

**ANALISIS DAN IMPLEMENTASI PENYELESAIAN *GAME*  
*MINESWEEPER* MENGGUNAKAN ALGORITMA  
*GREEDY BEST FIRST SEARCH***

**SKRIPSI**

**IRMA Y N SIGIRO  
061401069**



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

ANALISIS DAN IMPLEMENTASI PENYELESAIAN *GAME*  
*MINESWEEPER* MENGGUNAKAN ALGORITMA  
*GREEDY BEST FIRST SEARCH*

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat mencapai gelar  
Sarjana Komputer

IRMA Y N SIGIRO

061401069



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

## PERSETUJUAN

Judul : ANALISIS DAN IMPLEMENTASI  
PENYELESAIAN *GAME MINESWEEPER*  
MENGUNAKAN ALGORITMA *GREEDY BEST*  
*FIRST SEARCH*

Kategori : SKRIPSI

Nama : IRMA Y N SIGIRO

Nomor Induk Mahasiswa : 061401069

Program Studi : SARJANA (S1) ILMU KOMPUTER

Departemen : ILMU KOMPUTER

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

Diluluskan di  
Medan, 17 Maret 2011

Komisi Pembimbing :

Pembimbing 2

Pembimbing 1

Syahril Efendi, S.Si, MIT  
NIP. 196711101996021001

Drs. Sawaluddin, MIT  
NIP. 195912311998021001

Diketahui/Disetujui oleh  
Program Studi S1 Ilmu Komputer  
Ketua,

Dr. Poltak Sihombing, M.KOM  
NIP. 196203171991031001

**PERNYATAAN****ANALISIS DAN IMPLEMENTASI PENYELESAIAN *GAME*  
*MINESWEEPER* MENGGUNAKAN ALGORITMA  
*GREEDY BEST FIRST SEARCH*****SKRIPSI**

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

Medan, 17 Maret 2011

IRMA Y N SIGIRO  
061401069

## **PENGHARGAAN**

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena kasih dan karuniaNya yang selalu menyertai penulis sehingga kertas kajian ini berhasil diselesaikan dalam waktu yang telah ditetapkan.

Ucapan terima kasih penulis sampaikan kepada Drs. Sawaluddin, MIT dan Syahril Efendi, S.Si, MIT selaku pembimbing pada penyelesaian skripsi ini yang telah memberikan panduan dan penuh kepercayaan kepada penulis untuk menyempurnakan kajian ini. Ucapan terimakasih juga penulis sampaikan kepada Drs. Suyanto, M.Kom dan M. Andri B., ST, MCompSc, MEM selaku dosen penguji. Panduan ringkas, padat, dan profesional telah diberikan kepada penulis agar dapat menyelesaikan tugas ini. Ucapan terima kasih juga ditujukan kepada Ketua dan Sekretaris Departemen Ilmu Komputer, Dr. Poltak Sihombing, M.KOM dan Maya Silvi Lydia, B.Sc, M.Sc, Dekan dan Pembantu Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara, semua dosen pada Departemen Ilmu Komputer FMIPA USU, dan pegawai di Ilmu Komputer FMIPA USU. Tidak lupa juga penulis ucapkan terima kasih yang tak terhingga kepada kedua orang tua tercinta Rahman Sigiroy dan Tianur Damanik yang selalu memberikan cinta kasihnya dan dukungan, baik secara materil dan spiritual. Penulis juga mengucapkan terima kasih kepada k'Jeny, k'Desri, k'Vea, b'Leston, dan adek penulis Anwar Sigiroy yang penuh perhatian kepada penulis selama dalam studi serta rekan-rekan kuliah, khususnya Nurinda, Anne, Emnita, Yelly, Philip, kadar dan rekan-rekan yang lainnya yang tidak bisa disebutkan satu per satu oleh penulis yang selalu memberikan semangat dan dorongan kepada penulis. Semoga Tuhan Yang Maha Esa memberikan limpahan karunia kepada semua pihak yang telah memberikan bantuan, perhatian, serta dukungan kepada penulis dalam menyelesaikan skripsi ini.

Saya menyadari bahwa skripsi ini masih jauh dari kesempurnaan, oleh karena itu saya menerima saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini. Sehingga dapat bermanfaat bagi kita semuanya.

