

***PATH FINDING SEARCH PADA GAME MAZE DINAMIS
MENGUNAKAN ALGORITMA FLOOD FILL***

Skripsi



oleh
HENRICHI AMARAL DA COSTA
22104925

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI
INFORMASI UNIVERSITAS KRISTEN DUTA WACANA
2017

PATH FINDING SEARCH PADA GAME MAZE INAMIS MENGUNAKAN ALGORITMA FLOOD FILL

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana
Sebagai Salah Satu Syarat dalam Memperoleh Gelar
Sarjana Komputer

Disusun oleh

HENRICHI AMARAL DA COSTA
22104925

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI
INFORMASI UNIVERSITAS KRISTEN DUTA WACANA 2017**

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

**PATH FINDING SEARCH PADA GAME MAZE DINAMIS
MENGUNAKAN ALGORITMA FLOOD FILL**

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Informatika Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di 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, 21 Desember 2017



HENRICHI AMARAL DA COSTA
22104925


HALAMAN PERSETUJUAN

Judul Skripsi : PATH FINDING SEARCH PADA GAME MAZE
DINAMIS MENGGUNAKAN ALGORITMA
FLOOD FILL

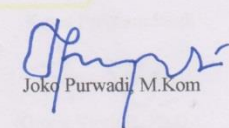
Nama Mahasiswa : HENRICHI AMARAL DA COSTA
N I M : 22104925
Matakuliah : Skripsi (Tugas Akhir)
Kode : TIW276
Semester : Gasal
Tahun Akademik : 2017/2018

Telah diperiksa dan disetujui di
Yogyakarta,
Pada tanggal 21 Desember 2017

Dosen Pembimbing I


R. Gunawan Santosa, Drs. M.Si.

Dosen Pembimbing II


Joko Purwadi, M.Kom

HALAMAN PENGESAHAN

PATH FINDING SEARCH PADA GAME MAZE DINAMIS
MENGUNAKAN ALGORITMA FLOOD FILL

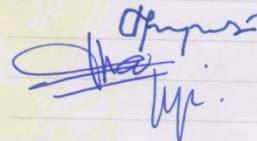
Oleh: HENRICHI AMARAL DA COSTA / 22104925

Dipertahankan di depan Dewan Penguji Skripsi
Program Studi Informatika Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana - Yogyakarta
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Komputer
pada tanggal 18 Desember 2017

Yogyakarta, 21 Desember 2017
Mengesahkan,

Dewan Penguji:


1. R. Gunawan Santosa, Drs. M.Si.
2. Joko Purwadi, M.Kom
3. Yuan Lukito, S.Kom., M.Cs.
4. Rosa Delima, S.Kom., M.Kom.



Dekan


(Budi Susanto, S.Kom., M.T.)

Ketua Program Studi


(Gloria Virginia, Ph.D.)

UCAPAN TERIMAKASIH

Pertama-tama penulis ucapkan syukur kepada Tuhan Yang Maha Esa karena dengan berkat dan karunia-Nya skripsi ini dapat diselesaikan. Skripsi yang berjudul “*Path Finding Search* pada *Maze* Dinamis Menggunakan Algoritma *Flood Fill*” merupakan salah satu syarat untuk mendapatkan gelar Sarjana Komputer. Diselesaikannya skripsi ini tidak lepas dari partisipasi dan bantuan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan rasa terima kasih yang setulus-tulusnya kepada :

1. Bapak Drs. R. Gunawan Santosa, M.Si dan Joko Purwadi, M.Kom selaku dosen pembimbing yang selalu gigih dan sabar dalam memberikan arahan.
2. Kepada seluruh anggota keluarga yang saya cintai, Bapak, Ibu, Adik, dan saudara-saudara saya yang selalu memberikan dukungan baik berupa nasehat, motivasi dan doa dari kedua orang tua saya.
3. Seluruh staf pengajar dan pegawai Fakultas Teknologi Informasi UKDW atas keramahan dan ilmu yang telah diberikan kepada saya.
4. Kepada sahabat-sahabat saya di KELAS GOKIL UKDW : Max, Ian Jacob , Aan Setiawan, Bharep Pramono, Yoshua Hendra, Damasus Bagus, Roby Chandra, Stanley Dethan, Prima Noviarso, Rico, Mahendra Yadnya, Bastio dkk yang memberikan keceriaan dan kebersamaan.

Seluruh pihak yang ikut membantu namun tidak bisa dituliskan oleh penulis. Penulis mengucapkan terima kasih yang sebesar-besarnya.

