Jurnal Riset Komputer (JURIKOM), Vol. 2 No. 6, Desember 2015
ISSN 2407-389X (Media Cetak)
Hal : 28-32

PENERAPAN ALGORITMA MINIMAX PADA PERMAINAN CHECKERS

Dahwila Syapnika [1], Edward Robinson Siagian [2]

[1] Mahasiswa Teknik Informatika STMIK Budi Darma
[2] Dosen Tetap STMIK Budi Darma
[1, 2] Jl. Sisimangaraja Np. 338 Simpang Limun Meda

ABSTRAK

Checkers merupakan jenis permainan game board,yang mengandalkan strategi sebagai elemen utamanya. Permainan ini dimainkan oleh dua orang pemain dengan tujuan menghabiskan kepingan lawan. Permainan checkers yang dibuat dengan AI (Artificial Intelligence) tertentu menerapkan algoritma Minimax. Penerapan algoritma Minimax dalam checkers dibuat berdasarkan prosedur Minimax untuk mendapatkan langkah terbaik dari posisi yang ada. Setiap posisi memiliki nilai yang dapat dihasilkan dari langkah terbaik, dengan berasumsi bahwa AI akan selalu mencoba memaksimalkan nilai, ketika lawan akan mencoba untuk meminimalkannya. Perancangan permainan checkers ini di buat dengan menggunakan Microsoft Visual Basic 6.0. Didalam menu permainan checkers terdapat beberapa menu. Dalam permainan checkers terdapat dua buah bidak yang berlainan warna. Setiap pemain memilih warna yang di inginkan. Setiap bidak dapat berubah menjadi raja setelah melewati garis akhir lawan.

Kata Kunci: checkers, Artificial Intellingence (AI), Minimax, alpha-beta, Microsoft Visual Basic

I. PENDAHULUAN

Bermain *game* merupakan salah satu aktifitas yang sangat disukai oleh sebagian besar masyarakat di dunia ini. Alasan mereka bermain game tentunya berbeda-beda, ada yang untuk melepas lelah, ada juga yang memang suka atau hobi bermain *game*. Dengan berkembangnya teknologi sekarang ini, *game – game* ini tidak hanya dapat kita jumpai pada kehidupan nyata, tapi juga dapat kita jumpai di dalam dunia maya. Jenisnya pun semakin banyak dan bervariasi. Salah satu yang cukup menarik perhatian adalah permainan komputer.

Permainan- permainan berbasis komputer ini juga bermacam-macam. Salah satu kelebihannya adalah kita tidak harus mencari orang untuk menjadi lawan tanding jika ingin bermain karena permainan berbasis komputer ini sudah mendukung *single-player mode* dimana kita dapat bermain sendiri melawan komputer yang dirancang untuk dapat berlaku seperti pemain manusia atau yang sering dikenal dengan *Artificial Inteligince* (AI). Contoh – contoh permainan yang menggunakan AI adalah permainan catur, *go, othello, checkers, bridge, tic-tac-toe* dan lain sebagainya. Untuk membuat pemain merasa seperti melawan pemain manusia lainnya, diperlukan suatu algoritma yang dapat membuat AI ini mampu mengambil keputusan yang terbaik agar dapat mengalahkan pemain atau setidaknya menghalau pemain menang. Algoritma minimax ini merupakan algoritma yang sangat sering dipakai untuk permasalah tersebut dalam permainan *checkers.*

Checkers ialah suatu permainan papan dengan menggunakan keterampilan murni dari dua pemain yang mengikuti sejumlah aturan-aturan dalam permainan, dan berusaha untuk memenangkan permainan dengan cara memakan semua bidak lawan dalam papan atau dengan membuat semua bidak lawan tidak dapat melakukan gerakan.

II. TEORITIS

A. Aturan Permainan Checkers

Menurut jimloy *Checkers* ialah suatu permainan papan dengan menggunakan keterampilan murni dari dua pemain yang mengikuti sejumlah aturan-aturan dalam permainan, dan berusaha untuk memenangkan permainan dengan cara memakan semua bidak lawan dalam papan atau dengan membuat semua bidak lawan tidak dapat melakukan gerakan.

