

**APLIKASI PERMAINAN MATH MAZE MENGGUNAKAN ALGORITMA
GROWING TREE SEBAGAI PEMBANGKIT DAN PENYELESAI
PERMAINAN**

SKRIPSI

Oleh:

Moh. Nuruddin Effendi

NIM. 09650105



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2013**

**APLIKASI PERMAINAN *MATH MAZE* MENGGUNAKAN ALGORITMA
GROWING TREE SEBAGAI PEMBANGKIT DAN PENYELESAI
PERMAINAN**

SKRIPSI

Diajukan Kepada:

Fakultas Sains dan Teknologi

Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang

Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:

Moh. Nuruddin Effendi

NIM. 09650105

JURUSAN TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

**UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG**

2013

**APLIKASI PERMAINAN *MATH MAZE* MENGGUNAKAN ALGORITMA
GROWING TREE SEBAGAI PEMBANGKIT DAN PENYELESAI
PERMAINAN**

Oleh:

Moh. Nuruddin Effendi

NIM. 09650105

Telah disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T
19780625 200801 2 006

M. Amin Hariyadi, M.T
19670118 200501 1 001

Tanggal, 10 Januari 2013

**Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika**

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

**APLIKASI PERMAINAN *MATH MAZE* MENGGUNAKAN ALGORITMA
GROWING TREE SEBAGAI PEMBANGKIT DAN PENYELESAI
PERMAINAN**

SKRIPSI

Oleh:

Moh. Nuruddin Effendi

NIM. 09650105

**Telah Dipertahankan di Depan Dewan Penguji Tugas akhir dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Tanggal, 21 Januari 2013

Susunan Dewan Penguji	Tanda Tangan
1. Penguji Utama : <u>Ririen Kusumawati, M.Kom</u> NIP. 19720309 200501 2 002	()
2. Ketua : <u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008	()
3. Sekretaris : <u>Hani Nurhayati, M.T</u> NIP. 19780625 200801 2 006	()
4. Anggota : <u>M. Amin Hariyadi, M.T</u> NIP. 19670118 200501 1 001	()

**Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika**

**Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002**

ORISINALITAS PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama : Moh. Nuruddin Effendi

NIM : 09650105

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : **APLIKASI PERMAINAN MATH MAZE**

**MENGGUNAKAN ALGORITMA GROWING TREE
SEBAGAI PEMBANGKIT DAN PENYELESAI
PERMAINAN**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 10 Januari 2013

Yang Membuat Pernyataan,

Moh. Nuruddin Effendi
NIM. 09650105



KEMAMPUAN ITU ADA BATASNYA, TETAPI USAHA TIDAK ADA BATASNYA
Hiruma - Eyeshield 21



PERSEMBAHAN

Dengan mengucapkan rasa syukur seraya mengharap ridho Allah SWT.

Kupersembahkan karya ini kepada:

Abi dan Umi tercinta

Sajat Akhwani, S.H dan Sulastri

Yang senantiasa mendidik, membina, membesarkan dengan kasih sayang dan segala pengorbanannya yang tak terhitung sehingga Saya bisa tumbuh besar dan berpendidikan.

Semoga Allah SWT melindungi dan menyayangi keduanya . . .

Serta kakak-kakak **Moch. Taufiq Ridlo, Eny Muhimmatul F.**, dan adik Saya

Murtadlo Anugerah Pamungkas

♦ Yang selalu mendukung Saya

Semoga mereka selalu dalam lindungan-Nya . . .

Amien Ya Robbal Alamien..

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puja dan puji syukur Alhamdulillah kami panjatkan kehadirat Allah SWT yang Maha Pengasih lagi Maha Penyayang yang telah memberikan rahmat, taufik, hidayah dan inayah-Nya kepada kita serta memberikan nikmat Islam dan Iman serta tak lupa nikmat kesehatan yang diberikan kepada penulis khususnya sehingga penulis dapat menyelesaikan karya ilmiah dengan judul “APLIKASI PERMAINAN *MATH MAZE* MENGGUNAKAN ALGORITMA *GROWING TREE* SEBAGAI PEMBANGKIT DAN PENYELESAI PERMAINAN”. Penelitian ini dimaksudkan untuk memenuhi salah satu syarat dalam meraih gelar Sarjana Komputer (S.Kom) di Fakultas Sains dan Teknologi Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Penulis menyadari bahwa banyak pihak yang telah membantu dalam menyelesaikan penulisan tugas akhir ini. Untuk itu, iringan doa dan ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Ibu Hani Nurhayati, M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan, serta memberikan saran, kemudahan dan kepercayaan dalam dalam penyelesaian tugas akhir ini.
2. Bapak M. Amin Haryadi, M.T selaku dosen pembimbing II yang telah memberi arahan, motivasi, masukan, saran serta bimbingan.
3. Bapak Prof. Dr. Imam Suprayogo, selaku rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

4. Bapak Prof. Dr. Sutiman, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
5. Ibu Ririen Kusumawati, M.Kom selaku ketua jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
6. Seluruh Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang yang telah mengalirkan ilmu, pengetahuan, pengalaman, dan wawasannya, sebagai pedoman dan bekal bagi penulis.
7. Teman-teman satu jurusan Teknik Informatika angkatan 2009 yang telah bersedia berbagi ilmu dan informasi selama menimba ilmu di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang ini, khususnya Harditya Rahmat Ramadhan yang telah mengalirkan ilmu, pengetahuan, dan pengalaman sehingga penulis mengetahui banyak hal tentang Teknologi Informasi dan Komunikasi.
8. Dan semua pihak yang tidak dapat penulis sebutkan satu-persatu. Penulis ucapkan terimakasih banyak atas bantuan, dan motivasinya.

Akhirnya atas segala kekurangan dari penyusunan skripsi ini, sangat diharapkan kritik dan saran yang bersifat konstruktif dari semua pembaca demi memperbaiki kualitas penulisan selanjutnya. Semoga apa yang telah tertulis di dalam skripsi ini dapat memberikan kontribusi yang bermanfaat dan menambah khasanah ilmu pengetahuan. Amien...

Wassalamu'alaikum Wr. Wb.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN.....	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
ABSTRAK	xv
BAB I.....	Error! Bookmark not defined.
1.1. Latar Belakang	Error! Bookmark not defined.
1.2. Rumusan Masalah	Error! Bookmark not defined.
1.3. Batasan Masalah.....	Error! Bookmark not defined.
1.4. Tujuan Penelitian.....	Error! Bookmark not defined.
1.5. Manfaat Penelitian.....	Error! Bookmark not defined.
1.6. Sistematika Penulisan.....	Error! Bookmark not defined.
1.7. Metode Penelitian.....	Error! Bookmark not defined.
BAB II	Error! Bookmark not defined.
1.1. Permainan dan Jenisnya	Error! Bookmark not defined.
1.1.1. Permainan (<i>Game</i>)	Error! Bookmark not defined.
1.1.2. Jenis-Jenis <i>Game</i>	Error! Bookmark not defined.

1.2.	Permainan Math Maze.....	Error! Bookmark not defined.
1.3.	Pendidikan Anak Usia Dini	Error! Bookmark not defined.
1.4.	Pelajaran Matematika	Error! Bookmark not defined.
1.5.	iOS (iPhone OS).....	Error! Bookmark not defined.
1.6.	Objective-C	Error! Bookmark not defined.
1.7.	Cocos2D	Error! Bookmark not defined.
1.8.	AI (Kecerdasan Buatan)	Error! Bookmark not defined.
1.8.1.	Sejarah Kecerdasan Buatan.....	Error! Bookmark not defined.
1.9.	Algoritma Growing Tree	Error! Bookmark not defined.
BAB III		Error! Bookmark not defined.
3.1.	Analisa dan Perancangan Sistem.....	Error! Bookmark not defined.
3.1.1.	Keterangan Umum	Error! Bookmark not defined.
3.1.2.	Rancangan Level Permainan.....	Error! Bookmark not defined.
3.2.	Perancangan Growing Tree	Error! Bookmark not defined.
3.3.	Perancangan Aplikasi	Error! Bookmark not defined.
3.3.1.	Keterangan Umum Game.....	Error! Bookmark not defined.
3.3.2.	<i>Storyboard Game</i>	Error! Bookmark not defined.
3.3.3.	Desain Karakter.....	Error! Bookmark not defined.
3.3.4.	Kebutuhan Sistem	Error! Bookmark not defined.
BAB IV		Error! Bookmark not defined.
4.1.	Implementasi	Error! Bookmark not defined.
4.2.	Implementasi Growing Tree.....	Error! Bookmark not defined.
4.2.1.	Pembangkitan Maze	Error! Bookmark not defined.
4.2.2.	Penyelesaian Maze	Error! Bookmark not defined.
4.3.	Merubah Jalur Solusi Menjadi Angka-Angka Pemandu	Error!
	Bookmark not defined.	

4.4.	Implementasi Aplikasi.....	Error! Bookmark not defined.
4.5.	Uji Coba	Error! Bookmark not defined.
4.5.1.	Uji Coba Growing Tree.....	Error! Bookmark not defined.
4.5.2.	Uji Coba Pengguna	Error! Bookmark not defined.
4.6.	Integrasi Islam dengan Growing Tree	Error! Bookmark not defined.
BAB V		Error! Bookmark not defined.
1.1.	Kesimpulan.....	Error! Bookmark not defined.
1.2.	Saran	Error! Bookmark not defined.
DAFTAR PUSTAKA		Error! Bookmark not defined.



DAFTAR GAMBAR

Gambar 2.1. <i>Math Maze</i> 8x8.....	Error! Bookmark not defined.
Gambar 2.2. Lapisan aplikasi iOS	Error! Bookmark not defined.
Gambar 2.3. Lapisan sistem operasi iOS	Error! Bookmark not defined.
Gambar 2.4. <i>Task Domains of Artificial Intelligence</i>	Error! Bookmark not defined.
Gambar 3.1. Flowchart pembangkitan sebuah <i>Maze</i>	Error! Bookmark not defined.
Gambar 3.2. Flowchart pencarian solusi dari <i>Maze</i> yang sudah dibangkitkan	Error! Bookmark not defined.
Gambar 3.3. Flowchart pembuatan papan permainan.....	Error! Bookmark not defined.
Gambar 3.4. <i>Storyboard Game Math Maze</i>	Error! Bookmark not defined.
Gambar 3.6. Flowchart permainan.....	Error! Bookmark not defined.
Gambar 3.5. Desain Karakter Utama	Error! Bookmark not defined.
Gambar 3.6. Desain Karakter Sampingan.....	Error! Bookmark not defined.
Gambar 3.7. <i>Class diagram</i>	Error! Bookmark not defined.
Gambar 4.1. Tampilan utama <i>Math Maze</i>	Error! Bookmark not defined.
Gambar 4.2. Tampilan menu bantuan bagian pertama	Error! Bookmark not defined.
Gambar 4.3. Tampilan menu bantuan bagian kedua.....	Error! Bookmark not defined.
Gambar 4.4. Tampilan menu pengaturan.....	Error! Bookmark not defined.
Gambar 4.5. Tampilan papan permainan pertama	Error! Bookmark not defined.
Gambar 4.6. Tampilan papan permainan kedua.....	Error! Bookmark not defined.
Gambar 4.7. Tampilan ketika kalah	Error! Bookmark not defined.
Gambar 4.8. Tampilan ketika menang	Error! Bookmark not defined.
Gambar 4.3. Grafik presentase responden kelas 1	Error! Bookmark not defined.
Gambar 4.4. Grafik presentase responden kelas 2	Error! Bookmark not defined.
Gambar 4.5. Grafik presentase responden kelas 3	Error! Bookmark not defined.
Gambar 4.6. Grafik presentase responden kelas 4	Error! Bookmark not defined.

Gambar 4.7. Grafik presentase responden secara keseluruhan ...**Error! Bookmark not defined.**



DAFTAR TABEL

Tabel 3.1. Tingkat kesulitan setiap level.....	Error! Bookmark not defined.
Tabel 4.1. Variasi bentuk <i>Maze</i> ukuran 5x5.	Error! Bookmark not defined.
Tabel 4.2. Variasi lintasan dari <i>Maze</i> yang dibangkitkan....	Error! Bookmark not defined.
Tabel 4.3. Pencarian solusi dengan posisi <i>start</i> dan <i>finish</i> yang berbeda	Error! Bookmark not defined.
Tabel 4.4. Tabel tabulasi kuisioner kelas 1	Error! Bookmark not defined.
Tabel 4.5. Tabel tabulasi kuisioner kelas 2	Error! Bookmark not defined.
Tabel 4.6. Tabel tabulasi kuisioner kelas 3	Error! Bookmark not defined.
Tabel 4.7. Tabel tabulasi kuisioner kelas 4	Error! Bookmark not defined.
Tabel 4.8. Tabel tabulasi kuisioner responden secara keseluruhan	Error! Bookmark not defined.

DAFTAR LAMPIRAN

Lampiran 1 Langkah-Langkah Membangun *Game Math Maze* . **Error! Bookmark not defined.**

Lampiran 2 Variasi Lintasan *Maze* Tanpa Dinding **Error! Bookmark not defined.**

Lampiran 3 Data Lengkap Responden Uji Coba ...**Error! Bookmark not defined.**

Lampiran 4 Keterangan Jawaban Kuisioner**Error! Bookmark not defined.**

Lampiran 5 Data Rekapitulasi Hasil Jawaban Kuisioner Responden **Error! Bookmark not defined.**

Lampiran 6 Dokumentasi Uji Coba Responden.....**Error! Bookmark not defined.**

ABSTRAK

Effendi, Moh. Nuruddin. 2013. **Aplikasi Permainan *Math Maze* Menggunakan Algoritma *Growing Tree* Sebagai Pembangkit dan Penyelesai Permainan.** Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Pembimbing: (I) Hani Nurhayati, M.T (II) M. Amin Hariyadi, M.T

Kata kunci: *Math Maze*, *Growing Tree*, Pembangkit, Penyelesai, iOS

Game adalah sesuatu yang dapat dimainkan dengan menggunakan aturan tertentu sehingga ada yang menang dan ada yang kalah. *Game* merupakan hiburan yang sangat menarik bagi seorang anak. Akan tetapi, bermain *game* secara berlebihan dapat memperburuk kemampuan anak. Padahal, saat usia dini perkembangan intelektual anak sangatlah cepat. Atas dasar ini, maka dilakukanlah penelitian dengan tujuan membangun aplikasi permainan *Math Maze* menggunakan algoritma *growing tree* yang memiliki unsur pendidikan di dalamnya.

Metode diperlukan dalam membangun *game* supaya menjadi lebih menarik. Salah satu metode yang dapat digunakan adalah *growing tree*. *Growing tree* merupakan algoritma yang memiliki kemiripan dengan algoritma backtracking apabila selalu memilih *node* terbaru (*newest growing tree*) dan akan mirip seperti algoritma prim's apabila memilih *node* secara random (*random growing tree*). Hasil dari algoritma ini adalah berupa solusi yang akan direpresentasikan dalam bentuk angka-angka pemandu sebagai aturan dalam bermain *Math Maze*.

Dari uji coba yang dilakukan, metode *growing tree* dapat digunakan untuk mencari solusi dan membangkitkan sebuah *Maze* sempurna dengan bentuk yang berbeda-beda, sehingga akan selalu menghasilkan satu solusi. Uji coba juga dilakukan kepada sejumlah pengguna untuk mengetahui penilaian pengguna terhadap aplikasi permainan *Math Maze*.

ABSTRACT

Effendi, Moh. Nuruddin. 2013. **Math Maze Game Applications Using Growing Tree Algorithm For Generating and Solving Game.** Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang, Supervisor: (I) Hani Nurhayati, M.T. (II) M. Amin Hariyadi, M.T.

Keywords: *Math Maze, Growing Tree, Generator, Solver, iOS*

Game is something that can be played using certain rules so that there is winner and loser. Game is exciting entertainment for a child. However, playing games excessively can worsen the child's ability. In fact, at an early age intellectual growth of children is very fast. For that reason, this research was undertaken with the aim of building Math Maze game applications using Growing Tree algorithm that has an element of education on it.

Method is needed to building a game to make the game more interesting. One of the methods that can be used is the growing tree algorithm. Growing tree is an algorithm that mimics the behaviour of the backtracking algorithm if you always choose the newest node (newest growing tree) and mimics the behaviour of the prim's algorithm if you always choose a node at random (random growing tree). The results of this algorithm is represented in the form of guide numbers as the based rule to play Math Maze game.

From the test conducted, growing tree method can be used to find solutions and generate perfect Maze with a different form, so it always found one solution. The test also conducted to a number of users to determine the users rating towards Math Maze game applications

BAB I PENDAHULUAN

1.1. Latar Belakang

Perkembangan kemajuan teknologi komputer saat ini telah berkembang pesat, sehingga banyak aplikasi-aplikasi modern yang telah dikembangkan. Demikian juga dengan *game* (permainan), saat ini tidak hanya terdapat pada permainan konsol saja akan tetapi sudah merambah pada komputer bahkan perangkat *mobile*. Seperti terlansir pada website berita *tekno.kompas.com*, bahwa Kris Antoni Hadiputra (Director Toge Productions) mengatakan “Indonesia merupakan pasar game dengan potensi yang besar. Apalagi mengingat besarnya populasi serta tingginya penetrasi perangkat *mobile*”. Menurut penelitian dalam Psychological Science Maret 2010 yang terlansir dalam website *duniaku.net* (2011), “konsumsi permainan multimedia secara intens memperburuk kemampuan anak untuk membaca dan menulis. Bahkan disebutkan paparan video game membuat fokus anak terpecah karena sibuk memikirkan cara untuk meningkatkan skor permainannya.” Menurut Tom Watson, salah seorang Menteri Sekretaris Kabinet di Inggris yang dilansir dari website *duniaku.net* (2011), bahwa “anak-anak akan mendapatkan pelajaran berharga dari video game ketimbang menonton televisi. Dengan bermain video game, anak-anak dapat belajar melatih pikiran, konsentrasi, menjawab tantangan, dan beradaptasi terhadap perubahan di sekitar mereka.”

