ISBN : 978-623-7297-39-0

ANALISIS KECERDASAN BUATAN PADA PERMAINAN CHECKER MENGGUNAKAN OPTIMASI ALGORITMA MINIMAX

Iliyen Uari[1)], Abdullah Muhazir[2)], Hermansyah Alam[3)], Budhi Santri Kusuma[4)]
[1)]Alumni Teknik Informatika Institut Teknologi Medan (ITM),
[2)]Dosen Teknik Informatika Institut Teknologi Medan (ITM)
[3)]Dosen Teknik Informatika Institut Teknologi Medan (ITM)
[4)]Dosen Universitas Medan Area (UMA)

Abstrak

Permainan (game) sering memanfaatkan kecerdasan buatan (artificial intelligence) untuk memberikan tantangan berupa kesempatan untuk melawanmesin 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 aturanpermainan yang digunakan, khususny untuk permainan yang melibatkan dua pemain seperti permainan checker. Dalam penelitian ini, algoritma minimax akan diimplementasikan ke dalam permainan checkers, dalam bentuk sebuah computer AI yang dapat mengambil keputusan langkah secara otomatis pada saat permainanberlangsung. Hasil implementasi algoritma minimax kedalam permainan checker yang diteliti pada penelitian ini mampu menunjukkan bagaimana proses berfikir (self thinking) dari computer AI berdasarkan konfigurasi tingkat kesulitan yang dipilih oleh pengguna.

Kata-Kata Kunci : *: Kecerdasan Buatan, Minimax, Games, Checker*

I. PENDAHULUAN

Permainan (*game*) sering memanfaatkan kecerdasan buatan (*artificial intelligence*) untuk memberikan tantangan di dalam permainan tersebut. Tantangan ini dapat berupa kesempatan untuk melawan mesin yang memiliki tingkat intelektualitas tertentu dalam berpikir, sehingga pengguna tidak harus mencari lawan tanding yang lain jika ingin bermain. Kemampuan berpikir yang diterapkan di dalam suatu permainan didukung oleh algoritma yang tentunya harus memiliki efisiensi dan efektifitas sesuai dengan peraturan permainan yang digunakan. Salah satu algoritmayang dapat diterapkan di dalam permainan adalah algoritma *minimax*.

Minimax dapat melakukan kalkulasi langkah dengan pertimbangan yang logis, berdasarkan aturan permainan yang digunakan. Secara khusus, algoritma ini sering digunakan untuk permainan yang melibatkan dua pemain, dimana untuk setiap langkah (*turn*), akan dilakukan kalkulasi untuk mencari langkah yang paling optimal dengan mempertimbang kan kemungkinan langkah yang akan dilakukan oleh lawan. Dalam algoritma *minimax*, seluruh kemungkinan langkah akan dievaluasi dalam sebuah pohon permaianan, dengan menggunakan fungsi heuristik untuk menentukan simpul *max* dan simpul *min* yang digunakan sebagai dasar untuk menentukan langkah di dalam sebuah permainan. Simpul *max* dan simpul *min* di dalam pohon kemungkinan ini digunakan untuk melihat tingkat keuntungan atau kerugian dari setiap kemungkinan langkah, sehingga dapat diambil satu langkah yang paling efisien dan efektif untuk mengalahkan lawan. Dengan melihat deskripsi dari algoritma ini, *minimax* sangat cocok untuk diterapkan ke dalam permainan seperti *othello, checkers, bridge, tic-tac-toe* atau *backgammon* yang hanya melibatkan dua pemain, dimana setiap pemain akan berusaha untuk mengalahkan pemain lawan dengan mempertimbangkan langkah yang paling efisien dan paling efektif. Dalam penelitian ini, algoritma *minimax* akan diimplementasikan ke dalam permainan *checkers*, dimana pengguna dapat memilih satu opsi untuk melawan komputer yang telah diberikan kemampuan berpikir berdasarkan cara kerja algoritma *minimax*. Aturan permainan *checkers* yang digunakan adalah aturan umum, dimana setiap pemain hanya dapat menjalankan koin yang dimilikinya secara diagonal dengan jumlah maximal langkah sebanyak dua kotak. Setiap koin yang berhasil mencapai batas akhir dari kotak lawan, maka akan diberikan promosi, dimana kointersebut dapat bergerak maju atau mundur. Permainan dinyatakanberakhir jika salah satu pemain berhasil menghabisi seluruh koin lawan atau berhasil seluruh koin yangdimilikinya berhasil mencapai batas akhir dari kotak lawan.