B. Algoritma

Dalam matematika dan komputasi, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, dalam hal ini berbeda dengan heuristik. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai. Desain dan analisis algoritma adalah suatu cabang khusus dalam ilmu komputer yang mempelajari karakteristik dan performa dari suatu algoritma dalam menyelesaikan masalah, terlepas dari implementasi algoritma tersebut. Dalam cabang disiplin ini algoritma dipelajari secara abstrak, terlepas dari sistem komputer atau bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan kriteria yang sama.Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Secara informal, algoritma yang dapat menyelesaikan

suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan masalahnya mempunyai kompleksitas yang tinggi.

C. Algoritma Minimax

Minimax merupakan algoritma yang digunakan untuk menentukan pilihan agar memperkecil kemungkinan kehilangan nilai maksimal. Algoritma ini dapat diterapkan dengan baik pada permainan yang melibatkan dua pemain yang saling bergantian seperti *tic-tac-toe, othello, Checker, catur, go*, dan permainan yang menggunakan strategi atau logika lainnya (Nadhira Ayuningtyas, 2008).

Algoritma Minimax merupakan salah satu implementasi dari pencarian DFS (*Depth-First Search*) dalam melakukan pencarian pada pohon. DFS akan menelusuri simpul paling dalam terlebih dahulu. Setelah simpul akar dibangkitkan, *algoritma* ini akan membangkitkan simpul pada tingkat kedua, yang akan dilanjutkan pada tingkat ketiga, dst.

D. Alpha-Beta Pruning

Dalam algoritma *Minimax*, pencarian dilakukan pada seluruh bagian pohon, sementara sebagian pohon tidak seharusnya diperiksa. *Alpha-beta pruning* merupakan modifikasi dari algoritma *Minimax*, yang akan mengurangi jumlah node yang dievaluasi oleh pohon pencarian. Pencarian untuk node berikutnya akan dipikirkan terlebih dahulu. Algoritma ini akan berhenti mengevaluasi langkah ketika terdapat paling tidak satu kemungkinan yang ditemukan dan membuktikan bahwa langkah tersebut lebih buruk jika dibandingkan dengan langkah yang diperiksa sebelumnya. Sehingga, langkah berikutnya tidak perlu dievaluasi lebih jauh. Dengan algoritma ini hasil optimasi dari suatu algoritma tidak akan berubah. Berikut merupakan pohon dengan algoritma *alpha-beta pruning*

E. Penerapan Algoritma Minimax Pada Permainan Chekers

Penerapan algoritma *Minimax* dalam *checkers* dibuat berdasarkan prosedur *Minimax* untuk mendapatkan langkah terbaik dari posisi yang ada. Setiap posisi memiliki nilai yang dapat dihasilkan dari langkah terbaik, dengan berasumsi bahwa AI akan selalu mencoba memaksimalkan nilai, ketika lawan akan mencoba untuk meminimalkannya.Ketika prosedur *minimax* mencapai akar pada pohon pencarian (posisi saat tersebut), akan menghasilkan langkah terbaik dengan asumsi lawan akan menggunakan kriteria evaluasi yang sama. Beberapa versi program yang dibuat kebanyakan telah menerapkan algoritma pemotongan *alpha-beta*.Terdapat dua macam metode, yang disebut *rote learning*. Metode tersebut memiliki penyimpan untuk setiap posisi yang ditemui selama permainan dengan tidak menghilangkan nilai yang ditentukan oleh prosedur *Minimax*. Hasilnya adalah jika terdapat posisi yang pernah ditenukan sebelumnya, akan dimunculkan sebagai posisi terminal pada pohon pencarian. Sehingga, pencarian akan semakin mudah karena nilai posisi diambil dari hasil pencarian yang telah dilakukan sebelumnya. Satu masalah awal yang ditemukan adalah program tidak mendukung untuk melangkah langsung menuju kemenangan. Solusi pencegahan adalah dengan mengurangi sedikit nilai posisi setiap tahap (disebut *ply*) pada analisis *Minimax*. Jika program berhadapan dengan pilihan posisi dengan nilai yang hanya dibedakan oleh *ply*, maka program akan secara otomatis melangkah pada pilihan yang paling menguntungkan..

