

Penerapan Algoritma Minimax Pada Permainan Checkers

Joel Parlingoman Aritonang^{1*}, Rinna Rachmatika¹

¹Fakultas Teknik, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46,

Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}aritonang.joel@gmail.com, ²rinnarachmatika@unpam.ac.id

(* : coresponding author)

Abstrak– Chekeres adalah salah satu jenis permainan game board, yang mana pada jenis game ini mengandalkan strategi sebagai elemen utamanya. Permainan checkers ini umumnya dimainkan oleh dua orang dengan memiliki tujuan adalah untuk menghabiskan kepingan yang dimiliki oleh lawan. Permainan checkers yang dibuat dengan menggunakan AI (Artificial Inteligent) tertentu menerapkan algoritma minimax. Algoritma minimax ini digunakan untuk menentukan pilihan agar memperkecil kemungkinan kehilangan nilai maksimal, yang akan mendeskripsikan dimana jika terdapat pemain yang mengalami pendapatan akan maka pemain lain akan mengalami kehilangan sebesar pendapatan tersebut. Untuk memperkecil lingkup pencarian pada algoritma minimax, dikembangkan suatu algoritma yang dinamakan alpha-betha sebagai solusi untuk mengurangi jumlah node pada pohon pencarian. Pada permainan checkers algoritma ini akan menentukan langkah yang diambil oleh AI agar menghasilkan pendapatan maksimum dengan mempertimbangkan kemungkinan langkah yang dapat dilakukan selanjutnya. Penerapan algoritma minimax dalam permainan checkers dibuat berdasarkan prosedur Minimax dengan tujuan untuk mendapatkan langkah terbaik dari posisi yang ada. Algoritma minimax memiliki dasar berupa zero-sum game, yang memiliki arti jika salah satu pemain dinyatakan menang maka pemain lainnya dinyatakan kalah, jika salah satu pemain mendapat 1 poin maka pemain lainnya mendapat -1 poin.

Kata Kunci: Checkers, Artificial Inteligent, (AI), Minimax, Zero-Sum, Alpha-Betha, Min, Max.

Abstract– *Chekeres is a type of board game, which in this type of game relies on strategy as its main element. This game of checkers is generally played by two people with the objective being to spend the opponent's pieces. A checkers game made using a certain AI (Artificial Intelligence) implements the minimax algorithm. This minimax algorithm is used to make choices in order to minimize the possibility of losing the maximum value, which will describe where if a player experiences income, other players will experience a loss of that income. To reduce the scope of the search on the minimax algorithm, an algorithm called alpha-betha was developed as a solution to reduce the number of nodes in the search tree. In the checkers game, this algorithm will determine the steps taken by AI to generate maximum income by considering the possible steps that can be taken next. The application of the minimax algorithm in the game of checkers is made based on the Minimax procedure with the aim of getting the best move from the existing position. The minimax algorithm is based on a zero-sum game, which means that if one player wins, the other player loses, if one player gets 1 point then the other player gets -1 point.*

Keywords: Checkers, Artificial Inteligent, (AI), Minimax, Zero-Sum, Alpha-Betha, Min, Max.

1. PENDAHULUAN

Permainan (*game*) sering memanfaatkan kecerdasan buatan (*artificial intelligence*) untuk memberikan tantangan berupa kesempatan untuk melawan mesin yang memiliki tingkat intelektualitas tertentu dalam berpikir. Minimax merupakan salah satu algoritma dalam bidang kecerdasan buatan yang dapat melakukan kalkulasi langkah dengan pertimbangan yang logis, berdasarkan aturan permainan yang digunakan, khususnya untuk permainan yang melibatkan dua pemain seperti permainan *checker*. Dalam penelitian ini, algoritma minimax akan diimplementasikan ke dalam permainan *checkers*, dalam bentuk sebuah komputer AI yang dapat mengambil keputusan langkah secara otomatis pada saat permainan berlangsung. Hasil implementasi algoritma minimax kedalam permainan *checker* yang diteliti pada penelitian ini mampu menunjukkan bagaimana proses berfikir (*self thinking*) dari komputer AI berdasarkan konfigurasi tingkat kesulitan yang dipilih oleh pengguna. (Uari et al., n.d.)

