IMPLEMENTASI ALGORITMA MINIMAX UNTUK ARTIFICIAL INTELEGENCE PADA PERMAINAN CATUR SEDERHANA

De Rosal Ignatius Moses Setiadi
Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Jl. Nakula I No. 5-11, Semarang
Email: moses.dinus@gmail.com

Abstrak

Aplikasi game komputer banyak digunakan oleh masyarakat. Beberapa game memerlukan orang lain untuk dapat dimainkan. Seperti pada jenis board game yang dimainkan oleh dua pemain, maka dibutuhkan metode untuk membuat pemain dapat merasa game dimainkan oleh dua pemain. Dalam paper ini, peneliti akan mengimplementasikan algoritma minimax dalam sebuah permainan catur mini dimana tiap pemain memiliki 7 bidak. Algoritma minimax umumnya menghitung semua kemungkinan yang ada di game, kadang-kadang sampai game selesai. Karena aplikasi yang dirancang dalam algoritma yang sederhana maka memerlukan penyederhanaan tetapi tidak mengurangi kemampuan secara signifikan. Jadi algoritma minimax dalam aplikasi ini akan diberi prioritas dan tidak harus menghitung semua kemungkinan yang ada jika ditemukan nilai maximum.
***Kata Kunci**: Board games, Minimax algorithm, Games tree algorithm*

Abstract

Games application in a computer that is widely used by all people. In some games it should be played with other players to be played. As in the type of game two player board games, and therefore needed a technique to make a player can feel the game as if performed by two players. In this paper we will be implemented the minimax algorithm in minichess game where each player has seven pawns. Minimax algorithms usually calculate all the possibilities that exist in the game, sometimes until the game finished. Because the application is designed in a simple algorithm would require a simplified but does not reduce performance significantly. So the minimax algorithm in this application will be given priority and not have to calculate all the possibilities that exist if it is found that the maximum value.

***Keyword**: Board games, Minimax algorithm, Games tree algorithm*

1. PENDAHULUAN

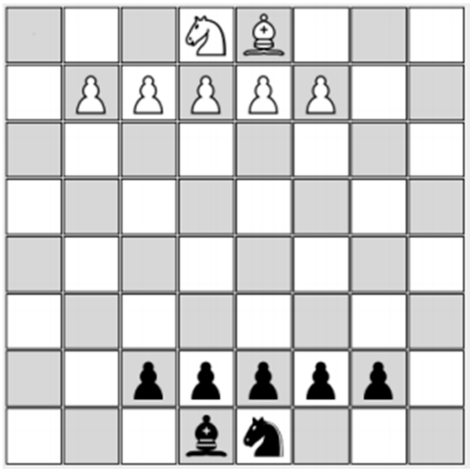
Aplikasi permainan yang ada dalam komputer merupakan aplikasi yang dikenal oleh banyak orang dari yang muda sampai yang tua. Aplikasi ini cukup mudah dan menyenenangkan sehingga sering digunakan untuk menghilangkan kepenatan. Penggunaan aplikasi permainan sering kali dilakukan secara personal atau seorang diri saja karena tidak ada lawan yang dapat diajak bermain bersama. Beberapa applikasi permainan harus dilakukan setidaknya oleh dua orang pemain terutama permainan *two player board games*, misalnya: permainan catur, tic tac toe, janggi, *battleship* dan lain sebagainya [1].

Artificial intelegence (AI) atau kecerdasan buatan biasanya digunakan

sebagai teknik untuk menggerakan komputer sebagai lawan bermain dalam applikasi permainan [2] [3]. Tujuan AI digunakan untuk mencari pola permainan, merancang startegi, mengkolaborasikan entity dalam kompter, dan berlajar dari pengalaman sebelumnya untuk mengambil keputusan. Dalam permaianan catur dapat dilihat bahwa kesuksesanan dari permainan merupakan kemenanggan dari satu pemain atau pemain lainya, dimana mencari hasil yang paling maksimal dari pemain dan meminimalisir hasil dari lawan [4]. Maka algoritma minimax diusulkan pada applikasi *board games* pada makalah ini.

