

# Algoritma Greedy dalam Permainan *Tic Tac Toe*

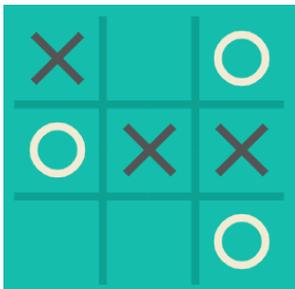
Rizky Ramadhana P. K. - 13520151  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13520151@std.stei.itb.ac.id

**Abstract**—Permainan *tic tac toe* merupakan permainan dua orang yang dimainkan di atas papan berukuran 3x3. Pemain bergiliran meletakkan tanda X dan O pada papan yang masih kosong. Pemain dikatakan menang apabila berhasil membuat garis. Pada makalah ini akan dirancang sebuah algoritma *greedy* untuk memainkan permainan *tic tac toe*. Algoritma *greedy* yang dibuat akan memilih langkah yang paling mendekati pemain ke kemenangan dan sekaligus menjauhkan lawan dari kemenangan. Ukuran seberapa dekat seseorang dengan kemenangan diukur dari jumlah langkah minimum yang harus dilakukan seseorang untuk mendapat kemenangan.

**Keywords**—*tic tac toe*; *greedy*; *algoritma*

## I. PENDAHULUAN

### A. Permainan *Tic Tac Toe*



Gambar 1 Ilustrasi permainan *tic tac toe*

Permainan *tic tac toe* merupakan permainan yang dimainkan oleh dua orang pada sebuah papan berukuran 3 x 3. Setiap pemain memiliki simbolnya sendiri-sendiri, biasanya X dan O. Umumnya, pemain dengan simbol X akan memulai permainan terlebih dahulu. Pemain bergiliran memberi tanda miliknya di papan permainan. Pemain dinyatakan menang apabila berhasil membuat satu garis horizontal, garis vertikal, atau garis diagonal terlebih dahulu menggunakan symbol yang dimilikinya. Pada permainan ini tidak selalu didapatkan pemenang pada akhir permainan, Bisa saja permainan berakhir seri, artinya papan sudah penuh namun belum ada pemain yang berhasil menyusun garis horizontal, vertikal, maupun diagonal menggunakan simbol miliknya.

Pada permainan ini terdapat 138 kemungkinan akhir, yang mana 91 di antaranya dimenangkan X, 44 di antaranya dimenangkan O, dan 3 sisanya tidak dimenangkan siapapun [1]. Permainan ini juga memiliki berbagai variasi, misalnya

variasi ukuran di mana ukuran papan yang dimainkan bukan 3x3 melainkan 4x4 atau bahkan 5x5. Terdapat pula permainan *tic tac toe* tiga dimensi di mana permainan ini dimainkan di sebuah kubus berukuran 3x3x3. Namun pada makalah ini, pembahasan difokuskan pada permainan *tic tac toe* 3x3.

### B. Algoritma *Greedy*

Salah satu strategi yang bisa digunakan untuk menyelesaikan sebuah permasalahan adalah algoritma *greedy*. Algoritma *greedy* akan memilih langkah terbaik dalam keadaan saat ini tanpa melihat ke depan ataupun ke belakang. Algoritma *greedy* hanya akan mempertimbangkan keadaan saat ini dan keadaan bila mengambil satu langkah tertentu. Dalam berbagai kemungkinan langkah akan dibandingkan langkah mana yang paling menguntungkan atau bisa disebut juga paling mendekati ke kemenangan. Algoritma *greedy* sering juga disebut sebagai algoritma yang mengambil keputusan yang menghasilkan keuntungan optimum lokal, tentunya tidak bisa dipastikan apakah keputusan tersebut juga akan mengarah ke optimum global. Sehingga tidak semua algoritma *greedy* bisa membersihkan solusi sebenarnya terhadap suatu permasalahan.

