

**IMPLEMENTASI ALGORITMA *SIDEWINDER* SEBAGAI  
*GENERATOR MAZE***

Skripsi



oleh  
**SEBASTIO AVIANT MATUTINA**  
**22104985**

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
2016

**IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI  
GENERATOR MAZE.**

Skripsi



oleh  
**SEBASTIO AVIANT MATUTINA**  
22104985

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
2016

**IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI  
GENERATOR MAZE.**

Skripsi



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

Disusun oleh

**SEBASTIO AVIANT MATUTINA**  
**22104985**

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
2016

**PERNYATAAN KEASLIAN SKRIPSI**

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

**IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI GENERATOR MAZE.**

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 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, 20 Desember 2016




**SEBASTIO AVIANT MATUTINA**  
22104985

## HALAMAN PERSETUJUAN

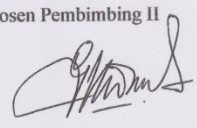
Judul Skripsi : IMPLEMENTASI ALGORITMA SIDEWINDER  
SEBAGAI GENERATOR MAZE.  
Nama Mahasiswa : SEBASTIO AVIANT MATUTINA  
N I M : 22104985  
Matakuliah : Skripsi (Tugas Akhir)  
Kode : TIW276  
Semester : Gasal  
Tahun Akademik : 2016/2017

Telah diperiksa dan disetujui di  
Yogyakarta,  
Pada tanggal 20 Desember 2016

Dosen Pembimbing I

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

Dosen Pembimbing II

  
R. Gunawan Santosa, Drs. M.Si.



**HALAMAN PENGESAHAN**

**IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI  
GENERATOR MAZE.**

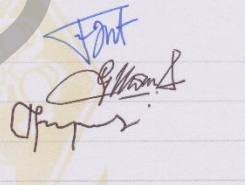
Oleh: SEBASTIO AVIANT MATUTINA / 22104985

Dipertahankan di depan Dewan Penguji 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 15 Desember 2016

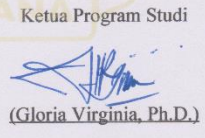
Yogyakarta, 20 Desember 2016  
Mengesahkan,

Dewan Penguji:

1. Antonius Rachmat C., S.Kom.,M.Cs.
2. R. Gunawan Santosa, Drs. M.Si.
3. Joko Purwadi, M.Kom
- 4.



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 “Implementasi Algoritma *Sidewinder* Sebagai *Generator Maze*” 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 Antonius R. C., S.Kom., M.Cs dan Drs. R. Gunawan Santosa, M.Si 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 Agnes Dita yang selalu ada disetiap saat.
5. 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, Richi Costa dkk yang memberikan keceriaan dan kebersamaan.

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

## INTISARI

### IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI GENERATOR MAZE

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*. Untuk membuat *Maze* itu mudah jika dibuat secara manual, namun untuk membuat *Maze* secara cepat, acak dan otomatis dibutuhkan tambahan algoritma tertentu yang memang digunakan sebagai pembuat *Maze*.

Dalam penelitian ini, penulis membuat sebuah sistem *Generator Maze* dengan mengimplementasikan algoritma *Sidewinder* sebagai pembuat *Maze*. Penulis menganalisis *Traversed Nodes* dan *Path Nodes* pada *Maze* dari hasil *Solver* menggunakan algoritma *Recursive Backtracker*. Dari *Nodes* tersebut *Maze* yang dibuat dengan algoritma *Sidewinder* dapat diuji kompleksitasnya.

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

Kata Kunci : *Maze*, *Solver*, Algoritma *Sidewinder*, Algoritma *Recursive Backtracker*



## DAFTAR ISI

PERNYATAAN KEASLIAN SKRIPSI.....	iv
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN .....	iv
UCAPAN TERIMAKASIH.....	iv
INTISARI .....	vii
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
DAFTAR GRAFIK.....	xiv
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.....	3
1.6 Sistematika Penulisan.....	3
BAB 2 .....	5
LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka .....	5
2.2 Landasan Teori .....	6
2.2.1 Konsep Sebuah Maze .....	6
2.3 Maze Solving.....	14
2.3.1 Algoritma Recursive Backtracker.....	14
BAB 3 .....	22
ANALISIS DAN PERANCANGAN SISTEM .....	22
3.1 Analisis Kebutuhan Sistem .....	22
3.1.1 Kebutuhan Fungsional .....	22
3.1.2 Kebutuhan Non Fungsional .....	22

