

SKRIPSI

**PEMBANGUNAN GIM SNAKE 360 BERBASIS WEB
DENGAN KODE TERBUKA**



Evelyn Wijaya

NPM: 2015730030

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2019**

UNDERGRADUATE THESIS

OPEN SOURCE SNAKE 360



Evelyn Wijaya

NPM: 2015730030

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2019**

LEMBAR PENGESAHAN

PEMBANGUNAN GIM SNAKE 360 BERBASIS WEB DENGAN KODE TERBUKA

Evelyn Wijaya

NPM: 2015730030

Bandung, 24 Mei 2019

Menyetujui,

Pembimbing

Dr. Veronica Sri Moertini

Ketua Tim Penguji

Anggota Tim Penguji

Chandra Wijaya, M.T.

Raymond Chandra Putra, 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:

PEMBANGUNAN GIM SNAKE 360 BERBASIS WEB DENGAN KODE TERBUKA

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 24 Mei 2019

Meterai Rp. 6000

Evelyn Wijaya
NPM: 2015730030

ABSTRAK

Penelitian ini membahas mengenai pembangunan permainan Open Source Snake 360. Permainan ini dibuat berdasarkan acuan dari permainan *Snake* yang sudah ada. Permainan *Snake* adalah permainan mengontrol gerakan ular untuk mendapatkan makanan yang tersebar di labirin. Umumnya pada permainan *Snake*, ular hanya dapat bergerak ke atas, ke bawah, ke kiri dan ke kanan saja. Selain itu labirin yang disediakan terbatas. Tujuan dari penelitian ini adalah membangun permainan *Snake* yang ularnya dapat bergerak ke segala arah dan orang lain dapat menambahkan labirin buatan sendiri.

Open Source Snake 360 adalah sebuah permainan yang dibuat menggunakan HTML5 Canvas dan *Javascript*. *jQuery* digunakan untuk mengakses labirin dan memuat labirin dari server. Pull request Github digunakan untuk menambahkan labirin buatan sendiri.

Berdasarkan penelitian yang sudah dilakukan dapat diperoleh kesimpulan bahwa ular sudah dapat bergerak ke segala arah dan orang lain dapat menambahkan labirin buatan sendiri. Orang lain dapat menambahkan labirin buatan sendiri dengan menggunakan *pull request* pada *GitHub*. Permainan ini juga sudah dapat memuat labirin-labirin yang dibuat oleh orang lain.

Kata-kata kunci: Permainan *Snake*, HTML5 Canvas, *Javascript*, *jQuery*, *GitHub*

ABSTRACT

This study discusses the construction of the Open Source Snake 360. This game is based on a reference to the existing game, Snake. Snake is a game that controls the movement of snake to get food that scattered in the maze. In Snake, the snake can only move up, down, left and right. In addition, the labyrinth provided is limited. The purpose of this research is to build a game of Snake whose snake can move in any direction and others can add mazes of their own.

Open Source Snake 360 is a game created using HTML5 Canvas and Javascript. jQuery is used to access the maze and load the maze from the server. Pull request GitHub is used to add mazes made by others.

Based on the research that has been done, it can be concluded that the snake can move in any direction and other people can add their own mazes. Others can add their own mazes by using pull request on GitHub. This game can also load mazes made by other people.

Keywords: Snake Game, HTML5 Canvas, Javascript, jQuery, GitHub

Dipersembahkan kepada orang tua

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul Pembangunan Gim Snake 360 Berbasis Web dengan Kode Terbuka. Skripsi ini ditulis untuk memenuhi syarat kelulusan pada Teknik Informatika Unpar.

