

**SKRIPSI**

**PEMBANGKIT LABIRIN ACAK UNTUK PERMAINAN  
MAZE RACE**



**JAMES TJANDRA**

**NPM: 2010730048**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
2017**



**UNDERGRADUATE THESIS**

**RANDOMIZED LABYRINTH GENERATOR FOR MAZE  
RACE GAME**



**JAMES TJANDRA**

**NPM: 2010730048**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND  
SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
2017**



**LEMBAR PENGESAHAN**

**PEMBANGKIT LABIRIN ACAK UNTUK PERMAINAN  
MAZE RACE**

**JAMES TJANDRA**

**NPM: 2010730048**

**Bandung, 10 Mei 2017**

**Menyetujui,**

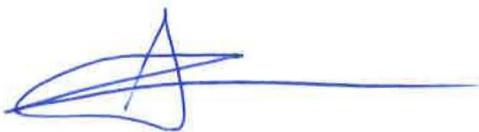
**Pembimbing**



**Luciana Abednego, M.T.**

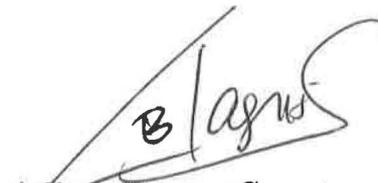


**Ketua Tim Penguji**



**Dott. Thomas Anung Basuki**

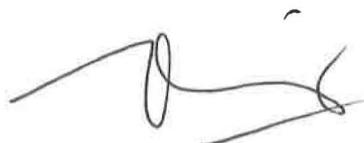
**Anggota Tim Penguji**



**Aditya Bagoes Saputra, M.T.**

**Mengetahui,**

**Ketua Program Studi**



**Mariskha Tri Adithia, P.D.Eng**



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### PEMBANGKIT LABIRIN ACAK UNTUK PERMAINAN MAZE RACE

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal 10 Mei 2017



James Tjandra  
NPM: 2010730048



## ABSTRAK

Pada jaman modern permainan yang membutuhkan ruang yang luas dan menghabiskan banyak sumber daya seperti permainan labirin mulai banyak menghilang. Namun perkembangan teknologi membuat permainan-permainan seperti *maze race* berkembang dalam bentuk simulasi komputer. Dalam simulasi komputer permainan *maze race*, pemain dapat melihat maze secara keseluruhan dari *bird view*. Jika pemain telah berhasil menemukan jalur keluar atau telah berhasil menjadi familiar dengan maze yang ada, permainan dapat menjadi membosankan dan tidak lagi menantang. Hal tersebut akan diatasi dengan labirin berbeda setiap kali permainan dijalankan. Labirin akan dibangkitkan dengan algoritma Prim Acak, yang menjamin labirin unik untuk setiap generasi. Juga untuk menyelesaikan permainan, komputer harus memiliki jalur gerakan yang sesuai dengan solusi untuk setiap labirin yang unik. Jalur gerakan bidak akan dicari dengan algoritma BFS.

**Kata-kata kunci:** Algoritma, Prim, BFS, labirin



## ABSTRACT

In modern times a game that requires vast space and spends a lot of resources like a maze game starts to disappear. However, the development of technology permits games like maze race evolve in the form of computer simulations. In the maze race game, player can view the overall maze from bird view. When a player has managed to find an exit point or has become familiar with an existing maze, the game can become boring and no longer challenging. This issue can be solved with a different maze each time the game is run. The maze will be generated with the Randomized Prim's algorithm, which ensures a unique maze for each generation. Also to complete the game, the computer must have a movement path corresponding to the solution for each unique maze. This path will be searched with a BFS algorithm.

**Keywords:** Algorithm, Prim, BFS, labyrinth



*Skripsi ini saya persembahkan untuk Papih dan Mamih saya,  
Tuhan Yang Maha Esa, serta Koko dan dede saya yang telah  
mentolerir saya selama 7 tahun ...*



## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya penulis berhasil menyelesaikan penyusunan skripsi yang berjudul "Pembangkit Labirin Acak untuk Permainan Maze Race". Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak, baik langsung maupun tidak langsung. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada.

- Keluarga tercinta, Papih, Mamih, Koko, Adik dan saudara-saudara semuanya yang tanpa henti memberikan hiburan dan dukungan dalam mengerjakan tugas akhir ini.
- Bu Joanna Helga atas kesabaran, bimbingan dan dorongan dalam proses pengerjaan skripsi.
- Bu Mariskha yang sudah memberikan kesempatan dan dorongan semangat dalam proses pengerjaan skripsi.
- Game Witcher 3 dan Fallout 4. Tanpa keduanya seharusnya saya lulus 2 tahun yang lalu.

