

**IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON**

SKRIPSI



**Disusun Oleh :
Widi Nugroho
08.18.143**

**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON

SKRIPSI

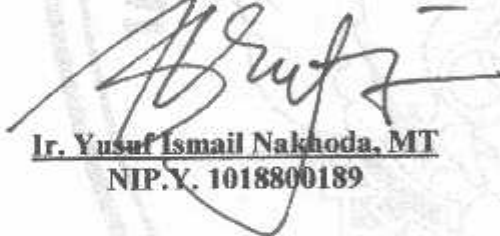
*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

Disusun Oleh :


WIDI NUGROHO
NIM : 08.18.143

Diperiksa dan Disetujui

Dosen Pembimbing I


Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189


Dosen Pembimbing II


Febriana Santi Wahyuni, S.Kom, M.Kom
NIP.P. 1031000425

Mengetahui

Ketua Program Studi Teknik Informatika S-1




Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

**PRODI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON

Widi Nugroho (NIM. 0818143)

Teknik Informatika S-1, Institut Teknologi Nasional Malang
e-mail : wdhee_sock@yahoo.com

Dosen Pembimbing : I. Ir. Yusuf Ismail Nakhoda, MT.
II. Febriana Santi Wahyuni, S.Kom, M.Kom.

Abstrak

Adventure Game adalah game bergenis petualangan dengan alur cerita yang berkesinambungan disertai perpindahan atau pergantian tempat dari satu wilayah ke wilayah yang lain. Kecerdasan buatan didefinisikan sebagai kecerdasan yang ditujukan untuk suatu entitas buatan. Sebuah kecerdasan yang diciptakan dan dimasukkan kedalam mesin atau objek komputer agar dapat melakukan tindakan alami selayaknya manusia. Pada penelitian ini akan dibangun sebuah *game* sederhana yang bergenis *adventure*, yaitu *game* Malam Jum'at Kliwon. Algoritma *Finite State Machine* (FSM) diterapkan kepada musuh, dengan memberikan sensor sebagai indikator respon terhadap pergerakan pemain. *Game adventure* ini dibangun dengan memanfaatkan *Adobe Flash CS3* dan bahasa pemrograman *Actionscript 2.0*, sehingga proses perancangan karakter, animasi, dan algoritma *FSM* dapat dicapai. Dalam *game* ini terdapat dua jenis musuh, yaitu musuh Hantu Anak Buah dan Hantu Boss. Perilaku yang mampu dilakukan oleh karakter Hantu Anak Buah meliputi mencari pemain, mengejar pemain, dan menghindari serangan pemain. Sedangkan, perilaku yang mampu dilakukan oleh karakter Hantu Boss meliputi mencari pemain, mengejar pemain dan menembak pemain, serta menghindari serangan pemain.

Tingkat keberhasilan yang diperoleh adalah 100% berdasarkan tabel pengujian kecerdasan Hantu Anak Buah dan tabel pengujian kecerdasan Hantu Boss. Saran yang dapat diberikan pada skripsi Implementasi *Finite State Machine* Pada *Game* Malam Jum'at Kliwon, meliputi kedepannya pada *game* ini terdapat *high score* untuk pemain dan kedepannya pada *game* ini terdapat fungsi *save* untuk pemain.

Kata kunci : *Actionscript 2.0*, *Adobe Flash CS3*, *Adventure Game*, *Finite State Machine*, *Game*.

Abstract

Adventure Game is type of game with a continuous storyline and movement or change of place from one area to another. Artificial intelligence is defined as intelligence devoted to an artificial entity. An intelligence that created and embedded into the machine or computer objects in order to perform proper actions of human nature. In this research will built a simple Adventure Game, the game is Malam Jum'at Kliwon. Finite State Machine (FSM) algorithm is applied to the enemies by adding sensors as an indicator of response to the player's movements. This game is built using Adobe Flash CS3 with ActionScript 2.0 as programming language, so that the process of character designing, animationing, and implementing of FSM algorithm can be achieved. In this game there are two enemy characters, the enemy of Ghost and Boss Ghost. Actions that can be done by the Ghost such as looking for the player, chasing the player, and evade from player's attacks. Meanwhile, actions that can be by the Boss Ghost such as looking for player, chasing player with shooting aim player, and evade from player's attacks.

The success rate was 100%, acquired by testing artificial intelligence's chart of Ghost and artificial intelligence's chart of Boss Ghost. The suggestions which can be given to thesis of Implementasi Finite State Machine Pada Game Malam Jum'at Kliwon, such as in the future this game there are high score and save function for the sake of player.

Keywords : *Actionscript 2.0, Adobe Flash CS3, Adventure Game, Finite State Machine, Game.*

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : WIDI NUGROHO

NIM : 08.18.143

Program Studi : Teknik Informatika S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Februari 2013

Yang membuat Pernyataan,



Widi Nugroho
NIM. 08.18.143

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas karunia, rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul **“IMPLEMENTASI FINITE STATE MACHINE PADA GAME MALAM JUM’AT KLIWON”**.

Skripsi ini dapat terselesaikan tidak terlepas dari dukungan berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Kepala Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing I yang telah memberikan saran dan bimbingannya dalam penyusunan skripsi ini.
5. Ibu Febriana Santi Wahyuni, S.Kom, M.Kom selaku Dosen Pembimbing II yang telah memberikan saran dan bimbingannya dalam penyusunan skripsi ini.
6. Rekan-rekan Teknik Informatika dan berbagai pihak yang turut membantu dalam penyelesaian laporan ini.

Semoga apa yang telah disajikan dapat memberikan manfaat dan pengetahuan bagi para pembaca. Segala kritik dan saran yang bersifat membangun, diterima dengan senang hati sebagai tambahan ilmu pengetahuan demi membangun kesempurnaan skripsi ini.

Malang, 13 Februari 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABTRAKSI	ii
SURAT PERNYATAAN ORISINALITAS	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xiii

BAB I PENDAHULUAN

1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan Penelitian	2
1.4. Batasan Masalah	2
1.5. Metode Penelitian	3
1.6. Sistematika Penulisan	3

BAB II LANDASAN TEORI

2.1. Game	5
2.2. Sejarah Game	6
2.3. Syarat Perancangan Game	10
2.4. Jenis Game	11
2.5. Kecerdasan Buatan	13
2.6. Metode Kecerdasan Buatan	13
2.6.1. Searching	13
2.6.2. Reasoning	16
2.6.3. Planning	17
2.6.4. Learning	18

2.7. Finite State Machine	19
2.8. Diagram Alir	19
2.9. Perangkat Lunak Pendukung	21
2.9.1. Adobe Flash CS3	21
2.9.2. Adobe Premiere Pro CS3	23

BAB III ANALISA DAN PERANCANGAN

3.1. Analisa	24
3.1.1. Kebutuhan Fungsional	24
3.1.2. Kebutuhan Non Fungsional	25
3.2. Perancangan Game	27
3.2.1. Alur Cerita	27
3.2.2. Alur Permainan	27
3.2.3. Perancangan Kecerdasan Musuh	31
3.3. Proses Perhitungan	33
3.3.1. Perhitungan kecerdasan musuh	33
3.3.2. Perhitungan score	38
3.4. Properti Game	39
3.4.1. Perancangan Menu Utama	39
3.4.2. Perancangan Tampilan Bantuan	40
3.4.3. Perancangan Tampilan Profil	41
3.4.4. Perancangan Tampilan Quit	42
3.4.4. Perancangan Tampilan Story Board	42
3.4.6. Perancangan Tampilan Arena Game	42
3.4.7. Perancangan Tampilan Ending	48
3.4.8. Perancangan Tampilan Game Over	49
3.4.9. Perancangan Tampilan Score	49
3.5. Komposisi Game	50
3.5.1. Perancangan Karakter Pemain	50
3.5.2. Perancangan Karakter Musuh	50
3.5.3. Perancangan Peluru	52

3.5.4. Perancangan Healthbar	54
3.5.5. Perancangan Audio.....	55

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi.....	56
4.1.1. Implementasi Menu Utama	56
4.1.2. Implementasi Tampilan Bantuan	57
4.1.3. Implementasi Tampilan Profil	58
4.1.4. Implementasi Tampilan Quit	59
4.1.5. Implementasi Arena Game	59
4.1.6. Implementasi Tampilan Ending	68
4.1.7. Implementasi Tampilan Score	68
4.1.8. Implementasi Tampilan Game Over	69
4.2. Pengujian.....	69
4.2.1. Pengujian Kontrol	69
4.2.2. Pengujian Kecerdasan Hantu Anak Buah	72
4.2.3. Pengujian Kecerdasan Hantu Boss	72
4.2.4. Pengujian Nilai Pemain	73
4.2.5. Pengujian Kebutuhan Fungsional	75
4.2.6. Pengujian Kebutuhan Non Fungsional.....	77

BAB V PENUTUP

5.1. Kesimpulan	86
5.2. Saran.....	87

DAFTAR PUSTAKA.....	88
----------------------------	-----------

DAFTAR TABEL

Tabel 2.1. Simbol Diagram Alir	20
Tabel 2.2. Tools Pada Adobe Flash CS3	22
Tabel 3.1. Tabel Jumlah Hantu	32
Tabel 3.2. Tabel FSM Hantu Anak Buah	34
Tabel 3.3. Tabel FSM Hantu Boss.....	36
Tabel 4.1. Tabel Pengujian Kecerdasan Hantu Anak Buah.....	72
Tabel 4.2. Tabel Pengujian Kecerdasan Hantu Boss	73
Tabel 4.3. Tabel Pengujian Kebutuhan Fungsional Hantu Anak Buah	76
Tabel 4.4. Tabel Pengujian Kebutuhan Fungsional Hantu Boss	76
Tabel 4.5. Tabel Pengujian Kebutuhan Non Fungsional Secara Performance	78
Tabel 4.6. Tabel Pengujian Non Fungsional Interface Pada Acer Aspire 4530	80
Tabel 4.7. Tabel Pengujian Non Fungsional Interface Pada Dell N 4050	81
Tabel 4.8. Tabel Pengujian Non Fungsional Interface Pada Acer Aspire 4738G	82
Tabel 4.9. Tabel Pengujian Non Fungsional Interface Pada Acer Aspire One AOA 150 ..	83
Tabel 4.10. Tabel Pengujian Non Fungsional Interface Pada Macbook Pro 13"	84
Tabel 4.11. Tabel Pengujian Non Fungsional Interface Pada Personal Computer.....	85

DAFTAR GAMBAR

Gambar 2.1. Contoh Alur Diagram Finite State Machine Sederhana	19
Gambar 2.2. Tampilan Workspace Adobe Flash CS3	22
Gambar 2.3. Tampilan Workspace Adobe Premiere Pro CS3	23
Gambar 3.1. Flowchart Perancangan Game	28
Gambar 3.2. Perancangan Kecerdasan Musuh	31
Gambar 3.3. Blok Diagram Finite State Machine Hantu Anak Buah	33
Gambar 3.4. Flowchart Finite State Machine Hantu Anak Buah	35
Gambar 3.5. Blok Diagram Finite State Machine Hantu Boss	36
Gambar 3.6. Flowchart Finite State Machine Hantu Boss	38
Gambar 3.7. Rancangan Tampilan Menu Utama	39
Gambar 3.8. Rancangan Button Menu Utama	39
Gambar 3.9. Rancangan Tampilan Peta	40
Gambar 3.10. Rancangan Tampilan Navigasi	40
Gambar 3.11. Rancangan Tampilan Petunjuk	41
Gambar 3.12. Rancangan Tampilan Profil	41
Gambar 3.13. Rancangan Tampilan Quit	42
Gambar 3.14. Rancangan Tampilan Story Board	42
Gambar 3.15. Rancangan Tampilan Arena.1	43
Gambar 3.16. Rancangan Tampilan Arena.2	43
Gambar 3.17. Rancangan Tampilan Arena Boss.1	44
Gambar 3.18. Rancangan Tampilan Arena.3	44
Gambar 3.19. Rancangan Tampilan Arena.4	44
Gambar 3.20. Rancangan Tampilan Arena.5	45
Gambar 3.21. Rancangan Tampilan Arena Boss.2	45
Gambar 3.22. Rancangan Tampilan Arena.6	45
Gambar 3.23. Rancangan Tampilan Arena.7	46
Gambar 3.24. Rancangan Tampilan Arena.8	46
Gambar 3.25. Rancangan Tampilan Arena Boss.3	46
Gambar 3.26. Rancangan Tampilan Arena.9	47

Gambar 3.27. Rancangan Tampilan Arena.10	47
Gambar 3.28. Rancangan Tampilan Arena.11	47
Gambar 3.29. Rancangan Tampilan Arena Boss.4	48
Gambar 3.30. Rancangan Tampilan Arena.12	48
Gambar 3.31. Rancangan Tampilan Ending	48
Gambar 3.32. Rancangan Tampilan Game Over	49
Gambar 3.33. Rancangan Tampilan Score	49
Gambar 3.34. Rancangan Karakter Pemain.....	50
Gambar 3.35. Rancangan Karakter Hantu Anak Buah	51
Gambar 3.36. Rancangan Karakter Hantu Boss.1	51
Gambar 3.37. Rancangan Karakter Hantu Boss.2	51
Gambar 3.38. Rancangan Karakter Hantu Boss.3	52
Gambar 3.39. Rancangan Karakter Hantu Boss.4	52
Gambar 3.40. Rancangan Amunisi Bawang Putih	52
Gambar 3.41. Rancangan Amunisi Bawang Merah	53
Gambar 3.42. Rancangan Amunisi Hantu Boss.1	53
Gambar 3.43. Rancangan Amunisi Hantu Boss.2	53
Gambar 3.44. Rancangan Amunisi Hantu Boss.3	53
Gambar 3.45. Rancangan Amunisi Hantu Boss.4	53
Gambar 3.46. Rancangan Healthbar Pemain	54
Gambar 3.47. Rancangan Healthbar Hantu Boss.1	54
Gambar 3.48. Rancangan Healthbar Hantu Boss.2	54
Gambar 3.49. Rancangan Healthbar Hantu Boss.3	54
Gambar 3.50. Rancangan Healthbar Hantu Boss.4.....	55
Gambar 3.51. Rancangan Suara Karakter Bajaj Ketika Berjalan	55
Gambar 4.1. Tampilan Menu Utama	56
Gambar 4.2. Tampilan Button Menu Utama	57
Gambar 4.3. Tampilan Peta	57
Gambar 4.4. Tampilan Navigasi	57
Gambar 4.5. Tampilan Petunjuk Permainan.....	58
Gambar 4.6. Tampilan Profil	58

Gambar 4.7. Tampilan Opsi Quit	59
Gambar 4.8. Tampilan Arena.1	60
Gambar 4.9. Tampilan Arena.2	60
Gambar 4.10. Tampilan Arena Boss.1	61
Gambar 4.11. Tampilan Arena.3	61
Gambar 4.12. Tampilan Arena.4	62
Gambar 4.13. Tampilan Arena.5	62
Gambar 4.14. Tampilan Arena Boss.2	63
Gambar 4.15. Tampilan Arena.6	63
Gambar 4.16. Tampilan Arena.7	64
Gambar 4.17. Tampilan Arena.8.....	64
Gambar 4.18. Tampilan Arena Boss.3	65
Gambar 4.19. Tampilan Arena.9	65
Gambar 4.20. Tampilan Arena.10	66
Gambar 4. 21. Tampilan Arena.11	66
Gambar 4. 22. Tampilan Arena Boss.4	67
Gambar 4. 23. Tampilan Arena.12.....	67
Gambar 4. 24. Tampilan Ending	68
Gambar 4.25. Tampilan Total Score	68
Gambar 4.26. Tampilan Game Over.....	69
Gambar 4.27. Pengujian Kontrol Arah Pemain	70
Gambar 4.28. Pengujian Kontrol Tembakan Bawang Putih	70
Gambar 4.29. Pengujian Kontrol Tembakan Bawang Merah	71
Gambar 4.30. Pengujian Button Pause	71
Gambar 4. 31. Score Pemain Untuk Hantu Anak Buah	73
Gambar 4. 32. Hantu Anak Buah Mati	74
Gambar 4. 33. Hantu Boss Mati	74
Gambar 4. 34. Score Pemain Untuk Hantu Boss	74
Gambar 4. 35. Score Akhir Pemain	75
Gambar 4. 36. Akumulasi Score Akhir Pemain.....	75

DAFTAR LAMPIRAN

- Lampiran.1. Berita Acara Ujian Skripsi
- Lampiran.2. Formulir Perbaikan Skripsi
- Lampiran.3 Formulir Bimbingan Skripsi
- Lampiran.4. Penggalan *actionsript* Hantu Anak Buah berjalan *vertical*
- Lampiran.5. Penggalan *actionsript* Hantu Anak Buah berjalan *horizontal*
- Lampiran.6. Penggalan *actionsript* Hantu Anak Buah mengejar pemain
- Lampiran.7. Penggalan *actionsript* Hantu Anak Buah menghindar
- Lampiran.8. Penggalan *actionsript* Hantu Boss berjalan *horizontal*
- Lampiran.9. Penggalan *actionsript* Hantu Boss mengejar dan menembak pemain
- Lampiran.10. Penggalan *actionsript* Hantu Boss menghindar

BAB I

PENDAHULUAN

1.1. Latar Belakang

Game merupakan media hiburan yang telah berkembang pesat di masyarakat saat ini. Perkembangan *game* dengan menampilkan grafis, dan sistem yang lebih baik menjadi hal utama untuk menarik minat para pencintanya.

Seiring dengan perkembangan *gadget*, *game* menjadi banyak diminati oleh kalangan anak-anak hingga dewasa. Namun, di sisi lain banyak riset yang menunjukkan adanya dampak negatif dalam bermain *game* khususnya pada kalangan anak-anak. Anak akan larut dalam dunia *game* itu sendiri, sehingga anak cenderung apatis atau acuh terhadap sekitarnya yang berdampak anak tersebut akan menjadi individu yang kurang pergaulan di kemudian hari. Selain itu, tak jarang anak akan meniru atau mengimplementasikan apa yang ada di dalam *game* ke dalam kehidupannya. Namun, sebenarnya tidak hanya dampak negatif yang ditimbulkan dari bermain *game* tetapi ada pula dampak positifnya, seperti mengurangi kejenuhan, membangun kerja sama tim, serta memperluas daya berfikir dalam memecahkan suatu masalah.

Salah satu unsur yang berperan penting dalam membangun sebuah *game* yaitu kecerdasan buatan. Kecerdasan buatan adalah bidang ilmu komputer yang meneliti dan mempelajari bagaimana agar komputer dapat bertindak seperti yang dilakukan oleh manusia. Dengan adanya kecerdasan buatan, elemen-elemen dalam *game* dapat berperilaku sealam mungkin layaknya manusia. Implementasi kecerdasan buatan dalam *game*, terletak pada salah satu elemennya yaitu NPC (*Non Playable Character*) atau musuh. Yang mana, musuh mampu mengenali aksi dari pemain sekaligus merespon dengan reaksi sesuai dengan acuannya.

Adventure Game adalah *game* berjenis petualangan dengan alur cerita yang berkesinambungan disertai perpindahan atau pergantian tempat dari satu wilayah ke wilayah yang lain. *Game* ini banyak diminati masyarakat, karena memiliki jalan cerita yang menarik, terdapat *level*, terdapat *item* yang dapat membantu atau merugikan pemain dalam permainan.

Finite State Machine adalah suatu model kecerdasan buatan yang memiliki sejumlah *state* (keadaan) dan dalam waktu tertentu dapat berubah sesuai dengan *input* yang diberikan. Dia mampu mengenali *input* dan menghasilkan transisi dari *state* satu ke *state* lain atau menghasilkan *output* berupa aksi.

1.2. Rumusan Masalah

Adapun rumusan masalah yang akan dibahas dalam skripsi ini adalah bagaimana mengimplementasikan *Finite State Machine* (FSM) sebagai kecerdasan untuk menentukan reaksi musuh dalam *game* Malam Jum'at Kliwon.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah mengimplementasikan *Finite State Machine* (FSM) sebagai perilaku musuh terhadap pemain. Dengan memberikan sensor pada musuh, kemudian dari sensor tersebut musuh mampu memberi reaksi kepada pemain.

1.4. Batasan Masalah

Adapun batasan masalah dalam membangun *game* Malam Jum'at Kliwon, antara lain:

1. *Platform* yang digunakan dalam membangun *game* adalah *Adobe Flash CS3* dengan bahasa pemrograman *Actionscript 2.0*.
 2. Algoritma yang diterapkan adalah *Finite State Machine* (FSM) untuk menentukan perilaku musuh terhadap pemain.
 3. *Game* yang akan dibangun adalah berjenis *Adventure game*.
 4. Kecerdasan terletak pada perilaku musuh ketika sensor mengetahui keberadaan pemain dan menghindari serangan pemain.
 5. Adanya Hantu Boss dan Hantu Anak Buah sebagai musuh yang harus dikalahkan oleh pemain.
-

1.5. Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut :

1. Studi literatur

Teknik pengumpulan data dengan mencari bahan-bahan referensi dari berbagai sumber sebagai landasan teori yang berhubungan dengan permasalahan yang akan dijadikan objek penelitian.

2. Analisa kebutuhan

Data dan informasi yang telah diperoleh akan dianalisa agar diperoleh suatu kerangka sebagai sumber acuan perancangan *game*.

3. Perancangan

Setelah pengumpulan data selesai, berlanjut pada tahap perancangan *game*.

4. Implementasi

Data-data yang telah terkumpul, kemudian diimplementasikan ke dalam *game* bersama dengan *Finite State Machine*.

5. Pengujian

Jika aplikasi telah selesai selanjutnya ketahap pengujian, hal ini dilakukan untuk mengetahui apakah *game* yang dirancang sudah benar atau masih terdapat kesalahan.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka sistematika penulisan dibagi ke dalam 5 bab, antara lain :

BAB I : Pendahuluan

Berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.

BAB II : Landasan Teori

Berisi tentang tinjauan pustaka mengenai permasalahan yang berhubungan dengan pembuatan *game* ini.

BAB III : Analisis dan Perancangan

Berisi mengenai analisis kebutuhan dan perancangan *game* yang akan dibuat.

BAB IV : Implementasi dan Pengujian

Berisi implementasi terhadap proses pembuatan *game* pada tugas akhir skripsi ini dari awal hingga akhir, serta melakukan pengujian *game* yang dibuat.

BAB V : Kesimpulan dan Saran

Berisi kesimpulan dari hasil penelitian yang dilakukan dan saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya.



BAB II

LANDASAN TEORI

2.1. Game

Menurut Agustinus Nilwan, *game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan *game*. Atau jika ingin membuat *game*, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.

Dalam bahasa Indonesia “*Game*” dapat diartikan sebagai permainan. Permainan adalah suatu kegiatan yang di dalamnya terdapat peraturan-peraturan dan sebuah sistem yang mana pemain terlibat dalam suatu konflik buatan. Agar tidak terjadi kecurangan di dalam sebuah permainan, maka peraturan dibuat untuk membatasi perilaku pemain dan menentukan jalannya permainan. Sehingga, setiap pemain memiliki peluang yang sama untuk memenangkan permainan.

Game merupakan permainan yang merujuk pada kelincahan intelektual atau *intellectual capability*. Kelincahan intelektual pada tingkatan tertentu merupakan ukuran sejauh mana *game* tersebut menarik untuk dimainkan. Tidak hanya pada kelincahan intelektualnya saja, *game* juga mampu melatih kemampuan seseorang berfikir dan bertindak dalam memecahkan suatu masalah dengan tepat dan cepat. Karena, di dalam sebuah *game* terdapat berbagai konflik atau masalah yang menuntut sang pemain untuk dapat menyelesaikannya dengan hasil maksimal.

Finite State Machine merupakan metodologi kecerdasan buatan yang memperlihatkan perilaku sistem dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks. (Setiawan, 2006).

2.2. Sejarah Game

Dalam perjalanannya, *game* juga memiliki sejarah yang cukup panjang. Berikut ini adalah sejarah *game* mulai dari generasi pertama hingga masa kini :

1. Game Generasi Pertama

Pada tahun 1972, saat itu banyak orang yang masih belum mengenal *console game* dan *computer game*. Namun, sebuah perusahaan bernama *Magnavox* meluncurkan *video game* pertama yaitu *Odyssey*. *Magnavox Odyssey*, *console game* pertama di dunia yang mengoperasikan *Pong*. Tidak lama setelah itu sebuah *game arcade* legendaris *Atari* berjudul “*Pong*” muncul. *Pong* merupakan sebuah *game* sederhana yang mengambil konsep permainan tenis, dengan 1 bola dan 2 papan di kiri dan di kanan.

Pada tahun 1975, akhirnya *Magnavox* menyerah dan menghentikan produksi *Odyssey*. Sebagai gantinya mereka mengikuti jejak *Atari* dengan memproduksi mesin dindong bernama *Odyssey 100*, yang khusus menyajikan *game Pong*.

2. Game Generasi Kedua

Pada tahun 1976, *Fairchild* mencoba menghidupkan kembali dunia *video game* dengan menciptakan VES (*Video Entertainment System*). VES adalah mesin pertama yang disebut “*console*”. *Console* ini menggunakan kaset magnetik yang disebut *cartridge*. Konsep ini kemudian diikuti oleh beberapa produsen lain, termasuk *Atari*, *Magnavox*, dan *RCA*, ketiga perusahaan tersebut juga meliris *console* serupa.

Pada tahun 1977, dunia *game console* menjadi tidak populer, *game* yang ada tidak berhasil menarik minat. *Fairchild* dan *RCA* mengalami kebangkrutan. Praktis, hanya ada *Atari* dan *Magnavox* yang masih bertahan di dunia *video game*. Kemudian, pada tahun 1978 *Magnavox* meluncurkan *Odyssey 2*. Seperti halnya *Odyssey* pertama, *console* ini pun gagal di pasaran. Tak lama berselang, *Atari* meluncurkan *console* legendaris, *Atari 2600*, yang terkenal dengan *game Space Invaders*. Pada tahun 1980, berbagai produsen *console* muncul dan mereka mengambil *Atari 2600* sebagai konsep dasar dan perkembangan dunia *game* pun semakin pesat.

Pada tahun 1983 dunia *video game* kembali ambruk. *Game* yang kurang kreatif membuat *console* kembali mendapat sambutan dingin, apalagi *Personal Computer* saat itu semakin canggih. Masyarakat lebih memilih membeli PC, karena selain untuk bermain PC juga produktif untuk bekerja.

3. Game Generasi Ketiga

Pada tahun 1983, perusahaan bernama *Famicom* (Jepang) menciptakan gebrakan baru, sebuah *console* bernama *Nintendo Entertainment System* (NES) diliris di akhir 1983. *Console* ini menampilkan gambar dan animasi resolusi tinggi untuk pertama kalinya. Setelah mendapat sambutan hangat di Jepang. *Famicom* memperluas pemasarannya ke Amerika, yang dikenal dengan NES. *Nintendo* memiliki chip pengaman pada *cartridge game* mereka, dengan demikian seluruh *game* yang dirilis haruslah sejin *Nintendo*. kemudian, muncul sebuah *game* yang sangat legendaris, yaitu *Super Mario Brothers* (Mario Bros), yang mana *game* ini masih tetap eksis hingga sekarang. Akhirnya, *Nintendo* dari *Famicom* berhasil merajai pasar *video game* di era generasi ketiga.

4. Game Generasi Keempat

Pada tahun 1988, NES mendapat sambutan hangat di seluruh dunia. Sampai akhirnya, sebuah perusahaan bernama *Sega* mencoba menyaingi produk *Nintendo*, *Sega* merilis *console next-generation* mereka, yaitu *Sega Mega Drive* (yang juga dikenal dengan *Sega Genesis*). *Console* ini menyajikan gambar yang lebih tajam dan animasi yang lebih halus dibandingkan NES. *Console* ini cukup berhasil memberi tekanan, tetapi NES tetap bertahan dengan angka penjualan tinggi.

Selanjutnya Pada tahun 1990, *Nintendo* kembali menggebrak dengan *console next-generation* mereka dengan meluncurkan produk SNES (*Super Nintendo Entertainment System*). Selama empat tahun, *Nintendo* dan *Sega* menjadi musuh bebuyutan, meskipun ada beberapa produsen seperti *SNK* dengan *NeoGeo*-nya, *NEC* dengan *TurboGrafx-16* dan *Phillips CD-i*.

Rivalitas yang legendaris, *Super NES* dengan *Mario Bros* sebagai ikonnya melawan *SEGA Mega Drive* dengan *Sonic The Hedgehog* sebagai ikonnya.

5. Game Generasi Kelima

Pada tahun 1990-1994, *Sega* dan *Nintendo* tetap bersaing. Berbagai *game* fenomenal diliris. SNES menyertakan *chip Super FX* pada *cartridge* mereka, dan *Sega* menggunakan *Sega Virtual Processor*. Keduanya bertujuan untuk meningkatkan kualitas grafis dari *game*. Alhasil, SNES dan *Sega* saling beradu dengan *game*-nya seperti *Donky Kong Country* (SNES) dan *Vectorman* (*Sega*).

Tetapi, pada 1993 sebuah perusahaan bernama *Panasonic* meliris *consolenya* yaitu *Panasonic 3DO*. Ini adalah *console* pertama yang menggunakan *CD* sebagai pengganti *cartridge*. Harganya yang sangat mahal membuat *console* ini tidak populer. *3DO* tidak bertahan lama dan harus segera menghentikan produksinya. *3DO* adalah *console* pertama yang menggunakan media *CD*.