Penulisan skripsi ini tidak dapat selesai pada waktunya tanpa bantuan dari berbagai pihak. Oleh karena itu penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Tuhan Yang Maha Esa yang memberikan kekuatan dan selalu menyertai dalam pembuatan skripsi ini.
2. Bapak Pascal Alfadian Nugroho selaku dosen pembimbing skripsi yang telah membimbing dan membantu penulis dalam pembuatan skripsi ini.
3. Ayah dan Ibu yang selalu mendoakan dan telah memberikan semangat dan motivasi kepada penulis agar dapat menyelesaikan skripsi ini.
4. Teman-teman Teknik Informatika yang telah mendukung dan memberikan semangat kepada penulis dalam pembuatan skripsi ini.
5. Semua pihak yang membantu penulis dan tidak sempat terucapkan.

Penulis berharap bahwa skripsi ini dapat memberikan informasi tentang pembangunan permainan berbasis web kepada masyarakat. Penulis menyadari bahwa skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis bersedia menerima kritik dan saran yang membangun.

Bandung, Mei 2019

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
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	2
2 LANDASAN TEORI	5
2.1 <i>Snake Game</i>	5
2.2 <i>Open Source</i>	6
2.3 <i>HTML5 Canvas</i>	6
2.4 <i>Javascript</i>	7
2.4.1 Menggambar pada <i>Canvas</i>	8
2.4.2 <i>Object Oriented Programming Javascript</i>	10
2.4.3 <i>Event</i>	12
2.4.4 Membuat Animasi	14
2.4.5 Fungsi <i>Callback</i>	15
2.5 <i>jQuery</i>	16
2.5.1 <i>Looping</i>	17
2.5.2 <i>Chaining</i>	17
2.5.3 Mendapatkan dan Mengubah Konten Elemen	18
2.5.4 Mendapatkan dan Mengubah Properti CSS	18
2.5.5 <i>Event</i>	19
2.5.6 Mengubah dan Mendapatkan Dimensi dari Elemen	20
2.6 <i>AJAX</i>	20
2.6.1 <i>Ajax Request dan Response</i>	21
2.6.2 <i>jQuery dan AJAX Request</i>	22
2.6.3 <i>jQuery dan AJAX Response</i>	22
2.7 <i>Version Control</i>	23
2.7.1 <i>Git</i>	24
2.7.2 <i>Git Branching</i>	27
2.7.3 <i>GitHub</i>	33
2.8 Teknik Mendeteksi Tabrakan	34

3	ANALISIS	35
3.1	Analisis Permainan <i>Snake</i> yang Sudah Ada	35
3.1.1	Ular dan Makanan	35
3.1.2	Pergerakan Ular	36
3.1.3	Labirin	37
3.2	Analisis Sistem yang Dibangun	37
3.2.1	Menggambar Ular dan Apel	37
3.2.2	Pergerakan Ular	40
3.2.3	Mengacak posisi apel	41
3.2.4	Menentukan Besar <i>Canvas</i>	42
3.2.5	Membuat Labirin	42
3.2.6	Pengecekan tabrakan (<i>Collision Detection</i>)	43
3.2.7	Memilih Labirin, Kecepatan Laju Ular dan Kecepatan Berbelok Ular	45
3.3	Analisis Berorientasi Objek	46
3.3.1	Skenario Permainan	46
3.3.2	Diagram Kelas	46
4	PERANCANGAN	49
4.1	Rancangan Diagram <i>Sequence</i>	49
4.2	Rancangan Diagram <i>State</i>	50
4.3	Rancangan Diagram Kelas Rinci	50
4.4	Rancangan Tampilan Antarmuka	55
4.4.1	Tampilan Menu Utama	55
4.4.2	Tampilan Bermain	56
4.4.3	Tampilan Permainan Berakhir	56
5	IMPLEMENTASI DAN PENGUJIAN	59
5.1	Implementasi	59
5.1.1	Lingkungan Perangkat Keras	59
5.1.2	Lingkungan Perangkat Lunak	60
5.1.3	Lingkungan Pengujian	60
5.1.4	Implementasi Antarmuka	61
5.2	Pengujian	66
5.2.1	Pengujian Fungsional	66
5.2.2	Pengujian Eksperimental	69
6	KESIMPULAN DAN SARAN	77
6.1	Kesimpulan	77
6.2	Saran	77
	DAFTAR REFERENSI	79
	A KODE PROGRAM	81
	B FILE README	89