Semoga segala bantuan dan dukungan dari semua pihak tersebut mendapat berkah dari Tuhan Yang Maha Esa. Akhir kata, penulis memohon maaf apabila terdapat kekurangan dalam penyusunan skripsi ini. Semoga skripsi ini berguna bagi semua pihak yang memerlukan.

Bandung, Mei 2017

Penulis



# DAFTAR ISI

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xix</b>
<b>DAFTAR TABEL</b>	<b>xxi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	3
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Labirin . . . . .	5
2.1.1 Dimensi . . . . .	5
2.1.2 Tesselation . . . . .	6
2.1.3 Routing . . . . .	8
2.1.4 Tekstur . . . . .	8
2.2 HTML5[1] . . . . .	9
2.2.1 JavaScript[2] . . . . .	10
2.3 Canvas . . . . .	11
2.4 Algoritma pembangkit labirin . . . . .	13
2.4.1 Teori Graf . . . . .	14
2.4.2 Kruskal's Algorithm[3] . . . . .	14
2.4.3 Randomized Prim's Algorithm[3] . . . . .	16
2.5 Maze Solver Algorithm[3] . . . . .	19
2.5.1 Random Mouse Algortihm . . . . .	19
2.5.2 Wall Follower . . . . .	20
2.5.3 BFS . . . . .	20
<b>3 ANALISIS</b>	<b>23</b>
3.1 Deskripsi labirin yang dibutuhkan . . . . .	23
3.2 Maze Race . . . . .	23
3.3 Analisis Kebutuhan Perangkat Lunak . . . . .	25
3.3.1 Deskripsi Perangkat Lunak . . . . .	25
3.3.2 Asumsi yang digunakan di Perangkat Lunak . . . . .	25
3.3.3 Diagram Use Case . . . . .	26
<b>4 PERANCANGAN</b>	<b>29</b>

4.1	Perancangan Antarmuka Aplikasi . . . . .	29
4.1.1	Antarmuka <i>scene</i> MenuScreen . . . . .	29
4.1.2	Antarmuka <i>scene</i> Generate . . . . .	31
4.1.3	Antarmuka <i>scene</i> Solve . . . . .	33
4.2	Deskripsi Fungsi . . . . .	33
<b>5</b>	<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>35</b>
5.1	Implementasi . . . . .	35
5.1.1	Lingkungan Implementasi Perangkat Keras . . . . .	35
5.1.2	Lingkungan Implementasi Perangkat Lunak . . . . .	35
5.1.3	Hasil Implementasi Aplikasi . . . . .	35
5.2	Pengujian Aplikasi . . . . .	38
5.2.1	Pengujian Fungsional . . . . .	39
5.2.2	Hasil Pengujian Fungsional . . . . .	39
<b>6</b>	<b>KESIMPULAN DAN SARAN</b>	<b>47</b>
6.1	Kesimpulan . . . . .	47
6.2	Saran . . . . .	47
	<b>DAFTAR REFERENSI</b>	<b>49</b>
	<b>A KODE PROGRAM</b>	<b>51</b>

## DAFTAR GAMBAR

2.1	2D labirin 2 dimensi	5
2.2	Labirin Delta	6
2.3	Labirin Sigma	7
2.4	Labirin Theta	7
2.5	Labirin Braid	8
2.6	labirin horizontal	9
2.7	labirin simetris	9
2.8	Javascript example	11
2.9	Javascript example 2	11
2.10	tampilan awal canvasTes.html	13
2.11	tampilan canvasTes.html setelah menerima input	13
2.12	graf sample	14
2.13	Kruskal Algoritma step 0 sel belum terhubung	15
2.14	Kruskal Algoritma step 1 sel (2,2) dan (2,3) telah terhubung	15
2.15	Kruskal Algoritma step 2 iterasi langkah 2 Algoritma kruskal	15
2.16	Kruskal Algoritma step 3 <i>tree</i> A dan E bergabung menjadi sebuah <i>tree</i> A	15
2.17	Kruskal Algoritma step 4 <i>tree</i> A dan <i>tree</i> C tersisa	16
2.18	Kruskal Algoritma step 5 seluruh sel telah terhubung	16
2.19	Prim Algoritma step 0 sel belum terhubung	17
2.20	Prim Algoritma step 1; sel dipilih random untuk <i>starting point</i>	17
2.21	Prim Algoritma step 2; tandai frontier/perbatasan	17
2.22	Prim Algoritma step 3; penandaan frontier dan pemilihan sel yang akan dimasukkan dalam labirin	18
2.23	Prim Algoritma step 4; menandai frontier	18
2.24	Prim Algoritma step 5; memilih sel secara acak	18
2.25	Prim Algoritma step 6; iterasi berulang-ulang	18
2.26	Perfect labirin	19
2.27	Maze solving 1; wall follower	20
2.28	BFS solving 1 start	20
2.29	BFS solving 2; tetangga pertama	21
2.30	BFS solving 3; tetangga kedua	21
2.31	BFS solving 4; setelah iterasi selesai	21
2.32	BFS solving 5; return jalur	22
3.1	Use-Case Diagram	26
4.1	Antarmuka <i>scene</i> MenuScreen	29
4.2	contoh labirin berukuran 2×2	30
4.3	<i>contoh abirin</i> 10×10	31
4.4	Antarmuka <i>scene</i> Generate	31
4.5	Labirin yang ditampilkan di bawah antarmuka	32
4.6	Antarmuka <i>scene</i> Solve	33