Akhirnya, Pada tahun 1994 *Atari* kembali meluncurkan *console* barunya untuk menandingi *Nintendo* dan *Sega*. *Atari Jaguar* jelas jauh lebih canggih ketimbang NES maupun *Mega Drive*, tetapi penggunaannya yang sulit menjadi batu sandungan. Pada tahun yang sama *Sony* meliris *console* super canggih, yaitu *PlayStation*. *Atari* pun bangkrut dan akhirnya melakukan *merger*. *Console* basis *CD* yang pertama kali menuai sukses adalah *Sony PlayStation*. *Console* Jepang ini segera mendapat sambutan hangat hingga saat ini.

6. Game Generasi Keenam

Pada tahun 1998 setelah jatuhnya *Nintendo* dan *Sega*, kini dunia *console* jadi milik *Sony*. *PlayStation* menjadi raja dan bisa dibilang tidak memiliki pesaing. *Sega* mencoba meluncurkan *Sega Dreamcast* untuk mematahkan dominasi *Sony PlayStation*, tetapi kembali gagal dan akhirnya tahun itu juga *Sega* mengundurkan diri dari dunia produsen *console*.

Di tahun 2000, *Sony* semakin merajalela ketika mereka berhasil merilis *console* barunya, bernama *PlayStation2* yang telah berbasis *DVD*. *Nintendo* mencoba bertahan di dunia *console* dengan meliris *GameCube*. *Console* ini tidak menggunakan *DVD* 12cm biasa, melainkan *DVD* yang berukuran lebih kecil, yaitu 8cm. Ukuran medianya yang tidak umum membuat *GameCube* kurang populer. Satu-satunya pesaing serius *PlayStation2* adalah *Xbox*. Sebuah *console* keluaran *Microsoft* yang menggebrak dengan tampilan *visual* yang sangat tajam dan berkualitas. Sayangnya *game Xbox* tidak sepopuler *PlayStation2*.

7. Game Generasi Ketujuh

Tahun 2005, *Xbox* terlambat meluncur ke pasaran dibandingkan *PlayStation2*, dan support *game* tenar juga sangat minim. Tetapi, *Microsoft* seolah belajar dari kesalahan. Pada saat *Sony* masih melakukan riset untuk *console PlayStation3* yang menggunakan *Blu-Ray*, *Microsoft* kali ini telah mengambil langkah inisiatif dengan mengeluarkan *Xbox 360*, sebuah *console* generasi terkini yang memanfaatkan media *HD-DVD*.

Xbox 360 hadir dengan segudang fitur istimewa, mulai dari grafis hingga kecerdasan yang terdapat dalam permainan (*Artificial Intelligence*). Yang akhirnya memperoleh *Best Game of The Years 2006*. Kali ini giliran *Sony* yang terlambat. *PlayStation3* dirilis pada November 2006, selang seminggu sebelum *Nintendo* meluncurkan terobosannya, yaitu *Nintendo Wii*. Posisi *PlayStation3* kurang menguntungkan. Selain harganya mahal, *console* ini juga harus didukung dengan kualitas *device* yang super canggih.

8. Game Generasi *Handheld*

Merebaknya popularitas *game* membuat berbagai perusahaan elektronik berusaha membuat terobosan baru. Diantaranya adalah membuat sebuah mesin *game* berukuran kecil, yang bisa dibawa ke mana pun. Sekitar tahun 1976-1979, sejarah *video game* saku ini bermula, beberapa piranti dari *Mattel* dirilis ke pasaran, tetapi tidak begitu populer. Demikian dengan *handheld* buatan *Milton Bradley* yang di lempar ke pasaran.

Pada tahun 1980-1984 perusahaan-perusahaan Jepang mulai merambah pasar *handheld*, tetapi tetap saja hasilnya. Berlanjut hingga 1984, *Game Boy* pun muncul. *Handheld* buatan *Nintendo* ini begitu diminati dan dinobatkan sebagai *handheld* pertama di dunia yang angka penjualan yang dapat dikatakan sukses.

Pada tahun 1989, *Atari* mengakhiri era *handheld* hitam putih. Produk andalannya adalah *Atari Lynx* yang membawa dimensi baru. Ini adalah *handheld* pertama yang mampu menampilkan warna, sekaligus animasi 3 dimensi (*3D*) yang sederhana. Pada awal 1990, dunia *handheld* semakin menggila, NEC, perusahaan elektronik terkemuka di Jepang membuat *handheld* yang mampu membuat animasi 3D lebih kompleks, Karena menerapkan konsep grafis 3D untuk PC.

Kemudian pada tahun 1998-2000, *Sony* merilis *Pocket Station* dan memberikan kejutan besar di dunia *console. Handheld* ini memiliki kualitas visual yang jauh lebih baik dibandingkan *handheld* lain yang ada di pasaran. Salah satunya adalah *Nintendo* dan *Gumpei Yokoi*, memutuskan untuk keluar dan bergabung dengan *Bandai*, kemudian merilis *WonderSwan* dan *WonderSwan Color*.

Akhirnya pada tahun 2004-2006, *Sony* merilis *handheld* pertama yang menggunakan cakram bernama PSP (*PlayStation Portable*) dan dibarengi dengan hadirnya *Nintendo DS*, yang menggunakan konsep *dual screen*. Lalu, disusul oleh *game Boy Micro* dan *Game Park XGP. Nintendo DS lite* dan *Pelican VG Pocket Caplet* menjadi *handheld* terbaru yang dilempar ke pasaran.

2.3. Syarat Perancangan Game

Berikut ini adalah syarat utama dalam merancang sebuah *game*, yaitu :

1. *Game are not linear*

Pada umumnya di dalam sebuah *game* terdapat tingkat kesulitan yang berbeda. Tetapi dalam menyelesaikan setiap kasus di dalam *game*, pemain tidak harus menyelesaikannya secara *linear* atau berurutan. Melainkan pemain dapat menyelesaikannya dengan mendahulukan cara yang dianggap mudah. Berdasarkan alur tersebut, maka *game* disajikan secara tidak *linear* (lurus).

2. *Game have a goal*

Di dalam merancang sebuah *game*, *game* harus memiliki tujuan atau hasil akhir permainan. Seperti halnya pada permainan catur, pemain harus dapat mengalahkan lawan dengan cara "*checkmate*". Atau seperti pada *game RPG*, pemain harus bertemu dengan salah seorang tokoh di *game* tersebut untuk memperoleh *item* yang berguna pada perjalanan selanjutnya.

3. *Game must be winnable*

Di dalam merancang sebuah *game*, *game* tersebut juga harus memiliki cara untuk dapat dimenangkan oleh pemain. Jadi, di dalam permainan tersebut antara pemain dengan musuh atau antara pemain dengan pemain memiliki kesempatan menang yang sama besar.

4. *Start, middle and ending part of the game*

Di dalam sebuah *game*, *game* wajib memiliki posisi awal (*starting point*) yang mana pada posisi ini pemain akan mengawali permainannya. Sedangkan untuk *middle*, merupakan alur perjalanan atau proses yang terjadi selama permainan sehingga sampai pada tujuan permainan atau akhir dari permainan tersebut (*ending*).

2.4. Jenis Game

Berikut ini adalah beberapa jenis *game* yang ada hingga saat ini :

1. Action Game

Action game merupakan jenis *game* yang menampilkan sebuah medan pertempuran atau sebuah kasus yang harus diselesaikan dengan cara bertempur dengan dilengkapi persenjataan yang komplit, disertai perpindahan tempat, memiliki tensi yang tinggi, dan umumnya diambil dari sudut pandang orang pertama. Contoh : *Sierra*, *Counter Strike*, *Xiao-Xiao*, *Resident Evil*, *Batman*, dan sebagainya.

2. Adventure Game

Adventure game adalah *game* berjenis petualangan dengan alur cerita yang berkesinambungan disertai perpindahan atau pergantian tempat dari satu wilayah ke wilayah yang lain dan biasanya disertai dengan teka-teki untuk mengakhiri sebuah misi. Contoh : *Super Mario Bros*, *Tiny Toon*, *TMNT*, *Donkey Kong*, dan sebagainya.

3. Casual Game

Casual game merupakan sebuah *game* hasil implementasi dari permainan tradisional maupun permainan sehari-hari dan biasanya menggunakan papan sebagai alasnya. Contoh : *Sudoku Café*, *Chessmaster 9000*, *Solitaire*, *Poker Texas*, *Shanghai Mahjong*, dan sebagainya.

4. Educational Game

Educational game merupakan *game* dengan menggunakan media pembelajaran interaktif sebagai intinya. Yang mana *game* ini ditujukan untuk memberikan pembelajaran, berkaitan dengan penerapan ilmu pengetahuan, ataupun memberi stimulus pada kinerja saraf dan otak pemain. Contoh : *Barbie Secret Agent*, *Blue's Clues Learning Time*, *Zuma*, *Angry Bird*, dan sebagainya.

5. Role Playing Game

Role Playing Game (RPG) merupakan jenis *game* yang memerankan sebuah tokoh (*avatar*) untuk menjalani sebuah cerita layaknya kehidupan sehari-hari yang nyata. *Game* ini tidak mampu diselesaikan dalam waktu satu hari atau sekali main, karena umumnya *game* jenis ini berkaitan dengan *level avatar*, menjalankan misi yang berbeda pada setiap tempat, arena permainan yang luas, dan meningkatnya level setiap musuh. Pada saat ini *game* RPG umumnya disajikan dengan grafis yang menarik, adanya transaksi jual-beli antara pemain dengan pelaku *game* lainnya, aturan pertarungan dilakukan dengan *turn-based*, dan adanya aksesoris untuk menambah kinerja *avatar*. Contoh : *GTA, Ragnarok, Pokemon, Zoid, Dragon Nest*, dan sebagainya.

6. Simulation Game

Simulation game adalah jenis *game* yang mengkondisikan pemain untuk menirukan kegiatan layaknya di dunia nyata secara alami. *Game* jenis ini mengacu dan bersumber sesuai kenyataan dan dapat diterapkan pula di dunia nyata. Contoh : *Grand Turismo, Playboy, Flight Simulator 2002, The Reconstruction Initiative*, dan sebagainya.

7. Sport Game

Sport game adalah jenis *game* dengan mengadopsi pada kegiatan olahraga di kehidupan nyata serta dilengkapi dengan aturan mainnya. *Game* jenis ini merupakan *game* yang melibatkan banyak kecerdasan. Selain kecerdasan yang diterapkan pada musuh, kecerdasan lainnya juga diterapkan pada wasit, juri, dan peraturan pertandingan (penentuan pelanggaran). Contoh : *Pro Evolution Soccer, Tony Hawk's Pro Skater, Street Ball, WWE*, dan sebagainya.

8. Strategy Game

Strategy game merupakan jenis *game* yang umumnya identik dengan situasi peperangan, biasanya menampilkan pemandangan dari atas (*angle from top*), arena tempur digambarkan seperti peta yang bersegmen, permainan dimulai dengan membangun sendiri markas dan perlengkapan pemain. *Game* ini tidak memainkan tensi yang tinggi, tapi menuntut pemain untuk berfikir logis, cepat dan tepat dalam memenangkan pertempuran. Biasanya, sudut pandang pemain adalah menjadi orang ketiga. Contoh : *Red Alert, Vietnam Squad Battles, Warcraft 3: Reign of Chaos, Soccer Manager*, dan sebagainya.

2.5. Kecerdasan Buatan

Kecerdasan buatan didefinisikan sebagai kecerdasan yang ditujukan untuk suatu entitas buatan. Sebuah kecerdasan yang diciptakan dan dimasukkan kedalam mesin atau objek komputer agar dapat melakukan tindakan alami selayaknya manusia. (Suyanto, 2011).

Kecerdasan buatan kini banyak digunakan dalam permainan komputer, dimana kecerdasan buatan berperan aktif dalam memberikan aksi dan reaksi kepada pemain. Untuk pembentukan kecerdasan buatan diperlukan sebuah algoritma atau metode sebagai strukturnya. Berdasarkan algoritma inilah sebuah *game* dapat disusun kecerdasan buataannya.

2.6. Metode Kecerdasan Buatan

2.6.1 Searching

1. Breadth First Search

Breadth First Search merupakan suatu metode pencarian dengan melakukan pencarian pada semua simpul dalam setiap *level* secara berurutan dari kiri ke kanan. Jika pada satu *level* belum ditemukan solusi, maka pencarian dilanjutkan pada *level* berikutnya.

2. Depth First Search

Depth First Search merupakan suatu metode pencarian yang dilakukan pada suatu simpul dalam setiap *level* mulai dari yang paling kiri. Jika pada *level* yang terdalam belum ditemukan solusi, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori.

3. Depth Limited Search

Depth Limited Search adalah suatu metode solusi yang membantu kelemahan *Depth First Search*, dengan membatasi kedalaman maksimum dari suatu jalur solusi. Tetapi, sebelum menerapkan metode ini, *programmer* harus mengetahui berapa *level* maksimum dari suatu solusi. Jika, batas kedalaman terlalu kecil, maka metode *Depth Limited Search* tidak dapat menemukan solusinya.

4. Uniform Cost Search

Uniform Cost Search adalah metode yang hampir sama dengan *Breadth First Search*, hanya saja pada *Uniform Cost Search* yang digunakan adalah urutan biaya dari paling kecil sampai dengan yang terbesar. *Uniform Cost Search* berusaha menemukan solusi dengan total biaya terendah yang dihitung berdasarkan biaya dari simpul asal ke simpul tujuan.

5. Iterative Deepening Search

Iterative Deepening Search adalah metode yang menggabungkan kelebihan *Breadth First Search* dengan kelebihan *Depth First Search*, tetapi konsekuensinya adalah *time complexity* yang lebih tinggi.

6. Bi-Directional Search

Bi-Directional Search adalah metode pencarian dalam menemukan solusi dengan cara menggabungkan dua alur pencarian, yaitu alur mundur (dari *goal* ke *start*) dengan alur maju (dari *start* ke *goal*).

7. Generate and Test

Generate and Test adalah metode yang menggunakan prosedur *Depth First Search*, karena suatu solusi harus dibangkitkan secara lengkap sebelum dilakukan *test*. Metode ini berbentuk sistematis, pencarian sederhana yang mendalam dari suatu ruang permasalahan. *Generate and Test* juga dapat dilakukan dengan pembangkitkan solusi secara acak, tetapi tidak akan menjamin akan ditemukannya solusinya.

8. Hill Climbing

Hill Climbing merupakan metode yang berkebalikan dengan *Generate and Test*. Metode ini bertumpu dengan *test* untuk membantu pembangkit menemukan solusi yang bisa langsung dihilangkan dalam ruang pencarian. *Hill Climbing* sering digunakan jika terdapat suatu fungsi heuristik yang baik untuk mengevaluasi *state*. Sebagai contoh, ketika seseorang berada di sebuah kota yang tidak terkenal tanpa peta dan orang tersebut ingin menuju ke pusat kota. Cara yang paling sederhana adalah dengan berpatokan dengan gedung-gedung tinggi yang umumnya terletak di pusat kota. Jadi, fungsi heuristik ini adalah menemukan jarak antara orang tersebut dengan gedung-gedung tinggi.

9. Simulated Annealing

Simulated Annealing adalah metode pencarian dengan memanfaatkan analogi cara pendinginan dan pembekuan metal menjadi sebuah struktur kristal dengan energi yang maksimal. *Simulated Annealing* menerapkan sebuah rumus probabilitas yang memungkinkannya dapat keluar dari *local minimum*. Fungsi tersebut diadopsi dari ilmu fisika, yang mana fungsi ini mempresentasikan distribusi *Boltzman* dari energi dalam sistem termodinamik. Sehingga, diperoleh probabilitas dari *level* energi yang diberikan dalam sebuah sistem temperatur.

10. Best First Search

Best First Search merupakan metode pencarian dengan cara memilih simpul baru yang memiliki biaya terkecil diantara semua *node*. *Best First Search* akan membangkitkan simpul berikutnya dari sebuah simpul yang terbaik diantara semua *node* yang pernah dibangkitkan. Penentuan simpul terbaik diperoleh melalui biaya perkiraan dari sebuah simpul hingga menuju *goal* atau dengan menggabungkan biaya perkiraan dengan biaya sebenarnya. Biaya perkiraan ini dapat diperoleh dengan menggunakan fungsi heuristik.

11. Greedy Best - First Search

Greedy Search merupakan salah satu metode *Best First Search* yang lebih sederhana. Yaitu, dengan memperhitungkan biaya perkiraannya saja tanpa menghitung biaya sebenarnya. Sehingga, pencorepan metode ini menjadi kurang optimal.

12. A*

*A** merupakan penggabungan dari metode *Uniform Cost Search* dengan metode *Greedy Best - First Search*. Biaya yang diperhitungkan didapat dari biaya sebenarnya ditambah biaya perkiraan.

13. Iterative Deepening A*

*Iterative Deepening A** merupakan metode pencarian dengan menerapkan batasan berupa *f-limit* (nilai gabungan antara biaya perkiraan dengan biaya sebenarnya). Pada metode ini, setiap iterasi akan mengembalikan nilai *f-limit* baru yang akan digunakan sebagai batasan pencarian untuk iterasi selanjutnya. Karena dilakukan secara iteratif, maka dalam penerapannya membutuhkan lama dengan *time complexity* yang tinggi.

Keunggulan dari metode *Iterative Deepening A** ini terletak pada penggunaan memori menjadi jauh lebih sedikit.

14. Simplified Memory Bounded A*

*Simplified Memory Bounded A** merupakan metode pencarian yang mampu mengingat semua biaya dari setiap iterasi sampai sejumlah simpul yang ada di dalam memori. Karena batasan memori dalam jumlah tertentu, sehingga *programmer* dapat membatasi pencarian hanya sampai pada simpul-simpul yang dapat dicapai dari *root* sepanjang suatu jalur yang memorinya masih mencukupi. Kemudian, mengembalikan suatu rute terbaik diantara rute-rute yang ada dalam batasan jumlah simpul tersebut.

15. Bi-Directional A*

*Bi-Directional A** adalah metode pencarian heuristik yang pada prinsipnya dapat memanfaatkan algoritma *A**. hanya saja pada metode *Bi-Directional A** dapat dikerjakan lebih cepat dan lebih menghemat dalam penggunaan memori, karena simpul yang dibangkitkan *Bi-Directional A** lebih sedikit. Jumlah simpul yang dibangkitkan oleh *Bi-Directional A** diperkirakan setengah dari jumlah simpul yang dibangkitkan *A**.

16. Modified Bi-Directional A*

*Modified Bi-Directional A** merupakan metode pencarian yang menerapkan fungsi heuristik dengan sedikit modifikasi pada pencarian maju (dari *start* ke *goal*).

17. Dynamic Weighting A*

*Dynamic Weighting A** adalah metode yang memberikan bobot dinamis terhadap suatu fungsi heuristik. Dengan pembobotan yang dinamis ini, *programmer* dapat mengasumsikan bahwa pada awal iterasi pencarian simpul dapat diarahkan kemana saja. Tetapi, ketika *goal* sudah dekat maka pencarian difokuskan ke arah *goal*.

2.6.2 Reasoning

1. Propositional Logic

Propositional Logic adalah sebuah metode logika penalaran sebagai pemecahan solusi yang menerapkan fakta benar, salah atau tidak diketahui. Pada metode ini, suatu simbol dinyatakan sebagai suatu fakta yang dapat bernilai benar atau salah atau tidak tahu. simbol proposisi ini dapat dihubungkan dengan *Boolean connectives* sehingga membentuk suatu kalimat. Tetapi dalam permasalahan yang kompleks metode ini tidak

sepenuhnya dapat diterapkan, karena *programmer* harus banyak membangun banyak fakta untuk mempresentasikan keadaan yang sederhana.

2. Predicate Logic

Predicate Logic merupakan metode penalaran dengan mempresentasikan fakta dan aturan kedalam suatu objek, relasi (*predicate*), *connectives* dan *quantifier* sehingga beberapa fakta sederhana dapat direpresentasikan ke dalam suatu kalimat logika. Pada *Predicate Logic* semua *predicate* bersifat tetap pada waktu kapanpun, sehingga pada masalah yang kompleks dimana suatu relasi dapat berubah sesuai dengan perubahan waktu.

3. Fuzzy Logic

Fuzzy Logic adalah metode logika penalaran metode untuk menangani masalah ketidakpastian. Yang dimaksud dengan ketidakpastian yaitu suatu masalah yang mengandung keraguan, ketidaktepatan, kurang lengkapnya informasi, dan nilai kebenarannya bersifat sebagian yang menerapkan nilai tingkat kebenaran antara 0 dan 1.

4. Finite State Machine

Finite State Machine merupakan salah satu logika penalaran yang memperlihatkan perilaku sistem dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi.

2.6.3 Planning

1. Goal Stack Planning

Goal Stack Planning adalah metode *planning* dengan mengumpulkan sebuah tumpukan (*stack*) untuk menampung keadaan-keadaan (keadaan *goal* dan keadaan yang mungkin terjadi selama proses pencarian solusi) dan operator-operator yang telah diajukan untuk memenuhi keadaan-keadaan tersebut.

2. Constraint Posting

Constraint Posting adalah metode *planning* yang hanya menggunakan satu tumpukan saja untuk menuju *goal*. Metode ini bekerja berdasarkan keyakinan bahwa, pada kebanyakan masalah, sub-sub masalah perlu untuk dikerjakan bersamaan. *Constraint Posting* melakukan penyelesaian melalui tiga tahap, yaitu menganalisa semua operator secara bertahap, mengurutkan operator secara parsial dan membuat variabel antar operator.

2.6.4 Learning

1. Neural Networks

Neural networks merupakan bentuk penerapan kemampuan syaraf otak manusia untuk mengenali fakta atau objek ke dalam teknik komputerisasi. *Neural networks* menyerupai otak manusia dalam dua hal, yaitu pengetahuan diperoleh jaringan melalui proses pembelajaran dan kekuatan hubungan antar sel syaraf yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan. *Neural networks* memiliki jaringan yang tersebar paralel dalam jumlah besar sehingga, memiliki kemampuan untuk menyimpan pengetahuan yang bersifat pengalaman.

2. Genetic Algorithms

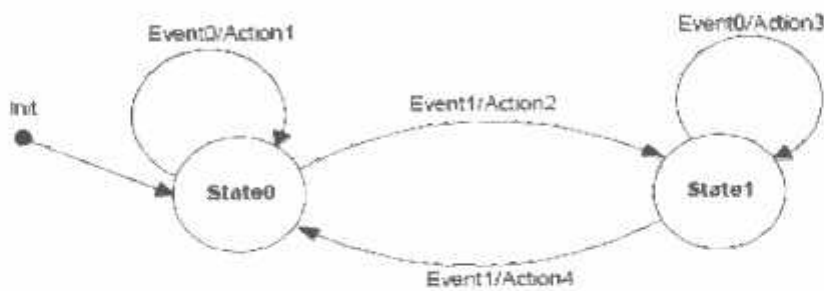
Genetic algorithms adalah metode kecerdasan yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Pada awalnya *Genetic algorithms* digunakan sebagai metode pencarian dengan *parameter* yang optimal. Tetapi, sesuai perkembangannya *Genetic algorithms* dapat diterapkan sebagai metode *learning*, peramalan dan pemrograman otomatis.

3. Decision Tree Learning

Decision tree learning adalah metode kecerdasan secara pembelajaran yang berusaha menemukan fungsi-fungsi yang bernilai *diskrit* dan tahan terhadap data-data yang terdapat kesalahan (*noisy data*) serta mampu mempelajari fungsi-fungsi *disjunctive*.

2.7. Finite State Machine

Finite State Machine merupakan metodologi kecerdasan buatan yang memperlihatkan perilaku sistem dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks. (Setiawan, 2006).


















Gambar. 2.1 Contoh alur diagram Finite State Machine sederhana



Pada gambar. 2.1, diagram tersebut memperlihatkan FSM dengan 2 buah *state* dan 2 buah *input* serta 4 buah aksi *output* yang berbeda : ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state.0*, pada keadaan ini sistem akan menghasilkan *Action.1* jika terjadi masukan *Event.0*, sedangkan jika terjadi *Event.1* maka *Action.2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State.1* dan seterusnya.

2.8. Diagram Alir

Diagram Alir adalah diagram yang menggambarkan bagaimana jalannya program maupun sistem kerja mulai dari awal hingga akhir. Setiap diagram alir harus memiliki titik awal dan titik akhir (*start and stop*). Diagram alir dibentuk dengan memanfaatkan simbol-simbol tertentu. Perancangan sebuah diagram alir umumnya sebagai bahan mentah sebelum proses *coding* sesungguhnya. Pada tabel 2.1 terlihat simbol-simbol yang digunakan pada diagram alir.

Tabel 2.1. Simbol Diagram Alir

Simbol	Nama	keterangan
	Dokumen	Digunakan untuk semua jenis dokumen.
	Dokumen rangkap	Menggambarkan dokumen asli beserta tembusannya.
	Berbagai dokumen	Menggambarkan berbagai jenis dokumen yang digabungkan bersama dalam satu paket.
	<i>Input</i> atau <i>output</i> data	Proses <i>input output</i> data, parameter dan informasi.
	Penghubung	Penghubung bagian-bagian diagram alir yang berada pada satu halaman.
	Kegiatan manual	Untuk menggambarkan suatu kegiatan manual.
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen sementara.
	Arsip permanen	Menunjukkan tempat penyimpanan dokumen secara permanen yang tidak akan diproses lagi.
	Proses	Proses perhitungan dan proses pengolahan data.
	<i>Keying</i> atau <i>typing</i>	Menggambarkan <i>input</i> data ke dalam komputer melalui <i>on-line terminal</i> .
	Pita magnetic	Menggambarkan arsip pada komputer yang berbentuk pita magnetik.
	<i>On-line storage</i>	Menggambarkan arsip pada komputer yang berbentuk <i>on-line</i>
	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data atau operasi perbandingan logika.
	Garis alir	Menunjukkan arah aliran proses pengolahan data atau sistem.
	Persimpangan garis alir	Jika dua garis alir bersimpangan, untuk menunjukkan arah masing-masing garis, salah satu garis dibuat melengkung.

	Pertemuan garis alir	Digunakan jika terdapat dua garis alir bertemu dan salah satu garis mengikuti garis lainnya.
	Mulai atau berakhir	Menggambarkan awal dan akhir suatu sistem.

2.9. Perangkat Lunak Pendukung

2.9.1 Adobe Flash CS3

Awalnya *Macromedia Flash* pertama kali di produksi pada tahun 1966. pada awal produksi, *Macromedia Flash* merupakan *software* untuk membuat animasi sederhana berbasis *GIF*. Seiring dengan perkembangannya, *Macromedia Flash* mulai digunakan untuk pembuatan desain situs *web*. *Macromedia Flash* telah diproduksi dengan beberapa versi hingga *Macromedia Flash* versi 8. Namun sejak diakuisisi oleh perusahaan raksasa *Adobe*, maka *software* multimedia ini berubah nama menjadi *Adobe Flash*. Kemudian muncullah versi barunya, yaitu *Adobe Flash CS3* atau dapat juga disebut sebagai *Flash9*.

Adobe Flash CS3 merupakan *software* yang memiliki kemampuan menggambar sekaligus menganimasikan sebuah objek sehingga mudah dipelajari. Selain digunakan untuk keperluan pembuatan animasi biasa dan animasi situs *web*, *Flash* juga dapat digunakan untuk membuat *game*, presentasi, dan animasi kartun. Sekarang juga sudah mulai berkembang penggunaan *Flash* untuk pembuatan *game* pada *mobile device* seperti *handpone*, *PDA*, dan lain-lain. (Amrullah, 2008).

Animasi yang dihasilkan *Flash* adalah animasi berupa *file movie swf*. *Movie* yang dihasilkan dapat berupa grafis atau teks. Grafis yang dimaksud disini adalah grafis yang berbasis *vector*. Jadi ketika diakses melalui internet, animasi yang akan ditampilkan lebih cepat dan terlihat halus. Selain itu, *Flash* juga memiliki kemampuan untuk mengimpor *file audio*, *video* maupun *file* gambar dari aplikasi lain. Seperti pada gambar. 2.2 dimana terlihat tampilan *workspace* pada *Adobe Flash CS3*.



Gambar. 2.2 Tampilan Workspace Adobe Flash CS3

Adobe Flash CS3 memiliki *tools* yang dapat digunakan untuk membuat sebuah proyek. *Tools* yang terdapat pada *Adobe Flash CS3* dapat dilihat pada tabel 2.2.

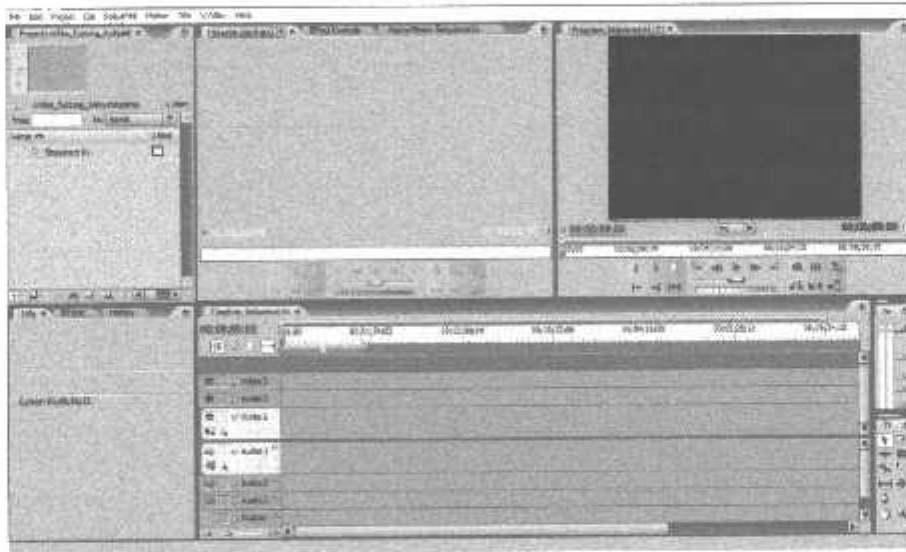
Tabel 2.2. *Tools* pada *Adobe Flash CS3*

Alat	Fungsi
<i>Arrow Tool</i>	Untuk memilih suatu objek.
<i>Sub Select Tool</i>	Untuk mengolah objek garis.
<i>Free Transform Tool</i>	Untuk mengubah ukuran suatu objek dengan fleksibel.
<i>Gradient Transform Tool</i>	Untuk mengatur gradiasi objek.
<i>Line Tool</i>	Untuk membuat objek garis.
<i>Text Tool</i>	Untuk membuat objek teks.
<i>Oval Tool</i>	Untuk membuat objek lingkaran.
<i>Rectangle Tool</i>	Untuk membuat objek persegi.
<i>Pencil Tool</i>	Untuk membuat objek bebas dengan garis.
<i>Ink Bottle</i>	Untuk memberi warna pada garis luar objek.
<i>Paint Bucket Tool</i>	Untuk memberi warna di dalam objek.

