PEMBUATAN GAME COLOR INVASION MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)

SKRIPSI

Disusun oleh :

EDWIN YUNANTO

12.18.113

PROGRAM STUDI TEKNIK INFORMATIKA S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2016

LEMBAR PERSETUJUAN

PEMBUATAN *GAME* COLOR INVASION MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)

SKRIPSI

Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai Gelar Sarjana Komputer Strata Satu (S-1)

Disusun Oleh :
EDWIN YUNANTO
12.18.113

Diperiksa dan Disetujui,

DosenPembimbing I	DosenPembimbing II
Joseph Dedy Irawan, ST, MT	**Sonny Prasetio, ST, MT**
NIP.1974041620050110002	**NIP.P. 1031000433**

Mengetahui
Ketua Program Studi Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP.1974041620050110002

PROGRAM STUDI TEKNIK INFORMATIKA S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2016

LEMBAR KEASLIAN

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Edwin Yunanto

NIM : 12.18.113

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa skripsi saya yang berjudul :**"PEMBUATAN *GAME COLOR INVASION* MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)"**Adalah skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 13 Januari 2016

Yang membuat pernyataan

Edwin Yunanto

PEMBUATAN *GAME COLOR INVASION* MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)

Edwin Yunanto

12.18.113

Jurusan Teknik Informatika S-1

Fakultas Teknologi Industri

Institut Teknologi Nasional Malang

Jln. Raya Karanglo Km 2 Malang

E-mail : Yunantoedwin@gmail.com

Dosen Pembimbing : 1. Joseph Dedy Irawan, ST.MT.

2. Sonny Prasetio, ST.MT.

Abstrak

Permainan atau sering disebut dengan game merupakan suatu sarana hiburan yang diminati dan dimainkan oleh banyak orang baik dari kalangan anak-anak, remaja maupun orang dewasa. Game ini terdiri dari game tradisional dan game moderen. Game tradisional merupakan segala bentuk permainan yang telah ada sejak zaman dahulu dan diwariskan secara turun-menurun dari generasi ke generasi. Biasanya game tradisional yang tumbuh dan berkembang dalam suatu masyarakat mencerminkan warna kebudayaan setempat. Sedangkan game modern merupakan game yang disajikan pada suatu piranti atau perangkat teknologi dan dimainkan secara virtual. Media teknologi yang digunakan, yaitu console, komputer, handphone dan elektronik lainnya. Seiring perkembangan teknologi, game tradisional ini sudah jarang dimainkan dan kalah bersaing dari game modern.

Game adventure ini di bangun menggunakan aplikasi Adobe Flash dan bahasa pemograman Action Script 2.0, sehingga proses pembuatan karakter, level, dan metode FSM dapat dicapai. Dalam game ini terdapat 4 level dan beberapa musuh yang memiliki sifat berbeda. Prilaku yang dilakukan oleh musuh yaitu melihat player, mengejar pemain dan menyerang pemain. Pada level terakhir game juga terdapat musuh boss, untuk mengalahkan boss , pemain secara tidak langsung dituntut untuk mengasah kemampuan analisa dan strateginya, karena musuh boss lebih sulit dikalahkan daripada musuh-musuh kecil pada level sebelumnya

Dalam proses implementasinya, gameini berhasil berjalan pada 5 komputer dengan spesifikasi yang berbeda. Dari hasil pengujian dapat disimpulkan Bahwa game ini dapat berjalan pada komputer dengan ram 512mb hingga 4gb dan dengan sistem operasidari windows xp hingga windows 10

Kata kunci : *Game Adventure, Action Script 2.0, Finite State Machine, Adobe Flash*

KATA PENGANTAR

Dengan memanjatkan puji syukur kami panjatkan kehadiran Dzat yang Maha Agung Allah SWT yang selalu memberikan Rahmat dan HidayahNya yang telah dilimpahkan, sehingga penulis dapat menyelesaikan Skripsi dengan judul **"Pembuatan *Game Color Invasion* Menggunakan Metode *Finite State Machine*"**dengan lancar tanpa menemukan hambatan yang berarti. Skripsi ini merupakan persyaratan kelulusan di program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang dan untuk mencapai gelar Sarjana Komputer.