5.1	Antarmuka <i>scene</i> MenuScreen . . . . .	36
5.2	Antarmuka <i>scene</i> Generate . . . . .	37
5.3	Antarmuka <i>scene</i> Solve . . . . .	38
5.4	generated labirin . . . . .	40
5.5	Labirin yang dibangkitkan dengan IE . . . . .	40
5.6	Labirin yang dibangkitkan dengan Mozilla . . . . .	41
5.7	Labirin dibangkitkan dengan Opera browser . . . . .	41
5.8	Labirin dibangkitkan dengan UCBrowser . . . . .	42
5.9	Labirin yang dibangkitkan pada <i>Asus mobile phone</i> web browser . . . . .	43
5.10	Labirin yang dibangkitkan pada <i>Asus mobile phone</i> google chrome . . . . .	44
5.11	Labirin yang dibangkitkan pada <i>Asus mobile phone</i> 4share web viewer . . . . .	45

## DAFTAR TABEL

3.1	Skenario Generate labirin . . . . .	27
3.2	Skenario solveLabirin . . . . .	27
3.3	Skenario clearSolution . . . . .	27
3.4	Skenario Input Movement . . . . .	27
4.1	Tabel nilai paramater labirin <i>default</i> . . . . .	30
4.2	Tabel paramater warna labirin <i>default</i> . . . . .	30



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*Labirin* atau *maze* adalah sebuah kumpulan dari jalur yang membingungkan di mana semua jalur terhubung satu sama lain. Pembuatan labirin sendiri awalnya memiliki tujuan untuk menghambat pergerakan musuh pada jaman peperangan dengan jalan berliku dan penanda jalan ambigu. Hasilnya adalah membuat orang yang masuk ke dalamnya menjadi tersesat. Dahulu kegunaan labirin adalah untuk menyesatkan orang yang masuk namun kebutuhan untuk menyesatkan mulai berkurang dengan berkurangnya musuh yang menginvasi. Sekarang labirin digunakan sebagai permainan, orang yang masuk ke dalam labirin mencoba mencari hiburan dengan mengejar orang lain dengan bermain *hide and seek* atau berusaha mencapai titik tertentu di dalam labirin.

Ada berbagai macam permainan yang dapat dinikmati dalam sebuah *maze*. Salah satu permainan yang dapat dinikmati adalah *maze race*. *maze race* memiliki aturan yang sangat sederhana : pemain akan ditempatkan dalam posisi yang berbeda secara acak di dalam labirin. Pemain akan dinyatakan sebagai pemenang jika sang pemain berhasil menjadi pemain pertama yang mencapai lokasi tujuan yang telah ditentukan di dalam labirin. Dalam permainan ini jumlah pemain tidak dibatasi, prinsipnya sama dengan maraton, tidak peduli berapa jumlah peserta atau berapa banyak yang berhasil mencapai titik akhir, yang mencapai titik akhir pertamalah yang menjadi sang pemenang.

Sayangnya pada jaman modern permainan yang membutuhkan ruang yang luas dan menghabiskan banyak sumber daya seperti permainan labirin mulai banyak menghilang. Namun perkembangan teknologi membuat permainan-permainan seperti *maze race* berkembang dalam bentuk simulasi komputer. Dalam simulasi komputer permainan *maze race*, pemain dan komputer akan mengontrol bidak permainannya masing masing. Perbedaan yang paling besar adalah dalam simulasi permainan komputer, pemain dapat melihat maze secara keseluruhan dari *bird-eye view* atau pandangan dari sudut pandang yang tinggi sehingga seolah-olah terlihat oleh burung yang terbang di langit. Sedangkan pada dunia nyata pemain akan maju tanpa mengetahui lika-liku *maze* . Sayangnya baik dalam simulasi maupun dunia nyata, jika pemain telah berhasil menemukan jalur keluar atau telah berhasil menjadi terbiasa dengan *maze* yang ada, permainan dapat menjadi membosankan dan tidak lagi menantang. Hal tersebut menjadi alasan mengapa diperlukan pembangkit *maze* baru untuk setiap permainan.