Pada makalah ini akan banyak dibahas tetang salah satu jenis *two player board games*, yaitu applikasi permainan catur. Applikasi yang digunakan disini merupakan permainan catur sederhana yang hanya terdiri dari tujuh bidak. Ketujuh bidak tersebut terdiri dari lima pion, satu menteri dan satu kuda.

Pada makalah ini menggunakan bahasa pemrograman java untuk membangun program serta dengan algoritma minimax dan *tree* sebagai struktur data. Sesuai dengan judul makalah ini, program catur yang dibuat cukup sederhana karena masing-masing pemain hanya memiliki tujuh bidak saja, yang terdiri dari lima pion, satu kuda dan satu menteri. Permainan akan berakhir jika salah satu bidak kuda mati atau kelima pion mati. Berikut merukan posisi awal dari permainan:



Gambar 1. *Posisi awal permainan*

Aplikasi permainan catur sederhana yang dibuat dengan bahasa pemrograman java, dibuat dengan tujuan menerapkan dan menguji algoritma minimax dengan struktur data *tree* pada applikasi permainan catur sederhana. Dalam aplikasi ini algoritma minimax digunakan untuk mengatur strategi permainan, Sehingga player yang menggunakan applikasi ini seakan-akan dapat bermain dengan player lain.

2. TINJAUAN PUSTAKA

2.1. *Board Games* dan Catur

Board Games merupakan permainan yang melibatkan counter atau potongan dipindahkan atau diletakkan pada permukaan premarked atau "papan", menurut aturan permainan yang tersedia. Permainan dapat didasarkan pada strategi murni, kesempatan (dadu bergulir misalnya) atau campuran keduanya, dan biasanya memiliki tujuan yang harus dicapai pemain [5].

Catur merupakan salah satu jenis *two player strategy board games* yang masih sangat populer sampai saat ini. Catur adalah permainan mental yang dimainkan oleh dua orang. Pecatur adalah orang yang memainkan catur, baik dalam pertandingan satu lawan

satu maupun satu melawan banyak orang (dalam keadaan informal). Sebelum bertanding, pecatur memilih biji catur yang akan ia mainkan. Terdapat dua warna yang membedakan bidak atau biji catur, yaitu hitam dan putih. Pemegang buah putih memulai langkah pertama, yang selanjutnya diikuti oleh pemegang buah hitam secara bergantian sampai permainan selesai [6].

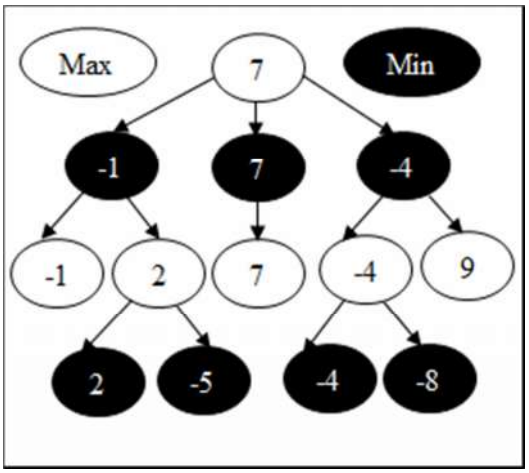
Ada berbagai jenis catur dengan gaya permainan yang berbeda pula. Akan tetapi catur yang dibahas merupakan catur yang digunakan pada umumnya yang terdiri dari delapan macam bidak dengan ukuran papan delapan kali delapan kotak. Dengan masing-masing pemain memiliki 16 bidak yang dapat dimainkan.