Keberhasilan penyelesaian laporan Skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Oleh karena itu pada kesempatan ini dengan segala kerendahan hati penyusun mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan kesehatan bagi penyusun sehinggan dapat mengerjakan laporan Skripsi.
2. Ayah dan ibunda tercinta, karena selalu berdo'a yang terbaik dan selalu memberikan dorongan baik secara moral maupun materiil untuk menyelesaikan Skripsi ini.
3. Dr. Ir. Lalu Mulyadi, MTA, selaku Rektor Institut Teknologi Nasional Malang.
4. Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Indrustri Institut Teknologi Nasional Malang.
5. Joseph Dedy Irawan, ST, MT., selaku Ketua Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
6. Sonny Prasetio, ST, MT, selaku Sekertaris Program Studi Teknik Informatika S-1Institut Teknologi Nasional Malang.
7. Joseph Dedy Irawan, ST, MT, selaku Dosen Pembimbing I Skripsi Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
8. Sonny Prasetio,ST,MT, selaku Dosen Pembimbing II Skripsi Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.

9. Sahabat sahabati tercinta yang telah memberikan banyak gagasan ide dan masukan.
10. Sanak famili serta keluarga yang selalu memberikan do'a restu, dorongan dan semangat.
11. Teman – teman dan semua yang tak mungkin disebutkan satu per satu yang telah membantu dalam penyelesaian penyusunan Skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan Skripsi ini. Untuk itu penyusun mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan laporan Skripsi ini.

Akhir kata penyusun mohon maaf yang sebesar-besarnya bilamana dalam penyusunan laporan Skripsi ini terdapat kekurangan serta kesalahan dalam penulisan. Semoga laporan Skripsi ini bermanfaat bagi pembaca.

Malang, 25 Januari 2016

(Edwin Yunanto)

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB III

ANALISIS DAN PERANCANGAN SITEM

3.1. Analisis Sistem

Merupakan suatu kegiatan yang menguraikan seluruh pokok masalah yang ada didalamnya. Analisa merupakan tahapan awal sebelum masuk ke tahapan perancangan, sedangkan perancangan merupakan hasil dari keseluruhan analisa yang dapat memberikan solusi dalam suatu permasalahan.

3.1.1. Analisis Masalah

Analisis masalah merupakan proses idenifikasi serta evaluasi terhadap *game* sejenis dan *Game* yang akan dibangun.Dalam *Game* Adventure, pemain diharuskan untuk menyelesaikan semua level yang ada dalam *Game*. Setiap level memiliki tingkat kesulitan tertentu dan memiliki syarat tertentu seperti menyelesaikan misi di setiap level untuk dapat menuju ke level selanjutnya. Disetiap *Game* Adventure, akan terasa kurang menarik jika kita hanya melawan musuh-musuh kecil yang mudah mati atau biasanya disebut creep. Oleh karena itu dibuatlah sebuah AI yang tidak mudah untuk dikalahkan player dan memiliki status diatas rata-rata bahkan melebihi player yang biasanya disebut boss. Boss berada di level akhir *game* dan untuk mengalahkannya pemain tidak bisa ceroboh dalam menyerang. Untuk mengalahkan boss , pemain secara tidak langsung dituntut untuk mengasah kemampuan analisa dan strateginya.

Game dengan genre Adventure dengan sideview atau sidecroller yaitu *Game* yang terlihat dari samping sudah tidak asing lagi dalam dunia *Game*, khususnya Indonesia. Dalam perkembangannya genre *Game* seperti ini telah berkembang pesat karena jenis *Game* seperti ini terbilang mudah dan menyenangkan untuk dimainkan oleh masyarakat terutama anak-anak. Beberapa *Game* yang memakai genre ini antara lain Super Mario, Captain Commando,Super Contra, dan Metal Slug.