2.9.2 Adobe Premiere Pro CS3

Adobe Premiere Pro CS3 adalah *software* yang digunakan untuk mengedit dan mendesain suatu *video*. Pada umumnya *Adobe Premiere Pro CS3* memang sering digunakan untuk mengedit suatu *movie*, tetapi perangkat lunak ini juga dapat digunakan untuk mengedit file *audio* yang umumnya berformat *WAV* maupun *MP3*.

Adobe Premiere Pro CS3 merupakan salah satu *software* pengolah *video* yang sangat populer dan diakui kecanggihannya. Kelengkapan fasilitas dan kemampuannya dalam mengolah *video*, menjadikan *software* ini banyak digunakan oleh *production house*. Seperti pada gambar. 2.3 terlihat tampilan *workspace* pada *Adobe Premiere Pro CS3*.



Gambar. 2.3 Tampilan Workspace Adobe Premiere Pro CS3

BAB III

ANALISA DAN PERANCANGAN

3.1. Analisa

Analisa merupakan tahapan awal didalam membangun sebuah *game*. Analisa adalah tahapan sebelum kepada perancangan. Dimana pada tahap ini, penulis menguraikan pokok permasalahan dan mengumpulkan informasi yang dibutuhkan untuk ke tahap perancangan.

3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional yang digunakan dalam membangun *game* ini meliputi aksi yang dimiliki oleh karakter Hantu Anak Buah dan Hantu Boss. Kebutuhan fungsional Hantu Anak Buah dalam *game* Malam Jum'at Kliwon ini, antara lain :

1. Terdapat dua macam aksi awal Hantu Anak Buah dalam mencari pemain, yaitu ada yang berjalan secara *vertical* dan ada yang berjalan *horizontal*.
2. Jika Hantu Anak Buah belum melihat pemain, maka Hantu Anak Buah akan tetap pada aksi mencari pemain.
3. Jika Hantu Anak Buah melihat posisi pemain, maka Hantu Anak Buah akan melakukan aksi mengejar pemain.
4. Ketika pemain melakukan aksi menembak. Jika Hantu Anak Buah melihat laju peluru pemain, maka Hantu Anak Buah dapat menghindarinya. Namun, jika Hantu Anak Buah tidak melihat laju tembakan tersebut maka ia akan mati.
5. Jika dalam pengejaran, pemain berhasil lolos dari pandangan Hantu Anak Buah. Maka, Hantu Anak Buah akan kembali mencari pemain.
6. Pada *game* ini, Hantu Anak Buah dalam satu area berjumlah lebih dari 1. Dengan konsep jika terkena tembakan pemain, maka langsung mati.
7. Jika jumlah Hantu Anak Buah dalam satu area sudah mati semua, maka akan muncul animasi arah panah untuk pemain melanjutkan ke tempat selanjutnya.

Sedangkan Kebutuhan fungsional Hantu Boss dalam *game* Malam Jum'at Kliwon, meliputi :

1. Aksi awal dalam mencari pemain, yaitu dengan berjalan secara *horizontal*.
2. Untuk Hantu Boss terdapat *healthbar* sebagai indikator nyawa.
3. Jika Hantu Boss belum melihat pemain, maka Hantu Boss tetap pada aksi mencari pemain.
4. Jika Hantu Boss melihat posisi pemain, maka Hantu Boss akan melakukan aksi mengejar dan menembak pemain.
5. Ketika pemain melakukan aksi menembak. Jika Hantu Boss melihat laju peluru pemain, maka Hantu Boss akan menghindarinya. Namun, jika Hantu Boss tidak melihat laju scrangan tersebut maka kesehatan pada *healthbar* Hantu Boss akan berkurang.
6. Jika dalam pengejaran, pemain berhasil lolos dari pandangan Hantu Boss. Maka, Hantu Boss akan kembali mencari pemain.
7. Jika kesehatan pada *healthbar* Hantu Boss = 0, maka Hantu Boss mati dan akan muncul animasi arah panah untuk pemain melanjutkan ke tempat selanjutnya.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional yang digunakan dalam membangun *game* ini meliputi kebutuhan non fungsional *performance* dan kebutuhan non fungsional *interface*. Kebutuhan non fungsional *performance* dalam *game* Malam Jum'at Kliwon ini, antara lain :

1. Dalam pembuatan *game* Malam Jum'at Kliwon menggunakan 35fps (satuan *frame per second*).
 2. Dalam setiap musuh Terdapat 3 sensor, sensor untuk mengenali pemain, sensor untuk mengenali peluru pemain, dan sensor untuk mengenali sasaran (*hit*).
 3. Pergerakan pemain dalam *game* sejauh 25pixel.
 4. Pergerakan laju peluru bawang putih sejauh 25pixel.
 5. Pergerakan laju peluru bawang merah sejauh 50pixel.
 6. Untuk senjata bawang putih, jika mengenai musuh maka akan ikut meledak bersama musuh. Untuk senjata bawang merah, jika mengenai musuh tidak akan ikut meledak bersama musuh. Namun, untuk bawang merah hanya dapat digunakan 1kali di setiap arena.
-

7. Pergerakan Hantu Anak Buah dalam mengejar pemain sejauh *25pixel*.
8. Pergerakan Hantu Anak Buah dalam menghindari serangan pemain sejauh *20pixel*.
9. Pergerakan Hantu Boss dalam mengejar pemain sejauh *30pixel*.
10. Pergerakan Hantu Boss dalam menghindari serangan pemain sejauh *20pixel*.
11. Pergerakan laju peluru Hantu Boss sejauh *40pixel*.
12. Pemain dan Hantu Boss menggunakan *healthbar* sebagai indikator nyawa.
13. Panjang sensor musuh yang digunakan untuk mengenali objek pemain sepanjang *5000pixel*.
14. Panjang sensor musuh yang digunakan untuk mengenali objek peluru pemain sepanjang *3000pixel*.
15. Kesehatan pemain pada *healthbar* akan berkurang apabila bersentuhan dengan setiap hantu ataupun terkena peluru Hantu Boss.
16. Kesehatan Hantu Boss pada *healthbar* akan bertambah apabila bersentuhan dengan pemain.
17. Durasi pada *game* sepanjang 15menit (waktu mundur). Apabila waktu yang ditentukan telah habis dan permainan belum selesai, maka pemain dinyatakan mati (*game over*).

Sedangkan Kebutuhan non fungsional *interface* dalam *game* Malam Jum'at Kliwon ini, antara lain :

1. Spesifikasi *monitor* yang digunakan dalam membangun *game* ini, yaitu sebesar *1280 x 800 pixel*.
 2. Spesifikasi *Processor* yang digunakan dalam membangun *game*, yaitu *AMD Turion X2 (64)*.
 3. Spesifikasi *RAM* yang digunakan dalam membangun *game*, sebesar *1024 MB*.
 4. Spesifikasi *VGA* yang digunakan dalam membangun *game*, yaitu *nVIDIA GeForce 9100M G*.
 5. Sistem operasi yang digunakan dalam membangun *game* adalah *Windows Xp.3 Dark Edition Rebirth Version*.
 6. *Size game* yang digunakan sebesar *1280 x 800 pixel*.
 7. *Game* dimainkan menggunakan *key* panah sebagai kontrol pemain. *Key A* sebagai senjata bawang putih dan *key S* sebagai senjata bawang merah. Sedangkan, *key space* digunakan sebagai *pause*.
-

8. *Platform* yang digunakan dalam membangun *game* adalah *Adobe Flash CS3* dengan bahasa pemrograman *Action Script 2.0*.
9. *Platform* yang digunakan dalam *editing audio* adalah *Adobe Premiere Pro CS3*.
10. *File* hasil *publish* dari *game* Malam Jum'at Kliwon akan dilakukan pengujian ke beberapa *laptop* atau *personal computer* dengan spesifikasi *hardware* dan sistem operasi yang berbeda.

3.2. Perancangan Game

Pada tahap perancangan, penulis menguraikan hasil dari seluruh analisa yang didapat menjadi karya abstrak yang selanjutnya dijadikan komposisi pada tahap implementasi dalam membangun *game* Malam Jum'at Kliwon.

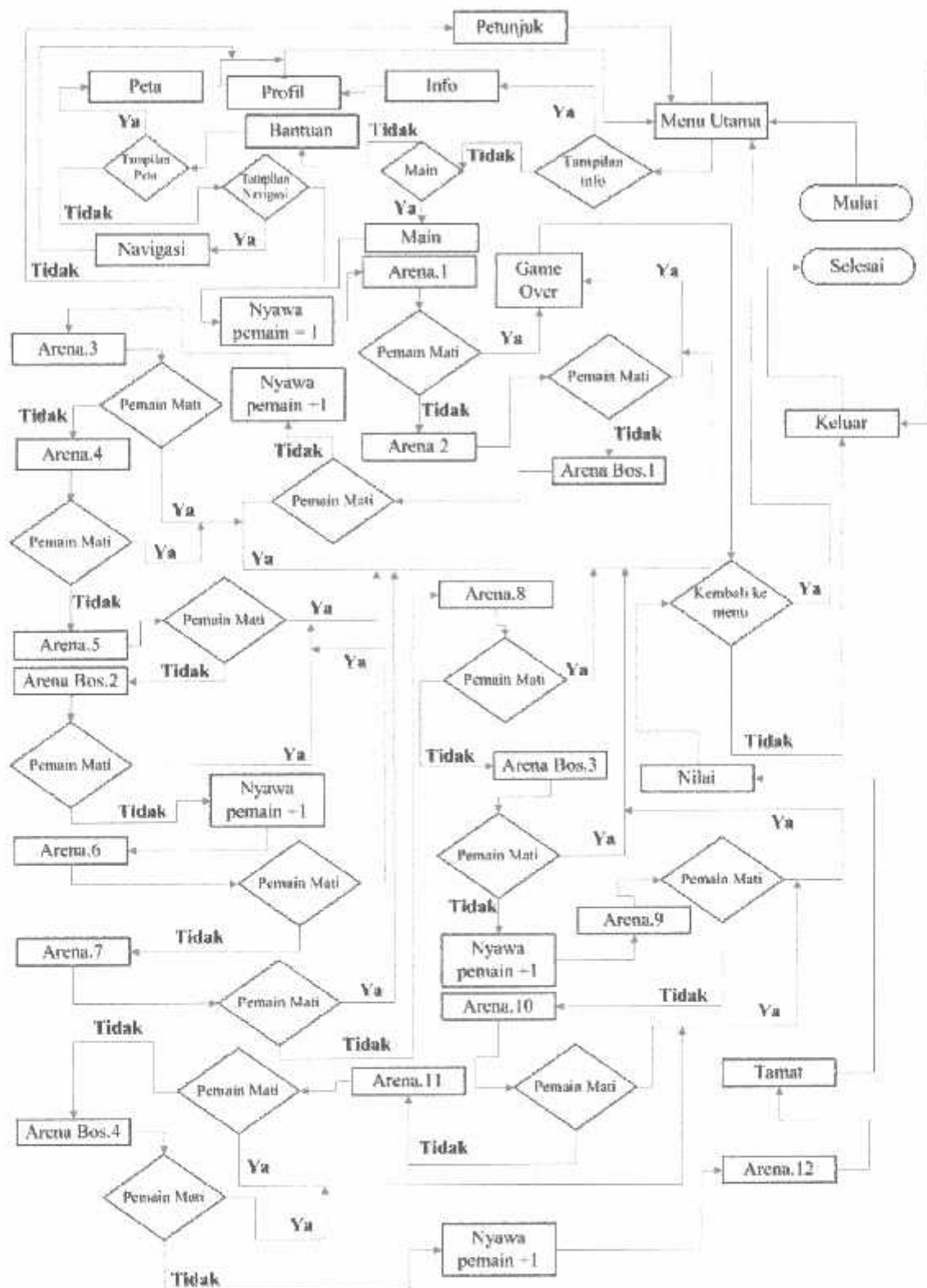
3.2.1 Alur Cerita

Bang Numpung adalah seorang supir bajaj yang ramah, disiplin dan rajin bekerja. Karena di Jakarta telah banyak masyarakat yang mampu membeli kendaraan pribadi, akhirnya penghasilan Bang Numpung setiap hari kian menurun.

Untuk mengejar target "setoran"-nya, maka Bang Numpung tetap mencari penumpang hingga larut malam. Sampai pada suatu hari di malam jum'at kliwon ketika Bang Numpung pulang ke rumahnya, ia harus melewati beberapa tempat angker di Jakarta. Ia pun dikejar oleh hantu-hantu jahil *penghuni* tempat tersebut. Misi utama dari *game* ini adalah membantu Bang Numpung sampai ke kembali rumahnya dengan cara mengalahkan hantu-hantu yang mengejanya.

3.2.2 Alur Permainan

Hal yang pertama kali dilakukan dalam membangun *game* Malam Jum'at Kliwon adalah membuat *flowchart* untuk menentukan alur permainan. Seperti yang terlihat pada gambar 3.1 *flowchart* alur *game* Malam Jum'at Kliwon.



Gambar. 3.1 Flowchart perancangan game

Flowchart alur *game* Malam Jum'at Kliwon dapat dijelaskan sebagai berikut :

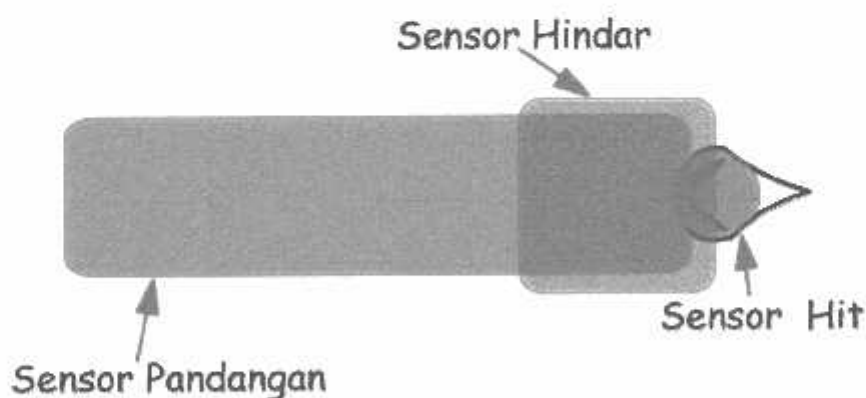
1. Ketika *game* ini dijalankan akan muncul menu utama dengan empat buah *button*, yaitu *button* main, *button* bantuan, *button* info dan *button* keluar.
 2. Ketika *button* bantuan diklik, disana terdapat tiga *button*, yaitu *button* peta untuk memunculkan tampilan peta, *button* navigasi untuk memunculkan tampilan navigasi dan *button* petunjuk untuk memunculkan tampilan petunjuk permainan. Kemudian, dari tampilan ini dapat kembali ke menu utama.
 3. Ketika *button* info diklik, maka akan memunculkan tampilan profil penulis, yaitu foto, nama lengkap, jurusan dan tahun pembuatan *game* Malam Jum'at Kliwon. Kemudian, dari tampilan ini dapat kembali ke menu utama.
 4. Ketika *button* keluar diklik, maka akan keluar dari permainan.
 5. Ketika *button* main diklik, maka pemain akan mengawali *game* mulai dari arena.1 dan pemain memiliki nyawa = 1.
 6. Selama di arena.1, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.2.
 7. Selama di arena.2, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena boss.1.
 8. Selama di arena boss.1, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.3. Pemain mendapat bonus 1 nyawa.
 9. Selama di arena.3, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.4.
-

10. Selama di arena.4, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.5.
 11. Selama di arena.5, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena boss.2.
 12. Selama di arena boss.2, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.6. Pemain mendapat bonus 1 nyawa.
 13. Selama di arena.6, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.7.
 14. Selama di arena.7, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.8.
 15. Selama di arena.8, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena boss.3.
 16. Selama di arena boss.3, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. pemain tidak mati, maka berlanjut ke arena.9. Pemain mendapat bonus 1 nyawa.
 17. Selama di arena.9, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.10.
-

18. Selama di arena.10, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.11.
19. Selama di arena.11, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena boss.4.
20. Selama di arena boss.4, apakah pemain mati? Jika ya maka pemain *game over*. Pada tampilan *game over* terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan. Jika pemain tidak mati, maka berlanjut ke arena.12 (zona aman). Pemain mendapat bonus 1 nyawa.
21. Setelah melewati zona aman, kemudian muncul tampilan ending yang disertai *button* untuk menuju kepada tampilan nilai (*score*). Pada tampilan nilai terdapat pilihan kembali ke menu utama. Jika ya, maka pemain kembali ke menu utama. Jika tidak maka pemain keluar dari permainan.

3.2.3 Perancangan Kecerdasan Musuh

Finite State Machine (FSM) adalah algoritma yang akan diterapkan ke dalam perilaku musuh dengan memanfaatkan sensor pandang dan sensor hindar seperti yang terlihat pada gambar 3.2.



Gambar. 3.2 Perancangan kecerdasan musuh

Pada *game* Malam Jum'at Kliwon terdapat 16 tempat diantaranya 4 tempat Hantu Boss, 11 tempat Hantu Anak Buah dan 1 tempat netral. Untuk jumlah Hantu dalam setiap arena, seperti yang tampak pada tabel 3.1.

Tabel 3.1. Jumlah Hantu

Nama arena	Jumlah hantu
Arena.1	4 Hantu Anak Buah
Arena.2	5 Hantu Anak Buah
Arena Boss.1	1 Hantu Boss
Arena.3	4 Hantu Anak Buah
Arena.4	5 Hantu Anak Buah
Arena.5	5 Hantu Anak Buah
Arena Boss.2	1 Hantu Boss
Arena.6	5 Hantu Anak Buah
Arena.7	4 Hantu Anak Buah
Arena.8	4 Hantu Anak Buah
Arena Boss.3	1 Hantu Boss
Arena.9	4 Hantu Anak Buah
Arena.10	5 Hantu Anak Buah
Arena.11	5 Hantu Anak Buah
Arena Boss.4	1 Hantu Boss
Arena.12	0 Hantu

3.3. Proses Perhitungan

3.3.1 Perhitungan Kecerdasan Musuh

Umumnya *Finite State Machine* digambarkan berupa blok diagram. Sedangkan, dalam proses perhitungan *Finite State Machine* dinyatakan dengan

$$\Sigma = \delta(Q, S, F) \quad (1)$$

dimana:

Σ = Himpunan input.

δ = Fungsi transisi.

Q = Himpunan semua state.

S = State awal.

F = State akhir.

Pada FSM Hantu Anak Buah satuan-satuan tersebut berlaku sebagai berikut :

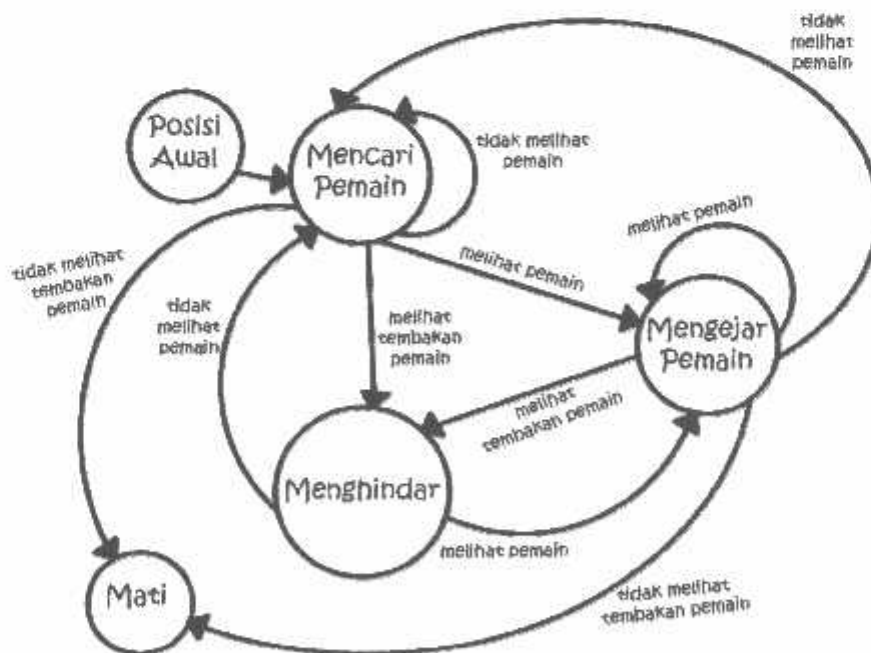
$Q = \{ \text{mencari pemain, mengejar pemain, menghindari, mati} \}$

$\Sigma = \{ \text{melihat pemain, tidak melihat pemain, melihat tembakan pemain, tidak melihat tembakan pemain} \}$

$S = \{ \text{mencari pemain} \}$

$F = \{ \text{mati} \}$

Untuk blok diagram FSM perilaku Hantu Anak Buah seperti yang terlihat pada gambar 3.3.



Gambar. 3.3 Blok diagram Finite State Machine Hantu Anak Buah

Untuk *lookup table* FSM Hantu Anak Buah dapat seperti yang tampak pada tabel 3.2.

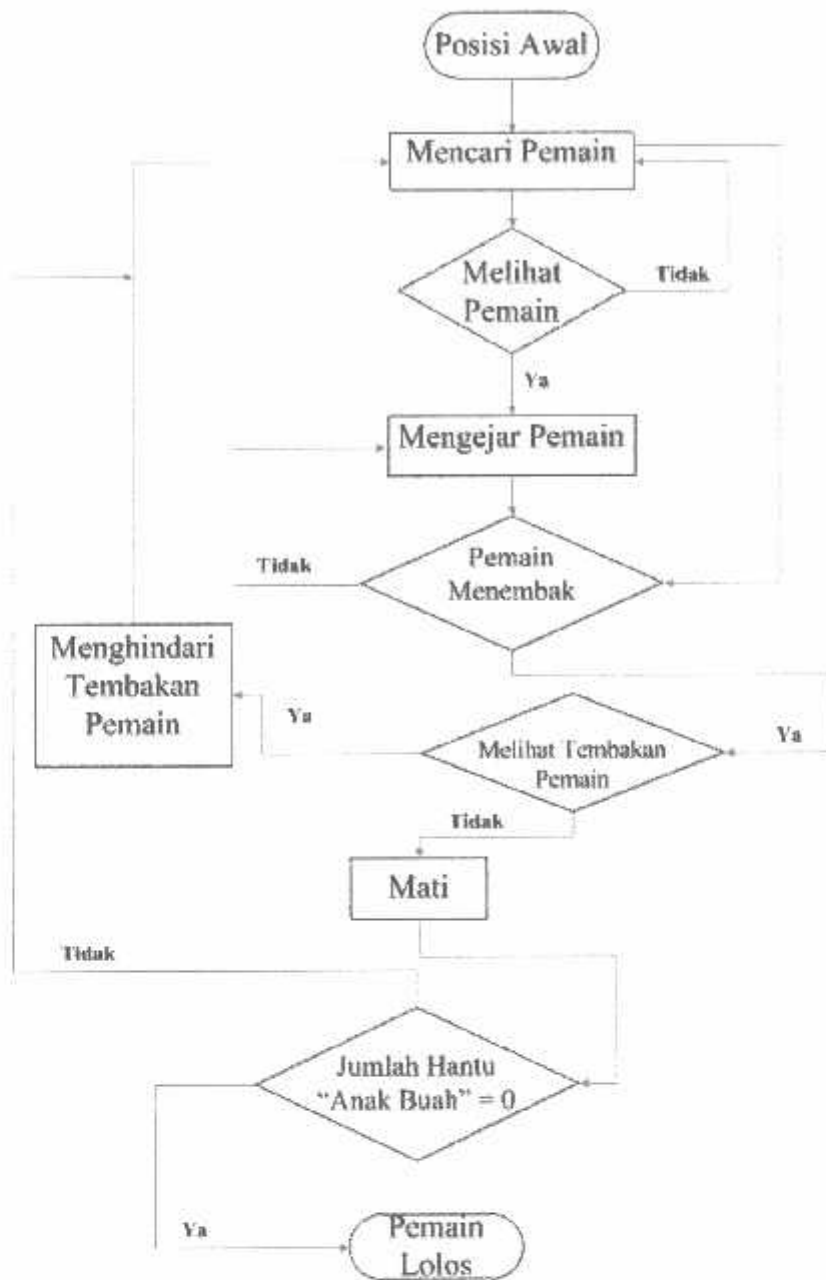
Tabel 3.2. Tabel FSM Hantu Anak Buah

δ	INPUT			
	Melihat pemain	Tidak melihat pemain	Melihat tembakan pemain	Tidak melihat tembakan pemain
Mencari pemain	Mengejar pemain	Mencari pemain	Menghindar	Mati
Mengejar pemain	Mengejar pemain	Mencari pemain	Menghindar	Mati
Menghindar	Mengejar pemain	Mencari pemain	-	-
Mati	-	-	-	-

Untuk flowchart FSM perilaku Hantu Anak Buah seperti yang terlihat pada gambar 3.4.

Flowchart FSM perilaku Hantu Anak Buah dapat dijelaskan sebagai berikut :

1. Posisi awal Hantu Anak Buah adalah mencari pemain.
2. Kemudian, apakah Hantu Anak Buah melihat pemain? Jika iya, maka Hantu Anak Buah akan mengejar pemain. Jika tidak, maka Hantu Anak Buah tetap pada aksi mencari pemain.
3. Ketika dalam pengejaran, apakah pemain melakukan aksi menembak? Jika tidak, maka Hantu Anak Buah akan tetap mengejar pemain. Jika pemain melakukan aksi menembak, apakah Hantu Anak Buah melihat laju tembakan pemain? Jika iya, maka Hantu Anak Buah akan menghindarinya. Jika tidak, maka Hantu Anak Buah akan mati.
4. Ketika Hantu Anak Buah menghindar dari laju tembakan pemain, ada kemungkinan pemain melarikan diri atau bersembunyi. Maka, Hantu Anak Buah kembali melakukan aksi awal mencari pemain.
5. Pada *game* Malam Jum'at Kliwon, jumlah Hantu Anak Buah lebih dari 1 sosok dalam satu arena. Jika jumlah Hantu Anak Buah dalam satu arena sudah mati semua, maka pemain lolos ke tempat selanjutnya.



Gambar. 3.4 Flowchart Finite State Machine Hantu Anak Buah

Pada FSM Hantu Boss satuan-satuan tersebut berlaku sebagai berikut :

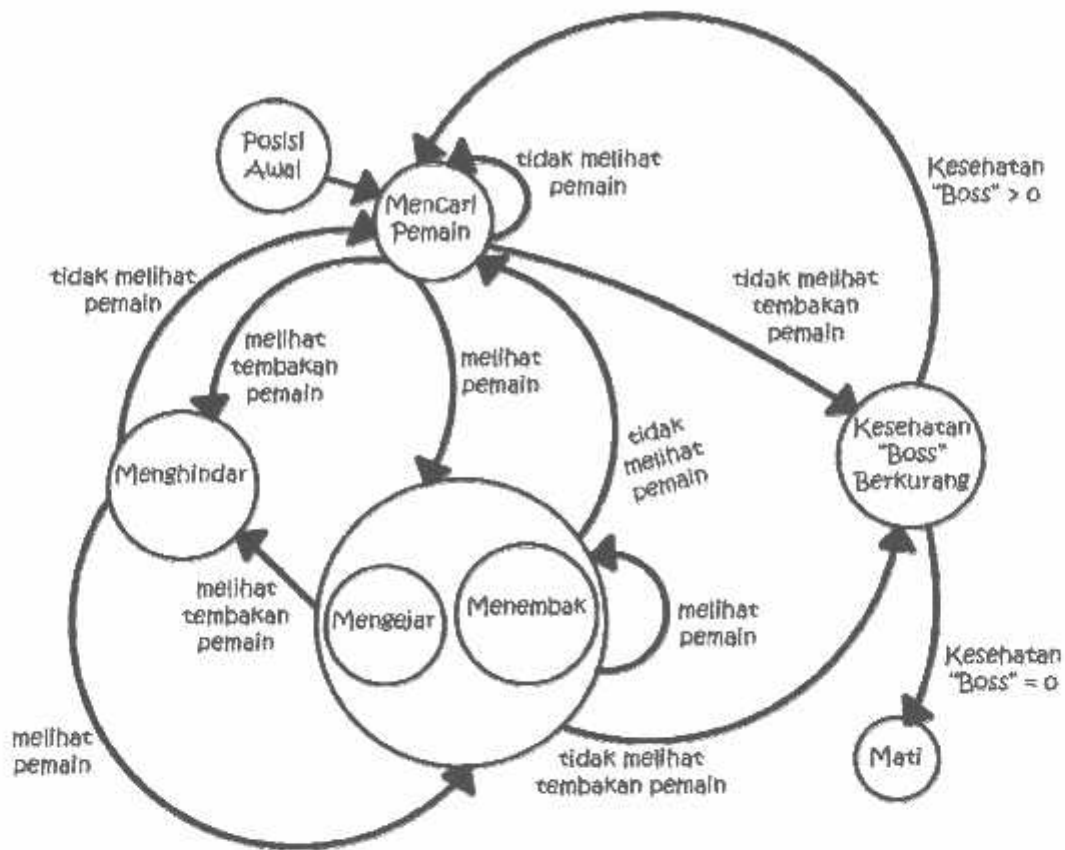
$Q = \{ \text{mencari pemain, mengejar dan menembak pemain, menghindari, kesehatan Boss berkurang, mati} \}$

$\Sigma = \{ \text{melihat pemain, tidak melihat pemain, melihat tembakan pemain, tidak melihat tembakan pemain, kesehatan Boss} > 0, \text{ kesehatan Boss} = 0 \}$

$S = \{ \text{mencari pemain} \}$

$F = \{ \text{mati} \}$

Untuk blok diagram FSM perilaku Hantu Boss seperti yang terlihat gambar 3.5.