2.2. Kecerdasan Buatan

Kecerdasan buatan (AI) adalah kecerdasan mesin dan cabang ilmu komputer yang bertujuan untuk menciptakannya. Buku AI mendefinisikan bidang ini sebagai "*the study and design of intelligent agents*" dimana agen cerdas adalah sistem yang merasakan lingkungannya dan mengambil tindakan yang memaksimalkan peluang yang sukses [7], [8], [9]. John McCarthy, yang menciptakan istilah dalam. 1956 [7], [8], [10] mendefinisikan sebagai "ilmu dan teknik membuat mesin cerdas." [11] AI sangat banyak digunakan dalam applikasi komputer seperti applikasi permainan. Pada *two player board games strategy* AI digunakan untuk mengatur startegi dan memutuskan langkah sehingga dapat mengimbangi permainan player. Sehingga player yang memainkan applikasi ini seakan-akan dapat bermain dengan player lain.

2.3. Algoritma Minimax

Algoritma minimax merupakan salah satu algoritma yang digunakan pada permainan dua player yang memiliki AI atau pada *zero sum games* [12], [13], seperti catur. Pada algoritma minimax, pengecekan akan dilakukan untuk mencari seluruh kemungkinan yang ada bahkan dapat dilakukan sampai akhir permainan. Pengecekan tersebut akan menghasilkan pohon permainan yang berisi semua kemungkinan tersebut. Tentunya dibutuhkan resource yang berskala besar untuk menangani komputasi pencarian pohon solusi tersebut berhubung kombinasi kemungkinan untuk sebuah permainan catur pada setiap geraknya sangat banyak sekali.
Pada algoritma minimax komputer akan menganalisis seluruh pohon permainan. Dan untuk setiap langkahnya, komputer akan memilih langkah yang paling membuat lawan mendapatkan keuntungan minimum, dan keuntungan maksimum bagi komputer itu sendiri [4], [12], [13]. Dalam penentuan keputusan tersebut dibutuhkan suatu nilai atau bobot yang dapat merepresentasikan kerugian atau keuntungan yang akan diperoleh pada setiap lagkah, sehingga langka yang memiliki nilai terbesar (keuntungan terbesar dan kerugian terkecil) akan dipilih. Berikut merupakan gambaran dari algoritma minimax.



Gambar 2. Algoritma Minimax

3. PEMBAHASAN

3.1. Metode yang Diusulkan

	0	1	2	3	4	5	6	7
0					Mai			
1		P1ai	P2ai	P3ai		P5ai		
2					P4ai			
3								
4				P2pl				
5			Kai		P3pl	P4pl		
6							P5pl	
7				Mpl	Kpl			

Gambar 3. Contoh kasus untuk menentukan langkah AI selanjutnya

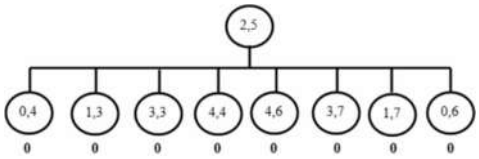
Keterangan:
K = Kuda
M = Menteri
P = Pion
pl = Player / pemain
ai = Komputer AI

Pertama kali sebelum menentukan langkah yang harus dipilih AI dilakukan pengecekan apakah posisi dari AI berbahaya dari semua bidak atau tidak, jika tidak maka AI akan mencari jalan terbaik untuk langkah selanjutnya.

Contoh:
Posisi kuda AI pada koordinat (2,5).

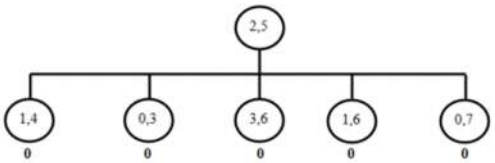
Formula untuk menentukan nilai adalah:

((Nilai AnakLv1 - bahaya2) + bahaya awal) + Nilai AnakLv2



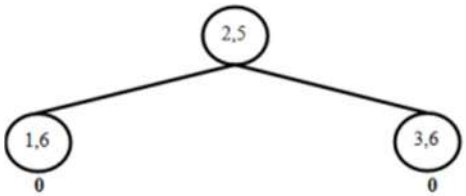
Gambar 4. *Nilai bahaya awal dari posisi kuda AI dari kuda player*