3.1.2. Pengenalan *Game Color Invasion*

Game yang akan dibangun adalah *game Color Invasion* dengan genre Adventure Sidescroller *game*. *Game* ini dibangun dengan mengaplikasikan teknologi sebagai sarana untuk meningkaatkan kemampuan koordinasi antara mata dan tangan serta analisa pemain. Fitur-fitur yang digunakan pada *game* ini adalah:

1. Menggunakan Grafik 2D.
2. Metode yang digunakan adalah *Finite State Machine*
3. *Game* ini bergenre *Adventure*
4. *FSM* terletak pada musuh, jika terdeteksi pada area musuh maka secara otomatis musuh akan curiga dan mulai mengejar lalu membunuh karakter utama.

3.1.3 *Game Story*

Game Color Invasion ini menceritakan tentang sebuah dunia berwarna hitam bernama *Dark World* yang diserang oleh dunia lain bernama *Vivid Wolrd* yang memiliki banyak warna. *Vivid World* berencana menjadikan *Dark World* Menjadi dunia penuh warna dan menghilangkan semua warna hitam di dunia *Dark World*, semua penghuni *Dark World* yang berwarna hitam akan diubah menjadi pasukan *Vivid World* dengan warna yang lain. Tetapi ada satu penghuni *Dark World* bernama Grey yang memiliki kekuatan yang bisa berganti warna menjadi warna lain, sehingga grey dapat melawan setiap pasukan *Vivid World*. Kekutan ini diperoleh Grey ketika dia akan diubah oleh pasukan *Vivid World* menjadi warna lain, akan tetapi DNA milik Grey bisa menyerap seluruh radiasi warna yang diterimanya dan dia tidak bisa terpengaruh oleh virus warna yang bisa merubahnya menjadi pasukan *Vivid World*. Grey berencana ingin mengembalikan *Dark World* Menjadi seperti semula, dengan mengumpulkan sebuah cairan bernama *Black Ink*, Grey bisa merubah *Dark world* menjadi berwarna hitam lagi.

3.1.4 *Game Rules*

Game rule merupakan aturan perintah, cara menjalankan, fungsi objek dan karakterdi dunia dalam *game*. Oleh karena itu, dalam memainkan suatu *game*, pemain harus patuh dan bermain sesuai aturan yang berlaku. Pada *game Color Invasion* ini pemain mengendalikan sebuah karakter yang dapat merubah warnanya sesuai keinginan pemain, tidak hanya melompat, menghindari rintangan dan membunuh musuh seperti *game adventure* pada umumnya, pada *game* ini pemain juga mencocokkan warna karakter dengan warna daratan dan rintangan yang dilewati karakter, jika warna karakter tidak sesuai dengan warna daratan maka Health Bar atau nyawa karakter akan berkurang, begitu juga dengan musuh karakter yang hanya bisa dibunuh jika warna karakter sama dengan warna musuh.Dalam *gameColor Invasion* ini terdapat 4 level yang harus di lewati agar dapat memenangkan *game* ini. Pada setiap level pemain akan menghadapi rintangan dan musuh untuk menuju gerbang ke level selanjutnya, setiap level memiliki perbedaan jumlah musuh dan jenis musuh yang mempunyai sifat berbeda. Pemain juga harus mengumpulkan cairan *ink*sampai indicator *ink bar* penuh untuk bisa membuka gerbang menuju level berikutnya, jika cairan *ink* belum penuh maka pemain tidak bisa membuka gerbang ke level berikutnya. Pada level 1 dan 2 pemain hanya bisa merubah warna menjadi 4 warna yaitu warna hitam, merah, kuning, dan biru, ketika sudah berada di level 3 dan 4 pemain bisa merubah warna tambahan lagi yaitu hijau dan oranye, di level 4 pemain juga akan menghadapi musuh *boss*.