Oleh sebab itu, *game* juga bisa dijadikan sebagai hiburan yang menarik dan mendidik, misalnya sebagai media pembelajaran/pendidikan (Randel, 1992). Dalam Al-Qur'an surat Ali Imran ayat 190 disebutkan, bahwa manusia diwajibkan untuk menggunakan akalanya.

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَآخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِأُولِي الْأَلْبَابِ

Artinya:

Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal (Ali Imran:190)

Ibnu Katsir dalam menafsirkan Q.S. Ali Imran ayat 190 menyebutkan bahwa “orang-orang yang berakal itu adalah orang-orang yang cerdas, karena hanya yang demikianlah yang dapat mengetahui segala sesuatu dengan hakikatnya masing-masing secara jelas dan gamblang (Tafsir Ibnu Katsir, juz 14).”

Artificial Intelligence (AI) atau kecerdasan buatan merupakan cabang ilmu komputer yang ditujukan untuk membuat perangkat lunak (*software*) dan perangkat keras (*hardware*) yang bisa sepenuhnya menirukan beberapa fungsi otak manusia. *Game* juga dapat diintegrasikan dengan unsur-unsur psikologi atau logika seperti halnya manusia. Menurut J. M. Randel dalam jurnalnya *The Effectiveness of Games for Educational Purposes*, *game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Dalam *game* juga bisa dimasukkan unsur-unsur pendidikan, misalnya matematika. Dengan menggunakan *game* sebagai media pembelajaran, maka akan memancing minat belajar anak terhadap materi pelajaran. Salah satu permainan yang dijadikan sebagai media pembelajaran adalah permainan *Math Maze*.

Permainan *Math Maze* hanya dimainkan oleh satu orang (*single player*). Permainan ini adalah permainan sederhana yang bertujuan menentukan jalur yang

tepat untuk mencapai tujuan yang telah ditetapkan. Pada dasarnya *Math Maze* merupakan perkembangan dari permainan *Maze* (labirin) dimana dinding-dinding yang terdapat pada labirin dihilangkan dan digantikan oleh angka-angka yang terdapat pada bagian kiri dan bagian atas papan permainan yang disebut sebagai angka pemandu (Teneng, dkk, 2010).

Permainan *Math Maze* telah banyak diteliti oleh para peneliti dengan menggunakan berbagai metode, salah satunya dengan menggunakan metode *backtracking* yang diteliti oleh Teneng, Joko Purwadi, dan Erick Kurniawan pada tahun 2010. Pada penelitian tersebut metode *backtracking* diimplementasikan pada pembangkitan *Maze* dan pembangkitan papan permainan. Selain *backtracking*, salah satu metode yang dapat digunakan untuk pembangkit *Maze* adalah *Growing Tree*. *Growing Tree* adalah suatu metode yang memiliki kemiripan dengan metode *backtracking (newest)* dan *prim's (random)*. Penelitian ini merupakan lanjutan dari penelitian yang telah dilakukan sebelumnya. Dalam penelitian ini metode yang digunakan adalah *Growing Tree (Newest)* sebagai pembangkit dan penyelesai permainan.

Oleh karena itu, diambil keputusan bahwa penelitian yang ingin dilakukan berjudul “***Aplikasi Permainan Math Maze Menggunakan Algoritma Growing Tree Sebagai Pembangkit dan Penyelesai Permainan***”, yaitu rancang bangun aplikasi permainan *Math Maze* berbasis perangkat *mobile* dengan menggunakan *platform* iOS.

1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang, maka perumusan masalah dalam penelitian ini, yaitu: Bagaimana membuat aplikasi permainan *Math Maze* menggunakan algoritma *Growing Tree*?

1.3. Batasan Masalah

Batasan masalah dalam pembuatan aplikasi ini meliputi:

- a. Bahasa pemrograman yang digunakan adalah *Objective-C*.
- b. *Game* yang dibangun bersifat *single player* dan berjenis *Maze*.
- c. Dimensi kotak *Math Maze* adalah 3x3.
- d. Algoritma yang digunakan adalah *Growing Tree (Newest)*.
- e. *Growing Tree (Newest)* digunakan sebagai pembangkit dan penyelesai *Maze*.
- f. Ditujukan kepada anak usia 7-8 tahun.
- g. *Platform* yang digunakan adalah iOS.

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk membuat suatu aplikasi permainan *Math Maze* menggunakan algoritma *growing tree* sebagai pembangkit dan penyelesai permainan.

1.5. Manfaat Penelitian

Menjadi salah satu permainan yang mampu mengasah logika dengan hiburan yang menarik. Serta mengembangkan permainan *Math Maze* sebagai permainan edukasi untuk menumbuhkan kembali motivasi belajar anak, khususnya pelajaran matematika.

1.6. Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, metodologi dan sistematika penyusunan laporan tugas akhir. Uraian di bab pendahuluan ini memberikan gambaran kepada pembaca terkait maksud dan tujuan dalam penelitian ini. Berangkat dari pemahaman bahwa *game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Hal tersebut dijawab dengan merancang dan membangun sebuah aplikasi permainan *Math Maze* menggunakan metode *growing tree* yang akan dipaparkan lebih lanjut di bab selanjutnya.

BAB II Tinjauan Pustaka

Bab ini menjelaskan tentang teori-teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah tentang teori yang berkaitan dengan pembahasan tentang aplikasi permainan *Math Maze*, pembelajaran matematika, dan metode *growing tree*. Teori yang dijabarkan ini yang akan menjadi acuan dalam melakukan perancangan sistem yang akan dibahas di bab III.

BAB III Analisa dan Perancangan Sistem

Bab ini menjelaskan tentang analisa perancangan sistem untuk membuat aplikasi permainan *Math Maze*. Perancangan aplikasi terdiri atas perancangan proses-proses utama dan desain aplikasi yang terdiri atas desain output,

desain input, desain antar muka dan desain karakter. Terdapat beberapa proses utama yaitu: pembangkitan sebuah *Maze*, pencarian solusi dari *Maze* yang telah dibangkitkan, dan membuat papan permainan *Math Maze*.

BAB IV Hasil dan Pembahasan

Menjelaskan tentang pengujian aplikasi permainan *Math Maze*. Perancangan dan desain aplikasi diimplementasikan dengan bahasa pemrograman Objective-C yang dibangun dengan IDE Xcode. Juga dipaparkan pengujian terhadap aplikasi yang dibuat untuk mengetahui aplikasi tersebut telah dapat menyelesaikan permasalahan sesuai dengan yang diharapkan. Uji coba dan evaluasi dilakukan terhadap tahapan aplikasi dan solusi yang didapatkan dalam melakukan penyelesaian *Maze*.

BAB V Penutup

Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan pembuatan aplikasi permainan *Math Maze* selanjutnya.

1.7. Metode Penelitian

Untuk mencapai tujuan yang telah dirumuskan, maka metode penelitian yang dilakukan dalam penulisan skripsi ini adalah *library research*, yaitu suatu metode penelitian dengan cara melakukan pengumpulan data secara teoritis dari buku-buku, artikel-artikel, jurnal-jurnal, dan berbagai literatur untuk mendukung penyusunan skripsi. Pengumpulan data juga dilakukan dengan uji coba aplikasi pada pengguna yang disertai dengan angket.



BAB II TINJAUAN PUSTAKA

2.1. Permainan dan Jenisnya

2.1.1. Permainan (*Game*)

Game merupakan kata dalam bahasa Inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah. Permainan merupakan bagian dari bermain dan bermain merupakan bagian dari permainan, keduanya saling berhubungan. permainan adalah sistem dimana pemain terlibat dalam konflik buatan yang terjadi dalam permainan yang merupakan rekayasa. *Game* adalah program komputer sangat kompleks yang merangsang otak untuk melakukan serangkaian tugas kognitif dan menghasilkan tingkat pemikiran yang lebih tinggi (Beck dan Wade, 2006).

Game memiliki peraturan yang bertujuan untuk menentukan alur dari permainan dan membatasi perilaku dari pemain. *Game* dapat melatih perkembangan otak, untuk meningkatkan konsentrasi, dan melatih untuk memecahkan masalah dengan tepat dan cepat. Saat ini *game* telah berkembang pesat, yang awalnya hanya terdapat pada konsol sekarang telah merambah pada komputer dan telepon selular. *Game* yang terdapat pada telepon selular disebut sebagai *mobile game*.

Mobile game adalah *game* yang dimainkan pada perangkat genggam, seperti ponsel dan PDA dengan fungsionalitas komunikasi nirkabel (Jeong dan J.Kim, 2009). Menurut Barbara Gruter dan Miriam Oks dalam jurnalnya *On the Physical*

Dimension of Social Interaction in Mobile Games, definisi *mobile game* terdiri dari 3 karakteristik:

1. Secara fisik pemain bergerak dalam dunia permainan.
2. Permainan berlangsung dalam dunia permainan, dimana dunia nyata dan dunia virtual bergabung menjadi satu.
3. Teknologi komputasi *game* dapat bekerja, baik pada pemain maupun kondisi objek.

2.1.2. Jenis-Jenis *Game*

Game menurut Lindsay Grace (2005) dapat diklasifikasikan menjadi beberapa jenis, antara lain:

1. *Action Game*

Game jenis ini adalah *game* yang menawarkan aksi sebagai daya tarik utama. Respon merupakan keterampilan utama yang dibutuhkan dalam memainkan *game* ini. Contoh dari *game* ini adalah Metal Gear.

2. *Adventure Game*

Game jenis ini adalah *game* dengan mengambil konsep petualangan. Pemain berjalan menuju ke suatu tempat dan disepanjang perjalanan pemain menemukan hal-hal baru untuk dieksplorasi. Sebagai contohnya adalah *game* Crash Bandicoot.

3. *Puzzle Game*

Jenis *game* ini memberikan tantangan kepada pemainnya dengan cara memecahkan masalah yang terdapat pada *game* yang dimainkan. Teka-teki merupakan daya tarik utama dalam *game* ini. Contoh dari jenis *game* ini adalah jigsaw.

4. *Role Playing Game*

Game bertipe ini ini lebih mengutamakan cerita dan biasanya pemain diajak masuk ke dalam cerita tersebut untuk menyelesaikan sebuah misi. Dalam *game* ini pemain akan berperan menjadi sebuah karakter dengan berbagai atribut, seperti kesehatan, intelegensi, kekuatan dan keahlian. Salah satu *game* yang terkenal dengan *RPG* adalah Final Fantasy.

5. *Simulation Game*

Game ini merupakan *game* yang mengambil simulasi seperti keadaan nyata. Dalam beberapa jenis *game* ini pemain diajak untuk menciptakan lingkungan yang diinginkan, seperti membangun kota. Contoh *game* yang populer dari jenis ini adalah The Sims.

6. *Strategy Game*

Merupakan jenis *game* yang melibatkan strategi dan logika melalui pemecahan masalah. Contoh dari *game* ini adalah *game Civilization*.

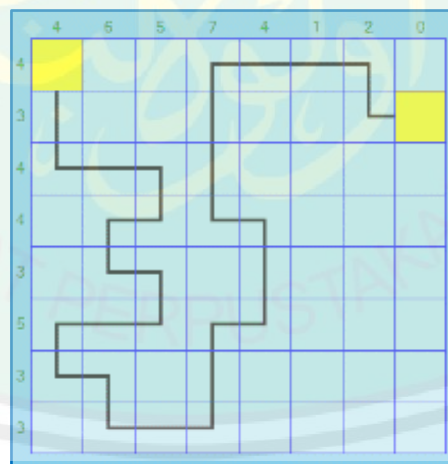
2.2. Permainan *Math Maze*

Math Maze merupakan permainan sederhana yang bertujuan menentukan jalur yang tepat untuk mencapai tujuan yang telah ditetapkan. Pada dasarnya *Math Maze* adalah permainan yang merupakan perkembangan dari permainan *Maze* (labirin). Perbedaannya adalah permainan ini menggunakan angka-angka yang terdapat pada bagian kiri dan bagian atas papan permainan yang disebut sebagai angka-angka pemandu. Angka-angka tersebut merupakan indikator yang menyatakan berapa banyak sel yang dilalui oleh jalur yang menuju tujuan akhir (Teneng, dkk, 2010). Dengan nilai pemandu pada bagian kiri dan atas papan permainan, maka pemain harus memperkirakan sel mana yang harus dipilih.

Apabila jumlah baris dan kolom tidak sesuai saat mencapai garis, maka pemain dianggap gagal menemukan tujuan.

Permainan *Math Maze* ini mempunyai beragam bentuk berdasarkan banyaknya kotak *Math Maze*, mulai dari 3x3, 5x5, 7x7, dan sebagainya. Berikut ini adalah peraturan permainan *Math Maze*:

1. Angka-angka pada bagian kiri dan atas papan permainan digunakan sebagai petunjuk untuk menemukan tujuan akhir.
2. Pemain hanya diperbolehkan memilih 4 jalur, yaitu atas, bawah, kanan, dan kiri.
3. Pemain tidak diperbolehkan memilih jalur secara diagonal.
4. Apabila pemain berhasil menemukan jalur untuk menuju tujuan akhir, maka pemain dianggap menang. Jika tidak, maka pemain dianggap kalah.



Gambar 2.1. *Math Maze* 8x8
(sumber: <http://www.codeproject.com/Articles/13611/Math-Maze>)

2.3. Pendidikan Anak Usia Dini

Pendidikan bertujuan untuk membentuk bangsa yang cerdas dan bermutu. Pendidikan orang tua terhadap anak akan sangat berpengaruh terhadap kreativitas anak. Menurut Alim Sumarno, anak yang memiliki bakat tertentu jika tidak

diberikan rangsangan-rangsangan atau motivasi dari orang tua dan lingkungannya, tidak akan mampu memelihara, apalagi mengembangkan bakatnya. Hal ini dapat diminimalkan dengan mengoptimalkan pendidikan anak sejak dini.

Menurut Prof. Dr. Utami Munandar, seorang pakar kreativitas Indonesia:

“Kapasitas otak anak pada usia 6 bulan sudah mencapai sekitar 50% dari keseluruhan potensi orang dewasa. Otak seorang anak ternyata sangat luar biasa. Pada masa ini, anak mengalami perkembangan intelektual otak yang sangat cepat. Tingkat perkembangan intelektual otak anak dari sejak lahir sampai usia 4 tahun mencapai 50%. Oleh karena itu, pada masa empat tahun pertama ini sering disebut juga sebagai Golden Age (Masa Keemasan), karena si anak mampu menyerap dengan cepat setiap rangsangan yang masuk. Anak akan mampu menghafal banyak sekali informasi, seperti perbendaharaan kata, nada, dan sebagainya. Hingga usia 8 tahun, anak telah memiliki intelektual otak sekitar 80%”.(Sumarno, 2011)

Melihat anak sekolah dasar yang lebih senang bermain dan masa keemasan anak terjadi pada usia muda, serta kapasitas intelektual otak anak yang mencapai 80% pada usia 8 tahun. Hendaknya diangkat menjadi sebuah *game* edukasi yang ditujukan untuk anak usia sekolah dasar, mengingat *game* yang ada saat ini banyak memuat konten yang tidak jelas tujuannya dan cenderung kepada kekerasan.

2.4. Pelajaran Matematika

Matematika merupakan salah satu bidang studi yang diajarkan di sekolah dasar (SD). Matematika berasal dari kata Yunani *mathematika* yang berarti mempelajari. Kata matematika berhubungan pula dengan kata lainnya, yaitu *mathein* atau *mathenein* yang artinya belajar (berpikir). Matematika lebih menekankan kegiatan dalam dunia rasional (penalaran), bukan menekankan dari hasil eksperimen atau hasil observasi. Pengertian matematika menurut para ahli berbeda-beda, beberapa pengertian matematika menurut para ahli:

Matematika adalah sebagai ilmu deduktif, bahasa seni, ratunya ilmu, ilmu tentang struktur yang terorganisasikan dan ilmu tentang pola dan hubungannya. (Ruseffendi, 1997).

Soedjadi (2000) memberikan enam definisi tentang matematika, yaitu:

- a. Matematika adalah cabang ilmu pengetahuan eksak dan terorganisir dengan baik.
- b. Matematika adalah ilmu tentang bilangan dan kalkulasi.
- c. Matematika adalah pengetahuan tentang penalaran logis dan berhubungan dengan baik.
- d. Matematika adalah ilmu pengetahuan fakta-fakta kuantitatif dan masalah tentang ruang bentuk.
- e. Matematika adalah ilmu pengetahuan struktur-struktur yang logik.
- f. Matematika adalah aturan-aturan yang ketat.

Sigit Dwi Kusrahmadi dalam jurnalnya *Dinamika Pendidikan* No. 1 / Th. XIV, mengatakan bahwa pendidikan pada anak usia sekolah dasar (sekitar umur 6-12 tahun) merupakan suatu hal yang fundamental bagi kesuksesan perkembangan pendidikan selanjutnya. Dalam jurnal tersebut, beliau juga menuliskan bahwa ada beberapa perkembangan dan karakteristik anak usia sekolah dasar, antara lain:

- a. Mereka (anak usia SD) secara alamiah memiliki rasa ingin tahu yang kuat dan tertarik akan dunia sekitar yang mengelilingi mereka sendiri.
- b. Anak usia dasar lebih senang bermain dan lebih suka bergembira.

- c. Anak SD bisa tergetar perasaannya dan terdorong untuk berprestasi sebagaimana mereka mengalami ketidakpuasan dan menolak kegagalan-kegagalan.
- d. Anak SD suka mengatur dirinya untuk menangani berbagai hal, mengeksplorasi sesuatu dan mencoba hal-hal baru.
- e. Anak SD belajar dengan cara bekerja, mengobservasi, berinisiatif, dan mengajar anak-anak lainnya.