Keterangan:
0 = aman
17 = kuda AI dapat diserang oleh kuda player



Gambar 5. Nilai bahaya awal dari posisi kuda AI dari menteri player

Keterangan:
0 = aman
17 = kuda AI dapat diserang oleh menteri player



Gambar 6. Nilai bahaya awal dari posisi kuda AI dari pion player

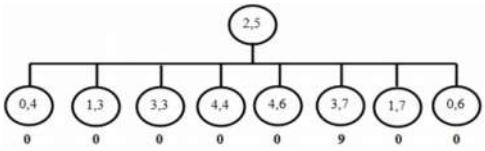
Keterangan:
0 = aman
21 = kuda AI dapat diserang oleh pion player

Dapat dilihat bahwa dari kasus diatas nilai bahaya awal dari kuda bernilai 0, yang berarti posisi kuda saat ini tidak berbahaya dan dapat menentukan langkah selanjutnya. Jika posisi kuda berbahaya, kuda akan mencari posisi yang paling

aman untuk menghindar dari serangan, disini merupakan salah satu prioritas yang dilakukan karena AI tidak melakukan pemanggilan algoritma minimax sebelum melakukan pengecekan terhadap keadaan AI sendiri.

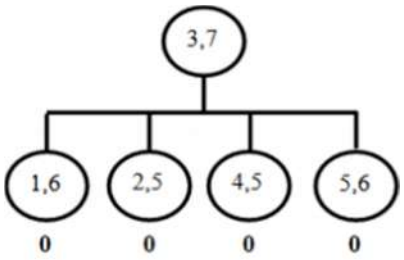
Untuk menentukan langkah selanjutnya yang akan dipilih kuda AI, maka algoritma minimax pada masing-masing bidak akan dipanggil, dengan urutan pemanggilan dari kuda, menteri dan terakhir pion. Berikut adalah langkah-langkah untuk menentukan langkah selanjutnya dari kuda AI. Pertama kali kuda AI akan melihat jalur yang dilewatinya dan melihat apakah jalur yang dilewati terdapat bidak musuh / player. Jika terdapat bidak player maka, jalur tersebut kemungkinan besar akan dipilih. Tetapi jalur itupun sebelum dipilih akan dipastikan dari kemungkinan bahaya yang dapat menyerang kuda, disinilah implementasi algoritma minimax. Berikut merupakan informasi nilai yang menentukan langkah kuda AI selanjutnya:

Menyerang Kuda = 10
Menyerang Menteri = 9
Menyerang Pion = 8
Tidak dapat menyerang =0

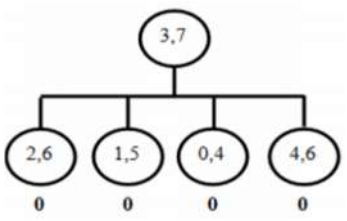


Gambar 7. Kemungkinan jalur yang dapat dilewati kuda AI

Keterangan:
Pada koordinat 3,7 bernilai 9 karena pada koordinat tersebut terdapat menteri dari player, maka koordinat ini akan dipilih, tetapi sebelum dipilih, akan dilakukan pengecekan dengan nilai yang sama dengan pengecekan awal. Akan tetapi jika koordinat tersebut tidak aman maka koordinat lain akan dipilih.

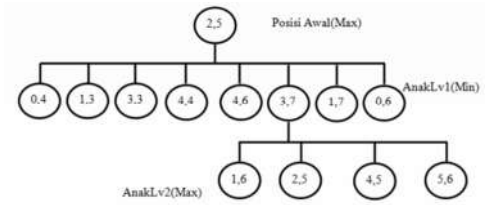


Gambar 8. Nilai bahaya2 dari posisi kuda AI dari kuda player dan nilai kemungkan jalur yang dilewati kuda AI



Gambar 9. Nilai bahaya2 dari posisi kuda AI dari menteri player

Dari gambar 8 dan gambar 9 diatas dapat dilihat bahwa koordinat 3,7 merupakan jalur yang paling menguntungkan untuk dipilih kuda AI dan paling merugikan bagi menteri player. Sedangkan pengecekan pada pion tidak dilakukan karena tidak ada pion yang dapat berada pada posisi 3,7.