3.1.5 *Game Goal*

Di dalam sebuah *game* pasti memiliki *goal* atau tujuan untuk bisa menyelesaikan setiap level pada *game*. Dalam *game Color Invasion* ini memiliki *goal* yang berbeda pada setiap levelnya. Goal yang harus diselesaikan pemain pada setiap level *game Color Invasion* ini adalah:

1. Level 1

Goal pada level 1 *game Color Ivasion* ini adalah, pemain harus mengumpulkan cairan *ink*sampai indikator *ink* penuh untuk bisa membuka gerbang ke level selanjutnya.

2. Level 2

Goal pada level 2 *game Color Ivasion* ini adalah, pemain harus menghancurkan tumpukan batu yang menghalangi jalan menuju gerbang ke level selanjutnya. Untuk menghancurkan tumpukan batu tersebut, pemain harus mencari beberapa bom di dalam dunia level 2 ini. Seperti pada level sebelumnya pemain juga harus mengumpulkan cairan *ink*sampai indikator *ink* penuh untuk bisa membuka gerbang ke level selanjutnya.

3. Level 3

Goal pada level 3 *game Color Ivasion* ini adalah, pemain harus membebaskan penghuni *Dark World*yang di kurung dalam penjara oleh musuh. Untuk membebaskannya pemain harus mencari kunci yang dibawa oleh musuh, jadi untuk mendapatkan kunci, pemain harus membunuh musuh yang yang memiliki kunci penjara tersebut . Seperti pada level sebelumnya pemain juga harus mengumpulkan cairan *ink*sampai indikator *ink* penuh untuk bisa membuka gerbang ke level selanjutnya.

4. Level 4

Goal pada level 3 *game Color Ivasion* ini adalah, pemain harus mengalahkan *boss* untuk mengaktifkan semua tabung cairan tinta hitam yang bisa merubah planet *Dark World* menjadi warna hitam lagi. Untuk mengalahkan *boss* pemain harus menghancurkan tabung warna agar warna boss berubah sesuai tabung warna yang dihancurkan pemain.

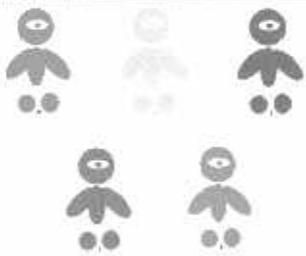
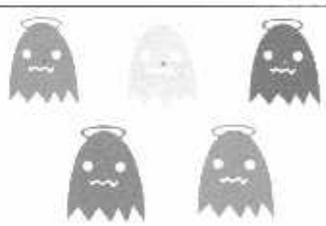
3.2. Perancangan Sistem

Perancangan sistem adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci tentang *Game Color Invasion.*

3.2.1 Perancangan karakter

Perancangan karakter merupakan pembahasan mengenai karakter yang terlibat dalam *Game Color Invasion.*Karakter pada *Game Color Invasion* dapat dilihat pada Tabel 3.1.