F. Microsoft Visual Basic

Merupakan sebuah bahasa pemrograman yang bersifat *event driven* dan menawarkan *Integrated Development Environment* (IDE) *visual* untuk membuat program aplikasi berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman *Common Object Model* (COM). *Visual Basic* merupakan turunan bahasa *BASIC* dan menawarkan pengembangan aplikasi komputer berbasis grafik dengan cepat, akses ke basis data menggunakan *Data Access Objects* (DAO), *Remote Data Objects* (RDO), atau *ActiveX Data Object* (ADO), serta menawarkan pembuatan kontrol *ActiveX* dan objek *ActiveX*. Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda. Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic* Program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi *eksternal* tambahan.IDE (*Integrated Development Environment*) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan

III. ANALISA dan PEMODELAN

A. Analisa

Kecerdasan buatan merupakan salah satu bidang ilmu yang penting pada ilmu komputer. Banyak permainan komputer yang memanfaatkan kecerdasan buatan untuk membuat permainan tersebut bertindak cerdas dengan memilih langkah terbaik pada permainan. Tetapi banyak orang yang memainkan permainan tersebut tidak mengerti cara kerja dari kecerdasan buatan itu sendiri.

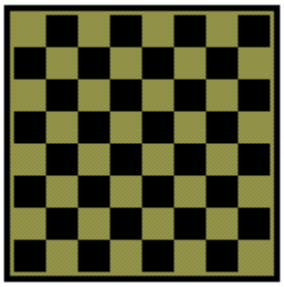
Dahulu *game* checkers ini dimainkan dengan menggunakan papan dengan ukuran 8x8, karena perkembangan terus meningkat terutama di dunia teknologi maka permainan checkers ini dapat dimainkan dengan menggunakan komputer. Dengan dimainkanya permaianan checkers di computer berarti memberikn kemudahan untuk

memainnkannya di bandingkan dengan menggunakan papan tersebut.

B. Storyboard

Checkers ialah suatu permainan papan dengan menggunakan keterampilan murni dari dua pemain yang mengikuti sejumlah aturan-aturan dalam permainan, dan berusaha untuk memenangkan permainan dengan cara memakan semua bidak lawan dalam papan atau dengan membuat semua bidak lawan tidak dapat melakukan gerakan.

Papan *checkers* tersusun oleh 64 persegi bergantian antara sisi gelap dan terang, yang disusun ke dalam satu kumpulan persegi yang terdiri dari 8 baris dan 8 kolom.



Gambar 1. Papan Checkers

C. Creative Strategy

Didalam permaian checkers perlu dilakukannya strategi untuk memenangkan permainan tersebut, karena permainan tersebut sangatlah bebrpengaruh kepada strategi yang kita lakukan.strateginya yaitu

1. Dengan mengubah bidak sebanyak-banyaknya menjadi raja, karena dalam menjadi raja bidak tersebut memiliki peluang terbesar untuk mengalahkan.
2. Dengan berjalan bidak di paling pinggir maka, bidak tersebut tidak akan dapat di makan oleh bidak lawan, karena semakin banyak bidak maka kesempatan menang lebih besa

IV. PERANCANGAN DAN PENGUJIAN

Tahapan pembentukan data awal dilakukan sebelum pembangunan aplikasi oleh *programmer*, rincian dari data awal sebelumnya telah direncanakan dalam tahap perancangan sistem. Tahapan ini membutuhkan waktu satu hari dalam pengerjaannya.

Perangkat lunak yang digunakan untuk mengimplementasikan sistem yang baru dibangun terdiri dari dua jenis yaitu perangkat lunak sistem operasi dan perangkat lunak pendukung. Perangkat lunak minimum yang dibutuhkan yaitu:

Tabel 1. Persiapan Perangkat Lunak

Jenis Perangkat Lunak	Komputer
Aplikasi Sistem Operasi	Windows XP SP2
Aplikasi Pendukung	Visual Basic 6

Agar aplikasi yang dibangun dapat dijalankan maka komputer yang akan menggunakan aplikasi ini harus sudah meng*instal* aplikasi system operasi. Untuk menginstal aplikasi system operasi, dapat dilakukan dengan melakukan instalasi aplikasi system operasi yang telah ada dalam paket instalasi aplikasi yang dibangun. Tahapan persiapan perangkat lunak dapat dilakukan dalam waktu satu hari dan cukup dilakukan oleh satu orang personil.