Untuk memaksimalkan algoritma yang lebih cocok dan mengimplementasikannya pada penempatan bidak dalam waktu yang sesingkat-singkatnya. Penelitian ini menggunakan metode eksperimen berupa studi kepustakaan. Aspek-aspek penelitian ini meliputi fitur-fitur yang dibutuhkan seperti *interface*, validasi *input* dan desain *output*. Hasil dari perancangan ini adalah

penerapan Permainan *Checkers* yang disematkan pada algoritma negascout dan algoritma minimax. Pengujian yang digunakan *White Box* adalah *Input*, *Output*, fungsi program, dan kecepatan algoritma negascout dan algoritma minimax dalam penempatan bidak di permainan *Checkers*. Setelah dilakukan pengujian, didapatkan hasil bahwa masing-masing algoritma memiliki kelebihan dan kekurangannya masing-masing, dimana pengujian dilakukan melalui peletakan gadai dengan waktu tercepat. Setelah pengujian perbandingan selesai, penulis menyimpulkan bahwa dalam penelitian ini, algoritma minimax memiliki waktu gerakan bidak yang lebih baik dalam penempatan, sedangkan algoritma negascout memiliki keputusan langkah yang lebih baik. (David, 2013)(Kurniawan Effendi et al., n.d.)(Plaat & Schaeffer, n.d.)

Setelah mengetahui perbandingan dari kedua algoritma tersebut maka ditemukanlah solusi bahwa Algoritma minimax merupakan salah satu contoh algoritma yang sangat sering dipakai untuk permasalahan tersebut. Dan permainan dapat menjadi salah satu contoh yang baik dan cukup sederhana untuk kita mengerti bagaimana cara kerja dan efeknya. Berdasarkan uraian di atas, penulis bermaksud untuk menerapkan algoritma minimax pada permainan *checkers*. Oleh karena itu, penulis mengambil topik tugas akhir dengan judul “*Penerapan Algoritma Minimax Pada Permainan Checkers*”.

2. METODOLOGI PENELITIAN

2.1 Metode Pengumpulan Data

Dalam melakukan penelitian ini, metode yang dipakai untuk pengumpulan data, menggunakan metode:

a. Wawancara

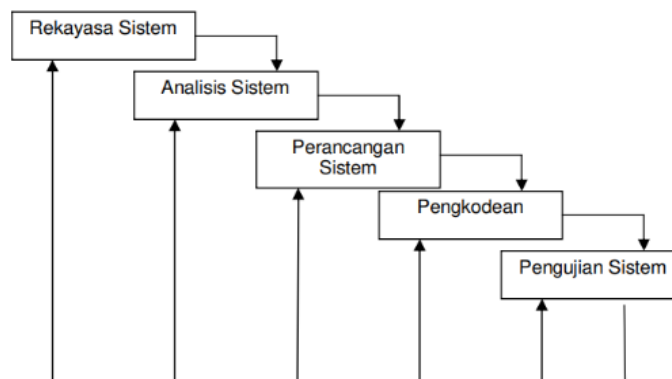
Metode ini dipakai untuk mengumpulkan informasi-informasi dari narasumber yaitu teman-teman serta saudara saya, mengenai pemahaman mereka dalam menyelesaikan suatu permainan khususnya permainan yang menggunakan strategi seperti *checkers*. Dari hasil wawancara yang saya dapatkan, dapat disimpulkan bahwa kebanyakan dari mereka, biasanya menyelesaikan suatu permainan strategi hanya menggunakan pemikiran serta pemahaman mereka saja. Padahal ada teknik algoritma yang dapat digunakan untuk menyelesaikan permainan tersebut.

b. Studi Pustaka

Metode ini digunakan untuk mengumpulkan data-data kepustakaan, yang mana data tersebut berupa teori mengenai aturan permainan *checker*, kecerdasan buatan (*artificial intelligence*), serta optimasi algoritma minimax dalam permainan *checkers*.