Menurut Ruseffendi (2006), minat anak terhadap matematika merupakan salah satu faktor untuk mengetahui sikap anak terhadap matematika. Artinya seorang anak yang berminat pada matematika akan menumbuhkan sikap positif terhadap matematika. Untuk menumbuhkan minat dan sikap positif inilah maka diperlukan antara lain kegunaan matematika bagi kehidupan sehari-hari dan bagaimana cara guru menyampaikan matematika kepada siswa. Cara guru menyampaikan matematika haruslah menarik, sehingga bisa menumbuhkan minat siswa terhadap matematika. Misalnya memahami operasi logis dengan bantuan benda-benda konkrit ataupun alat peraga yang nantinya akan menghasilkan pengalaman-pengalaman berpikir sehingga akan memberikan kesanggupan kepada siswa untuk memecahkan masalah persoalan dalam kehidupan sehari-hari.

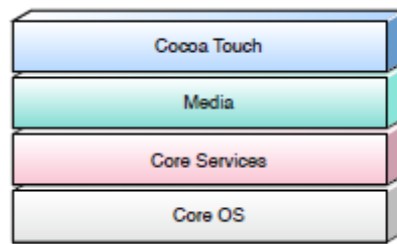
Konsep pendidikan matematika ini bisa diimplementasikan dalam sebuah *game* edukasi, dengan harapan motivasi belajar siswa terhadap pembelajaran matematika bisa meningkat. *Mobile game* yang mulai digandrungi kalangan anak-anak hingga orang dewasa ini bisa menjadi ladang yang subur untuk meningkatkan pendidikan menjadi lebih interaktif.

2.5. iOS (iPhone OS)

iOS merupakan sistem operasi yang berjalan pada perangkat keras yang dibuat dan dikembangkan oleh perusahaan APPLE Inc. Arsitektur iOS mirip dengan arsitektur dasar yang dimiliki oleh MAC OS X (Sistem Operasi milik APPLE Inc.). Lapisan aplikasi pada sistem operasi iOS dapat dilihat pada gambar 2.1. Sedangkan lapisan pada sistem operasi iOS dapat dilihat pada gambar 2.2. Lapisan sistem paling bawah adalah lapisan dimana layanan fundamental dan teknologi dari aplikasi bergantung. Dan pada lapisan yang lebih tinggi, mengandung teknologi yang lebih mutakhir daripada lapisan yang berada dibawahnya (Apple Inc Developer, 2011.).



Gambar 2.2. Lapisan aplikasi iOS
(sumber: Apple Inc. Developer, iOS Technology Review)



Gambar 2.3. Lapisan sistem operasi iOS
(sumber: Apple Inc. Developer, iOS Technology Review)

Terdapat empat lapisan pada sistem operasi iOS, yaitu:

1. *Cocoa Touch Layer*

Berisi *framework* untuk membangun aplikasi iOS. Lapisan ini mendefinisikan aplikasi infrastruktur dan dukungan untuk teknologi seperti *multitasking*, input berbasis sentuhan, pemberitahuan dan sebagainya.

2. *Media Layer*

Berisi teknologi grafis, audio, dan video untuk membangun pengalaman multimedia terbaik di dalam perangkat *mobile*.

3. *Core Service*

Berisi sistem servis fundamental yang dibutuhkan oleh semua aplikasi.

4. *Core Os*

Berisi fitur-fitur dengan tingkat level yang rendah dan digunakan untuk berurusan dengan masalah keamanan dan perangkat keras.

2.6. Objective-C

Objective-C adalah bahasa pemrograman yang merupakan turunan dari bahasa pemrograman C. Objective-C adalah salah satu bahasa pemrograman yang mendukung *object-oriented programming* (bahasa pemrograman berorientasi objek). Gaya bahasa pemrograman ini adalah mengorganisasikan program menjadi koleksi-koleksi objek yang saling terhubung antara objek yang satu

dengan objek yang lain. Sebuah objek adalah sekumpulan variabel yang memodelkan sesuatu dalam suatu prosedur. Masing-masing objek berkomunikasi dengan cara mengirimkan pesan untuk melakukan eksekusi terhadap *method* yang terdapat pada objek lain (Kochan, 2009)

2.7. Cocos2D

Cocos2D adalah framework open source untuk bahasa pemrograman Objective-C yang digunakan untuk membuat permainan 2 dimensi pada iOS dan Mac OS X. Cocos2D menggunakan OpenGL ES dalam pembentukan grafisnya, sehingga kecepatan dan performa yang baik pada perangkat iOS. Sebelum framework ini tersedia untuk bahasa pemrograman Objective-C, Cocos2D lebih dahulu terdapat pada bahasa pemrograman Python. Akhirnya pada akhir tahun 2008, Cocos2D resmi dibuka untuk bahasa pemrograman Objective-C. Dan pada saat ini, sudah cukup banyak bahasa pemrograman yang menggunakan *framework* ini, contohnya Ruby, Java (Android), dan Mono (C#/.Net) (Strougo dan Wenderlich, 2011).

2.8. AI (Kecerdasan Buatan)

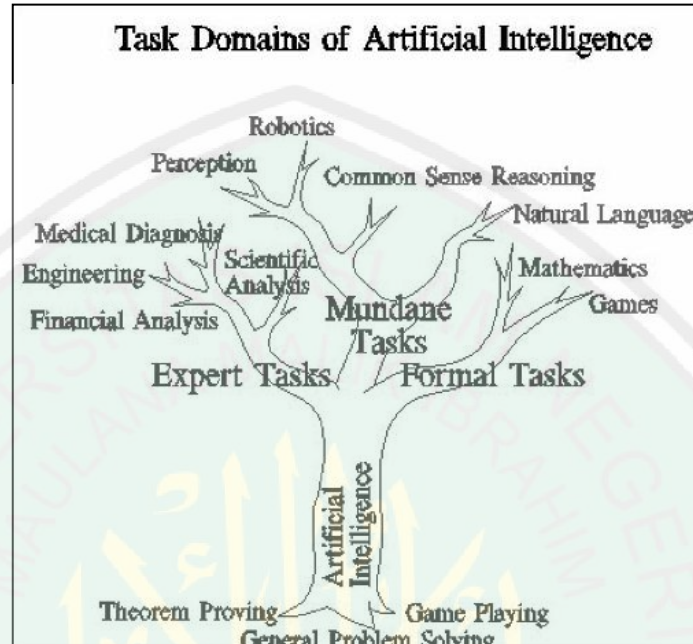
Definisi *Artificial Intelligence* (AI) menurut Ian Millington dalam bukunya *Artificial Intelligence for Games* adalah “membuat komputer untuk bisa melakukan pekerjaan seperti manusia ataupun hewan”. *Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang ilmu komputer yang ditujukan untuk membuat perangkat lunak (*software*) dan perangkat keras (*hardware*) yang bisa sepenuhnya menirukan beberapa fungsi otak manusia.. AI dapat dipandang dalam berbagai perspektif, antara lain:

- a. Dari perspektif Kecerdasan (*Intelligence*) AI adalah bagaimana membuat mesin yang “cerdas” dan dapat melakukan hal-hal yang sebelumnya dapat dilakukan oleh manusia
- b. Dari perspektif bisnis, AI adalah sekelompok alat bantu (*tools*) yang berdaya guna, dan metodologi yang menggunakan tool-tool tersebut guna menyelesaikan masalah-masalah bisnis.
- c. Dari perspektif pemrograman (*Programming*), AI termasuk didalamnya adalah studi tentang pemrograman simbolik, pemecahan masalah, proses pencarian (*search*).
- d. Dari perspektif penelitian (*research*), Riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan (*game*) catur, membuktikan teori, dan general problem solving (untuk tugas-tugas sederhana).

2.8.1. Sejarah Kecerdasan Buatan

Menurut Tyler Cowen and Michelle Dawson, pada tahun 1950-an Alan Turing, seorang pionir AI dan ahli matematika Inggris melakukan percobaan Turing (*Turing Test*) yaitu sebuah komputer melalui terminalnya ditempatkan pada jarak jauh. Di ujung yang satu ada terminal dengan software AI dan diujung lain ada sebuah terminal dengan seorang operator. Operator itu tidak mengetahui kalau di ujung terminal lain dipasang software AI. Mereka berkomunikasi dimana terminal di ujung memberikan respon terhadap serangkaian pertanyaan yang diajukan oleh operator. Dan sang operator itu mengira bahwa ia sedang berkomunikasi dengan operator lainnya yang berada pada terminal lain. Turing beranggapan bahwa jika mesin dapat membuat seseorang percaya bahwa dirinya

mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia).



Gambar 2.4. Task Domains of Artificial Intelligence
(sumber: Silfianti, Pengantar Kecerdasan Buatan)

2.9. Algoritma Growing Tree

Algoritma *growing tree* merupakan algoritma dengan struktur *tree* yang memiliki kemiripan dengan algoritma *backtracking* dan *prim's*. Terdapat dua jenis dari algoritma ini, yaitu *growing tree (newest)* yang selalu memilih *node* (titik) paling baru dari *node* sebelumnya dan *growing tree (random)* yang memilih *node* secara random dari semua *node* yang ada (Buck, 2011).

Berikut ini adalah jenis dari algoritma *growing tree*:

b. Growing Tree (Newest)

1. Masukkan *graph* terhubung dengan *vertices* V dan *edges* E .
2. Inisialisasi $V_{\text{new}} = \{X\}$, dimana X adalah *vertices* dari V , $E_{\text{new}} = \{\}$.
3. Pilih sebuah *edge* (u,v) secara random dari V_{new} .

4. Ulangi langkah 2,3 sampai $V_{\text{new}} = V$.
5. Tambahkan v pada V_{new} dan (u,v) pada E_{new} .

c. *Growing Tree (Random)*

1. Masukkan *graph* terhubung dengan *vertices* V dan *edges* E .
2. Inisialisasi $V_{\text{new}} = \{X\}$, dimana X adalah *vertices* dari V , $E_{\text{new}} = \{\}$.
3. Pilih sebuah *edge* (u,v) secara random.
4. Ulangi langkah 2,3 sampai $V_{\text{new}} = V$.
5. Tambahkan v pada V_{new} dan (u,v) pada E_{new} .





BAB III

ANALISA DAN PERANCANGAN SISTEM

1.1. Analisa dan Perancangan Sistem

Pada penelitian ini akan dibangun sebuah aplikasi permainan *Math Maze*. *Math Maze* merupakan permainan logika berbasis *Maze* dan angka. Tujuan dari permainan ini adalah untuk menuju tujuan akhir dengan menggunakan panduan angka yang terdapat pada bagian kiri dan bagian atas papan permainan. Pada penelitian ini, permainan *Math Maze* yang pada dasarnya hanya menggunakan garis sebagai jalur akan digantikan oleh karakter yang akan berjalan menuju tujuan akhir melalui *Maze* tersebut. Berikut ini desain dari permainan *Math Maze*:

1.1.1. Keterangan Umum

Permainan *Math Maze* ini dikembangkan pada platform iOS yang merupakan sistem operasi berbasis *mobile*. Hal ini bertujuan untuk mempermudah pemain dalam bermain *Math Maze* yang dapat dimainkan kapan saja dan dimana saja. Permainan *Math Maze* yang merupakan permainan logika dan berjenis *Maze* terkadang menghabiskan waktu lama untuk menemukan jalur menuju tujuan akhir, sehingga pengembangan permainan menggunakan sistem yang berbasis *mobile* dapat membantunya.

Dalam permainan ini terdapat unsur pendidikan. Pemain dikenalkan dengan angka yang tidak hanya digunakan untuk menyelesaikan rumus matematika, tetapi dapat juga digunakan sebagai penyelesai suatu permasalahan. Permainan ini secara tidak langsung juga mengajak pemain untuk mengasah logikanya dalam menyelesaikan sebuah *Maze* yang menggunakan angka sebagai penyelesai

permainan. Penggunaan logika sangat penting untuk menyelesaikan permainan ini. Sedangkan untuk objek penelitiannya, yaitu penggunaan metode *growing tree* dalam pembangkitan dan penyelesaian *Maze*, sehingga pemain tidak merasa bosan dengan *Maze* yang sama.

1.1.2. Rancangan Level Permainan

Pada permainan ini terdapat tiga level yang dibentuk berdasarkan tingkat kesulitan, waktu serta jumlah langkah yang tersedia untuk menyelesaikan *Math Maze*. Berikut ini tingkat kesulitan dalam permainan ini:

Tabel 3.1. Tingkat kesulitan setiap level

Tingkat Kesulitan	Waktu	Jumlah Langkah
Mudah	3 menit	Jumlah langkah = total jalur solusi X 2
Sedang	2 menit	Jumlah langkah = total jalur solusi + (total jalur solusi / 2)
Sulit	1 menit	Jumlah langkah = total jalur solusi

Pada level mudah waktu yang tersedia adalah 3 menit dan jumlah langkah yang tersedia adalah jumlah jalur solusi X 2. Sedangkan pada level sedang waktu yang tersedia adalah 2 menit dan jumlah langkah yang tersedia adalah jumlah jalur solusi + (jumlah jalur solusi / 2). Untuk tingkat level sulit waktu yang tersedia adalah 1 menit dan jumlah langkah yang tersedia adalah sama dengan jumlah jalur solusi. Pada setiap level juga terdapat skor yang akan bertambah apabila pemain mengambil *item* (buah apel) yang terdapat pada jalur solusi.

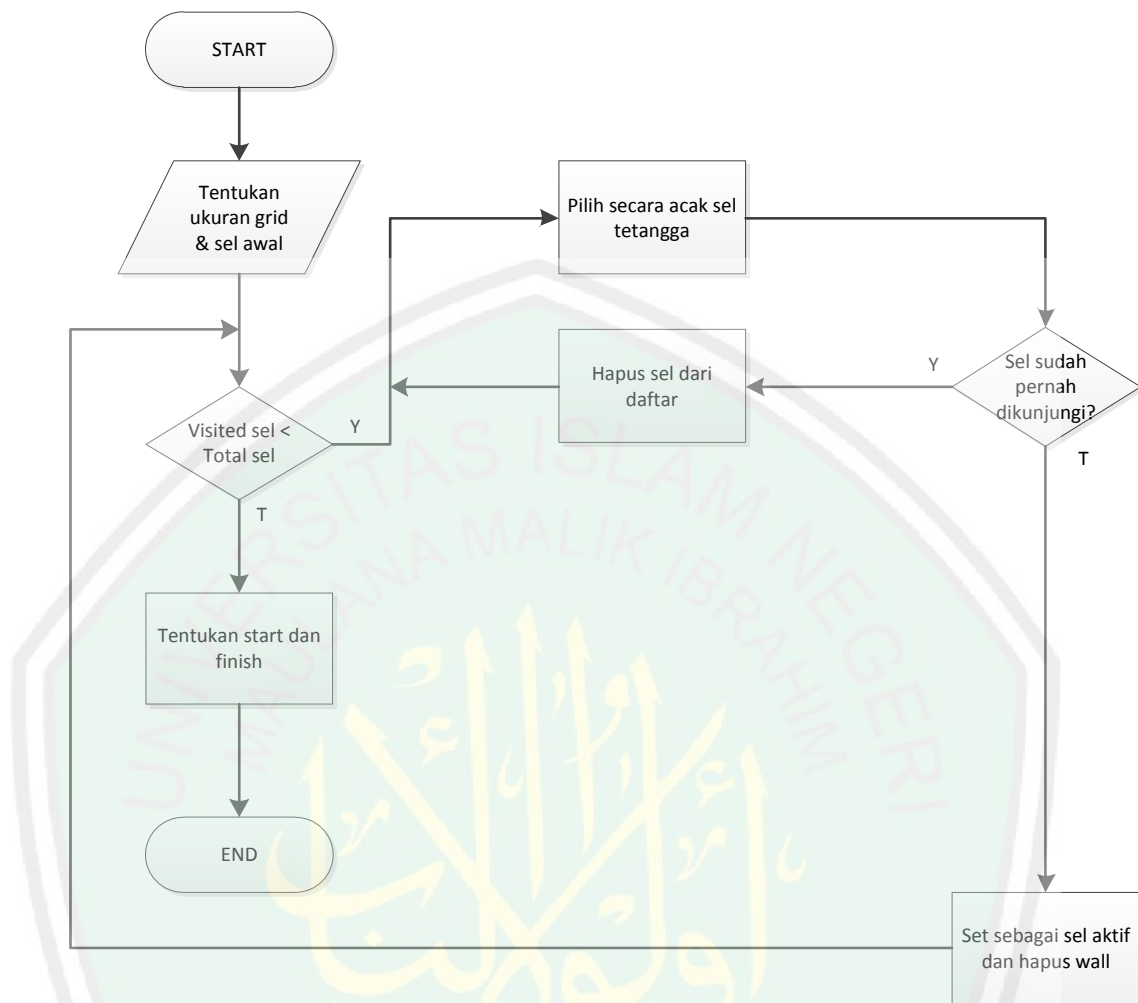
1.2. Perancangan Growing Tree

Permasalahan dalam permainan *Math Maze* adalah bagaimana membangun *Maze* dan mencari solusi dari *Maze* yang dibangkitkan dan dari hasil solusi yang ditemukan akan dijadikan sebagai angka-angka pemandu dalam memainkan *Math Maze*. Terdapat tiga proses utama dalam membangun permainan *Math Maze*, yaitu pembangkitan sebuah *Maze*, pencarian solusi dari *Maze* yang telah dibangkitkan, dan membuat papan permainan *Math Maze*. Algoritma *growing tree* digunakan untuk menentukan dua dari tiga proses utama, yaitu membangkitkan sebuah *Maze* dan mencari solusi dari *Maze* yang telah dibangkitkan.

1. Pembangkitan sebuah *Maze*.

Proses pembangkitan sebuah *Maze* yang dilakukan dengan *growing tree* adalah dengan prosedur berikut:

- a. Tentukan ukuran *grid* dan pilih salah satu sel pada *grid*, yang kemudian disebut sebagai sel aktif.
- b. Pilih secara acak sel terdekat atau sel tetangga dengan gerakan yang diperbolehkan adalah horizontal dan vertikal. Tidak diperbolehkan memilih sel tetangga secara diagonal.
- c. Jika sel tetangga belum pernah didatangi, maka pindah ke sel tersebut (sel aktif). Jika tidak ada sel tetangga yang didatangi, maka hapus sel tersebut (sel tidak aktif).
- d. Lakukan langkah ini sampai semua sel pada *grid* sudah didatangi.
- e. Tentukan posisi *start* dan *finish*.



