

**IMPLEMENTASI ALGORITMA *NEGASCOUT* UNTUK
PERMAINAN *CHECKERS***

Tugas Akhir



Oleh

Aditya Kurniawan Effendi

22074279

Program Studi Teknik Informatika Fakultas Teknologi Informasi

Universitas Kristen Duta Wacana

Tahun 2012

**IMPLEMENTASI ALGORITMA *NEGASCOUT* UNTUK
PERMAINAN *CHECKERS***

Tugas Akhir



Diajukan kepada Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai salah satu syarat dalam memperoleh gelar
Sarjana Komputer



Disusun oleh :

Aditya Kurniawan Effendi

22074279

Program Studi Teknik Informatika

Universitas Kristen Duta Wacana

Tahun 2012

PERNYATAAN KEASLIAN TUGAS AKHIR


Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul :

Implementasi Algoritma *Negascout* untuk Permainan *Checkers*

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan sarjana Program studi Teknik Informatika, Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya

Yogyakarta, 1 Mei 2012



Aditya Kurniawan Effendi

(_____)
22074279



HALAMAN PERSETUJUAN

Judul : Implementasi Algoritma *Negascout* untuk Permainan
Checkers
Nama : Aditya Kurniawan Effendi
NIM : 22074279
Mata Kuliah : Tugas Akhir
Kode : TIW276
Semester : Genap
Tahun Akademik : 2011/2012

Telah diperiksa dan disetujui
Di Yogyakarta,
Pada Tanggal... 30-4-2012



Dosen Pembimbing I

Rosa Delima, S.Kom, M.Kom

Dosen Pembimbing II

Antonius Rachmat C, S.Kom., M.Cs.

HALAMAN PENGESAHAN

SKRIPSI

IMPLEMENTASI ALGORITMA *NEGASCOUT* UNTUK PERMAINAN
CHECKERS

Oleh : Aditya Kurniawan Effendi / 22074279

Dipertahankan di depan dewan Penguji Tugas Akhir/Skripsi
Program Studi Teknik Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana – Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu

Syarat memperoleh gelar
Sarjana Komputer
Pada tanggal

18 Mei 2012

Yogyakarta, 22/5/2012
Mengesahkan,

Dewan Penguji :

1. Rosa Delima, S.Kom., M.Kom.

2. Antonius Rachmat C, S.Kom., M.Cs.

3. Restyandito, S.Kom., MSIS.

4. Aloysius Airlangga Bajuadji, S.Kom., M.Eng.



Dekan

(Drs. Wimmie Handiwidjojo, MIT)

Ketua Program Studi

(Nugroho Agus H., S.Si, M.Si.)

UCAPAN TERIMA KASIH

Dengan Selesainya Tugas Akhir ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

- a. Tuhan Yang Maha Esa yang telah memberkati dan memberikan kekuatan dalam setiap cobaan dan tantangan yang harus dihadapi untuk menyelesaikan tugas akhir ini.
- b. **Ibu Rosa Delima, S.Kom., M.Kom.** selaku dosen Pembimbing I yang telah memberikan bimbingan, pengarahan dan masukan kepada penulis, juga kepada
- c. **Bapak Antonius Rachmat C, S.Kom,M.Cs.**selaku dosen Pembimbing II yang telah membimbing dengan sabar sejak awal pembuatan tugas akhir ini.
- d. Keluarga tercinta yang telah memberikan dukungan baik dalam bentuk dana, masukan, dan juga doa hingga tugas akhir ini selesai.
- e. Teman-teman terdekat yang telah memberikan bantuan dalam penyelesaian masalah yang terjadi dalam tugas akhir ini.

Penulis menyadari bahwa program dan laporan tugas akhir ini memiliki banyak kekurangan dan belum sempurna. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun dari pembaca sekalian, sehingga suatu saat dapat memberikan karya yang lebih baik lagi.



Yogyakarta, Mei 2012

Penulis

ABSTRAK

Negascout ditemukan oleh Alexander Reinefeld pada tahun 1983. Algoritma ini adalah optimalisasi dari algoritma *Minimax* dengan menggunakan *zero-width window* untuk menentukan node mana yang akan dipotong. Berbeda dengan *Alpha-Beta Pruning* yang menggunakan jendela pencarian yang lebih besar yaitu alpha dan beta.