Pada skripsi ini dikembangkan sebuah permainan *maze race*. Perangkat lunak yang dibuat memiliki dua komponen utama. Hal pertama yang harus dibuat adalah pembangkit labirin acak sebagai rangka utama permainan. Hal kedua yang harus dibuat adalah algoritma yang berperan sebagai penggerak bidak komputer. Maka secara garis besar perangkat lunak ini merupakan permainan *maze race* dengan labirin yang acak dan memiliki *computer player* yang siap melawan pemain dalam *bird view*.

### 1.2 Rumusan Masalah

Berdasarkan deskripsi yang telah diberikan dapat disimpulkan beberapa rumusan masalah.

- Bagaimana cara membangkitkan labirin secara acak?
- Bagaimana cara mengimplementasikan permainan *maze race* dalam bentuk simulasi komputer?
- Bagaimana cara mengimplementasikan algoritma yang dapat mencari jalur terpendek sebagai sistem penggerak komputer lawan bermain.

### 1.3 Tujuan

Setelah meneliti rumusan masalah yang tercantum, dapat disimpulkan bahwa skripsi ini memiliki tujuan sebagai berikut :

- Menghasilkan perangkat lunak dapat menghasilkan labirin yang berbeda setiap kali permainan dijalankan.
- Perangkat lunak dapat digunakan oleh pemain dengan media elektronik komputer, dan pemain dapat menggerakkan bidak sesuai dengan keinginan.
- Bidak komputer lawan bermain akan bergerak sesuai dengan jalur terpendek yang dihasilkan oleh algoritma.

### 1.4 Batasan Masalah

Setelah meneliti tujuan serta rumusan masalah yang tersedia, dapat ditambahkan beberapa batasan masalah yang membatasi skripsi ini. Berikut adalah batasan masalah yang telah ditentukan :

- Ukuran dimensi labirin jika tidak ditentukan oleh pemain akan berdimensi 25 sel  $\times$  25 sel, dan hanya menghasilkan 1 jenis labirin dengan 4 buah dinding berbentuk persegi empat.
- Bidak yang digerakan oleh pemain adalah bidak yang berada di ujung kiri atas dan berwarna hitam, sedangkan bidak yang digerakan oleh *cpuPlayer* adalah bidak yang berada di ujung kiri bawah berwarna hijau.
- Bidak *cpuPlayer* lawan bermain akan bergerak sesuai dengan kecepatan yang konstan, tanpa menyesuaikan dengan dimensi labirin yang dibangkitkan.

### 1.5 Metodologi

#### 1. Studi Literatur

Melakukan studi pustaka untuk mengetahui teori-teori yang berkaitan dengan penelitian yang dilakukan mengenai algoritma pembangkit labirin dan algoritma untuk mencari jalur terpendek di dalam labirin.

#### 2. Pengumpulan Data

Melakukan pengamatan pada kriteria labirin agar labirin yang dihasilkan mampu memberikan kepuasan bermain bagi siapapun yang menggunakan.

#### 3. Perancangan Aplikasi

Dari hasil studi literatur dan pengumpulan data akan dibuat deskripsi program, kemudian dilakukan perancangan awal aplikasi yang akan dibuat, sehingga dihasilkan perancangan antarmuka yang siap untuk diimplementasi.

4. Pembuatan Aplikasi  
Rancangan aplikasi diimplementasi dengan basis HTML5 menggunakan JavaScript.
5. Uji Coba dan Evaluasi  
Melakukan uji coba dan evaluasi hasil aplikasi untuk memeriksa apakah terjadi kesalahan dan apakah terdapat hasil yang tidak akurat.
6. Penulisan Tugas Akhir  
Penulisan tugas akhir yang sudah diuji coba dan dievaluasi.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan dari tugas akhir ini adalah sebagai berikut:

1. Bab Pendahuluan  
Bab 1 berisi latar belakang, rumusan masalah, tujuan, batasan masalah, dan metodologi penelitian.
2. Bab Dasar Teori  
Bab 2 berisi teori-teori dasar yang menunjang penelitian, yaitu mengenai jenis-jenis labirin, HTML5, JavaScript, teori dasar graf, mengenai algoritma pembangkit labirin (Prim, dan Kruskal) dan teori *maze solver algorithm* untuk mencari jalan yang harus dilalui (*Random Mouse Algorithm, Wall follower, Breadth-first search*).
3. Bab Analisis  
Bab 3 berisi analisis mengenai hasil pengumpulan data dan pembuatan rancangan algoritma pembangkit labirin.
4. Bab Perancangan  
Bab 4 berisi perancangan aplikasi permainan *maze race* mulai dari masukan hasil labirin yang dibangkitkan hingga kemampuan pergerakan visual permainan.
5. Bab Implementasi dan Pengujian  
Bab 5 berisi implementasi dan pengujian permainan *maze race*.
6. Bab Kesimpulan dan Saran  
Bab 6 berisi kesimpulan dan saran dari penelitian yang dilakukan.