II. METODE PENELITIAN

Dalam melakukan penelitian ini, metode pengumpulan data yang dilakukan adalah sebagai berikut: Penelitian Kepustakaan. Data yang dikumpulkan dari hasil penelitian kepustakaan adalah berupa teori mengenai aturan permainan *checker*, kecerdasan buatan (*artificial intelligence*). Aplikasi permainan *checker* yang menggunakan optimasi algoritma *minimax. Dapat dilihat sebagai berikut:*

2.1 Rancangan Proses

Berdasarkan hasil analisa sistem yang dilakukan, dirancang sebuah alur proses yang akan digunakan dalam perancangan aplikasi. Adapun bentuk rancangan proses algoritma *minimax* yang diimplementasikan ke dalam aplikasi permainan *checker* ini sebagaimana terlihat pada *flowchart.*

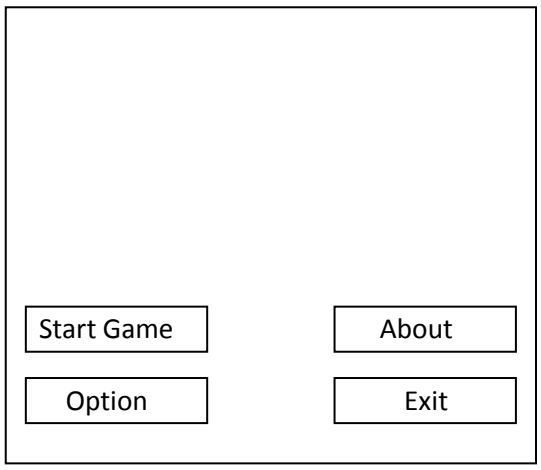
ISBN : 978-623-7297-39-0

2.2 Rancangan Antarmuka

Dalam rancangan antarmuka ini, penulis merancang beberapa *form* yang akan digunakan sebagai media interaksi antara sistem dengan pengguna. Ada empat *form* yang dirancang pada aplikasi permainan *checker* dengan optimasi algoritma *minimax* ini, yaitu *form* Main, *form* Game, *form* Option dan *form* About.

1. Rancangan *Form* Main

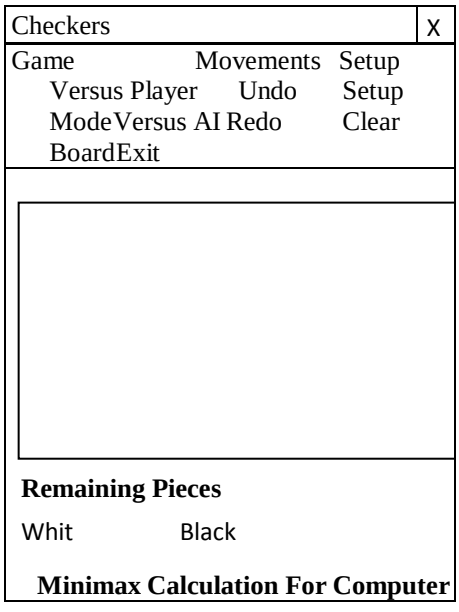
Form Main dirancang sebagai*form* induk dari aplikasi ini, *form* ini dirancang untuk tampil pertama kali dan sebagai media untuk memilih opsi menu seperti memulai permainan, memilih opsi tingkat kesulitan AI serta tampilan info permainan. Adapun bentuk rancangan*form* Main berikut:

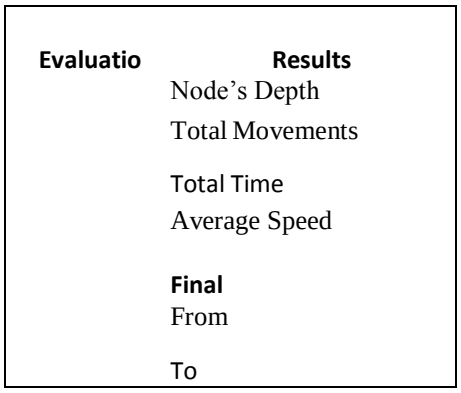


Gambar 1. Rancangan *Form* Main

1. Rancangan *Form* Game

Form Game dirancang sebagai *form* yang menampilkan permainan *checker*. *Form* ini juga berfungsi untuk menampilkankalkulasi langkah yang diambil oleh *computer* AI dengan menggunakan optimasi algoritma *minimax*. Adapun bentuk rancangan *form* Gamesebagaimana terlihat dibawah:





Gambar 2. Rancangan *Form* Game

1. Rancangan *Form* Option

Form Option dirancang untuk menampilkan konfigurasi *computer* AI yang digunakan di dalam penentuan langkah pada permainan *checker*. Pada *form* ini terdapat beberapa pilihan yang dapat diubah oleh pengguna seperti batas waktu pencarian alternatif langkah, jumlah maksimum alternatif langkah yang ditelusuri, jumlah maksimum kedalaman *node* yang diperiksa, serta jumlah maksimum nilai probabilitas (nilai MAX atau MIN) yang digunakan di dalam pencarian langkah yang paling optimum. Adapun bentuk rancangan *form* Option sebagaimana terlihat pada

2. Rancangan *Form* About

Form About dirancang untuk menampilkan informasi mengenai permainan *checker* dengan menggunakan optimasi algoritma *minimax* ini. Adapun bentuk rancangan *form* About sebagaimana terlihat pada gambar berikut:

Gambar 3. Tampilan *Form* Main

Selanjutnya, dilakukanpemilihan menu untuk masing- masing pilihan yang ada seperti Start Game, Option dan About. Untuk menu Start Game, pada saat dilakukan penekanan tombol Start Game pada

ISBN : 978-623-7297-39-0

form Main, hasil yang diperoleh adalah munculnya *form*Game sebagaimana terlihat pada gambar berikut:

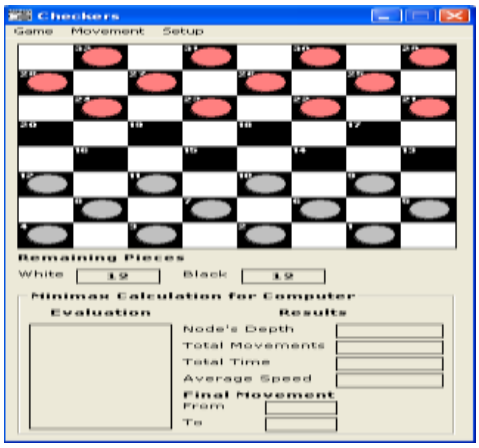
Gambar 4. Tampilan *Form* Option

Tahap akhir dari pengujian iniadalah melakukan penekanan terhadap tombol About. Hasil yang diperoleh adalah munculnya *form*About sebagaimana terlihat pada gambar di bawah:

Gambar 5. Tampilan *Form* About

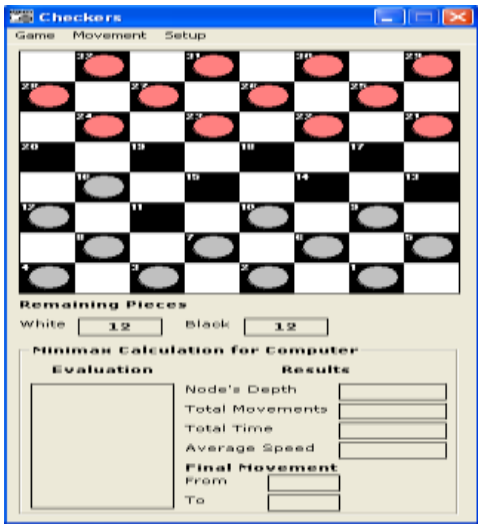
Pengujian Permainan *Checker*

Dalam pengujian ini, akandiuji apakah aplikasi permainan telahdapat melakukan *self thinking* untuk mencari langkah optimum pada permainan *checker*. Langkah pertama dalam pengujian ini adalah mengakses *form* Game dan memilih sub menu Versus AI dari menu Game, sebagaimana terlihat pada Gambar dibawah:



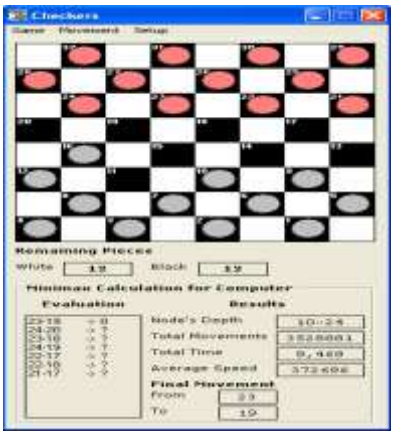
Gambar 6. Memilih Sub Menu Versus AI

Selanjutnya, dilakukanpemindahan koin hitam (*black*) dari posisi 11 menuju posisi 16, sebagaimana terlihat pada gambar berikut:



Gambar 7. Memindahkan Koin Hitam (*Black*)

Setelah langkah pemindahandilakukan, terlihat bahwa *computer* AI melakukan langkah antisipasi dengan menggerakan koin putih (*white*) dari posisi 23 menuju posisi 19, sebagaimana terlihat pada Gambar 8.



Gambar 8. Pergerakan *Computer* AI

ISBN : 978-623-7297-39-0

Gambar 9. Memilih Sub Menu Undo

Dari Gambar 8, terlihat bahwa aplikasi permainan telah mampu untuk menunjukkan hasil kalkulasi nilai MIN atau MAX pada kotak informasi Evaluation. Dari beberapa alternatif kemungkinan langkah yang dapat diambil oleh *computer* AI, terlihat bahwamemindahkan posisi koin putih dari posisi 23 menuju posisi 19 merupakan nilai yang paling optimum dengan nilai 0.

Selain itu, sistem juga dapat menunjukkan berapa tingkatkedalaman *node* yang diperiksa, jumlah total langkah yang diperiksa serta waktu yang dibutuhkan untuk menghasilkan nilai akhirnya.

Pengujian berikutnya adalah menguji apakah aplikasi permainantelah dapat melakukan proses *undo* dan *redo*. Proses *undo* adalah opsi dimana pemain dapat membatalkan langkah terakhir yang diambil oleh *computer* AI maupun langkah dari pemain itu sendiri. Pada pengujianproses *undo* ini, dilakukan pemilihan sub menu Undo dari menu Movements, sebagaimana terlihat pada gambar.

Hasil Proses Undo Langkah yang sama dilakukan untuk proses *redo*, namum yang dipilih adalah sub menu Redo dari menu Movement. Hasil yang diperoleh dari proses pengujian iniadalah posisi koin putih dari posisi 23kembali dipindahkan ke posisi 19.

Berdasarkan hasil pengujian aplikasi yang dilakukan, diperoleh hasil pembahasan mengenai kinerja algoritma *minimax* pada aplikasi permainan *checker* yang dihasilkan. Adapun hasil pembahasan yang penulis peroleh adalah sebagai berikut :

1. Dalam permainan *checker*, posisi koin pada papan sangat mempengaruhi pengambilan keputusan langkah yang akan diambil pada algoritma *minimax*. Semakin dekat posisi koin pada pinggirpapan, akan semakin tinggi nilai dari kemungkinan langkah tersebut, yangselanjutnya akan berpengaruhpada nilai MAX pada algoritma *minimax*.Sebaliknya, semakin jauh posisi koinj dari pinggi papan, maka nilai yang dipengaruhinya adalah nilai MIN pada algoritma *minimax*.
2. Kedalaman *node* yangdiperiksa oleh algoritma *minimax* mempengaruhitingkat akurasi pengambilan keputusan langkah. Semakin dalam *node* yang diperiksa, setiap kemungkinan langkah akan dikalkulasi dan dapat diambil nilai yang lebih akurat.
3. Aplikasi yang dihasilkan telah mampu untuk menunjukkan bagaimana optimasi algoritma *minimax* mempengaruhi kinerja sebuah aplikasi permainan *checker*. Dengan menggunakan algoritma*minimax* ini, permainan *checker* dapat dimainkan tanpa harus tersedianya dua orang pemain. Dengan hanya satu orang pemain saja, permainan tetap dapat berlangsung, dengan *computer* AI menggantikan pemain yang lainnya.