Dalam penelitian ini algoritma *Negascout* akan diimplementasikan pada permainan *Checkers*. Permainan ini adalah salah satu permainan *zero-sum game* yang berarti pemain dinyatakan menang jika pemain kalah, seandainya pemain membutuhkan 1 poin untuk menang, maka untuk kalah pemain dinyatakan kalah jika mendapatkan nilai -1. Dalam permainan ini setiap pemain dapat mengetahui semua kondisi permainan, seperti jumlah bidak yang dimiliki pemain dan lawan, posisi bidak, dan langkah apa saja yang dimiliki oleh pemain maupun lawan.

Penerapan algoritma *Negascout* pada permainan ini akan dibandingkan dengan algoritma *Alpha-Beta Pruning* untuk melihat performanya. Kedua algoritma digunakan untuk mencari solusi terbaik dari 5 kondisi papan yang diberikan dan diberikan batas *depth* 4, 6 dan 8.

Dari 86% hasil pengujian yang dilakukan algoritma *Negascout* membutuhkan waktu lebih singkat dari algoritma *Alpha-Beta Pruning* pada pengujian yang dilakukan. Selain itu hasil pengujian juga menunjukkan algoritma *Negascout* menghasilkan satu node solusi yang berbeda dengan algoritma *Alpha-Beta Pruning*. Dilihat dari nilai heuristik yang dihasilkan algoritma *Negascout* menghasilkan solusi yang sama baiknya dengan algoritma *Alpha-Beta Pruning*.

DAFTAR ISI

HALAMAN JUDUL.....	
PERNYATAAN KEASLIAN SKRIPSI.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
UCAPAN TERIMA KASIH.....	iv
ABSTRAK.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	ix
DAFTAR GRAFIK.....	x
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Hipotesis Masalah.....	2
1.5. Tujuan Penelitian.....	2
1.6. Metode/Pendekatan.....	3
1.7. Sistematika Penulisan.....	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1. Tinjauan Pustaka.....	5
2.2. Landasan Teori.....	6
2.2.1. <i>Checkers</i>	6
2.2.2. <i>Minimax</i>	7
2.2.3. <i>Alpha-Beta Pruning</i>	8
2.2.4. <i>Negascout</i>	9
2.2.5. Fungsi Evaluasi.....	11
BAB 3 ANALISIS DAN PERANCANGAN SISTEM.....	13
3.1. Kebutuhan Perangkat Keras dan Lunak.....	13
3.1.1. Kebutuhan Perangkat Keras.....	13

3.1.2. Kebutuhan Perangkat Lunak	13
3.2. Perancangan Kerja Sistem.....	13
3.2.1. Fungsi Evaluasi dan Implementasi Algoritma <i>Negascout</i>	13
3.2.2. Perancangan Proses	18
3.2.2.1. Perancangan Proses Permainan <i>Checkers</i>	18
3.2.2.2. Algoritma <i>Negascout</i> Pada Permainan <i>Checkers</i>	20
3.3. Perancangan Antar Muka	22
3.4. Perancangan Struktur Data.....	24
3.5. Perancangan Pengujian Sistem.....	25
BAB 4 IMPLEMENTASI DAN ANALISIS SISTEM.....	27
4.1. Hasil Implementasi Sistem.....	27
4.1.1. Tampilan Form Permainan <i>Checkers</i>	27
4.2. Implementasi <i>Negascout</i> dan <i>Alpha-Beta</i> Pada Permainan <i>Checkers</i>	31
4.3. Analisis Sistem	33
4.3.1. Cara Pengujian.....	33
4.3.2. Hasil Perbandingan <i>Negascout</i> dan <i>Alpha-Beta</i>	35
4.3.3. Hasil Pengujian <i>User</i> Melawan <i>Negascout</i> dan <i>Alpha-Beta</i>	39
BAB 5 KESIMPULAN DAN SARAN	41
5.1. Kesimpulan.....	41
5.2. Saran.....	42