2.2 Metode Pengembangan Sistem

Dalam metode perancangan penelitian ini penulis menggunakan rancangan sistem aplikasi pengembangan perangkat lunak dengan model air terjun (*Waterfall*), Pada model air terjun sistem aplikasi pengembangan dilakukan pertahap dengan memberlakukan bahwa tahap berikutnya dapat dijalankan setelah tahap sebelumnya telah selesai maka dapat dikatakan semua tahap saling berhubungan satu dengan yang lainnya.



Gambar 1. Alur Metode *Waterfall*

3. ANALISA DAN PEMBAHASAN

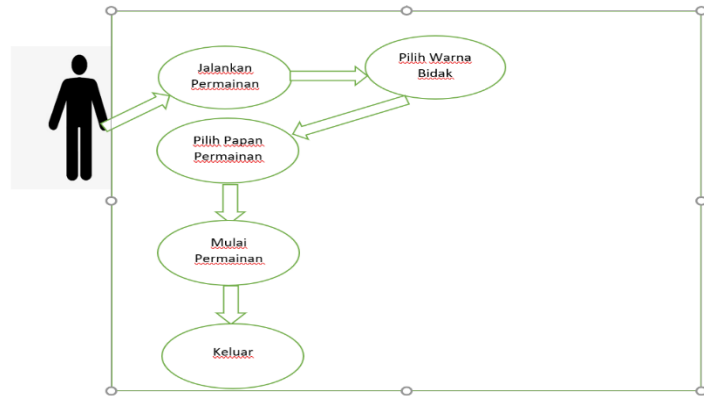
3.1 Analisis Masalah

Strategi di belakang Minimax algoritma adalah komputer berasumsi bahwa kedua pemain akan main sejauh kemampuan mereka. Maka, jika lawan mempunyai pilihan suatu gerak tidak baik atau suatu yang baik gerak, komputer akan mempunyai lawan memilih yang baik gerak. sungguh mencoba untuk memilih gerak yang mengakibatkan nilai titik yang paling rendah. Konsep relatif nyata dan sederhana ini menjadi rahasia di belakang minimax pohon. Jika komputer di program untuk mencari maksimal, maka rangkaian atau gerakan yang terbaik akan selalu ditemukan yang mengakibatkan nilai titik yang paling tinggi. Demikian juga dengan minimal, maka rangkaian atau gerakan yang terbaik akan selalu ditemukan yang mengakibatkan nilai titik yang paling rendah. (Ritonga & Rony, 2016).

3.2 Desain Sistem

3.2.1 Use Case Diagram

Adapun rancangan *use case diagram* yang dibangun pada aplikasi game permainan *checkers* sebagai berikut.



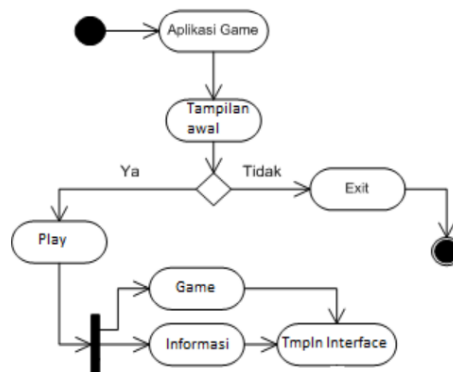
Gambar 2. Use Case Diagram Permainan Checkers

3.2.2 Activity Diagram

Activity diagram adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan. Pada pemodelan UML, diagram ini dapat digunakan untuk menjelaskan alur kerja sistem.

a. Activity Diagram Menu Game

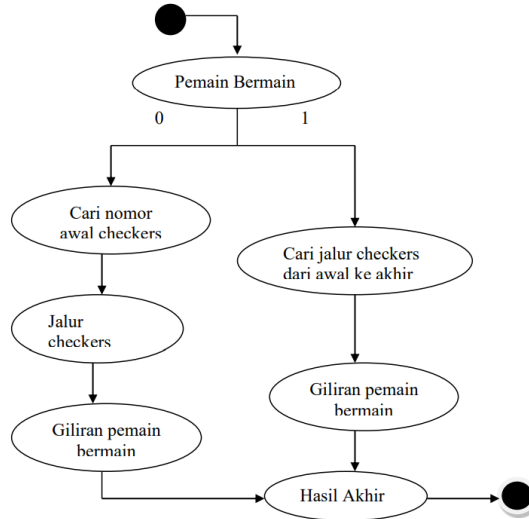
Berikut ini merupakan *activity diagram menu game* pada aplikasi permainan *checkers* dengan metode atau algoritma minmax yang dirancang.