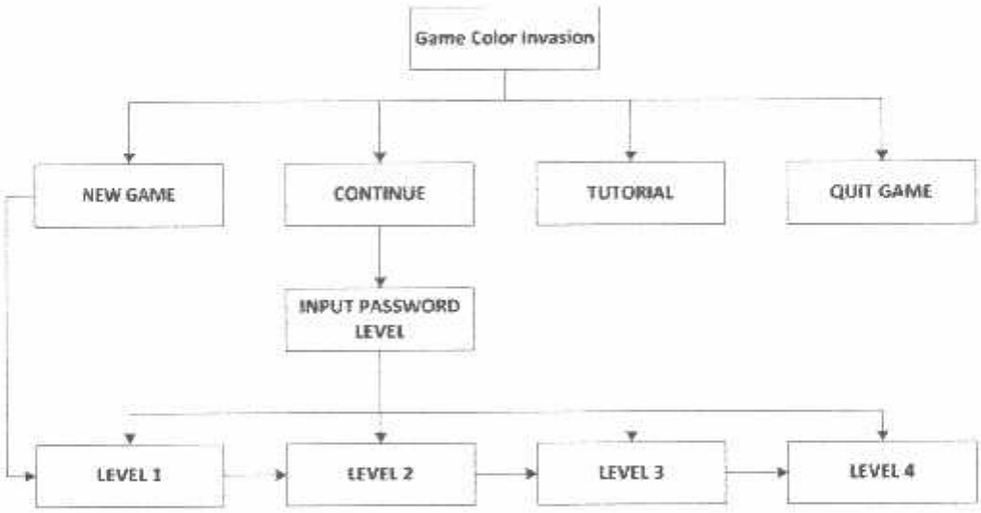
Tabel 3.1. PerancanganKarakter

No	Gambar	Keterangan
1		Karakter Utama
2		Musuh Walker
3		Musuh Ghost

4		Musuh Mecha
5		Musuh Boss

3.2.2 Perancangan Struktur Menu

Perancangan Struktur menu adalah perancangan tata urutan menu dari *Game Color Invasion* seperti yang terlihat pada Gambar 3.1.



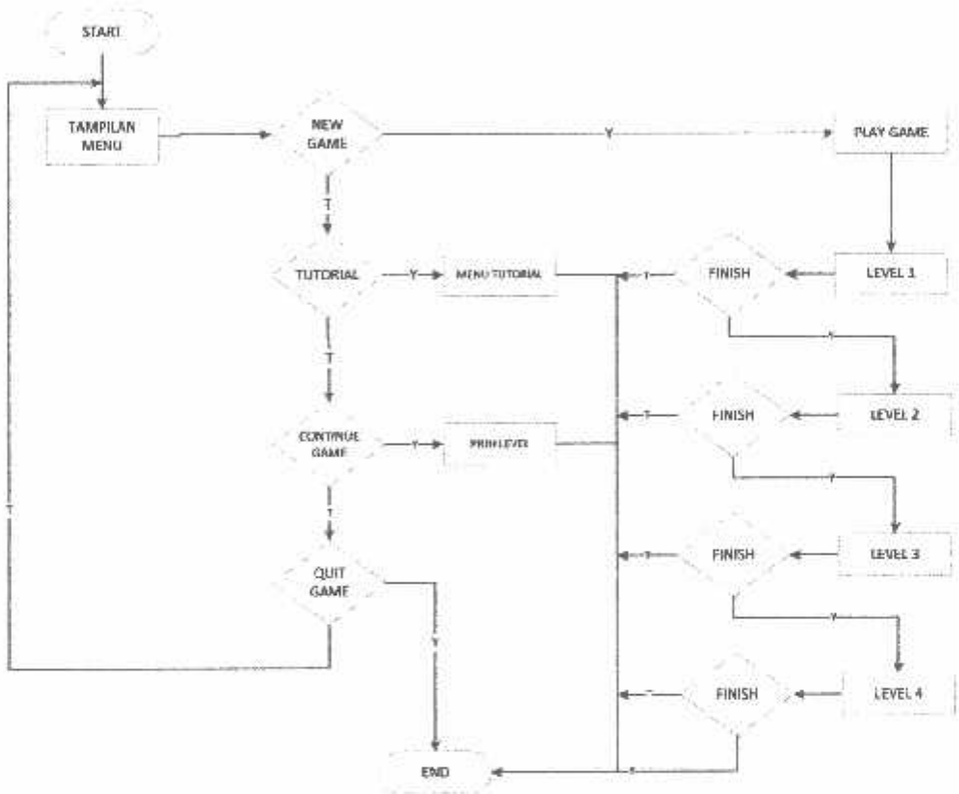
Gambar 3.1. Perancangan Struktur Menu

Pada *game Color Invasion* terdapat 4 menu utama yaitu *New Game, Continue,Tutorial,* dan *QuitGame.* Sebelum memulai *game* baru, diharuskan ke menu *Tutorial* agar mengetahui cara bermain dan tombol – tombol yang ada pada *game Color Invasion*, sehingga pemain dapat lebih mudah dalam bermain. Dan pada menu *Continue*pemain bisa memainkan level yang ingin dimainkan dengan memasukkan password level yang

diinginkan, pemain bisa mendapatkan password level setelah memenangkan level tersebut.

