

**GAME ENDLES RUNING SI KANCIL
MENGUNAKAN METODE FINITE STATE MACHINE**

SKRIPSI



**MILIK
PERPUSTAKAAN
ITN MALANG**

**Disusun Oleh :
Musyaffa Al - Hafi
13.18.109**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

2017

UNITED STATES DEPARTMENT OF AGRICULTURE
NATIONAL BUREAU OF ECONOMIC RESEARCH

EXHIBIT

MILK
PERPUSTAKAAN
TNI MALANG

1950
No. 1 - 1950
1950

THE NATIONAL BUREAU OF ECONOMIC RESEARCH
NATIONAL BUREAU OF ECONOMIC RESEARCH
NATIONAL BUREAU OF ECONOMIC RESEARCH
1950

LEMBAR PERSETUJUAN DAN PENGESAHAN
GAME ENDLES RUNING DENGAN MENGGUNAKAN
METODE FINITE STATE MACHINE

SKRIPSI

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

Disusun Oleh :

MUSYAFFA AL - HAFI

13.18.109

Diperiksa dan Disetujui,

Dosen Pembimbing I

Dosen Pembimbing II

Karina Auliasari, ST, M.Eng

NIP. P 1031000426

Rofila El Maghfiroh, S.Si.MSc

NIP.P 1031000505

Mengetahui,

Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT

NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2017

LEMBAR KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI

Sebagai mahasiswa Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang, yang bertanda tangan di bawah ini, saya:

Nama : Musyaffa Al – Hafi
NIM : 13.18.109
Program Studi : Teknik Informatika S-1
Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa skripsi saya dengan judul ” ***GAME ENDLES RUNING SI KANCIL MENGGUNAKAN METODE FINITE STATE MACHINE***” merupakan karya asli dan bukan merupakan duplikat dan mengutip seluruhnya karya orang lain. Apabila di kemudian hari, karya asli saya disinyalir bukan merupakan karya asli saya, maka saya akan bersedia menerima segala konsekuensi apa pun yang diberikan Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Malang, 10 Januari 2017

Yang membuat pernyataan




Musyaffa Al - Hafi
NIM. 13.18.109

GAME ENDLES RUNING SI KANCIL MENGUNAKAN METODE FINITE STATE MACHINE

Musyaffa Al - Hafi (1318109)
Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km. 2 Tasikmadu, Malang
Email : allhaafi@gmail.com

Dosen Pembimbing : 1. Karina Auliasari, ST, M.Eng

2. Rofila El Maghfiroh, S.Si.M.Sc

Abstrak

Indonesia adalah negara yang kaya akan wisatanya, sumber daya alamnya, budayanya, bahasanya, legenda dan cerita rakyatnya, dan masih banyak lagi. Dalam legenda tersebut, sikancil diceritakan sangat suka mengambil timun, contohnya adalah Cerita Legenda Si kancil Suka Mengambil Timun. Akan tetapi semakin bertambahnya zaman dan semakin berkembangnya teknologi, semakin banyak masyarakat yang terjun kedalam modernisasi tersebut, maka akan semakin banyak juga orang-orang meninggalkan kekayaan tradisi dan cerita legenda Indonesia.

Dalam pembuatan game ini penulis menggunakan game engine unity3D, dengan menerapkan metode Finite State Machine. Finite State Machine digunakan pada karakter player yang digerakan oleh kecerdasan buatan untuk mendukung game tersebut. Finite State Machine dapat digunakan untuk menentukan gerakan dan aksi dari player. Salah satu gerakan dan aksinya adalah pengambilan poin.

Target yang dicapai adalah dapat dimainkan oleh anak-anak usia dini bahkan hingga orang dewasa. Dari pengujian performa game dapat dijalankan dikomputer dengan aspek RAM 1-4GB, VGA minimal 512, dengan OS Windows 7 hingga 8. Dari hasil pengujian pengguna 7 dari 10 orang menyatakan gameplay game "Endles Runing Si Kancil" menarik dengan presentasi 70% dan sisa 3 lainnya menyatakan kurang menarik dengan persentase 30%. Untuk genre platfromer 7 dari 10 orang menyatakan menarik dengan presentasi 70%, 2 menyatakan kurang menarik dengan persentase 20% , dan 1 menyatakan tidak menarik dengan presentasi 10% dan untuk desain karakter 6 dari 10 menyatakan menarik dengan persentase 60%, 4 menyatakan kurang menarik dengan persentase 40%.

Kata kunci : Cerita Legenda, Game, Finite State Machine, Game Engine Unity3D .

KATA PENGANTAR

Puji syukur kehadiran Tuhan yang maha kuasa, karena telah memberikan hikmat-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul *GAME ENDLES RUNNING SI KANCIL MENGGUNAKAN METODE FINITE STATE MACHINE* sesuai dengan waktu yang ditentukan.

Penyusunan skripsi ini merupakan salah satu persyaratan untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika, Fakultas Teknologi Industri di Institut Teknologi Nasional Malang.

Pada penyusunan skripsi ini kami mengucapkan banyak terima kasih sebesar-besarnya kepada :

1. Dr. Ir. Lalu Mulyadi, MT, selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
4. Suryo Adi Wibowo, ST, MT, selaku Sekertaris Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
5. Karina Auliasari, ST, M.Eng, selaku Dosen Pembimbing I, yang selalu memberikan masukan.
6. Rofila El Magfiroh, S.Si. M.Sc, selaku Dosen Pembimbing II, yang selalu memberikan masukan.
7. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.
8. Semua teman seperjuangan yang telah membantu dalam terselesaikannya skripsi ini.

Penyusun menyadari bahwa skripsi masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca, Semoga skripsi ini bisa bermanfaat bagi pembaca

Malang, 12 Januari 2017

Penulis.

DAFTAR ISI

ABSTRAK	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan	2
1.4. Batasan Masalah.....	2
1.5. Metode Penelitian.....	2
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI	
2.1. Definisi Game	5
2.2. Jenis – Jenis Game	5
1 Game Action.....	5
2 Game Action-Adventure.....	5
3 Game Simulasi, Kontruksi, dan Manajemen.	6
4 Role Playing Games (RPG).. ..	6
5 Game Strategi.. ..	6
6 Game Racing.. ..	6
7 Game Sport.. ..	6
8 Game Puzzle.. ..	7
9 Game Endles Runner.. ..	7
2.3. Kecerdasan Buatan (<i>Artificial Intelligence</i>).....	7
2.4. Jenis – Jenis Kecerdasan Buatan.....	8
2.4.1 Decision Marking.....	8
1 Finite State Machines (FSM).....	8
2 Fuzzy Logic.. ..	9

2.4.2 Pathfinding.....	9
1 Algoritma A Star (A*).....	9
2 Algoritma Dijkstra.....	9
3 Algoritma Minmax.....	9
4 Breadth Search First (BFS).....	11
5 Depth Search First (DFS).....	11
2.5. Unity 3D.....	11
BAB III ANALISIS DAN PERANCANGAN SISTEM	
3.1. Analisa dan Kebutuhan Sistem	13
3.1.1 Cerita Game.....	13
3.1.2 Desain Level.....	13
3.1.3 Analisa Target User.....	14
3.1.4 Analisa Konsep Game.....	14
3.1.5 Analisa Kebutuhan Perangkat.....	14
3.1.6 Analisa Kebutuhan Perangkat Lunak.....	14
3.1.7 Analisa Kebutuhan Perangkat Keras.....	14
3.2. Perancangan Game.....	15
3.2.1 Story Line.....	15
3.2.2 Desain Level.....	15
3.2.3. Kontrol Game	16
3.2.4 Desain Sistem	16
3.2.5 Struktur Menu.....	17
3.2.6 Flowchart.....	18
3.2.7 Perancangan Alur Finite State Machine	18
3.2.8 Desain GUI (Grapichal User Interface).....	20
3.2.9 Desain Karakter	21
3.2.10 Perancangan Environment.....	22
BAB IV IMPLEMENTASI DAN PENGUJIAN	
4.1 Hasil Implementasi Sistem Pendukung Keputusan.....	27
4.1.1 Tampilan Menu Utama.....	27
4.1.2 Tampilan Menu Petunjuk	28
4.1.3 Tampilan Menu Keluar	28

4.2	Pengujian Game Play	29
4.2.1	Pengujian Karakter Dengan Dunia Game	29
4.2.2	Pengujian Animasi Karakter	29
4.2.3	Pengujian Karakter Pengambila Koin	30
4.2.4	Pengujian Tombol Dalam Game	32
4.2.5	Pengujian Tampilan Game Over	32
4.3	Pengujian Artificial Intelligence (AI).....	33
4.4	Pengujian Performance	33
4.5	Pengujian Control Player	34
4.6	Pengujian Terhadap Pengguna	35
BAB V PENUTUP		
5.1.	Kesimpulan	36
5.2.	Saran.....	36
DAFTAR PUSTAKA		37
LAMPIRAN - LAMPIRAN		

DAFTAR GAMBAR

Gambar 2.1. Contoh Diagram <i>Finite State Machine</i>	8
Gambar 3.1. Kontrol <i>Game</i>	16
Gambar 3.2. Desain Sistem.....	17
Gambar 3.3. Struktur <i>Menu</i>	17
Gambar 3.4. <i>Flowchart Game</i>	18
Gambar 3.5. Penerapan FSM Pada <i>Player</i>	19
Gambar 3.6. Penerapan FSM Pada Musuh	19
Gambar 3.7. Tampilan <i>Desain GUI Menu Utama</i>	20
Gambar 3.8. Tampilan <i>Desain GUI Menu Petunjuk</i>	21
Gambar 3.9. Tampilan <i>Desain GUI Menu Game Over</i>	21
Gambar 4.1. Pengujian Tampilan Menu Utama.....	27
Gambar 4.2. Pengujian tampilan Menu Petunjuk	28
Gambar 4.3. Pengujian Tampilan Menu Keluar	28
Gambar 4.4. Pengujian Karakter Dalam <i>Game</i>	29
Gambar 4.5. Pengujian Animasi Karakter Berlari	30
Gambar 4.6. Pengujian Animasi Karakter Melompat.....	30
Gambar 4.7. Sebelum Mengambil Koin, Piala, dan Kunci.....	31
Gambar 4.8. Sesudah Mengambil Koin, Piala dan Kunci.....	31
Gambar 4.9. Pengujian Kontrol Dalam <i>Game</i>	32
Gambar 4.10 Pengujian Tampilan <i>Game Over</i>	32

DAFTAR TABEL

Tabel 3.1 Perancangan Karakter	22
Tabel 3.2 Perancangan <i>Environtmenr</i>	23
Tabel 4.1 Pengujian AI (<i>Artifical Intellegent</i>)	34
Tabel 4.2 Pengujian FSM (<i>Finite State Machine</i>)	34
Tabel 4.3 Pengujian Tombol.....	35
Tabel 4.4 Pengujian <i>Performance</i>	35
Tabel 4.5 Pengujian <i>Control Player</i>	36
Tabel 4.6 Pengujian Terhadap Pengguna	37

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia teknologi pada saat ini berkembang dengan sangat pesat. Perkembangan teknologi mencakup semua bidang kehidupan manusia seperti kesehatan, pangan, industri, dan lain-lain. Begitupun dengan perkembangan teknologi dalam bidang *game*, saat ini sudah sangat bagus karena *game* saat ini seperti kondisi nyata.

Perlu kita ketahui Indonesia adalah negara yang kaya akan wisatanya, sumber daya alamnya, budayanya, bahasanya, legenda dan cerita rakyatnya, dan masih banyak lagi. Dalam legenda tersebut, si kancil diceritakan sangat suka mengambil timun, contohnya adalah Cerita Legenda Si kancil Suka Mengambil Timun. Akan tetapi semakin bertambahnya zaman dan semakin berkembangnya teknologi, semakin banyak masyarakat yang terjun kedalam modernisasi tersebut, maka akan semakin banyak juga orang-orang meninggalkan kekayaan tradisi dan cerita legenda Indonesia.

Berdasarkan uraian diatas, peneliti tertarik untuk membuat "*Game Endles Runing Si Kancil*", dengan gendre *Endles Runner*. Peneliti berharap *game* ini dapat menjadi salah satu bentuk usaha untuk melestarikan budaya indonesia, selain itu, bisa untuk berpartisipasi dalam dunia *game* khususnya di Indonesia. Dalam "*Game Endles Run Si Kancil*" disajikan dengan visualisasi 3d (tiga dimensi).

1.1. Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat diambil beberapa rumusan masalah dalam penelitian ini, yaitu sebagai berikut:

1. Bagaimana cara merancang *Game Endles Runging Si Kancil* ?
2. Bagaimana mengimplementasikan AI dan metode *Finite State Machine* pada *Game Endles Runging Si Kancil*?
3. Bagaimana implementasi *Game Endles Runging Si Kancil* menggunakan *Unity Game Engine*?

1.2. Batasan Masalah

Batasan masalah pada pembuatan *game Endles Runing Si Kancil* adalah sebagai berikut:

1. Jumlah *stage* dalam *game Endles Runing Si Kancil* ini dibuat sebanyak dua *stage*.
2. Ringtangan dalam *game Endles Runing Si Kancil* ini bersifat passif.
3. *Game Endles Runing Si Kancil* akan dibuat *game* berbasis *desktop*.

1.3. Tujuan

Tujuan yang dapat dicapai dalam pembuatan *game Endles Runing Si Kancil* adalah sebagai berikut:

1. Mengimplementasikan AI pada *game Endles Runing Si Kancil*.
2. Mengimplementasikan metode *Finite State Machine* pada *game Endles Runing Si Kancil*.
3. Mengimplementasikan *game Endles Runing Si Kancil* menggunakan *Unity Game Engine*.