DAFTAR TABEL

Tabel 4.1. Perbandingan <i>Negascout</i> dan <i>Alpha-Beta</i> Dengan <i>Depth</i> 4.....	35
Tabel 4.2. Perbandingan <i>Negascout</i> dan <i>Alpha-Beta</i> Dengan <i>Depth</i> 6.....	36
Tabel 4.3. Perbandingan <i>Negascout</i> dan <i>Alpha-Beta</i> Dengan <i>Depth</i> 8.....	37
Tabel 4.4. Hasil Pengujian <i>User</i> Melawan <i>Negascout</i>	39
Tabel 4.5. Hasil Pengujian <i>User</i> Melawan <i>Alpha-Beta</i>	39

© UKDW

DAFTAR GAMBAR

Gambar 2.1. Papan <i>Checkers</i>	6
Gambar 2.2. <i>Search Tree Minimax</i>	8
Gambar 2.3. <i>Search Tree Alpha-Beta Pruning</i>	9
Gambar 2.4. Pseudocode <i>Negascout</i>	10
Gambar 2.5. <i>Search Tree Negascout</i>	11
Gambar 3.1. Contoh Kondisi Papan.....	15
Gambar 3.2. Pohon Pencarian Permainan <i>Checkers</i>	16
Gambar 3.3. <i>Flowchart</i> Proses Permainan.....	19
Gambar 3.4. <i>Flowchart</i> Algoritma <i>Negascout</i>	21
Gambar 3.5. Rancangan Form Utama.....	23
Gambar 3.6. Rancangan Form Pohon Pencarian	23
Gambar 3.7. Class Kotak	24
Gambar 3.8. <i>Array of Class</i> Kotak.....	24
Gambar 4.1. Form Utama	27
Gambar 4.2. Menu New Game	28
Gambar 4.3. Menu Permainan Baru.....	28
Gambar 4.4. Menu Perbandingan.....	28
Gambar 4.5. Tabel Informasi	29
Gambar 4.6. Tabel Solusi Komputer.....	29
Gambar 4.7. Tabel History Langkah.....	30
Gambar 4.8. Papan Permainan	30
Gambar 4.9. Form Pohon	31
Gambar 4.10. Pseudocode <i>Negascout</i>	32
Gambar 4.11. Pseudocode <i>Alpha-Beta</i>	32

DAFTAR GRAFIK

Grafik 4.1. Perbandingan Jumlah Node	38
Grafik 4.2. Perbandingan Waktu.....	38

© UKDW

BAB 1

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan berkembangnya teknologi komputer, teknologi perangkat lunak juga berkembang. Salah satu teknologi perangkat lunak yang berkembang adalah aplikasi permainan. Dewasa ini sudah bermunculan berbagai macam jenis aplikasi permainan. Dari permainan klasik seperti kartu, *board games*, sampai permainan-permainan lainnya yang lebih kompleks.

Semua permainan itu diperlukan sebuah kecerdasan yang memungkinkan komputer untuk bermain melawan manusia. Cabang ilmu komputer yang berusaha untuk menirukan kecerdasan manusia adalah kecerdasan buatan atau *Artificial Intelligence*(A.I.).

Salah satu permainan yang menggunakan A.I. adalah *checkers*, Pada permainan ini pemain dinyatakan menang jika pemain lain kalah, seandainya pemain membutuhkan 1 poin untuk menang, maka untuk kalah pemain membutuhkan poin -1. Permainan seperti ini disebut juga dengan *zero-sum game* yang berarti kemenangan pemain adalah kekalahan pemain lainnya. Dalam permainan ini setiap pemain dapat mengetahui semua kondisi permainan, seperti jumlah bidak yang dimiliki pemain dan lawan, posisi bidak, dan langkah apa saja yang dimiliki oleh pemain maupun lawan. Oleh karena itu maka permainan *checkers* ini sangat cocok jika mengimplementasikan algoritma *negascout* untuk penentuan langkah komputer. *Negascout* ditemukan oleh Alexander Reinefeld tahun 1983. Algoritma ini berusaha memotong node-node dengan melihat dahulu node yang akan dicek, apabila node tersebut memiliki nilai yang lebih baik maka akan dilakukan pencarian ulang dengan menggunakan *search window* seperti algoritma *alpha-beta pruning* untuk mengetahui nilai asli dari node tersebut.

