

Analisis Algoritma *Minimax* Menggunakan Optimasi *Alpha-Beta Pruning* Pada Permainan Tic-Tac-Toe

Nur Wilda Angraini Sary
Fakultas Sains dan Teknologi
Universitas Al-Azhar Indonesia
Jakarta, Indonesia
nwildaas@gmail.com

Abstrak— Dengan semakin berkembangnya ilmu pengetahuan dan teknologi muncul berbagai permainan berbasis komputer yang menggunakan prinsip kecerdasan buatan (*Artificial Intelligence*). Tic-Tac-Toe merupakan permainan sederhana yang umum dikenal. Permainan ini dimainkan oleh dua orang dimana pemain saling mengatur strategi agar lawan tidak memenangkan permainan atau paling tidak menghasilkan hasil yangimbang. Permainan Tic-Tac-Toe ini termasuk dalam *zero-sum game* yaitu jika seseorang menang, maka pemain yang lain kalah. Jika total keuntungan dari pemain dan total kerugian dikurangi maka akan berjumlah nol.

Untuk membuat permainan Tic-tac-toe tidak terkalahkan atau minimal seri, diperlukan algoritma yang dapat menghitung semua kemungkinan pergerakan pada komputer. Algoritma *Minimax* merupakan algoritma yang digunakan untuk pengambilan keputusan dengan tujuan memperkecil kemungkinan kehilangan nilai maksimal. *Alpha-Beta Pruning* merupakan cara yang digunakan untuk optimasi jumlah simpul yang dieksplorasi oleh algoritma *Minimax*. Penerapan algoritma *Minimax* dengan dan atau tanpa optimasi *Alpha-Beta Pruning* pada permainan Tic-Tac-Toe bertujuan untuk membandingkan apakah penggunaan optimasi tersebut dalam menghitung kemungkinan pergerakan pada papan Tic-Tac-Toe yang tersedia untuk *AI player* merupakan langkah dan solusi terbaik (menang atau minimal seri).

Kata Kunci —Permainan, Tic-Tac-Toe, Algoritma, *Minimax*, *Alpha-Beta Pruning*

I. PENDAHULUAN

A. Latar Belakang

Game merupakan sebuah aktivitas hiburan dengan tujuan untuk bersenang-senang, mengisi waktu luang, atau sekedar melepas lelah. Permainan biasanya dimainkan secara individu ataupun bersama-sama (kelompok). Seiring berkembangnya IPTEK (Ilmu Pengetahuan dan Teknologi), saat ini terdapat banyak permainan berbasis komputer yang menggunakan prinsip kecerdasan buatan (*Artificial Intelligence*).

Konsep kecerdasan buatan (*Artificial Intelligence*) dibuat dan diterapkan kedalam suatu mesin komputer agar dapat melakukan pekerjaan dan berpikir layaknya manusia. Permainan *Tic-Tac-Toe* merupakan salah satu permainan yang menggunakan prinsip kecerdasan buatan tersebut.

Permainan Tic-Tac-Toe ini termasuk kedalam *zero-sum game*. Dalam permainan dengan informasi jelas, setiap pemain tahu hasil dari semua pergerakan sebelumnya[7] dikarenakan hanya satu pemain yang mendapatkan poin maksimal. Sedangkan pemain lainnya mendapatkan hasil yang rendah [1]. Jika pemain menang maka pemain mendapatkan poin +1 dan pemain lainnya kalah mendapat poin -1 atau bisa juga 0 jika keduanya tidak dapat mengalahkan salah satunya. Permainan Tic-Tac-Toe dimainkan oleh dua pemain yaitu *AI player* dan manusia dengan bidak X dan O. Permainan tersebut dimainkan sedemikian hingga terbentuk garis sejajar secara vertikal, horizontal, dan diagonal.

Dalam penerapan ini batasan yang akan dilakukan adalah pada papan berukuran 3x3 dengan permainan yang dapat dimainkan oleh dua orang pemain yaitu *AI player* dan *human player*.