## ABSTRAK

*Minesweeper* merupakan salah satu permainan (*game*) yang pada umumnya ditemukan pada sistem operasi Windows. *Minesweeper* ini cukup sulit diselesaikan, sehingga sangat jarang seorang pemain bisa dipastikan memenangkan permainan ini. Tema yang diangkat dalam permainan ini adalah menemukan seluruh ranjau yang tersebar pada petak yang telah disediakan tanpa meledakkannya. Ranjau dapat ditemukan dengan menggunakan petunjuk yang tertera, yakni nilai yang ditampilkan pada petak yang telah terbuka. Dalam hal ini, algoritma pencarian terbaik pertama (*best first search*) digunakan untuk membantu menyelesaikan masalah ini. Algoritma pencarian terbaik pertama (*best first search*) merupakan teknik pencarian solusi yang menggabungkan kelebihan dari algoritma pencarian melebar pertama (*breadth first search*) dengan algoritma pencarian mendalam pertama (*depth first search*), sehingga metode ini merupakan gabungan dari kedua metode tersebut. Semua kemungkinan solusi dibuat dalam bentuk pohon ruang keadaan terlebih dahulu, kemudian dengan menggunakan algoritma *best first search* dilakukan pencarian berdasarkan nilai heuristik dari tiap-tiap cabang pada pohon. Cabang dengan nilai heuristik terbaik atau lebih menjanjikan menuju solusi akan terlebih dahulu dikerjakan. Dalam kajian ini pencarian dilakukan di dalam kotak seleksi ukuran 4x4, dimana kotak seleksi ini dapat membantu *user* dalam menyelesaikan permainan. Dengan menggunakan bantuan kotak seleksi *user* membutuhkan waktu yang lebih singkat dalam menyelesaikan permainan ini, khususnya pada level *intermediate* dan *expert*.

## **MINESWEEPER GAME RESOLUTION ANALYSIS AND IMPLEMENTATION USING GREEDY BEST FIRST SEARCH ALGORITHM**

### **ABSTRACT**

Minesweeper is a game generally found in Windows operating systems. Minesweeper is difficult to solve, that is the reason why it's rarely a game player can win this game. The goal of this game is to uncover all free cells and leaving all mined cells covered. Mined cells can be found by using the information contained on the uncovered cells. In this paper presented a heuristic algorithm exactly best first search method for helping to solve these problems. Best first search method is a method of searching that combine the excesses of breadth first search(BFS) method and depth first search(DFS) method. That's why this method often called as a combination of both of that two methods. All of solution possibilities are presented in tree state space first, then searching is done on every state based on it's heuristic value of every state by using best first search algorithm. A state with best heuristic value or more promising to find a solution is done first. On this study it's described the searching for box selection as large as 4 x 4. This box selection will help the game player to finish/win the game. By using the help of this box selection, a user will finish this game in shorter time different with using user's ability only.

## DAFTAR ISI

	Halaman
Persetujuan	ii
Pernyataan	iii
Penghargaan	iv
Abstrak	v
Abstract	vi
Daftar Isi	vii
Daftar Tabel	x
Daftar Gambar	xi
 Bab 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	3
1.7 Sistematika Penulisan	4
 Bab 2 Tinjauan Pustaka	6
2.1 Algoritma Pencarian	6
2.1.1 Algoritma Pencarian Mendalam Pertama ( <i>Depth First Search</i> )	7
2.1.2 Pencarian Heuristik	9
2.1.3 Algoritma Pencarian Terbaik Pertama ( <i>Best First Search</i> )	12
2.2 Analisis Algoritma	16
2.3 Kecerdasan Buatan / <i>Artificial Intelligence</i> (AI)	17
2.3.1 Sejarah <i>Artificial Intelligence</i> (AI)	19
2.3.2 Pemrograman Algoritma Konvensional dan Pemrograman AI	19
2.3.3 Lingkup <i>Artificial Intelligence</i> (AI) pada Aplikasi Komersial	20
2.4 <i>Game Playing</i>	22
2.5 <i>NP-Complete</i>	23
 Bab 3 Pembahasan dan Perancangan	25
3.1 Gambaran Umum <i>Game Minesweeper</i>	25