Gambar 3. Activity Diagram Permainan Checkers

b. Activity Diagram Permainan Checkers

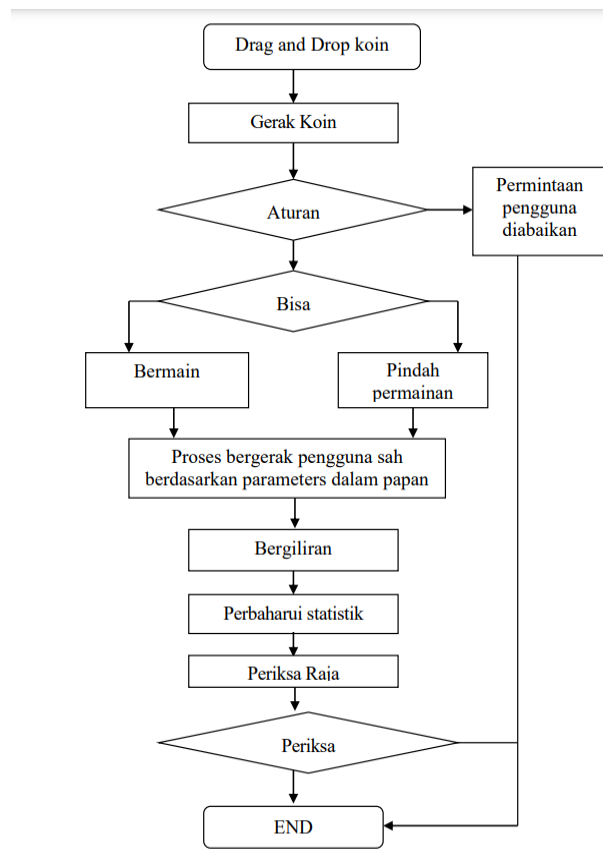
Berikut ini merupakan *activity diagram* pada aplikasi permainan *checkers* yang dirancang.



Gambar 4. Activity Diagram Aplikasi Permainan Checkers

3.3 Flowchart Permainan Checkers

Berikut ini merupakan *flowchart* permainan pada aplikasi permainan *checkers* yang dirancang :



Gambar 5. Flowchart Aplikasi Permainan Checkers

4. IMPLEMENTASI

4.1 Tampilan Hasil

Pada bab ini akan dijelaskan tampilan hasil dari aplikasi yang telah dibuat, yang digunakan untuk memperjelas tentang tampilan-tampilan yang ada pada aplikasi *game checkers* dengan metode minimax. Hasil dari perancangan permainan ini yang pertama di tampilkan ketika di jalankan ialah tampilan *form menu* utama permainan yang terdiri dari pilihan mulai game, pilihan warna bidak, pilihan ukuran papan permainan dan keluar. Pilihan mulai game digunakan untuk memulai permainan sedangkan pilihan warna bidak sebagai penentu bidak yang ingin digunakan pemain dalam melawan AI dan pilihan ukuran papan untuk mengetahui papan yang ingin digunakan oleh pemain dalam melawan AI, baik itu 8x8 atau 12x12.

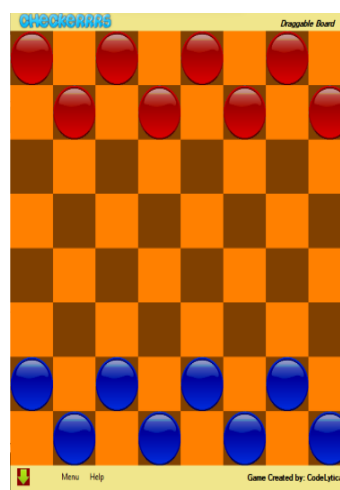
4.1.1 Tampilan Form Awal Permainan



Gambar 6. Tampilan Awal *Game Checkers*

Form ini muncul setelah program di *debuging* atau dijalankan. Pada tampilan ini, terdapat menu tampilan pilih warna bidak, pilih papan, *play* dan keluar.