B. Rumusan Masalah

Pada tugas akhir makalah dengan judul “Analisis Algoritma *Minimax* Menggunakan Optimasi *Alpha-Beta Pruning* pada Permainan Tic-Tac-Toe” memiliki rumusan masalah sebagai berikut :

1. Bagaimana penerapan algoritma *minimax* dengan *alpha-beta pruning* pada permainan Tic-Tac-Toe untuk langkah keputusan *AI player* agar menang atau minimal seri?
2. Apakah penggunaan *alpha-beta pruning* menghasilkan solusi optimal dibanding hanya dengan menggunakan algoritma *minimax* saja?

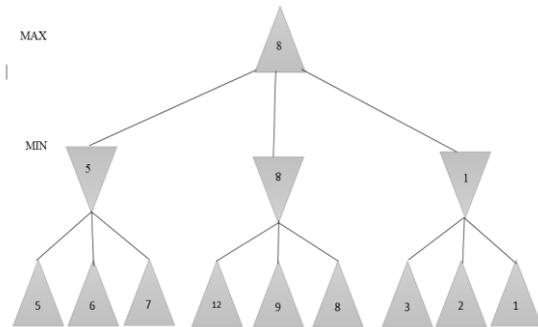
C. Tujuan

Tujuan dari penulisan ini adalah untuk mengetahui langkah yang akan diambil *AI player* dalam mengambil keputusan agar tidak dapat dikalahkan atau minimal dengan nilai seri. Dan mengetahui apakah penggunaan *Alpha-Beta Pruning* merupakan solusi untuk mencapai hasil yang paling optimal dibanding hanya menggunakan algoritma *minimax* saja.

II. LANDASAN TEORI

A. Algoritma Minimax

Min-max adalah algoritma pengambilan keputusan yang menggunakan teori keputusan, teori permainan untuk menghitung langkah optimal [1]. Algoritma minimax menggunakan prinsip algoritma eksplorasi *depth-first* dengan kedalaman terbatas dari pohon *game* [3]. Algoritma minimax (Gambar1) menghitung keputusan minimax dari keadaan saat ini. Hal ini menggunakan perhitungan rekursif sederhana nilai minimax dari masing-masing keadaan yang terganti, yang secara langsung menerapkan persamaan yang menentukan. Rekursi berlanjut sampai ke daun pohon, dan kemudian nilai minimax disimpan sampai pengulangan berhenti [3]. Dengan asumsi lawan yang sempurna untuk permainan deterministik, strategi yang layak adalah pindah ke posisi dengan nilai minimum yang tertinggi. Melalui nilai tersebut dapat dicapai hasil terbaik [2].



Gambar 1

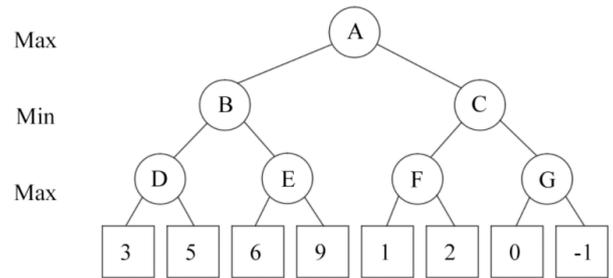
a) Keputusan Optimal Dalam Permainan

Dalam pohon permainan, strategi optimal dapat ditentukan dari nilai minimax masing-masing simpul, yang di tulis sebagai Minimax (n). Nilai minimax dari simpul adalah utilitas (untuk MAX) sebagai keadaan yang sesuai, dengan asumsi bahwa kedua pemain bermain optimal sampai akhir permainan. Nilai minimax pada setiap pangkalan merupakan nilai utilitas. MAX mengacu pada kondisi nilai *maximum* (α), sedangkan MIN mengacu kepada kondisi dari nilai minimum (β) [3].