3.1.1 Aturan Permainan <i>Minesweeper</i>	27
3.1.2 Masalah <i>Minesweeper</i> Sebagai Masalah <i>NP-Complete</i>	30
3.2 Pembuatan Simulasi <i>Game Minesweeper</i>	30
3.2.1 <i>Flowchart</i> Pembuatan Simulasi <i>Game</i>	31
3.2.1.1 <i>Flowchart</i> SebarRanjau (Menentukan Posisi Ranjau)	31
3.2.1.2 <i>Flowchart</i> RandomRange (Menentukan Angka Random)	33
3.2.1.3 <i>Flowchart Depth First Search</i> untuk CariAreaBebasRanjau	34
3.2.1.3.1 <i>Flowchart</i> PeriksaStatus	35
3.2.1.3.2 Prosedur MasukkanTetangga	36
3.2.1.3.3 <i>Flowchart</i> CariAreaBebas	37
3.2.2 Representasi Pohon Pencarian <i>Depth First Search</i>	38
3.3 Analisis Penyelesaian <i>Game</i> dalam Kotak Seleksi	39
3.3.1 Ruang Keadaan	39
3.3.2 Analisa Pembentukan Ruang Keadaan	41
3.3.3 Aturan Penentuan Heuristik dalam Ruang Keadaan	43
3.3.4 Algoritma <i>Greedy Best First Search</i> untuk Pencarian Petak Aman dan Petak Ranjau	46
3.3.5 Konsep Dasar Penyelesaian <i>Game Minesweeper</i>	48
3.4 Analisis Kompleksitas (Waktu Eksekusi) Algoritma <i>Greedy Best First Search</i>	55
3.5 Perancangan Program Aplikasi	59
3.5.1 Perancangan <i>Data Flow Diagram</i>	61
3.5.2 Perancangan Antarmuka	64
Bab 4 Implementasi dan Pengujian	67
4.1 Implementasi Sistem	67
4.2 Spesifikasi Perangkat Lunak	67
4.3 Spesifikasi Perangkat Keras	67
4.4 Tampilan Aplikasi	68
4.4.1 Tampilan Utama	68
4.4.2 Tampilan <i>Menu</i>	69
4.4.3 Tampilan <i>Custom</i>	69
4.4.4 Tampilan <i>Input Nama</i>	70
4.4.5 Tampilan <i>Best Times</i>	70
4.4.6 Tampilan <i>How To Play</i>	71
4.4.7 Tampilan <i>About</i> Kotak Seleksi	71
4.4.8 Tampilan <i>About The Analyser</i>	72
4.5 Pengujian Penyelesaian <i>Greedy Best First Search</i> dalam Kotak Seleksi	73
4.5.1 Pengujian Ketika Ditemukan Solusi Optimal	74
4.5.2 Pengujian Ketika Tidak Ditemukan Solusi Optimal	76

4.5.3 Pengujian Ketika Tidak Ditemukan Solusi Apapun	78
4.6 Pengujian Penggunaan Aplikasi dan Fasilitas Kotak Seleksi Oleh <i>User</i>	78
Bab 5 Kesimpulan dan Saran	82
5.1 Kesimpulan	82
5.2 Saran	83
Daftar Pustaka	84
Lampiran : <i>Listing</i> Program	85

**DAFTAR TABEL**

	Halaman
Tabel 2.1 Beberapa Fungsi Heuristik Sederhana	11
Tabel 2.2 Pemrograman AI Vs. Pemrograman Konvensional	20
Tabel 3.1 Ketentuan Umum <i>Game Minesweeper</i>	31
Tabel 3.2 Aturan Penentuan Heuristik pada <i>Game Minesweeper</i>	43
Tabel 3.3 <i>Pseudocode</i> Algoritma <i>Greedy Best First Search</i>	55
Tabel 3.4 Spesifikasi proses diagram alir level 0	62
Tabel 3.5 Spesifikasi proses diagram alir level 1	63
Tabel 4.1 Hasil Pengujian Oleh <i>User</i> pada <i>Level Beginner</i>	79
Tabel 4.2 Hasil Pengujian Oleh <i>User</i> pada <i>Level Intermediate</i>	79

