

**ANALISIS DAN IMPLEMENTASI KECERDASAN BUATAN
PADA PERMAINAN CHECKER MENGGUNAKAN
ALGORITMA MINIMAX DENGAN NEGASCOUT**

SKRIPSI

**MUHAMMAD AIDIL AKBAR
061401020**



**PROGRAM STUDI S1 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SUMATERA UTARA
MEDAN
2011**

**ANALISIS DAN IMPLEMENTASI KECERDASAN BUATAN
PADA PERMAINAN CHECKER MENGGUNAKAN
ALGORITMA MINIMAX DENGAN NEGASCOUT**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat mencapai gelar
Sarjana Komputer

**MUHAMMAD AIDIL AKBAR
061401020**



**PROGRAM STUDI S1 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SUMATERA UTARA
MEDAN
2011**

PERSETUJUAN

Judul : ANALISIS DAN IMPLEMENTASI KECERDASAN
BUATAN PADA PERMAINAN CHECKER
MENGUNAKAN ALGORITMA MINIMAX
DENGAN NEGASCOUT

Kategori : SKRIPSI

Nama : MUHAMMAD AIDIL AKBAR

Nomor Induk Mahasiswa : 061401020

Program Studi : SARJANA (S1) ILMU KOMPUTER

Departemen : ILMU KOMPUTER

Fakultas : MATEMATIKA DAN ILMU PENGETAHUAN
ALAM (FMIPA) UNIVERSITAS SUMATERA
UTARA

Diluluskan di
Medan,

Komisi Pembimbing :

Pembimbing 2

Pembimbing 1

M. Andri B, ST, M.CompSc, MEM
NIP 197510082008011011

Drs. Marihat Situmorang, M.Kom
NIP 196312141989031001

Diketahui/Disetujui oleh
Program Studi S1 Ilmu Komputer
Ketua,

Prof. Dr. Muhammad Zarlis
NIP 195707011986011003

PERNYATAAN

ANALISIS DAN IMPLEMENTASI KECERDASAN BUATAN PADA PERMAINAN CHECKER MENGGUNAKAN ALGORITMA MINIMAX DENGAN NEGASCOUT

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan,

Muhammad Aidil Akbar
061401020

PENGHARGAAN

Puji syukur penulis sampaikan kehadiran Allah SWT, yang telah memberikan rahmat dan hidayah-Nya serta segala sesuatu dalam hidup, sehingga penulis dapat menyelesaikan penyusunan skripsi ini, sebagai syarat untuk memperoleh gelar Sarjana Komputer, Program Studi S1 Ilmu Komputer Departemen Ilmu Komputer Universitas Sumatera Utara.

Ucapan terima kasih penulis sampaikan kepada Drs. Marihat Situmorang, M.Kom selaku pembimbing pertama dan M. Andri B, ST, M.CompSc, MEM selaku pembimbing kedua yang telah banyak meluangkan waktunya dalam memberikan masukan-masukan kepada penulis. Ucapan terima kasih juga ditujukan kepada Prof. Dr. Iryanto, M.Si dan Dra. Mardiningsih, M.Si yang telah bersedia menjadi dosen penguji yang telah banyak memberikan kritik dan saran agar skripsi ini mendekati kesempurnaan. Ucapan terima kasih juga ditujukan kepada Ketua dan Sekretaris Departemen Ilmu Komputer, Prof. Dr. Muhammad Zarlis dan Syahriol Sitorus, S.Si, MIT, Dekan dan Pembantu Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, semua dosen serta pegawai di Program Studi S1 Ilmu Komputer Departemen Ilmu Komputer FMIPA USU.

Tidak lupa penulis ucapkan terima kasih kepada keluarga tercinta, Ayahanda H. Arzil Alwi dan Ibunda Hj. Mahraini Rangkuti yang memberikan dukungan, doa, dan semangat tanpa henti. Untuk abang dan kakak penulis, M. Abduh Aswin, S.Kom dan Khairiah, S.E. Terima kasih penulis ucapkan kepada teman-teman yang selalu memberikan dukungan, Rury Handayani, Anggarani Novitasari, Wenty Suma, Izhari Ishak Aksa, Alfarisi, Surya Wijaya serta teman-teman RCS dan seluruh teman-teman yang tidak disebutkan namanya. Sekali lagi penulis mengucapkan terima kasih kepada semua pihak yang membantu dalam penyelesaian skripsi ini yang tidak dapat disebutkan satu persatu, terima kasih atas ide, saran dan motivasi yang diberikan.

