

## **APLIKASI ALGORITMA MINIMAX PADA PERANCANGAN GAMES OHELLO (MINIMAX ALGORITHM APPLICATION ON DESIGN OF GAME OHELLO)**

**Muhammad Akbar Wiguna<sup>1</sup>, Agung Toto Wibowo<sup>2</sup>, Endro Ariyanto<sup>3</sup>**

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

---

### **Abstrak**

Game othello adalah game logic yang cukup terkenal yang berasal dari daerah jepang. Game ini dimainkan oleh dua orang. Game othello dimainkan pada arena papan kotak-kotak persegi dengan pin hitam dan putih di atas arena. Pin warna hitam harus melewati pin warna putih agar pin putih dapat diubah menjadi pin hitam dan sebaliknya. Permainan akan berakhir jika semua kotak arena sudah berisi pin, atau seluruh pin yang ada di atas arena berwarna sama. Pemenang adalah pemain yang memiliki jumlah pin yang lebih banyak di atas arena. Game ini menuntut artificial intelligence yang tinggi sehingga mulai tahun 1977 diadakan kejuaraan dunia hingga sekarang.

Game ini dibuat dalam 3 bentuk papan, 8x8, 14x14, dan 20x20. Game ini dirancang agar manusia dapat melawan Artificial Intelligence (AI). Oleh karena itu, pada sisi AI akan diterapkan algoritma minimax. Algoritma minimax merupakan algoritma khusus game yang menggunakan teknik adversarial search. Yaitu teknik menganalisis permainan berlawanan. Algoritma ini akan memaksimalkan posisi AI dan meminimalkan posisi human player. Algoritma minimax akan membaca kemungkinan langkah human player dan AI dalam bentuk tree. Kedalaman tree mempengaruhi kecerdasan AI dan performansi sistem. Pada tugas akhir ini, dilakukan analisis terhadap algoritma minimax, yaitu analisis performansi sistem yang didapat dari kecepatan rata-rata waktu eksekusi sistem, dan analisis kecerdasan AI dengan melakukan uji coba terhadap 15 orang sebagai lawan AI.

Berdasarkan observasi yang telah dilakukan, Pada analisis performansi sistem, algoritma minimax membutuhkan waktu 2,089 detik pada kedalaman keempat dari tree algoritma minimax. Pada analisis kecerdasan, AI telah dapat memenangi permainan dengan persentase 60% pada kedalaman keempat dari tree algoritma minimax. Selain itu, kebutuhan memori algoritma minimax adalah sebesar 412 byte dikali jumlah maksimum node tree yang dibangun algoritma minimax.

**Kata Kunci : Game Othello, Algoritma Minimax**

---

**Telkom**  
University

### Abstract

Othello Game is a fairly well-known logic game which originated from the Japanese. The game is played by two people. Othello Game is played on the arena board square boxes with black and white pin above the arena. Black pin must pass through a white pin to pin white pin can be changed into black and vice versa. The game ends when all the arena box already contains a pin, or all pins are there on the arena of the same color. The winner is the player who has a greater number of pins on the arena. This game requires a high artificial intelligence that began in 1977 world championships were held today.

This game is made in 3 shape of the board, 8x8, 14x14, and 20x20. This game is designed for humans to fight Artificial Intelligence (AI). Therefore, on the side of the AI will apply minimax algorithm. Minimax algorithm is a special algorithm that uses a technique game adversarial search. That is the opposite technique to analyze the game. This algorithm will maximize the position of AI and minimize human position player. Minimax algorithm will read the possibility of human players and AI step in tree shape. The depth of the tree affects the intelligence of the AI and the system performance. In this thesis, an analysis of the minimax algorithm, system performance analysis obtained from the average speed of execution time systems, and analysis of intelligence with AI carried out tests on 15 people as opposed to AI.

Based on the observations that have been done, the system performance analysis, minimax algorithm takes 2.089 seconds on the fourth of the tree depth minimax algorithms. In intelligence analysis, the AI has to win the game with a percentage of 60% on the fourth of the tree depth minimax algorithms. In addition, the minimax algorithm the memory requirement of 412 bytes is multiplied by the maximum number of tree nodes that are built minimax algorithm.

Keywords : Game Othello, Algoritma Minimax.

---

# 1. Pendahuluan

## 1.1 Latar Belakang

Kecerdasan buatan dapat diterapkan atau diimplementasikan ke dalam berbagai bentuk aplikasi. Bentuk implementasi yang paling mudah untuk diukur tingkat keberhasilan dan cukup digemari oleh sebagian besar orang yaitu pada *game* atau permainan. Salah satu *game* yang menuntut *artificial intelegent* yang bagus adalah *game othello*. *Game othello* adalah salah satu *game* tradisional yang cukup terkenal yang berasal dari daerah Jepang. *Game* ini dimainkan oleh dua orang.

Permainan ini di luar negeri lebih dikenal dengan nama *Reversi*. Saat ini *othello* tidak hanya dimainkan secara tradisional, namun sudah banyak dibuat dalam bentuk program pada komputer. Permainan-permainan berbasis komputer ini juga bermacam-macam. Salah satu kelebihanannya adalah tidak harus mencari orang untuk menjadi lawan tanding jika ingin bermain karena permainan berbasis komputer ini sudah mendukung *single player mode* dimana pemain dapat bermain sendiri melawan komputer yang dirancang untuk dapat berlaku seperti pemain manusia atau yang lebih dikenal dengan kecerdasan buatan atau *artificial intelegence (AI)*.