Algoritma *greedy* memiliki enam komponen [2], yang pertama adalah himpunan kandidat. Himpunan kandidat merupakan himpunan yang anggotanya berupa langkah-langkah yang bisa diambil pada setiap keadaan. Yang kedua adalah himpunan solusi, yaitu himpunan yang berisikan langkah-langkah yang sudah dipilih. Komponen yang ketiga adalah fungsi solusi, yaitu fungsi yang mengecek apakah himpunan kandidat yang sudah dipilih merupakan solusi persoalan atau tidak. Yang keempat adalah fungsi seleksi, yaitu fungsi yang memilih langkah apa yang akan diambil selanjutnya. Untuk persoalan yang sama, bisa saja terdapat fungsi seleksi yang berbeda-beda. Yang kelima adalah fungsi kelayakan, yaitu fungsi yang bertugas untuk mengecek apakah kandidat yang baru saja dipilih layak untuk dimasukkan kedalam himpunan solusi atau tidak. Yang keenam adalah fungsi obyektif, yaitu fungsi yang diminimumkan atau dimaksimumkan. Secara garis besar, algoritma *greedy* akan mencari langkah yang paling mengoptimalkan fungsi obyektif menggunakan fungsi seleksi dengan domain himpunan kandidat. Calon solusi tersebut akan diperiksa menggunakan fungsi kelayakan dan bila layak akan dimasukkan ke dalam himpunan solusi. Di setiap langkah akan

diperiksa apakah himpunan solusi yang dimiliki saat ini sudah merupakan solusi menggunakan fungsi solusi.

### C. Karya Tulis Sejenis

Pada makalah sejenis, juga telah dirumuskan strategi dalam permainan *tic tac toe* menggunakan algoritma *branch and bound* [3]. Pada makalah tersebut permainan *tic tac toe* dianalogikan dengan pencarian pada sebuah pohon di mana di dalamnya terdapat *cost function* untuk menentukan cabang mana yang akan dikunjungi terlebih dahulu. Pada makalah lain juga telah dirumuskan strategi dalam permainan *tic tac toe* menggunakan algoritma *minimax* [4]. Pada dasarnya, setiap algoritma yang ditulis pada makalah di atas akan memilih mana langkah yang terbaik untuk giliran saat ini. Posisi makalah ini juga sama dengan makalah-makalah yang disebutkan sebelumnya, yaitu ingin mencari langkah yang terbaik untuk giliran saat ini. Hanya saja kedua makalah di atas akan menelusuri kemungkinan-kemungkinan di langkah-langkah berikutnya untuk menentukan langkah saat ini. Sedangkan makalah ini akan menggunakan algoritma *greedy* di mana penentuan langkah saat ini hanya dipengaruhi oleh keadaan saat ini tanpa melihat ke depan ataupun ke belakang.

## II. ISI

### A. Pemetaan Algoritma Greedy ke Permainan Tic Tac Toe

#### 1) Himpunan kandidat

Himpunan kandidat pada permainan ini adalah himpunan yang berisi posisi kotak-kotak yang masih kosong. Kotak-kotak tersebut lah yang bisa menjadi solusi (langkah terbaik) karena pada permainan *tic tac toe* tidak diperbolehkan mengisi kotak yang sudah diisi sebelumnya.

#### 2) Himpunan solusi

Himpunan solusi pada permainan ini adalah himpunan posisi-posisi yang telah dipilih melalui fungsi seleksi. Posisi-posisi tersebut diharapkan bila diisi akan memberi kemenangan.

#### 3) Fungsi solusi

Fungsi solusi pada permainan ini adalah fungsi yang mengecek apakah sudah ada pemenang pada keadaan saat ini atau papan sudah penuh. Bila belum ada pemenang dan papan belum penuh, maka fungsi solusi akan mengembalikan nilai *false*.

#### 4) Fungsi seleksi

Fungsi seleksi adalah fungsi yang dieksekusi untuk menentukan calon solusi. Pada permainan ini, tujuan yang ingin dicapai adalah memenangkan permainan. Sehingga fungsi seleksi harus dibuat sedemikian hingga mendekatkan kita ke kemenangan dan menjauhkan lawan dari kemenangan. Pada makalah ini akan dibahas dua alternatif fungsi seleksi yang memungkinkan. Implementasi lebih detail mengenai alternatif tersebut akan dijelaskan di bagian selanjutnya.