4.1.2 Tampilan Permainan *Checkers*



Gambar 7. Tampilan *Game Checkers*

Form ini menampilkan data *game checkers* ketika kita memilih *play* permainan. Setelah kita meng-klik *menu button play*, maka kita akan langsung memulai permainan, dimana disini kita akan melangkah terlebih dahulu.

4.1.3 Tampilan Form Menu



Gambar 8. Form Menu Game Checkers

Form ini menampilkan pilihan-pilihan yang dapat digunakan oleh pemain mulai dari *menu new game*, *main menu*, *restart game*, dan *exit*.

5. KESIMPULAN

Kesimpulan yang dapat diambil berdasarkan hasil analisis perancangan, implementasi, pengujian, dan evaluasi pada judul penelitian “Penerapan Algoritma Minimax Pada Permainan Checkers” yaitu sebagai berikut:

- a. Dalam pembuatan permainan *checkers* menggunakan *AI*, permainan ini menerapkan algoritma minimax. Algoritma minimax memiliki dasar berupa *zero-sum game*, artinya jika salah satu pemain dinyatakan menang maka pemain lainnya dinyatakan kalah, jika salah satu pemain mendapat 1 poin maka pemain lainnya mendapat -1 poin. Dalam jenis permainan ini setiap pemain memiliki pengetahuan penuh tentang kondisi pemain lain, seperti setiap gerakan yang dimiliki pemain lain, pion apa dan posisi pion pemain lain.
- b. Hasil penerapan algoritma minimax dengan menambahkan nilai *Alpha-Betha*, algoritma minimax akan memiliki pohon pencarian secara lebih singkat, Algoritma ini digunakan untuk menentukan pilihan agar memperkecil kemungkinan kehilangan nilai maksimal, yang akan mendeskripsikan dimana jika terdapat pemain yang mengalami pendapatan akan maka pemain lain akan mengalami kehilangan sebesar pendapatan tersebut.

REFERENCES

- Coppin, Ben. (2004). *Artificial Intelligence Illuminated*. Jones And Bartlett Publishers.
- David. (2013). PERBANDINGAN ALGORITMA MINIMAX DAN NEGASCOUT PADA PERMAINAN CATUR SEDERHANA. In *Jurnal Informatika Mulawarman* (Vol. 8, Issue 2).
- Kurniawan Effendi, A., Delima, R., & C, A. R. (N.D.). *03-Implementasi Algoritma Negascout Untuk Permainan Checkers*.
- Kurniawan, M., Pamungkas, A., & Hadi, S. (2016). *ALGORITMA MINIMAX SEBAGAI PENGAMBIL KEPUTUSAN DALAM GAME TIC-TAC-TOE*. 6–7.
- Kusumadewi, S. (2003). *Artificial Intelligence (Teknik Dan Aplikasinya)*.
- Plaat, A., & Schaeffer, J. (N.D.). *An Algorithm Faster Than Negascout And SSS* In Practice*. [Http://www.This.Net/~Frank/Pstill.Html](http://www.this.net/~frank/pstill.html)
- Ritonga, & Rony, M. (2016). *Perancangan Game Checkers Dengan Penerapan Algoritma Minimax*.
- Syapnika, D., & Siagian, E. R. (2015). PENERAPAN ALGORITMA MINIMAX PADA PERMAINAN CHECKERS. In *JURIKOM* (Vol. 2, Issue 6).
- Uari, I., Muhazir, A., Alam, H., & Santri Kusuma, B. (N.D.). *ANALISIS KECERDASAN BUATAN PADA PERMAINAN CHECKER MENGGUNAKAN OPTIMASI ALGORITMA MINIMAX*.
- Waruwu, C., & Purba, D. (2017). *Implementasi Algoritma Minimax Dalam Game Othello*.