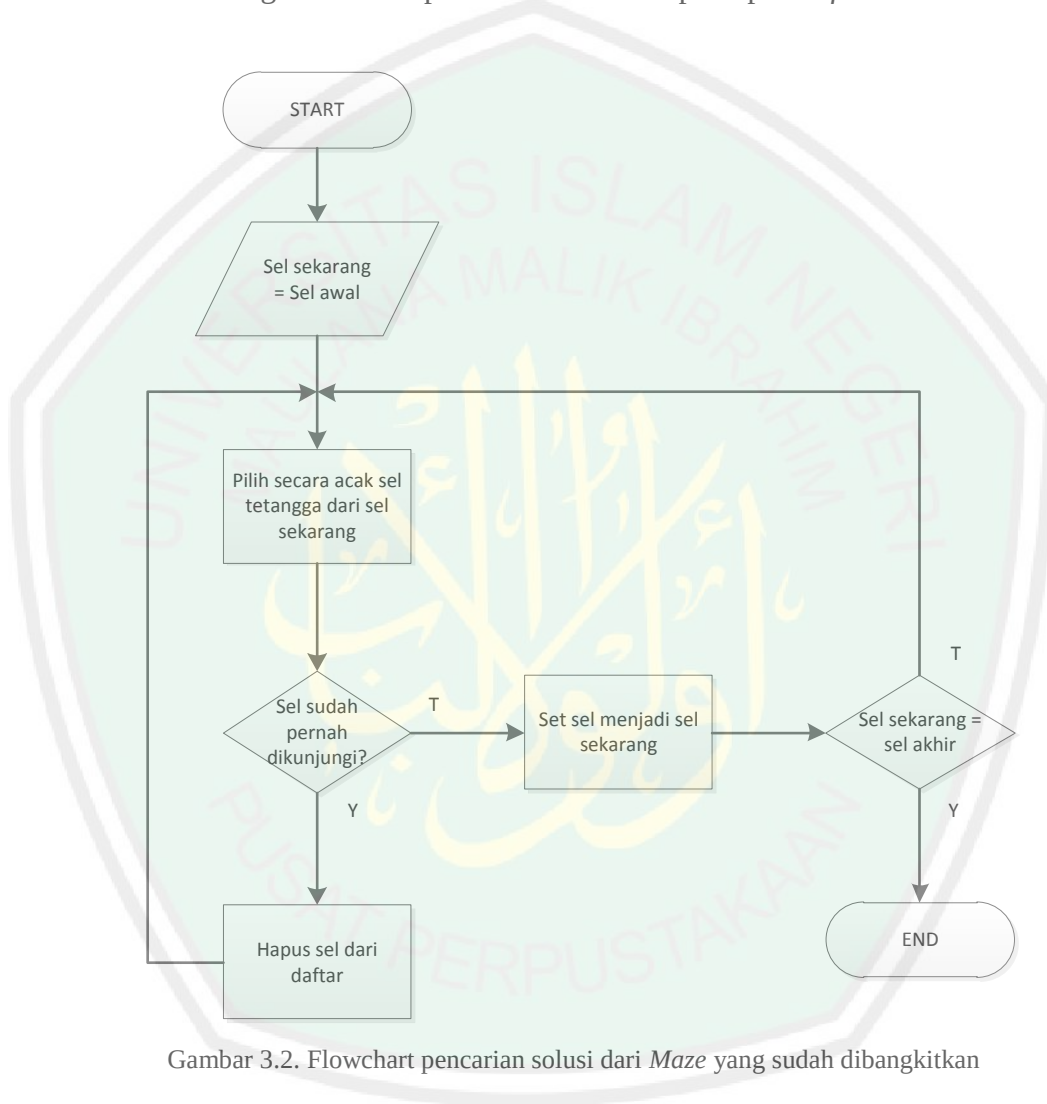
Gambar 3.1. Flowchart pembangkitan sebuah Maze

2. Pencarian solusi dari Maze yang sudah dibangkitkan.

Apabila Maze sudah dibangkitkan maka selanjutnya adalah mencari solusinya. Untuk mencari solusinya hampir sama dengan langkah-langkah ketika Maze dibangkitkan. Langkah-langkah yang dilakukan adalah sebagai berikut:

- a. Dari sel *start*, pilih secara acak sel tetangga terdekat yang bisa dikunjungi (sel aktif). Gerakan yang diperbolehkan adalah secara horizontal dan vertikal.

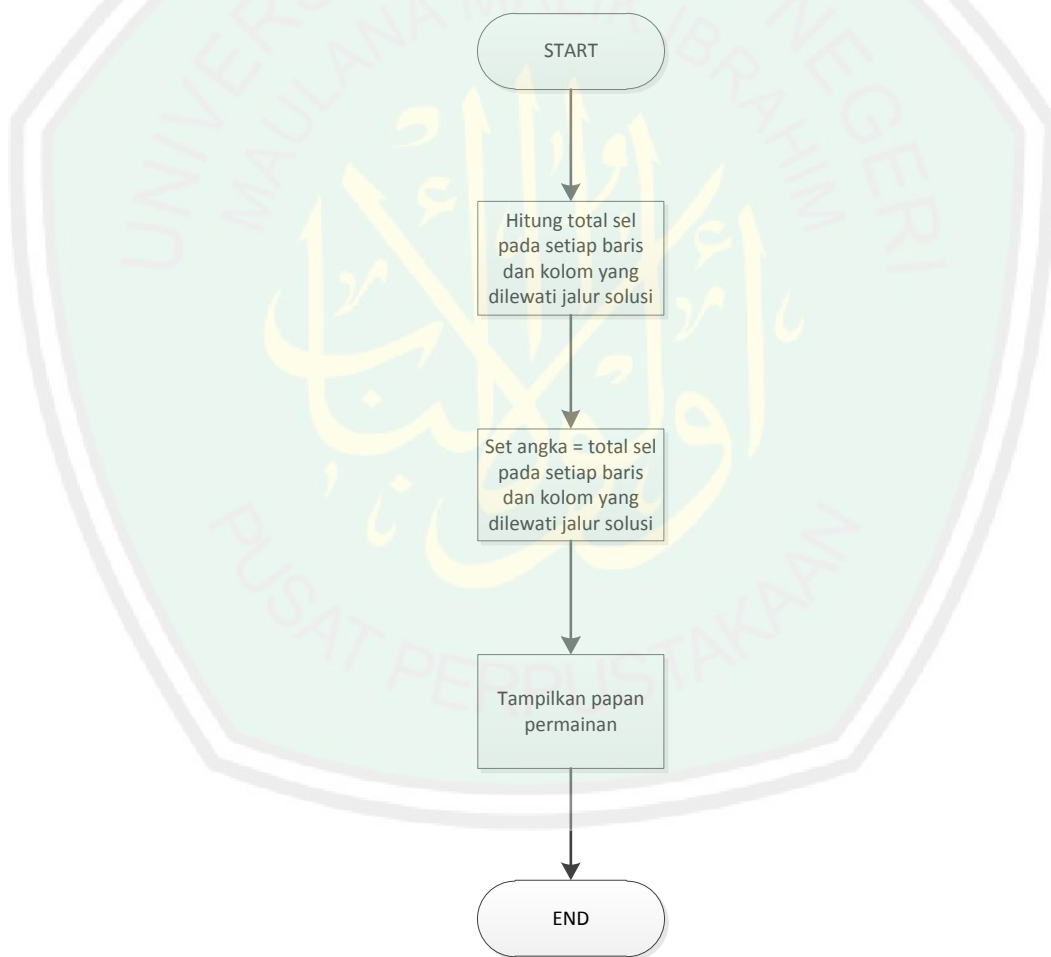
- b. Apabila sel tetangga tersebut belum pernah didatangi, maka pindah ke sel tersebut (sel aktif). Jika tidak ada sel tetangga yang didatangi, maka hapus sel tersebut (sel tidak aktif).
- c. Lakukan langkah ini sampai menemukan sel pada posisi *finish*.



Gambar 3.2. Flowchart pencarian solusi dari *Maze* yang sudah dibangkitkan

3. Membuat papan permainan *Math Maze*.

Proses ini adalah proses pembuatan papan permainan *Math Maze* dengan memanfaatkan *Maze* yang sudah dibangkitkan dan solusinya sudah ditemukan. Langkah-langkahnya adalah menghitung total sel pada baris dan kolom yang dilewati oleh jalur solusi. Jumlah sel pada baris dan kolom itulah yang nantinya akan menjadi angka-angka sebagai panduan bermain dalam permainan *Math Maze*.



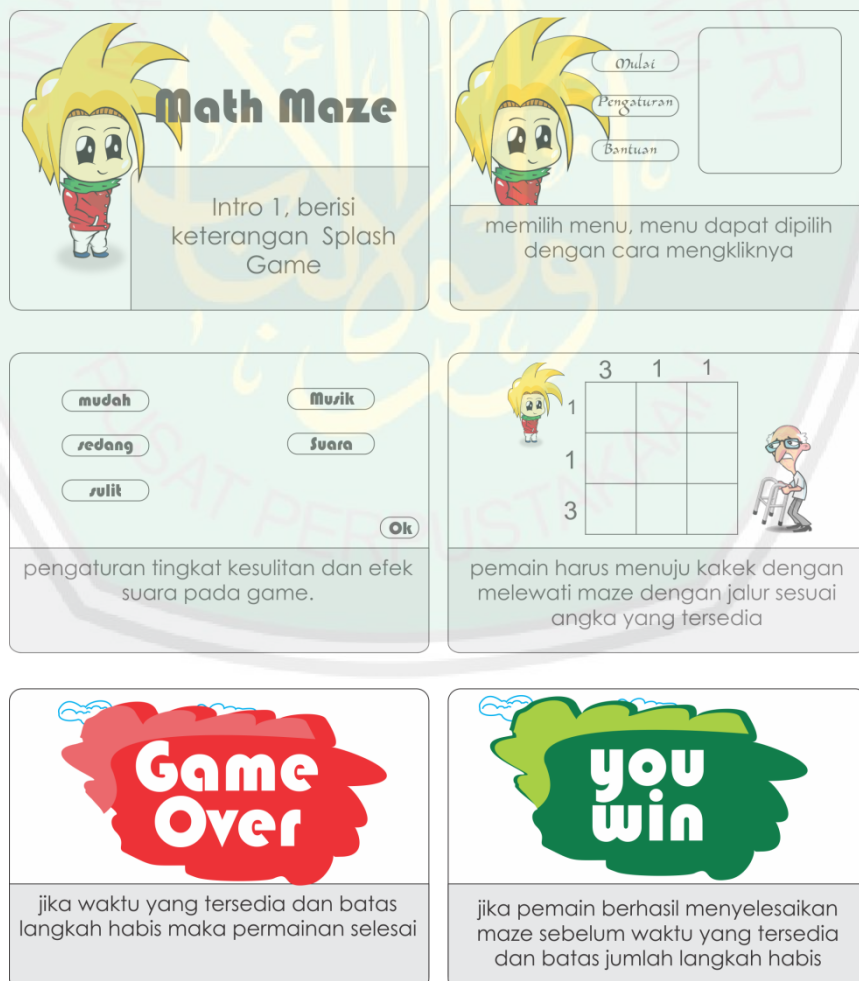
Gambar 3.3. Flowchart pembuatan papan permainan

3.3. Perancangan Aplikasi

3.3.1. Keterangan Umum Game

Permainan ini ber-genre *board game* (permainan papan) akan tetapi terdapat karakter yang dapat dijalankan oleh pemain, sehingga permainan menjadi lebih menarik. Lingkungan *game* berbentuk *Maze* yang terletak di hutan. Pemain hanya memiliki satu tujuan, yaitu melewati *Maze* menggunakan angka-angka pemandu yang tersedia sebelum waktu yang tersedia habis untuk menemui Kakek yang berada di luar *Maze*.

3.3.2. Storyboard Game

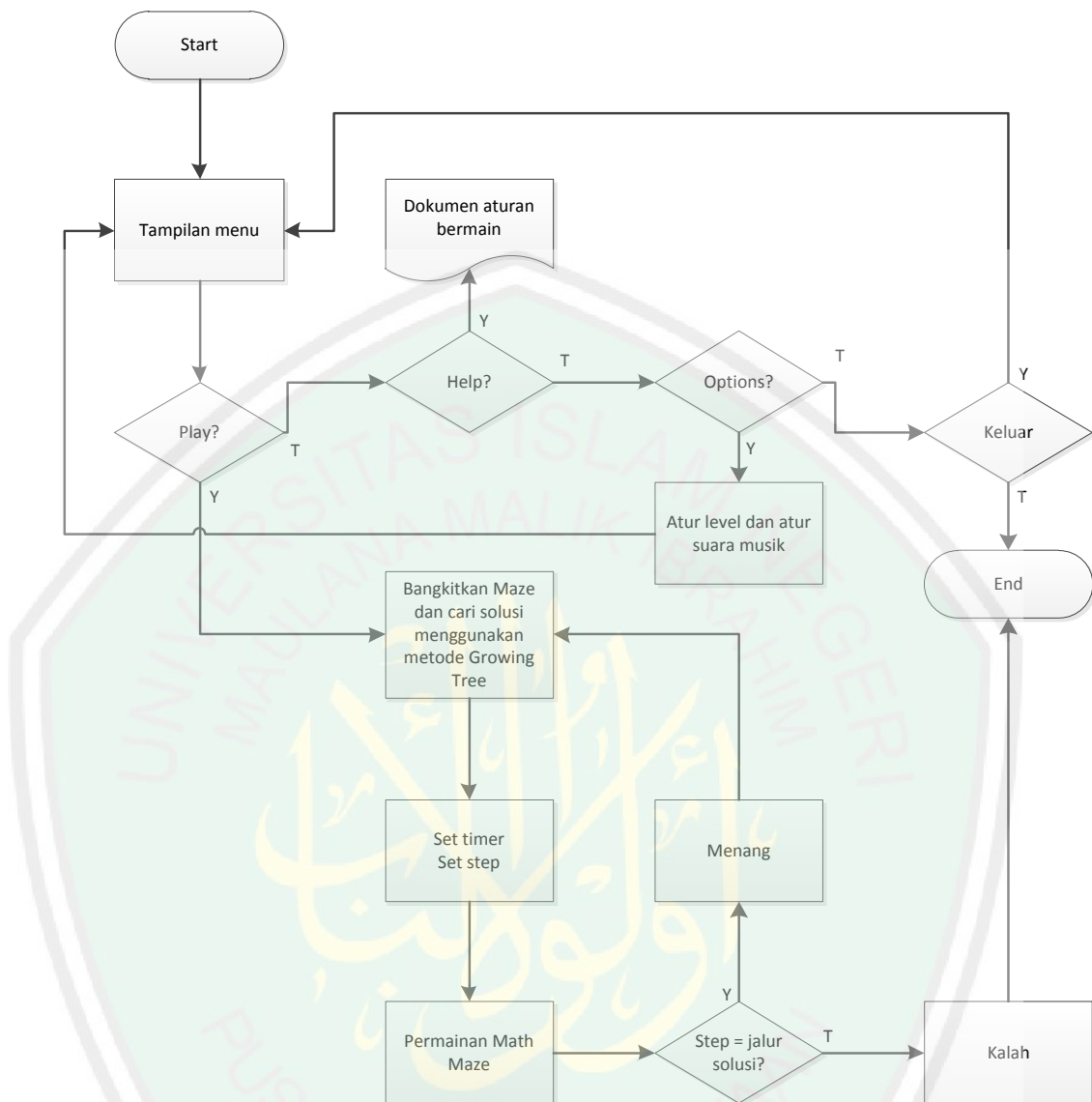


Gambar 3.4. Storyboard Game Math Maze

Keterangan dari tombol menu antara lain:

- a. Menu mulai merupakan menu yang akan melanjutkan pengguna untuk memainkan permainan *Math Maze*.
- b. Menu pengaturan merupakan menu yang berisi untuk memilih tingkat kesulitan yang diinginkan serta menu untuk memilih menghidupkan atau mematikan musik dan efek suara.
- c. Menu bantuan merupakan menu yang berisi petunjuk permainan yang harus dipahami oleh pengguna, sehingga mempermudah dalam menyelesaikan permainan.

Pada rancangan tampilan *options* terdapat tiga pilihan tingkat kesulitan, yaitu level mudah, sedang, dan sulit. Selain itu juga terdapat menu untuk memilih menghidupkan atau mematikan suara musik dan efek suara. Sedangkan untuk antarmuka utama, pemain diberikan *Maze* ukuran 3x3 dengan angka-angka pemandu yang terdapat pada sisi kiri dan atas papan permainan. Apabila pemain berhasil menyelesaikan *Maze* sesuai dengan aturan yang berlaku, maka pemain akan dianggap dapat menyelesaikan permainan. Apabila tidak, maka pemain akan dianggap tidak dapat menyelesaikan permainan. Berikut ini adalah diagram alur dari permainan *Math Maze*:



Gambar 3.6. Flowchart permainan

3.3.3. Desain Karakter

Terdapat dua karakter dalam *game Math Maze* ini. Pertama, karakter utama yang nantinya akan dibantu oleh pemain untuk menentukan tujuan akhir dari *Maze* dengan panduan angka-angka yang terdapat pada sisi kiri dan atas papan permainan untuk menemui Kakek.