3.2 Rancangan Kerja Sistem .....	23
3.2.1 Diagram Alir(Flowchart) .....	23
3.2.2 Diagram Alir Utama Sistem .....	24
3.2.3 Diagram Alir Algoritma <i>Sidewinder</i> .....	25
3.2.4 Diagram Alir Algoritma <i>Recursive Backtracker</i> .....	26
3.2.5 Algoritma Sistem .....	27
3.3 Rancangan Kerja Sistem .....	27
3.3.1 Perancangan Input .....	27
3.3.2 Penerapan Algoritma .....	27
3.4 Perancangan Struktur Data .....	28
3.5 Rancangan Antar Muka Sistem .....	30
3.5.1 Tampilan Awal .....	30
3.5.2 Tampilan Generate Maze .....	30
3.5.3 Tampilan Acak Pintu keluar .....	32
3.5.4 Tampilan Solve Maze .....	32
3.6 Perancangan Pengujian sistem .....	33
BAB 4 .....	35
IMPLEMENTASI DAN ANALISIS SISTEM .....	35
4.1 Implementasi Sistem .....	35
4.1.1 Antarmuka sistem .....	35
4.1.2 Tampilan Halaman Map Generate .....	36
4.1.3 Tampilan Halaman <i>Set Route</i> .....	37
4.1.4 Tampilan Halaman <i>Solver</i> .....	38
4.1.5 Tampilan Halaman Tree .....	39
4.2 Analisis Sistem .....	41
4.2.1 Data Percobaan .....	42
4.2.2 Analisis Traversed Nodes .....	46
4.2.3 Analisis Path Nodes .....	47
4.2.4 Analisis Algoritma <i>Sidewinder</i> .....	47
BAB 5 .....	49

KESIMPULAN.....	49
5.1 Kesimpulan.....	49
5.2 Saran.....	49
DAFTAR PUSTAKA .....	50
LAMPIRAN.....	51

©UKDW

## DAFTAR GAMBAR

Gambar 2.1: Ilustrasi Struktur <i>Maze</i> .....	6
Gambar 2.2: <i>Maze</i> Tidak Sempurna .....	7
Gambar 2.3: <i>Maze</i> Sempurna.....	7
Gambar 2.4 : <i>Step 1 Sidewinder Algorithm</i> .....	8
Gambar 2.5 : <i>Step 2 Sidewinder Algorithm</i> .....	8
Gambar 2.6 : <i>Step 3 Sidewinder Algorithm</i> .....	9
Gambar 2.7 : <i>Step 4 Sidewinder Algorithm</i> .....	9
Gambar 2.8 : <i>Step 5 Sidewinder Algorithm</i> .....	9
Gambar 2.9 : <i>Step 6 Sidewinder Algorithm</i> .....	10
Gambar 2.10 : <i>Step 7 Sidewinder Algorithm</i> .....	10
Gambar 2.11 : <i>Step 8 Sidewinder Algorithm</i> .....	10
Gambar 2.12 : <i>Step 9 Sidewinder Algorithm</i> .....	11
Gambar 2.13 : <i>Step 10 Sidewinder Algorithm</i> .....	11
Gambar 2.14 : <i>Step 11 Sidewinder Algorithm</i> .....	11
Gambar 2.15 : <i>Step 12 Sidewinder Algorithm</i> .....	12
Gambar 2.16 : <i>Step 13 Sidewinder Algorithm</i> .....	12
Gambar 2.17 : <i>Step 14 Sidewinder Algorithm</i> .....	12
Gambar 2.18 : <i>Step 15 Sidewinder Algorithm</i> .....	13
Gambar 2.19 : <i>Step 16 Sidewinder Algorithm</i> .....	13
Gambar 2.20 : <i>Step 17 Sidewinder Algorithm</i> .....	13
Gambar 2.21 : <i>Pseudocode Recursive Backtacker</i> .....	14
Gambar 2.22 : Contoh Pohon Ruang Status .....	15
Gambar 2.23 : Contoh <i>Maze Solver</i> .....	16
Gambar 2.24 : Contoh <i>Maze Solver</i> .....	16
Gambar 2.25 : Contoh <i>Maze Solver</i> .....	17
Gambar 2.26 : Contoh <i>Maze Solver</i> .....	17
Gambar 2.27 : Contoh <i>Maze Solver</i> .....	18
Gambar 2.28 : Contoh <i>Maze Solver</i> .....	19