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Pohon Pencarian Mendalam Pertama ( <i>Depth First Search</i> )	7
Gambar 2.2 Ilustrasi Algoritma <i>Best First Search</i>	14
Gambar 2.3 Penerapan Konsep Kecerdasan Buatan di Komputer	18
Gambar 3.1 Papan Permainan <i>Minesweeper</i>	25
Gambar 3.2 Ruang Keadaan <i>User</i> Terkena Ranjau	26
Gambar 3.3 Ruang Keadaan <i>User</i> Mengklik Petak Bebas Ranjau	26
Gambar 3.4 Ruang Keadaan <i>User</i> Mengklik Petak Sekitar Ranjau	27
Gambar 3.5 Ruang Keadaan Akhir Dari Permainan	29
Gambar 3.6 <i>Flowchart</i> Menentukan Posisi Ranjau	32
Gambar 3.7 <i>Flowchart</i> Fungsi <i>RandomRange</i>	33
Gambar 3.8 <i>Flowchart</i> <i>Depth First Search</i> untuk <i>CariAreaBebasRanjau</i>	34
Gambar 3.9 <i>Flowchart</i> Sub Prosedur <i>PeriksaStatus</i>	35
Gambar 3.10 <i>Flowchart</i> <i>CariAreaBebas</i>	37
Gambar 3.11 Ilustrasi Pohon Pencarian <i>Depth First Search</i>	38
Gambar 3.12 Posisi - Posisi Kotak Seleksi Pada Papan <i>Game Minesweeper</i>	40
Gambar 3.13 Analisa Pembentukan Ruang Keadaan	42
Gambar 3.14 <i>Flowchart</i> Heuristik Pencarian Petak Ranjau dan Petak Aman	45
Gambar 3.15 <i>Flowchart</i> Algoritma <i>Greedy Best First Search</i>	47
Gambar 3.16 Konsep Dasar Penyelesaian pada $pRow = 1$ dan $pCol = 1$	48
Gambar 3.17 Konsep Dasar Penyelesaian pada $pRow = 1$ dan $pCol = nCols - 3$	49
Gambar 3.18 Konsep Dasar Penyelesaian pada $pRow = nRows - 3$ dan $pCol = 1$	50
Gambar 3.19 Konsep Dasar Penyelesaian pada $pRow = nRows - 3$ dan $pCol = nCols - 3$	51
Gambar 3.20 Konsep Dasar Penyelesaian pada $pRow = 1$ dan $1 < pCol < nCols - 3$	51
Gambar 3.21 Konsep Dasar Penyelesaian pada $pRow = nRows - 3$ dan $1 < pCol < nCols - 3$	52
Gambar 3.22 Konsep Dasar Penyelesaian pada $1 < pRow < nRows - 3$ dan $pCol = 1$	53
Gambar 3.23 Konsep Dasar Penyelesaian pada $1 < pRow < nRows - 3$ dan $pCol = nCols - 3$	54
Gambar 3.24 Konsep Dasar Penyelesaian pada $1 < pRow < nRows - 3$ dan $1 < pCol < nCols - 3$	55

Gambar 3.25 Diagram alir level 0	61
Gambar 3.26 Diagram alir level 1	62
Gambar 3.27 Rancangan <i>Form</i> Utama	65
Gambar 3.28 Tampilan <i>Form Custom</i>	65
Gambar 3.29 Tampilan <i>Form Input</i> Nama	66
Gambar 3.30 Tampilan <i>Form Best Times</i>	66
Gambar 4.1 Tampilan Utama Aplikasi <i>Level Beginner</i>	68
Gambar 4.2 <i>Menu</i> Aplikasi	69
Gambar 4.3 Tampilan <i>Custom</i>	69
Gambar 4.4 Tampilan <i>Input</i> Nama	70
Gambar 4.5 Tampilan <i>Best Times</i>	70
Gambar 4.6 Tampilan <i>How To Play</i>	71
Gambar 4.7 Tampilan <i>About</i> Kotak Seleksi	72
Gambar 4.8 Tampilan <i>About The Analyser</i>	73
Gambar 4.9 Tampilan Ketika Ditemukan Solusi Optimal	74
Gambar 4.10 Ilustrasi Pencarian <i>Greedy Best First Search</i> Ketika Solusi Optimal	75
Gambar 4.11 Tampilan Ketika Tidak Ditemukan Solusi Optimal	76
Gambar 4.12 Ilustrasi Pencarian <i>Greedy BestFS</i> Ketika Solusi Tidak Optimal	77
Gambar 4.13 Pengujian Ketika Tidak Ditemukan Solusi Apapun	78
Gambar 4.14 Grafik Hasil Pengujian <i>Level Beginner</i>	80
Gambar 4.15 Grafik Hasil Pengujian <i>Level Intermediate</i>	80