Pengujian pada sistem yang baru dibangun dilakukan melalui dua tahap yaitu pengujian *aplha* (fungsional) dan pengujian *betha*. Metode yang digunakan dalam pengujian *aplha* adalah metode *blackbox* yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun, sedangkan metode untuk pengujian *betha* adalah dengan memberikan kuesioner kepada pengguna yang menggunakan aplikasi yang dibangun. Tahap pengujian sistem dilakukan dalam waktu 15 hari dengan jumlah personil satu orang yang bertindak sebagai *programmer* dan sebagai analis.

Gambar 2. Desain Permainan

A. Pengujian *Aplha*

Pengujian *aplha* adalah pengujian yang fokus pada persyaratan fungsionaldari perangkat lunak yang dibangun.

Pengujian sistem sesuai dengan rencana yang telah ditentukan menghasilkan data sebagai berikut:

Tabel 2. Pengujian Permainan

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	pengamatan	Kesimpulan
Klik tombol mulai	Untuk memulai aplikasi permainan	Permainan dpt dimulai	Diterima
Klik tombol tipe permainan	Untuk memilih jenis pemain pada permainan	Tombol dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol nama pemain	Untuk Mengisi data nama pemain	Tombol dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol balik papan	Untuk membalik posisi papan permainan	Tombol dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol kembali	Untuk kembali pada saat melangkah di permainan	Tombol dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol keluar	Untuk keluar dari aplikasi	Tombol keluar dapat berfungsi sesuai yang diharapkan	Diterima

Hasil pengujian dari pengujian *Aplha* yang telah dilakukan, menunjukan bahwa aplikasi yang dibangun sudah memenuhi persyaratan fungsional.

Akan tetapi pada prosesnya masih memungkinkan untuk terjadinya kesalahan, namun frekuensi kesalahan masih relatif kecil, dikarenakan dalam perancangan telah dirancang sistem untuk menangani kemungkinan kesalahan proses yang terjadi. Secara fungsional sistem yang telah dibangun sudah dapat menghasilkan keluaran yang diharapkan.

B. Pengujian *Betha*

Pengujian *betha* merupakan pengujian yang dilakukan secara objektif, pengujian ini dilakukan oleh *user* yang akan menggunakan aplikasi yang dibangun. Pengujian dilakukan terhadap 10 orang yang akan berhubungan dengan aplikasi yang dibangun. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulannya terhadap penilaian penerapan sistem yang baru. Kuesioner terdiri dari 5 pertanyaan (contoh kuesioner terlampir) dengan menggunakan skala jawaban 1 sampai 7, dengan ketentuan skala untuk setiap pertanyaan sebagai berikut:

1. Bagaimana tingkat kesulitan pada saat menjalankan permainan i

No	Keterangan
1	Sangat mudah
2	Mudah
3	Cukup mudah
4	Biasa-biasa saja
5	Agak sulit
6	Sulit
7	Sangat sulit

2. Bagaimana tampilan dari perangkat lunak yang dibangun:

No	Keterangan
1	Sangat bagus
2	Bagus
3	Cukup bagus
4	Biasa-biasa saja
5	Kurang bagus
6	Tidak bagus
7	Sangat tidak bagus

3. Bagaimana waktu yang diperlukan computer untuk berfikir/melangkah pada saat kita bermain melawan computer di permainan :

No	Keterangan
1	Sangat cepat
2	Cepat
3	Cukup cepat
4	Biasa-biasa saja
5	Agak lambat
6	Lambat
7	Sangat lambat

4. Bagaimana informasi yang ditampilakn oleh perangkat lunak yang dibangun:

No	Keterangan
1	Sangat akurat
2	Akurat
3	Cukup akurat
4	Biasa-biasa saja
5	Kurang akurat
6	Tidak akurat
7	Sangat tidak akurat

Berdasarkan hasil pengujian *Betha*, dicari prosentase masing-masing jawaban dengan menggunakan rumus:

Y=P/Q*100%

Keterangan:

P = banyaknya jawaban responden tiap soal

Q = jumlah responden

Y = nilai prosentase

Hasil perhitungan prosentase masing-masing jawaban sebagai berikut:

Untuk Pertanyaan No 1

"Bagaimana proses pengolahan data menggunakan perangkat lunak yang dibangun"

Tabel 4.3 Hasil Pengujian *Betha* Pertanyaan No 1

Kategori Jawaban	Frekuensi Jawaban	Jumlah populasi Sampel	Jumlah Persentase
Sangat mudah	5	10	50 %
Mudah	4	10	40 %
Cukup mudah	1	10	10 %
Biasa-biasa saja	0	10	20 %
Agak sulit	0	10	0 %
Sulit	0	10	0 %
Sangat sulit	0	10	0 %

Untuk Pertanyaan No 2

"Bagaimana tampilan dari perangkat lunak yang dibangun"

Tabel 4.4 Hasil Pengujian *Betha* Pertanyaan No 2

Kategori Jawaban	Frekuensi Jawaban	Jumlah Populasi Sampel	Jumlah Persentase
Sangat bagus	0	10	0 %
Bagus	6	10	60 %
Cukup bagus	3	10	30 %
Biasa-biasa saja	1	10	10 %
Kurang bagus	0	10	0 %
Tidak bagus	0	10	0 %
Sangat tidak bagus	0	10	0 %
Sangat bagus	0	10	0 %
Bagus	6	10	60 %
Cukup bagus	3	10	30 %
Biasa-biasa saja	1	10	10 %
Kurang bagus	0	10	0 %
Tidak bagus	0	10	0 %
Sangat tidak bagus	0	10	0 %

Tabel 4.6 Hasil Pengujian *Betha* Pertanyaan No 4

Kategori Jawaban	Frekuensi Jawaban	Jumlah Populasi Sampel	Jumlah Persentase
Sangat akurat	1	10	10 %
Akurat	8	10	80 %
Cukup akurat	1	10	10 %
Biasa-biasa saja	0	10	0 %
Kurang akurat	0	10	0 %
Tidak akurat	0	10	0 %
Sangat tidak akurat	0	10	0 %

Berdasarkan hasil pengujian *betha* dapat disimpulkan bahwa Aplikasi permainan ini cukup mudah untuk dimainkan, memiliki tampilan yang bagus, permainan yg cukup cepat dan informasi yang dihasilan lebih akurat.

V. KESIMPULAN

Berdasarkan hasil pengujian, maka dapat diambil beberapa kesimpulan sebagai berikut:

a. Permainan *checkers* merupakan permainan yang dimainkan oleh dua orang dengan tujuan untuk menghabiskan kepingan yang dimiliki lawan.
b. Dalam pembuatannya dengan AI, permainan ini menerapkan algoritma Minimax. Algoritma Minimax memiliki dasar berupa *zero-sum game*, dimana jika pemain mendapatkan nilai tertentu maka pemain lain akan kehilangan nilai yang sama dengan pemain tersebut.
c. Dengan menambahkan nilai *alpha-beta* algoritma *Minimax* akan memiliki pohon pencarian yang lebih singkat sehingga akan membutuhkan waktu singkat untuk melakukan aksinya.

VI. DAFTAR PUSTAKA

1. Ayuningtyas, Nadhira. 2008. "Algoritma Minimax dalam Permainan Checker". Dalam *Strategi Algoritmik*. Bandung, Indonesia: Institut Teknologi Bandung.
2. Coppin, Ben. 2004. *Artificial Intelligence Illuminated*. United States of America: Jones and Bartlett.
3. Millington, Ian. 2006. *Artificial Intelligence For Games*. United States of America: Morgan Kaufmann.
4. Stuart Russel and Peter Norvig. 2003. *Artificial Inteligence A Modern Approach.* Second Edition. United States of America: Prentice Hall.
5. Tim Jones, M. 2008. *Artificial Intelligence A Systems Approach*. Hingham Massachusetts: David Pallai.
6. http://www.jimloy.com/checkers/checkers.html,tanggal akses 20/052011