1.2 Metodologi Penelitian

Langkah-langkah yang digunakan dalam penyusunan skripsi disini menggunakan metode penelitian berikut :

1. Studi Literatur

Pada tahap ini dipelajari literature dan perencanaan serta konsep awal untuk merancang *game* yang akan dibuat yaitu didapat dari referensi buku, internet, maupun sumber-sumber yang lain.

2. Pengumpulan Data dan Analisis

Pada tahap ini adalah proses pengumpulan data yang dibutuhkan untuk pembuatan *game*, serta melakukan analisa atau pengamatan pada data yang sudah terkumpul untuk selanjutnya diolah lebih lanjut.

3. Analisa dan Perancangan Sistem

Setelah selesai pada tahap pengumpulan data dan analisis maka tahap selanjutnya adalah melakukan analisa dan perancangan sistem. Pada tahap ini adalah proses perancangan dari *game* yang akan dibuat untuk selanjutnya akan diproses lebih lanjut.

4. Pembuatan Game

Setelah tahap perancangan system maka tahap selanjutnya adalah pembuatan game. Pada tahap ini rancangan yang sebelumnya telah dibuat akan diterapkan pada program yang akan dibuat. Pembuatan game ini menggunakan pemrograman Unity 3D, C# dan *Endles Runing* sebagai metode penalaran pada program ini.

5. Uji Coba Game

Setelah game selesai dibuat maka dilakukan pengujian program untuk mengetahui apakah game tersebut telah bekerja dengan benar dan sesuai dengan sistem yang dibuat.

6. Pembuatan Kesimpulan

Pada tahap akhir ini adalah pembuatan kesimpulan atau ringkasan dari skripsi ini dan kesimpulan tentang program yang telah dibuat.

1.3 Sistematika Penulisan

Untuk mempermudah memahami pembahasan pada penulisan skripsi ini, maka sistematika penulisan yang diperoleh sebagai berikut :

BAB I : Pendahuluan

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metode penelitian dan sistematika penulisan.

BAB II : Tinjauan Pustaka

Bab ini berisi referensi mengenai permasalahan yang berhubungan dengan penelitian ini.

BAB III : Analisis dan Perancangan Sistem

Bab ini berisi mengenai perancangan game *Endles Runing Si Kancil Genre Endles Runner* yang akan dibuat.

BAB IV : Implementasi dan Pengujian

Bab ini berisi implementasi metode penalaran pada game *Endles Runing Si Kancil*, serta melakukan pengujian terhadap game tersebut.

BAB V : Penutup

Baba ini berisi kesimpulan dari hasil penelitian yang dilakukan dan saran yang dapat digunakan untuk bahan pengembangan penelitian berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Game

Dalam bahasa Indonesia *game* diartikan sebagai permainan. Permainan adalah kegiatan kompleks yang didalamnya terdapat peraturan, *Play* dan budaya. Sebuah permainan adalah sistem dimana pemain terlibat konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan. (Lestari, 2012).

2.2 Jenis-Jenis Game

Seiring dengan perkembangan industri *game*, jenis-jenis *game* semakin bervariasi. Perbedaan jenis *game* terletak pada *gameplay*, interaksi, dan kategori. *Gameplay* merupakan sebuah sistem yang berjalan pada *game*. Sistem tersebut meliputi *story line*, cara bermain, *menu*, *area* permainan, dan lain sebagainya. Bisa jadi dua *game* memiliki banyak persamaan. Namun, yang membedakannya adalah bagaimana pemain memainkan dan berinteraksi dengan *game*. Berdasarkan hal-hal tersebut, *game* dikategorikan menjadi berbagai macam-macam jenis *game* yang meliputi :

1. Game Action

Genre ini merupakan macam *game* yang paling populer. *Game* jenis ini membutuhkan kemampuan refleks pemain dan *timing* yang tepat. Salah satu subgenre *action* yang populer adalah *First Person Shooter* (FPS). Banyak sekali *game* sukses di pasaran yang termasuk dalam genre ini. *Game* jenis ini memerlukan kecepatan berpikir. *Game* ini dibuat seolah-olah pemain yang berada dalam suasana tersebut. Contoh *game* genre ini, misalnya *Half Life*, *Crysis*, *Call of Duty: Modern Warfare*, dan lain sebagainya.

2. Game Action-Adventure

Genre ini memadukan *gameplay* aksi dan petualangan. Pemain diajak untuk menelusuri gua bawah tanah sambil mengalahkan musuh, mencari artefak kuno, menyeberangi sungai dan sebagainya. Saat ini kebanyakan genre ini sudah mengadopsi 3D. *Tom Clancy*, *Splinter Cell*, *Hitman*, *Tomb Rider* dan *Prince of Persia* termasuk dalam *game* ini.

3. Game Simulasi, Konstruksi, dan Manajemen

Pemain dalam *game* ini diberi keleluasaan untuk membangun, berekspansi, dan mengatur komunitas fiksi atau proyek tertentu dengan bahan baku yang terbatas. Contohnya adalah *SimCity*, *The Sims*, dan *Spore*.

4. Role Playing Games (RPG)

Dalam RPG pemain dapat memilih satu karakter untuk dimainkan. Seiring dengan naiknya *level game*, karakter tersebut dapat berubah, bertambah *skill*-nya, bertambah senjatanya, bertambah hewan peliharaannya dan lain sebagainya. *Final Fantasy*, *World of Warcraft*, *Fallout*, dan *Dragon Quest* termasuk dalam *genre* ini.

5. Game Strategi

Awal mula *genre* ini adalah *board game*. *Genre* strategi menitikberatkan pada kemampuan berpikir dan organisasi. *Game* strategi dibedakan menjadi dua, yaitu *Turn Based Strategy* dan *Real Time Strategy*. *Genre Real Time Strategy* mengharuskan pemain membuat keputusan dan secara bersamaan pihak lawan juga beraksi hingga menimbulkan serangkaian kejadian dalam waktu yang sebenarnya. Contohnya adalah *Age of Empires*, *Starcraft*, *Rise of Nation* dan *Command and Conquer*. Sementara itu *genre Turn Based Strategy* mengharuskan pemain bergantian menjalankan taktiknya. Saat pemain mengambil langkah, pihak lawan menunggu. Demikian juga sebaliknya. Termasuk dalam *genre* ini adalah *Heroes of Might and Magic*, *Front Mission* dan *Master of Orion*.

6. Game Racing

Pemain dapat memilih kendaraan, mendandani lalu melaju di arena balap. Tujuannya hanya satu, yaitu mencapai garis *finish* tercepat. Misalnya *Need For Speed*, *Grand Turismo*, *Top Gear*, *Daytona* dan lain sebagainya.

7. Game Sport

Genre ini membawa olahraga ke dalam komputer atau konsol. Biasanya *gameplay* dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya. Termasuk dalam *genre* ini adalah *FIFA*, *Winning Eleven*, *PESS*, *NBA*, *Tony Hawk Pro Skater*, dan lain-lain.

8. Game Puzzle

Genre *puzzle* menyajikan teka-teki, menyamakan warna bola, perhitungan matematika, menyusun balok, dan sebagainya. Misalnya *Tetris*, *Bejeweled*, *Minesweeper*, dan *Bomberman*.

9. Game Endles Runner

Game bergenre *Endless Runner* merupakan game dengan permainan unik dimana pemain diharuskan terus berlari dan berlari sambil menghindari berbagai rintangan dan juga musuh yang menghadang. *Game* berjenis satu ini tidak memiliki akhir dengan kata lain permainan akan terus bergulir sampai kalah. Tujuan dari game berjenis ini mengumpulkan skor paling tinggi, semakin jauh dari titik awal permainan maka semakin besar skor yang raih dan semakin tinggi levelnya.

2.3 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Menurut John McCarthy, 1956, AI : untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik. Manusia cerdas (pandai) dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu akan lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan menalar

yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik (Dahria, 2008).

2.4 Jenis-Jenis Kecerdasan Buatan

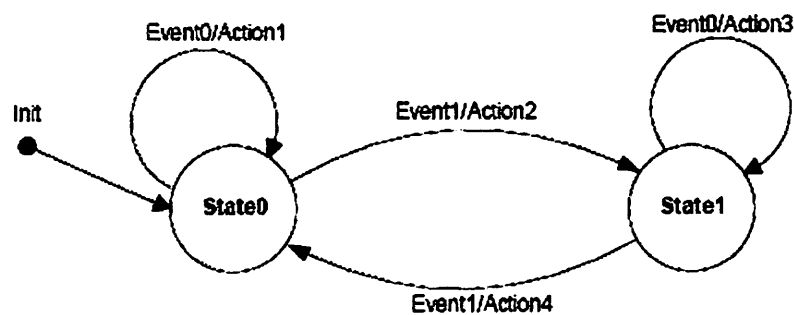
Adapun beberapa jenis kecerdasan buatan meliputi :

2.4.1 Decision Making

Jenis-jenis kecerdasan buatan *decision making* meliputi :

1. Finite State Machines (FSM)

FSM (*Finite State Machine*) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *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, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. 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. Contoh diagram state sederhana ditunjukkan pada gambar 2.1.



Gambar 2.1. Contoh Diagram Finite State Machine

Diagram tersebut memperlihatkan FSM dengan dua buah *state* dan dua buah *input* serta empat buah aksi output yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*,

sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

Secara formal FSM dinyatakan oleh 5 tuple atau $M=(Q, \Sigma, \delta, S, F)$, (Utdirartama, 2001) dimana:

Q = himpunan *state*/kedudukan

Σ = himpunan symbol *input*/masukan/abjad

δ = fungsi transisi

S = *state awal*/ kedudukan awal (*initial state*), $S \subseteq Q$

F = himpunan *state akhir*, $F \subseteq Q$

2. Fuzzy Logic

Fuzzy Logic diperkenalkan oleh Prof. Lotfi Zadeh pada tahun 1965. Merupakan metode yang mempunyai kemampuan untuk memproses variabel yang bersifat kabur atau yang tidak dapat dideskripsikan secara eksak/pasti seperti misalnya tinggi, lambat, bising. Dalam *fuzzy logic variabel* yang bersifat kabur tersebut direpresentasikan sebagai sebuah himpunan yang anggotanya adalah suatu nilai *crisp* dan derajat keanggotaannya (*membershipfunction*) dalam himpunan tersebut (Hermawanto, 2008).

2.4.2 Pathfinding

Jenis-jenis kecerdasan buatan *Pathfinding* meliputi :

1. Algoritma A Star(A*)

Algoritma A* adalah sebuah algoritma yang telah diperkaya. Dengan menerapkan suatu heuristik, algoritma ini membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan. Algoritma A* membangkitkan simpul yang paling mendekati solusi. Simpul ini kemudian disimpan ke dalam list sesuai dengan urutan yang paling mendekati solusi terbaik. Kemudian, simpul pertama pada list diambil, dibangkitkan dan kemudian disimpan ke dalam list sesuai dengan urutan yang terbaik untuk solusi. List simpul ini disebut dengan simpul terbuka(*open node*). (Yoseph, 2006).

2. Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip dari algoritma *Dijkstra* adalah dengan pencarian dua lintasan yang paling kecil. Algoritma *Dijkstra* memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik yang diketahui (dari titik awal) diperbarui bila ternyata didapat titik yang baru yang memberikan jarak terpendek. Syarat algoritma ini adalah bobot sisinya yang harus *non-negatif* (Sholichin, 2013).

3. Algoritma Minimax

Algoritma *Minimax* merupakan algoritma yang digunakan untuk menentukan pilihan agar memperkecil kemungkinan kehilangan nilai maksimal. Algoritma ini diterapkan dalam permainan yang melibatkan dua pemain seperti *tic tac toe*, *checkers*, *go* dan permainan yang menggunakan strategi atau logika lainnya. Hal ini berarti permainan-permainan tersebut dapat dijelaskan sebagai suatu rangkaian aturan dan premis. Algoritma ini mulai dikembangkan dari teori *game zero-sum*. Teori ini mendeskripsikan situasi dimana jika terdapat pemain yang mengalami pendapatan, pemain lain akan mengalami kehilangan dengan nilai yang sama dari pendapatan tersebut, dan sebaliknya. Jumlah pendapatan dari pemain yang dikurangi dengan jumlah kehilangan akan berjumlah nol. Teori *minimax* menyatakan: Untuk setiap dua orang pemain dalam *zero-sum game*, terdapat nilai V dari strategi yang dimiliki pemain seperti :

1. Strategi yang ditentukan pemain kedua akan menghasilkan konsekuensi kemungkinan untuk pemain pertama, V
2. Strategi yang ditentukan pemain pertama akan menghasilkan konsekuensi kemungkinan untuk pemain pertama, $-V$

Secara setara, strategi pemain pertama akan memastikan suatu nilai V tanpa memperdulikan strategi pemain kedua, dan bersamaan dengan itu pemain kedua akan memastikan dirinya kehilangan nilai sebesar $-V$. Algoritma *Minimax* merupakan algoritma dasar pencarian DFS (*Depth-First Search*) untuk melakukan *traversal* dalam pohon. DFS akan mengekskansi simpul paling dalam terlebih dahulu. Setelah simpul akar dibangkitkan, algoritma ini akan

membangkitkan simpul pada tingkat kedua, yang akan dilanjutkan pada tingkat ketiga, dst. Dalam melakukan traversal, misalkan dimulai dari suatu simpul i , maka simpul selanjutnya yang akan dikunjungi adalah simpul tetangga j , yang bertetangga dengan simpul k , selanjutnya pencarian dimulai lagi secara rekursif dari simpul j . Ketika telah mencapai simpul m , dimana semua simpul yang bertetangga dengannya telah dikunjungi, pencarian akan dirunutbalik ke simpul terakhir yang dikunjungi sebelumnya dan mempunyai simpul j yang belum dikunjungi. Selanjutnya pencarian dimulai kembali dari j . Ketika tidak ada lagi simpul yang belum dikunjungi yang dapat dicapai dari simpul yang telah dikunjungi maka pencarian selesai. (Nadhira, 2008).