Gambar. 3.5 Blok diagram Finite State Machine Hantu Boss

Sedangkan, untuk *lookup table* FSM Hantu Boss dapat seperti yang tampak pada tabel 3.3.

Tabel 3.3. Tabel FSM Hantu Boss

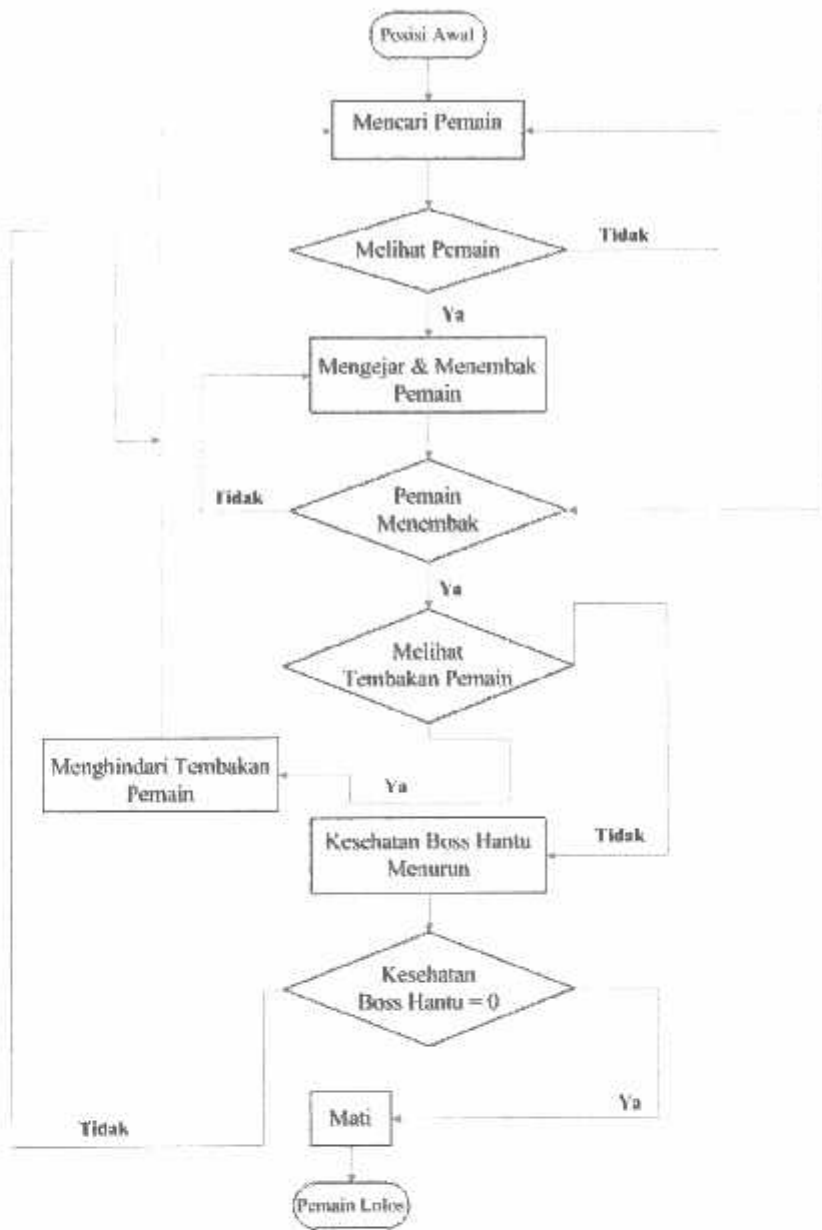
δ	INPUT					
	Melihat pemain	Tidak melihat pemain	Melihat tembakan pemain	Tidak melihat tembakan pemain	Kesehatan Boss > 0	Kesehatan Boss = 0
Mencari pemain	Mengejar & menembak pemain	Mencari pemain	Menghindar	Kesehatan Boss berkurang	-	-
Mengejar & Menembak pemain	Mengejar & menembak pemain	Mencari pemain	Menghindar	Kesehatan Boss berkurang	-	-

Menghindar	Mengejar & menembak pemain	Mencari pemain	-	-	-	-
Kesehatan Boss berkurang	-	-	-	-	Mencari pemain	Mati
Mati	-	-	-	-	-	-

Untuk *flowchart* FSM perilaku Hantu Boss seperti yang terlihat pada gambar 3.6.

Flowchart FSM perilaku Hantu Boss dapat dijelaskan sebagai berikut :

1. Posisi awal Hantu Boss adalah mencari pemain.
2. Kemudian, apakah Hantu Boss melihat pemain? Jika iya, maka Hantu Boss akan mengejar dan menembak pemain. Jika tidak, maka Hantu Boss tetap pada aksi mencari pemain.
3. Ketika dalam pengejaran, apakah pemain melakukan aksi menembak? Jika tidak, maka Hantu Boss akan tetap mengejar dan menembak pemain. Jika pemain melakukan aksi menembak, apakah Hantu Boss melihat laju tembakan pemain? Jika iya, maka Hantu Boss akan menghindarinya. Jika tidak, maka kesehatan pada *healthbar* Hantu Boss akan berkurang atau menurun.
4. Ketika Hantu Boss menghindar dari laju tembakan pemain, ada kemungkinan pemain melarikan diri atau bersembunyi. Maka, Hantu Boss kembali melakukan aksi awal mencari pemain.
5. Jika kesehatan pada *healthbar* Hantu Boss = 0, maka Hantu Boss mati dan pemain lolos ke tempat selanjutnya.



Gambar. 3.6 Flowchart Finite State Machine Hantu Boss

3.3.2 Perhitungan Score

Perhitungan *score* pemain dalam *game* Malam Jum'at Kliwon meliputi beberapa variabel yang terdiri dari :

Nyawa pemain = 1

$Score = 10point/Hantu\ Anak\ Buah$

$Score\ bonus = 1000point/Hantu\ Boss$

$Nyawa\ bonus = Nyawa\ pemain + 1/Hantu\ Boss$

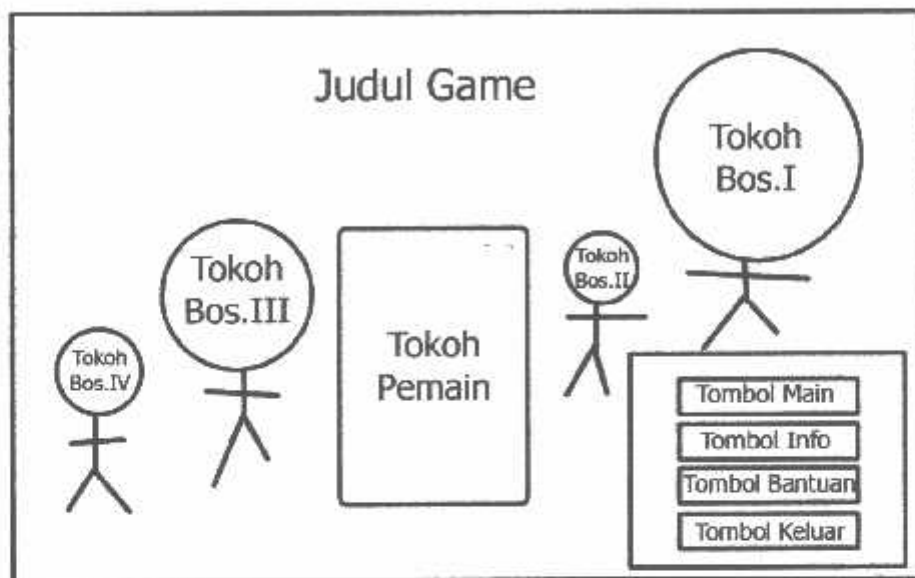
Total *score* pemain =

$(Jumlah\ Score\ Hantu\ Anak\ Buah + Jumlah\ Score\ Hantu\ Boss) * Jumlah\ Nyawa\ pemain$

3.4. Properti Game

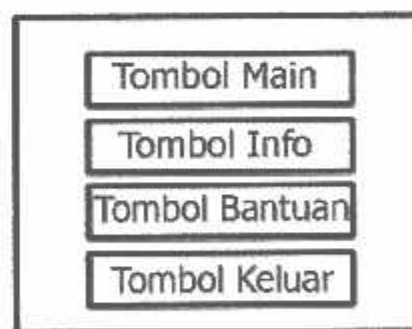
3.4.1 Perancangan Menu Utama

Perancangan menu utama adalah tahapan sebelum merancang *game*. Seperti pada gambar 3.7 memperlihatkan tokoh utama dan 4 Hantu Boss sebagai musuh besarnya.



Gambar. 3.7 Rancangan tampilan menu utama

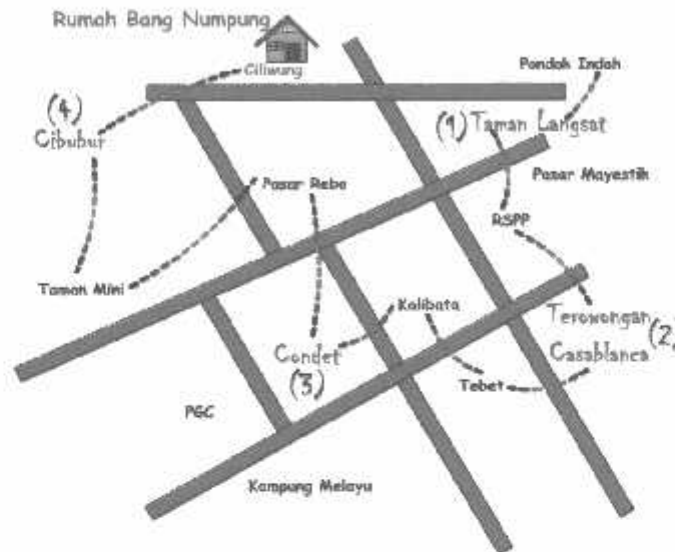
Pada perancangan tampilan menu terdapat empat *button* yaitu *button* main yang menghubungkan ke halaman permainan, *button* bantuan yang memunculkan tampilan peta, navigasi, maupun petunjuk permainan, *button* info yang memunculkan tampilan profil. Sedangkan pada *button* keluar untuk menutup permainan ini. Seperti yang terlihat pada gambar 3.8



Gambar. 3.8 Rancangan button menu utama

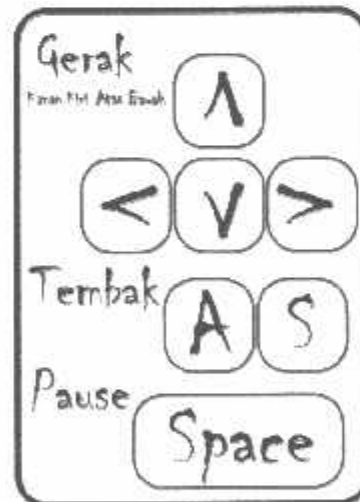
3.4.2 Perancangan Tampilan Bantuan

Tampilan bantuan terdiri dari tampilan peta, tampilan navigasi, dan tampilan petunjuk *game*. Seperti pada gambar 3.9, perancangan tampilan peta berguna untuk mengetahui rute dan tahapan hantu yang harus di lalui.



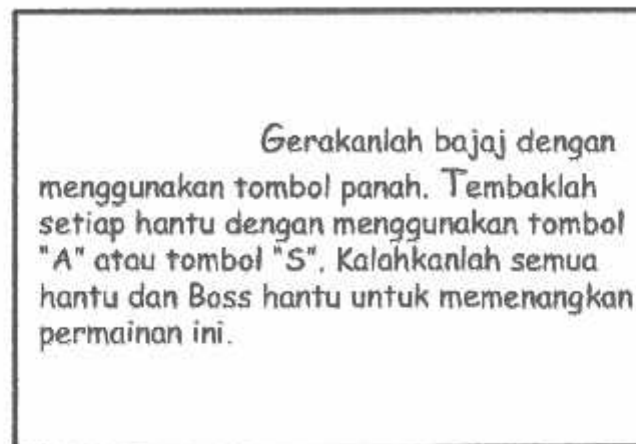
Gambar. 3.9 Rancangan tampilan peta

Tampilan navigasi ditujukan kepada pemain tentang *key* yang digunakan selama permainan. Perancangan tampilan navigasi terlihat seperti pada gambar 3.10.



Gambar. 3.10 Rancangan tampilan navigasi

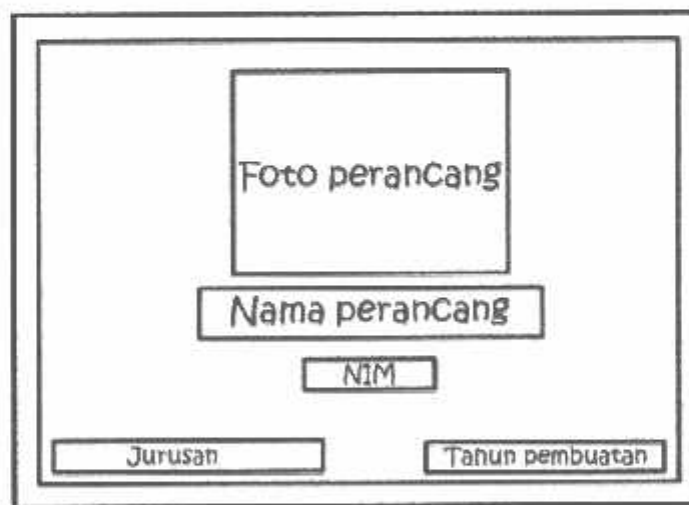
Tampilan petunjuk adalah tampilan yang menyajikan petunjuk cara melakukan permainan pada *game* Malam Jum'at Kliwon. Perancangan tampilan petunjuk terlihat seperti pada gambar 3.11.



Gambar. 3.11 Rancangan tampilan petunjuk

3.4.3 Perancangan Tampilan Profil

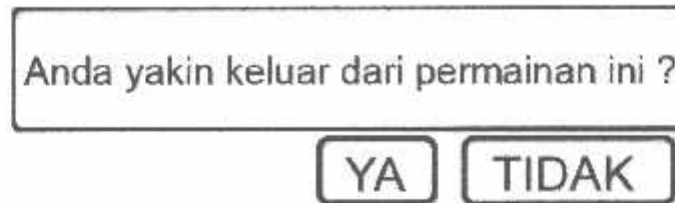
Tampilan yang menampilkan profil dari penulis berupa foto, nama lengkap, jurusan dan tahun pembuatan *game* Malam Jum'at Kliwon. Perancangan tampilan profil terlihat seperti pada gambar 3.12.



Gambar. 3.12 Rancangan tampilan profil

3.4.4 Perancangan Tampilan Quit

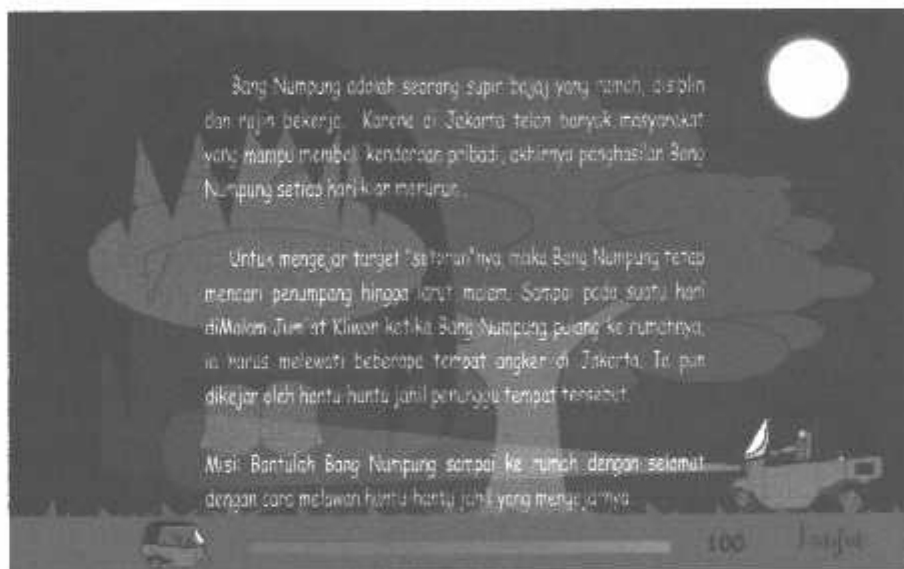
Tampilan *quit* ditujukan sebagai opsi apakah pemain akan keluar permainan atau tidak. Pada tampilan ini terdapat dua *button* pilihan ya atau tidak. Perancangan tampilan *quit* terlihat pada gambar 3.13.



Gambar. 3.13 Rancangan tampilan *quit*

3.4.5 Perancangan Tampilan Story Board

Pada gambar 3.14 perancangan tampilan alur cerita. Tampilan ini adalah tampilan sebelum arena.1 yang menuliskan alur cerita atau *story board* dari *game* Malam Jum'at Kliwon.

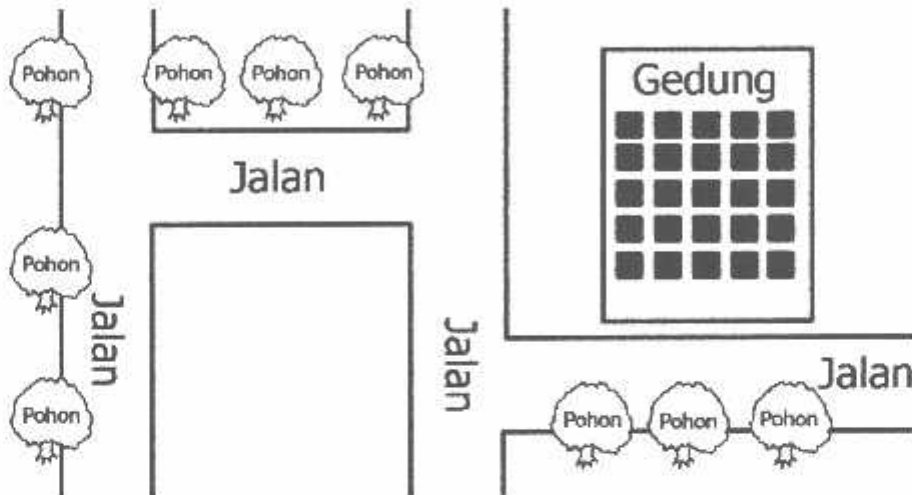


Gambar. 3.14 Rancangan tampilan *story board*

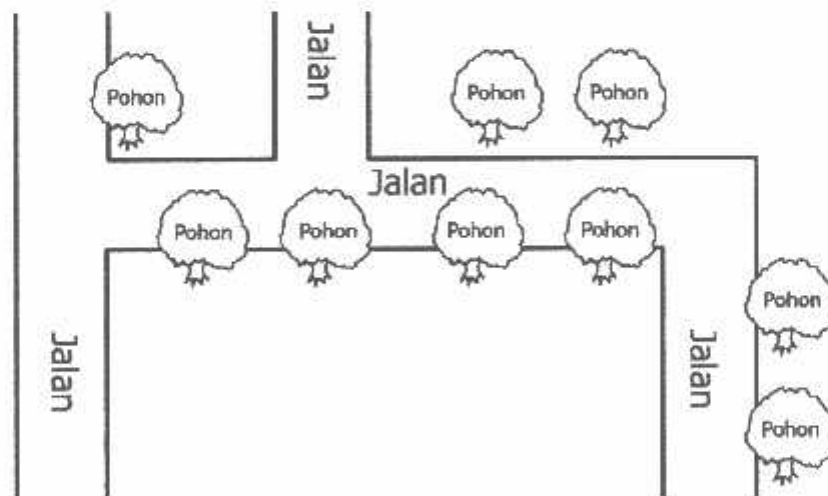
3.4.6 Perancangan Arena Game

Dalam *game* Malam Jum'at Kliwon terdapat 16 tempat diantaranya 4 tempat Hantu Boss, dua belas tempat Hantu Anak Buah dan satu tempat netral. Arena ini merupakan lokasi terjadinya konflik pemain dengan musuh dengan tujuan untuk berpindah ke tempat berikutnya. Seperti yang terlihat pada gambar 3.15 perancangan tampilan arena.1, gambar 3.16 perancangan tampilan arena.2, gambar 3.17 perancangan

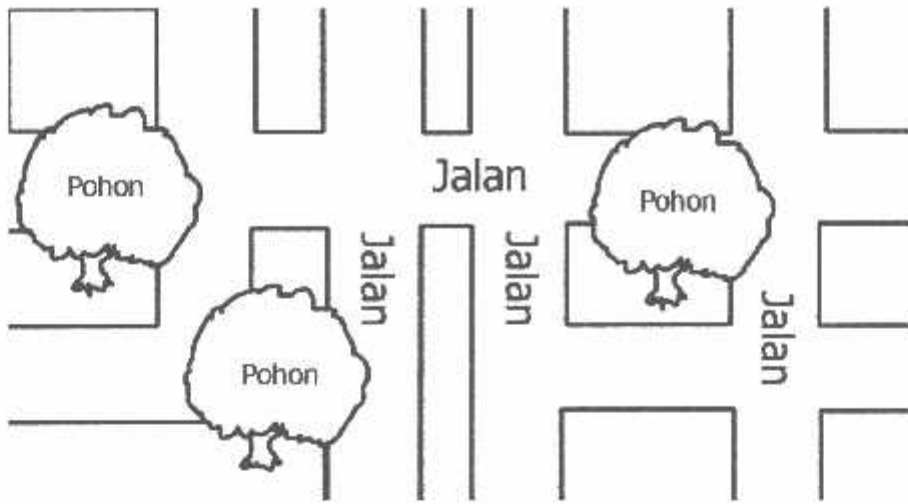
tampilan arena Boss.1, gambar 3.18 perancangan tampilan arena.3, gambar 3.19 perancangan tampilan arena.4, gambar 3.20 perancangan tampilan arena.5, gambar 3.21 perancangan tampilan arena Boss.2, gambar 3.22 perancangan tampilan arena.6, gambar 3.23 perancangan tampilan arena.7, gambar 3.24 perancangan tampilan arena.8, gambar 3.25 perancangan tampilan arena Boss.3, gambar 3.26 perancangan tampilan arena.9, gambar 3.27 perancangan tampilan arena.10, gambar 3.28 perancangan tampilan arena.11, gambar 3.29 perancangan tampilan arena Boss.4, gambar 3.30 perancangan tampilan arena.12.



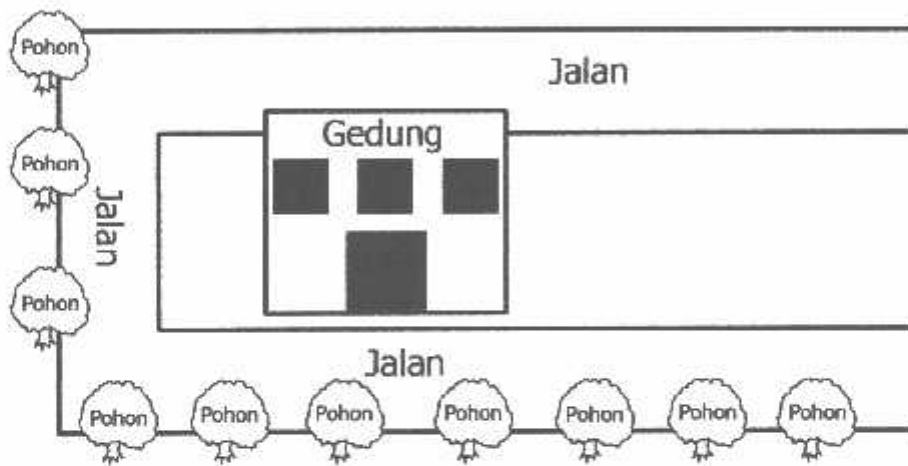
Gambar. 3.15 Rancangan tampilan arena.1



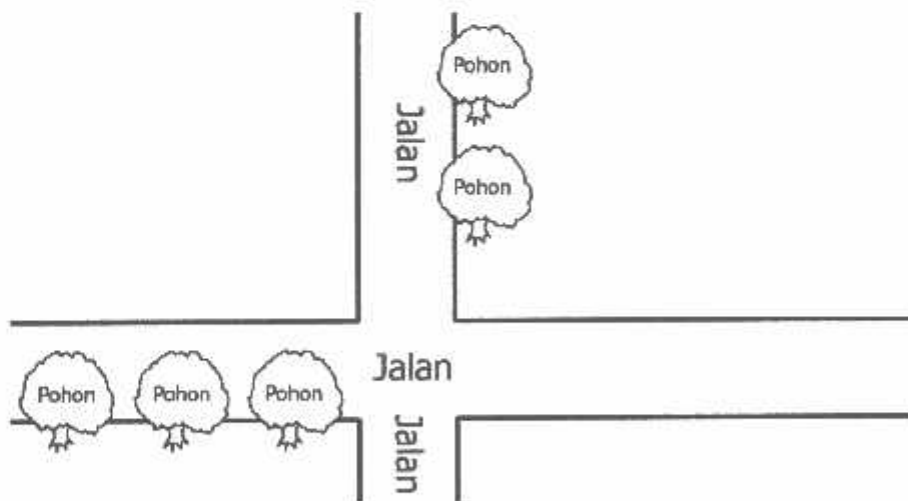
Gambar. 3.16 Rancangan tampilan arena.2



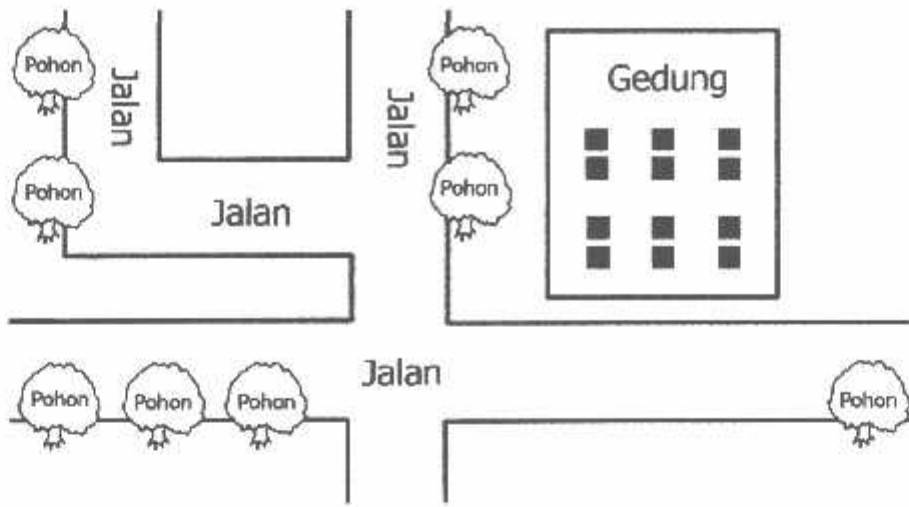
Gambar. 3.17 Rancangan tampilan arena Boss.1