Gambar 10. *Gambaran keseluruhan dari tree jalur yang akan dilewati kuda AI*

Berikut merupakan keterangan dari gambar:
Koordinat (1,6) → ((9-0)+0)+0)=9
Koordinat (2,5) → ((9-0)+0)+0)=9
Koordinat (4,5) → ((9-0)+0)+0)=9
Koordinat (5,6) → ((9-0)+0)+0)=9
Maka nilai minimal dari (3,7) adalah 9. Koordinat (3,7) akan dipilih karena memiliki nilai maksimum 9.

Untuk langkah selanjutnya akan dilakukan pengecekan pada semua bidak apakah

terdapat nilai yang lebih menguntungkan dari posisi tersebut.

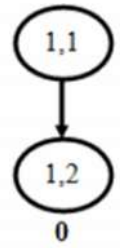
Posisi Menteri AI berada pada koordinat (0,4), akan tetapi pada posisi tersenut menteri tidak dapat bergerak kemana pun. Apabila menteri dapat berjalan, maka formula yang digunakan sama dengan kuda AI. Perbedaanya hanya model langkah yang digunakan yaitu kuda = langkah L dan mentri adalah langkah diagonal. Sedangan untuk pion sedikit berbeda dengan kuda dan meteri. Pada pion dilakukan yaitu bagaimana cara pion untuk bergerak maju atau bertahan dan bagaimana pion melakukan serangan. Adapun formula yang digunakan untuk melakukan serangan:

Nilai anak-nilai bahaya

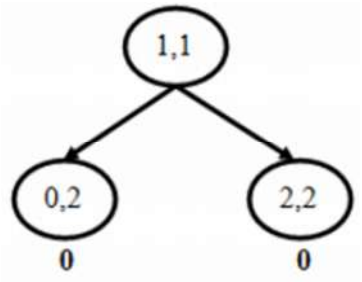
Sedangkan formula untuk melangkah maju atau betahan adalah:

Nilai bahaya parent - nilai bahaya child

Berikut merupan contoh gerakan yang dilakukan oleh Pion 1 AI yang berada pada koordinat (1,1).



Gambar 11. *Langkah maju atau bertahan pada P1ai*



Gambar 12. *Langkah menyerang pada Pion AI*

Keterangan:
Koordinat (1,2)` → 0 – 0 = 0
Koordinat (0,2) → 0 – 0 = 0
Koordinat (2,2) → 0 – 0 = 0

Karena nilai yang terbesar adalah nilai yang ada pada kuda AI, maka kuda AI yang akan dijalankan pada koordinat (3,7).

3.2. Hasil

Aplikasi yang dihasilkan akan dijalankan pada console sedangkan unruk graphical user interface (GUI), dijalankan dengan javax swing. Berikut merupakan gambaran dari bagaimana aplikasi permainan ini dijalankan.

1. Memulai Permainan

Gambar 13. *Tampilan awal saat memulai permainan*

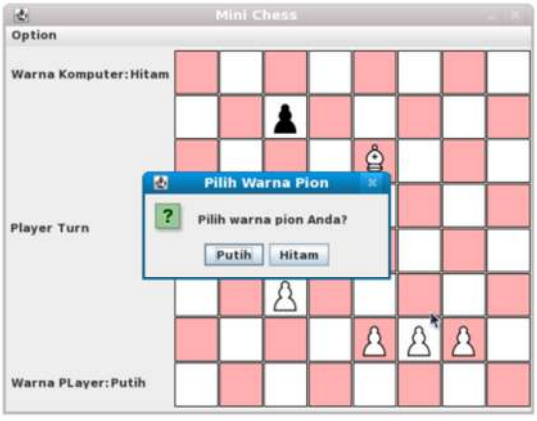
Gambar diatas merupakan tampilan awal saat permaiaan dilakukan. Sebelum meminkan permainan ini dapat memilih warna bidak dengan menekan tombol *option*.