B. Alpha-Beta Pruning

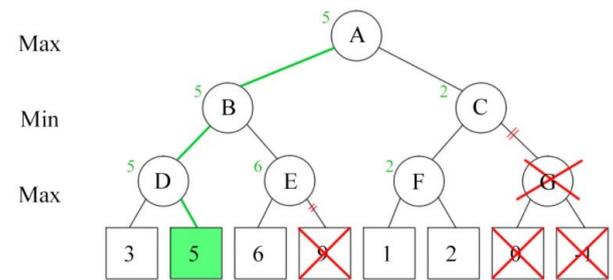
Alpha-Beta Pruning sebenarnya bukan algoritma baru, metode untuk mengurangi jumlah node yang dieksplorasi oleh algoritma minimax [2] menggunakan nilai *alpha* dan *beta*. *Alpha-Beta Pruning* merupakan teknik untuk mengurangi waktu komputasi dengan faktor pencarian yang sangat besar. Hal ini memungkinkan pencarian lebih cepat dan bahkan masuk ke tingkat yang lebih dalam di pohon permainan. Teknik tersebut memotong cabang di pohon permainan yang tidak perlu dicari karena sudah ada langkah yang lebih baik yang tersedia. Maka teknik ini disebut *Alpha-Beta Pruning* karena melewati dua parameter tambahan dalam fungsi minimax, yaitu *alpha* (α) dan *beta* (β) [3,4]. Selama pencarian

berlangsung nilai *alpha* (α) dan *beta* (β) akan selalu di perbaharui.



Gambar 2

Penggunaan optimasi algoritma minimax dan Alpha-beta pruning



Gambar 3

Pada (Gambar 3.2) [4] inisial panggilan awal dari A. Nilai alpha di sini adalah $-\infty$ dan nilai beta adalah $+\infty$. Nilai ini diturunkan ke simpul berikutnya pada pohon. Pada A *maximizer* harus memilih nilai max dari B dan C, jadi A memanggil B terlebih dahulu. Pada B *minimizer* harus memilih nilai min D dan E dan dimulai pada D. Pada D, terlihat pada anak kirinya yang merupakan simpul daun. Node ini mengembalikan nilai 3. Sekarang nilai alpha pada D adalah max ($-\infty, 3$) yaitu 3. D sekarang terlihat pada anak kanannya yang mengembalikan nilai 5. Pada D, $\alpha = \max(3, 5)$ yaitu 5. Sekarang nilai simpul max D adalah 5. Sekarang E melihat anak kirinya yaitu 6. Pada E, $\alpha = \max(-\infty, 6)$ yaitu 6. Disini kondisinya menjadi *true* $\beta = 5$ dan $\alpha = 6$. Jadi $\alpha \leq \beta$ benar. Setelah itu E kembali ke B. Perhatikan bagaimana pencarian tidak peduli nilai anak kanan E karena *minimizer* harus bernilai 5 atau lebih rendah. Jadi begitu *maximizer* melihat 6 *minimizer* akan memilih mendapatkan 5 di sisi kiri B. Begitu juga dengan 9 pencarian yang di potong. E mengembalikan nilai 6 ke B. Pada B, $\beta = \min(5, 6)$ yaitu 5. Nilai simpul B juga 5. Kemudian A memanggil C. C memeriksa simpul F terlebih dahulu dan mendapatkan nilai max 2. Maka nilai minimal pada C saat itu ≥ 2 , jadi simpul G langsung dipotong atau diabaikan. Setelah itu A memandingkan nilai max dari terendah pada B dan C (5,2) dan didapat nilai $\alpha \beta$ adalah 5.