3.2.3 Perancangan Alur *Game*

Perancangan alur *game* berfungsi untuk mengetahui alur proses dari alur program dimulai dari *start* prorgram hingga *end* program seperti yang ditunjukkan pada Gambar 3.2.

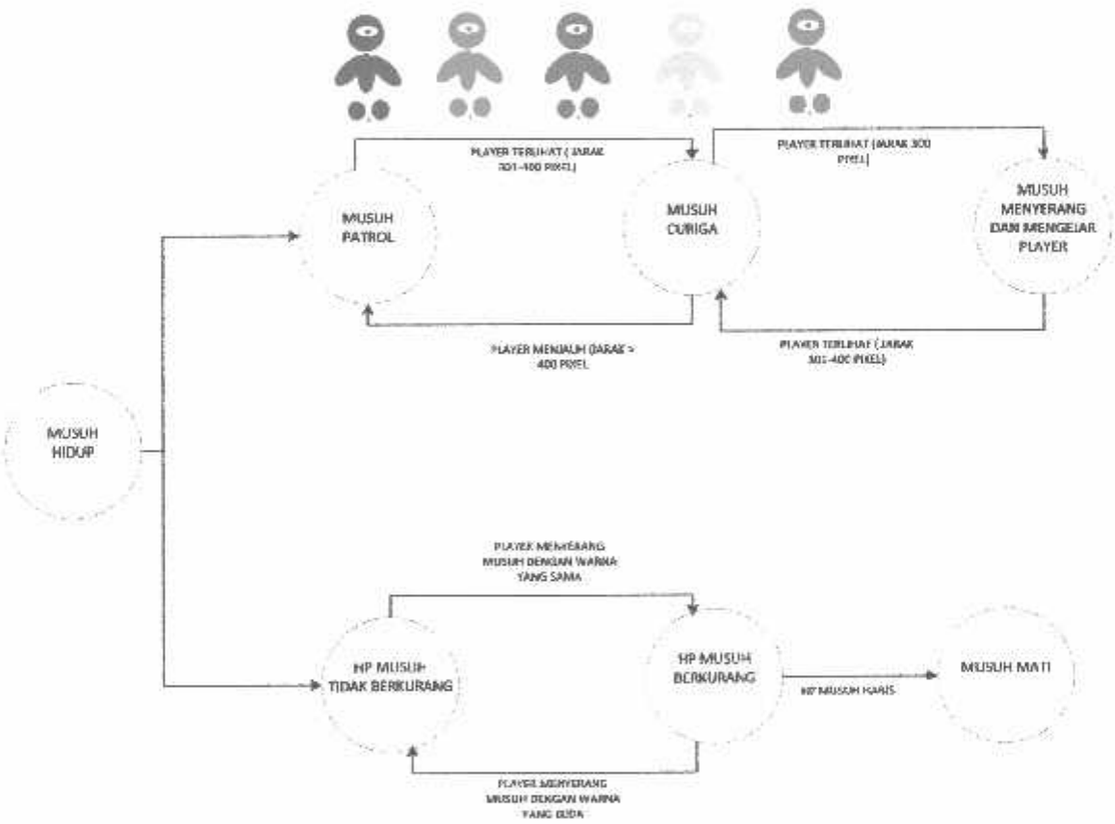


Gambar 3.2. Alur *Game Color Invasion*

Pada Gambar 3.2 , program dimulai dari start kemudian masuk ke 4 menu utama yaitu *New Game, Continue, Tutorial*, dan *QuitGame*. Jika pemain memulai *New Game* maka akan langsung menuju level 1, dan jika selesai akan melanjutkan ke level 2 sampai ke stage terakhir. Jika pemain kalah, akan di arahkan ke tampilan *Game over* dan menuju ke *New Game* atau *game* baru.