Gambar 2.29 : Contoh <i>Maze Solver</i> .....	20
Gambar 3.1 : Alir Utama Sistem.....	23
Gambar 3.2 : Diagram Alir Algoritma <i>Sidewinder</i> .....	24
Gambar 3.3 : Diagram Alir Algoritma <i>Recursive Backtracker</i> .....	25
Gambar 3.4 : Tampilan Halaman Utama .....	29
Gambar 3.5 : Tampilan Proses <i>Generate Maze</i> .....	30
Gambar 3.6 : Rancangan Halaman Menang .....	30
Gambar 3.7 : Tampilan Acak Pintu Keluar.....	31
Gambar 3.8 : Tampilan Proses <i>Solve Maze</i> .....	31
Gambar 3.9 : Tampilan Hasil Pencarian Jalan Keluar .....	32
Gambar 4.1 : Halaman Utama.....	33
Gambar 4.2 : Menu .....	34
Gambar 4.3 : Proses <i>Generator</i> .....	35
Gambar 4.4 : Halaman Pilih Rute.....	35
Gambar 4.5 : Proses <i>Solver</i> .....	36
Gambar 4.6 : Tampilan <i>Solver</i> .....	37
Gambar 4.7 : Hasil <i>Print Maze</i> .....	37
Gambar 4.8 : Hasil <i>Print Maze</i> .....	38
Gambar 4.9 : <i>Log</i> .....	39
Gambar 4.10 : <i>Tree</i> .....	39
Gambar 4.11 : Ukuran <i>Maze</i> Yang Dianalisis .....	41

## DAFTAR TABEL

Tabel 3.1 : Struktur Data.....	27
Tabel 3.2 : Daftar Fungsi .....	28
Tabel 4.1 : Tabel Hasil Analisis Waktu <i>Generate Maze</i> Berdasarkan Ukuran <i>Maze</i> Dengan Algoritma <i>Sidewinder</i> .....	41
Tabel 4.2(a) : Tabel <i>Maze</i> Ukuran Persegi .....	41
Tabel 4.2(b) : Tabel <i>Maze</i> Ukuran Persegi .....	42
Tabel 4.3 : Hasil Perbandingan Nilai <i>Traversed Node</i> Dan <i>Path Node</i> Pada <i>Maze</i> Berbentuk Persegi Panjang Vertical .....	42
Tabel 4.4 : Hasil Perbandingan Nilai <i>Traversed Node</i> Dan <i>Path Node</i> Pada <i>Maze</i> Berbentuk Persegi Panjang Horizontal .....	43



## DAFTAR GRAFIK

Grafik 4.3 : Hasil Analisis Perbandingan Rata-Rata <i>Traversed Nodes Maze</i> Persegi Panjang Vertikal Dan Persegi Panjang Horizontal .....	44
Grafik 4.4 : Hasil Analisis Perbandingan Rata-Rata <i>Path Nodes Maze</i> Persegi Panjang Vertikal Dan Persegi Panjang Horizontal .....	45

©UKDW

## INTISARI

### IMPLEMENTASI ALGORITMA SIDEWINDER SEBAGAI GENERATOR MAZE

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*. Untuk membuat *Maze* itu mudah jika dibuat secara manual, namun untuk membuat *Maze* secara cepat, acak dan otomatis dibutuhkan tambahan algoritma tertentu yang memang digunakan sebagai pembuat *Maze*.

Dalam penelitian ini, penulis membuat sebuah sistem *Generator Maze* dengan mengimplementasikan algoritma *Sidewinder* sebagai pembuat *Maze*. Penulis menganalisis *Traversed Nodes* dan *Path Nodes* pada *Maze* dari hasil *Solver* menggunakan algoritma *Recursive Backtracker*. Dari *Nodes* tersebut *Maze* yang dibuat dengan algoritma *Sidewinder* dapat diuji kompleksitasnya.