DAFTAR ISI

PERYATAAN KEASLIAN SKRIPSI	iii
HALAMAN PRSETUJUAN	iv
HALAMAN PENGESAHAN	v
UCAPAN TERIMAKASIH.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
INTISARI	xiii
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Metode Penelitian.....	2
1.6 Sistematika Penulisan.....	3
BAB 2	5
LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka	5
2.2.1 Kecerdasan Buatan	6
2.2.2 Konsep Sebuah <i>Maze</i>	6
2.2.3 Permainan <i>Maze</i>	7
2.2.4 <i>Maze Solving</i>	8
2.2.5 Algoritma <i>Flood Fill</i>	9
BAB 3	21
ANALISIS DAN PERANCANGAN SISTEM	21
3.1 Analisis Kebutuhan Sistem	21
3.1.1 Kebutuhan Fungsional	21
3.1.2 Spesifikasi Pembuatan Permainan	21

3.2 Rancangan Kerja Sistem	22
3.2.1 Diagram Alir (<i>Flowchart</i>).....	22
3.2.2 Diagram Alir Utama Sistem	22
3.2.3 Diagram Alir Algoritma <i>Flood Fill</i>	23
3.3 Rancangan Proses Kerja Permainan	24
3.3.1 Rancangan Input	24
3.3.2 Penerapan Algoritma	25
3.4 Rancangan Antara Muka Permainan	25
3.4.1 Tampilan Menu Utama Permainan.....	25
3.4.2 Tampilan Menu Utama Permainan.....	26
BAB 4	27
IMPLEMENTASI DAN ANALISIS SISTEM.....	27
4.1 Implementasi Sistem	27
4.1.1 Antarmuka Sistem	27
4.1.2 Tampilan Halaman <i>Maze</i>	28
4.1.3 Tampilan Halaman <i>Set Route</i>	29
4.1.4 Tampilan Halaman <i>About</i>	29
4.1.5 Tampilan Halaman <i>Help</i>	30
4.2 Implementasi Algoritma Program.....	31
4.2.1 Fungsi Algoritma <i>Flood Fill</i>	31
4.2.2 Fungsi Side Winder	32
4.2.3 Fungsi Reset Tiles	33
4.2.4 Fungsi Clear Path.....	34
4.3 Analisis Sistem	34
4.3.1 Data Percobaan	34
4.3.2 Analisa <i>Node</i> (Jumlah <i>Node</i> yang Dilalui)	39
4.3.3 Analisa Algoritma <i>Flood Fill</i>	40
BAB 5	41
KESIMPULAN.....	41
5.1 Kesimpulan.....	41
5.2 Saran	41

DAFTAR PUSTAKA	42
LAMPIRAN	43

©UKDW

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi Struktur Maze.....	6
Gambar 2. 2 Maze tidak Sempurna.....	7
Gambar 2. 3 Maze Sempurna	7
Gambar 2. 4 Tower Of Druaga	8
Gambar 2. 5 Mummy Maze	8
Gambar 2. 6 Air yang mengalir dalam maze	9
Gambar 2. 7 Pencarian Jalur	9
Gambar 2. 8 Aturan Pemberian bobot Nilai	10
Gambar 2. 9 Aturan Pemberian bobot Nilai	11
Gambar 2. 10 Aturan Pemberian bobot Nilai	11
Gambar 2. 11 Pemberian bobot Nilai pada Maze	12
Gambar 2. 12 Pemberian bobot Nilai pada Maze	12
Gambar 2. 13 Pemberian bobot Nilai pada Maze	12
Gambar 2. 14 Pemberian bobot Nilai pada Maze	13
Gambar 2. 15 Contoh Player bergerak dari titik mulai menuju titik tujuan.....	14
Gambar 2. 16 Cell tetangga dengan bobot nilai terkecil yang terhalang dinding	14
Gambar 2. 17 Nilai cell yang tela di Update.....	15
Gambar 2. 18 Pemberian Nilai Awal.....	16
Gambar 2. 19 Pemberian Bobot Nilai Tahap 2.....	16
Gambar 2. 20 Pemberian Bobot Nilai Tahap 3.....	16
Gambar 2. 21 Pemberian Bobot Nilai Tahap 4 dan 5	17
Gambar 2. 22 Pemberian Bobot Nilai Tahap 6 dan 7	17
Gambar 2. 23 Pemberian Bobot Nilai Tahap 8 dan 9.....	17
Gambar 2. 24 Pemberian Bobot Nilai Tahap 10 dan 11	18
Gambar 2. 25 Pemberian Bobot Nilai Tahap 12 dan 13	18
Gambar 2. 26 Pemberian Bobot Nilai Tahap 14 dan 15	18
Gambar 2. 27 Pemberian Bobot Nilai Tahap 16 dan 17	19
Gambar 2. 28 Pemberian Bobot Nilai Tahap 18 dan 19	19
Gambar 2. 29 Pemberian Bobot Nilai Tahap 20 dan 21	19
Gambar 2. 30 Proses Pencarian Jalan Keluar	20

