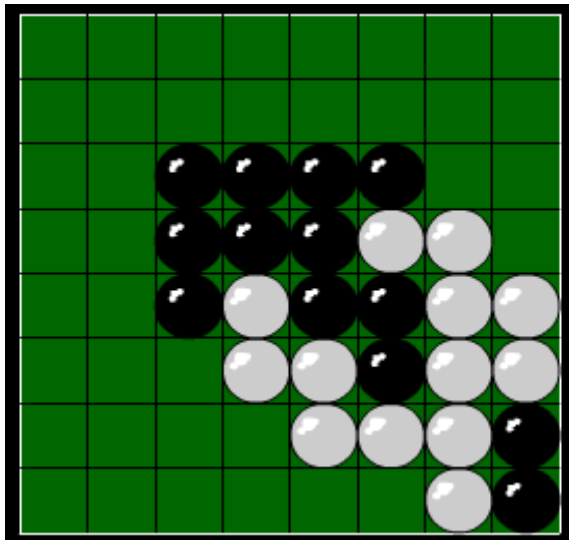
Terlihat pada gambar, belum ada keping hitam yang ada di ujung, maka algoritma memakai fungsi objektif minimasi. Dilakukan iterasi dari kotak (1,1) sampai (8,8). Koordinat yang meminimasi keping adalah koordinat (4,3), (5,3), (6,3), (4,7), (5,7), (6,7). Dipilih koordinat yang pertama kali didapat yaitu (4,3).



Gambar 4.4 Keadaan setelah proses algoritma *greedy*

(Sumber : othelloonline.org dengan olahan sendiri)

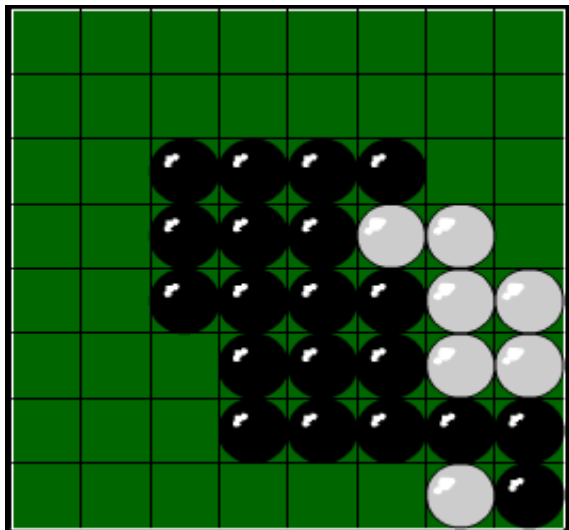
Contoh lain:



Gambar 4.5 Keadaan awal, giliran hitam

(Sumber : othelloonline.org dengan olahan sendiri)

Terdapat keping hitam di ujung papan, maka *greedy* memakai fungsi objektif memaksimalkan koordinat yang memaksimalkan jumlah keping hitam adalah (4,7) dengan menambah 7 buah keping.



Gambar 4.6 Keadaan setelah proses algoritma *greedy*

(Sumber : othelloonline.org dengan olahan sendiri)

V. KESIMPULAN DAN SARAN

Algoritma *greedy* merupakan algoritma yang populer dalam penanganan masalah optimasi. Algoritma *greedy* meninjau langkah per langkah untuk diambil optimum lokalnya yang diharapkan membawa menuju optimum global. Algoritma *greedy* dapat digunakan juga dalam permainan othello dimana permainan ini mempunyai tujuan untuk mendapatkan keping sebanyak-banyaknya di akhir permainan. Kedua optimasi dapat dilakukan dalam *greedy* untuk permainan ini. Optimasi untuk meminimasi dalam tahap awal permainan, dan optimasi maksimasi pada saat pemain sudah mendapat koordinat ujung yang merupakan kunci untuk memenangkan permainan.

Algoritma ini dapat dikembangkan lebih lanjut dengan memperhatikan pemosisian keping, karena dengan posisi yang strategis ke pinggir, maka ruang gerak lawan akan terbatas dan ruang gerak pemain akan semakin banyak.

VI. REFERENSI

- [1] Munir, Rinaldi. Strategi Algoritma. Bandung: Informatika Bandung 2009, Bab.3
- [2] <http://www.ketemu.ga/2017/09/othello-game-reversi.html>, diakses tanggal 13 Mei 2018

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

Juan Felix Parsaoran Tarigan
13516143