DAFTAR GAMBAR

2.1	Permainan Snake pada telepon genggam <i>Nokia</i>	5
2.2	Permainan <i>Slither.io</i> pada <i>Android</i>	5
2.3	Posisi kotak biru pada <i>canvas</i> terhadap <i>origin</i>	8
2.4	Perbedaan <i>quadratic Bézier curve</i> dan <i>cubic Bézier curve</i>	10
2.5	Local Version Control	23
2.6	Centralized Version Control	24
2.7	Distributed Version Control	24
2.8	Working tree, staging area, dan Git directory	25
2.9	Siklus hidup pada status <i>file</i>	26
2.10	Commit dan tree dari file yang dicommit	28
2.11	Commit dan parent dari commit	28
2.12	<i>Pointer HEAD</i> menunjuk <i>branch master</i>	29
2.13	<i>Pointer HEAD</i> beserta <i>branch testing</i>	29
2.14	3 <i>snapshot</i> yang digunakan dalam <i>three way merge</i>	30
2.15	<i>Merge commit</i>	30
2.16	Perbedaan pada <i>branch</i> lokal dan <i>remote</i>	31
2.17	<i>Update remote-tracking branches</i> menggunakan perintah <i>git fetch</i>	31
2.18	<i>Rebasing commit C4</i> ke <i>C3</i>	32
2.19	<i>Merge branch</i> setelah <i>rebasing</i>	32
2.20	Tombol 'Fork'	33
3.1	Ular pada <i>Slither.io</i>	35
3.2	Makanan pada <i>Slither.io</i>	36
3.3	Ular pada <i>Snake Nokia</i>	36
3.4	Makanan biasa(A) dan makanan bonus(B) pada <i>Snake Nokia</i>	36
3.5	Ular sedang melaju dengan cepat(<i>speed up</i>)	37
3.6	Peta labirin pada <i>Slither.io</i>	37
3.7	Koordinat bagian tubuh ular pada <i>array</i>	38
3.8	Tubuh ular setelah digambar menggunakan garis	38
3.9	Bagian pada apel(lingkaran merah) yang akan dibuat menggunakan kurva	38
3.10	Pembagian gambar apel dengan layout persegi beserta ukuran pada setiap bagian	39
3.11	<i>Start point</i> , <i>control point</i> dan <i>end point</i> untuk menggambar apel bagian kiri atas	40
3.12	<i>Start point</i> , <i>control point</i> dan <i>end point</i> untuk menggambar apel bagian kiri bawah	40
3.13	Ilustrasi ular sebelum bergerak maju(A) dan setelah bergerak maju(B)	40
3.14	Gambar apel yang terpotong sesudah mengacak posisi apel	42
3.15	Menggambar dinding menggunakan simbol pada file teks	43
3.16	Ular ingin melewati jalur yang diapit oleh 2 buah dinding	43
3.17	Daerah tabrakan pada apel	44
3.18	Daerah tabrakan berbentuk persegi pada apel	44
3.19	Posisi kepala ular pada sebuah daerah labirin	45
3.20	Diagram <i>use case</i> dari permainan <i>Snake 360</i>	46
3.21	Diagram class dari permainan <i>Snake 360</i>	47