III. IMPLEMENTASI DAN HASIL

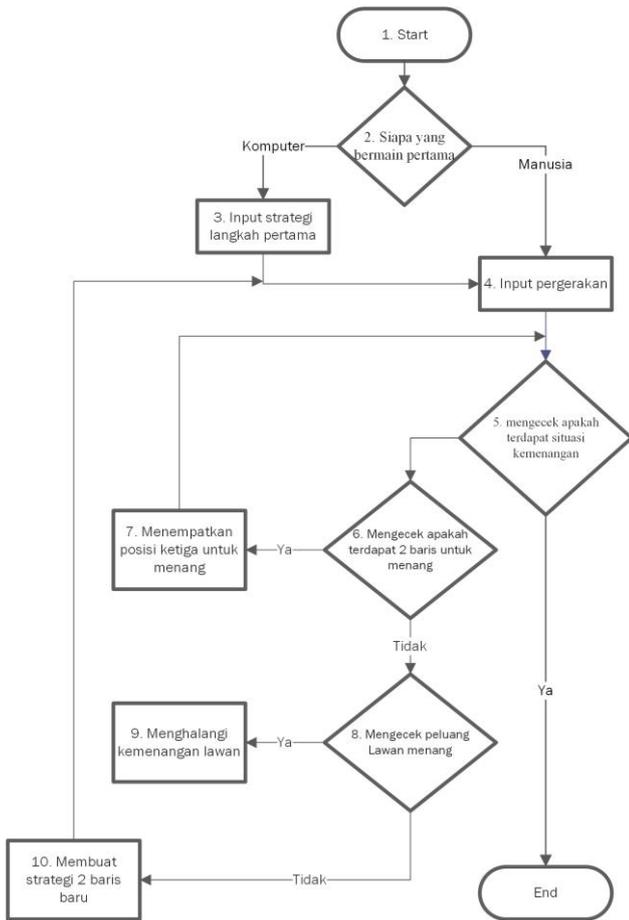
A. Rancangan Sistem

a) Strategi Untuk Algoritma Permainan

Untuk menang atau mencegah kekalahan AI player harus secara konsisten melakukan langkah-langkah sesuai prioritas. Prioritas tersebut berupa :

1. Membuat tiga baris diagonal, vertikal, atau horizontal.
2. Menahan lawan agar tidak membentuk tiga baris sempurna yaitu diagonal, vertikal, atau horizontal.
3. Menciptakan strategi yang dapat mempunyai dua kemungkinan menang
4. Mencegah posisi lawan yang mempunyai pola yang bisa membuatnya menang.

b) Strategi algoritma Minimax

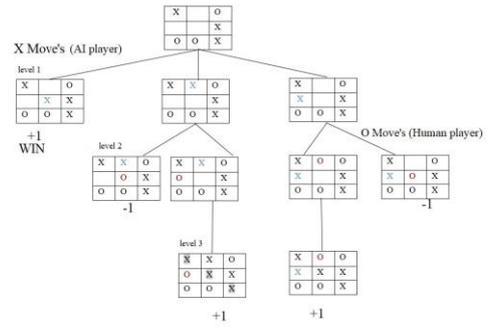


Gambar 3.1

Flow chart untuk permainan Ti-Tac Toe dengan Algoritma *minimax*

B. Implementasi

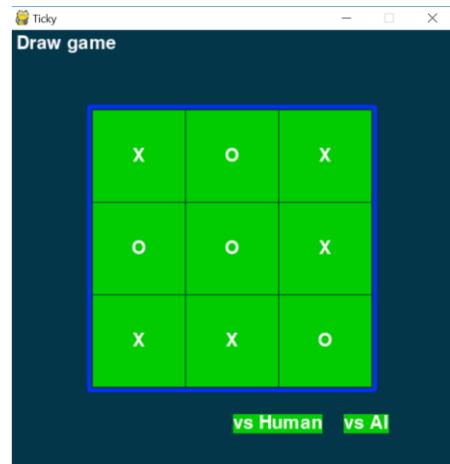
a) Implementasi Algoritma Minimax Tic-Tac-Toe



Gambar 3.2 Kondisi permainan pada saat terdapat tiga kemungkinan langkah pencarian untuk AI player

Pada (Gambar 3.1) ilustrasi diatas tersedia tiga kemungkinan pilihan pencarian untuk AI player. Dalam simulasi ini AI player sebagai "X" mendapat giliran untuk menentukan langkah yang akan dipilih. Jika AI player menempatkan posisi X di posisi papan (2,2) pada level 1 maka AI player akan langsung menangkan permainan. Kemungkinan kedua di level 1, jika AI player menempatkan X pada posisi papan di (1,2) maka ada kemungkinan O sebagai "human player" memenangkan permainan tersebut dengan menempatkan O di posisi (2,2). Namun jika O ditempatkan di posisi (2,1) maka X dapat menempatkan posisinya di (2,2) pada level 3. Dan pada kemungkinan ketiga di level 1, jika X ditempatkan pada posisi (2,1) ada dua kemungkinan O akan menempatkan posisinya di (2,1) yang artinya X akan menang atau O menempatkan di (2,2) yang artinya X kalah. Dari simulasi tersebut pada saat papan permainan dalam keadaan memiliki tiga peluang untuk menang atau kalah adalah memilih kemungkinan pertama pada posisi (2,2).