III. KESIMPULAN DAN SARAN

3.1 Kesimpulan

1. Algoritma *minimax* yangdiaplikasikan ke dalam permainan *checker* ini menyebabkan permainan dapat dilakukan dengan hanyasatu pemain saja. Dengan *computer* AI yang menggunakan algoritma *minimax* untuk mementukan langkah koin, pemain yang lain dapat digantikan.
2. Hasil akurasi penentuan langkah pada permainan *checker* dengan optimasialgoritma *minimax* ini dipengaruhi oleh beberapa hal, seperti kedalaman *node*yang diperiksa, posisi koin pada papan permainan, jumlahlangkah yang diperiksa serta waktu berfikir yang diberikan.Keseluruhan parameter ini akan mempengaruhi nilai MAX atau MIN dari *leaf node*akhir yang dihasilkan, yang selanjutnya akan mempengaruhi nilai akhir darilangkah tersebut.

3.2 Saran

1. Aplikasi ini dapatdikembangkan sehingga dapat dimainkan pada jaringankomputer. Hal ini akan menambah daya tarik dari aplikasi permainan *checker*tersebut.
2. Dapat digunakan metode *artificial intelligence* yang lain diluar algoritma *minimax*, sehingga dapat dibandingkan hasil keputusan langkah yang dihasilkan oleh kedua metode tersebut.
3. Dapat ditambahkan metode lain ke dalam algoritma *minimax* yang digunakan, seperti mengganti metode *deep first search* yang digunakan, dan dibandingkan hasilnya dengan metode tersebut.

DAFTAR PUSTAKA

[1]. Akbar, M.A, 2011, *Analisis Dan Implementasi Kecerdasan Buatan Pada Permainan Checker MenggunakanAlgoritma Minimax Dengan Negascout*, Tugas Akhir, Universitas Sumatera Utara.

[2]. Ayuni, Wulan, 2011, *Implementasi Konsep Kecerdasan BuatanDalam Rancang Bangun Game Brick Breaker*, Tugas Akhir, Universitas Sumatera Utara.

[3]. Beny, 2010, *PengembanganSoftware Game MenggunakanRPG Maker VX*, Tugas Akhir, Universitas Sumatera Utara.

[4]. Nuzulla, B. & Solichin, A., 2012, *Implementasi AlgoritmaSteepest Ascent Hill Climbing Dengan Optimasi Minimax Pada Permainan Tic Tac Toe Berbasis Android*, Prosiding Seminar Nasional Multi Disiplin Ilmu, 65-73.

[5]. Prasetyo, R.M.K. 2012, *Penyelesaian Permainan Checkers Pada Mobile Device Berbasis Android Menggunakan Algoritma Iterative Deepening Search*, Tugas Akhir, UniversitasSumatera Utara.
[6]. Ramadhan, A., 2007, *Visual Basic6.0*, PT. Elex MediaKomputindo, Jakarta.
[7]. Rizky, S., 2011, *Konsep Dasar Rekayasa Perangkat Lunak*, Prestasi Pustaka Publisher, Jakarta.
[8]. Setiadi, D. R. I. M., 2012, *Implementasi Algoritma Minimax Untuk Artificial Intelligence Pada Permainan Catur Sederhana*, Techno.COM, **Vol. 11 No.2**, 99-107.
[9]. Simamarmata, J., 2010, *Rekayasa Perangkat Lunak*, Penerbit Andi, Yogyakarta.
[10]. Sutojo, T., Mulyanto, E., Suhartono, V. 2011, *Kecerdasan Buatan*, Penerbit Andi, Yogyakarta.
[11]. Toba, H. dan Kurniawan, B., 2008, *Aplikasi Permainan Capsah Banting dengan Penerapan Algoritma Minimax*, Jurnal Informatika, **Vol. 4 No. 1**, 12-27
[12]. Zarlis, M. & Handrizal, 2008, Algoritma dan Pemrograman : Teori dan Praktik Dalam Pascal, Edisi Kedua,USU Press, Medan.