4.1	Diagram <i>sequence</i> untuk memuat labirin	49
4.2	Diagram state proses bermain dari Open Source <i>Snake 360</i>	50
4.3	Diagram class rinci dari Open Source <i>Snake 360</i>	51
4.4	Rancangan tampilan menu utama	55
4.5	Rancangan tampilan menu utama jika pemain salah memasukkan data	56
4.6	Rancangan tampilan bermain	56
4.7	Rancangan tampilan permainan berakhir	57
5.1	Tampilan memilih branch repository untuk dihosting	60
5.2	Tampilan setelah melakukan hosting di GitHub	61
5.3	Tampilan menu utama pada <i>desktop</i>	61
5.4	Tampilan menu utama pada <i>smartphone</i>	62
5.5	Tampilan menu utama jika pemain salah memasukkan data labirin pada <i>desktop</i>	62
5.6	Tampilan menu utama jika pemain salah memasukkan data labirin pada <i>smartphone</i>	63
5.7	Tampilan bermain pada <i>desktop</i>	64
5.8	Tampilan bermain pada <i>smartphone</i>	64
5.9	Tampilan permainan berakhir pada desktop	65
5.10	Tampilan permainan berakhir pada smartphone	66
5.11	Tampilan hasil <i>pull request</i> milik penguji 1	69
5.12	Tampilan pengujian labirin ke 6 yang dibuat oleh penguji 1	70
5.13	Tampilan hasil <i>pull request</i> milik penguji 2	71
5.14	Tampilan pengujian labirin ke 7 yang dibuat oleh penguji 2	71
5.15	Tampilan pengujian labirin ke 8 yang dibuat oleh penguji 2	72
5.16	Tampilan pengujian labirin ke 9 yang dibuat oleh penguji 2	72
5.17	Tampilan hasil pull request milik penguji 3	73
5.18	Tampilan pengujian labirin yang dibuat oleh penguji 3	73
5.19	Tampilan hasil <i>pull request</i> milik penguji 4	74
5.20	Tampilan pengujian labirin yang dibuat oleh penguji 4	74
5.21	Tampilan hasil <i>pull request</i> milik penguji 5	75
5.22	Tampilan pengujian labirin yang dibuat oleh penguji 5	75

DAFTAR TABEL

5.1	Pengujian Fungsional pada Tampilan Menu Utama	67
5.2	Pengujian Fungsional Tampilan Bermain pada Desktop	67
5.3	Pengujian Fungsional Tampilan Bermain pada <i>Smartphone</i>	68
5.4	Pengujian Fungsional Tampilan Permainan Berakhir	68

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Snake merupakan sebuah permainan yang pertama kali dibuat oleh Peter Trefonas pada tahun 1978. Konsep *Snake* berasal dari permainan arkade yaitu *Blockade*. Awalnya *Snake* hanya dapat dimainkan pada komputer pribadi. Namun pada tahun 1997, *Snake* dapat dimainkan pada telepon genggam *Nokia*¹. Cara bermain *Snake* adalah pemain menggerakkan ular pada sebuah labirin. Ular tersebut harus mendapatkan makanan sebanyak-banyaknya tanpa menabrak dinding atau ular itu sendiri. Setiap memakan makanan, tubuh ular akan memanjang dan pemain akan semakin sulit untuk menggerakkan ular tersebut dengan bebas karena tubuh ular semakin lama akan menutupi labirin tersebut.

HTML(*Hyper Text Markup Language*) adalah sebuah bahasa markah yang digunakan untuk membuat halaman web. HTML5 merupakan HTML versi 5 dan penerus dari HTML4, XHTML1, dan DOM *level 2* HTML. HTML5 memiliki beberapa elemen baru, salah satunya adalah HTML5 Canvas. HTML5 Canvas adalah tempat untuk menggambar *pixel-pixel* yang dapat ditulis menggunakan bahasa pemrograman *JavaScript*. *JavaScript* adalah bahasa pemrograman tingkat tinggi yang digunakan untuk membuat halaman web menjadi lebih interaktif. *jQuery* merupakan *library JavaScript* yang cepat, kecil dan kaya dengan fitur. *jQuery* membuat hal-hal seperti traversal dan manipulasi dokumen HTML, penanganan *event*, animasi dan *Ajax* jauh lebih sederhana dengan API(*Application Programming Interface*) yang mudah untuk digunakan pada banyak *browser*. *GitHub* adalah layanan *web hosting* bersama untuk proyek pengembangan perangkat lunak yang menggunakan sistem *version control* yaitu *Git*. Dengan adanya *GitHub*, *programmer* dapat mengetahui perubahan yang pada *repository* tersebut. *Pull request* adalah sebuah fitur *GitHub* yang digunakan sebagai tempat diskusi antara kontributor dengan pemilik proyek/*owner*. Proyek yang diajukan oleh kontributor akan digabungkan dengan proyek milik *owner* jika *owner* merasa senang dengan perubahan yang sudah dibuat oleh kontributor.