Dalam pengimplementasian menggunakan Algoritma minimax skala 3x3 digunakan program yang berasal dari GitHub [6] menggunakan bahasa python dengan modul gamepy .



Gambar 3.3

Permainan ini dimainkan sebanyak 40 kali dengan komputer vs komputer dengan hasil selalu draw

b) Implementasi Algoritma Minimax menggunakan Alpha-Beta Pruning pada Tic-Tac-Toe

Dalam pengimplementasian algoritma ini digunakan program yang berasal dari GitHub [5] menggunakan bahasa Java.

c) Studi kasus implementasi permainan pada algoritma minimax dengan atau tanpa alpha-beta pruning

1. Memainkan permainan tersebut sebanyak 40 kali secara acak kemudian menghitung rata-ratanya.

C. Hasil Studi Kasus

Ukuran Board	Variable Waktu Rata-rata (40x Iterasi)		Score
	Minimax	Minimax+Alpha-beta pruning	
3x3	03.24 s	03.00 s	Draw

IV. KESIMPULAN

Dalam percobaan kasus di atas, penggunaan algoritma *minimax* + *alpha-beta pruning* menghasilkan solusi optimal dalam hal waktu. Penggunaan algoritma *minimax* dengan atau tanpa *alpha-beta pruning* akan menghasilkan solusi optimal untuk menghitung pergerakan pada AI player. Jika ingin melakukan pencarian secara cepat dan untuk memperluas skala bidak permainan dapat digunakan optimasi *minimax* dan *alpha-beta pruning* dengan memangkas setiap nodes yang tidak diperlukan.

Penggunaan *Alpha-Beta Pruning* dalam algoritma *minimax* bisa menjadi solusi optimal untuk pencarian secara efisien namun belum tentu efektif. Penggunaan *Alpha-Beta Pruning* dalam algoritma *minimax* pada permainan pencarian adalah tentang seberapa cepat dalam pengambilan keputusan. Optimasi *Alpha-Beta Pruning* bukan tentang memperkirakan langkah terbaik dalam pencarian keputusan melainkan

memotong pohon carian agar membuatnya menjadi efisien dalam hal waktu.

Studi penelitian dimasa mendatang, diharapkan algoritma *minimax* dengan *alpha-beta pruning* disimulasikan di permainan Tic-Tac-Toe dengan ukuran papan yang lebih luas dengan kombinasi algoritma *greedy*.

ACKNOWLEDGMENT

Ucapan terimakasih penulis sampaikan kepada Bapak Ali Akbar S selaku Dosen matakuliah Kecerdasan Buatan atas bimbingan dan dorongannya sehingga penulis dapat menyelesaikan tugas makalah ini. Penulis dapat belajar banyak tentang penerapan algoritma untuk kecerdasan khususnya pada Permainan Tic-Tac-Toe.

REFERENCES

- [1] R. Garg, D.P. Nayak, "Game Of Tic-Tac-Toe: Simulation Using Min-Max Algorithm," UIET. Panjab University, vol. 8, No.7, July 2017.
- [2] E. Mel'ko and B. Nagy, "Optimal Strategy in Games with Chance nodes," Acta Cybern.,18(2),pp. 171-192,2007.
- [3] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," Prentice-Hall, 1995. (3rd edition, 2010), pp 161-169.
- [4] Admin, "Minimax Algorithm in Game Theory | Set 4 (Alpha-Beta Pruning)," [Online]. Available: <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/>
- [5] C.Lazo , "Tic-Tac-Toe," 15 Maret 2017 . [Online]. Available : <https://github.com/LazoCoder/Tic-Tac-Toe> [Diakses: 8 Februari 2018].
- [6] Memoiry , "Ticky (A Tic-tac toe game)," 31 Maret 2017 . [Online]. Available : <https://github.com/memoiry/Ticky> [Diakses: 8 Februari 2018].