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

Kata Kunci : *Maze*, *Solver*, Algoritma *Sidewinder*, Algoritma *Recursive Backtracker*

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang Masalah

Di era teknologi komputer yang semakin berkembang, komputer tidak hanya digunakan untuk ke butuhan penghitungan tetapi dapat digunakan untuk kebutuhan multimedia dan hiburan yang lain. Salah satunya adalah kebutuhan komputer hiburan berupa game, pada saat ini banyak game yang diminati oleh semua kalangan mulai dari anak – anak sampai dengan dewasa. Setiap game memiliki genre antara lain *puzzle*, *role playing game(RPG)*, *adventure* , *horror*, *shooter*, *race*. Game bergenre *puzzle* merupakan game untuk mengasah otak, karena *puzzle* game dapat meningkatkan daya kreatifitas dan tingkat intelektual pemainnya. Salah satu jenis game bergenre *puzzle* adalah game labirin, tantangan pada game labirin adalah labirin itu sendiri. Semakin sulit sebuah *map* labirin maka semakin sulit mencari jalan keluarnya.

Membuat sebuah *map* labirin bukanlah hal yang susah, akan tetapi menjadi 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 (Kristiadi, 2015).

Dari permasalahan diatas, membuat *map* labirin yang memiliki bentuk yang berbeda-beda dapat dilakukan dengan menggunakan algoritma yang mampu membuat labirin secara acak tiap kali dihasilkan. *Programmer* tidak akan kesulitan dengan membuat *map* labirin yang berbeda-beda secara manual dengan menambahkan algoritma pembuat labirin kedalam program. Membuat *map* labirin secara otomatis dan acak tidak hanya mempermudah *programmer* saja tapi dapat juga memberikan pengalaman bermain yang berbeda-beda kepada *player* ketika *map* yang ditemui dalam suatu game tidak pernah sama, karena labirin yang dihasilkan selalu acak sehingga tidak menghasilkan bentuk labirin yang sama. Dari *map* labirin yang dibuat belum tentu

juga *programmer* dapat mengetahui jalan keluar sebuah labirin dengan cepat sehingga diperlukan juga algoritma yang dapat digunakan untuk mencari jalan keluar.

Dari permasalahan tersebut, penulis ingin mengimplementasikan algoritma *Sidewinder* sebagai *generator* pembuatan sebuah labirin dan menggunakan algoritma *Recursive Backtracker* sebagai *solver* jalan keluar dari labirin tersebut.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang masalah diatas, rumusan masalah dalam penelitian ini adalah:

- a. Apakah algoritma *Sidewinder* dapat diimplementasikan sebagai *generator maze* ?
- b. Apakah algoritma *Recursive Backtracker* dapat diimplementasikan sebagai *solver* sebuah *maze* ?
- c. Apakah bentuk *maze* dapat mempengaruhi nilai *traversed node* dan *path node* ?

## 1.3 Batasan Masalah

Dalam penelitian ini peneliti memberikan batasan masalah sebagai berikut :

- a. Ukuran maksimal map yang diuji adalah 30x30.
- b. Hanya terdapat 1 titik awal dan 1 titik akhir.
- c. Untuk analisis posisi *start* berada di pojok kiri atas.
- d. Untuk analisis posisi *end* berada pada pojok kanan bawah.

## 1.4 Tujuan Penelitian

Tujuan pembuatan tugas akhir ini adalah

- a. Mengimplementasikan algoritma *Sidewinder* untuk membuat sebuah *maze*.
- b. Mengimplementasikan algoritma *Recursive Backtracker* sebagai solusi pencarian jalan keluar sebuah *maze*.
- c. Mengukur tingkat ketepatan algoritma *Sidewinder* dan algoritma *Recursive Backtraker* sebagai pembuat *maze* dan sebagai solusi pencarian jalan keluar sebuah *maze*.

## 1.5 Metode Penelitian

Studi pustaka dilakukan dengan membaca referensi-referensi dan artikel maupun jurnal –jurnal yang berhubungan dengan salah yang dihadapi untuk menunjang perancangan dan pembuatan program serta penulisan tugas akhir

### a. Studi pustaka dan literature

Pada tahap ini dilakukan untuk memperoleh informasi dengan memanfaatkan berbagai macam sumber pustaka melalui buku, artikel, jurnal ilmiah, dan sumber lain yang berkaitan dengan *maze*, algoritma *Sidewinder*, dan algoritma *Recursive Backtracker*.

### b. Pembuatan Sistem

Pada tahap ini berisi perancangan antarmuka sistem dan mengimplementasikan algoritma *Sidewinder* dan algoritma *Recursive Backtracker* ke dalam bahasa program. Bahasa pemrograman yang digunakan adalah HTML5 dan javascript.