Gambar. 3.18 Rancangan tampilan arena.3



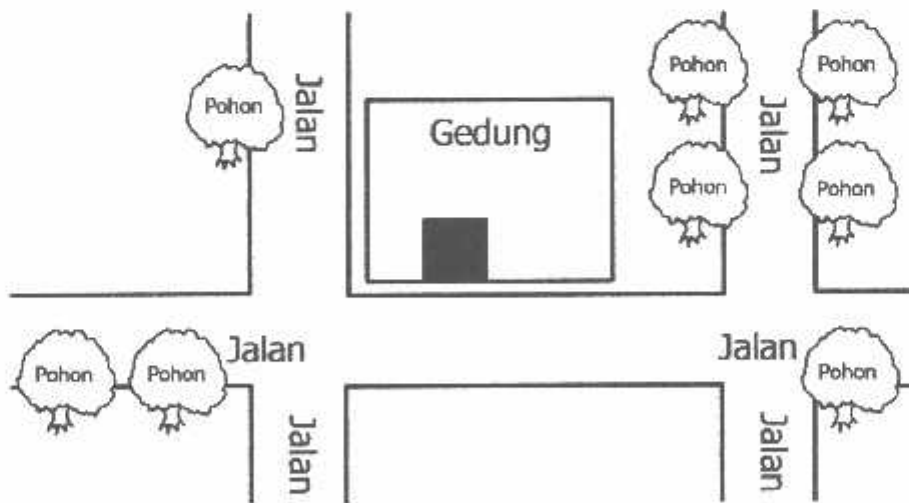
Gambar. 3.19 Rancangan tampilan arena.4



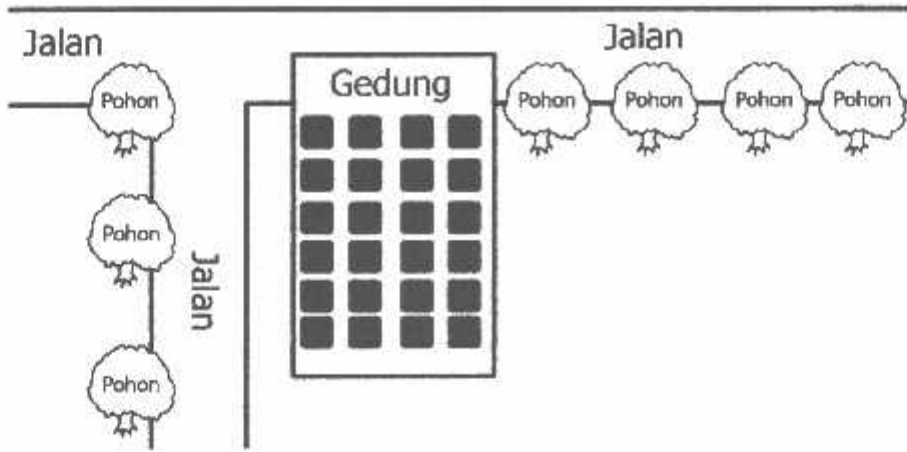
Gambar. 3.20 Rancangan tampilan arena.5



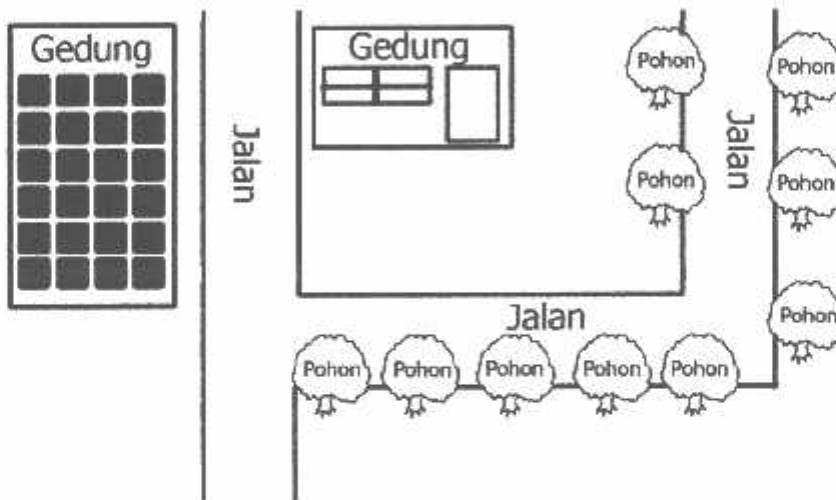
Gambar. 3.21 Rancangan tampilan arena Boss.2



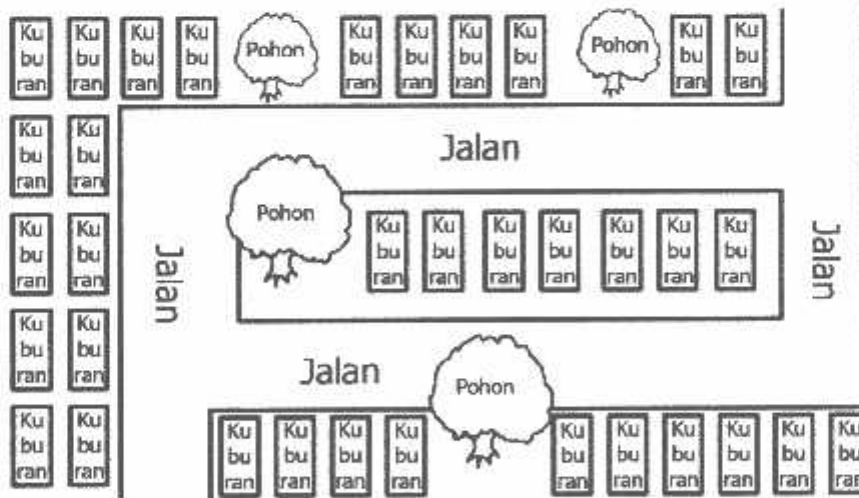
Gambar. 3.22 Rancangan tampilan arena.6



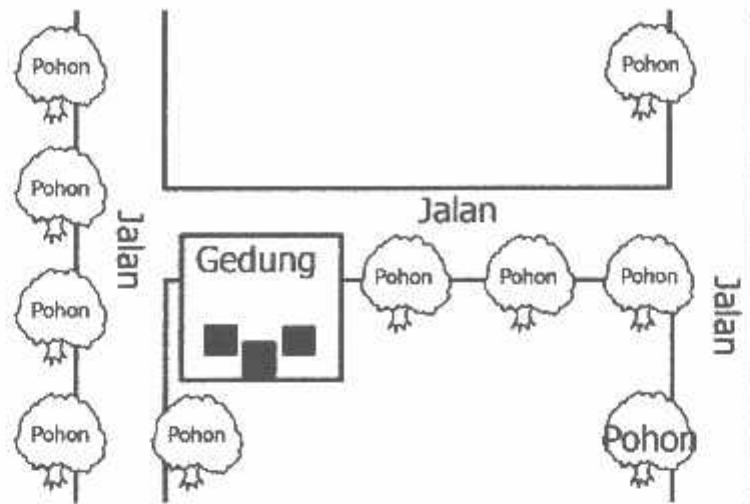
Gambar. 3.23 Rancangan tampilan arena.7



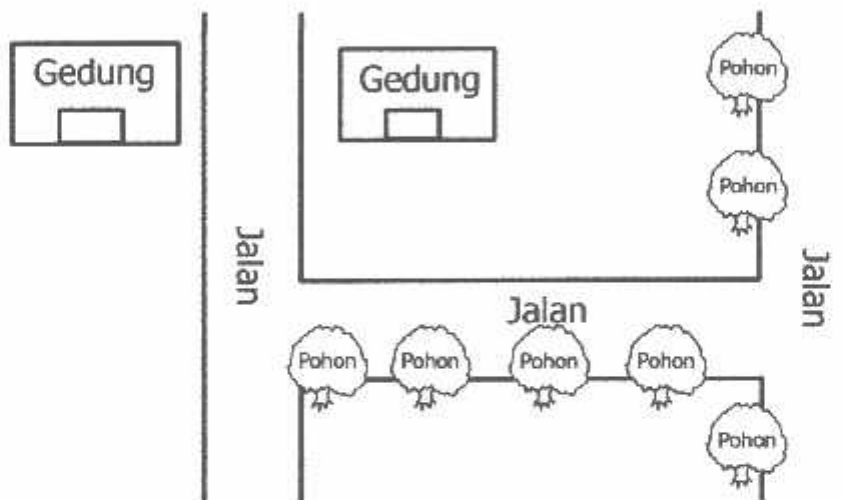
Gambar. 3.24 Rancangan tampilan arena.8



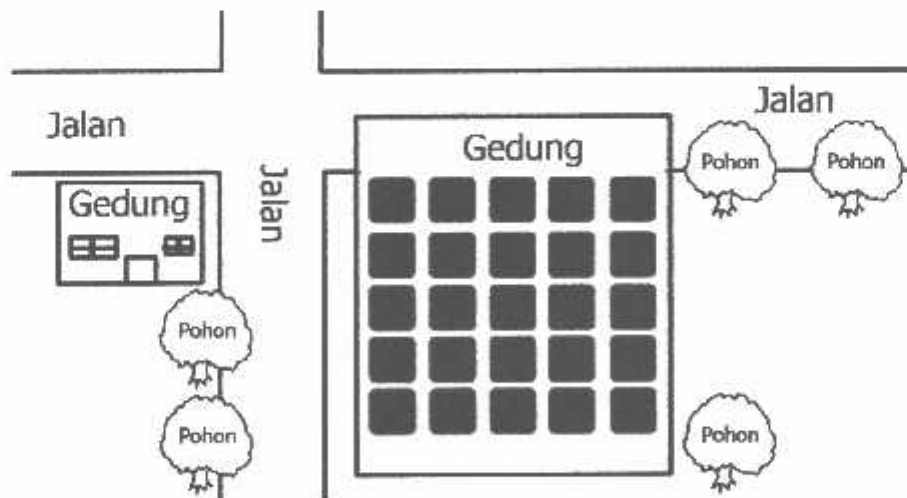
Gambar. 3.25 Rancangan tampilan arena Boss.3



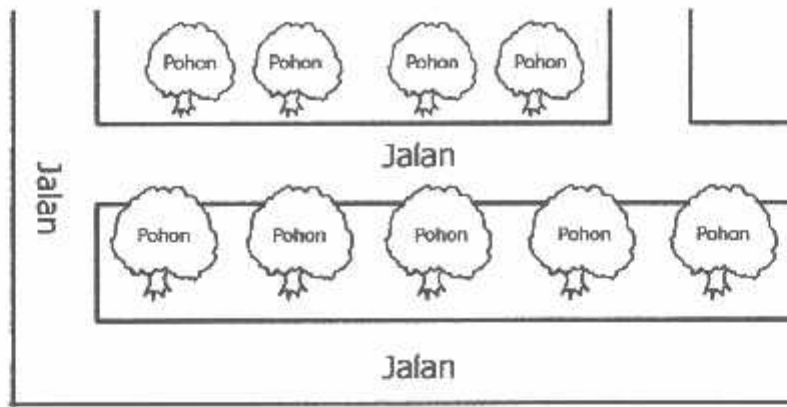
Gambar. 3.26 Rancangan tampilan arena.9



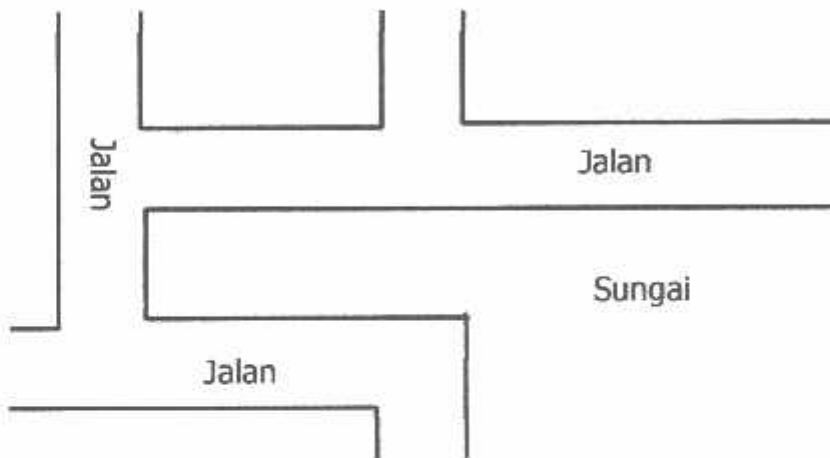
Gambar. 3.27 Rancangan tampilan arena.10



Gambar. 3.28 Rancangan tampilan arena.11



Gambar. 3.29 Rancangan tampilan arena Boss.4



Gambar. 3.30 Rancangan tampilan arena.12

3.4.7 Perancangan Tampilan Ending

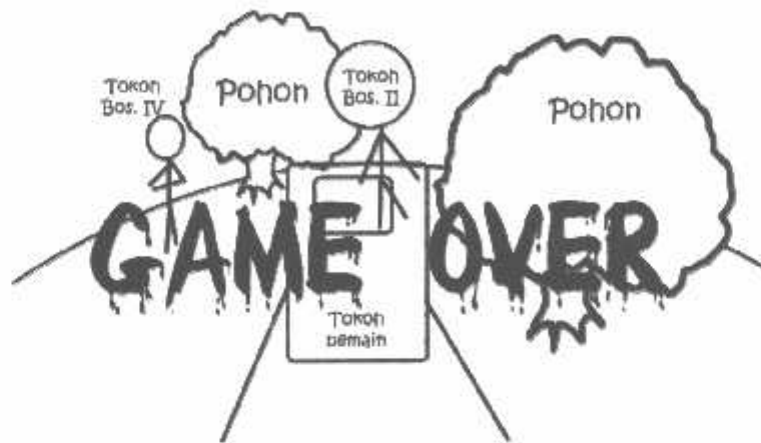
Ending adalah adegan yang muncul setelah pemain mengalahkan semua hantu. Adegan ini muncul sebagai pertanda berakhirnya *game* ini. Gambar 3.31 menampilkan perancangan tampilan *ending* pada *game* Malam Jum'at Kliwon.



Gambar. 3.31 Rancangan tampilan ending

3.4.8 Perancangan Tampilan Game Over

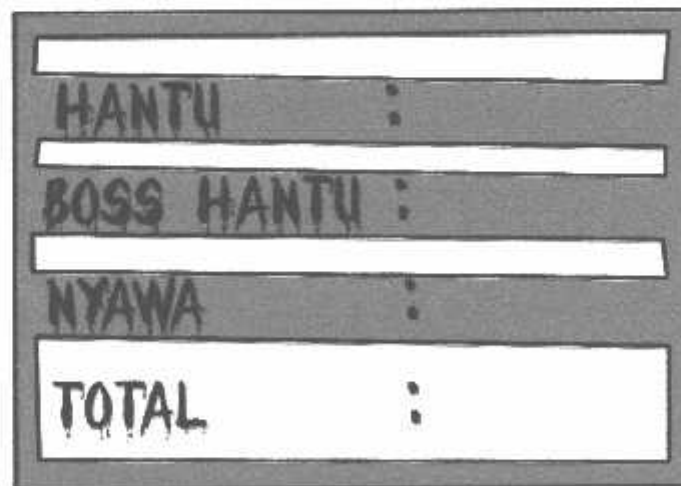
Tampilan *game over* adalah tampilan yang akan muncul ketika pemain mati. Tampilan ini adalah pertanda bahwa, pemain gagal dalam menyelesaikan *game* ini. Untuk perancangan tampilan *game over* pada *game* Malam Jum'at Kliwon terlihat pada gambar 3.32.



Gambar. 3.32 Rancangan tampilan game over

3.4.9 Perancangan Tampilan Score

Tampilan *score* adalah tampilan yang memperlihatkan hasil akumulasi *score* akhir yang diperoleh pemain. Seperti pada gambar 3.33 perancangan tampilan *score* pada *game* Malam Jum'at Kliwon.

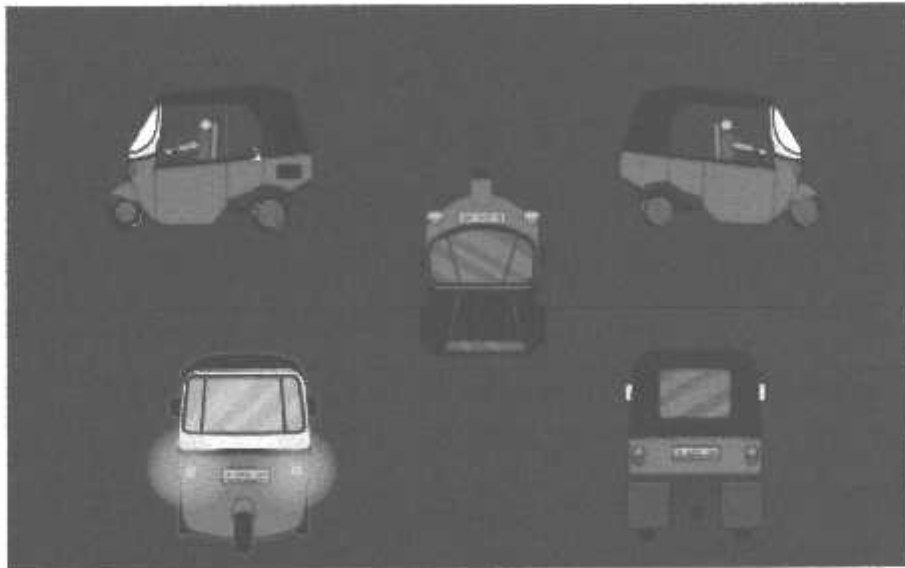


Gambar. 3.33 Rancangan tampilan score

3.5. Komposisi Game

3.5.1 Perancangan Karakter Pemain

Seperti pada gambar 3.34 karakter pemain dalam *game* Malam Jum'at Kliwon adalah bajaj yang dilengkapi bawang putih dan bawang merah sebagai senjata untuk menembak para hantu.



Gambar. 3.34 Rancangan karakter pemain

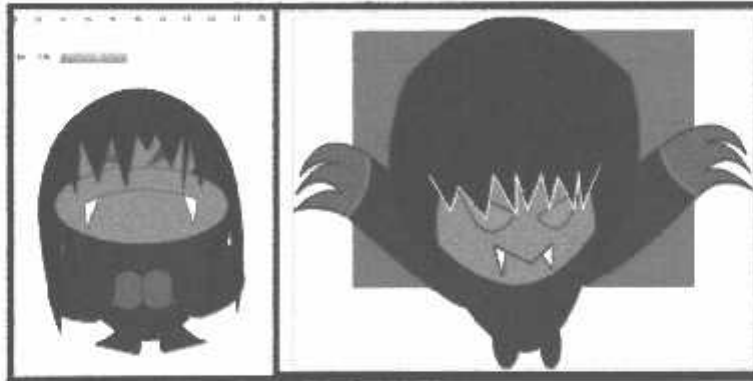
3.5.2 Perancangan Karakter Musuh

Musuh yang harus dihadapi dalam *game* Malam Jum'at Kliwon ini terdiri dari 50 Hantu Anak Buah. Untuk Hantu Anak Buah hanya terdiri dari satu model seperti pada gambar 3.35.

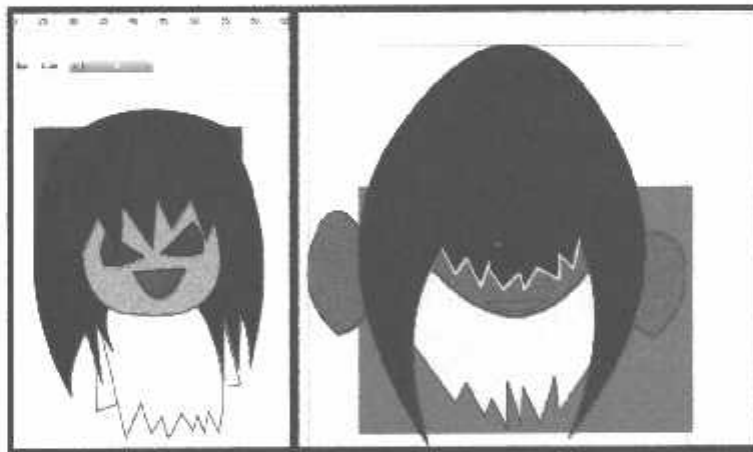
Karakter Hantu Boss terdiri atas 4 Hantu Boss, yaitu pada Boss.1 dengan karakter Gendruwo penghuni Taman Langsung seperti pada gambar 3.36. Sedangkan pada Boss.2 dengan karakter Kuntilanak sebagai penghuni Terowongan *Cassablanca* seperti yang terlihat pada gambar 3.37. Untuk karakter Boss.3 pada gambar 3.38 yaitu Setan Kober (kober = kuburan dalam bahasa Betawi) sosok bola api sebagai penghuni kuburan Condet. Boss yang terakhir seperti pada gambar 3.39, yaitu Mumun karakter pocong (hantu bungkus dalam bahasa Melayu) sebagai penghuni Bumi Perkemahan Cibubur.



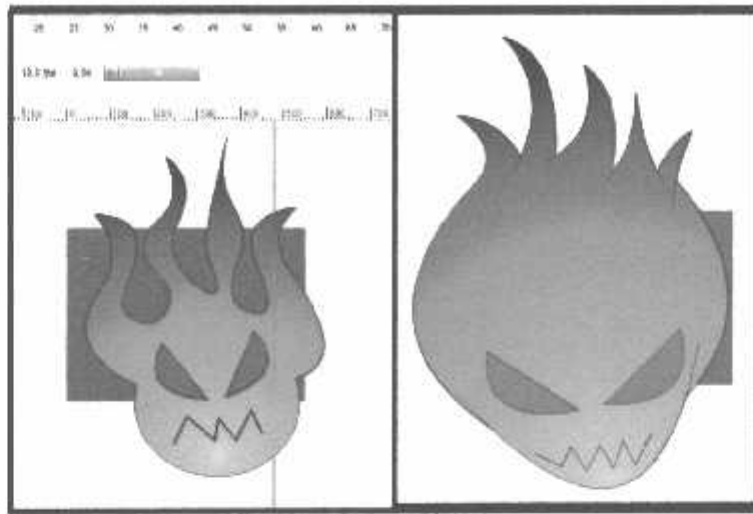
Gambar. 3.35 Rancangan karakter Hantu Anak Buah



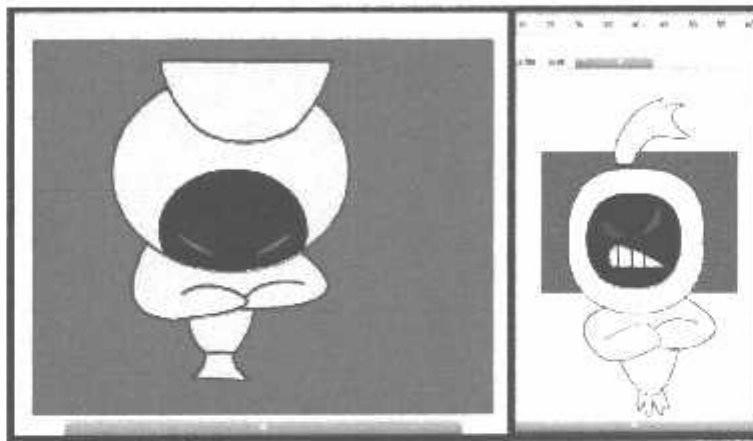
Gambar. 3.36 Rancangan karakter Hantu Boss.1



Gambar. 3.37 Rancangan karakter Hantu Boss.2



Gambar. 3.38 Rancangan karakter Hantu Boss.3



Gambar. 3.39 Rancangan karakter Hantu Boss.4

3.5.3 Perancangan Peluru

Di dalam *game* Malam Jum'at Kliwon, pemain dan Hantu Boss memiliki benda yang digunakan sebagai amunisi dalam baku tembak selama di dalam arena yang sama. Untuk karakter bajaj dilengkapi dengan amunisi bawang putih seperti pada gambar 3.40 dan amunisi bawang merah seperti pada gambar 3.41.



Gambar. 3.40 Rancangan amunisi bawang putih



Gambar. 3.41 Rancangan amunisi bawang merah

Sedangkan untuk karakter Hantu Boss.1 dilengkapi dengan amunisi lingkaran seperti pada gambar 3.42. karakter Hantu Boss.2 dilengkapi dengan amunisi paku seperti pada gambar 3.43, untuk karakter Hantu Boss.3 dilengkapi dengan amunisi api seperti pada gambar 3.44, dan pada karakter Hantu Boss.4 dilengkapi dengan amunisi boomerang seperti pada gambar 3.45.



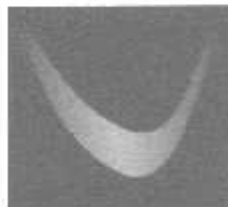
Gambar. 3.42 Rancangan amunisi Hantu Boss.1



Gambar. 3.43 Rancangan amunisi Hantu Boss.2



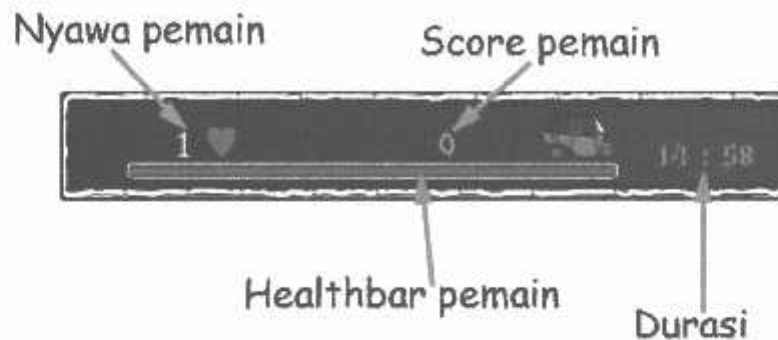
Gambar. 3.44 Rancangan amunisi Hantu Boss.3



Gambar. 3.45 Rancangan amunisi Hantu Boss.4

3.5.4 Perancangan Healthbar

Untuk indikator nyawa pemain dan Hantu Boss pada *game* Malam Jum'at Kliwon ditandai dengan *healthbar*. Seperti pada gambar 3.46 adalah perancangan indikator *healthbar* yang dimiliki pemain.



Gambar. 3.46 Rancangan healthbar pemain

Pada gambar 3.47 adalah perancangan indikator *healthbar* yang dimiliki oleh Hantu Boss.1. untuk gambar 3.48 adalah perancangan indikator *healthbar* yang dimiliki oleh Hantu Boss.2. Sedangkan pada gambar 3.49 adalah perancangan indikator *healthbar* yang dimiliki oleh Hantu Boss.3, dan pada gambar 3.50 adalah perancangan indikator *healthbar* yang dimiliki oleh Hantu Boss.4.



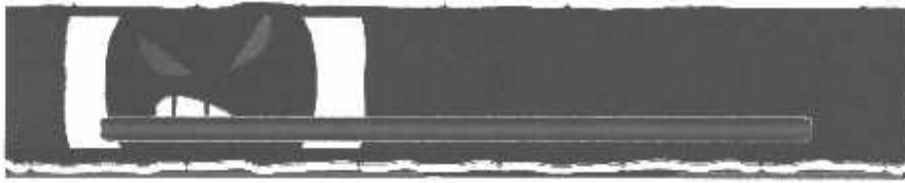
Gambar. 3.47 Rancangan healthbar Hantu Boss.1



Gambar. 3.48 Rancangan healthbar Hantu Boss.2



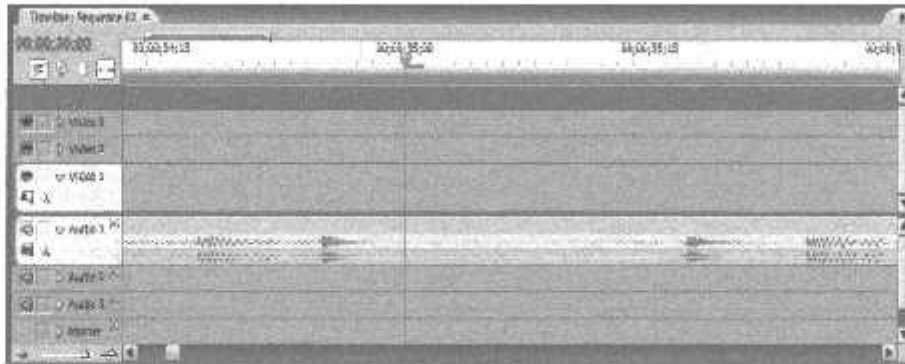
Gambar. 3.49 Rancangan healthbar Hantu Boss 3



Gambar. 3.50 Rancangan healthbar Hantu Boss 4

3.5.5 Perancangan Audio

Pada gambar 3.51 adalah perancangan suara bajaj ketika berjalan dengan menggunakan suara *vespa* yang diedit dalam *Adobe Premiere Pro CS.3*.



Gambar. 3.51 Rancangan suara karakter bajaj ketika berjalan.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Implementasi adalah tahapan dalam membangun dan menjalankan *game* Malam Jum'at Kliwon dengan menerapkan *Finite State Machine* sebagai agen cerdas musuh.

4.1.1 Implementasi Menu Utama

Pada tampilan menu terdapat 4 *button* yaitu *button* main yang menghubungkan ke halaman permainan, *button* bantuan yang memunculkan tampilan peta, navigasi, maupun petunjuk permainan, *button* info yang memunculkan tampilan profil perancang. Sedangkan pada *button* keluar untuk menutup permainan ini. Tampilan menu utama dapat dilihat pada gambar 4.1 dan pada gambar 4.2 tampilan *button* yang terdapat halaman menu utama.



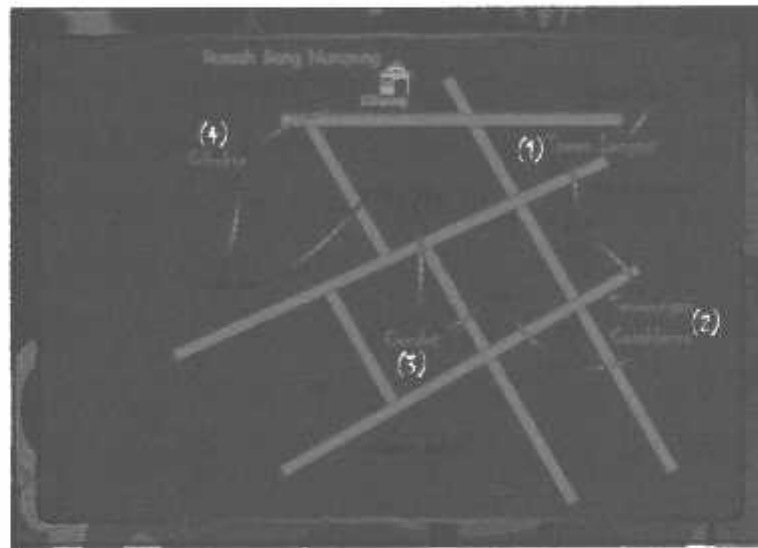
Gambar. 4.1 Tampilan menu utama



Gambar. 4.2 Tampilan button menu utama

4.1.2 Implementasi Tampilan Bantuan

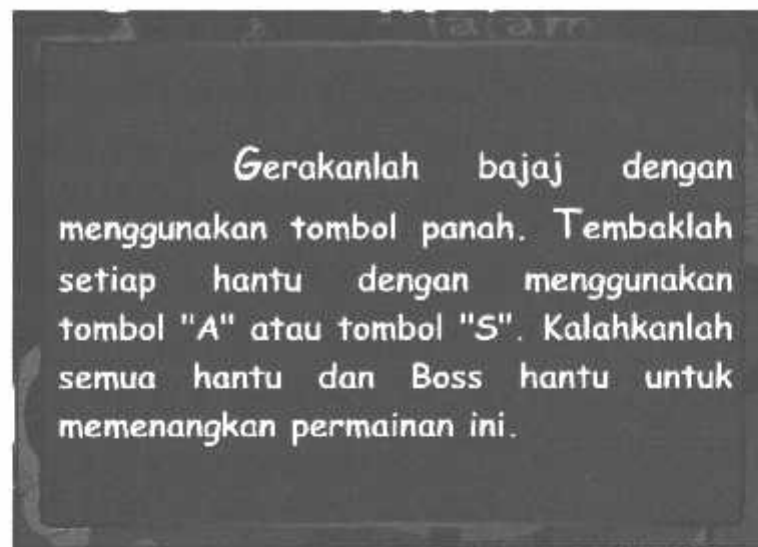
Tampilan peta seperti pada gambar 4.3 tampilan navigasi seperti pada gambar 4.4 dan tampilan petunjuk permainan seperti pada gambar 4.5.