Pada permainan *Snake*, umumnya pergerakan ular hanya atas, bawah, kiri, dan kanan saja. Pada skripsi ini, peneliti akan membuat permainan *Snake* yang ularnya dapat bergerak ke segala arah dan orang lain dapat menambahkan labirin menggunakan mekanisme *pull request GitHub*. Dengan begitu, orang lain dapat menambahkan labirin sesuai dengan keinginannya dan pemain tidak akan cepat bosan karena labirin yang disediakan cukup banyak dan variatif.

1.2 Rumusan Masalah

Rumusan dari masalah yang akan dibahas pada skripsi ini adalah sebagai berikut:

- Bagaimana membangun permainan *Snake* menggunakan HTML5 Canvas?
- Bagaimana cara menyimpan labirin pada *file* eksternal?

¹[https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))

- Bagaimana cara menggunakan *pull request* pada *GitHub* agar orang lain dapat menambahkan labirin?

1.3 Tujuan

Tujuan-tujuan yang hendak dicapai melalui penulisan skripsi ini adalah sebagai berikut:

- Dapat membangun permainan *Snake* menggunakan HTML5 Canvas.
- Dapat menyimpan labirin pada *file* eksternal.
- Dapat menggunakan *pull request* pada *GitHub* agar orang lain dapat menambahkan labirin.

1.4 Batasan Masalah

Beberapa batasan yang dibuat terkait dengan pengerjaan skripsi ini adalah sebagai berikut:

- *Web browser* yang digunakan sudah mendukung HTML5 Canvas.
- Permainan ini hanya dapat dimainkan oleh 1 orang.

1.5 Metodologi

Metodologi pada penelitian ini adalah sebagai berikut:

1. Melakukan studi literatur tentang HTML5 Canvas, *Open Source*, *JavaScript*, *jQuery*, *AJAX*, dan *Version Control*.
2. Melakukan analisis dan menentukan objek-objek.
3. Merancang algoritma untuk menggambar tubuh ular, pergerakan ular dan membuat labirin.
4. Mengimplementasikan keseluruhan algoritma.
5. Menambahkan labirin menggunakan *pull request* pada *GitHub*.
6. Melakukan pengujian.
7. Melakukan penarikan kesimpulan.
8. Menuliskan dokumen skripsi.

1.6 Sistematika Pembahasan

Sistematikan penulisan setiap bab pada penelitian ini adalah sebagai berikut:

1. Bab 1 berisikan latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, dan sistematika pembahasan dari penelitian yang dilakukan.
2. Bab 2 berisikan dasar-dasar teori yang menunjang penelitian ini. Teori yang digunakan adalah: pengertian *Snake Game*, *Open Source*, HTML5 Canvas, *Javascript*, *jQuery*, *AJAX*, dan *Version Control*.
3. Bab 3 berisikan analisis sistem yang sudah ada, analisis sistem yang dibangun dan analisis berorientasi objek. Analisis sistem yang sudah ada berisikan analisis permainan *Snake Nokia* dan *Slither.io*. Analisis berorientasi objek berisikan skenario permainan dan diagram kelas.

4. Bab 4 berisikan perancangan perangkat lunak yang dibangun. Perancangan yang dilakukan meliputi perancangan diagram *sequence*, perancangan diagram kelas dan perancangan tampilan antarmuka.
5. Bab 5 berisikan implementasi dan pengujian perangkat lunak.
6. Bab 6 berisikan kesimpulan dan saran.