4. Breadth Search First (BFS)

Breadth First Search adalah suatu metode yang melakukan pencarian secara melebar yang mengunjungi simpul secara preorder yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon graf berakar, maka semua simpul pada aras d dikunjungi lebih dahulu sebelum simpul-simpul pada aras $d+1$ (Prasetyo, 2014).

5. Depth Search First (DFS)

Depth First Search (DFS) adalah suatu metode pencarian pada sebuah pohon dengan menelusuri satu cabang sebuah pohon sampai menemukan solusi. Pencarian dilakukan pada satu node dalam setiap *level* dari yang paling kiri dan dilanjutkan pada *node* sebelah kanan. Jika solusi ditemukan maka tidak diperlukan proses backtracking yaitu penelusuran balik untuk mendapatkan jalur yang diinginkan. Pada metode DFS pemakaian memori tidak banyak karena hanya *node-node* pada lintasan yang aktif saja yang disimpan. Selain itu, jika solusi yang dicari berada pada *level* yang dalam dan paling kiri, maka DFS akan menemukannya secara cepat (Prasetyo, 2014).

2.5 Unity 3D

Unity merupakan suatu aplikasi yang digunakan untuk mengembangkan game multi platform yang didesain untuk mudah digunakan. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. *Editor* pada *Unity* dibuat dengan *user interface* yang sederhana. *Editor* ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan ranking teratas untuk editor game. Grafis pada *unity* dibuat dengan *grafis* tingkat tinggi untuk *OpenGL* dan *directX*. *Unity* mendukung semua *format file*, terutamanya format umum seperti semua format dari art applications. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada Mac OS x dan windows dan dapat menghasilkan game untuk *Mac, Windows, Wii, iPhone, iPad* dan *Android*.

Unity secara rinci dapat digunakan untuk membuat *video game 3D, real time animasi 3D* dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. *Editor Unity* dapat menggunakan *plugin* untuk *web player* dan menghasilkan *game browser* yang didukung oleh *Windows* dan *Mac*. *Plugin web player* dapat juga dipakai untuk *widgets Mac*. *Unity* juga akan mendukung *console* terbaru seperti *PlayStation 3* dan *Xbox 360*. Pada tahun 2010, telah memperoleh *Technology Innovation Award* yang diberikan oleh *Wall Street Journal* dan tahun 2009, *Unity Technology* menjadi 5 perusahaan game terbesar. Tahun 2006, menjadi juara dua pada *Apple Design Awards*.

Server aset dari *Unity* dapat digunakan semua *scripts* dan *aset game* sebagai solusi dari versi kontrol dan dapat mendukung proyek yang terdiri atas banyak *gigabytes* dan ribuan dari *file multi-megabyte*. *Editor Unity* dapat menyimpan metadata dan versi mereka, itu dapat berjalan , pembaharuan dan didalam perbandingan versi grafis. *Editor Unity* dapat diperbaharui dengan sesegera mungkin seperti file yang telah dimodifikasi. *Server aset Unity* juga cocok pada *Mac, Windows* dan *Linux* dan juga berjalan pada *PostgreSQL, database server opensource*.

Perizinan atau *license* dari *Unity* ada dua bentuk. Ada *Unity* dan *Unity Pro*. Versi *Unity* tersedia dalam bentuk gratis, sedang versi *Unity Pro* hanya dapat dibeli. *Versi Unity Pro* ada dengan fitur bawaan seperti *efek post processing* dan *render efek texture*. Versi *Unity* merupakan yang gratis memperlihatkan aliran

untuk *game web* dan layar splash untuk game yang berdiri sendiri. *Unity* dan *Unity Pro* menyediakan tutorial, isi, contoh *project*, wiki, dukungan melalui forum dan perbaruan kedepannya. *Unity* digunakan pada *iPhone*, *iPod* dan *iPad* *operating system* yang mana *iOS* ada sebagai add-ons pada *Unity editor* yang telah ada lisensinya, dengan cara yang sama juga pada *Android*.

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis dan Kebutuhan Sistem

Game yang dibangun merupakan game *Endels Runner 3 Dimensi* yang terdiri dari 2 *Stage* yang dapat dipilih sebagai arena permainan. Berikut ini adalah analisa sistem yang ada pada game ini :

1. Sistem *single player*.
2. Grafik game 3 Dimensi.
3. Interaksi menggunakan *keyboard*.
4. *Enemy Follow* diimplementasikan menggunakan pada karakter musuh.
5. Game ini dibuat menggunakan *Unity3D* dengan bahasa pemrograman C#.
6. Game ini ditargetkan untuk berjalan pada *Platform Microsoft Windows 7* atau yang lebih baru.

3.1.1. Cerita Game

Untuk alur cerita game ini yaitu :

- a. Disuatu hari terdapat seorang petani yang sedang merawat timun.
- b. Ketika pak tani sedang bercock tanam, kemudian datanglah Si Kacil.
- c. Kemudian Si Kancil mengambil timun.
- d. Pak tani mengetahui perbuatan Si Kancil, kemudian pak tani mengejanya.

3.1.2. Desain Level

Game ini terbagi menjadi 2 *stage*, masing-masing *stage* berbeda ruangnya dan cara permainan dalam game ini sama. Beberapa aturan dari game ini yaitu :

- a. Player mempunyai aksi utama yaitu, *Running* (Lari), *Jump* (Lompat).
- b. Player harus berlari sejauh – jauhnya dan mengambil poin, kunci dan piala.
- c. Jika *enemy* menyentuh *player* maka akan *game over*.
- d. Terdapat *score*, jika *score* bertambah banyak maka level naik dan *player* berlari akan semakin kencang.
- e. Terdapat piala disetiap akhir *stag*.
- f. Ketika *game over*, jika *player* mempunyai kunci maka *player* bisa menggunakan kunci tersebut untuk meneruskan game.

3.1.3 Analisis Target User

Analisis pengguna digunakan untuk mengetahui spesifikasi pengguna untuk dapat memainkan *Game Endles Runing Si Kancil* yaitu :

1. Pengguna untuk usia 8 tahun ke atas.
2. Berlatar belakang pendidikan SD dan SMP
3. Pengguna mengerti cara pengoprasian computer.

3.1.4 Analisa Konsep Game

1. Genre

Ganre Dalam *game Endles Runing Si Kancil* menggunakan *Endles Run*, yaitu dimana game terlihat dari belakang dengan desain 3D.

3.1.5 Analisis kebutuhan perangkat

Analisis kebutuhan perangkat menggambarkan kebutuhan yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta user sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan meliputi :

3.1.6 Analisis Kebutuhan Perangkat Lunak

1. Spesifikasi perangkat lunak bagi pengembang yang digunakan dalam membangun *game Endles Runing Si Kancil* yaitu meliputi :
 - a. Sistem Operasi Windows 7
 - b. Software Unity 5.0.0
 - c. Software *Adobe Phototshop CS6*
 - d. Software *CorelDraw X8*
 - e. Adobe Blender
2. Sedangkan spesifikasi perangkat lunak bagi pemain yang memainkan game *Petualangan Si Kabayan Menyelamatkan Hewan Punah* adalah dengan sistem operasi *Windows XP, Windows 7, 8, 8.1, dan 10.*

3.1.7 Analisis Kebutuhan Perangkat Keras

1. Spesifikasi *hardware* atau perangkat keras yang dibutuhkan oleh pengembang yaitu meliputi :

- a. Prosesor dengan kecepatan 2.3 Ghz.
 - b. RAM 2 gigabyte.
 - c. *Hardisk* 20 gigabyte.
 - d. VGA Card 1024 megabyte.
 - e. Monitor.
 - f. *Mouse* dan *Keyboard*.
 - g. *Speaker*.
2. Spesifikasi *hardware* atau perangkat keras yang dibutuhkan oleh pemain adalah :
- a. Prosesor minimum dengan kecepatan 1.8 Ghz.
 - b. RAM 2 gigabyte.
 - c. Hardisk 10 gigabyte.
 - d. VGA Card 512 megabyte.
 - e. Monitor.
 - f. *Mouse* dan *Keyboard*.
 - g. *Speaker*.

3.2 Perancangan Game

Perancangan *game* adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci tentang *Game Endles Runing Si Kancil*. Perancangan *Game Endles Runing Si Kancil* meliputi :

3.2.1 Story Line

Disuatu hari ada seorang petani yang sedang merawat timun diperkebunan. Ketika pak tani sedang merawat timun, kemudian datanglah Si Kacil, Si Kancil kemudian mengambil timun yang sedang pak tani rawat, Pak tani mengetahui perbuatan si kancil tersebut, kemudian pak tani mengejarnya, si kancil pun lari.

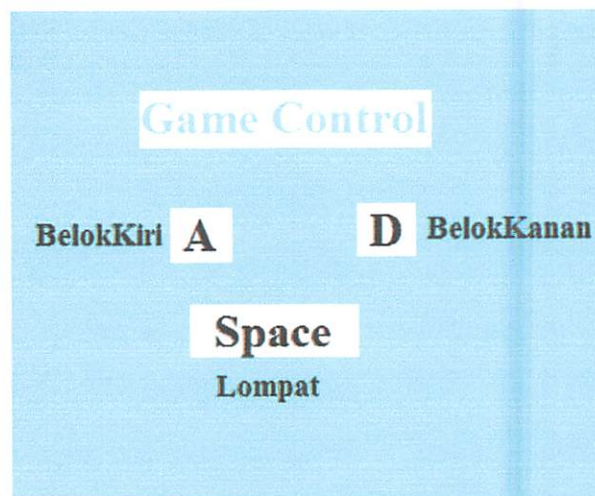
3.2.2 Desain Level

Dalam *game Endles Runing Si Kancil* ini, *score* akan menentukan tingkat level dan kesulitan. Beberapa aturan dari *game* ini yaitu :

1. Player mempunyai aksi utama yaitu, *Running* (Lari), *Jump* (Lompat), Belok kanan, dan Belok kiri.
2. Player harus berlari sejauh – jauhnya dan mengambil poin, kunci dan piala.
3. Jika *enemy* (Petani) menyentuh *player* maka akan *game over*.
4. Terdapat *score*, jika *score* bertambah 10 maka level game naik dan *player* berlari semakin cepat.
5. Terdapat piala disetiap akhir *stag*.
6. Ketika *game over*, jika *player* mempunyai kunci maka *player* bisa menggunakan kunci tersebut untuk meneruskan game.

3.2.3 Kontrol Game

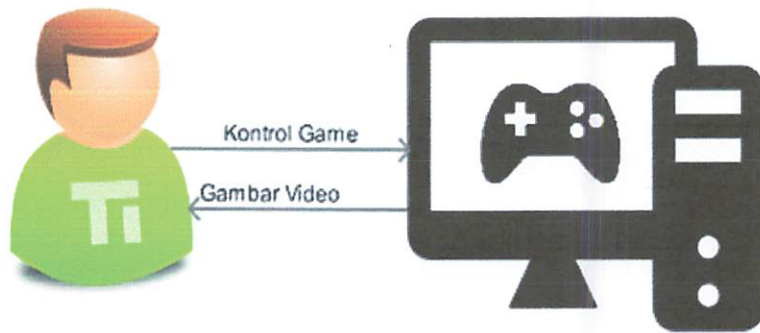
Game ini dimainkan menggunakan *Keyboard* seperti pada Gambar 3.1.



Gambar 3. 1 Kontrol Game

3.2.4 Desain Sistem

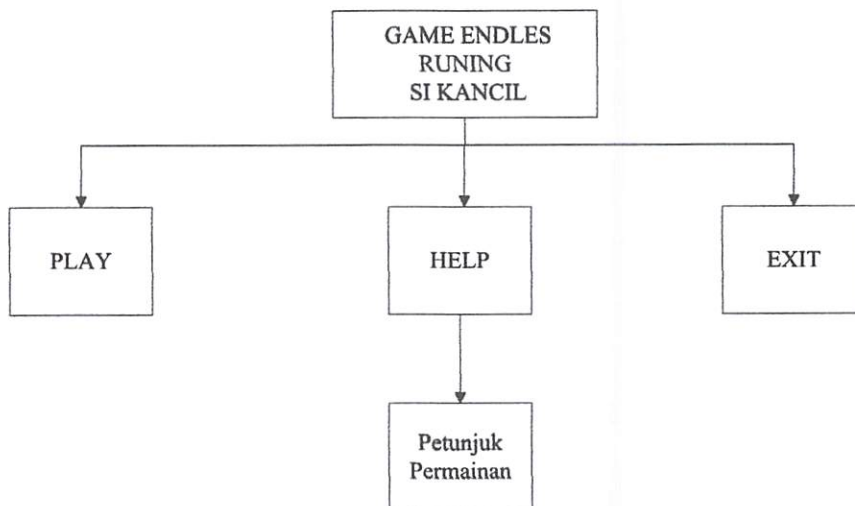
Game Endles Runing Si Kancil ini merupakan game *personal computer*. Desain sistem dari game ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Desain Sistem

3.2.5 Struktur Menu

Pada game *Endles Run Si Kancil* ini terdiri dari beberapa menu yaitu Play, Help dan Exit. Diagram struktur menu dapat dilihat pada Gambar 3.3.