Gambar. 4.3 Tampilan peta



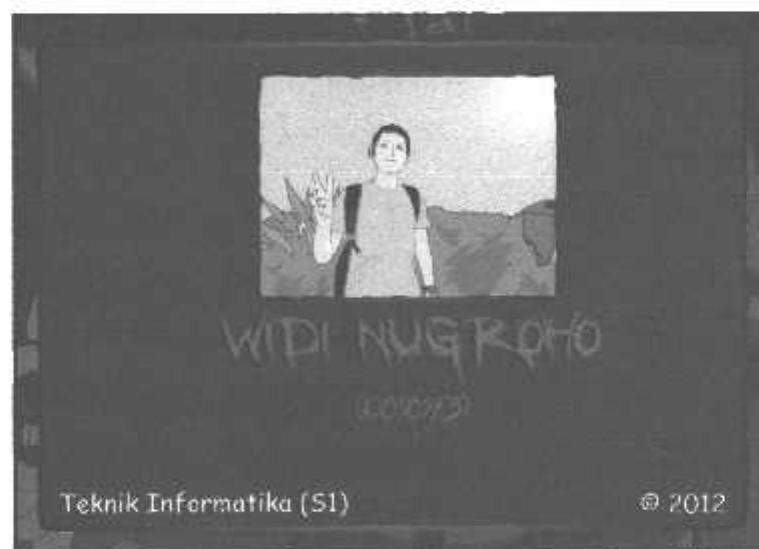
Gambar. 4.4 Tampilan navigasi



Gambar. 4.5 Tampilan petunjuk permainan

4.1.3 Implementasi Tampilan Profil

Seperti pada gambar 4.6, tampilan profil merupakan tampilan yang memberi informasi tentang data diri penulis.



Gambar. 4.6 Tampilan profil

4.1.4 Implementasi Tampilan Quit

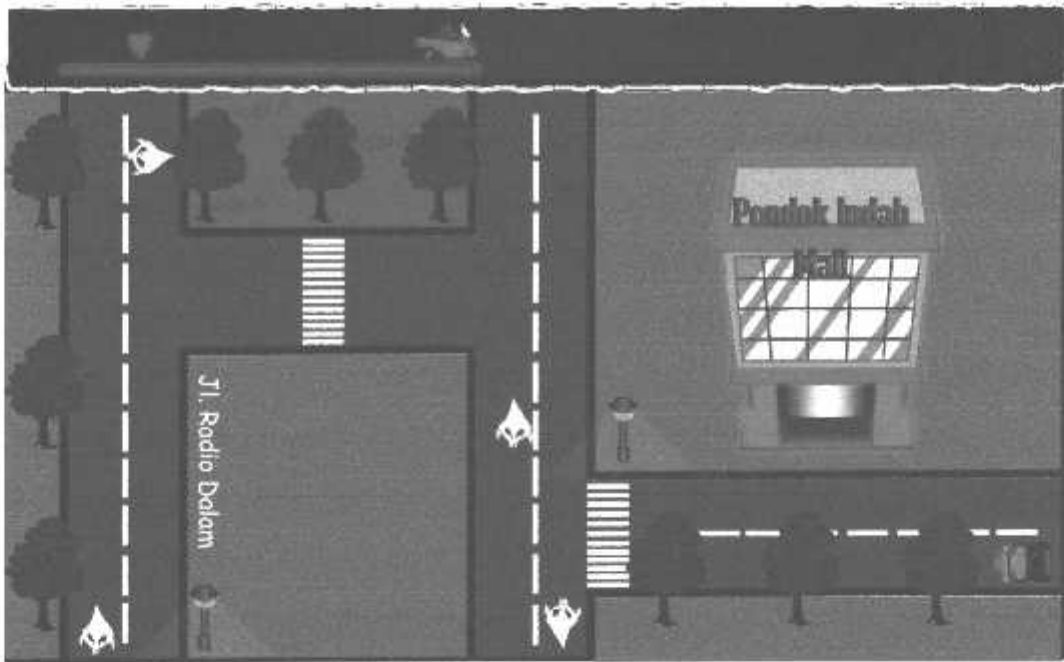
Tampilan *quit* ditujukan sebagai opsi apakah pemain akan keluar permainan atau tidak. Pada tampilan ini terdapat dua *button* pilihan ya atau tidak. Jika *button* Ya diklik, maka keluar dari permainan. Jika *button* Tidak diklik, maka tampilan ini akan hilang dan kembali pada halaman menu utama. Tampilan keluar terlihat pada gambar 4.7.



Gambar. 4.7 Tampilan opsi quit

4.1.5 Implementasi Arena Game

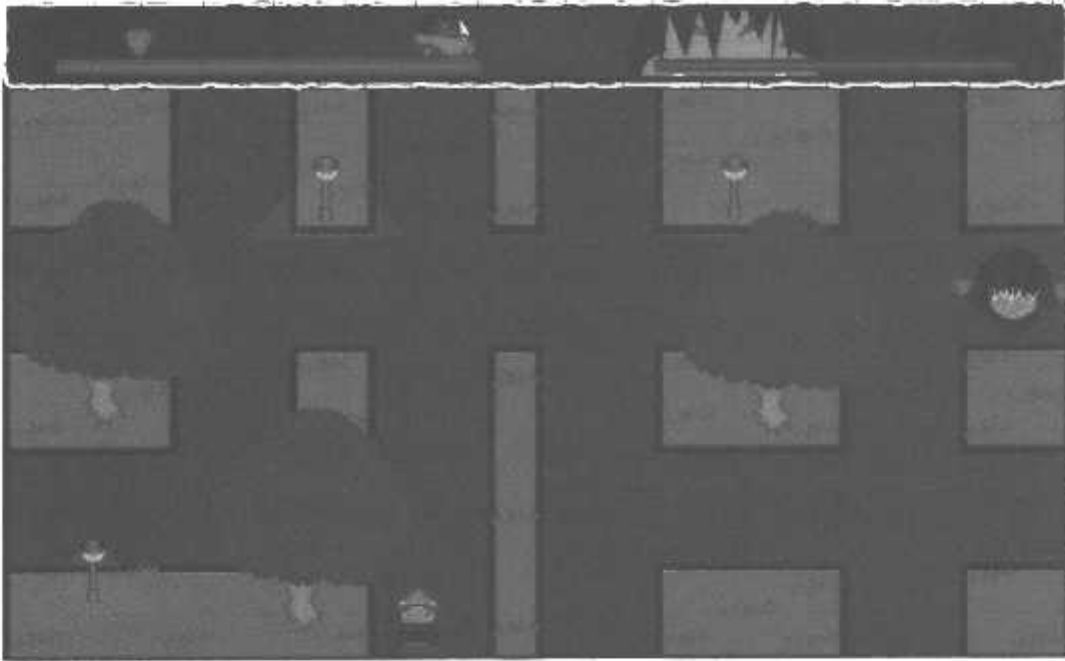
Dalam *game* Malam Jum'at Kliwon terdapat 16 tempat diantaranya 4 tempat Hantu Bos, 11 tempat Hantu Anak Buah dan 1 tempat netral. Seperti yang terlihat pada gambar 4.8 tampilan arena.1, gambar 4.9 tampilan arena.2, gambar 4.10 tampilan arena Boss.1, gambar 4.11 tampilan arena.3, gambar 4.12 tampilan arena.4, gambar 4.13 tampilan arena.5, gambar 4.14 tampilan arena Boss.2, gambar 4.15 tampilan arena.6, gambar 4.16 tampilan arena.7, gambar 4.17 tampilan arena.8, gambar 4.18 tampilan arena Boss.3, gambar 4.19 tampilan arena.9, gambar 4.20 tampilan arena.10, gambar 4.21 tampilan arena.11, gambar 4.22 tampilan arena Boss.4, gambar 4.23 tampilan arena.12.