Gambar 3. 1 Flowchart Diagram Sistem Utama	23
Gambar 3. 2 Flowchart Diagram alier Flood Filll	24
Gambar 3. 3 Tampilan Halaman Utama	25
Gambar 3. 4 Tampilan Halaman Maze	26
Gambar 3. 5 Tampilan Solve Maze	26
Gambar 4. 1 Halaman Utama	27
Gambar 4. 2 Menu	28
Gambar 4. 3 Proses Generate Maze	29
Gambar 4. 4 Halaman Solve	29
Gambar 4. 5 Halaman About	30
Gambar 4. 6 Halaman Help	30
Gambar 4. 7 Ukuran Maze yang Dinamis	35
Gambar 4. 8 Grafik Perbandingan Rata-Rata Nodes pada Maze Persegi panjang Vertikal Dan Maze Persegi Panjang Horizontal.....	40

© UTKDN

DAFTAR TABEL

Tabel 4.1 (a) Tabel Maze Ukuran Persegi	35
Tabel 4. 1 (b) Tabel Maze Ukuran Persegi	36
Tabel 4. 2 (a) Tabel Persegi panjang Vertikal	37
Tabel 4. 2 (b) Tabel Persegi panjang Vertikal	37
Tabel 4. 3 (a) Tabel Persegi panjang Horizontal	38
Tabel 4. 3 (b) Tabel Persegi panjang Horizontal	38

©UKDW

INTISARI

PATH FINDING SEARCH PADA GAME MAZE DINAMIS MENGGUNAKAN ALGORITMA FLOOD FILL

Salah satu aspek dalam sebuah game yang mampu memberikan tingkat kesulitan berbeda-beda adalah desain *Map*. *Gameplay* sebuah *Game* yang bergantung pada desain *Map* adalah game bertema *Maze*. Membuat sebuah map labirin bukan hal yang susah, akan tetapi menjadi hal yang berbeda ketika membuat beberapa *map* labirin yang berbeda untuk setiap game stage. Programmer sering merasa kesulitan ketika membuat *map* yang berbeda-beda untuk setiap game *stage*, karena harus menata ulang map labirin dan membuat secara manual, tetapi jika jalur map labirin berbeda maka penyelesaian untuk solusi pencarian jalan keluar juga akan berbeda oleh karena itu peneliti menggunakan algoritma *flood fill* untuk digunakan sebagai solusi pencarian jalan keluar.

Dalam penelitian ini, penulis membuat sebuah sistem *Game Maze* dengan mengimplementasikan algoritma *Flood Fill* sebagai *Solve* dalam *maze* dinamis. Penulis menganalisis perbandingan *Nodes* pada *maze* persegi panjang Vertikal dan *meze* Persegi panjang Horizontal.

Sistem yang dibangun 100% mampu menghasilkan *Maze* secara dinamis, sistem juga mampu mencari jalan keluar dengan menggunakan algoritma *flood fill* dari *Maze* yang dibuat. Bentuk *Maze* mempengaruhi kompleksitas *Maze*.

Kata Kunci : *Maze, Solver, Algoritma Flood Fill*

INTISARI

PATH FINDING SEARCH PADA GAME MAZE DINAMIS MENGGUNAKAN ALGORITMA FLOOD FILL

Salah satu aspek dalam sebuah game yang mampu memberikan tingkat kesulitan berbeda-beda adalah desain *Map*. *Gameplay* sebuah *Game* yang bergantung pada desain *Map* adalah game bertema *Maze*. Membuat sebuah map labirin bukan hal yang susah, akan tetapi menjadi hal yang berbeda ketika membuat beberapa *map* labirin yang berbeda untuk setiap game stage. Programmer sering merasa kesulitan ketika membuat *map* yang berbeda-beda untuk setiap game *stage*, karena harus menata ulang map labirin dan membuat secara manual, tetapi jika jalur map labirin berbeda maka penyelesaian untuk solusi pencarian jalan keluar juga akan berbeda oleh karena itu peneliti menggunakan algoritma *flood fill* untuk digunakan sebagai solusi pencarian jalan keluar.

Dalam penelitian ini, penulis membuat sebuah sistem *Game Maze* dengan mengimplementasikan algoritma *Flood Fill* sebagai *Solve* dalam *maze* dinamis. Penulis menganalisis perbandingan *Nodes* pada *maze* persegi panjang Vertikal dan *meze* Persegi panjang Horizontal.