1.2. Perumusan Masalah

Dalam penelitian ini ada dua rumusan masalah, yaitu :

1. Bagaimana mengimplementasikan algoritma *negascout* pada permainan *checkers* ?
2. Membandingkan performa algoritma *negascout* dengan algoritma *minimax alpha-beta pruning*.

1.3. Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut :

1. Aplikasi hanya dapat berjalan pada OS Windows.
2. Aplikasi menggunakan VB.NET
3. Aplikasi tidak terhubung ke jaringan.
4. Komputer hanya dapat memainkan satu sisi.
5. Aplikasi tidak dapat digunakan untuk mode 2 pemain manusia.
6. Aplikasi dapat menggunakan satu atau dua algoritma untuk pencarian solusi.

1.4. Hipotesis Masalah

Di dalam pengerjaan tugas akhir ini, penulis memiliki tujuan membuktikan bahwa algoritma *negascout* dapat diimplementasikan untuk permainan *checker* dan menghasilkan hasil pencarian solusi yang sama dengan algoritma *minimax alpha-beta pruning* dengan waktu yang lebih singkat.

1.5. Tujuan Penelitian

Penelitian ini bertujuan untuk mengimplemantasikan algoritma *negascout* untuk permainan *checkers* dan membandingkan performa *negascout* dengan *minimax alpha-beta pruning*.

1.6. Metode / Pendekatan

Metode atau pendekatan yang digunakan dalam penyelesaian tugas akhir ini adalah sebagai berikut :

1. Metode penelitian memanfaatkan studi pustaka dari sumber-sumber yang mendukung untuk pengimplementasian algoritma *negascout* untuk permainan *checkers*.
2. Mengimplementasikan teori-teori yang telah dipelajari selama perkuliahan ke dalam program yang berhubungan dengan pembuatan tugas akhir.
3. Pembuatan program permainan *checkers* dengan menggunakan algoritma *minimax alpha-beta pruning* dan *negascout*.
4. Pengujian dilakukan menggunakan program yang dibuat untuk membandingkan besar *search tree* dan waktu yang dibutuhkan untuk mencari solusi.
5. Penulisan laporan tugas akhir.

1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam tugas akhir ini terbagi menjadi lima (5) bab, yaitu:

Pada bab 1 Pendahuluan, yang memberikan gambaran umum mengenai apa yang diteliti dalam penulisan tugas akhir ini. Pendahuluan memuat latar belakang masalah, perumusan masalah, batasan masalah, hipotesis, tujuan penelitian, metode penelitian, dan sistematika penulisan laporan.

Pada bab 2 Tinjauan Pustaka,yang terdiri dari tinjauan pustaka dan landasan teori tentang permainan *checker* dan algoritma *negascout*.

Pada bab 3 Perancangan Sistem, Mencakup tahap perancangan sistem yang akan dibuat seperti kebutuhan *hardware* dan *software*,spesifikasi sistem, arsitektur sistem, *flowchart* diagram, algoritma yang digunakan dalam membuat sistem, kamus data, rancangan antarmuka dan rancangan pengujian sistem.

Pada bab 4 Implementasi dan Analisis Sistem, akan diuraikan mengenai hasil riset atau implementasi dan pembahasan atau analisis dari riset.

Pada bab 5 Kesimpulan dan Saran, berisi kesimpulan dari hasil penelitian yang dilakukan dan saran untuk memberikan hasil yang lebih baik lagi dalam penelitian yang sejenis.

© UKDW

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan implementasi serta setelah melakukan analisa algoritma *Negascout* pada permainan *Checkers* didapatkan kesimpulan sebagai berikut:

- Dari semua pengujian yang dilakukan 86% hasil pengujian menunjukkan penerapan algoritma *Negascout* pada permainan *Checkers* menghasilkan jumlah node yang lebih sedikit dan waktu pencarian yang lebih singkat dibandingkan dengan algoritma *Alpha-Beta Pruning* terutama pada *depth* 8.
- Penerapan algoritma *Negascout* pada permainan *Checkers* menghasilkan satu node solusi yang berbeda dengan algoritma *Alpha-Beta Pruning* dengan nilai heuristik yang sama.
- Melihat dari nilai heuristik yang dihasilkan algoritma *Negascout* pada pengujian yang dilakukan algoritma ini menghasilkan solusi yang sama dengan algoritma *Alpha-Beta Pruning*.
- Apabila algoritma *Negascout* melakukan pencarian ulang pada level-level awal, maka algoritma ini akan mengecek ulang semua anak node dari node yang dilakukan pencarian ulang. Apabila pencarian ulang terjadi pada level atas maka *Negascout* akan menghasilkan node yang lebih banyak dibandingkan dengan pencarian ulang pada level-level bawah.
- Dari pengujian komputer lawan *user* yang dilakukan dapat dilihat penerapan algoritma *Negascout* pada permainan *Checkers* menghasilkan kecerdasan komputer yang hampir sama dengan algoritma *Alpha-Beta Pruning* dengan waktu respon yang lebih cepat.
- Dari pengujian komputer lawan *user*, komputer memiliki persentase kemenangan lebih besar dengan *depth* 6 dibandingkan dengan *depth* 4.

Sedangkan *depth* 8 memerlukan waktu terlalu lama sehingga tidak memungkinkan untuk digunakan ketika melawan *user*.

5.2. Saran

Saran terhadap pengembangan sistem untuk mencapai hasil yang lebih baik antara lain :

- Fungsi evaluasi yang digunakan belum cukup bagus, hal ini dapat dilihat dari pengujian kedua dimana komputer masih dapat dikalahkan oleh *user*. Untuk itu diperlukan fungsi baru yang dapat merepresentasikan kondisi papan lebih akurat untuk menghasilkan A.I. yang lebih baik. Fungsi evaluasi yang digunakan menghitung faktor-faktor seperti : jumlah bidak, jenis bidak, lokasi bidak dan jumlah langkah yang tersedia. Mungkin dengan ditambahkan faktor lain seperti awal permainan atau akhir permainan dapat meningkatkan representasi kondisi papan.
- Dari 13% hasil pengujian menunjukkan *Negascout* menghasilkan jumlah node yang lebih banyak dari *Alpha-Beta Pruning*. Hal tersebut dikarenakan *Negascout* melakukan terlalu banyak pencarian ulang yang disebabkan jeleknya *move ordering* yang ada. Untuk menghasilkan performa *Negascout* yang lebih baik diperlukan penambahan *move ordering* yang baik.

DAFTAR PUSTAKA

- Ayuningtyas, N. (2008). *Algoritma Minimax Dalam Permainan Checkers*. Diakses 2 Januari 2012, dari <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2007-2008/Makalah2008/MakalahIF2251-2008-031.pdf>
- Ertel, W. (2011). *Introduction To Artificial Intelligence*. Germany: Springer.
- Gunawan, Kristian, Y., & Andika, H. (2009). *Game Playing Untuk Othello Dengan Menggunakan Algoritma Negascout dan MTDf*. Diakses 2 Januari 2012, dari <http://journal.uui.ac.id/index.php/Snati/article/viewFile/1278/1088>
- Millington, I. (2006). *Artificial Intelligence For Games*. San Francisco : Morgan Kaufmann.
- Reinefeld, A. (1983). *An improvement to the scout tree algorithm*. Diakses 2 Januari 2012, dari <http://www.top-5000.nl/ps/An%20improvement%20to%20the%20scout%20tree%20search%20algorithm.pdf>
- Reinefeld, F. (2011). *How To Win at Checkers*. Diakses 2 Januari 2012, dari <http://www.bobnewell.net/checkers/howtowin.pdf>
- Schaeffer, J. (2009). *One Jump Ahead*. New York : Springer.
- Schaeffer, J., Lake, R. (1996). *Solving the Game of Checkers*. Diakses 2 Januari 2012, dari <http://library.msri.org/books/Book29/files/schaeffer.pdf>
- Schaeffer, J., Lake, R., Lu, P., & Bryant, M. (n.d). *Chinook : The World Man-Machine Checkers Champion*. Diakses 2 Januari 2012, dari http://webdocs.cs.ualberta.ca/~jonathan/publications/ai_publications/aimag96.pdf