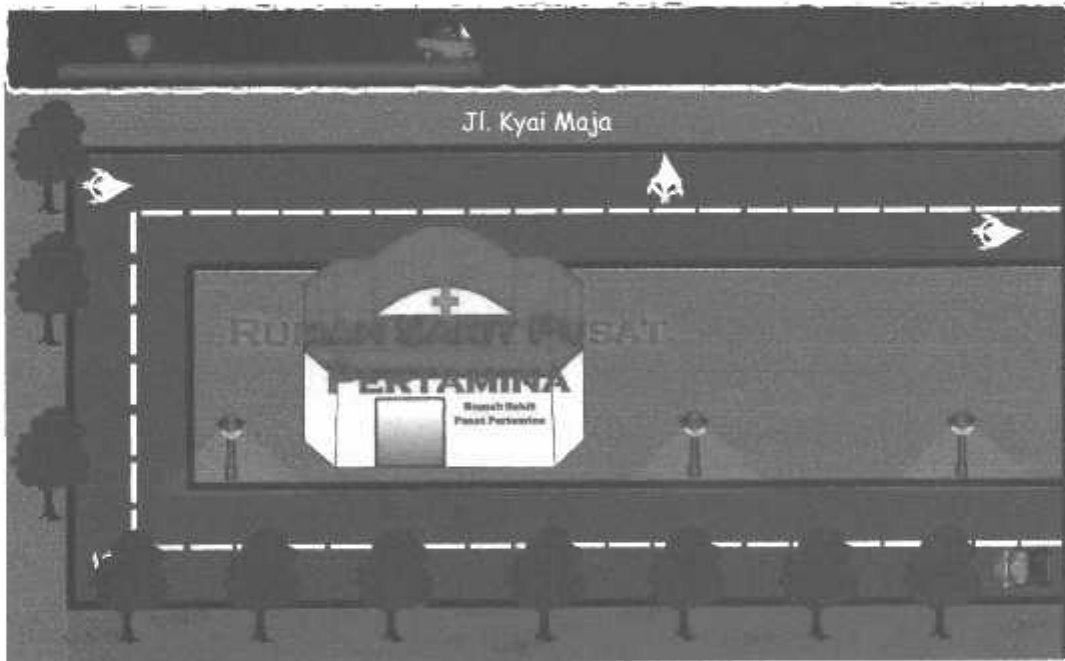
Gambar. 4.8 Tampilan arena.1



Gambar. 4.9 Tampilan arena.2



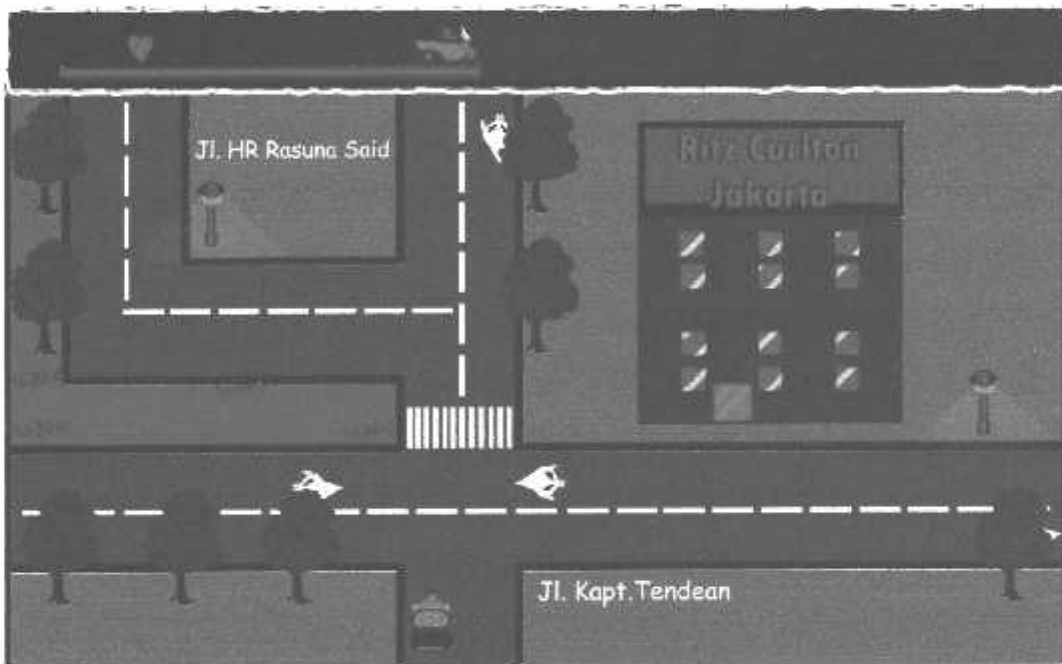
Gambar. 4.10 Tampilan arena Boss.1



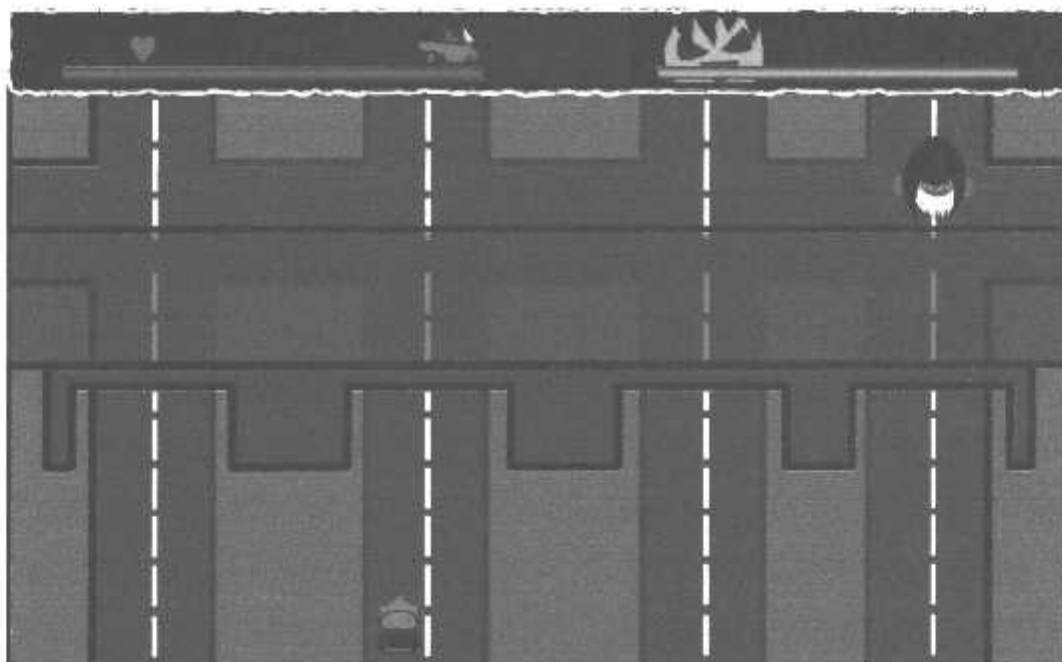
Gambar. 4.11 Tampilan arena.3



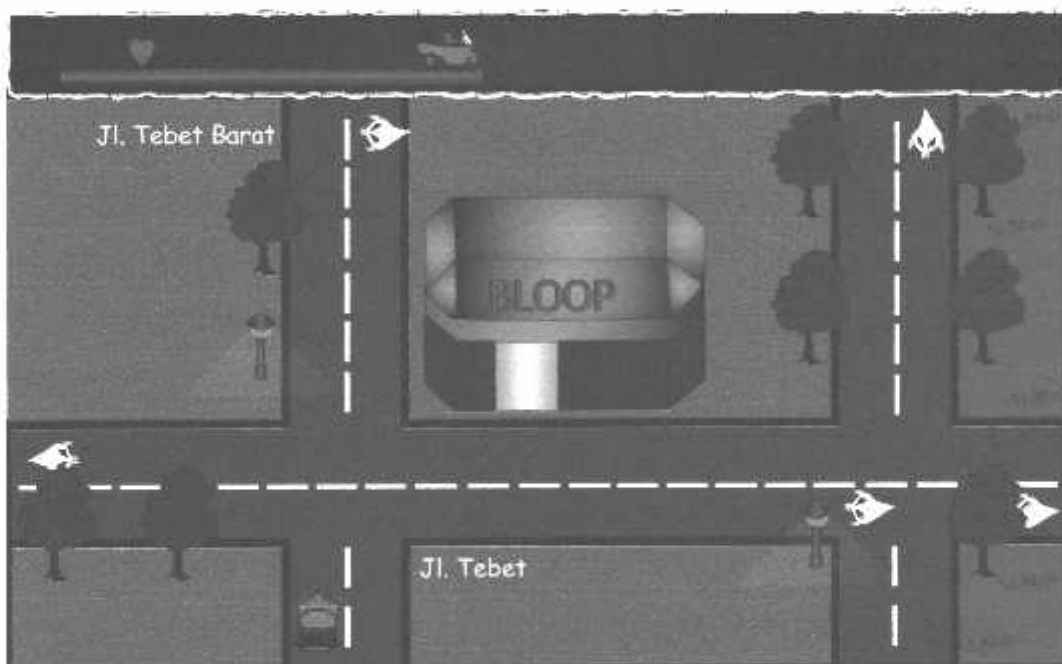
Gambar. 4.12 Tampilan arena.4



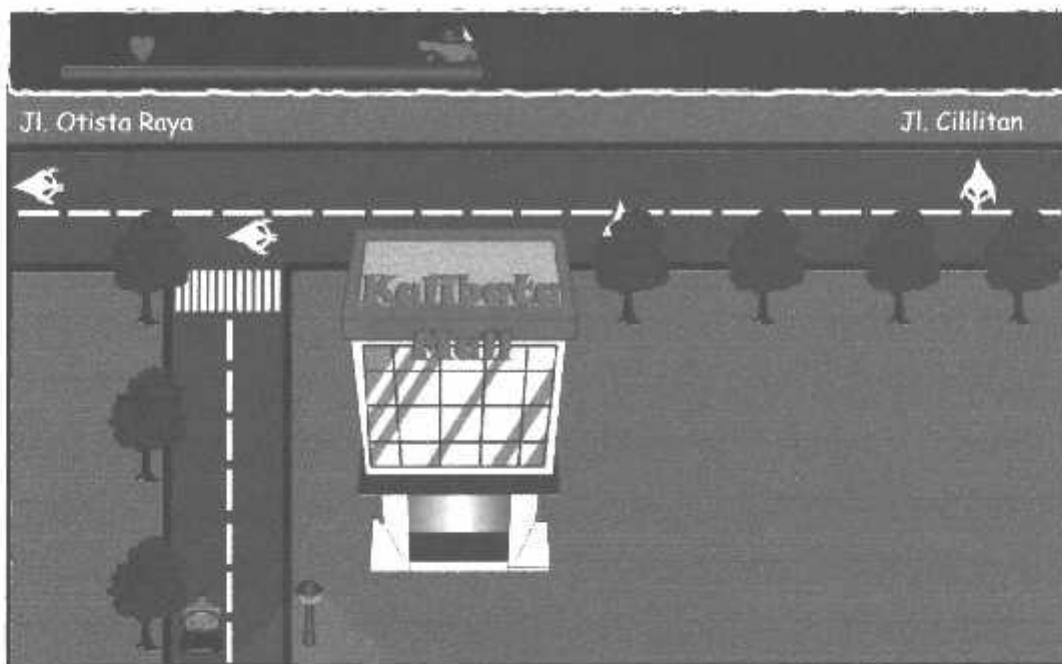
Gambar. 4.13 Tampilan arena.5



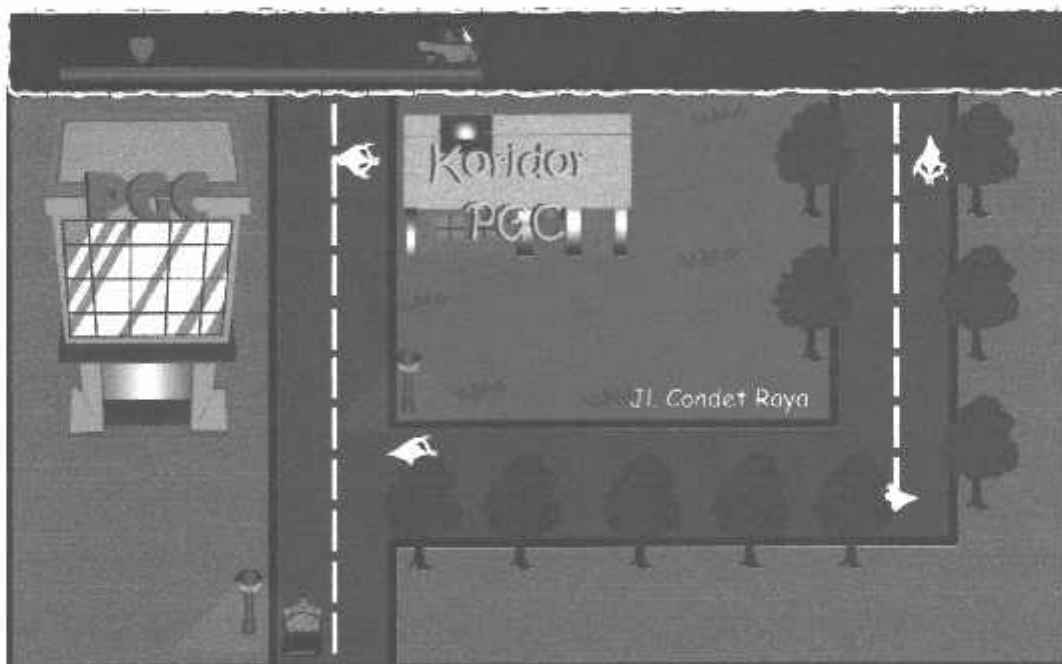
Gambar. 4.14 Tampilan arena Boss.2



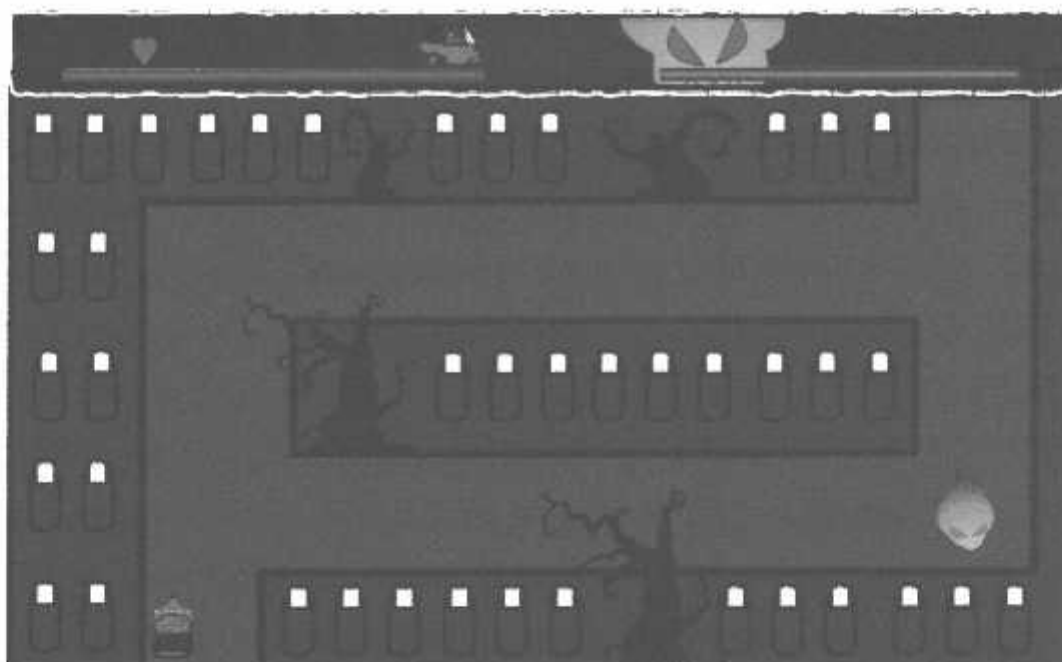
Gambar. 4.15 Tampilan arena.6



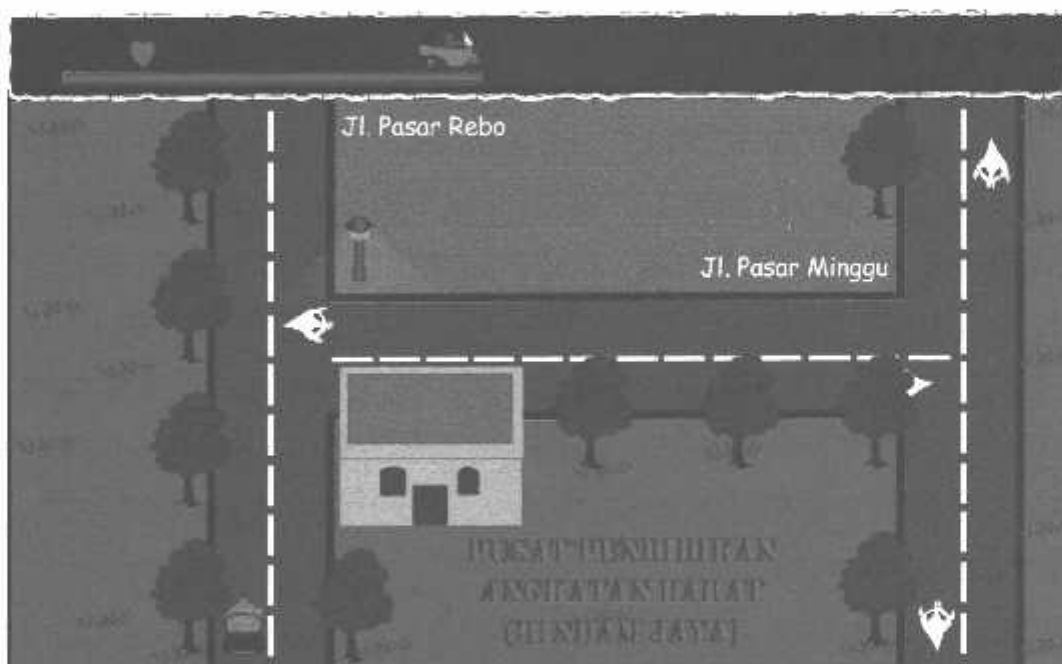
Gambar. 4.16 Tampilan arena.7



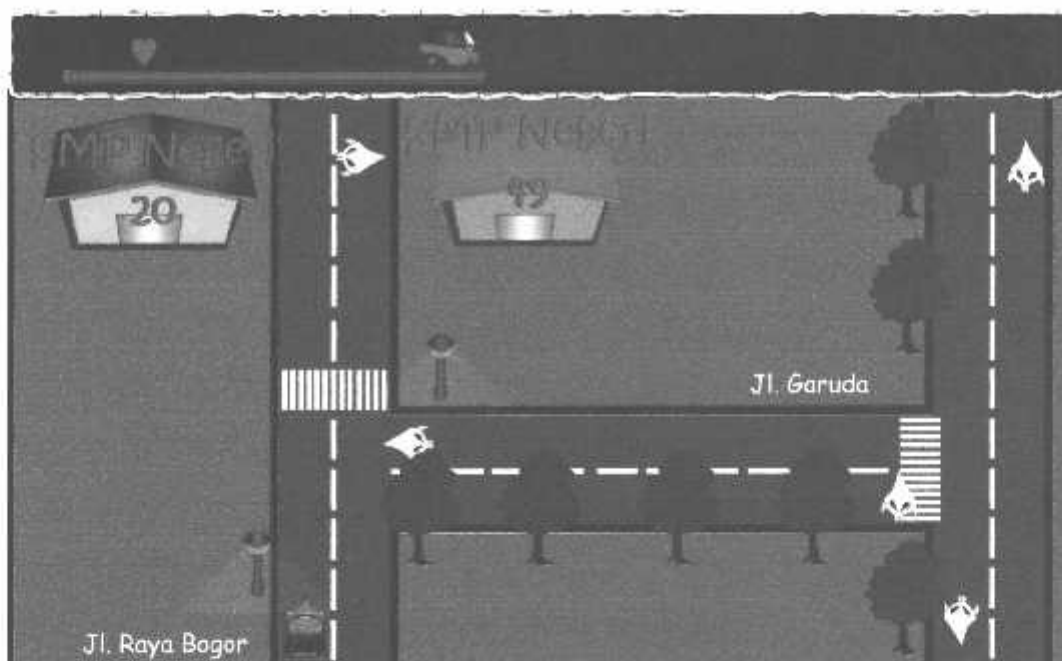
Gambar. 4.17 Tampilan arena.8



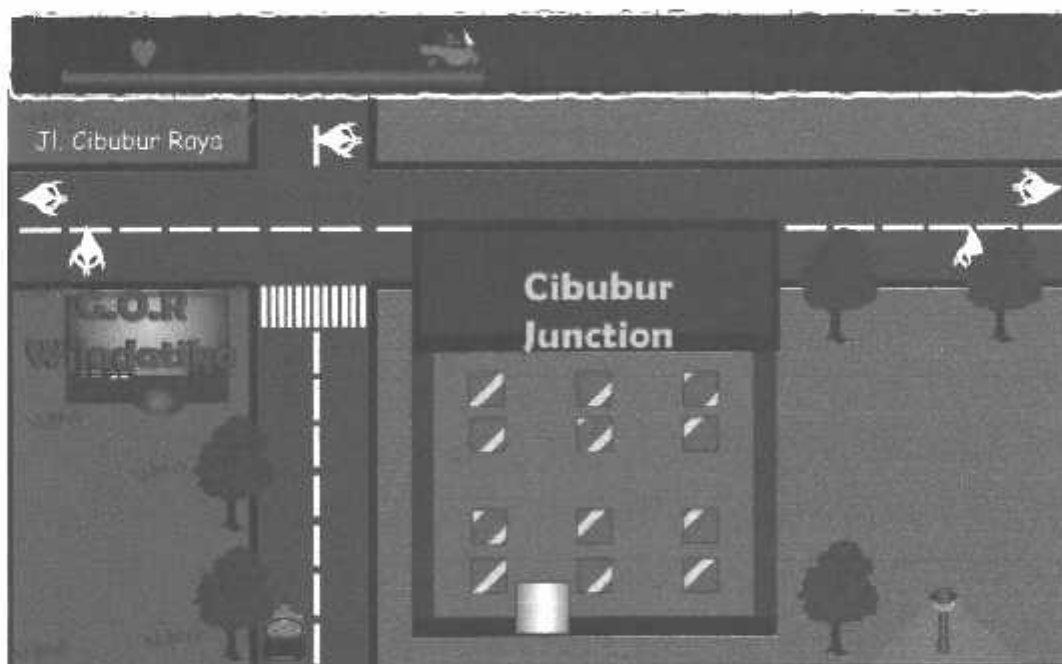
Gambar. 4.18 Tampilan arena Boss.3



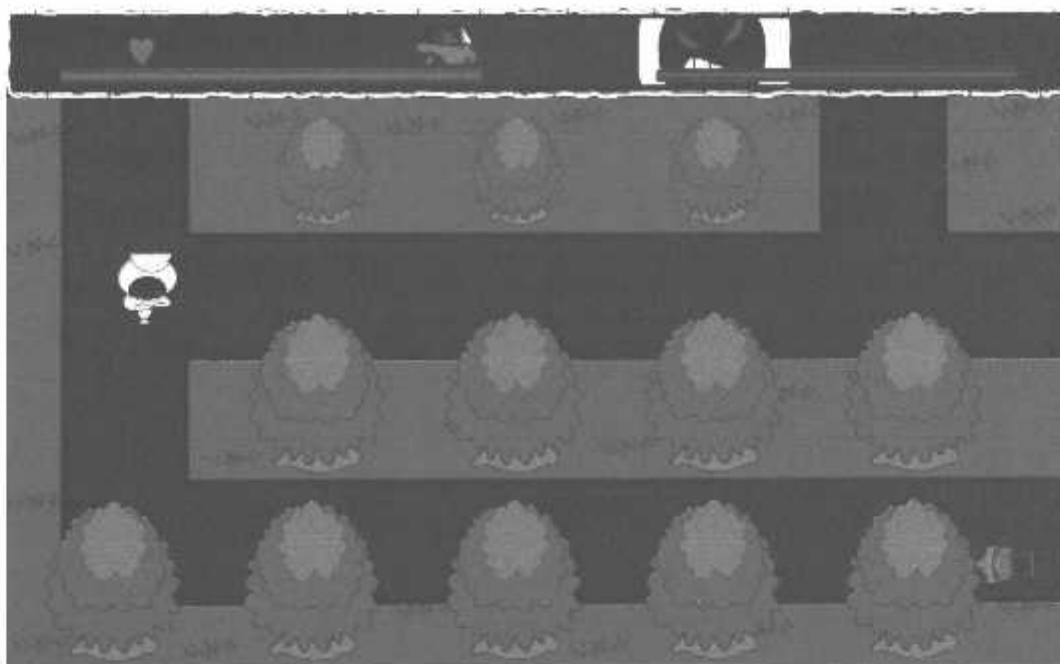
Gambar. 4.19 Tampilan arena.9



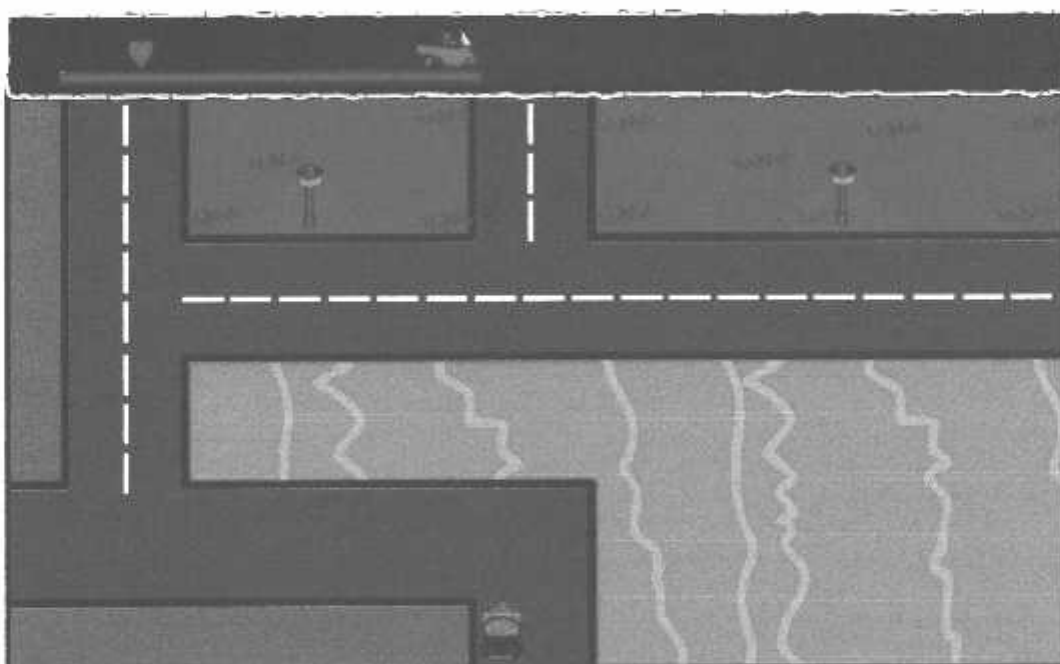
Gambar. 4.20 Tampilan arena.10



Gambar. 4.21 Tampilan arena.11



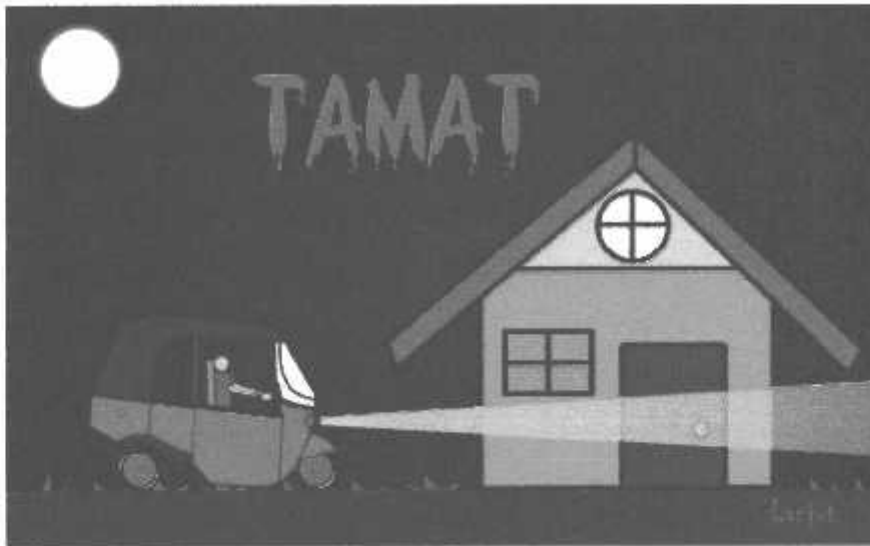
Gambar. 4.22 Tampilan arena Boss.4



Gambar. 4.23 Tampilan arena.12

4.1.6 Implementasi Tampilan Ending

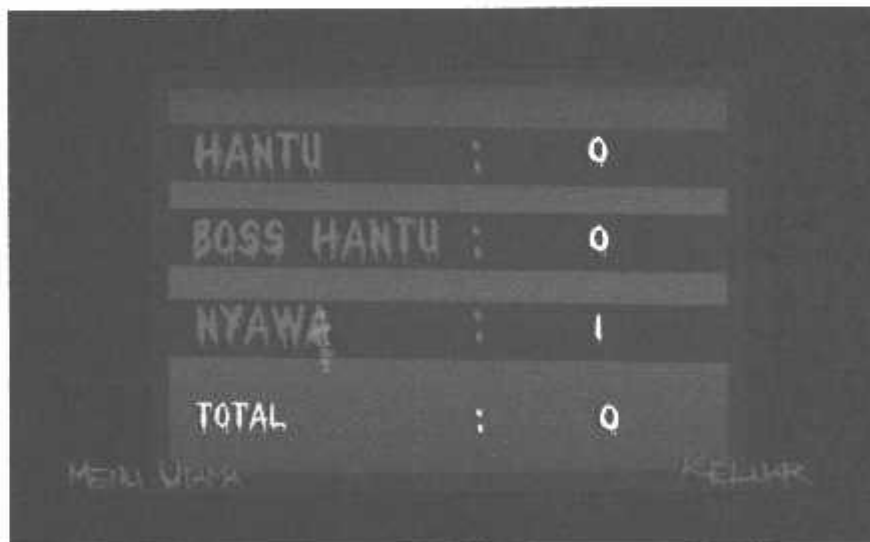
Ending adalah adegan yang muncul sebagai pertanda berakhirnya *game*. Seperti pada gambar 4.24 tampilan *ending* pada *game* Malam Jum'at Kliwon.



Gambar. 4.24 Tampilan Ending

4.1.7 Implementasi Tampilan Score

Tampilan ini menampilkan akumulasi *score* pemain, seperti pada gambar 4.25. Kolom Hantu menampilkan akumulasi dari 10*point*/Hantu Anak Buah, kolom Boss Hantu menampilkan akumulasi dari 1000*point*/Hantu Boss, kolom Nyawa menampilkan Jumlah Nyawa pemain yang tersisa, dan kolom Total untuk menampilkan akumulasi dari total *score* pemain.



Gambar. 4.25 Tampilan total score

4.1.8 Implementasi Tampilan Game Over

Halaman *game over* muncul ketika pemain mati. Tampilan ini pertanda bahwa, pemain gagal dalam menyelesaikan *game*. Tampilan *game over* terlihat pada gambar 4.26.



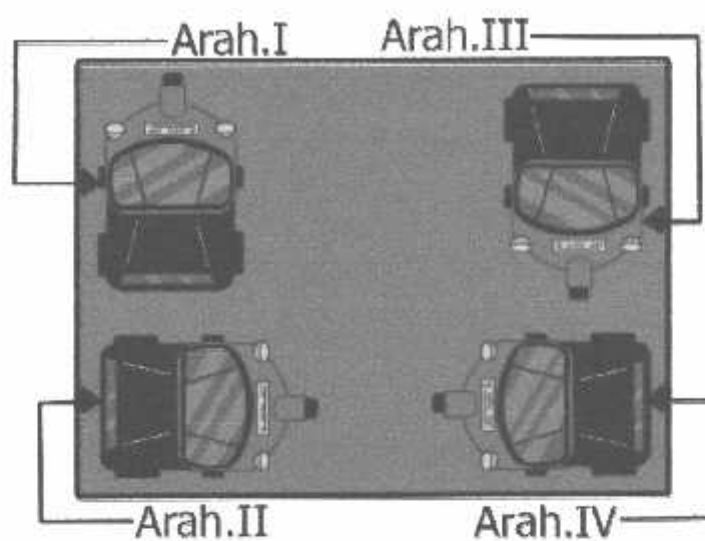
Gambar. 4.26 Tampilan game over

4.2. Pengujian

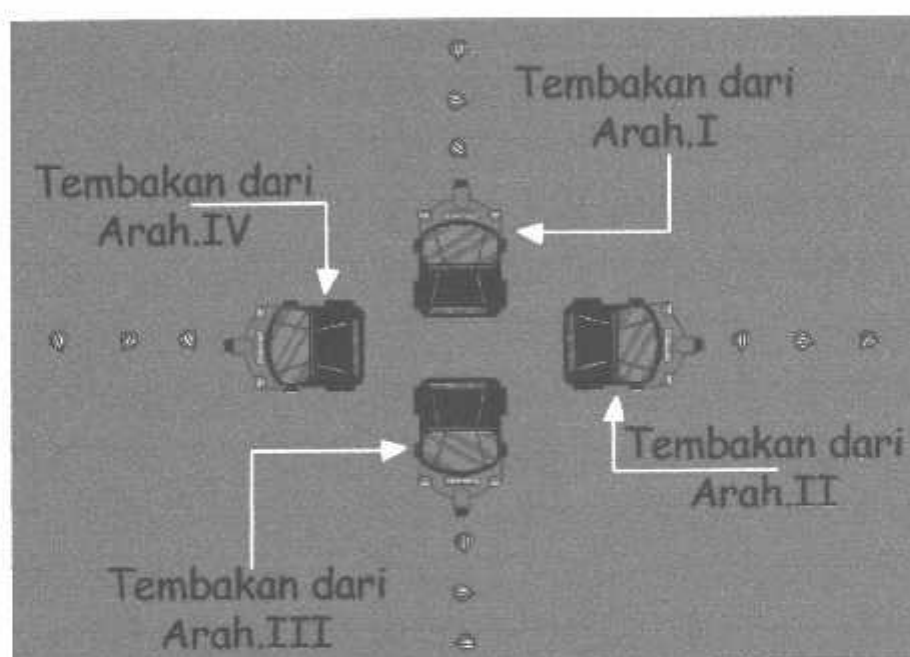
Pengujian adalah tahapan untuk melihat hasil kesesuaian dari kebutuhan fungsional dan kebutuhan non fungsional pada implementasi *game* Malam Jum'at Kliwon apakah sesuai dengan yang diharapkan.

4.2.1 Pengujian Kontrol

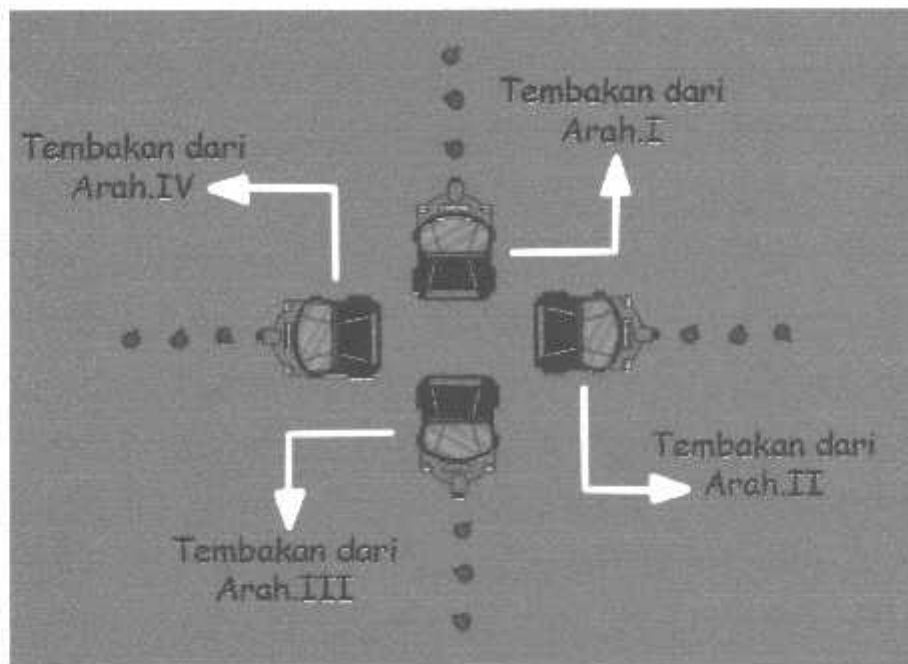
Uji kontrol karakter, digunakan *key* arah panah, yaitu arah atas, arah bawah, arah kanan, dan arah kiri. Setiap arah diberikan indikator nilai seperti yang terlihat pada gambar gambar 4.27 tentang arah bajaj. Selain itu, *key* A diperuntukan untuk menembak ke arah musuh dengan amunisi bawang putih seperti yang terlihat pada gambar 4.28 dan *key* S diperuntukan untuk menembak ke arah musuh dengan amunisi bawang merah seperti yang terlihat pada gambar 4.29. Untuk *key spacebar* diperuntukan untuk *pause* seperti pada gambar 4.30.



Gambar. 4.27 Pengujian kontrol arah pemain



Gambar. 4.28 Pengujian kontrol tembakan bawang putih



Gambar. 4.29 Pengujian kontrol tembakan bawang merah



Gambar. 4.30 Pengujian button pause

Pada tampilan *pause* terdapat 3 *button*, yaitu *button* lanjut untuk kembali melanjutkan permainan, *button* menu utama untuk kembali ke menu utama, dan *button* keluar untuk keluar dari permainan ini.

4.2.2 Pengujian Kecerdasan Hantu Anak Buah

Pengujian kecerdasan Hantu Anak Buah dilakukan oleh penulis. Dengan melakukan permainan sebanyak 10 kali, kemudian penulis menghimpun aksi yang dilakukan oleh pemain beserta reaksi yang dilakukan oleh Hantu Anak Buah. Hasil dari pengujian ini berdasarkan penilaian sesuai (√) atau tidak sesuai (-) beserta keterangannya. Hasil dari pengujian kecerdasan Hantu Anak Buah dapat dilihat pada tabel 4.1.

Tabel 4.1. Tabel Pengujian Kecerdasan Hantu Anak Buah

No	Aksi Pemain	Reaksi Hantu Anak Buah	Hasil	Keterangan
1	Pemain bersembunyi.	Mencari pemain	sesuai (√)	Keberadaan pemain tidak diketahui Hantu Anak Buah.
2	Pemain berada pada arah pandang Hantu Anak Buah.	Mengejar pemain	sesuai (√)	Jika ada ruang gerak bagi Hantu Anak Buah.
3	Pemain berada pada arah pandang Hantu Anak Buah.	Diam	sesuai (√)	Jika tidak ada ruang gerak bagi Hantu Anak Buah.
4	Pemain menembak pada arah pandang Hantu Anak Buah.	Menghindar	sesuai (√)	Hantu Anak Buah melihat laju tembakan dari pemain.
5	Pemain menembak dari arah belakang Hantu Anak Buah.	Mati	sesuai (√)	Hantu Anak Buah tidak melihat laju tembakan dari pemain.

4.2.3 Pengujian Kecerdasan Hantu Boss

Pengujian kecerdasan Hantu Boss dilakukan oleh penulis. Dengan melakukan permainan sebanyak 10 kali, kemudian penulis menghimpun aksi yang dilakukan oleh pemain beserta reaksi yang dilakukan oleh Hantu Boss. Hasil dari pengujian ini berdasarkan penilaian sesuai (√) atau tidak sesuai (-) beserta keterangannya. Hasil dari pengujian kecerdasan Hantu Boss dapat dilihat pada tabel 4.2.

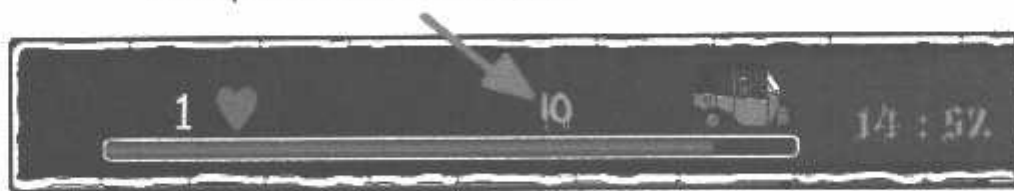
Tabel 4.2. Tabel Pengujian Kecerdasan Hantu Boss

No	Aksi Pemain	Reaksi Hantu Boss	Hasil	Keterangan
1	Pemain bersembunyi.	Mencari pemain	sesuai (√)	Keberadaan pemain tidak diketahui Hantu Boss.
2	Pemain berada pada arah pandang Hantu Boss.	Mengejar & Menembak pemain	sesuai (√)	Jika ada ruang gerak bagi Hantu Boss.
3	Pemain berada pada arah pandang Hantu Boss.	Menembak pemain	sesuai (√)	Jika tidak ada ruang gerak bagi Hantu Boss.
4	Pemain menembak pada arah pandang Hantu Boss.	Menghindar	sesuai (√)	Hantu Boss melihat laju tembakan dari pemain.
5	Pemain menembak dari arah belakang Hantu Boss.	Kesehatan pada <i>healthbar</i> Hantu Boss berkurang	sesuai (√)	Hantu Boss tidak melihat laju tembakan dari pemain.
6	Tembakan pemain mengenai Hantu Boss, saat darah Hantu Boss = 1.	Mati	sesuai (√)	Jika kesehatan Hantu Boss pada <i>healthbar</i> <1 atau = 0, maka Hantu Boss mati.

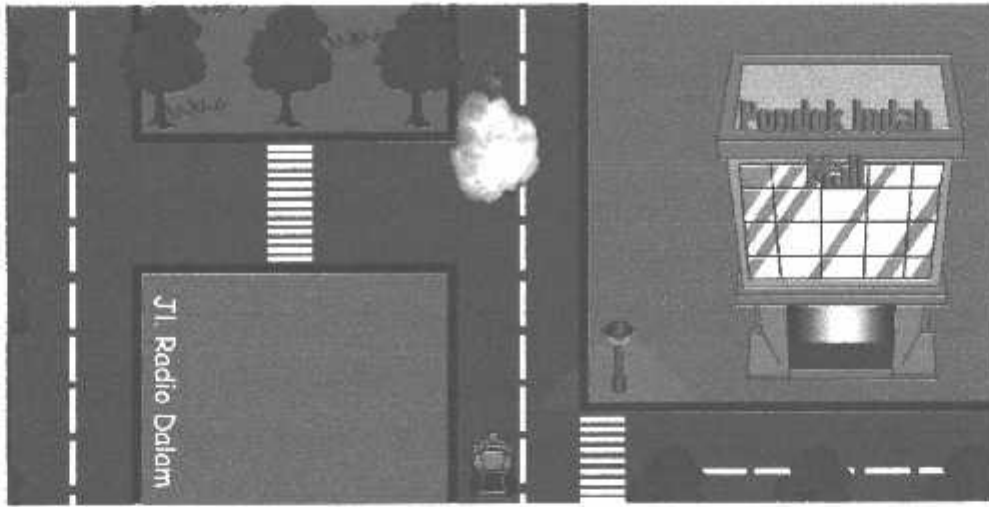
4.2.4 Pengujian Nilai Pemain

Score pemain akan bertambah 10*point* seperti pada gambar 4.31, untuk setiap Hantu Anak Buah yang mati seperti pada gambar 4.32. Untuk setiap Hantu Boss yang mati seperti pada gambar 4.33, maka *score* pemain akan bertambah 1000*point* dan pemain mendapatkan bonus 1nyawa seperti pada gambar 4.34.

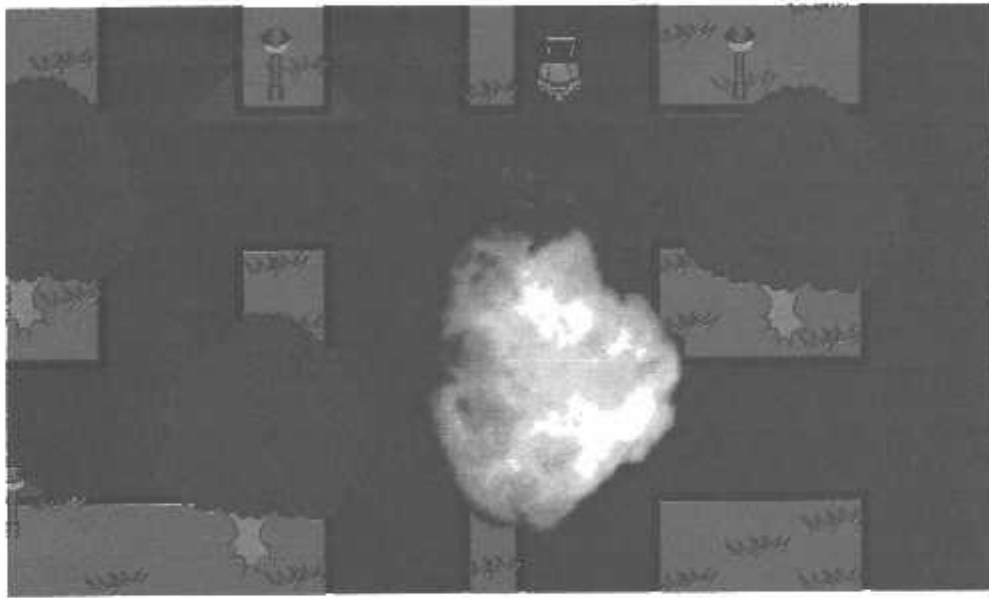
Score pemain bertambah 10 point



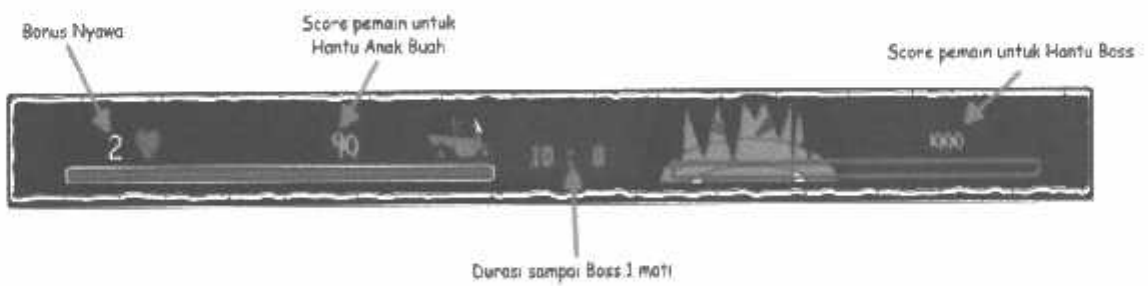
Gambar. 4.31 Score pemain untuk Hantu Anak Buah



Gambar. 4.32 Hantu Anak Buah mati

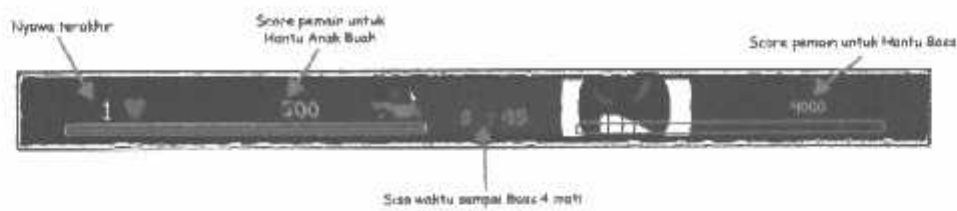


Gambar. 4.33 Hantu Boss mati

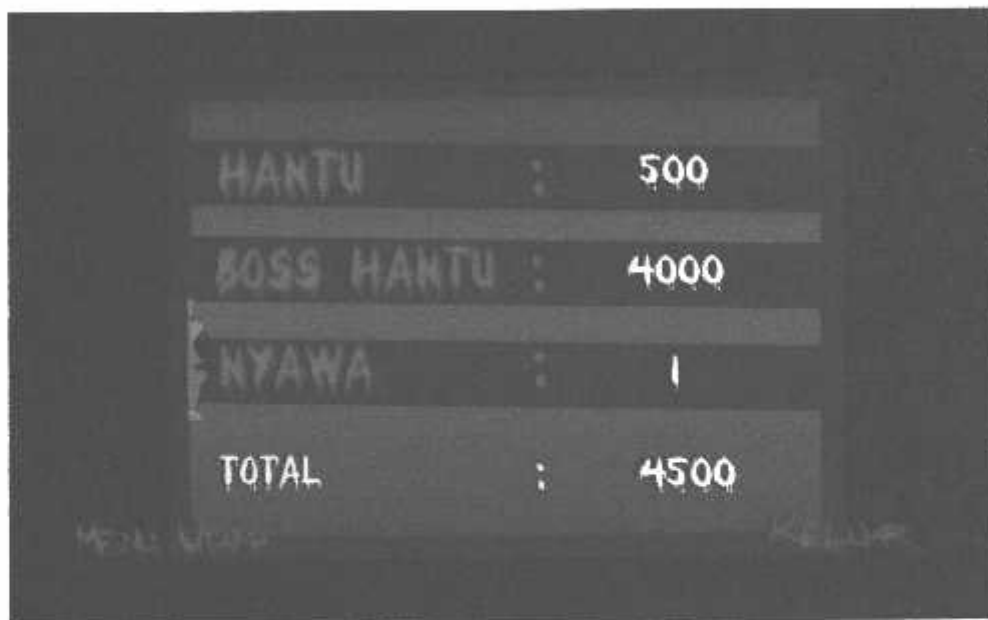


Gambar. 4.34 Score pemain untuk Hantu Boss

Setelah semua hantu mati, maka nilai akhir pemain terlihat seperti pada gambar 4.35, kemudian diakumulasikan dan ditampilkan pada halaman *score* seperti pada gambar 4.36.