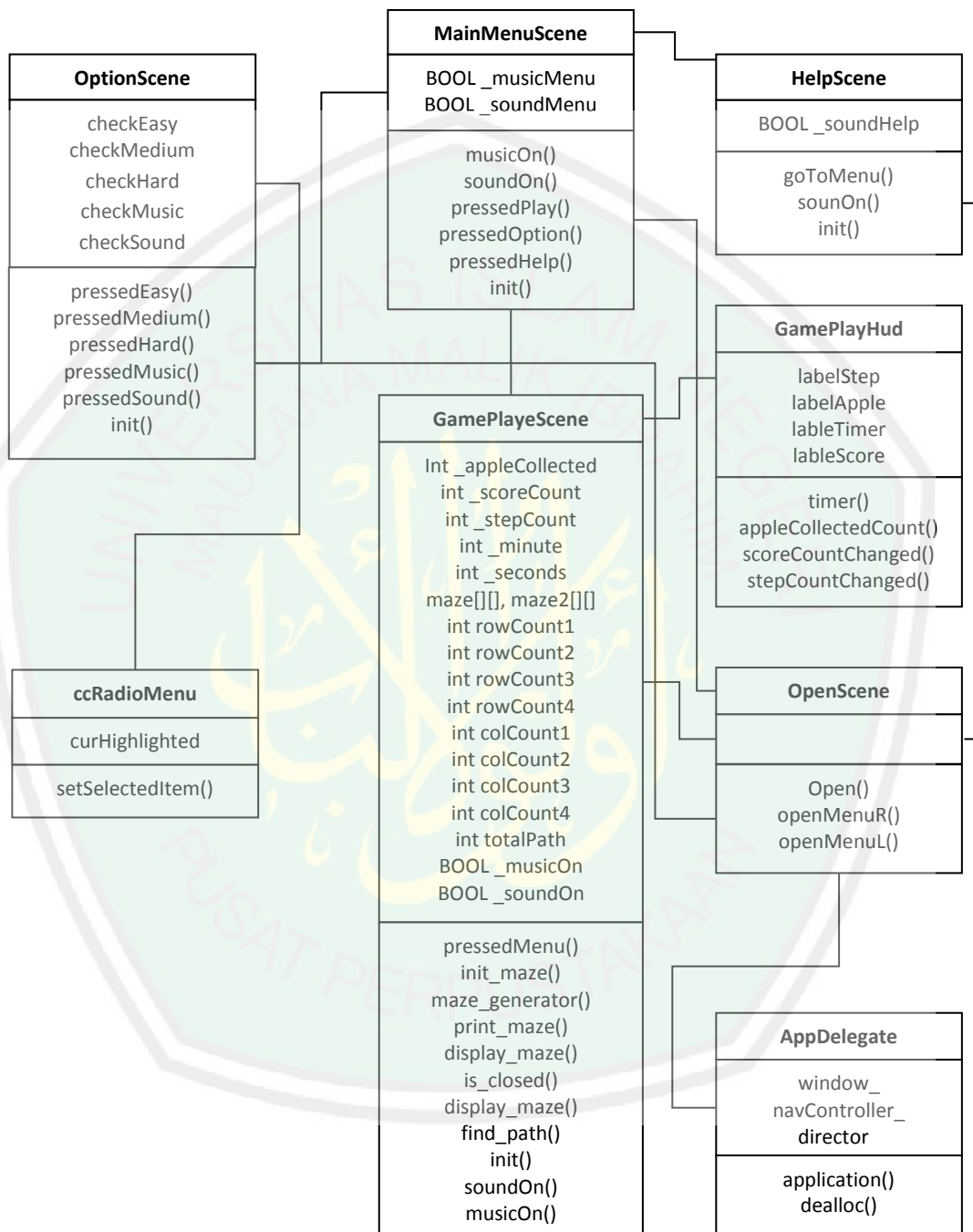


Gambar 3.5. Desain Karakter Utama

Kedua, karakter sampingan yang merupakan seorang Kakek. Karakter Kakek akan dijadikan sebagai tujuan utama dalam *game*. Pemain harus mencapai lokasi Kakek sebelum waktu yang ditentukan habis.



Gambar 3.6. Desain Karakter Sampingan



Gambar 3.7. Class diagram

3.3.4. Kebutuhan Sistem

Pada bagian kebutuhan sistem ini, dibahas tentang kebutuhan sistem perangkat keras maupun perangkat lunak yang mendukung dalam pembuatan dan uji coba aplikasi *Math Maze*.

a. Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk mendukung proses pengembangan sistem aplikasi permainan *Math Maze* adalah:

1. PC / Laptop dengan spesifikasi minimal, processor DualCore 1.75 GHz dan RAM 2 GB, HD 120 GB, digunakan untuk pembuatan aplikasi permainan *Math Maze*.
2. Smartphone / Perangkat *Mobile* berbasis iOS, dibutuhkan untuk melakukan uji coba.

b. Kebutuhan Perangkat Lunak (*Software*)

Beberapa perangkat lunak yang dibutuhkan untuk mendukung proses pengembangan sistem aplikasi permainan *Math Maze* adalah:

1. iOS SDK (*Software Development Kit*), yang diperlukan sebagai alat bantu dan API dalam mengembangkan aplikasi iOS menggunakan bahasa Objective-C.
2. Objective-C, digunakan untuk dapat melakukan kompilasi aplikasi iOS.
3. Xcode, merupakan software yang dibutuhkan untuk melakukan pengkodean aplikasi iOS. Xcode yang digunakan adalah Xcode versi 4.2.
4. MAC OSX Snow Leopard, merupakan sistem operasi APPLE yang digunakan untuk menjalankan aplikasi Xcode versi 4.2. Sistem operasi yang digunakan adalah MAC OSX Snow Leopard versi 10.6.7.

5. Adobe Illustrator, digunakan untuk membuat desain dari permainan *Math Maze*.
6. Tiled, aplikasi yang digunakan untuk membuat papan permainan *Math Maze*.





BAB IV

HASIL DAN PEMBAHASAN

4.1. Implementasi

Pada bab ini membahas tentang implementasi dari perancangan yang telah dibuat sebelumnya, meliputi implementasi algoritma dan implementasi aplikasi *Math Maze*. Serta melakukan pengujian terhadap aplikasi yang dibuat untuk mengetahui apakah aplikasi tersebut telah berjalan sesuai dengan yang diharapkan atau tidak. Pada bab ini juga membahas tentang integrasi Islam dengan algoritma yang digunakan, yaitu *growing tree*.

4.2. Implementasi Growing Tree

Implementasi merupakan proses pembangunan komponen-komponen pokok suatu sistem, yang didasarkan pada desain dan rancangan yang telah dibuat sebelumnya. *Growing tree* akan diimplementasikan pada proses pembangkitan dan penyelesaian *Maze*. Terdapat beberapa parameter yang harus ditentukan untuk membangkitkan sebuah *Maze*, yaitu jumlah maksimum grid (*MAXS*) dan maksimum sel (*CELL*) yang kemudian akan dilakukan inisialisasi *Maze* terlebih dahulu. Berikut ini kode implementasinya:

```
#define MAXS      5
#define CELL      25

#define WALL      1
#define PATH      0

void init_maze(int maze[MAXS][MAXS])
{
    for (int x=0; x < MAXS; x++) {
        for (int y=0; y < MAXS; y++) {
            if (x%2 == 0 || y%2 == 0) {
                maze[x][y] = WALL;
            } else {
                maze[x][y] = PATH;
            }
        }
    }
}
```

```

    }
}

```

Dari implementasi kode di atas, nilai $x\%2$ atau $y\%2$ yang bernilai sama dengan 0 akan dijadikan sebagai dinding (*WALL*). Sedangkan nilai dari $x\%2$ atau $y\%2$ yang bernilai sama dengan 1 akan dijadikan sebagai ruang kosong yang nantinya akan digunakan sebagai tempat untuk membangkitkan *Maze*. Inisialisasi *Maze* yang dihasilkan sebagai berikut:

```

#####
# # #
#####
# # #
#####

```

4.2.1. Pembangkitan Maze

Pada bagian ini akan dibahas mengenai implementasi metode *growing tree* pada proses pembangkitan *Maze* yang digunakan untuk membentuk sebuah *Maze* secara dinamis, sehingga bentuk *Maze* akan berbeda-beda. Setelah dilakukan proses inisialisasi *Maze*, maka selanjutnya yang dilakukan adalah proses pembentukan sebuah *Maze*. Untuk membangkitkan sebuah *Maze* terdapat beberapa parameter utama yang diperlukan, yaitu posisi koordinat dimana *Maze* akan mulai dibangkitkan (x, y), dan ukuran dari *Maze* yang akan dibangkitkan (n), dan nilai untuk sel yang sudah dikunjungi (*visited*). Nilai awal dari *visited* haruslah sama dengan 1, karena posisi awal ketika *Maze* akan dibentuk pasti dikunjungi. Berikut adalah kode implementasinya:

```

void maze_generator(int indeks, int
maze[MAXS][MAXS], int remove_x[CELL], int
remove_y[CELL], int x, int y, int n, int visited)
{
    if(visited < n * n)
    {

```

```

int neighbour_valid = -1;
int neighbour_x[4];
int neighbour_y[4];
int step[4];

int x_next;
int y_next;

if(x - 2 > 0 && is_closed(maze, x - 2, y))
// leftside
{
    neighbour_valid++;
    neighbour_x[neighbour_valid]=x - 2;
    neighbour_y[neighbour_valid]=y;
    step[neighbour_valid]=1;
}

if(y - 2 > 0 && is_closed(maze, x, y - 2))
// upside
{
    neighbour_valid++;
    neighbour_x[neighbour_valid]=x;
    neighbour_y[neighbour_valid]=y - 2;
    step[neighbour_valid]=2;
}

if(y + 2 < n * 2 + 1 && is_closed(maze, x, y
+ 2)) // downside
{
    neighbour_valid++;
    neighbour_x[neighbour_valid]=x;
    neighbour_y[neighbour_valid]=y + 2;
    step[neighbour_valid]=3;
}

if(x + 2 < n * 2 + 1 && is_closed(maze, x +
2, y)) // rightside
{
    neighbour_valid++;
    neighbour_x[neighbour_valid]=x + 2;
    neighbour_y[neighbour_valid]=y;
    step[neighbour_valid]=4;
}

if(neighbour_valid == -1)
{
    // remove cell
    x_next = remove_x[indeks];
    y_next = remove_y[indeks];
    indeks--;
}

```

```

        if(neighbour_valid != -1)
        {
            int randomization = neighbour_valid + 1;
            int random = rand()%randomization;
            x_next = neighbour_x[random];
            y_next = neighbour_y[random];
            indeks++;
            remove_x[indeks] = x_next;
            remove_y[indeks] = y_next;

            int rstep = step[random];

            if(rstep == 1)
                maze[x_next+1][y_next] = PATH;
            else if(rstep == 2)
                maze[x_next][y_next + 1] = PATH;
            else if(rstep == 3)
                maze[x_next][y_next - 1] = PATH;
            else if(rstep == 4)
                maze[x_next - 1][y_next] = PATH;
            visited++;
        }

        maze_generator(indeks, maze, remove_x,
remove_y, x_next, y_next, n, visited);
    }

    void print_maze(int maze[MAXS][MAXS])
    {
        int rnd = randomNum() % NROWS;
        if (rnd == 0) {
            rnd = 3;
        }
        for(int a = 0; a < NROWS+1; a++)
        {
            for(int b = 0; b < MCOLS+1; b++)
            {
                if (a==1 && b==0) {
                    maze[a][b] = 'S';
                    maze2[a][b] = maze[a][b];
                } else if (b==4 && rnd==a) {
                    maze[rnd][b] = 'G';
                    maze2[a][b] = maze[a][b];
                    finish = rnd;
                } else if(maze[a][b] == WALL) {
                    maze[a][b] = '#';
                    maze2[a][b] = maze[a][b];
                } else {
                    maze[a][b] = '.';
                    maze2[a][b] = maze[a][b];
                }
            }
        }
    }

```



```

        printf("\n");
    }
}

int is_closed(int maze[MAXS][MAXS], int x, int y)
{
    if(maze[x - 1][y] == WALL && maze[x][y - 1] ==
WALL &&
        maze[x][y + 1] == WALL && maze[x + 1][y] ==
WALL)
        return 1;

    return 0;
}

```

Dari implementasi kode di atas, nilai n yang diinputkan adalah 5, sehingga ukuran dari *Maze* yang akan dibangkitkan adalah 5x5. Untuk membentuk *Maze* sempurna haruslah menggunakan bilangan ganjil supaya dinding-dinding dari *Maze* yang dibangkitkan terbentuk sempurna. Proses penentuan sel tetangga yang akan dikunjungi dari sel awal, ditentukan secara random (horizontal dan vertikal). Jika jumlah sel yang dikunjungi (*visited*) kurang dari jumlah sel ($n * n$) yang terdapat pada *Maze*, maka akan dilakukan proses pembentukan *Maze* dengan memilih tetangga terdekat secara horizontal dan vertikal. Akan tetapi, apabila nilai *visited* sama dengan nilai $n*n$, maka proses pembentukan *Maze* telah selesai. Artinya semua sel yang terdapat pada *Maze* telah dikunjungi. Berikut adalah contoh dari *Maze* yang dihasilkan:

```

#####
S.#.#
#.#.G
#...#
#####

```

4.2.2. Penyelesaian Maze

Metode *growing tree* juga diimplementasikan untuk melakukan penyelesaian dari *Maze* yang telah dibangkitkan sebelumnya. Langkah pertama, inisialisasi parameter berupa koordinat posisi *START* dan *FINISH* yang ditentukan secara random saat proses pembangkitan *Maze* selesai. Berikut adalah kode implementasinya:

```
int find_path(int x, int y)
{
    // x,y is outside maze?
    if ( x < 0 || x > MCOLS || y < 0 || y > NROWS )
return FALSE;

    // x,y is the goal?
    if ( maze2[y][x] == 'G' ) return TRUE;

    // x,y is not open?
    if ( maze2[y][x] != '.' && maze2[y][x] != 'S' )
return FALSE;

    // mark x,y part of solution path.
    maze2[y][x] = '+';

    // north
    if ( find_path(x, y - 1) == TRUE ) return TRUE;

    // east
    if ( find_path(x + 1, y) == TRUE ) return TRUE;

    // south
    if ( find_path(x, y + 1) == TRUE ) return TRUE;

    // west
    if ( find_path(x - 1, y) == TRUE ) return TRUE;

    // unmark x,y as part of solution path.
    maze2[y][x] = 'x';

    return FALSE;
}
```

Pencarian jalur solusi akan dimulai dari titik *start*. Pencarian solusi dimulai dari tetangga terdekat secara vertikal (utara dan selatan) dan horizontal (barat dan

timur). Arah utara dan selatan memiliki inisialisasi 'x, y-1' dan 'x, y+1'. Sedangkan untuk arah barat dan timur memiliki inisialisasi 'x-1, y' dan 'x+1, y'. Untuk setiap jalur yang terlewati akan diberi tanda '+'. Akan tetapi ketika jalur tersebut menemui jalan buntu (*dead end*), maka jalur tersebut akan dihapus dari daftar solusi dan diberi tanda 'x', sehingga posisi jalur akan kembali pada saat sebelum menemui *dead end* sampai jalur tersebut mencapai *finish*. Berikut adalah hasil dari jalur solusi yang didapat dari proses pembangkitan *Maze* yang dilakukan sebelumnya.

```
#####
S+#x#
#+#+G
#+++#
#####
```

4.3. Merubah Jalur Solusi Menjadi Angka-Angka Pemandu

Dari hasil solusi yang didapat, maka langkah yang dilakukan selanjutnya adalah merubah jalur solusi menjadi angka-angka yang terdapat pada bagian kiri dan atas papan permainan. Angka-angka ini digunakan sebagai angka-angka pemandu dalam menyelesaikan permainan *Math Maze*. Berikut ini adalah kode implementasinya:

```
void display_maze(void)
{
    for (int x=0; x<NROWS+1; x++) {
        for (int y=0; y<MCOLS+1; y++) {
            printf("%c", maze2[x][y]);
            if (maze2[x][y] == '+') {
                totalPath++;
                if (x == 1 && (y > 0 && y < MCOLS))
                {
                    rowCount1++;
                } else if (x == 2 && (y > 0 && y <
                MCOLS)) {
                    rowCount2++;
                }
            }
        }
    }
}
```

```

    } else if (x == 3 && (y > 0 && y <
MCOLS)) {
        rowCount3++;
    }
    if ((x > 0 && x < NROWS) && y == 1)
    {
        colCount1++;
    } else if ((x > 0 && x < NROWS) && y
== 2) {
        colCount2++;
    } else if ((x > 0 && x < NROWS) && y
== 3) {
        colCount3++;
    } else if ((x > 0 && x < NROWS) && y
== 4) {
        colCount4++;
    }
    }
    printf("\n");
}
return;
}

```

Untuk mendapatkan angka-angka pemandu, maka dilakukan proses perhitungan jumlah sel yang dilalui oleh jalur solusi pada setiap baris dan kolom. Setiap sel pada baris dan kolom yang dilalui jalur solusi akan ditandai '+'. Seperti pada hasil yang didapat, baris ke-1 memiliki 1 tanda '+', sehingga angka yang didapat adalah 1. Pada baris ke-2 jumlah tanda '+' adalah 2, sehingga angka yang di dapat adalah 2. Kolom ke-1 memiliki 3 tanda '+', oleh sebab itu angka yang didapat adalah 3. Begitu juga untuk setiap baris dan kolom yang lain.

```

312
#####
1S+#x#
2#+##+G
3#+###
#####

```

Dari *Maze* ukuran 5x5 yang telah dibangkitkan, maka akan didapatkan *Math Maze* dengan ukuran 3x3. Untuk mengetahui langkah-langkah pembuatan *game Math Maze* lebih detail dapat dilihat pada **lampiran 1**.

4.4. Implementasi Aplikasi

Pada bagian ini akan dibahas tentang implementasi dari rancangan *Math Maze* yang telah dilakukan sebelumnya ke dalam aplikasi perangkat *mobile* berbasis iOS. Pada bagian ini juga akan dijelaskan secara detail dari tiap *scene* yang terdapat pada aplikasi *Math Maze*. Berikut ini hasil implementasinya:



Gambar 4.1. Tampilan utama *Math Maze*

Pada saat pemain menjalankan aplikasi *Math Maze*, maka pemain akan dihadapkan pada tampilan utama, seperti pada gambar 4.1. Terdapat tiga menu pada tampilan utama. Menu *play* berfungsi untuk memainkan *Math Maze* yang akan diarahkan pada tampilan utama permainan. Menu *options* berfungsi untuk mengatur tingkat kesulitan dan efek suara. Menu *help* berfungsi untuk mengetahui cara memainkan *Math Maze*.