Gambar 3.3 Struktur Menu

Pada Gambar 3.3, game *Endles Runing Si Kancil* memiliki 3 menu utama, yaitu menu *Play*, *Help*, dan *Exit*. Dimana menu *Play* memiliki fungsi memulai game. Menu *Help* berisi tentang penjelasan control yang digunakan dalam dunia game. Lalu pada menu *exit* berfungsi sebagai sarana untuk keluar dari dunia game.

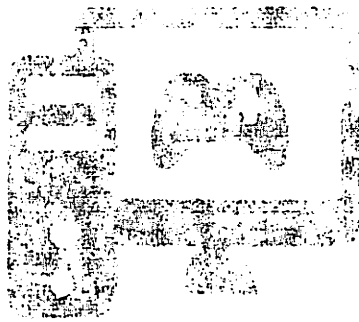


Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

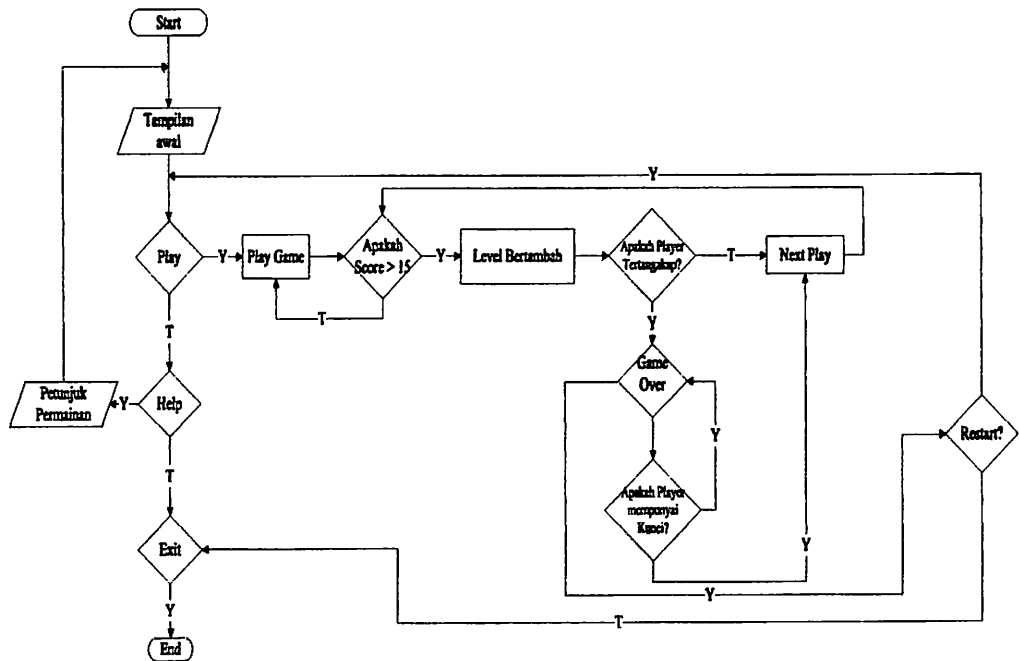
Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

Fig. 1. A vintage television set with a large screen and a wooden cabinet base.

3.2.6 Flowchart

Pada tahap perancangan diagram alir pada game. Dimulai ketika *player* membuka game akan menuju menu utama yang memiliki 3 menu pilihan yaitu *Play*, *Help* dan *Exit*. Seperti ditunjukkan pada Gambar 3.4.



Gambar 3.4 Flowchart Game

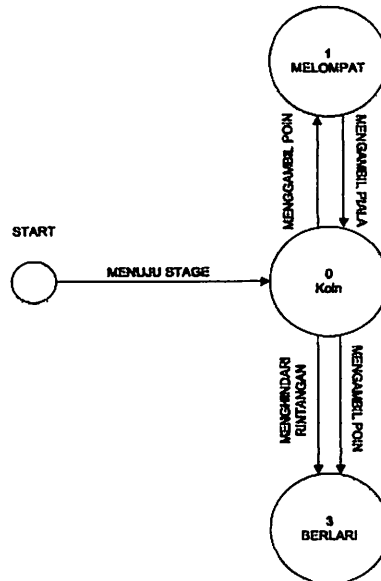
Pada Gambar 3.4 program dimulai dari *start* kemudian masuk ke 3 menu utama yaitu *Play*, *Help* dan *Exit*. Jika pemain memulai Game maka akan langsung menuju *stage 1* dan level rendah, jika poin *player* bertambah 15 maka *level* permainan bertambah cepat. Jika pemain kalah, akan di arahkan ke tampilan *Game over*, jika *player* mempunyai *freekey* (kunci) maka *player* bisa melanjutkan game dengan cara menggunakan *freekey*, jika tidak menuju ke tampilan *game over*, dan akan diarahkan pada utama *game*.

3.2.7 Perancangan Alur Finite State Machine

Dalam game *Endles Runing Si Kancil* memiliki 2 jenis metode FSM, yaitu metode FSM pada musuh karakter *player*, metode FSM pada musuh (pak tani).

1. Penerapan FSM pada karakter Player

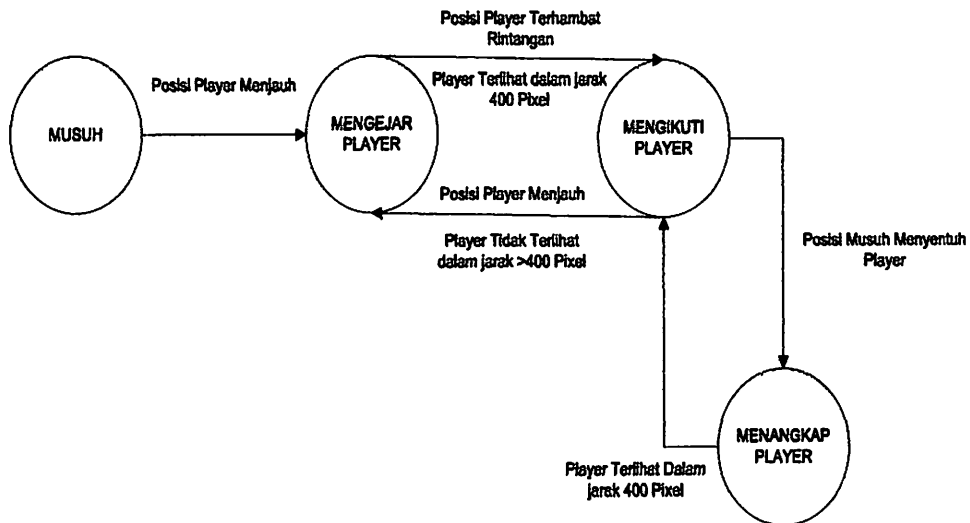
Penerapan alur FSM pada karakter *player* memiliki sifat jika karakter *player* menyentuh pada koin, piala, dan *freekey* (kunci bebas) maka *player* akan mengambil koin, piala, dan *freekey* (kunci bebas) tersebut. seperti yang terlihat pada Gambar 3.5 berikut:



Gambar 3.5 Penerapan FSM Pada *Player*

2. Penerapan FSM pada karakter musuh

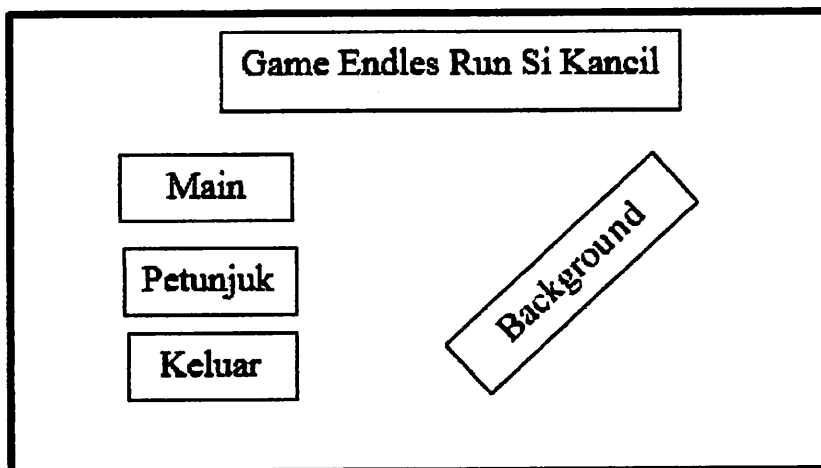
Penerapan alur FSM pada karakter musuh memiliki sifat jika karakter *player* menjauh maka musuh akan mengejar, jika *player* terhambat rintangan musuh akan mendekat pada jarak 300-400 *pixel* dari musuh, maka musuh akan mengikuti gerak dari karakter *player*, dan ketika karakter musuh menyentuh *player* maka musuh akan menangkap *player*. Seperti yang terlihat pada Gambar 3.6 berikut:



Gambar 4.6. Penerapan FMS Pada Musuh

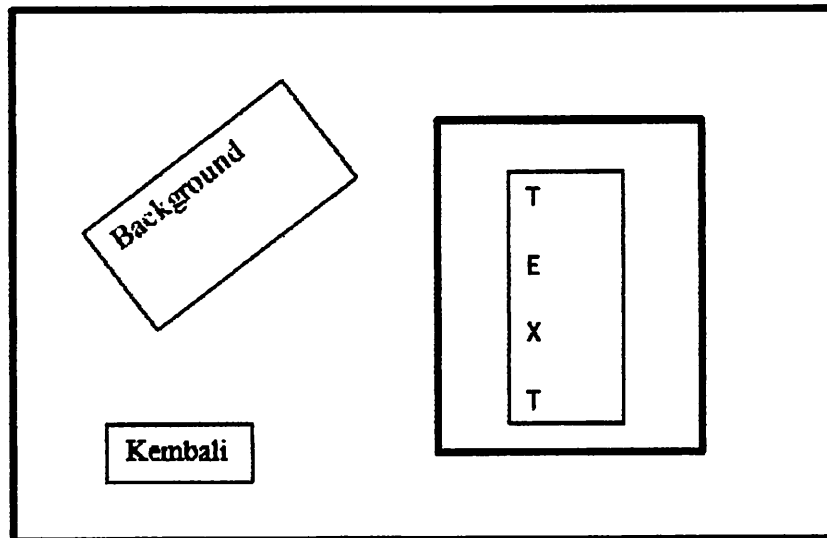
3.2.8 Desain GUI (Graphical User Interface)

Desain awal dari tampilan di dalam *game* meliputi menu utama. Main, Petunjuk dan Keluar. Pada Gambar 3.7 merupakan desain tampilan menu utama.



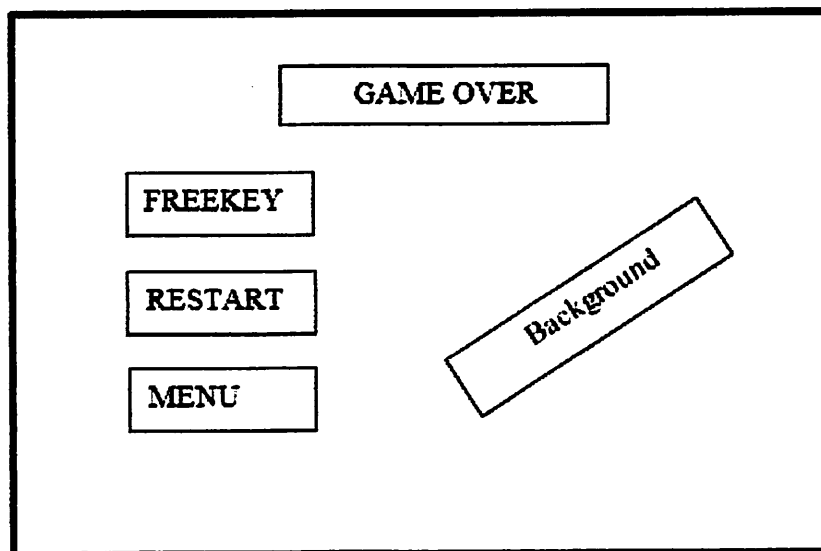
Gambar 3.7 Tampilan Desain GUI Menu Utama

Pada Gambar 3.8 berikut merupakan desain tampilan Petunjuk untuk membantu cara bermain *game Endles Runing Si Kancil*.



Gambar 3.8 Tampilan Desain GUI Menu Petunjuk

Pada Gambar 3.9 berikut merupakan desain tampilan *menu Game Over* pada game *Endles Runing Si Kancil*.