ABSTRAK

Checker ialah suatu permainan papan yang 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. Penulis membangun aplikasi permainan Checker berbasis Kecerdasan Buatan menggunakan algoritma Minimax dengan Negascout. Negascout mengefisienkan waktu eksekusi program dan memungkinkan agen untuk berpikir secara cerdas. Aplikasi ini dikembangkan dengan menggunakan metode perancangan UML dan bahasa pemrograman Java.

ANALYSIS AND IMPLEMENTATION OF ARTIFICIAL INTELLIGENCE IN CHECKER USING MINIMAX ALGORITHM WITH NEGASCOUT

ABSTRACT

Checker is a board game that uses pure skill between two players which follow certain rules in the game, and each player tries to win the game by eliminating all the opponent's pieces in a board or by making all the pieces of the opponent can not move. Checker game application based on Artificial Intelligence using Minimax algorithm with Negascout. Negascout allows efficient program execution time and causes agents to think intelligently. This application is developed using UML and Java programming language.

DAFTAR ISI

	Halaman
Persetujuan	ii
Pernyataan	iii
Penghargaan	iv
Abstrak	v
Abstract	vi
Daftar Isi	vii
Daftar Tabel	ix
Daftar Gambar	x
Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penulisan Tugas Akhir	4
1.5 Manfaat Penulisan Tugas Akhir	4
1.6 Metodologi Penelitian	4
1.7 Sistematika Penulisan	5
Bab 2 Tinjauan Pustaka	6
2.1 Kecerdasan Buatan	6
2.1.1 Tujuan Akhir Kecerdasan Buatan	7
2.2 Agen Cerdas	8
2.2.1 Karakteristik Lingkungan Agen	8
2.3 Permainan Checker	11
2.3.1 Papan dalam Permainan Checker	11
2.3.2 Bidak dalam Permainan Checker	11
2.3.3 Metode Permainan	12
2.3.4 Bidak Pion	12
2.3.5 Bidak Raja	14
2.3.6 Gerakan Menangkap Secara Umum	15
2.3.7 Kemenangan Permainan	16
2.3.8 Permainan Seri	16
2.4 Pohon Permainan	17
2.5 Algoritma Minimax	19
2.6 Negascout	22
2.6.1 Pemotongan Alpha	23
2.6.2 Pemotongan Beta	24
2.6.3 Pemotongan Alpha Beta	24
2.6.4 <i>Zero-Width Test</i>	24

Bab 3	Analisis dan Perancangan	26
3.1	Fungsi Evaluasi pada Checker	26
3.1.1	Nilai Bidak	27
3.1.2	Nilai Melompat	27
3.1.3	Nilai Posisi	27
3.1.4	Nilai Penobatan Raja	28
3.2	Pohon Permainan dan Nilai Evaluasi	28
3.3	Analisis Algoritma Minimax	32
3.4	Analisis Algoritma Negascout	34
3.4.1	<i>Pseudo Code</i>	36
3.4.2	Kinerja Algoritma	37
3.5	Pemodelan Visual Menggunakan UML	37
3.5.1	Identifikasi <i>Use Case Diagram</i>	36
3.5.1.1	<i>Use Case</i> Permainan Baru	39
3.5.1.2	<i>Use Case</i> Pindah Bidak	41
3.5.1.3	<i>Use Case</i> Help About	43
3.5.1.4	<i>Use Case</i> Keluar Permainan	45
3.5.2	Perancangan <i>Class Diagram</i> Checker	46
Bab 4	Implementasi dan Pengujian	49
4.1	Implementasi	49
4.1.1	Konfigurasi Perangkat Keras	49
4.1.2	Konfigurasi Perangkat Lunak	50
4.1.3	Hasil Eksekusi Aplikasi	50
4.2	Pengujian Agen Cerdas	55
4.2.1	Sampel Posisi Pertama	55
4.2.2	Sampel Posisi kedua	58
4.2.3	Sampel Posisi Ketiga	61
4.2.4	Pengujian Waktu Pencarian	64
Bab 5	Kesimpulan dan Saran	65
5.1	Kesimpulan	65
5.2	Saran	66
	Daftar Pustaka	67