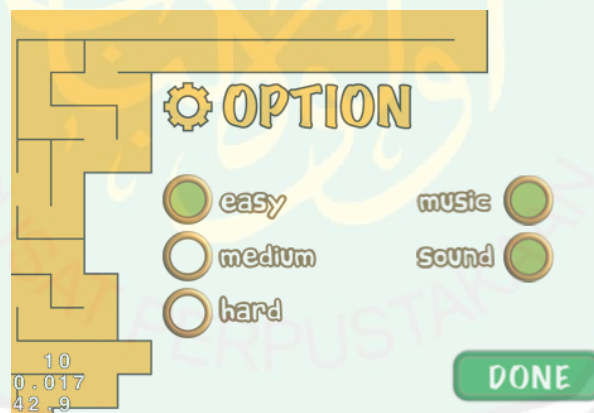
Gambar 4.2. Tampilan menu bantuan bagian pertama

Gambar 4.2 adalah menu *help* bagian pertama. Pada menu ini pemain dikenalkan terlebih dahulu kepada karakter yang terdapat pada *game* serta komponen apa saja yang terdapat pada *game Math Maze*. Terdapat dua karakter dalam *game* ini. Pertama, karakter utama yang bernama ‘Haidar’. Haidar merupakan karakter yang akan dimainkan oleh pemain yang memiliki tujuan untuk keluar dari *Maze* dengan mengambil buah apel sesuai dengan jumlah yang diinginkan oleh Kakek. Kedua, karakter Kakek. Kakek merupakan karakter sampingan yang memberi perintah pada Haidar untuk mengambil buah apel sesuai dengan jumlah yang diinginkan. Kemudian terdapat dua komponen dalam *game* ini. Komponen pertama adalah buah apel. Buah apel ini merupakan item yang diinginkan oleh Kakek yang nantinya akan diambil oleh Haidar. Komponen kedua adalah angka-angka pemandu yang terdapat pada sisi-sisi papan permainan. Angka-angka ini digunakan sebagai panduan supaya Haidar bisa keluar dari *Maze* dan mendapatkan buah apel sesuai dengan jumlah yang diinginkan oleh Kakek.



Gambar 4.3. Tampilan menu bantuan bagian kedua

Pada menu *help* bagian kedua, seperti yang terdapat pada gambar 4.3. Pemain diberikan sebuah isyarat. Apabila pemain mengalami kesalahan dalam memilih langkah, maka skor dari pemain akan berkurang. Semakin banyak kesalahan langkah yang dilakukan oleh pemain, maka skor yang berkurang akan semakin banyak pula.



Gambar 4.4. Tampilan menu pengaturan

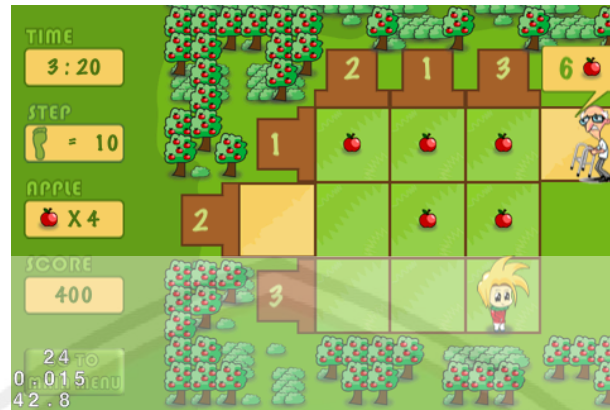
Pada tampilan *options* seperti pada gambar 4.4. Terdapat beberapa tombol yang digunakan untuk pengaturan seting dalam *game*. Untuk pengaturan tingkat kesulitan terdapat tiga tombol, yaitu mudah (*easy*), sedang (*medium*), dan sulit (*hard*). Perbedaan dari tingkat kesulitan ini adalah dari waktu (*timer*) dan jumlah langkah (*step*) yang disediakan dalam menyelesaikan permainan. Apabila pemain tidak mengatur tingkat kesulitan, maka secara otomatis tingkat kesulitan yang

dipilih adalah tingkat kesulitan dengan level mudah. Dalam menu *options* ini juga disediakan pengaturan musik (*music*) dan efek suara (*sound*), dimana pemain bisa menghidupkan atau mematikan musik dan efek suara dari game ini. Setelah pengaturan selesai dilakukan, maka pemain dapat menekan tombol *done* untuk kembali ke menu utama dan *Math Maze* siap dimainkan.



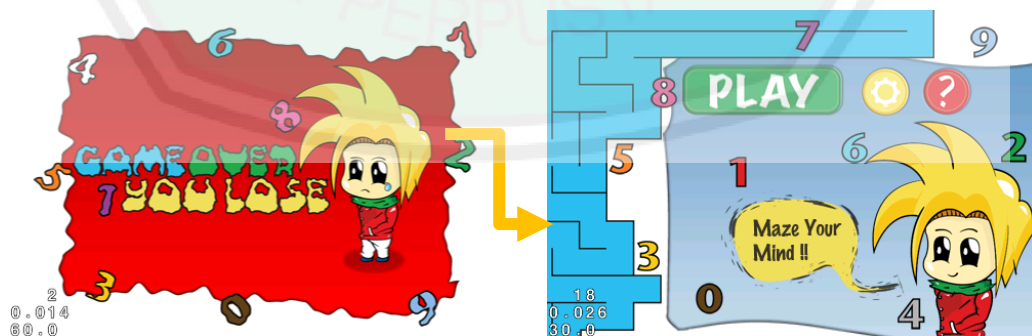
Gambar 4.5. Tampilan papan permainan pertama

Gambar 4.5 merupakan tampilan utama dari permainan *Math Maze* dengan waktu yang telah disediakan dan langkah yang terbatas, sesuai dengan tingkat kesulitan yang dipilih pada menu *options*. Pada gambar 4.5, Kakek menginginkan enam buah apel sehingga pemain harus membantu Haider untuk mengumpulkan enam buah apel sesuai dengan keinginan Kakek serta melewati *Maze* dengan menggunakan angka-angka pemandu yang terdapat pada sisi-sisi papan permainan.



Gambar 4.6. Tampilan papan permainan kedua

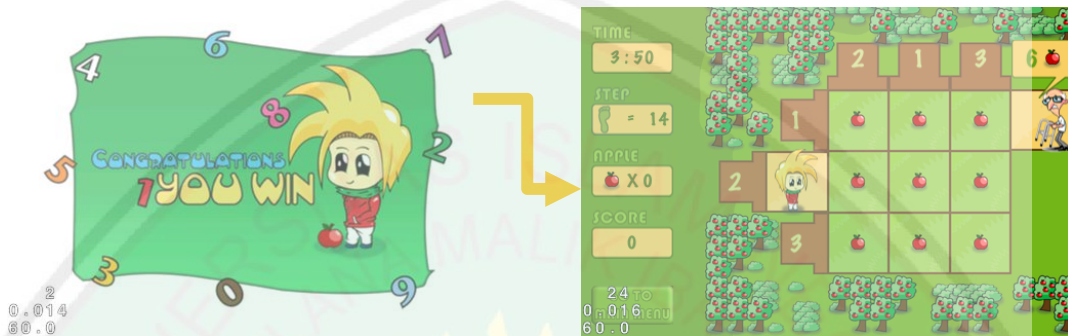
Dapat dilihat pada gambar 4.6 ketika pemain melakukan perpindahan posisi, maka jumlah langkah (*step*) yang dimiliki pemain akan berkurang. Pada gambar 4.5, jumlah *step* awal yang dimiliki pemain adalah 14. Namun pada gambar 4.6, jumlah *step* yang dimiliki pemain tersisa 10. Hal ini disebabkan karena pemain melakukan perpindahan posisi sebanyak empat kali. Dapat dilihat juga waktu (*time*) yang semakin berkurang. Ketika *time* = 0 atau *step* = 0, maka pemain dianggap gagal dalam menyelesaikan *Maze* dan tidak bisa memenuhi keinginan Kakek untuk mendapatkan jumlah apel yang diinginkan. Sehingga akan tampil pesan bahwa pemain gagal dan pemain akan diarahkan pada menu utama kembali, seperti pada gambar 4.7



Gambar 4.7. Tampilan ketika kalah

Akan tetapi ketika pemain mampu menyelesaikan *Maze* dan mendapatkan buah apel sesuai dengan keinginan Kakek sebelum *time* = 0 dan sebelum *step* = 0,

maka pemain akan dianggap mampu menyelesaikan *Maze* dan mampu memenuhi keinginan Kakek. Sehingga akan tampil pesan bahwa pemain berhasil menyelesaikan permainan dan pemain akan diarahkan kembali pada menu utama permainan, seperti pada gambar 4.8.



Gambar 4.8. Tampilan ketika menang

4.5. Uji Coba

Setelah melakukan implementasi terhadap semua rancangan ke dalam sebuah aplikasi, selanjutnya yang perlu dilakukan adalah proses uji coba. Langkah ini dilakukan untuk menguji fungsi-fungsi yang terdapat pada aplikasi. Uji coba dilakukan untuk menguji dari metode *growing tree* yang telah diterapkan pada permainan *Math Maze*. Selain itu, uji coba juga dilakukan kepada beberapa pengguna untuk mengetahui seberapa besar penilaian dan komentar dari para pengguna terhadap aplikasi ini.

4.5.1. Uji Coba Growing Tree

Uji coba algoritma *growing tree* dilakukan pada pembentukan *Maze* dan pencarian solusi yang didapat. Berikut ini adalah daftar tabel uji coba yang dilakukan pada algoritma *growing tree*.

Tabel 4.1. Variasi bentuk *Maze* ukuran 5x5.

No	Titik awal (x,y) / (kolom, baris)	Bentuk Maze
1	1,1	##### # # # # # # # # #####
2	1,1	##### # # ### # # # #####
3	3,1	##### # # # # # # # # #####
4	3,1	##### # # # ### # # #####
5	1,3	##### # # # # # # # # #####
6	1,3	##### # # ### # # # #####
7	3,3	##### # # # # # # # # #####
8	3,3	##### # # # ### # # #####

Dari hasil uji coba pembentukan *Maze* pada tabel 4.1 dengan posisi titik awal yang berbeda-beda. *Maze* yang dibentuk memiliki empat varian. Hal ini disebabkan *Maze* memiliki bentuk yang sama pada nomor 1-3, 2-6, 4-8, dan 5-7.

Tabel 4.2. Variasi lintasan dari *Maze* yang dibangkitkan.

No	Start (x,y) / (baris, kolom)	Finish (x,y) / (baris, kolom)	Variasi lintasan
1	1,0	1,4	2
2	2,0	2,4	2
3	3,0	3,4	2
4	1,0	2,4	2
5	2,0	3,4	2
6	3,0	1,4	2
7	1,0	3,4	2
8	2,0	1,4	2
9	3,0	2,4	2
10	1,4	1,0	2
11	2,4	2,0	2
12	3,4	3,0	2
13	1,4	2,0	2
14	2,4	3,0	2
15	3,4	1,0	2
16	1,4	3,0	2
17	2,4	1,0	2
18	3,4	2,0	2
19	0,1	4,1	2
20	0,2	4,2	2
21	0,3	4,3	2
22	0,1	4,2	2
23	0,2	4,3	2
24	0,3	4,1	2
25	0,1	4,3	2
26	0,2	4,1	2
27	0,3	4,2	2
28	4,1	0,1	2
29	4,2	0,2	2
30	4,3	0,3	2
31	4,1	0,2	2
32	4,2	0,3	2
33	4,3	0,1	2
34	4,1	0,3	2
35	4,2	0,1	2
36	4,3	0,2	2

Dari tabel 4.2 dapat dilihat, bahwa dengan posisi *start* dan *finish* yang berbeda-beda akan selalu didapatkan dua macam variasi jalur solusi. Hal ini disebabkan karena *Maze* yang hanya memiliki empat varian.

Tabel 4.3. Pencarian solusi dengan posisi *start* dan *finish* yang berbeda

No	Start (x,y) / (baris, kolom)	Finish (x,y) / (baris, kolom)	Ditemukan solusi?	Jumlah solusi
1	1,0	1,4	YA	1
2	2,0	2,4	YA	1
3	3,0	3,4	YA	1
4	1,0	2,4	YA	1
5	2,0	3,4	YA	1
6	3,0	1,4	YA	1
7	1,0	3,4	YA	1
8	2,0	1,4	YA	1
9	3,0	2,4	YA	1
10	1,4	1,0	YA	1
11	2,4	2,0	YA	1
12	3,4	3,0	YA	1
13	1,4	2,0	YA	1
14	2,4	3,0	YA	1
15	3,4	1,0	YA	1
16	1,4	3,0	YA	1
17	2,4	1,0	YA	1
18	3,4	2,0	YA	1
19	0,1	4,1	YA	1
20	0,2	4,2	YA	1
21	0,3	4,3	YA	1
22	0,1	4,2	YA	1
23	0,2	4,3	YA	1
24	0,3	4,1	YA	1
25	0,1	4,3	YA	1
26	0,2	4,1	YA	1
27	0,3	4,2	YA	1
28	4,1	0,1	YA	1
29	4,2	0,2	YA	1
30	4,3	0,3	YA	1
31	4,1	0,2	YA	1
32	4,2	0,3	YA	1
33	4,3	0,1	YA	1
34	4,1	0,3	YA	1
35	4,2	0,1	YA	1

No	Start (x,y) / (baris, kolom)	Finish (x,y) / (baris, kolom)	Ditemukan solusi?	Jumlah solusi
36	4,3	0,2	YA	1

Dari tabel 4.3 dapat dilihat bahwa setiap posisi *start* dan *finish* yang berbeda-beda, solusi tetap ditemukan dan jumlah solusi hanya satu pada masing-masing koordinat yang telah ditentukan. Sebagai contoh pada no.1, posisi *start* dimulai dari kolom (0), baris (1) dan posisi *finish* terdapat pada kolom (4) baris (1), telah ditemukan solusi dengan jumlah satu jalur solusi. Dari hasil percobaan tersebut, maka *Maze* yang dibangkitkan adalah *Maze* sempurna yang hanya memiliki satu jalur solusi.

4.5.2. Uji Coba Pengguna

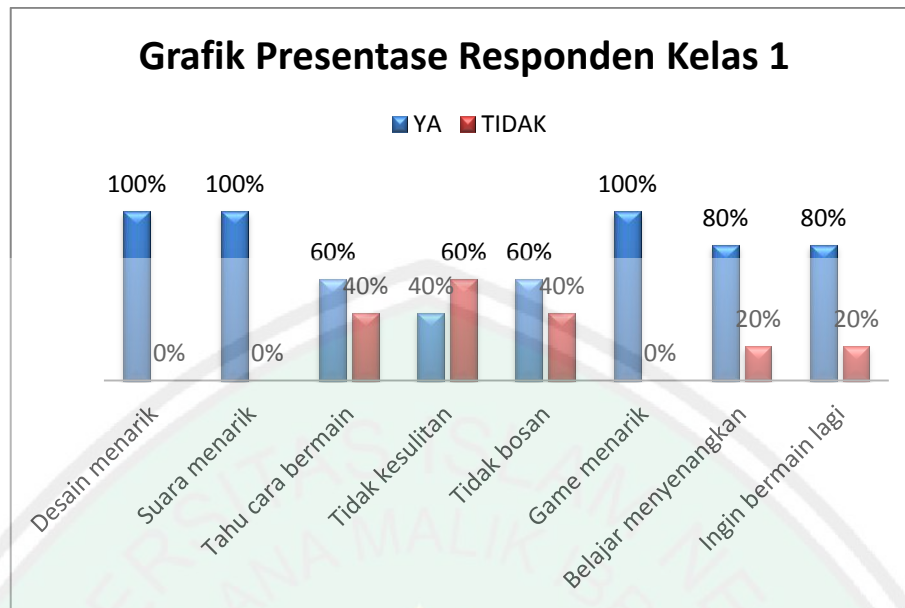
Pengujian aplikasi permainan *Math Maze* juga dilakukan pada responden yang berada pada lingkungan SDN Pagentan 3 Jalan Ronggolawe Gang 2 Kecamatan Singosari Malang. Pemilihan responden dilakukan secara acak sebanyak 20 responden dari kelas 1 sampai kelas 4, dimana masing-masing kelasnya dipilih sebanyak 5 responden. Pengujian ini dilakukan untuk mengetahui penilaian dari pengguna terhadap ketertarikan dari aplikasi tersebut. Untuk penilaian responden tentang aplikasi permainan *Math Maze* secara spesifik, dapat dilihat pada hasil tabulasi kuisioner responden dari masing-masing kelas.

Tabel 4.4. Tabel tabulasi kuisioner kelas 1

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
1	Desain menarik	5	100%	0	0%

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
2	Musik dan efek suara menarik	5	100%	0	0%
3	Mengetahui cara memainkan <i>Math Maze</i>	3	60%	2	40%
4	Tidak kesulitan dalam memainkan <i>Math Maze</i>	2	40%	3	60%
5	<i>Math Maze</i> tidak membosankan	3	60%	2	40%
6	Game menarik dan menyenangkan	5	100%	0	0%
7	Lebih senang belajar berhitung bersama <i>Math Maze</i>	4	80%	1	20%
8	Ingin memainkan <i>Math Maze</i> lagi	4	80%	1	20%
Total		77.5%		22.5%	

Berdasarkan tabel 4.4 di atas tentang penilaian dari total 5 responden kelas 1 terhadap kuisioner yang diberikan. Sebanyak 100% responden menilai aplikasi tersebut menarik, baik dari segi desain tampilan, cara bermain dan efek suara yang dikeluarkan. Sedangkan sebanyak 60% responden mengetahui cara memainkan *Math Maze* dan tidak bosan dalam memainkannya. Sebanyak 40% responden tidak mengalami kesulitan dalam memainkan aplikasi ini. Dan sebanyak 80% responden menilai, dengan aplikasi *Math Maze* belajar terasa menyenangkan dan ingin memainkan *Math Maze* lagi. Dari total responden secara keseluruhan, sebanyak 77.5% responden menjawab ‘Ya’ dan sebanyak 22.5% responden menjawab ‘Tidak’. Berikut ini hasil grafik dari tabel tabulasi responden kelas 1.