Sistem yang dibangun 100% mampu menghasilkan *Maze* secara dinamis, sistem juga mampu mencari jalan keluar dengan menggunakan algoritma *flood fill* dari *Maze* yang dibuat. Bentuk *Maze* mempengaruhi kompleksitas *Maze*.

Kata Kunci : *Maze*, *Solver*, Algoritma *Flood Fill*

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi yang semakin berkembang khususnya pada media elektronik komputer. Ada begitu banyak hal yang dikerjakan menggunakan media elektronik ini, salah satunya sebagai media hiburan. Salah satu hiburan yang disajikan dalam komputer berupa permainan. Ada begitu banyak kalangan dalam masyarakat yang suka dengan hiburan seperti ini mulai dari anak-anak hingga orang dewasa. Ada begitu banyak jenis game yang di sukai oleh kalangan masyarakat antara lain yaitu : permainan *Puzzle*, *Role Playing Game*, *Adventure* dan masih banyak jenis permainan lainnya. Salah satu jenis permainan game yang disukai yaitu permainan *Puzzle*, yaitu permainan game labiri. Labirin adalah sebuah permainan yang mencari jalan keluar di mana dalam labirin ada banyak rintangan.

Membuat sebuah *map* labirin bukanlah hal yang susah, akan tetapi menjadi hal yang berbeda ketika membuat beberapa Map yang berbeda untuk setiap game *stage*. Programmer sering merasa kesulitan ketika membuat *Map* yang berbeda-beda untuk setiap *stage*, karena harus menata ulang *Map* dan membuat secara manual.

Permasalahan di atas, disarankan untuk mempermudah programmer dalam pembuatan game berbentuk labirin. Programmer tidak perlu lagi membuat labirin secara manual karena itu merepotkan, tetapi cukup dengan menginputkan *source code* dan kemudian secara otomatis algoritma itu sendiri yang akan membuat map. Bahkan programmer tidak mengetahui desain labirin yang akan muncul dalam permainan. Dari *map* labirin yang dibuat belum tentu juga programmer dapat mengetahui jalan keluar sebuah labirin dengan cepat sehingga diperlukan algoritma yang dapat digunakan untuk mencari jalan keluar dari labirin yang dibentuk secara acak.

Pada permasalahan diatas, penulis ingin menggunakan algoritma *Flood Fill* dan mengimplementasikan algoritma tersebut sebagai *solve* pencarian jalan keluar pada labirin.

1.2 Perumusan Masalah

Berdasarkan latar belakang masalah diatas, rumusan masalah dalam penelitian ini yaitu apakah algoritma *Flood Fill* dapat diimplementasikan dalam *path finding search* untuk mencari jalan keluar pada permainan labirin.

1.3 Batasan Masalah

Adapun batasan sistem yang akan dibuat dalam penelitian ini adalah sebagai berikut :

- a. Ukuran maksimal map yang diuji adalah 30x30.
- b. Hanya terdapat 1 titik awal (*start*) dan 1 titik akhir (*finish*).
- c. Titik mulai (*start*) berada pada kotak (*cell*) pertama yaitu pojok kiri atas.
- d. Posisi titik akhir (*finish*) berada pada kotak (*cell*) akhir yaitu di pojok kanan bawah.
- e. Pengaturan kotak jalur dilakukan secara dinamis.

1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai oleh peneliti adalah sebagai berikut :

Mengimplementasikan algoritma *flood fill* sebagai solusi pencarian jalan keluar pada *maze*, algoritma *flood fill* dapat digunakan untuk mengetahui tingkat ketepatan dan presisi dalam pencarian jalan keluar pada *maze*. Menghasilkan aplikasi game *maze* dan menggunakan algoritme *flood fill* untuk menyelesaikan masalah-masalah dalam *path finding search* dalam sebuah game *maze*.