DAFTAR TABEL

	Halaman
Tabel 3.1 Proses menghitung nilai evaluasi	31
Tabel 3.2 Analisis algoritma Mimimax	33
Tabel 3.3 Analisis algoritma Negascout	35
Tabel 3.4 Tabel <i>pseudo code</i> Negascout	36
Tabel 3.5 Dokumentasi Naratif <i>Use Case</i> Permainan Baru	39
Tabel 3.6 Dokumentasi Naratif <i>Use Case</i> Pindah Bidak	41
Tabel 3.7 Dokumentasi Naratif <i>Use Case</i> Help About	43
Tabel 3.8 Dokumentasi Naratif <i>Use Case</i> Keluar Permainan	45
Tabel 3.9 Penjelasan Kelas-Kelas Pada <i>Class Diagram</i> Checker	48
Tabel 4.1 Pohon Permainan untuk Sampel Pertama	56
Tabel 4.2 Proses Minimax untuk Sampel Pertama	57
Tabel 4.3 Pohon Permainan untuk Sampel kedua	59
Tabel 4.4 Proses Minimax untuk Sampel Kedua	60
Tabel 4.5 Pohon Permainan untuk Sampel Ketiga	62
Tabel 4.6 Proses Minimax untuk Sampel Ketiga	63
Tabel 4.7 Pengujian waktu pada algoritma Minimax	64
Tabel 4.8 Pengujian waktu pada algoritma Negascout	64

DAFTAR GAMBAR

	Halaman
Gambar 1.1 Papan dan Bidak dalam Checker, serta posisi awal permainan	2
Gambar 2.1 Agen berinteraksi dengan lingkungan	8
Gambar 2.2 Bentuk papan Checker dan kondisi awal permainan	12
Gambar 2.3 Bidak pion	13
Gambar 2.4 Gerakan sederhana pion	13
Gambar 2.5 Gerakan menangkap pion	13
Gambar 2.6 Bidak raja	14
Gambar 2.7 Gerakan sederhana raja	14
Gambar 2.8 Gerakan menangkap raja	15
Gambar 2.9 Gerakan beberapa lompatan	15
Gambar 2.10 Contoh pohon permainan tic-tac-toe	18
Gambar 2.11 Langkah pemain pada algoritma Minimax	20
Gambar 2.12 Langkah lawan pada algoritma Minimax	20
Gambar 2.13 Ilustrasi cara kerja algoritma Minimax	21
Gambar 2.14 Pohon permainan dengan pemotongan alpha	23
Gambar 2.15 Pohon permainan dengan algoritma Negascout	25
Gambar 3.1 Strategi perpindahan bidak Checker	27
Gambar 3.2 Kondisi permainan pada Checker	29
Gambar 3.3 Pohon permainan pada Checker	29
Gambar 3.4 Pohon dengan nilai evaluasi pada setiap langkah	30
Gambar 3.5 Pohon dengan nilai evaluasi pada <i>leaf node</i>	31
Gambar 3.6 Proses perubahan nilai node dengan algoritma Minimax	32
Gambar 3.7 Pohon permainan menggunakan algoritma Negascout	34
Gambar 3.8 <i>Use case diagram</i> Checker	38
Gambar 3.9 <i>Activity diagram</i> Permainan Baru	40
Gambar 3.10 <i>Activity diagram</i> Pindah Bidak	42
Gambar 3.11 <i>Activity diagram</i> Help About	44
Gambar 3.12 <i>Activity diagram</i> Keluar Permainan	46
Gambar 3.13 <i>Class diagram</i> Checker	47
Gambar 4.1 Tampilan awal aplikasi permainan Checker	50
Gambar 4.2 Tampilan langkah yang mungkin untuk bidak yang telah dipilih	52
Gambar 4.3 Tampilan ketika agen mendapat giliran melangkah	53
Gambar 4.4 Tampilan jendela New Game	53
Gambar 4.5 Tampilan akhir dari permainan ketika bidak lawan telah habis	54
Gambar 4.6 Tampilan About The Checker	54
Gambar 4.7 Sampel posisi pertama	55
Gambar 4.8 Sampel posisi kedua	58
Gambar 4.9 Sampel posisi ketiga	61