2. Jendela Aturan Permainan

Gambar 14. *Gambar saat menampilkan jendela peraturan permainan*

Jendela aturan permainan dapat ditampilkan dengan memilih penuh option lalu memilih *rules of the game*.

3. Mengubah warna dan memulai permainan baru



Untuk memilih warna, player dapat memilih menu option > new game. Maka JOptionPane akan ditampilkan dengan memberikan button untuk memilih warna bidak. Langkah terakhir adalah memilih permainan baru.

4. Pergerakan kuda AI

Gambar 15. *Pergerakan kuda*

Gambar diatas menunjukan bagaimana kuda berjalan, ketika kuda di klik, maka warna koordinat yang akan dituju berwarna hijau.

5. Pergerakan Meteri AI

Gambar 16. *Tujuan yang mungkin dapat dilakukan oleh meneti.*

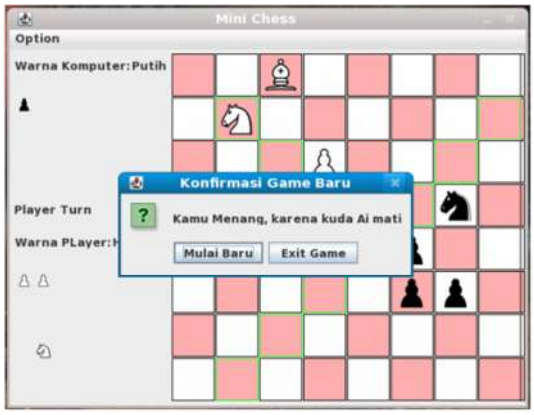
Gambar diatas menunjukan bagaimana menteri berjalanan, sedangkan oordinat yang berwarna hijau adalah area yang dapat digunakan untuk memindah menteri.

6. Pergerakan Pion AI

Gambar 17. Pergerakan pada pion

Dalam permainan ini pion tidak dapat berjalan lebih dari satu kali. Pion dapat bergerak maju, bertahan, dan menyerang pada bagian kiri atau kanannya.

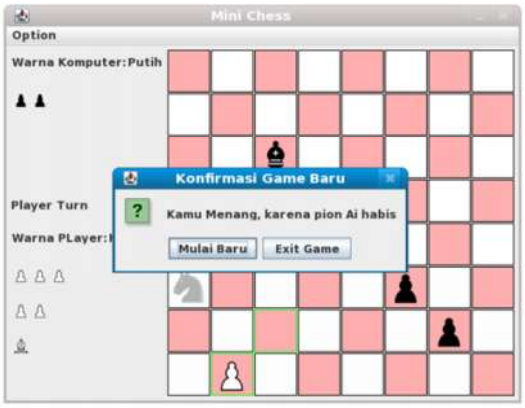
7. Player menang dengan membunuh kuda



Gambar 18. *Player menang ketika kuda AI berhasil dibunuh.*

Seperti pada peraturan pada aturan permainan. Player akan menang ketika membunuh kuda lawan. Selanjutnya JOptionPane digunakan untuk menampilkan pesan dan menghentikan permainan.