Untuk membuat pemain merasa seperti melawan pemain lainnya ketika melawan komputer, maka diperlukan suatu algoritma yang dapat membuat AI mampu mengambil keputusan yang terbaik agar dapat mengalahkan pemain atau setidaknya menghalangi pemain untuk menang. Oleh karena itu diterapkanlah algoritma *minimax*.

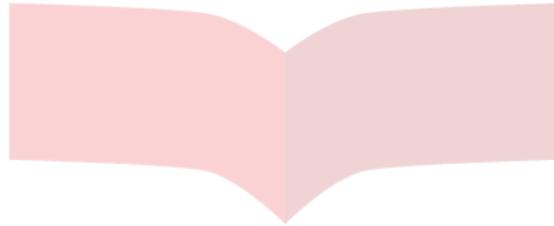
Algoritma *minimax* menggunakan teknik *adversarial search* yang merupakan teknik yang menganalisis permainan yang berlawanan untuk menentukan siapa yang dapat memenangkan permainan. Permainan ini dimainkan oleh dua orang tanpa adanya koalisi (berjalan bersama). Diterapkannya algoritma *minimax* karena algoritma ini memaksimalkan posisi pemain (*AI Player*) dan meminimalkan posisi lawan (*Human Player*). Pada algoritma *minimax*, akan dibangkitkan sebuah pohon permainan yang berisi semua kemungkinan langkah yang akan dipilih berikutnya oleh *AI player* dan *human player* sampai akhir permainan.

Dalam penentuan keputusan tersebut dibutuhkan suatu nilai yang merepresentasikan kerugian atau keuntungan yang akan diperoleh jika langkah tersebut dipilih. Oleh karena itu, pada algoritma *minimax* ini digunakan sebuah fungsi evaluasi (*evaluation function*) untuk membangkitkan nilai *fitness* yang merepresentasikan hasil permainan yang akan terjadi jika langkah tersebut dipilih. Dari nilai-nilai *fitness* inilah komputer akan menentukan simpul mana dari pohon permainan yang akan dipilih. Tentunya simpul yang akan dipilih tersebut adalah simpul dengan nilai *fitness* yang akan menuntun permainan ke hasil akhir yang menguntungkan bagi komputer.

Dalam penyusunan tugas akhir ini, akan diimplementasikan algoritma *minimax* sebagai salah satu cara untuk membangun *artifiacial intelegent* yang cukup bagus untuk menjalankan *game othello*.

## Daftar Istilah

<i>Game-Tree</i>	Pohon keputusan pada game, yang menggambarkan kemungkinan langkah dari Player1 dan Player2
<i>Node</i>	Simpul pada <i>tree</i> , yang dibangun dari <i>root</i> hingga daun



Telkom  
University

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil pengujian dan analisis yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut :

1. Tingkat kemenangan AI sudah mencapai 60% pada tingkat kesulitan 4 langkah pada game atau pada kedalaman ke 4 pada pencarian pohon algoritma *minimax* sehingga jika memiliki memori yang lebih besar, dapat menambah kedalaman pohon pencarian sehingga tingkat kemenangan lebih besar.
2. Dilihat dari segi komputasi, algoritma *Minimax* membutuhkan waktu yang cukup besar. Pada kedalaman 4 pohon pencarian algoritma *minimax* membutuhkan waktu 2,089 detik.

### 5.2 Saran

Saran yang dapat diberikan untuk melakukan pengembangan berikutnya yaitu :

1. Dapat menggunakan kapasitas memori/RAM yang lebih besar agar bisa menggunakan kedalaman yang lebih dalam pada tree algoritma *minimax* sehingga persentase kemenangan AI lebih besar.
2. Dapat dilakukan optimasi dengan algoritma *alpha-beta pruning*, *negascout* atau *MTDF* yang dapat memberikan hasil performansi yang lebih baik sehingga dapat mengurangi waktu eksekusi yang besar ketika AI dijalankan.

## Daftar Pustaka

- [1] Billings, Jessica. 2008. *"The Minimax Algorithm"*. CS 330, Spring 2008. Bryn Mawr College.
- [2] Department of Computer Science, Australian National University. 2009. *"Introduction to Programming and Algorithms"*.
- [3] Fang, Randi. 2003. *"Othello : From Beginner to Master"*. Dikutip : 16 Oktober 2009, [online].
- [4] Leberknight, david. 2003. *"Object oriented programming & design – minimax"*.
- [5] Minimax Explained AI Depot. Di unduh pada : <http://ai-depot.com/articles/minimax-explained/> , 2 Agustus 2010.
- [6] Minimax – Wikipedia The Free encyclopedia. Di unduh pada : <http://en.wikipedia.org/wiki/Minimax> , 3 agustus 2010.
- [7] Munir, Rinaldi. 2006. Strategi Algoritmik. Teknik Informatika ITB : Bandung.
- [8] Siswanto. 2000. "AI". STMIK Budi Luhur.
- [9] Suyanto, 2007. "Artificial Intelligence". Informatika : Bandung.
- [10] Reversi – Wikipedia The Free encyclopedia. Di unduh pada : <http://en.wikipedia.org/wiki/Reversi> , 3 agustus 2010.
- [11] Russel, Peter Norvig. "Artificial Intelligence: A Modern Approach, 2/e". Pearson Education India. ISBN 8177583670, 9788177583670.