Gambar. 4.35 Score akhir pemain



Gambar. 4.36 Akumulasi score akhir pemain

4.2.5 Pengujian Kebutuhan Fungsional

Game yang telah dibuat kemudian diujikan kepada *user* untuk dimainkan. Selanjutnya *user* diminta untuk memberikan penilaian kesesuaian dan komentar *user* terhadap *game* ini. Pengujian dilakukan kepada 10 orang *user* dan penilaian dilakukan dengan cara melakukan pengisian kuesioner oleh masing-masing *user*. Untuk hasil kesesuaian dari setiap *user*, maka akan diberi point 10%. Kuesioner berisi tujuh pertanyaan seputar kebutuhan fungsional (kemampuan musuh) Hantu Anak Buah dan Hantu Boss dengan penilaian sesuai (✓) atau tidak sesuai (-) menurut 10 orang *user*. Hasil dari pengujian *user* tersebut dikalkulasikan ke dalam bentuk persentase. Hasil dari pengujian *user* terhadap kebutuhan fungsional Hantu Anak Buah secara persentase

dapat dilihat pada tabel 4.3 dan untuk hasil dari pengujian *user* terhadap kebutuhan fungsional Hantu Boss secara persentase dapat dilihat pada tabel 4.4.

Tabel 4.3. Tabel Pengujian Kebutuhan Fungsional Hantu Anak Buah

No	Fungsi	Hasil
1	Terdapat dua macam aksi awal Hantu Anak Buah dalam mencari pemain, yaitu ada yang berjalan secara <i>vertical</i> dan ada yang berjalan <i>horizontal</i> .	100 %
2	Jika Hantu Anak Buah belum melihat pemain, maka Hantu Anak Buah akan tetap pada aksi mencari pemain.	100 %
3	Jika Hantu Anak Buah melihat posisi pemain, maka Hantu Anak Buah akan melakukan aksi mengejar pemain.	100 %
4	Ketika pemain melakukan aksi menembak. Jika Hantu Anak Buah melihat laju peluru pemain, maka Hantu Anak Buah dapat menghindarinya. Namun, jika Hantu Anak Buah tidak melihat laju tembakan tersebut maka ia akan mati.	100 %
5	Jika dalam pengejaran, pemain berhasil lolos dari pandangan Hantu Anak Buah. Maka, Hantu Anak Buah akan kembali mencari pemain.	100 %
6	Jumlah Hantu Anak Buah > 1.	100 %
7	Jika jumlah Hantu Anak Buah = 0 (mati semua), maka muncul animasi panah untuk pemain melanjutkan ke arena selanjutnya.	100 %
Total kesesuaian menurut user		100 %

Tabel 4.4. Tabel Pengujian Kebutuhan Fungsional Hantu Boss

No	Fungsi	Hasil
1	Aksi awal dalam mencari pemain, yaitu dengan berjalan secara <i>horizontal</i> .	100 %
2	Untuk Hantu Boss terdapat <i>healthbar</i> sebagai indikator nyawa.	100 %
3	Jika Hantu Boss belum melihat pemain, maka Hantu Boss tetap pada aksi mencari pemain.	100 %
4	Jika Hantu Boss melihat posisi pemain, maka Hantu Boss akan melakukan aksi mengejar dan menembak pemain.	100 %

5	Ketika pemain melakukan aksi menembak. Jika Hantu Boss melihat laju peluru pemain, maka Hantu Boss akan menghindarinya. Namun, jika Hantu Boss tidak melihat laju serangan tersebut maka kesehatan pada <i>healthbar</i> Hantu Boss akan berkurang.	100 %
6	Jika dalam pengejaran, pemain berhasil lolos dari pandangan Hantu Boss. Maka, Hantu Boss akan kembali mencari pemain.	100 %
7	Jika kesehatan pada <i>healthbar</i> Hantu Boss = 0, maka Hantu Boss mati dan akan muncul animasi arah panah untuk pemain melanjutkan ke tempat selanjutnya.	100 %
Total kesesuaian menurut user		100 %

4.2.6 Pengujian Kebutuhan Non Fungsional

Pengujian kebutuhan non fungsional dilakukan oleh penulis. Pengujian seputar kebutuhan non fungsional secara *performance* dan secara *interface* dilakukan untuk mendapatkan kesesuaian berdasarkan tahap perancangan. Penilaian dengan memberikan pernyataan sesuai (✓) atau tidak sesuai (-) beserta keterangannya. Setelah *game* Malam Jum'at Kliwon selesai dibuat, maka diperoleh hasil ukuran *file*, yaitu *file.Fla* berukuran 402MB, *file.Exe* berukuran 4,61MB, dan *file.Swf* berukuran 2,27MB. Hasil dari pengujian kebutuhan non fungsional secara *performance* dapat dilihat pada tabel 4.5, pengujian terhadap kebutuhan non fungsional secara *interface* pada *Acer Aspire 4530* dapat dilihat pada tabel 4.6, sedangkan untuk pengujian pada *Dell N 4050* dapat dilihat pada tabel 4.7, untuk pengujian pada *Acer Aspire 4738G* dapat dilihat pada tabel 4.8, untuk pengujian pada *Acer Aspire One AOA 150* dapat dilihat pada tabel 4.9, untuk pengujian pada *Macbook Pro 13"* dapat dilihat pada tabel 4.10, dan untuk pengujian pada *Personal Computer* dapat dilihat pada tabel 4.11.

Tabel 4.5. Tabel Pengujian Kebutuhan Non Fungsional Secara Performance

No	Fungsi	Hasil	Keterangan
1	Dalam pembuatan <i>game</i> Malam Jum'at Kliwon menggunakan 35fps.	sesuai (√)	<i>Frame/second.</i>
2	Dalam setiap hantu Terdapat tiga sensor, sensor untuk mengenali pemain, sensor untuk mengenali peluru pemain, dan sensor untuk mengenali sasaran (<i>hit</i>).	sesuai (√)	<ul style="list-style-type: none"> • Sensor pemain berfungsi sebagai acuan mengejar /menembak pemain. • Sensor peluru berfungsi sebagai acuan dalam menghindari serangan pemain. • Sensor hit untuk menyatakan kena/tidaknya musuh terhadap serangan dari pemain.
3	Pergerakan pemain dalam <i>game</i> sejauh 25pixel.	sesuai (√)	Saat ditekan <i>key</i> arah panah.
4	Pergerakan laju peluru bawang putih dalam <i>game</i> sejauh 25pixel.	sesuai (√)	Setiap peluru yang keluar.
5	Pergerakan laju peluru bawang merah sejauh 50pixel.	sesuai (√)	Setiap peluru yang keluar.
6	Untuk senjata bawang putih, jika mengenai musuh maka akan ikut meledak bersama musuh. Untuk senjata bawang merah, jika mengenai musuh tidak akan ikut meledak bersama musuh. Namun, untuk bawang merah hanya dapat digunakan 1kali di setiap arena.	sesuai (√)	<ul style="list-style-type: none"> • Senjata bawang putih bersifat tidak terbatas jumlahnya pada setiap arena. • Senjata bawang merah bersifat terbatas hanya 1kali pada setiap arena.
7	Pergerakan Hantu Anak Buah dalam mengejar pemain sejauh 25pixel.	sesuai (√)	Aksi mengejar dilakukan ketika melihat keberadaan pemain.

8	Pergerakan Hantu Anak Buah dalam menghindari serangan pemain sejauh $20pixel$.	sesuai (√)	Aksi menghindar dilakukan ketika melihat laju dari tembakan pemain.
9	Pergerakan Hantu Boss dalam mengejar pemain sejauh $30pixel$.	sesuai (√)	Aksi mengejar dilakukan ketika melihat keberadaan pemain.
10	Pergerakan Hantu Boss dalam menghindari serangan pemain sejauh $20pixel$.	sesuai (√)	Aksi menghindar dilakukan ketika melihat laju dari tembakan pemain.
11	Pergerakan laju peluru Hantu Boss sejauh $40pixel$.	sesuai (√)	Setiap peluru Boss yang keluar.
12	Pemain dan Hantu Boss menggunakan <i>healthbar</i> sebagai indikator nyawa.	sesuai (√)	Kesehatan pada <i>healthbar</i> akan berkurang apabila terkena serangan lawan.
13	Panjang sensor musuh yang digunakan untuk mengenali objek pemain sepanjang $5000pixel$.	sesuai (√)	Dinyatakan pada <i>scene.1</i> .
14	Panjang sensor musuh yang digunakan untuk mengenali objek peluru pemain sepanjang $3000pixel$.	sesuai (√)	Dinyatakan pada <i>scene.1</i> .
15	Kesehatan pemain pada <i>healthbar</i> akan berkurang apabila bersentuhan dengan setiap musuh ataupun terkena peluru Hantu Boss.	sesuai (√)	Bila kesehatan pemain pada <i>healthbar</i> = 0, maka pemain mati atau <i>game over</i> .
16	Kesehatan Hantu Boss pada <i>healthbar</i> akan bertambah apabila bersentuhan dengan pemain.	sesuai (√)	Bila kesehatan Hantu Boss = 100, maka Hantu Boss <i>fit</i> kembali dan harus ditembak dari awal sampai kesehatan Hantu Boss = 0, untuk lolos ke tempat berikutnya.
17	Durasi pada game sepanjang 15menit (waktu mundur). Apabila waktu yang ditentukan telah habis dan permainan belum selesai, maka pemain dinyatakan mati (<i>game over</i>).	sesuai (√)	<i>Countdown</i> .

Tabel 4.6. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Acer Aspire 4530

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yang digunakan dalam membangun <i>game</i> , yaitu sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game akan dibuat fullscreen sesuai spesifikasi 1280 x 800 <i>pixel</i> .
2	Spesifikasi <i>Processor</i> yang digunakan dalam membangun <i>game</i> , yaitu <i>AMD Turion X2 (64)</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> yang digunakan dalam membangun <i>game</i> , sebesar 1024 MB.	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yang digunakan dalam membangun <i>game</i> , yaitu <i>nVIDIA GeForce 9100M G</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan dalam membangun <i>game</i> adalah <i>Windows Xp.3 Dark Edition Rebirth Version</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Sesuai dengan spesifikasi <i>display</i> .
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawang putih dan <i>key S</i> sebagai senjata bawang merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	<i>Platform</i> yang digunakan dalam membangun <i>game</i> adalah <i>Adobe Flash CS3</i> dengan bahasa pemrograman <i>Action Script 2.0</i> .	sesuai (√)	Pemanfaatan dalam proses perancangan karakter, animasi, dan algoritma <i>FSM</i>
9	<i>Platform</i> yang digunakan dalam <i>editing audio</i> adalah <i>Adobe Premiere Pro CS3</i> .	sesuai (√)	Pemanfaatan dalam proses perancangan audio pada <i>game</i> .
10	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Adobe Flash Player</i> dan <i>Mozila FireFox browser</i> .
11	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.

Tabel 4.7. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Dell N 4050

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yaitu sebesar 1366 x 768 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
2	Spesifikasi <i>Processor</i> yaitu <i>Intel Core i5-2450M</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> sebesar 8 <i>GB</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yaitu <i>Intel HD 4000</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan adalah <i>Ubuntu 12.04</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawang putih dan <i>key S</i> sebagai senjata bawang merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Mozila FireFox browser</i> .
9	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada aplikasi <i>wine</i> .

Tabel 4.8. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Acer Aspire 4738G

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yaitu sebesar 1366 x 768 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
2	Spesifikasi <i>Processor</i> yaitu <i>Intel Core i3-380M</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> sebesar 2 <i>GB</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yaitu <i>AMD RADEON HD 6370 M</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan adalah <i>Windows 7 Enternity</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawah putih dan <i>key S</i> sebagai senjata bawah merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Adobe Flash Player</i> dan <i>Mozilla FireFox browser</i> .
9	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.

Tabel 4.9. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Acer Aspire One AOA 150

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yaitu sebesar 1024 x 600 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
2	Spesifikasi <i>Processor</i> yaitu <i>Intel Atom Processor N270</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> sebesar 512 <i>MB</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yaitu <i>Mobile Intel (R) 945 Express Chipset Family</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan adalah <i>Windows XP Sp2 Professional</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawang putih dan <i>key S</i> sebagai senjata bawang merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Adobe Flash Player</i> dan <i>Mozilla FireFox browser</i> .
9	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.

Tabel 4.10. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Macbook Pro 13"

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yaitu sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
2	Spesifikasi <i>Processor</i> yaitu <i>Intel Core i5 with L3 cache</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> sebesar 4 <i>GB</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yaitu <i>Intel HD Graphic 3000</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan adalah <i>Mac Mountain Lion update 10.8.2</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawang putih dan <i>key S</i> sebagai senjata bawang merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Mozila FireFox browser</i> dan <i>Safari 6.0.1 browser</i> .
9	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada aplikasi <i>wine</i> dan <i>crossover</i> .

Tabel 4.11. Tabel Pengujian Kebutuhan Non Fungsional Interface Pada Personal Computer

No	Fungsi	Hasil	Keterangan
1	Spesifikasi <i>display</i> yaitu sebesar 1366 x 768 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
2	Spesifikasi <i>Processor</i> yaitu <i>AMD phenom (tm) II x4955</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> sebesar 4 <i>GB</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yaitu <i>ATI Radeon HD 5700 series</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
5	Sistem operasi yang digunakan adalah <i>Windows 8</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
6	<i>Size game</i> yang digunakan sebesar 1280 x 800 <i>pixel</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key A</i> sebagai senjata bawang putih dan <i>key S</i> sebagai senjata bawang merah. Sedangkan, <i>key space</i> digunakan sebagai <i>pause</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
8	Pengujian <i>file.Swf</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Adobe Flash Player</i> dan <i>Mozilla FireFox browser</i> .
9	Pengujian <i>file.Exe</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.

BAB V

PENUTUP

5.1. Kesimpulan

Kesimpulan yang diperoleh dalam Implementasi Finite State Machine Pada Game Malam Jum'at Kliwon, meliputi :

1. Berdasarkan hasil pengujian menggunakan algoritma *finite state machine*, musuh dapat memberikan respon terhadap pemain berdasarkan sensor pandangan dan sensor hindar. Dari tingkat keberhasilan yang diperoleh adalah 100% sesuai tabel pengujian kecerdasan Hantu Anak Buah dan tabel pengujian kecerdasan Hantu Boss.
2. Algoritma *finite state machine* dapat diterapkan untuk menghasilkan perilaku yang bervariasi pada setiap musuh. Sesuai hasil pengujian dari tabel kecerdasan Hantu Anak Buah dan tabel kecerdasan Hantu Boss ditambah hasil pengujian dari tabel kebutuhan fungsional Hantu Anak Buah dan tabel kebutuhan fungsional Hantu Boss.
3. Hasil *publish* pada game Malam Jum'at Kliwon yaitu berupa *file.Exe* dengan *size* 4,61MB dan *file.Swf* dengan *size* 2,27MB. Semua *file* dari hasil *publish* dapat berjalan normal sesuai fungsional pada *laptop* maupun *personal computer* dengan spesifikasi *hardware* dan sistem operasi yang berbeda berdasarkan tabel pengujian non fungsional *interface*.
4. Pada *file.exe*, mampu dieksekusi tanpa aplikasi tambahan untuk sistem operasi yang berbasis *windows*. Sedangkan, untuk sistem operasi *Ubuntu* dan *Mac* harus menggunakan aplikasi *wine* untuk mengeksekusi *file.exe*. Untuk *file.swf* dapat dieksekusi pada *mozilla browser* untuk setiap jenis *laptop* maupun *personal computer* dengan spesifikasi *hardware* dan sistem operasi yang berbeda berdasarkan tabel pengujian non fungsional *interface*.

5.2. Saran

Saran yang dapat diberikan pada tugas akhir Implementasi Finite State Machine Pada Game Malam Jum'at Kliwon, meliputi :

1. Kedepannya pada game ini terdapat *high score* untuk pemain.
 2. Kedepannya pada game ini terdapat fungsi *save* untuk pemain.
 3. Kedepannya pada game ini terdapat variasi senjata yang lebih banyak untuk pemain.
-

DAFTAR PUSTAKA

1. Agustinus Nilwan, 2012. *Definisi game*.
 URL : <http://uwiesulistya.blogspot.com/2012/03/definisi-game.html>
 (diakses tanggal 10 september 2012)
 2. Amrullah, 2008. *Sekilas Tentang Flash*, Universitas Sumatera Utara, Medan.
 URL : <http://repository.usu.ac.id/bitstream/123456789/27967/4/Chapter%20II.pdf>
 (diakses tanggal 20 Februari 2013)
 3. Anggra, 2008. *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*, Gava Media, Jogjakarta.
 4. Anisahicha, 2011. *Sejarah Game*.
 URL : <http://anisahicha.wordpress.com/2011/10/30/sejarah-game/>
 (diakses tanggal 20 September 2012)
 5. David M. Bourg and Glenn Seeman, 2004. *AI for Game Developers*, O'Reilly Media, USA.
 6. Setiawan, 2006. *Pengertian Finite State Machine*, Universitas Sumatera Utara, Medan.
 URL :
http://www.google.com/url?sa=t&rct=j&q=pengertian%20fsm&source=web&cd=1&cad=rja&ved=0CDEQFjAA&url=http%3A%2F%2Frepository.usu.ac.id%2Fbitstream%2F123456789%2F24728%2F3%2FChapter%2520II.pdf&ci=VKkTUaWEYWKrgfJIYDgCA&usq=AFQjCNFw9DUjCagKWisuxsd9WQ_AoUmtVQ&bv m=by.42080656,d.bmk
 (diakses tanggal 20 Oktober 2012)
 7. Suyanto, 2011. *Artificial Intelligence*, INFORMATIKA, Bandung.
 8. Taylor and Francis Group, 2006. *Modeling Software With Finite State Machines : A Practical Approach*, A Uerbach Publication, New York.
 9. Wandah, 2008. *Dasar Pemrograman Flash Game*, Gava Media, Jogjakarta.
 10. Warung Flash, 2009. *Fscommand*.
 URL: <http://warungflash.com/2009/02/fscommand/>
 (diakses tanggal 20 September 2012)
-

11. Zen, Revaldo, 2008. *Pemodelan Sederhana*, FMIPA UI, Depok.

URL :

http://www.google.co.id/url?sa=t&rct=j&q=buku%20teori%20%20finite%20state%20machine&source=web&cd=1&cad=rja&ved=0CB4QFjAA&url=http%3A%2F%2Fwww.lontar.ui.ac.id%2Ffile%3Ffile%3Ddigital%2F126504-MAT.023-08-Pemodelan%2520sederhana-Literatur.pdf&ei=_utVUKXkFlrzQer1IGwDQ&usg=AFQjCNHCVidttVVpUHgHcLkhl-zKy7xPDQ

(diakses tanggal 16 September 2012)

LAMPIRAN



INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNT (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : J. Dendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : J. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

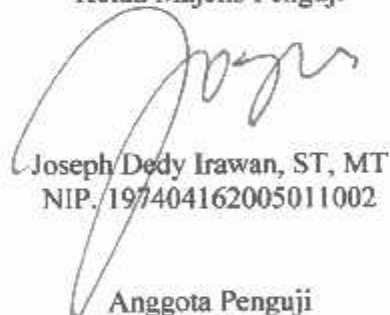
BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama : Widi Nugroho
NIM : 08.18.143
Jurusan : Teknik Informatika S-1
Judul Skripsi : **IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Senin
Tanggal : 18 Februari 2013
Nilai : 89,53 (A)

Panitia Ujian Skripsi
Ketua Majelis Penguji



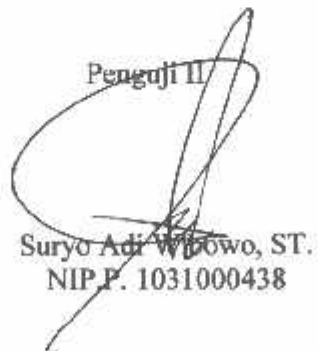
Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002
Anggota Penguji

Penguji I



Michael Ardita, ST, MT.
NIP.P. 1031000434

Penguji II



Suryo Adi Wibowo, ST.
NIP.P. 1031000438



INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigora-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Widi Nugroho
NIM : 08.18.143
Jurusan : Teknik Informatika S-1
Judul Skripsi : IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON

TANGGAL	PENGUJI	URAIAN	PARAF
18 Februari 2013	I	<ol style="list-style-type: none"> Memperbaiki Flowchart alur permainan pada BAB III. Menambah Flowchart FSM Hantu Anak Buah dan Flowchart Hantu Boss. Memperbaiki penulisan Daftar Pustaka. 	
	II	<ol style="list-style-type: none"> Menambah isi dari keterangan pada setiap tabel pengujian. Mencantumkan ukuran file game dengan .Exe dan .Swf dari hasil publish .Fla Menambah tabel hasil pengujian Non Fungsional Interface pada komputer/laptop dengan spesifikasi hardware dan sistem operasi yang berbeda. Menambah kesimpulan sesuai seluruh tabel hasil pengujian. 	

Anggota Penguji

Penguji I

Michael Ardita, ST, MT.
NIP.P.1031000434

Penguji II

Suryo Adi Wibowo, ST
NIP.P. 1031000438

Mengetahui,

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT.
NIP. Y. 1018800189

Dosen Pembimbing II

Febriana Santi Wahyuni, S.Kom, M.Kom.
NIP.P. 1031000425



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : J. Dendang Sigat-gara No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : J. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Widi Nugroho
NIM : 08.18.143
Masa Bimbingan : 17 Oktober 2012 s/d 17 April 2013
Judul Skripsi : IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	10 Januari 2013	Demo Game	
2	12 Januari 2013	Revisi Game	
3	19 Januari 2013	Pengajuan Laporan Skripsi	
4	22 Januari 2013	Revisi Laporan Skripsi	
5	31 Januari 2013	Pengajuan Makalah Seminar Hasil	
6	1 Februari 2013	Revisi Makalah Seminar Hasil	
7	12 Februari 2013	Revisi Laporan Skripsi untuk Ujian Komprehensif	
8			

Malang, 20 Maret 2013
Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT.
NIP. Y. 1018800189

**INSTITUT TEKNOLOGI NASIONAL
MALANG****FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK**PT. BNI (PERSERO) MALANG
BANK NIAGA MALANGKampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Kawarglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang**FORMULIR BIMBINGAN SKRIPSI**

Nama : Widi Nugroho
NIM : 08.18.143
Masa Bimbingan : 17 Oktober 2012 s/d 17 April 2013
Judul Skripsi : **IMPLEMENTASI FINITE STATE MACHINE PADA GAME
MALAM JUM'AT KLIWON**

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	17 Oktober 2012	Pengajuan Laporan Skripsi Bab.I-Bab.III	<i>toni</i>
2	20 Oktober 2012	Revisi Laporan Skripsi Bab.I-Bab.III	<i>toni</i>
3	10 Januari 2013	Demo Game	<i>toni</i>
4	12 Januari 2013	Pengajuan Laporan Skripsi Bab.IV-Bab.V	<i>toni</i>
5	15 Januari 2013	Revisi Laporan Skripsi Bab.IV-Bab.V	<i>toni</i>
6	26 Januari 2013	Pengajuan Makalah Seminar Hasil	<i>toni</i>
7	31 Januari 2013	Revisi Makalah Seminar Hasil	<i>toni</i>
8	12 Februari 2013	Revisi Laporan Skripsi untuk Ujian Komprehensif	<i>toni</i>

Malang, 20 Maret 2013
Dosen Pembimbing IIFebriana Santi Wahyuni, S.Kom, M.Kom.
NIP.P. 1031000425

Lampiran.4. Penggalan *actionsript* Hantu Anak Buah berjalan *vertical*

```
waktujeda3_hantu1--;  
hantu1._y=-kec_hantu1_x;  
  
if(hantu1._y<=batasatas)  
{  
    waktujeda3_hantu1=50;  
    hantu1._y = batasatas;  
    waktujeda1_hantu1--;  
  
    if(waktujeda1_hantu1<=1)  
    {  
        kec_hantu1_x =-kec_hantu1_x;  
        waktujeda1_hantu1 =20;  
        hantu1._rotation =-90;  
    }  
}  
  
if(hantu1._y>=batasbawah)  
{  
    waktujeda3_hantu1 =50;  
    hantu1._y=batasbawah;  
    waktujeda2_hantu1--;  
  
    if(waktujeda2_hantu1<=1)  
    {  
        waktujeda2_hantu1=30;  
        hantu1._rotation =90;  
        kec_hantu1_x =-kec_hantu1_x;  
    }  
}  
  
else if (waktujeda3_hantu1<1)  
{  
    waktujeda3_hantu1 = 50;  
    hantu1._rotation -= 90;  
}
```

Lampiran.5. Penggalan *actionsript* Hantu Anak Buah berjalan *horizontal*

```
waktujeda3_hantu4 -= 1;
hantu4._x += kec_hantu4x;

if (hantu4._x > bataskanan)
{
    hantu4._x = bataskanan;
    waktujeda1_hantu4 -= 1;
    waktujeda3_hantu4 = 30;

    if (waktujeda1_hantu4 <= 1)
    {
        waktujeda1_hantu4 = 10;
        hantu4._rotation = 90;
        hantu4._x -= kec_hantu4_x;
        kec_hantu4x = 0;
    }
}

hantu4._x -= kec_hantu4_x;

if (hantu4._x < bataskiri)
{
    hantu4._x = bataskiri;
    waktujeda2_hantu4 -= 1;
    waktujeda3_hantu4 = 30;

    if (waktujeda2_hantu4 <= 1)
    {
        waktujeda2_hantu4 = 20;
        hantu4._rotation = -90;
        hantu4._x -= kec_hantu4_x;
        kec_hantu4x = 10;
    }
}

else if (waktujeda3_hantu4 < 1)
{
    waktujeda3_hantu4 = 50;
    hantu4._rotation -= 90;
}
```

Lampiran.6. Penggalan *actionsript* Hantu Anak Buah mengejar pemain

```
if(hantu30.sensor.hitTest(bajaj))  
  
    {  
  
        if (this._x<bajaj._x)  
        {  
            this._x += kec;  
        }  
  
        if (this._x>bajaj._x)  
        {  
            this._x -= kec;  
        }  
  
        if (this._y<bajaj._y)  
        {  
            this._y += kec;  
        }  
  
        if (this._y>bajaj._y)  
        {  
            this._y -= kec;  
        }  
  
        else { hantu30.stop(); }  
  
    }
```


Lampiran.7. Penggalan *actionsript* Hantu Anak Buah menghindar

```
if(_root.bawang_lagi.hitTest(this.sensoor) || _root.bawang_merah.hitTest(this.sensoor))
{
    if(_root.arah == 1)
    {
        if(_root.bajaj._x > this._x)
        { this._x -= menghindar; }

        else
        { this._x += menghindar; }
    }

    else if(_root.arah == 3)
    {
        if(_root.bajaj._x > this._x)
        { this._x -= menghindar; }

        else
        { this._x += menghindar; }
    }

    else if(_root.arah == 2)
    {
        if(_root.bajaj._y > this._y)
        { this._y -= menghindar; }

        else
        { this._y += menghindar; }
    }

    else if(_root.arah == 4)
    {
        if(_root.bajaj._y > this._y)
        { this._y -= menghindar; }

        else
        { this._y += menghindar; }
    }
}
```

Lampiran.8. Penggalan *actionscript* Hantu Boss berjalan *horizontal*

```
waktujeda3_bos1 -= 1 ;
bos1._x += kec_bos1x ;

if (bos1._x > bataskanan)
{
    bos1._x = bataskanan ;
    waktujeda1_bos1 -= 1 ;
    waktujeda3_bos1 = 30 ;

    if (waktujeda1_bos1 <= 1)
    {
        waktujeda1_bos1 = 10 ;
        bos1._x -= kec_bos1_x ;
        kec_bos1x = 0 ;
    }
}

bos1._x -= kec_bos1_x ;

if (bos1._x < bataskiri)
{
    bos1._x = bataskiri ;
    waktujeda2_bos1 -= 1 ;
    waktujeda3_bos1 = 30 ;

    if (waktujeda2_bos1 <= 1)
    {
        waktujeda2_bos1 = 20 ;
        bos1._x -= kec_bos1_x ;
        kec_bos1x = 10 ;
    }
}

else if (waktujeda3_bos1 < 1)
{
    waktujeda3_bos1 = 60 ;
    bos1._rotation -= 90 ;

    _root.arah_bos1-- ;
    if (_root.arah_bos1 <= 0) _root.arah_bos1 = 4 ;
    trace(_root.arah_bos1) ;
}
}
```

Lampiran.9. Penggalan *actionsript* Hantu Boss mengejar dan menembak pemain

```
if (bos1.sensor.hitTest(bajaj))
{
    suara_bos1 = new Sound(); suara_bos1.attachSound("suarabos1"); suara_bos1.start();

    if (_root.bos1_lagi_nembak == false)
    {
        trace("nembak nihhhhhhhhh");

        _root.bos1_lagi_nembak = true;

        duplicateMovieClip
    ( root.amunisi_bos1, "amunisi_bos11", -16300);
    }

    if (this._x < bajaj._x)
    {
        this._x += kec;
    }

    if (this._x > bajaj._x)
    {
        this._x -= kec;
    }

    if (this._y < bajaj._y)
    {
        this._y += kec;
    }

    if (this._y > bajaj._y)
    {
        this._y -= kec;
    }

    else { bos1.stop(); }
}
```

Lampiran.10. Penggalan *actionscript* Hantu Boss menghindar

```
if(_root.bawang_lagi.hitTest(this.sensor) || _root.bawang_merah.hitTest(this.sensor) &&
!pause_game)
{
    if (_root.arah == 1)
    {
        if (_root.bajaj._x > this._x)
        { this._x -= menghindar; }

        else
        { this._x += menghindar; }
    }

    else if (_root.arah == 3)
    {
        if (_root.bajaj._x > this._x)
        { this._x -= menghindar; }

        else
        { this._x += menghindar; }
    }

    else if (_root.arah == 2)
    {
        if (_root.bajaj._y > this._y)
        { this._y -= menghindar; }
        else
        { this._y += menghindar; }
    }

    else if (_root.arah == 4)
    {
        if (_root.bajaj._y > this._y)
        { this._y -= menghindar; }

        else
        { this._y += menghindar; }
    }
}
```