#### 5) Fungsi kelayakan

Pada permainan ini, khususnya pada makalah ini, fungsi kelayakan akan selalu mengembalikan nilai *true*. Karena sudah diasumsikan bahwa posisi yang masuk ke

dalam himpunan kandidat adalah posisi yang kosong. Sehingga fungsi seleksi tidak mungkin menghasilkan posisi yang tidak valid.

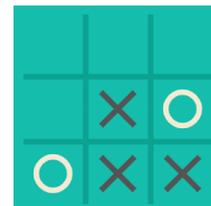
#### 6) Fungsi obyektif

Obyektif yang ingin dicapai adalah memenangkan permainan.

### B. Alternatif Fungsi Seleksi

#### 1) Alternatif pertama

Alternatif pertama adalah dengan memberikan poin pada setiap langkah yang memungkinkan. Langkah yang memberikan poin maksimum lah yang akan dipilih. Sebagai contoh, memblokir lawan akan diberikan poin 4. Contohnya pada gambar di bawah, menempatkan O pada baris pertama kolom kedua akan diberikan poin empat.



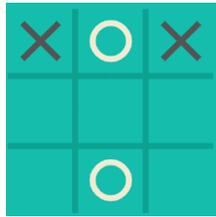
Gambar 2 Ilustrasi gerakan memblokir lawan

Contoh pemberian poin lain adalah langkah yang membuat kesempatan baru untuk memenangkan permainan akan diberikan poin 3. Seperti contoh pada gambar di bawah, menempatkan X di baris pertama kolom ketiga akan mendapatkan poin ketiga karena menciptakan kesempatan baru untuk menang.



Gambar 3 Ilustrasi gerakan menciptakan kesempatan baru untuk menang

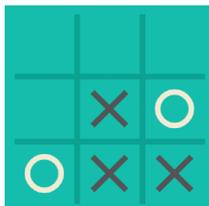
Sebuah langkah mungkin saja dikategorikan lebih dari satu jenis. Contohnya dapat diamati pada gambar berikut, menempatkan X pada baris kedua kolom kedua selain memblokir lawan juga menciptakan dua kesempatan baru untuk menang (dengan membuat diagonal ke kiri bawah atau ke kanan bawah). Sehingga poin yang didapatkan bila menempatkan X pada posisi tersebut adalah sama dengan poin memblokir lawan ditambah dua kali poin menciptakan kesempatan baru untuk menang. Posisi yang mendapatkan poin paling besar lah yang akan dipilih sebagai langkah optimum.



Gambar 4 Ilustrasi gerakan dengan banyak kategori

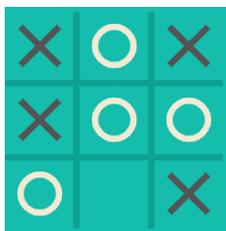
## 2) Alternatif kedua

Pada alternatif ini perlu dikenal istilah “langkah minimum untuk menang”, yaitu jumlah langkah paling kecil yang dibutuhkan seseorang untuk memenangkan permainan tersebut. Contohnya pada gambar di bawah, langkah minimum untuk menang milik X adalah 1 sedangkan langkah minimum untuk menang milik O adalah 2 (perlu mengisi kolom pertama baris pertama dan kedua).



Gambar 5 Ilustrasi langkah minimum untuk menang

Pada awal permainan, langkah minimum untuk menang selalu bernilai 3 karena pemain harus membuat paling tidak tiga tanda berjejer. Terdapat kasus khusus juga dalam menghitung langkah minimum untuk menang. Bila salah satu pemain menang, maka langkah minimum untuk menang pada saat itu adalah -4. Hal ini dilakukan agar algoritma memprioritaskan langkah yang menutup permainan dengan kemenangan. Contoh lain dapat diamati pada gambar di bawah. Bila pada saat ini pemain tidak mungkin menang, sebanyak apa pun langkah yang ia lakukan, maka langkah minimum untuk menangnya diberi nilai 4. Pada gambar di bawah, langkah minimum untuk menang milik X adalah 4.