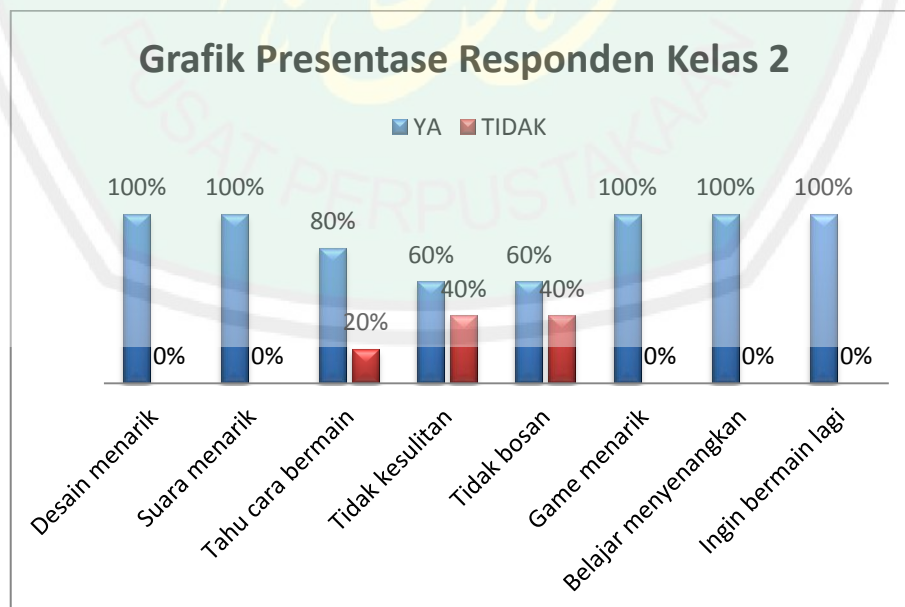
Gambar 4.3. Grafik presentase responden kelas 1

Tabel 4.5. Tabel tabulasi kuisioner kelas 2

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
1	Desain menarik	5	100%	0	0%
2	Musik dan efek suara menarik	5	100%	0	0%
3	Mengetahui cara memainkan <i>Math Maze</i>	4	80%	1	20%
4	Tidak kesulitan dalam memainkan <i>Math Maze</i>	3	60%	2	40%
5	<i>Math Maze</i> tidak membosankan	3	60%	2	40%
6	Game menarik dan menyenangkan	5	100%	0	0%
7	Lebih senang belajar berhitung bersama <i>Math Maze</i>	5	100%	0	0%
8	Ingin memainkan <i>Math Maze</i> lagi	5	100%	0	0%

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
Total		87.5%		12.5%	

Berdasarkan tabel 4.5 di atas tentang penilaian dari total 5 responden kelas 2 terhadap kuisioner yang diberikan. Sebanyak 100% responden menilai aplikasi tersebut menarik, baik dari segi desain tampilan, cara bermain dan efek suara yang dikeluarkan, serta responden merasa dengan *Math Maze* belajar terasa menyenangkan dan ingin memainkan *Math Maze* lagi. Sebanyak 80% responden mengetahui cara memainkan *Math Maze*. Sedangkan sebanyak 60% responden tidak mengalami kesulitan dalam memainkan *Math Maze* dan tidak bosan dalam memainkannya. Dari total responden secara keseluruhan, sebanyak 87.5% responden menjawab ‘Ya’ dan sebanyak 12.5% responden menjawab ‘Tidak’. Berikut ini hasil grafik dari tabel tabulasi responden kelas 2.



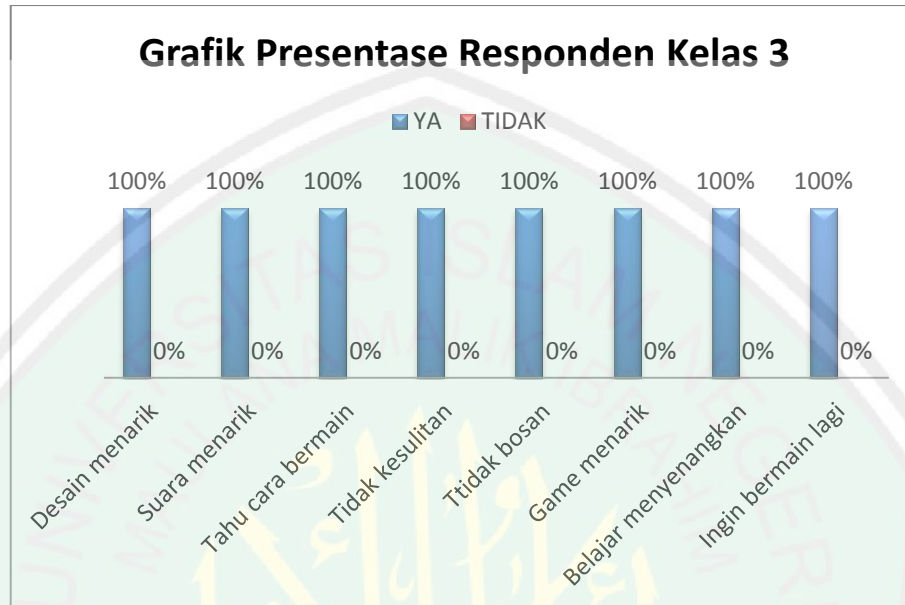
Gambar 4.4. Grafik presentase responden kelas 2

Tabel 4.6. Tabel tabulasi kuisioner kelas 3

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
1	Desain menarik	5	100%	0	0%
2	Musik dan efek suara menarik	5	100%	0	0%
3	Mengetahui cara memainkan <i>Math Maze</i>	5	100%	0	0%
4	Tidak kesulitan dalam memainkan <i>Math Maze</i>	5	100%	0	0%
5	<i>Math Maze</i> tidak membosankan	5	100%	0	0%
6	Game menarik dan menyenangkan	5	100%	0	0%
7	Lebih senang belajar berhitung bersama <i>Math Maze</i>	5	100%	0	0%
8	Ingin memainkan <i>Math Maze</i> lagi	5	100%	0	0%
Total		100%		0%	

Berdasarkan tabel 4.6 di atas tentang penilaian dari total 5 responden kelas 3 terhadap kuisioner yang diberikan. Sebanyak 100% responden menilai aplikasi tersebut menarik, baik dari segi desain tampilan, cara bermain dan efek suara yang dikeluarkan, serta responden mengetahui cara memainkan *Math Maze* dan tidak merasa kesulitan dalam memainkannya. Dan sebanyak 100% responden menilai, dengan aplikasi *Math Maze* belajar terasa menyenangkan dan ingin memainkan *Math Maze* lagi serta tidak bosan dalam memainkannya. Dari total responden

secara keseluruhan, sebanyak 100% responden menjawab ‘Ya’. Berikut ini hasil grafik dari tabel tabulasi responden kelas 3.



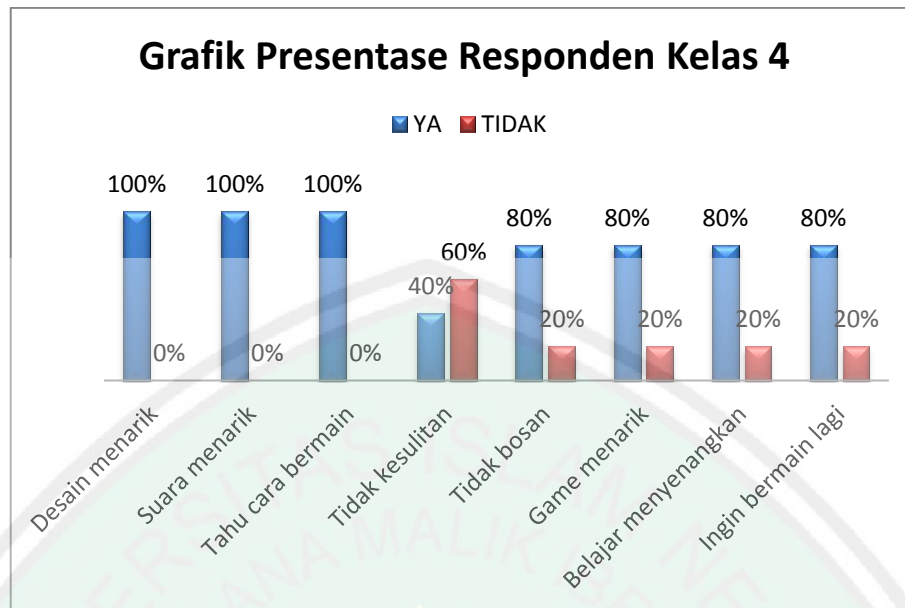
Gambar 4.5. Grafik presentase responden kelas 3

Tabel 4.7. Tabel tabulasi kuisioner kelas 4

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
1	Desain menarik	5	100%	0	0%
2	Musik dan efek suara menarik	5	100%	0	0%
3	Mengetahui cara memainkan <i>Math Maze</i>	5	100%	0	0%
4	Tidak kesulitan dalam memainkan <i>Math Maze</i>	2	40%	3	60%
5	<i>Math Maze</i> tidak membosankan	4	80%	1	20%
6	Game menarik dan menyenangkan	4	80%	1	20%

No	Keterangan	Ya		Tidak	
		Frekuensi	Presentase	Frekuensi	Presentase
7	Lebih senang belajar berhitung bersama <i>Math Maze</i>	4	80%	1	20%
8	Ingin memainkan <i>Math Maze</i> lagi	4	80%	1	20%
Total		82.5%		17.5%	

Berdasarkan tabel 4.7 di atas tentang penilaian dari total 5 responden kelas 4 terhadap kuisioner yang diberikan. Sebanyak 100% responden menilai aplikasi tersebut menarik, baik dari segi desain tampilan, cara bermain dan efek suara yang dikeluarkan, serta mengetahui cara memainkan *Math Maze*. Sebanyak 40% responden tidak mengalami kesulitan dalam memainkan *Math Maze*. Sedangkan sebanyak 80% responden menilai, dengan aplikasi *Math Maze* belajar terasa menyenangkan dan ingin memainkan *Math Maze* lagi, serta tidak merasa kesulitan dalam memainkannya. Dari total responden secara keseluruhan, sebanyak 82.5% responden menjawab ‘Ya’ dan sebanyak 17.5% responden menjawab ‘Tidak’. Berikut ini hasil grafik dari tabel tabulasi responden kelas 4.



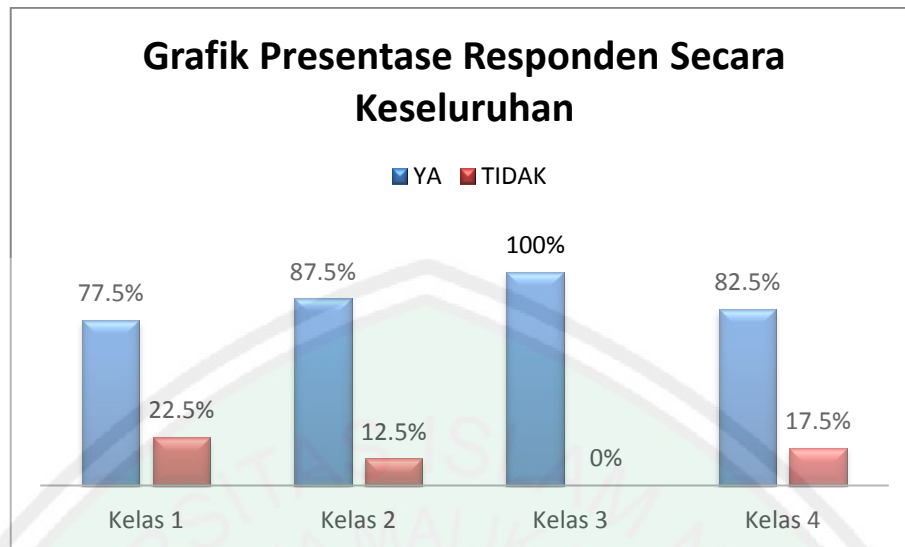
Gambar 4.6. Grafik presentase responden kelas 4

Dari tabel tabulasi responden masing-masing kelas, maka didapatkan tabel tabulasi responden secara keseluruhan. Berikut ini tabel tabulasi responden secara keseluruhan dari kelas 1 sampai kelas 4:

Tabel 4.8. Tabel tabulasi kuisioner responden secara keseluruhan

No	Keterangan	Frekuensi		Presentase	
		Ya	Tidak	Ya	Tidak
1	Kelas 1	31	9	77.5%	22.5%
2	Kelas 2	35	5	87.5%	12.5%
3	Kelas 3	40	0	100%	0%
4	Kelas 4	33	7	82.5%	17.5%
Total				87%	13%

Berdasarkan tabel 4.8. Secara keseluruhan aplikasi *Math Maze* menarik dan menyenangkan bagi responden. Dari total 20 responden, sebanyak 87% responden menyukai *Math Maze* dan sebanyak 13% responden tidak menyukai *Math Maze*. Berikut ini grafik dari tabel tabulasi responden secara keseluruhan.



Gambar 4.7. Grafik presentase responden secara keseluruhan

4.6. Integrasi Islam dengan Growing Tree

Growing tree merupakan algoritma dengan struktur *tree* yang memiliki kemiripan dengan algoritma *backtracking* dan algoritma *prim's*. Algoritma ini dapat digunakan untuk membangkitkan sebuah *Maze* dan mampu mencari solusi dari *Maze* yang telah dibangkitkan. Disini dapat dibuktikan bahwa setiap permasalahan pastilah memiliki solusi. Sebagaimana firman Allah SWT dalam surat Al-Baqarah ayat 287:

لَا يُكَلِّفُ اللَّهُ نَفْسًا إِلَّا وُسْعَهَا لَهَا مَا كَسَبَتْ وَعَلَيْهَا مَا اكْتَسَبَتْ رَبَّنَا لَا تُؤَاخِذْنَا إِن نَّسِينَا أَوْ أَخْطَأْنَا رَبَّنَا وَلَا تَحْمِلْ عَلَيْنَا إَصْرًا كَمَا حَمَلْتَهُ عَلَى الَّذِينَ مِن قَبْلِنَا رَبَّنَا وَلَا تُحَمِّلْنَا مَا لَا طَاقَةَ لَنَا بِهِ ۖ وَاعْفُ عَنَّا وَارْحَمْنَا ۖ أَنْتَ مَوْلَانَا فَانصُرْنَا عَلَى الْقَوْمِ الْكَافِرِينَ ﴿٢٨٧﴾

Artinya:

Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. ia mendapat pahala (dari kebajikan) yang diusahakannya dan ia mendapat siksa (dari kejahatan) yang dikerjakannya. (mereka berdoa): "Ya Tuhan Kami, janganlah Engkau hukum Kami jika Kami lupa atau Kami tersalah. Ya Tuhan Kami, janganlah Engkau bebaskan kepada Kami beban yang berat sebagaimana Engkau bebaskan kepada orang-

orang sebelum kami. Ya Tuhan Kami, janganlah Engkau pikulkan kepada Kami apa yang tak sanggup Kami memikulnya. beri ma'aflah kami; ampunilah kami; dan rahmatilah kami. Engkaulah penolong Kami, Maka tolonglah Kami terhadap kaum yang kafir." (Q.S Al-Baqarah: 286)

Dari ayat di atas dapat diketahui bahwa Allah tidak akan membebani hambanya melainkan sesuai dengan batas kemampuannya. Allah memberikan permasalahan kepada hambanya supaya hambanya berpikir. Sebagaimana firman Allah:

وَفِي الْأَرْضِ قِطْعٌ مُتَجَاوِرَاتٌ وَجَنَّاتٌ مِّنْ أَعْنَابٍ وَزُرْعٌ وَنَخِيلٌ صِنَوَانٌ وَغَيْرُ صِنَوَانٍ
يُسْقَىٰ بِمَاءٍ وَاحِدٍ وَنُفِضِلُ بَعْضَهَا عَلَىٰ بَعْضٍ فِي الْأُكُلِ ۚ إِنَّ فِي ذَٰلِكَ لَآيَاتٍ لِّقَوْمٍ
يَعْقِلُونَ ﴿١١١﴾

Artinya:

Dan di bumi ini terdapat bagian-bagian yang berdampingan, dan kebun-kebun anggur, tanaman-tanaman dan pohon korma yang bercabang dan yang tidak bercabang, disirami dengan air yang sama. Kami melebihkan sebahagian tanam-tanaman itu atas sebahagian yang lain tentang rasanya. Sesungguhnya pada yang demikian itu terdapat tanda-tanda (kebesaran Allah) bagi kaum yang berfikir.

Ibnu Katsir dalam menafsirkan Q.S. Ar-Ra'd ayat 4 menyebutkan bahwa:

"Perbedaan pada buah-buahan dan tanam-tanaman ini adalah hal bentuk, warna, rasa, bau, daun-daun, dan bunga-bunganya. Semuanya menyandarkan kehidupannya dari satu sumber, yaitu air, akan tetapi kejadiannya berbeda-beda dengan perbedaan yang cukup banyak tak terhitung. Dalam kesemuanya itu terkandung tanda-tanda kekuasaan ALLAH bagi orang yang menggunakan pikirannya. Hal ini menandakan bahwa dunia ini berisi sains yang perlu difikirkan (Tafsir Ibnu Katsir, juz 13:111)."

Ibnu Katsir menyebutkan bahwa manusia haruslah menggunakan pikirannya.

Manusia dibekali akal dan pikiran, semata-mata supaya digunakan untuk berpikir.