### c. Pengujian dan Analisis

Pada tahap ini dilakukan pengujian kedua algoritma yang telah digunakan dalam penelitian ini. Algoritma tidak akan diuji sekaligus, melainkan diuji satu persatu. Pengujian dilakukan dengan cara melakukan percobaan beberapa kali dalam *generator maze* dan *solver maze*.

### d. Analisis Hasil Percobaan dan Evaluasi

Tahapan ini berisi tentang analisis pengujian sistem yang telah dilakukan sebelumnya. Hasil akhir analisis menampilkan presentase keberhasilan *algoritma Sidewinder* sebagai *generator maze* dan *algoritma Recursive Backtracker* sebagai *solver maze*.

## 1.6 Sistematika Penulisan

Bab 1 PENDAHULUAN, Bab ini berisi gambaran umum penelitian. Pendahuluan terdiri dari Latar Belakang, Perumusan Masalah, Batasan Masalah , Tujuan Penelitian, Metode Penelitian, dan Sistematika Penulisan.

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. Rancangan sistem yang dibuat berupa spesifikasi dari sistem, rancangan antarmuka sistem berupa input dan output.

Bab 4 IMPLEMENTASI DAN ANALISIS SISTEM, Bab ini memuat hasil riset/implementasi, dan pembahasan/analisis dari riset tersebut yang sifatnya terpadu.

Bab 5 KESIMPULAN DAN SARAN, Kesimpulan merupakan pernyataan singkat dan tepat yang dijabarkan dari hasil analisis kegiatan riset/implementasi dalam penyusunan skripsi. Saran memuat aktifitas yang dilakukan untuk mengembangkan kinerja sistem saat ini.

©UKDW



## BAB 5

### KESIMPULAN

#### 5.1 Kesimpulan

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

- a. Algoritma *Sidewinder* terbukti 100% mampu menghasilkan *perfect maze*.
- b. Algoritma Recursive Backtracker 100% akurat mencari jalan keluar dari sebuah maze yang digenerate menggunakan Algoritma *Sidewinder*.
- c. *Maze* persegi panjang vertikal lebih rumit dari *maze* persegi panjang horisontal yang dapat dilihat dari nilai *path nodes* dan nilai *traversed nodes* lebih tinggi pada *maze* persegi panjang vertikal.

#### 5.2 Saran

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

- a. Algoritma *Sidewinder* dapat diimplementasikan langsung kedalam sebuah game sebagai pembuat *map* secara dinamis.
- b. Dapat dilakukan penelitian lebih lanjut menggunakan algoritma *Recursive Backtracker* untuk mencari jalan keluar pada *imperfect maze*.

## DAFTAR PUSTAKA

- Buck, J. (2011). *Maze Generation: Sidewinder algorithm*. From The Buckblog assorted ramblings by Jamis Buck: <http://weblog.jamisbuck.org/2011/2/3/maze-generation-sidewinder-algorithm#>
- Kristiadi, V. N. (2015). PERBANDINGAN ALGORITMA "GROWING TREE" DAN "HUNT AND KILL" PADA PEMBUATAN RANDOM MAP MAZE.
- Lee , H. L., Lee , C. F., & Chen , L. H. (2010, August 3). A perfect maze based steganographic method. *The Journal of Systems and Software*, 2528-2535.
- Prayoga, H. (2014). Implementasi Algoritma Backtracking Pada Game Puzzle Kakuro.
- Pribadi , O. (2015). Maze Generator Dengan Menggunakan Algoritma Depth-First-Search. *Jurnal TIMES*, IV(1).
- Sasongko, A. C. (2008). Penerapan Metode Recursive Backtracker Sebagai Creator Dan Solver Dalam Game Maze.
- Setiowati, Y., Martiana, E., & Pramitasari, A. D. (2012). Penerapan Algoritma Backtrack pada Game Edukasi Labirin Matematika Berbasis Mobile. *The 14th Industrial Electronics Seminar 2012 (IES 2012)*.
- Sharma , K., & Munshi , C. (2015, February). A Comprehensive and Comparative Study Of Maze-Solving Techniques by Implementing Graph Theory. *IOSR Journal of Computer Engineering*, IV(1), 24-29.
- Sukumar , T., & Santha , D. (2012, July). Maze Based Data Hiding Using Back Tracker Algorithm. *International Journal of Engineering Research and Applications*, II(4), 499-504.
- Tando , A. (2012). Perbandingan Algoritma Depth-First Search dan Algoritma Hunt-and-Kill dalam Pembuatan Labirin.