Gambar 6 Ilustrasi kasus X tidak mungkin menang

Diinginkan fungsi seleksi akan menghasilkan langkah yang paling mendekati pemain ke kemenangan sehingga langkah yang dipilih harus meminimumkan langkah minimum untuk menang milik pemain. Diinginkan pula fungsi seleksi akan menghasilkan langkah yang paling menjauhkan lawan dari kemenangan sehingga langkah yang dipilih harus memaksimalkan langkah

minimum untuk menang milik lawan. Kedua hal tersebut harus dipenuhi secara bersamaan, sehingga dibuatlah parameter  $-p+1$  yang harus dimaksimalkan. Dengan  $p$  adalah langkah minimum untuk menang milik pemain dan  $l$  adalah langkah minimum untuk menang milik lawan.

## C. Solusi yang Dipilih

Dari kedua alternatif yang telah dibahas sebelumnya, dipilih alternatif kedua dengan alasan sebagai berikut. Pemberian poin pada alternatif pertama mengandung unsur subjektif yang lebih besar, bisa saja seseorang berpendapat bahwa gerakan yang satu lebih penting daripada gerakan lainnya. Rasio poin antar berbagai jenis gerakan juga menjadi salah satu aspek yang subjektif. Hal ini juga berpotensi menjadikan lingkup pembahasan yang terlalu melebar karena *tuning* rasio poin antara satu jenis gerakan dan gerakan lainnya merupakan hal lain yang mungkin untuk didiskusikan. Sedangkan alternatif kedua memberi pendekatan yang lebih sederhana dan intuitif, yaitu untuk mendekati pemain dan menjauhkan lawan dari kemenangan. Mendekatkan pemain ke kemenangan artinya memilih langkah yang menghasilkan langkah minimum untuk menang milik pemain paling kecil. Menjauhkan pemain ke kemenangan artinya memilih langkah yang menghasilkan langkah minimum untuk menang milik lawan paling besar. Untuk mengakomodasi dua kepentingan tersebut, dibuatlah parameter  $-p+1$  yang harus dimaksimalkan sesuai pembahasan di bagian sebelumnya.

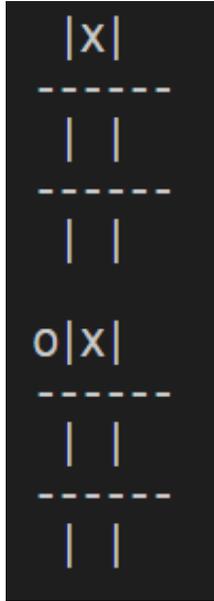
Berikut adalah potongan dari kode strategi *greedy* yang diimplementasi dalam bahasa Java. Implementasi *method* dengan nama yang trivial tidak dituliskan di sini.

```
public int[] getBestMove(String player){
    Board next;
    int max=-999;
    int step;
    int idx[] = {0,0};
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            if(board.isEmpty(i,j)){
                next = board.copy();
                next.set(player,i,j);
                step =
                    next.minStepToWin(player)
                    +next.minStepToWin(getOpponent(player));
                if(step>max){
                    max=step;
                    idx[0]=i;
                    idx[1]=j;
                }
            }
        }
    }
    return idx;
}
```

## III. HASIL PENGUJIAN DAN PEMBAHASAN

Pada bagian ini akan diperlihatkan beberapa hasil keputusan algoritma yang sudah dibuat dan penjelasan mengenai alasan mengapa langkah tersebut dipilih. Pada

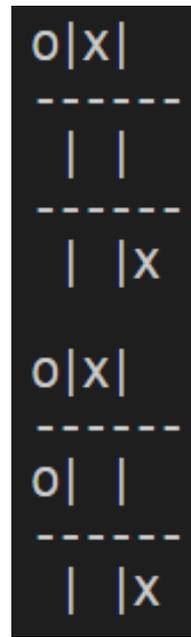
gambar di bawah, pemain (O) memilih baris pertama kolom pertama karena langkah minimum untuk menang milik O dan X sama-sama dua sehingga parameter  $-p+1$  bernilai 0. Sebenarnya ada beberapa posisi lain, seperti baris pertama kolom ketiga, juga memberikan nilai  $-p+1$  yang sama. Namun sebagai konvensi, bila ada posisi yang menghasilkan nilai  $-p+1$  yang sama, maka algoritma akan memilih posisi yang lebih awal ditelusuri.