Gambar 3.9 Tampilan Desain GUI Menu Game Over

3.2.9 Desain Karakter

Desain awal dari karakter meliputi karakter utama dan musuh disetiap *stage*, pembuatan karakter menggunakan Addobe Blender. Karakter utama dalam game *Endles Runing Si Kancil* ditunjukkan pada Tabel 3.1


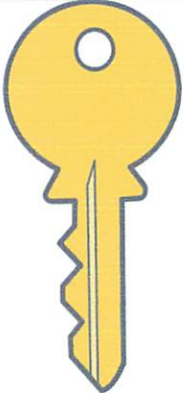

Tabel 3.1. Perancangan Karakter


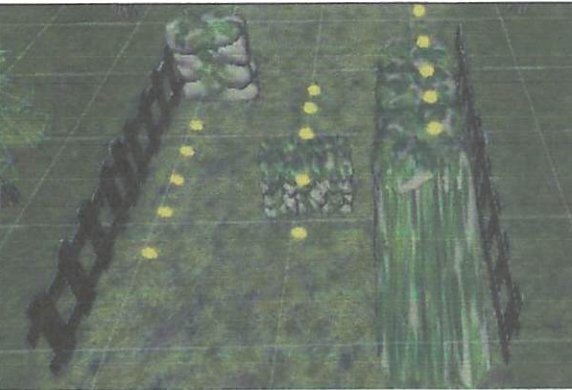
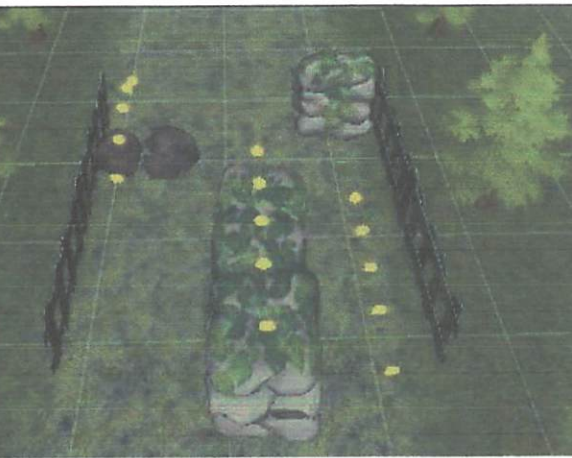
No	Gambar	Keterangan
1		Karakter utama pada <i>game endless running</i> si kancil , yang akan berlari dari kejaran musuh (<i>enemy</i>)
2		Karakter musuh yang akan mengejar <i>player</i> pada <i>game endkes running</i> si kancil




3.2.10 Perancangan *Environment*

Pengenalan *level environment* adalah pembahasan mengenai lingkungan yang terdapat pada *game Endles Runing Si Kancil*. Berikut penjabaran *environment* yang dapat dilihat pada Table 3.2 berikut :

Tabel 3.2. Perancangan *enviromtment*

No	Gambar	Keterangan
1		<p>Koin, jika koin <i>player</i> lebih dari 15 maka akan naik level atau tambah cepat</p>
2		<p>Freekey, jika <i>player</i> tertangkap maka <i>player</i> bisa menggunakan kunci tersebut untuk bebas</p>
3		<p>Piala yang terdapat pada akhir masing – masing stage</p>

4		Rintangan 1 yang akan menghadang lari dari <i>player</i>
5		Rintangan 2 yang akan menghadang lari dari <i>player</i>
6		Rintangan 3 yang akan menghadang lari dari <i>player</i>

7	 A top-down view of a game level. A path of yellow markers leads through a grassy area. There are several dark, rounded obstacles and two black metal fences on either side of the path.	<p>Rintangan 4 yang akan menghadang lari dari <i>player</i></p>
8	 A top-down view of a game level. A path of yellow markers leads through a grassy area. There are several dark, rounded obstacles and two black metal fences on either side of the path. A large, rectangular, textured obstacle is in the center of the path.	<p>Rintangan 5 yang akan menghadang lari dari <i>player</i></p>
9	 A third-person view of a player character in a game. The player is standing in a wide, grassy field with many small trees. In the background, there are several tall, pointed mountains under a clear blue sky. Other player characters are visible in the distance.	<p>Tampilan <i>stage 1</i> secara keseluruhan, yang mengambil suasana perkebunan</p>

10		Tampilan <i>stage 2</i> secara keseluruhan yang mengambil suasana laut
----	---	--

BAB IV

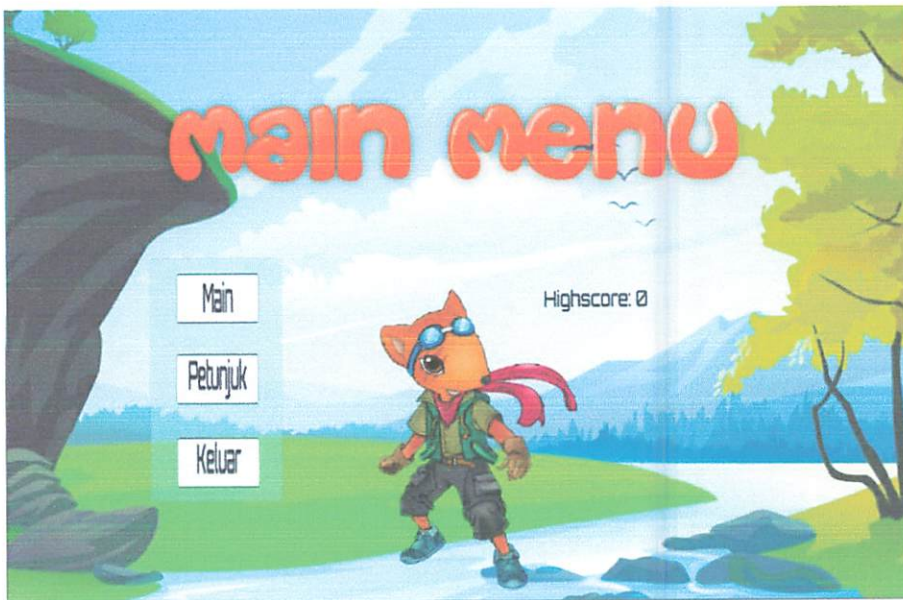
IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Implementasi sistem adalah proses penerapan rancangan sistem yang telah dibuat menjadi suatu aplikasi yang bisa dijalankan pada kenyataannya. Disamping itu, implementasi ini juga berfungsi untuk mengetahui tingkat keberhasilan dari rancangan yang telah dibuat. Implementasi aplikasi ini dibagi menjadi beberapa bagian diantaranya :

4.1.1 Tampilan Menu Utama

Tampilan menu utama adalah tampilan awal yang muncul pada saat membuka *Game Endles Runing Si Kancil*. Pada tampilan awal ini berisi tombol main untuk memulai *game*. Lalu ada tombol keluar untuk keluar dari permainan, tombol petunjuk berisi tentang cara memainkan *game*. Seperti pada Gambar 4.1.



Gambar 4.1 Pengujian Tampilan Menu Utama

4.1.2 Tampilan Menu Petunjuk

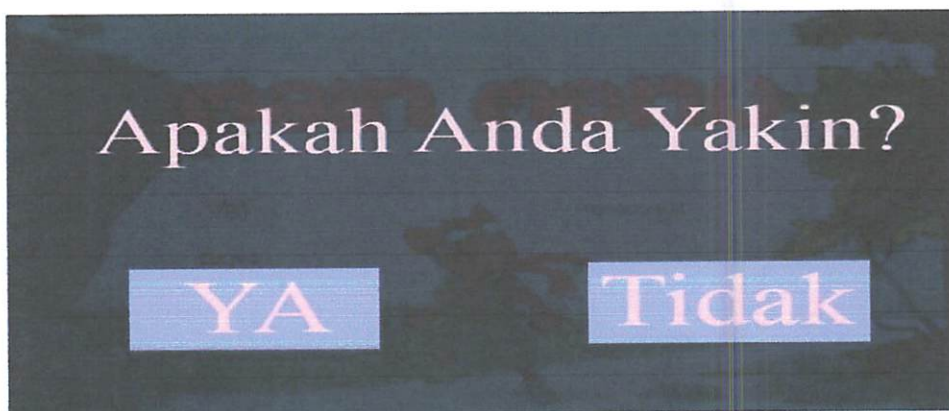
Tampilan menu Petunjuk adalah tampilan menu untuk melihat seluruh navigasi untuk memainkan *Game Endles Runing Si Kancil*. Adapun tampilan dari menu Kontrol seperti pada Gambar 4.2. berikut:



Gambar 4.2 Pengujian Tampilan Menu Petunjuk

4.1.3 Tampilan Menu Keluar

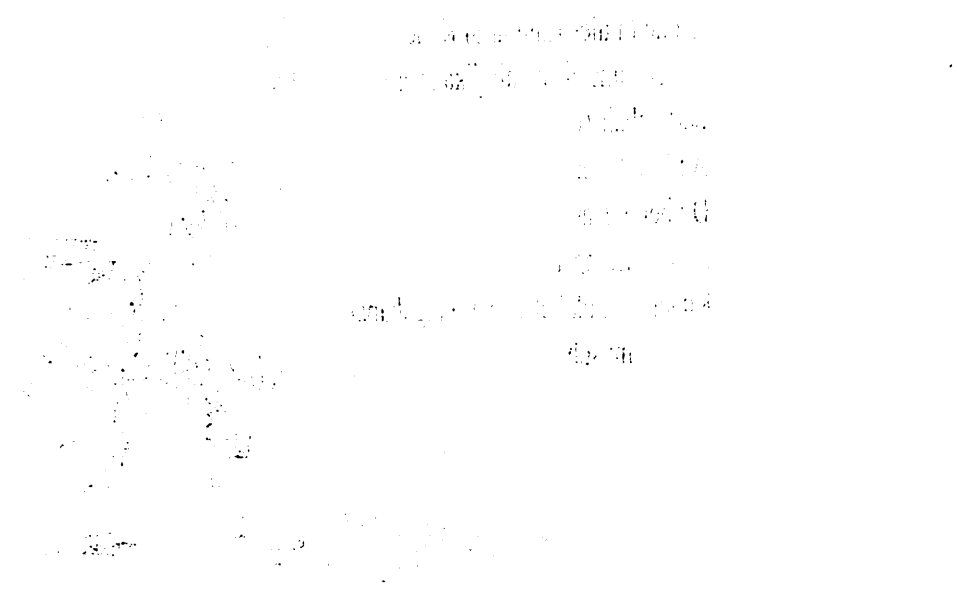
Pengujian menu keluar menampilkan tentang menu untuk keluar dari dunia game. Seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4.3 Pengujian Tampilan Menu Keluar

Definieren Sie die Begriffe: 1.1)

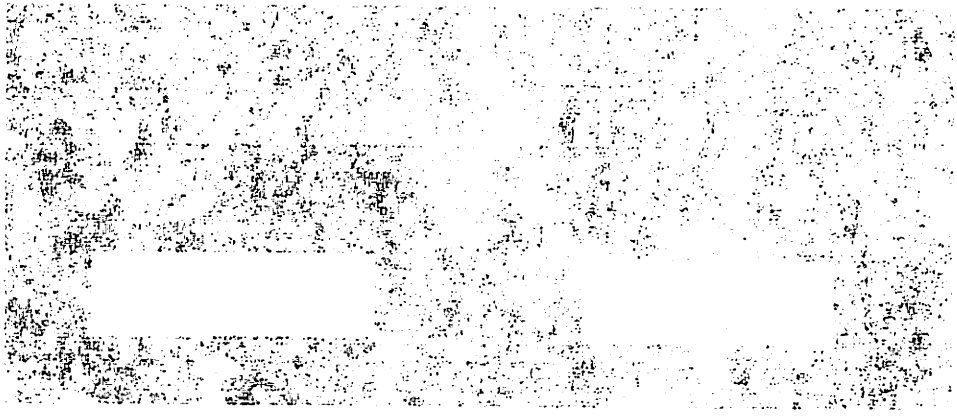
... (faded text) ...



... (faded text) ...

... (faded text) ...

... (faded text) ...



... (faded text) ...

4.2. Pengujian Game play

Pengujian *gameplay* adalah pengujian bagaimana *Game Endles Runing Si Kancil* berjalan sesuai dengan rancangan sistem yang telah dibuat.

4.2.1 Pengujian Karakter Dengan Dunia Game

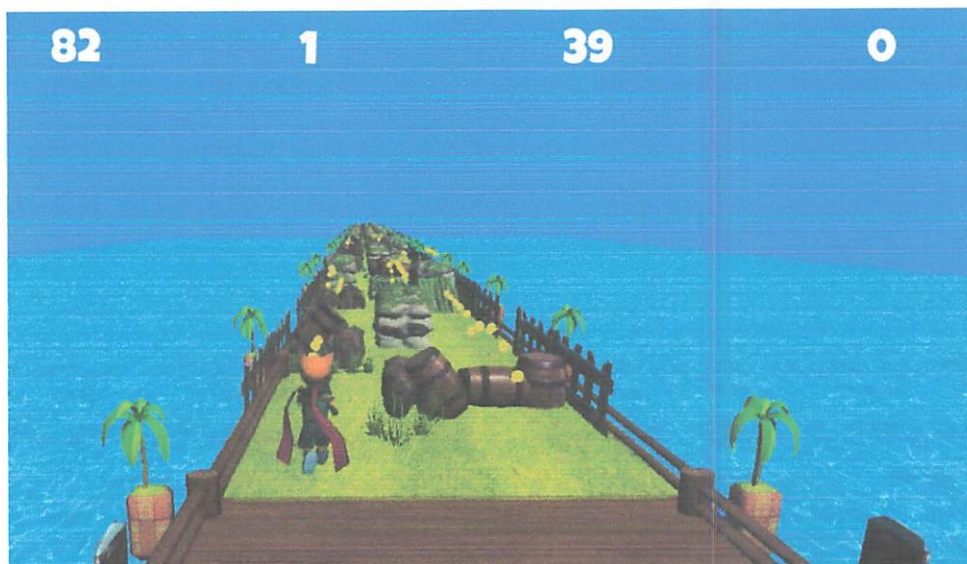
Pengujian karakter dengan dunia *game* menggambarkan interaksi antara karakter dengan lingkungan di dalam *game*. Pada gambar diatas menunjukan bahwa karakter dapat berjalan dengan baik diatas *ground* yang dibuat. Seperti yang ditunjukkan pada Gambar 4.3 berikut:



Gambar 4.4 Pengujian Karakter Dalam Game

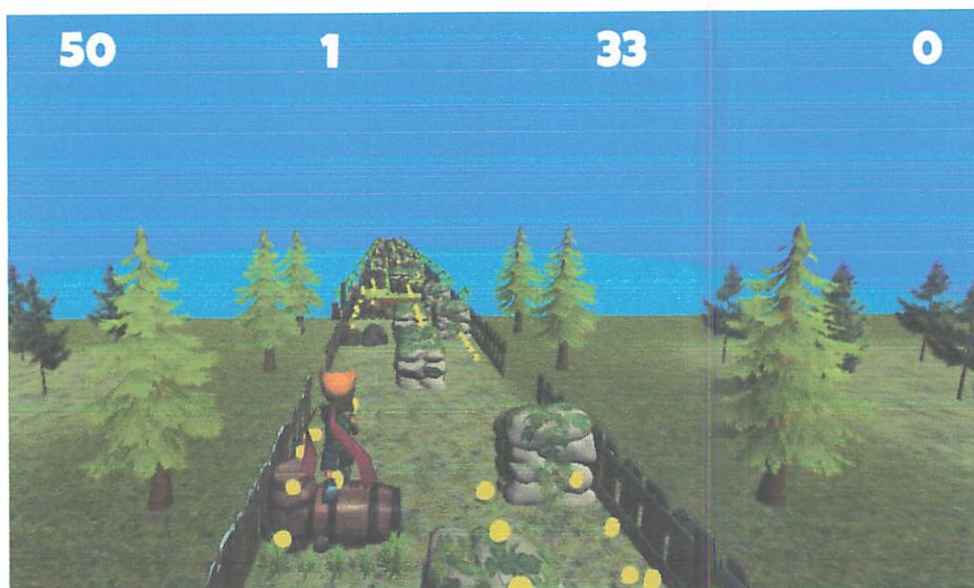
4.2.2 Pengujian Animasi Karakter

Pengujian animasi karakter menggambarkan ketika karakter berlari, dan melompat. Seperti yang ditunjukkan pada Gambar 4.5 berikut :



Gambar 4.5 Pengujian Animasi Karakter Berlari

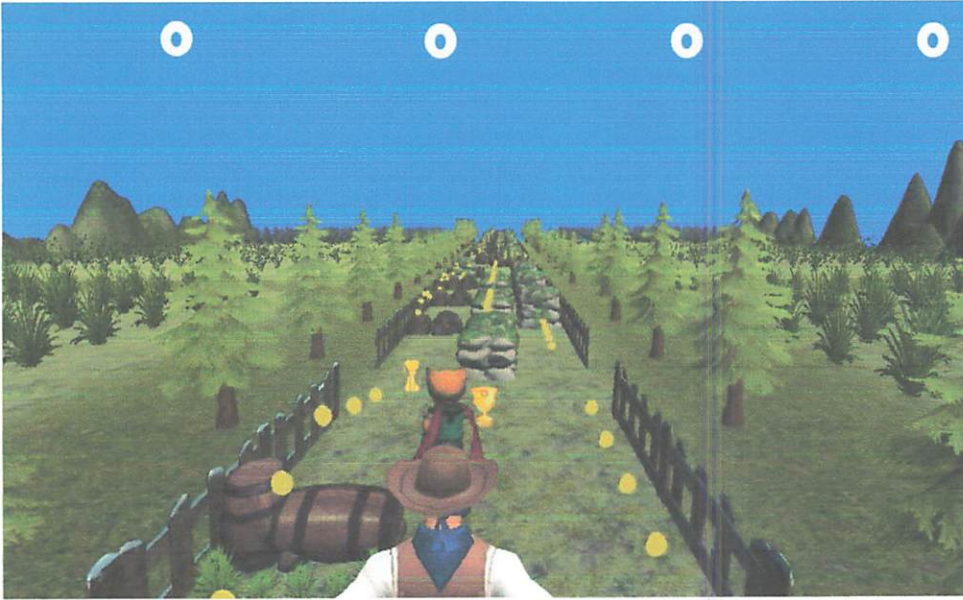
Pada Gambar 4.6 berikut adalah gambar animasi karakter pada saat melompat



Gambar 4.6 Animasi Karakter Melompat

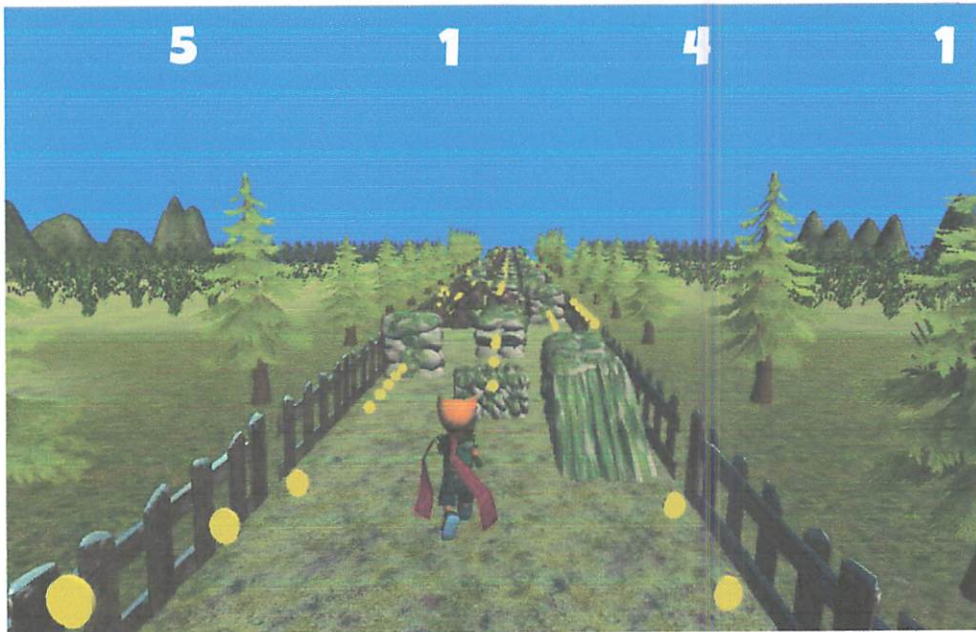
4.2.3 Pengujian Karakter Pangambilan Koin

Pengujian karakter pada saat pengambilan koin, piala, dan *freekey* (kunci kebebasan). Seperti pada Gambar 4.7 berikut:



Gambar 4.7 Sebelum Mengambil Koin, Piala, dan Kunci

Pada gambar 4.8 berikut adalah gambar pada saat karakter ketika sudah mengambil koin, piala, dan *freekey* (Kunci bebas)



Gambar 4.8 Sesudah mengambil Koin, Piala, dan Kunci

4.2.4 Pengujian Tombol Dalam Game

Pengujian menu control menampilkan penjelasan tentang control apa saja yang digunakan dengan jelas untuk menjalankan game. Seperti yang ditunjukkan pada gambar 4.9 berikut:



Gambar 4.9 Pengujian Kontrol Dalam Game

4.2.5 Pengujian Tampilan Game Over

Pengujian tampilan Gameover menampilkan tentang sebuah tanda bahwa *player* telah gagal dalam memainkan atau kalah. Seperti yang ditunjukkan pada gambar 4.10. berikut :



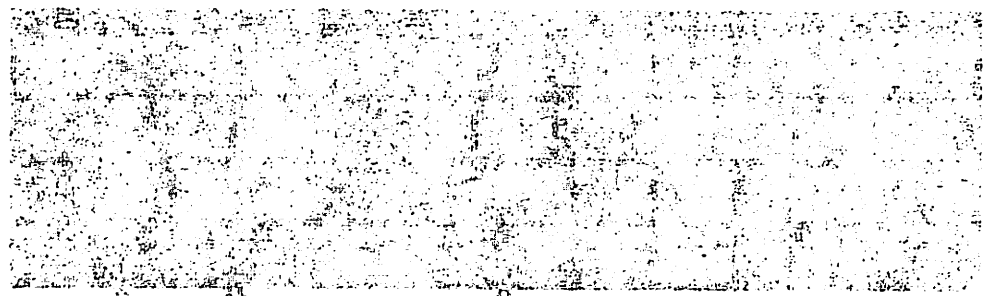
Gambar 4.10. Pengujian Tampilan Game Over

... ..

... ..

... ..

... ..



... ..

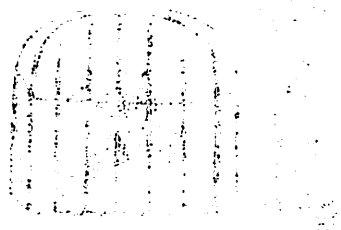
... ..

... ..

... ..

... ..

... ..



... ..

4.3. Pengujian Artificial Intelligence (AI)

Pengujian AI adalah pengujian mengenai fungsi yang berkaitan dengan AI (*Artificial Intelligence*) yang ada dalam *Game Endles Runing Si Kancil*. Hasil pengujian dari AI *Finite State Machine* dapat dilihat pada Tabel 4.1 berikut :

Tabel 4.1. Pengujian AI (*Artificial Intelligence*)

NO	FUNGSI	KEADAAN	KEJADIAN	AKSI	HASIL
1	Pada karakter <i>Player</i>	Mengambil koin	Player pada saat menyentuh koin	Mengambil Koin	Sesuai
2	Pada karakter <i>Player</i>	Mengambil piala	Player pada sata menyentuh piala	Mengambil piala	Sesuai
3	Pada karakter <i>Player</i>	Mengambil kunci	Player pada saat menyentuh kunci	Mengambil kunci	Sesuai

Berdasarkan Tabel 4.1. disimpulkan bahwa semua AI (*Artificial Intelligence*) yang ada dalam *Game Endles Runing Si Kancil* berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.4. Pengujian FSM (*Finite State Machine*)

Pengujian *Finite State Machine* adalah pengujian mengenai fungsi yang berkaitan dengan FSM (*Finite State Machine*) yang ada dalam *Game Endles Runing Si Kancil*. Hasil pengujian dari AI *Finite State Machine* dapat dilihat pada Tabel 4.2 berikut :

Tabel 4.2. Pengujian FSM (*Finite State Machine*)

NO	FUNGSI	KEADAAN	KEJADIAN	AKSI	HASIL
1	Pada Karakter Musuh	Mengejar Player	Player pada jarak 400 pixel dari musuh	Musuh mengikuti dan menangkap	Sesuai
			Player pada jarak >400 pixel dari musuh	player Musuh kembali mengejar	

Berdasarkan Tabel 4.2. disimpulkan bahwa semua FSM (*Finite State Machine*) yang ada dalam *Game Endles Runing Si Kancil* berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.5. Pengujian Tombol

Pengujian tombol adalah pengujian mengenai fungsi tombol yang ada pada *game Endles Runing Si Kancil* pada saat dijalankan (*Run*). Hasil pengujian dari tombol dapat dilihat pada Tabel 4.3 berikut :

Tabel 4.3. Pengujian Tombol

NO	FUNGSI	HASIL
1	Button Main	Sesuai
2	Button Petunjuk	Sesuai
3	Button Keluar	Sesuai
4	Button Restat	Sesuai

Berdasarkan Tabel 4.3. disimpulkan bahwa semua tombol yang ada dalam *Game Endles Runing Si Kancil* berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.6 Pengujian Back Sound Game

Pengujian *Back Sound Game* adalah pengujian *sound* yang ada dalam *Game Endles Runing Si Kancil* pada saat dimainkan. Hasil pengujian *back sound* pada *game* dapat dilihat pada Tabel 4.4 berikut :

Table 4.4. Pengujian Back Sound Game

NO	FUNGSI	HASIL
1	Menu awal game	Sesuai
2	Game dimainkan	Sesuai
3	Player mengambil koin	Sesuai
4	Player mengambil piala	Sesuai
5	Player mengambil kunci	Sesuai
6	Player game over	Sesuai

Berdasarkan Tabel 4.4. disimpulkan bahwa semua *Back sound* yang ada dalam *Game Endles Runing Si Kancil* berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan

4.7 Pengujian Performance

Pengujian *performance* adalah pengujian yang dilakukan pada kinerja atau respon perangkat keras. Pengujian *performance* dimaksudkan untuk mengetahui apakah aplikasi berjalan pada suatu perangkat keras tertentu dengan spesifikasi perangkat keras yang berbeda-beda. Pengujian dilakukan menggunakan 5 komputer dengan spesifikasi berbeda dan menggunakan monitor beresolusi 1366 x 768 *pixel*. Hasil dari pengujian *performance* dari *Game Endles Runing Si Kancil* terdapat pada Tabel 4.5 Berikut :

Tabel 4.5. Pengujian Performance

No	Processor	RAM	VGA	OS	Keterangan
1	Intel core i7-Q740 1.73 GHz	4 GB	2 GB	Windows 10 Pro	Berjalan Lancar
2	Intel core i3- 4010U 1.70GHz	4 GB	4 GB	Windows 10 Pro	Berjalan Lancar
3	AMD A8-4500M APU 1.96GHz	4 GB	4 GB	Windows 8.1 Pro	Berjalan Lancar
4	AMD 16-3400M 2.90GHz	8 GB	1 GB	Windows 7 Ultimate	Berjalan Lancar

5	Intel core i3-2328M 2.20GHz	4 GB	1 GB	Windows 8.1 Pro	Berjalan Lancar
---	-----------------------------	------	------	-----------------	-----------------

Dari Tabel 4.5 dapat disimpulkan bahwa *Game Endles Runing Si Kancil* dapat dijalankan pada komputer dengan RAM 2 GB hingga 8 GB dan monitor beresolusi 1366 x 768 pixel, juga dengan menggunakan Windows 7,8.1 dan Windows 10

4.8 Pengujian Control Player

Pengujian *control player* adalah pengujian fungsi dari setiap tombol yang sudah diterapkan untuk menggerakkan karakter utama. Hasil pengujian *control player* dapat dilihat pada Tabel 4.6 berikut :