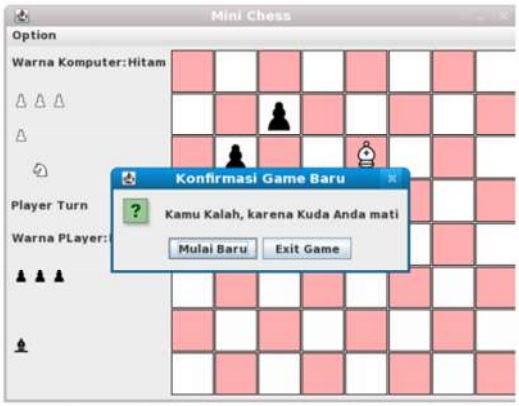
8. Player menang dengan membunuh semua pion



Gambar 19. *Player menang ketika seluruh pion AI dapat berhasil dibunuh.*

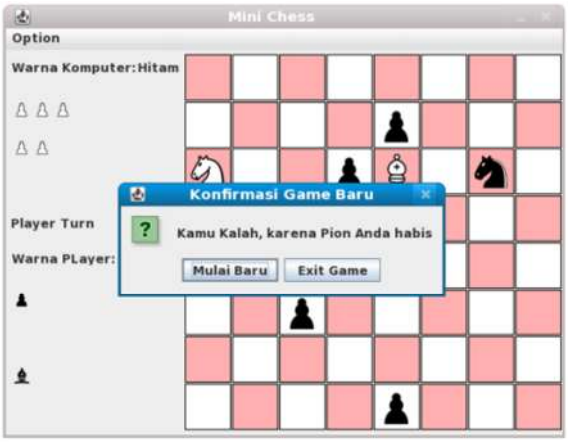
Selain membunuh kuda, permainan dapat berakhir ketika pion AI habis. Selanjutnya JOptionPane akan muncul danmenawarkan permainan baru datau keluar dari permaianan.

9. Player menang dengan membunuh kuda



Gambar 20. *AI menang ketika kuda Player berhasil dibunuh.*

10. Player menang dengan membunuh semua pion



Gambar 21. *AI kalah ketika pion AI habis.*

4. SIMPULAN

Dari makalah ini telah dibuat applikasi permainan sederhana. Dari hasil percobaan dapat disimpulkan sebagai berikut:

1. Applikasi dapat menerapkan algoritma minimax pada komputer AI.
2. Penggunaan algoritma minimax cukup efektif digunakan pada *two player boad games strategi* dan dapat mengimbangi permainan player.
3. Applikasi dapat dikembangkan lagi dengan kombinasi algoritma lain untuk meningkatkan efektifitas dan efesiensi kinerja AI.

DAFTAR PUSTAKA

[1] Wikipedia. (2012, May) List of board games - Wikipedia, the free encyclopedia. [Online]. http://en.wikipedia.org/wiki/List_of_board_games

[2] Millington I, *Artificial Intelegence for Games*. San Fransisko, United States of America: Morgan Kaufmann Publisers Inc, 2006.

[3] Smed J, *Algorithms and Networking for Computer*.: John Wiley & Sons, 2006.

[4] Silvia Garcia Diez, Jerrome Laforge, and Marco Saerens, "An Optimally Randomized Minimax Algorithm," February 2010.

[5] Wikipedia. (2012, May) Board game - Wikipedia, the free encyclopedia. [Online]. http://en.wikipedia.org/wiki/Board_game

[6] Wikipedia. (2012, May) Catur - Wikipedia bahasa Indonesia, ensiklopedia bebas. [Online]. http://id.wikipedia.org/wiki/Catur

[7] David Poole, Alan Mackworth, and Randy Goebel, *Computational Intelligence: A Logical Approach*. New York, United States of America: Oxford University Press, 1998.

[8] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, New Jersey, United States of America: Prentice Hall, 2003.

[9] Nils Nilsson, *Artificial Intelligence: A New Synthesis.*: Morgan Kaufmann Publishers, 1998.

[10] George Luger and William Stubblefield, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 5th ed.: The Benjamin/Cummings Publishing Company, Inc, 2004.

[11] John McCarthy. (2007, November)

What Is Artificial Intelligence? [Online]. http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

[12] Hamed Ahmadi. An Introduction to Game Tree Algorithms. [Online]. http://www.hamedahmadi.com/gametree/

[13] Wikipedia. (2012, May) Minimax - Wikipedia, the free encyclopedia. [Online]. http://en.wikipedia.org/wiki/Minimax