Gambar 7 Ilustrasi pemilihan langkah yang rasional

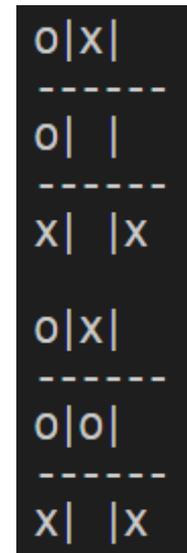
Langkah ini merupakan langkah yang cukup rasional bila dinilai menurut akal manusia.

Pada contoh lainnya, algoritma juga memberikan langkah yang cukup rasional seperti yang dapat dilihat pada gambar di bawah. Algoritma memilih baris kedua kolom pertama karena langkah minimum untuk menang milik pemain (O) bernilai 1 dan langkah minimum untuk menang milik lawan bernilai 2 sehingga nilai parameter  $-p+1$  adalah 1. Sebenarnya posisi lain, seperti baris ketiga kolom pertama, juga memberikan nilai parameter  $-p+1$  yang sama. Namun menurut konvensi yang telah dibuat, algoritma akan memilih posisi yang pertama kali ditemui.



Gambar 8 Ilustrasi langkah yang rasional

Namun, terdapat beberapa kasus di mana algoritma ini menghasilkan langkah yang tidak rasional seperti pada gambar di bawah ini. Pemain (O) memilih baris kedua kolom kedua karena langkah minimum untuk menang milik O adalah 1 dan langkah minimum untuk menang milik X adalah 1. Sehingga menghasilkan parameter  $-p+1$  sebesar 0 (yang dalam kasus ini sudah maksimal). Sebenarnya terdapat posisi lain yang menghasilkan  $-p+1$  sebesar 0 juga dan merupakan langkah yang lebih rasional. Posisi tersebut adalah baris ketiga kolom kedua yang menghasilkan langkah minimum untuk menang milik O dan X sebesar 2 sehingga nilai  $-p+1$  juga 0. Tetapi karena baris kedua kolom kedua ditelusuri terlebih dahulu, maka algoritma memilih posisi tersebut.



Gambar 9 Ilustrasi langkah yang tidak rasional

#### IV. PENUTUP

##### 1) Kesimpulan

- Permainan *tic tac toe* dapat dimainkan dengan strategi *greedy* dimana elemen-elemen *greedy* dari permainan ini telah dijelaskan di bagian II.A.
- Salah satu fungsi seleksi yang memungkinkan untuk permainan *tic tac toe* adalah memilih posisi yang memaksimalkan jumlahan dari negatif dari langkah minimum untuk menang milik pemain dengan langkah minimum untuk menang milih lawan.

##### 2) Saran

- Pada beberapa kasus masih belum didapatkan langkah yang rasional, umumnya disebabkan karena nilai maksimum parameter  $-p+1$  terdapat di beberapa posisi. Perlu strategi tambahan untuk memilih posisi yang tepat ketika beberapa posisi memiliki parameter  $-p+1$  yang sama.
- Perlu dicoba strategi yang sama namun untuk ukuran ataupun dimensi papan *tic tac toe* lainnya. Mengingat belum ada makalah yang membahas hal tersebut.

LINK VIDEO

<https://youtu.be/opAVIJguSZo>

#### UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan pada keluarga dan tim dosen IF2211 Strategi Algoritma yang telah membantu dalam penulisan makalah ini hingga selesai. Semoga makalah ini bisa membawa manfaat bagi banyak orang.

#### REFERENSI

- [1] T. Bolon, How to never lose at Tic-Tac-Toe, New York: Book Country, 2013.
- [2] R. Munir, "Algoritma Greedy (Bagian 1)," 2021. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf). [Accessed 21 May 2022].
- [3] A. H. Hutama, "Perancangan Lawan Bermain Permainan Tic-tac-toe dengan Menggunakan Algoritma Branch and Bound," 2016. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2015-2016/Makalah-2016/MakalahStima-2016-080.pdf>. [Accessed 21 May 2022].
- [4] Y. Valentino, "Penyelesaian Game Tic-Tac-Toe Dengan Algoritma Minimax," 2019. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018->

2019/Makalah/Makalah-Stima-2019-070.pdf. [Accessed 21 May 2022].

#### 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, 22 Mei 2022



Rizky Ramadhana P. K.  
13520151