Tabel 4.6. Pengujian Control Player

Tombol	Fungsi	Hasil
➡	Menggerakkan player ke kanan	Sesuai
⬅	Menggerakkan player ke kiri	Sesuai
Spasi	Untuk menyerang dengan pedang	Sesuai

Dari Tabel 4.6 menunjukkan bahwa semua fungsi dari *control player* Berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.9 Pengujian Terhadap Pengguna

Pengujian dilakukan untuk mengetahui apakah Game sudah berjalan dengan baik atau belum. Pengujian dilakukan terhadap 10 responden. Kuisisioner berisi pertanyaan tentang game yang dibuat penulis. Hasil dari pertanyaan terhadap responden dapat dilihat pada Tabel 4.7 berikut :

Tabel 4.7. Pengujian Terhadap Pengguna

Pertanyaan	Penilaian				
	1	2	3	4	5
Bagaimana tampilan desain	0%	10%	10%	80%	0%

karakter pada <i>game</i> ?					
Bagaimana tampilan desain animasi pada <i>game</i> ?	0%	20%	10%	70%	0%
Bagaimana tampilan menu pada <i>game</i> ?	0%	0%	30%	70%	0%
Bagaimana kontrol <i>player</i> saat memainkan <i>game</i> ?	0%	10%	30%	20%	40%
Bagaimana gerakan musuh saat mengejar <i>player</i> ?	0%	0%	30%	60%	10%
Apakah alur cerita pada <i>game</i> sudah baik?	0%	0%	70%	10%	20%
Bagaimana aspek hiburan pada <i>game</i> ?	0%	0%	60%	30%	10%

Dari hasil pengujian pengguna 7 dari 10 orang menyatakan gameplay game “Endles Runing Si Kancil” menarik dengan presentasi 70% dan sisa 3 lainnya menyatakan kurang menarik dengan persentase 30%. Untuk genre platfromer 7 dari 7 orang menyatakan menarik dengan presentasi 70%, 2 menyatakan kurang menarik dengan persentase 20% , dan 1 menyatakan tidak menarik dengan presentasi 10%, dan untuk desain karakter 6 dari 10 menyatakan menarik dengan persentase 60%, 4 menyatakan kurang menarik dengan persentase 40%.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah pembuatan *game Endles Runing Si Kancil*, maka penulis dapat mengambil kesimpulan :

1. Hasil pengujian metode menunjukkan bahwa implementasi *finite state machine* pada *game* berjalan dengan baik dengan presentase 70% dan 30% menyatakan cukup.
2. Hasil pengujian kontrol pada *player* 20% menyatakan kurang, 20% menyatakan cukup, dan 60% menyatakan baik.
3. Berdasarkan hasil kuisisioner menunjukkan bahwa desain karakter, desain animasi, dan desain menu 20% menyatakan cukup 80% menyatakan baik.

5.2 Saran

Adapun saran sebagai acuan terhadap penelitian atau pengembangan selanjutnya, diantaranya :

1. Dapat dikembangkan menjadi *game* berbasis *mobile* sehingga dapat dimainkan pada sistem operasi *android*.
2. Menambahkan beberapa atribut, *stage* dan karakter *player* sehingga *user* dapat memilih karakter sesuai dengan keinginan.
3. Menambah rintangan aktif pada masing-masing *stage* agar permainan lebih menantang dan menarik.

DAFTAR PUSTAKA

- Abdullah, D., Reza, B., & Erliana, C. I. (2015). Game *Endles Run* Berbasis Role Playing Game Dengan Metode Finite State Machine. *Universitas Malikussaleh*, 1-11.
- Setiawan, (2013). Agen Cerdas Kompetitif Berbasis Finite State Machine Dalam *Game Endles Runner*. *Techno.Com, Vol.9 No.2*, 53-61.
- Rahadian, M. F., Suyatno, A., & Maharani, S. (2016). PENERAPAN METODE FINITE STATE MACHINE PADA GAME "THE RELATIONSHIP". *Jurnal Informatika Mulawarman*, 14-22.
- Rostianingsih, S., Budhi, G. S., & Wijaya, H. K. (2013). GAME SIMULASI FINITE STATE MACHINE UNTUK PERTANIAN DAN PETERNAKAN. *Universitas Kristen Petra*, 1-6.
- Widyatama, (2015). *Endles Runner. Player Controller. Control Game. Institut Teknologi Bandung*, 1-13.

LAMPIRAN

IMPERIAL

1. Source code

Source code menu utama

```
using UnityEngine;

using UnityEngine.UI;

using System.Collections;

public class Menu : MonoBehaviour {

    public Text highscoreText;

    // Use this for initialization

    [SerializeField]

    private AudioSource sound;

    void Start ()

    {

        sound.Play();

        highscoreText.text = "Highscore: " +

(int) (PlayerPrefs.GetFloat("Highscore"));

    }

    // Update is called once per frame

    void Update () {

    }

}
```

```

public void Play()
{
    Application.LoadLevel ("Skripsi");
}

public void keluar()
{
    Application.LoadLevel (Application.loadedLevel);
}

public void help()
{
    GameObject.Find ("petunjuk").GetComponent<Image>
().enabled = true;

    GameObject.Find
("petunjuk").GetComponentInChildren<Text> ().enabled = true;
//      GameObject.Find ("petunjuk").GetComponent<Image>
().enabled = false;
//      GameObject.Find
("petunjuk").GetComponentInChildren<Text> ().enabled = false;
}
}

```

Source code player kontroler

```

using UnityEngine;

using System.Collections;

```



```

public class PlayerMotor : MonoBehaviour
{
    private CharacterController controller;

    private Vector3 moveVector;

    [SerializeField]
    private float speed = 10.0f;

    private float verticalVelocity = 0.0f;

    private float gravity = 12.0f;

    private float animationDuration = 4.0f;

    private float startTime;

    Animator anim;

    private int lane;

    Vector3 kembali,kembali2, kembali3;

    private bool isDead = false ;

    // Use this for initialization

    void Start ()
    {
        kembali = this.transform.position;

        kembali2 = GameObject.Find
("Cowboy").transform.position;

        kembali3 = GameObject.Find
("TileManager").transform.position;

        lane = 0;

        controller = GetComponent<CharacterController> ();
    }
}

```

```

        startTime = Time.time;

        anim = GetComponent<Animator>();

        return;

    }

    // Update is called once pe

    void Update () {

        anim.SetBool ("isJump",false);

        if(Input.GetKeyDown(KeyCode.Space)){

            anim.SetBool ("isJump", true);

            //GetComponent<Rigidbody>().AddForce(Vector3.up * 2000);

            controller.Move(new Vector3 ( 0, 200, 0)*
Time.deltaTime);

        }

        //////////////////////////////////////

        if (Input.GetKeyDown (KeyCode.UpArrow)){

            anim.SetBool("isJump", true);

            Invoke("stopJumping", 0.1f);

        }

//

        if (isDead)

            return;

        if (Time.time - startTime < animationDuration)

        {

```

```
        controller.Move(Vector3.forward * speed *
Time.deltaTime);

    }

    moveVector = Vector3.zero;

    if (controller.isGrounded)
    {
        verticalVelocity = -0.5f;
    }
    else
    {
        verticalVelocity -= gravity * Time.deltaTime;
    }

    //x - Left and Right

    moveVector.x = Input.GetAxisRaw("Horizontal") *
speed;

    //y - Up and Down

    moveVector.y = verticalVelocity;

    //z - forward and Backward

    moveVector.z = speed;

    controller.Move (moveVector * Time.deltaTime);

}

// private void onControllerColliderHit
(ControllerColliderHit hit)
```

```
// {  
  
//     if (hit.point.z > transform.position.z +  
controller.radius)  
  
//         Death ();  
  
// }  
  
private void Death()  
  
{  
  
    isDead = true;  
  
}  
  
public void SetSpeed(float modifier)  
  
{  
  
    speed = 15.0f + modifier;  
  
}  
  
void stopJumping()  
  
{  
  
    anim.SetBool("Jumping", false);  
  
}  
  
void OnTriggerEnter(Collider other)  
  
{  
  
    if (other.CompareTag ("end")) {  
  
        transform.position = kembali;  
  
        GameObject.Find  
("Cowboy").gameObject.transform.position = kembali2;  
  
    }  
  
}
```

```
        GameObject.Find
("TileManager").gameObject.transform.position = kembali3;

        }

    }

}
```

Source code enemy (musuh)

```
using UnityEngine;

using System.Collections;

public class enemy : MonoBehaviour {

    public Transform orang; // Target Object to follow

    public float speed = 10.0F; // Enemy speed

    private Vector3 coboy;

    private bool challenged = false; // If the enemy is
    Challenged to follow by the player

    [SerializeField]

    private Canvas deathScene;

    float maju=10.0f;

    // Use this for initialization

    void Start () {

    }

}
```

```

// Update is called once per frame

void Update ()
{
    transform.Translate (Vector3.forward * (speed
*Time.deltaTime));

    if (challenged) {
        coboy = orang.transform.position -
transform.position;

        coboy = coboy.normalized;    // Get Direction
to Move Towards

        transform.Translate (coboy * speed,
Space.World);
    }
}

// Will be triggered as soon as player would touch the
Enemy Object

void OnTriggerEnter( Collider other)
{
    if (other.CompareTag ("Player"))
    {
        challenged=true;

        GameObject.Find
("DeathScene").GetComponent<Mati>().enabled = true;

        deathScene.enabled = true;
    }
}

```

```
        }

    }

    public void SetSpeed(float modifier)
    {
        speed = 15.0f + modifier;
    }
}
```

Source code tile manager (Rintangan)

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class TileManager : MonoBehaviour
{
    public GameObject[] tilePrefabs;

    private Transform playerTransform;
    private float spawnZ = -6.0f;
    private float tileLength = 30.0f;
    private float safeZone = 200.0f;
    private int amnTilesOnScreen = 30;
```

```
private int LastPrefabIndex = 0;

private List<GameObject> activeTiles;

// Use this for initialization

private void Start () {

    activeTiles = new List<GameObject> ();

    //playerTransform =
gameObject.FindGameObjectsWithTag ("Player").transform;

    playerTransform = GameObject.FindGameObjectWithTag
("Player").transform;

    for (int i = 0; i < amnTilesOnScreen; i++)
    {

        if (i < 2)

            SpawnTile (0);

        else

            SpawnTile ();

    }

}

// Update is called once per frame

private void Update () {
```



```
        if (playerTransform.position.z - safeZone > (spawnZ
- amnTilesOnScreen * tileLength))

        {

            SpawnTile ();

            //DeleteTile();

        }

    }

    private void SpawnTile(int prefabIndex = -1)

    {

        GameObject go;

        if (prefabIndex == -1)

            go = Instantiate (tilePrefabs
[RandomPrefabIndex ()])as GameObject;

        else

            go = Instantiate (tilePrefabs
[prefabIndex])as GameObject;

        //go = Instantiate (tilePrefabs
[RandomPrefabIndex()]) as GameObject;

        //go.transform.SetParent (transform);

        go.transform.parent = transform;

        go.transform.position = Vector3.forward * spawnZ;

        spawnZ += tileLength;

        activeTiles.Add (go);

    }
```

```
// private void DeleteTile()
// {
//     Destroy (activeTiles [0]);
//     activeTiles.RemoveAt (0);
// }
private int RandomPrefabIndex()
{
    if (tilePrefabs.Length <= 1)
        return 0;

    int randomIndex = LastPrefabIndex;
    while (randomIndex == LastPrefabIndex)
    {
        randomIndex = Random.Range (0,
tilePrefabs.Length);
    }
    LastPrefabIndex = randomIndex;
    return randomIndex;
}
}
```

Source code koin

```
using UnityEngine;
```

```
using System.Collections;

public class Coin : MonoBehaviour {

    int CoinCounter;

    GameObject Koin;

    private GameObject textKoin;

    [SerializeField]

    private AudioSource suara;

    // Use this for initialization

    void Start () {

        textKoin = GameObject.Find ("TextCoin");

        CoinCounter = 0;

        ;

    }

    // Update is called once per frame

    void Update ()

    {

        Koin = GameObject.FindGameObjectWithTag("Coin");

        textKoin.GetComponent<GUIText> ().text =

        ((int)CoinCounter).ToString();
```

```

    }

    void OnTriggerEnter(Collider hit){

        if (hit.tag == "Coin")

        {

            Destroy (hit.gameObject);

            //hit.gameObject.GetComponent<MeshRenderer>().enabled =
false;

            CoinCounter++;

            suara.Play();

            Debug.Log ("coin counter = " + CoinCounter);

        }

    }

}

```

Source code terophy (Piala)

```

using UnityEngine;

using System.Collections;

public class Teropy : MonoBehaviour {

    int PialaCounter;

    GameObject piala;

```

```
private GameObject textPiala;

[SerializeField]

private AudioSource bunyi;

// Use this for initialization

void Start () {

    textPiala = GameObject.Find ("TextPiala");

    PialaCounter = 0;

}

// Update is called once per frame

void Update ()

{

    piala = GameObject.FindGameObjectWithTag("piala");

    textPiala.GetComponent<GUIText> ().text =

((int)PialaCounter).ToString();

}

void OnTriggerEnter(Collider hit){

    if (hit.tag == "piala")

    {

        Destroy (piala.gameObject);

    }

}
```

```
        //hit.gameObject.GetComponent<MeshRenderer>().enabled =
false;

        PialaCounter++;

        //Debug.Log ("coin counter = " +
PialaCounter);
    }
}
}
```

Source code score

```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public class Score : MonoBehaviour {

    private int difficultyLevel = 1;

    private int MaxDifficultyLevel = 10;

    private int skoreToNextLevel = 10;

    private bool isDead = false ;