1.5 Metode Penelitian

Metode penelitian yang akan digunakan adalah :

a. Studi pustaka Pada bagian studi Pustaka dilakukan dengan membaca referensi-referensi yang berisi berbagai informasi yang berkaitan dengan *maze* dan algoritma *flood fill* dari berbagai macam sumber seperti buku, artikel, jurnal dan sumber lainnya.

b. Pembuatan Sistem

Pada bagian tahapan sistem ini akan berisi tampilan atau antarmuka sistem dan mengimplementasikan algoritma *flood fill* kedalam Bahasa pemrograman, Bahasa pemrograman yang digunakan adalah HTML5, JavaScript dan untuk mempercantik tampilan digunakan CSS.

c. Pengujian dan Analisis

Dalam pengujian system ini akan dilakukan pengujian algoritma *flood fill*, pengujian dilakukan dengan cara melakukan percobaan berkali-kali dalam sebuah permainan pencarian jalan keluar dengan bentuk *maze* yang berubah-ubah..

d. Analisis Hasil Percobaan dan Evaluasi

Pada analisis hasil percobaan ini akan dilakukan analisis pengujian yang telah dilakukan sebelumnya. Hasil analisis akan menghasilkan dan mempersentasikan berhasil atau tidaknya algoritma *flood fill* dalam menyelesaikan masalah pencarian jalan keluar

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini dikelompokkan menjadi 5 bab, yaitu :

Bab 1 PENDAHULUAN, BAB ini berisi gambaran umum tentang penelitian yang dilakukan, dimana mencakup antara lain latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan penelitian.

Bab 2 LANDASAN TEORI, BAB ini berisi tentang tinjauan pustaka dan landasan teori yang digunakan dalam penelitian ini. Tinjauan pustaka menguraikan berbagai teori yang didapat dari berbagai sumber terkait dengan penelitian ini.

Bab 3 PERANCANGAN SISTEM , BAB ini berisi tentang rancangan sistem yang dibangun dalam penelitian ini. Rancangansistem yang dibuat berupa spesifikasi dari system, rancangan antarmuka sistem berupa input dan output..

Bab 4 IMPLEMENTASI SISTEM, BAB ini berisi tentang implementasi system yang telah dirancang sebagaimana sudah dirancang pada bab III.

Bab 5 KESIMPULAN DAN SARAN, BAB ini berisi tentang kesimpulan hasil penelitian yang telah dilakukan dan saran yang mungkin dapat dilakukan untuk pengembangan penelitain selanjutnya.

©UKDW

BAB 5

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan analisis sistem, maka dapat disimpulkan sebagai berikut :

- a. Algoritma *Flood Fill* terbukti 100% dapat menjadi solusi pencarian jalan keluar pada sebuah *maze* dinamis.
- b. Bentuk *maze* dinamis yang kompleksitasnya tinggi dapat diselesaikan dengan menggunakan algoritma *Flood Fill*.
- c. Bentuk *maze* dapat mempengaruhi waktu penyelesaian.
- d. Dari gambar 4.8 terlihat bahwa *nodes* $n \times 5$ memiliki rata-rata banyak *nodes* lebih sedikit untuk dilewati dibandingkan dengan *nodes* $5 \times n$ karena berdasarkan pada random proses jalur yang dibuat.

5.2 Saran

Saran yang diberikan penulis untuk pengembangan sistem selanjutnya adalah sebagai berikut :

- a. Algoritma *Flood Fill* dapat diimplementasikan langsung kedalam sebuah game sebagai solusi pencarian jalan keluar.
- b. Dapat dilakukan penelitian lebih lanjut perbandingan antara algoritma *Flood Fill* dengan algoritma lain.

DAFTAR PUSTAKA

- A Rahman, A. H. (2010). Penerapan Algoritma Flood Fill untuk Menyelesaikan Maze pada line Follower Robot.
- Bekti, S. H. (2009). Pencarian Shortest Path Dinamik dengan Algoritma Bellman-Base Flood Fill dan Implementasinya pada Robot Micromouse.
- Lee, H. L. (2010). A Perfect Maze based steganographic method. *The Journal Of Systems and Software*, 2528-2535.
- Namco. (2016, January 3). *Tower of Druaga*. Dipetik Oktober 5, 2016, dari UVL: <http://www.uvlist.net>
- Sasongko, A. C. (2008). Penerapan Metode Recursive Backtracker Sebagai Creator Dan Solver Dalam Game Maze.
- Sebatio, M. A. (2016). Implementasi Algoritma Sidewinder sebagai Generator Maze. Yogyakarta, Yogyakarta, Indonesia.
- Setiawan, H. C. (2012, September 26). *Pengantar Kecerdasan Buatan*. Dipetik November 7, 2016, dari Hanifahyas: <http://www.hanifahyas.blogspot.co.id/2012/09/pengantar-kecerdasan-buatan-atau.html>
- Teams. (2013, November 2). *Download Games Mummy Maze Deluxe*. Dipetik oktober 5, 2016, dari Cheat Afrix: <http://www.cheat-afrix.com/2013/10/download-games-mummy-maze-deluxe-full.html>
- Utomo, A. S. (2016). Perbandingan Algoritma Flood Fill dan Dijkstra's pada Maze Mapping untuk Robot Line Follower. 124-128.