Dari ayat-ayat di atas dapat diketahui bahwa setiap permasalahan pastilah memiliki solusi jika manusia tersebut mau menggunakan akal dan pikirannya.

Begitu pula dengan algoritma *growing tree* yang akan terus mencari solusi hingga solusi tersebut ditemukan.



BAB V

PENUTUP

5.1. Kesimpulan

Algoritma *growing tree* cocok dan bisa digunakan untuk mencari solusi dan membangkitkan sebuah *Maze* sempurna dengan bentuk yang berbeda-beda. Dari uji coba yang dilakukan pada proses pembangkitan *Maze*, terdapat empat varian dari *Maze* yang dibentuk. Dari setiap posisi *start* dan *finish* yang ditentukan, terdapat dua variasi lintasan untuk mencapai tujuan, sebagaimana terdapat pada tabel 4.2. Untuk jalur solusi yang ditemukan pada setiap *start* dan *finish* akan selalu menghasilkan satu jalur solusi, sebagaimana terdapat pada tabel 4.3. Sedangkan dari uji coba pengguna mengenai permainan *Math Maze*. Pengguna menyatakan bahwa *Math Maze* merupakan permainan yang menarik dan menyenangkan, serta cocok untuk pembelajaran berhitung dasar. Hal ini dibuktikan dengan hasil tabulasi responden secara keseluruhan yang terdapat pada tabel 4.8. Dari total 20 responden, sebanyak 87% responden menyukai *Math Maze* dan 13% responden tidak menyukai *Math Maze*.

5.2. Saran

Tentunya masih banyak kekurangan dalam penelitian aplikasi permainan *Math Maze* ini. Oleh karena itu, terdapat beberapa saran dari penulis untuk pengembangan selanjutnya, di antaranya:

1. Menerapkan metode *growing tree (random)* dalam proses pembangkitan dan penyelesaian *Maze*, karena penulis hanya menerapkan metode *growing tree (newest)* atau menerapkan algoritma yang lebih baik.

2. Mengembangkan aplikasi permainan *Math Maze* ini menjadi lebih menarik, baik dari segi tampilan, alur permainan, dan level permainan.
3. Mengembangkan aplikasi permainan *Math Maze* untuk *platform Android* ataupun *Windows Phone*.
4. Menerapkan pembelajaran huruf hijaiyah pada aplikasi permainan *Math Maze* sebagai media pembelajaran dasar.





DAFTAR PUSTAKA

- Apple Inc Developer. 2011. *iOS Technology Review*. California: Apple Inc.
- Beck, John C., Mitchell Wade. 2006. *The Kids Are Alright: How The Gamers Generations is Changing The Workplace*. diterjemahkan oleh: Isman H. Suryaman. PT. Grasindo: Jakarta.
- Buck, Erik. M., Donald A. Yacktman. 2010. *Cocoa Design Patterns*. USA: Pearson Education Inc.
- Buck, Jamis. 2011. Maze Generation: Growing Tree Algorithm. <http://weblog.jamisbuck.org/2011/1/27/maze-generation-growing-tree-algorithm> diakses pada 30 Mei 2012.
- Cowen, Tyler, Michelle Dawson. 2009. *What does the Turing test really mean? And how many human beings (including Turing) could pas.*
- Grace, Lindsay. 2005. Game Type and Game Genre. http://www.lgrace.com/documents/Game_types_and_genres.pdf diakses pada 20 Juni 2012 pukul 09.11 WIB.
- Gruter, Barbara, Miriam Oks. 2008. *On the Physical Dimension of Social Interaction in Mobile Games*. Bremen: University of Applied Sciences Bremen Flughafenallee.
- Jeong, Eui Jun, Dan J. Kim. *Definitions, Key Characteristics, and Generations of Mobile Games*. 2009. USA: IGI Global.
- Kochan, Stephen G. 2009. *Programming in Objective-C 2.0*. edisi 2. USA: Pearson Education Inc.
- Kompas. 2012. Saatnya Developer Game Indonesia Mencuri Start. <http://tekno.kompas.com/read/xml/2012/05/28/14464935/Saatnya.Developer.Game.Indonesia.Mencuri.Start> diakses pada tanggal 14 Januari 2013 pukul 22.00 WIB.
- Kusrahmadi, Sigit Dwi. 2007. *Pentingnya Pendidikan Moral Bagi Anak Sekolah Dasar*. Jurnal Dinamika Pendidikan No. 1 / Th. XIV.
- Millington, Ian. 2006. *Artificial Intelligence For Games*. San Francisco: Elseveir Inc.
- Randel, J. M., B. A. Morris, Wetzel, C. Douglas Wetzel, dan B. V. Whitehill. 1992. *The Effectiveness of Games for Educational Purposes: A Review of Recent Research*. California Navy Personnel Research and Development Center. No. 3, Vol. 23, hal. 261-276.
- Ruseffendi. 1997. *Pendidikan Matematika 3*. Jakarta: Universitas Terbuka.

- _____. 2006. *Pengantar kepada Membantu Guru Mengembangkan Kompetensinya dalam Pengajaran Matematika untuk Meningkatkan CBSA*. edisi revisi. Bandung: Tarsito.
- Clair, Robert. 2011. *Learning Objective-C 2.0: A Hands-On Guide to Objective-C for Mac and iOS Developers*. USA: Pearson Education Inc.
- Soedjadi, R. 2000. *Kiat-Kiat Pendidikan Matematika di Indonesia*. Jakarta: Dirjen Dikti Departemen Pendidikan Nasional.
- Strougo, Rod, Ray Wenderlich. 2011. *Learning Cocos2D: A Hands-On Guide to Building iOS Games with Cocos2D, Box2D, and Chipmunk*. USA: Pearson Education Inc.
- Sumarno. 2011. Mengembangkan Bakat Anak. Universitas Surabaya. <http://blog.elearning.unesa.ac.id/alim-sumarno/mengembangkan-bakat-anak> diakses 16 Juni 2012 pukul 06.24 WIB.
- Teneng, Joko Purwadi, Erick Kurniawan. 2010. *Penerapan Algoritma Backtracking Pada Permainan Math Maze*. Universitas Kristen Duta Wacana Yogyakarta.
- Zigma, Hawke. 2011. Pentingnya Mengenalkan Perkembangan Game di Dunia Pendidikan. <http://www.duniaku.net/2011/05/23/pentingnya-mengenalkan-perkembangan-game-di-dunia-pendidikan/> diakses pada 12 Januari 2013 pukul 10.00 WIB.

Lampiran 1

Langkah-Langkah Membangun *Game Math Maze*

a. Inisialisasi Maze

1. Tentukan ukuran grid dari Maze berukuran (x,y) .

Misal, apabila ukuran grid = 5, maka *Maze* tersebut berukuran 5x5.

2. Lakukan perulangan terhadap Maze berukuran (x,y) .
3. Letakkan dinding (#) pada posisi (x,y) dengan nilai x atau y apabila $\% 2 = 0$.
4. Letakkan ruang kosong pada posisi (x,y) dengan nilai x atau y apabila $\% 2 \neq 0$.
5. Ulangi langkah 2-4 untuk *Maze* berukuran (x,y) .

Berikut ini adalah contoh dari hasil inisialisasi Maze yang dilakukan:

```
#####
#   #
#####
#   #
#####
```

b. Pembangkitan Maze

1. Tentukan titik awal yang digunakan sebagai acuan awal dalam proses pembangkitan Maze.

Aturan untuk menentukan posisi titik awal pembangkitan *Maze* adalah sebagai berikut:

- a. Titik awal pembangkitan *Maze* harus dimulai dari ruang kosong yang telah dibangkitkan pada saat inisialisasi *Maze*. Misal, terdapat Maze berukuran 5x5, maka koordinat *start* yang bisa digunakan adalah (1,1), (1,3), (3,1), atau (3,3)

2. Tentukan total sel = grid x grid.

Misal, terdapat *Maze* berukuran 5x5 maka total sel yang dimiliki adalah 25.

3. Pilih secara acak sel tetangga terdekat secara horizontal ataupun vertikal.
4. Jika sel belum pernah dikunjungi, maka hapus dinding dan pindah ke sel tersebut. Jika sel sudah pernah dikunjungi, maka hapus sel dari daftar.
5. Ulangi langkah 3 dan 4 hingga total sel yang dikunjungi = total sel pada grid.
6. Tentukan posisi *start* (S) dan *finish* (G).

Aturan untuk menentukan posisi *start* dan *finish* adalah sebagai berikut:

- a. Tentukan posisi *start* dan *finish* secara bersebrangan. Misal, pada *Maze* berukuran 5x5 apabila *start* terdapat pada koordinat (1,1) maka *finish* harus terdapat pada koordinat yang bersebrangan, yaitu (1,4), (2,4), dan (3,4) seperti pada contoh dari hasil *Maze* yang telah dibangkitkan di bawah.
- b. Posisi *start* atau *finish* tidak boleh berada pada bagian pojok dari *Maze*. Misal, pada *Maze* berukuran 5x5. Posisi *start* atau *finish* tidak boleh berada pada koordinat (0,0), (0,4), (4,0), (4,4)

Berikut ini adalah contoh dari hasil *Maze* yang telah dibangkitkan:

```
#####
S.#.#
#.#.G
#...#
#####
```

c. Pencarian Jalur Solusi

1. Lakukan pencarian dari posisi *start* yang telah ditentukan sebelumnya.

Misal, apabila koordinat *start* (1,1) maka pencarian jalur solusi dimulai dari koordinat (1,1) dengan memilih tetangga terdekat yang bisa dikunjungi.

2. Pilih secara acak sel tetangga terdekat secara horizontal ataupun vertikal.

3. Jika sel belum pernah dikunjungi, maka pindah ke sel tersebut. Jika sel menemukan jalan buntu (*dead end*) ataupun sel sudah pernah dikunjungi, maka hapus sel dari daftar (x).
4. Ulangi langkah 2 sampai 3 hingga menemukan posisi *finish*.
5. Tandai jalur solusi yang didapatkan dengan tanda (+).

Berikut ini adalah contoh dari hasil pencarian jalur solusi yang didapatkan:

```
#####
S+##x#
#+##+G
#+++#
#####
```

d. Mengubah Jalur Solusi Menjadi Angka-Angka Pemandu

1. Hitung total sel yang dikunjungi jalur solusi (+) pada setiap baris dan kolom dari Maze berukuran (x,y).
2. Tampilkan papan permainan *Math Maze*.
3. Tampilkan angka-angka pemandu sebagai indikator untuk mencapai tujuan.

Berikut ini adalah contoh dari hasil angka-angka yang didapatkan:

```
312
#####
1S+##x#
2#+##+G
3#+++#
#####
```

Dari *Maze* berukuran 5x5 akan didapatkan *Math Maze* berukuran 3x3. Hal ini disebabkan karena permainan *Math Maze* tidak membutuhkan dinding-dinding luar yang terdapat pada *Maze*, kecuali posisi *start* dan *finish*. *Math Maze* hanya menggunakan area dalam dari *Maze* yang dibangkitkan. Untuk *Maze* berukuran

7x7 akan didapatkan *Math Maze* berukuran 5x5, begitu juga untuk *Maze* berukuran 9x9 akan didapatkan *Math Maze* berukuran 7x7, dan seterusnya. Berikut ini adalah papan permainan *Math Maze* berukuran 3x3 dari hasil pembangkitan *Maze* berukuran 5x5.

	3	1	2
1	S	+	#x
2	+	#	+G
3	+	+	+

Lampiran 2

Variasi Lintasan *Maze* Tanpa Dinding

Contoh *Maze* berukuran 5x5 tanpa menggunakan dinding:

#####
S...G
#...#
#...#
#####

Beberapa hasil variasi lintasan *Maze* berukuran 5x5 tanpa menggunakan dinding:

##### S123G #...# #...# #####	##### S125G #.34# #...# #####	##### S127G #.36# #.45# #####
##### S129G #438# #567# #####	##### S1.5G #234# #...# #####	##### S1.7G #236# #.45# #####
##### S1.7G #2.6# #345# #####	##### S1.7G #256# #34.# #####	##### S167G #25.# #34.# #####
##### S145G #23.# #...# #####	##### S189G #276# #345# #####	

start (1,0) dan *finish* (1,4)

##### S123# #..4G #...# #####	##### S12.# #.34G #...# #####	##### S1..# #234G #...# #####
##### S12.# #.36G #.45# #####	##### S12.# #438G #567# #####	##### S1..# #2.6G #345# #####
##### S1..# #256G #34.# #####	##### S167# #258G #34.# #####	##### S1..# #236G #.45# #####

start (1,0) dan finish (2,4)

##### #...# S123G #...# #####	##### #1..# S125G #.34# #####	##### #.34# S125G #...# #####
##### #...# S1.5G #234# #####	##### #234# S1.5G #...# #####	##### #...# S145G #23.# #####
##### #23.# S145G #...# #####	##### #.56# S147G #23.# #####	##### #23.# S147G #.56# #####

start (2,0) dan finish (2,4)

##### S123# #..4# #..5G #####	##### S1..# #2..# #345G #####	##### S123# #.54# #.67G #####
##### S123# #654# #789G #####	##### S12.# #43.# #567G #####	##### S1..# #234# #..5G #####
##### S1..# #23.# #.45G #####	##### S167# #258# #349G #####	##### S1..# #256# #347G #####
##### S12.# #.34# #..5G #####	##### S12.# #.3.# #.45G #####	

start (1,0) dan finish (3,4)

Tabel Variasi Lintasan Maze Tanpa Dinding

No	Start (x,y) / (baris, kolom)	Finish (x,y) / (baris, kolom)	Variasi lintasan
1	1,0	1,4	11
2	2,0	2,4	9
3	3,0	3,4	11
4	1,0	2,4	9
5	2,0	3,4	9
6	3,0	1,4	11
7	1,0	3,4	11
8	2,0	1,4	9
9	3,0	2,4	9
10	1,4	1,0	11
11	2,4	2,0	9
12	3,4	3,0	11
13	1,4	2,0	9
14	2,4	3,0	9
15	3,4	1,0	11

No	Start (x,y) / (baris, kolom)	Finish (x,y) / (baris, kolom)	Variasi lintasan
16	1,4	3,0	11
17	2,4	1,0	9
18	3,4	2,0	9
19	0,1	4,1	11
20	0,2	4,2	9
21	0,3	4,3	11
22	0,1	4,2	9
23	0,2	4,3	9
24	0,3	4,1	11
25	0,1	4,3	11
26	0,2	4,1	9
27	0,3	4,2	9
28	4,1	0,1	11
29	4,2	0,2	9
30	4,3	0,3	11
31	4,1	0,2	9
32	4,2	0,3	9
33	4,3	0,1	11
34	4,1	0,3	11
35	4,2	0,1	9
36	4,3	0,2	9

Dari hasil variasi lintasan dengan *Maze* tanpa dinding dihasilkan, bahwa apabila posisi *start* atau *finish* semakin mendekati dinding luar, maka semakin banyak pula variasi lintasan yang dihasilkan. Apabila posisi *start* atau *finish* semakin menjauh dari dinding luar, maka semakin sedikit pula variasi lintasan yang dihasilkan. Dengan asumsi, bahwa tidak boleh mengulangi jalur yang sudah terlewati.

Lampiran 3

Data Lengkap Responden Uji Coba

No	Nama	Jenis Kelamin	Kelas
1	Robi	L	1
2	Rafdan	L	1
3	Aril	L	1
4	Rian	L	1
5	Navisa	P	1
6	Reva	P	2
7	Findi	P	2
8	Mela	P	2
9	Galuh	P	2
10	Farhan	L	2
11	Wulan	P	3
12	Adi	L	3
13	Arya	L	3
14	Elita	P	3
15	Reza	L	3
16	Ibnu	L	4
17	Abel	L	4
18	Dika	L	4
19	Tahir	L	4
20	Salsa	P	4

(SDN Pagentan 3 Jalan Ronggolawe Gang 2 Kecamatan Singosari Malang)

Lampiran 4

Keterangan Jawaban Kuisisioner

No Pertanyaan	Pertanyaan	Keterangan Jawaban
P1	Apakah desain <i>game</i> menarik?	a. Ya
		b. Tidak
P2	Apakah efek suara dan desain <i>game</i> menarik?	a. Ya
		b. Tidak
P3	Apakah Anda mengetahui cara memainkan <i>game Math Maze</i> ?	a. Ya
		b. Tidak
P4	Apakah Anda merasa kesulitan dalam memainkan <i>game Math Maze</i> ?	a. Ya
		b. Tidak
P5	Apakah Anda merasa bosan memainkan <i>game Math Maze</i> ?	a. Ya
		b. Tidak
P6	Apakah <i>game Math Maze</i> menarik dan menyenangkan?	a. Ya
		b. Tidak
P7	Apakah Anda lebih senang belajar berhitung dengan <i>game Math Maze</i> ?	a. Ya
		b. Tidak
P8	Apakah Anda ingin memainkan <i>game Math Maze</i> lagi?	a. Ya
		b. Tidak

Lampiran 5

Data Rekapitulasi Hasil Jawaban Kuisioner Responden

No Res	P1	P2	P3	P4	P5	P6	P7	P8
01	a	a	a	b	b	a	a	a
02	a	a	b	a	a	a	a	b
03	a	a	b	a	a	a	b	a
04	a	a	a	a	b	a	a	a
05	a	a	a	a	b	a	a	a
06	a	a	a	a	a	a	a	a
07	a	a	a	b	b	a	a	a
08	a	a	a	b	b	a	a	a
09	a	a	b	b	a	a	a	a
10	a	a	a	a	b	a	a	a
11	a	a	a	b	b	a	a	a
12	a	a	a	b	b	a	a	a
13	a	a	a	b	b	a	a	a
14	a	a	a	b	b	a	a	a
15	a	a	a	b	b	a	a	a
16	a	a	a	b	b	a	a	a
17	a	a	a	b	b	a	a	a
18	a	a	a	a	a	b	b	b
19	a	a	a	b	b	a	a	a
20	a	a	a	a	b	a	a	a

Lampiran 6

Dokumentasi Uji Coba Responden