    //private CharacterController controller;

    public Text SkoreText;

    public DeathMenu deathMenu;
```

```
float Skore;

GameObject text;

//GameObject text2;

GameObject Player;

GameObject Enemy;

// Use this for initialization

void Start () {

    Skore = 0.0f;

    text = GameObject.Find ("Score");

    //text2 = GameObject.Find ("Score2");

    Player = GameObject.FindGameObjectWithTag("Player");

    Enemy = GameObject.FindGameObjectWithTag ("enemy");

}

// Update is called once per frame

void Update () {

    if (Skore >= skoreToNextLevel)

        LevelUp ();

    Skore += Time.deltaTime * difficultyLevel;

    text.GetComponent<GUIText> ().text =

((int)Skore).ToString();

    PlayerPrefs.SetFloat ("Highscore", Skore);
```

```
        //text2.GetComponent<GUIText> ().text =
        ((int)Skore).ToString();

    }

    void LevelUp()
    {
        if (difficultyLevel == MaxDifficultyLevel)
            return;

        skoreToNextLevel *= 2;

        difficultyLevel++;
//
        GetComponent<PlayerMotor>().SetSpeed(difficultyLevel);

        Player.GetComponent<PlayerMotor> ().SetSpeed
(difficultyLevel);

        Enemy.GetComponent<enemy> ().SetSpeed
(difficultyLevel);

        Debug.Log (difficultyLevel);
    }

    public void onDeath()
    {
        isDead = true;

        PlayerPrefs.SetFloat ("Highscore", Skore);

        Debug.Log(PlayerPrefs.GetFloat ("Highscore",
Skore));
//
        deathMenu.ToggleEndMenu (Skore);
    }
}
```



```
}
```

Source code freekey (kunci bebas)

```
using UnityEngine;

using System.Collections;

public class FreeKey : MonoBehaviour {

    public int PialaCounter;

    GameObject piala;

    private GameObject textKey;

    // Use this for initialization

    void Start () {

        textKey = GameObject.Find ("TextKunci");

        PialaCounter = 0;

    }

    // Update is called once per frame

    void Update ()

    {

        piala = GameObject.FindGameObjectWithTag("key");

        textKey.GetComponent<GUIText> ().text =

        ((int)PialaCounter).ToString();

    }

}
```

```

    }

    void OnTriggerEnter(Collider hit){

        if (hit.tag == "key")

        {

            Destroy (piala.gameObject);

            //hit.gameObject.GetComponent<MeshRenderer>().enabled =
false;

            PialaCounter++;

            //Debug.Log ("coin counter = " +
PialaCounter);

        }

    }

}

```

Source code ketika mati

```

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class Mati : MonoBehaviour {

    string score1, score2, teropi, key;

    public Text scoreVal;

    public Text coinVal;

```

```
public Text pialaVal;

public Button free;

[SerializeField]
Canvas deathScene;

[SerializeField]
private FreeKey kunci;

public GameObject enemy;

// Use this for initialization
void Start () {

}

// Update is called once per frame
void Update ()
{
    Time.timeScale = 0;

    score1 = GameObject.Find
("Score").GetComponent<GUIText> ().text;

    score2 = GameObject.Find
("TextCoin").GetComponent<GUIText> ().text;

    teropi = GameObject.Find
("TextPiala").GetComponent<GUIText> ().text;
```

```
        key = GameObject.Find
("TextKunci").GetComponent<GUIText> ().text;

        if (key == "1")
        {
            free.interactable = true;
        }
        else
        {
            free.interactable =false;
        }

        scoreVal.text = score1;

        coinVal.text = score2;

        pialaVal.text = teropi;

        GameObject.Find ("Player 1").GetComponent<Score>
().enabled = false;

        GameObject.Find ("Player 1").GetComponent<Coin>
().enabled = false;

        GameObject.Find ("Player 1").GetComponent<Teropy>
().enabled = false;

        //GameObject.Find ("Player
1").GetComponent<PlayerMotor> ().enabled = false;
    }

    public void lanjutGame()
```

```

{

    Time.timeScale = 1;

    deathScene.enabled = false;

    GameObject.Find ("DeathScene").GetComponent<Mati>
().enabled = false;

    key = "0";

    kunci.PialaCounter = 0;

    enemy.transform.position = new Vector3
(enemy.transform.position.x, enemy.transform.position.y,
enemy.transform.position.z - 20);

    GameObject.Find ("Player 1").GetComponent<Score>
().enabled = true;

    GameObject.Find ("Player 1").GetComponent<Coin>
().enabled = true;

    GameObject.Find ("Player 1").GetComponent<Teropy>
().enabled = true;

}
}

```

Source code death menu (Menu Game Over)

```

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class DeathMenu : MonoBehaviour {

```

```
// Update is called once per frame

void Update ()
{

}

public void Restart()
{

    //SceneManager.LoadScene
(SceneManager.GetActiveScene ());

    Application.LoadLevel ("Skripsi");

}

public void ToMenu()
{

    //Application.LoadScene ("Menu");

    Application.LoadLevel ("Menu");

}
```

2. Lembar Persetujuan Dosen Pembimbing I



PT BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

Kampus I : Jl. Bendungan Sigura-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

Malang, 26 September 2016

Nomor : ITN-806/IX.INF/TA/2016
Lampiran : ---
Perihal : *Bimbingan Skripsi*
Kepada : Yth. Bpk/Ibu Karina Auliasari, ST.M.Eng
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : MUSYAFFA AL HAFFI
Nim : 1318109
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

26 September 2016 S/D 25 Maret 2017

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua

Joseph Dedy Irawan, ST., MT.
NIP : 197404162005011002

Form S-4a

3. Lember Persetujuan Dosen Pembimbing II



PT BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

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

Kampus I : Jl. Bendungan Sigura-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

Malang, 26 September 2016

Nomor : ITN-806/IX.INF/TA/2016
Lampiran : ---
Perihal : *Bimbingan Skripsi*
Kepada : Yth. Bpk/Ibu Rofila Ei Magfiroh, S.Si, M.Sc
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : MUSYAFFA AL HAFFI
Nim : 1318109
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

26 September 2016 S/D 25 Maret 2017

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,


Joseph Dedy Irawan, ST., MT.
NIP : 197404162005011002

Form S-4a

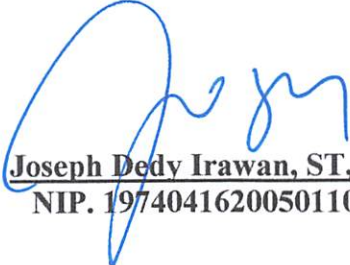


**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Musyaffa Al - Hafi
NIM : 13.18.109
JURUSAN : Teknik Informatika S-1
JUDUL : *GAME ENDLES RUNING MENGGUNAKAN METODE
FSM(FINITE STATE MACHINE)*


Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Rabu
Tanggal : 18 Januari 2017
Nilai : 77

Panitia Ujian Skripsi :
Ketua Majelis Penguji

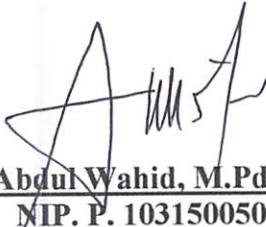

Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I


Suryo Adi Wibowo, ST, MT
NIP. 1031500438

Dosen Penguji II


Abdul Wahid, M.Pd.I
NIP. P. 1031500507



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Musyaffa Al - Hafi
NIM : 13.18.109
JURUSAN : Teknik Informatika S-1
JUDUL : *GAME ENDLES RUNING MENGGUNAKAN METODE
FSM(FINITE STATE MACHINE)*

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	18 Januari 2017	1. Pengujian FSM 2. Pengujian AI 3. Jebakan di masing - masing stage	
2.	Penguji II	18 Januari 2017	1. Game statis 2. Game melatih focus dan strategi	

Dosen Penguji I

Suryo Adi Wibowo, ST, MT
NIP. 1031500438

Dosen Penguji II

Abdul Wahid, M.Pd.I
NIP. P. 1031500507

Dosen Pembimbing I

Karina Auliasari, ST.M.Eng
NIP. 1031000426

Dosen Pembimbing II

Rofila El Magfiroh, S.Si, M.Sc
NIP.P 1031000425



FORMULIR BIMBINGAN SKRIPSI

Nama : Musyaffa Al – Hafi
Nim : 1318109
Masa Bimbingan : 26 September – 25 maret 2017
Judul Skripsi : Game Endles Runing Si Kancil Menggunakan Metode
FSM (Finite State Machine)

No	Tanggal	URAIAN	PARAF PEMBIMBING
1	19 - 10 - 2016	Komponen – komponen game	Li
2	26 - 10 - 2016	Pembahasan stage 1	Li
3	2 - 11 - 2016	Pembahasan animasi player	Li
4	9 - 11 - 2016	Pembahasan misi pada stage	Li
5	7 - 12 - 2016	Fix bab IV hasil	Li
6	8 - 11 - 2016	Fix bab IV Hasil dan Pengujian	Li
7	10 - 01 - 2017	Pembahasan seminar hasil	Li
8	11 - 01 - 2017	Fix bab V	Li
9	12 - 01 - 2017	Persetujuan maju ujian kompre	Li
10			

Malang, 19 - Oktober – 2016

Dosen Pembimbing

Karina Auliasari, ST.M.Eng
NIP. 1031000426



FORMULIR BIMBINGAN SKRIPSI

Nama : Musyaffa Al – Hafi
Nim : 1318109
Masa Bimbingan : 26 September – 25 maret 2017
Judul Skripsi : Game Endles Runing Si Kancil Menggunakan Metode
FSM (Finite State Machine)

No	Tanggal	URAIAN	PARAF PEMBIMBING
1	20 - 10 - 2016	Revisi laporan	 25/17
2	27 - 10 - 2016	Pembahasan AI	
3	1 - 11 - 2016	Revisi laporan	
4	9 - 11 - 2016	Pembahasan game	
5	7 - 12 - 2016	Pembahasan seminar progres	
6	8 - 11 - 2016	Bab IV Hasil dan Pengujian	
7	12 - 12 - 2017	Fix bab IV dan Hasil	
8	11 - 01 - 2017	Pembahasan seminar hasil	
9	12 - 01 - 2017	Revisi laporan	
10	12 - 01 - 2017	Laporan fix	

Malang, 19 - Oktober – 2016

Dosen Pembimbing



Rofila El Magfiroh, S.Si, M.Sc
NIP.P 1031000425

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : *Meh. jani*
Status : *Mahasiswa*

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game	✓		
3	Kontrol pada game		✓	
4	Implementasi FSM pada musuh		✓	
5	Fitur game		✓	
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh	✓		

Saran

Desain game over

Lembar Kuisisioner

Game Endles Runing Si Kancil

Nama : IWAN LAPARA

Status : MTK

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game		✓	
3	Kontrol pada game		✓	
4	Implementasi FSM pada musuh		✓	
5	Fitur game		✓	
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh	✓		

Saran

PERBANYAK IBADAH

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Moh. Fadil Rahman

Status : Mahasiswa 2

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game	✓		
3	Kontrol pada game	✓		
4	Implementasi FSM pada musuh		✓	
5	Fitur game		✓	
6	Desain Main Menu		✓	
7	Desain Game over		✓	
8	Desain stage		✓	
9	Desain Musuh	✓		

Saran

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Oktopian

Status : Mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game	✓		
3	Kontrol pada game	✓		
4	Implementasi FSM pada musuh		✓	
5	Fitur game		✓	
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh		✓	

Saran

Menambahkan karakter player. User agar dapat memilih

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Wahyu

Status : Mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game		✓	
2	Desain Animasi pada Game		✓	
3	Kontrol pada game		✓	
4	Implementasi FSM pada musuh		✓	
5	Fitur game		✓	
6	Desain Main Menu		✓	
7	Desain Game over		✓	
8	Desain stage		✓	
9	Desain Musuh	✓	✓	

Saran

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Jaya

Status : Mhs

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game	✓		
3	Kontrol pada game	✓		
4	Implementasi FSM pada musuh		✓	
5	Fitur game			✓
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh	✓		

Saran

tambah fitur game

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Farah

Status : Mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game		✓	
2	Desain Animasi pada Game		✓	
3	Kontrol pada game		✓	
4	Implementasi FSM pada musuh	✓		
5	Fitur game		✓	
6	Desain Main Menu		✓	
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh		✓	

Saran

perjelas alur permainan

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : wahyudi

Status : mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game		✓	
2	Desain Animasi pada Game		✓	
3	Kontrol pada game	✓		
4	Implementasi FSM pada musuh	✓		
5	Fitur game			✓
6	Desain Main Menu		✓	
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh			✓

Saran

- tambahkan fitur
- tingkatkan desain rintangan.

Lembar Kuisisioner
Game Endles Runing Si Kancil

Nama : Hasan

Status : Mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game		✓	
3	Kontrol pada game		✓	
4	Implementasi FSM pada musuh			✓
5	Fitur game		✓	
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage		✓	
9	Desain Musuh		✓	

Saran

Tingkat kesulitan musuh perlu ditingkatkan.

Lembar Kuisioner
Game Endles Runing Si Kancil

Nama : Merry sion

Status : Mahasiswa

NO	PERTANYAAN	PENILAIAN		
		BAIK	CUKUP	KURANG
1	Desain Karakter Game	✓		
2	Desain Animasi pada Game	✓		
3	Kontrol pada game	✓		
4	Implementasi FSM pada musuh		✓	
5	Fitur game			✓
6	Desain Main Menu	✓		
7	Desain Game over	✓		
8	Desain stage			✓
9	Desain Musuh	✓		

Saran

Meningkatkan stage