3.2.4 Alur *Finite State Machine*

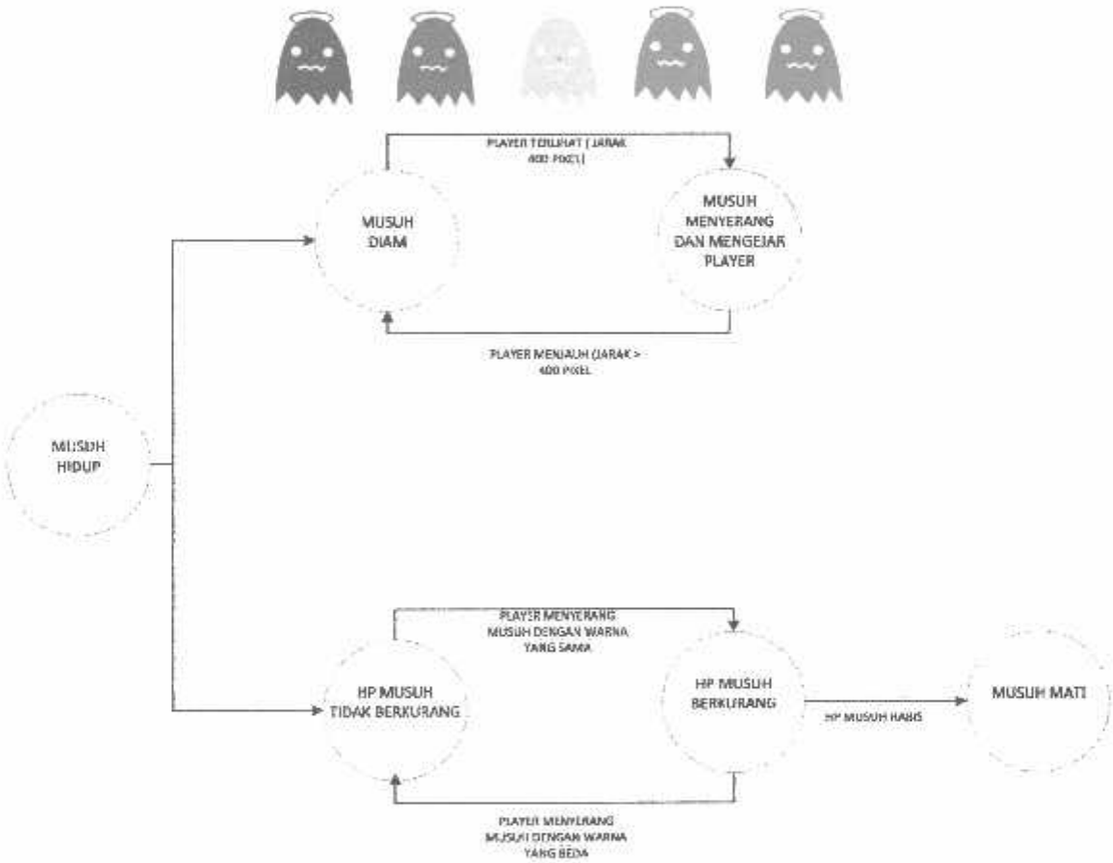
Finite State Machine merupakan salah satu logika penalaran yang memperlihatkan perilaku system dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada suatu saat, system akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh system ketika menanggapi masukan yang terjadi. Pada *game Color Invasion* ini metode *Finite State Machine* diterapkan pada karakter musuh, masing-masing karakter musuh memiliki alur Finite State Machine yang berbeda.



Gambar 3.3.Alur diagram *Finite State Machine* Pada Karakter Musuh *Walker*

Pada Gambar 3.3 memperlihatkan alur FSM pada karakter musuh walker. Sesuai dengan alur diagram diatas, karakter musuh walker memiliki

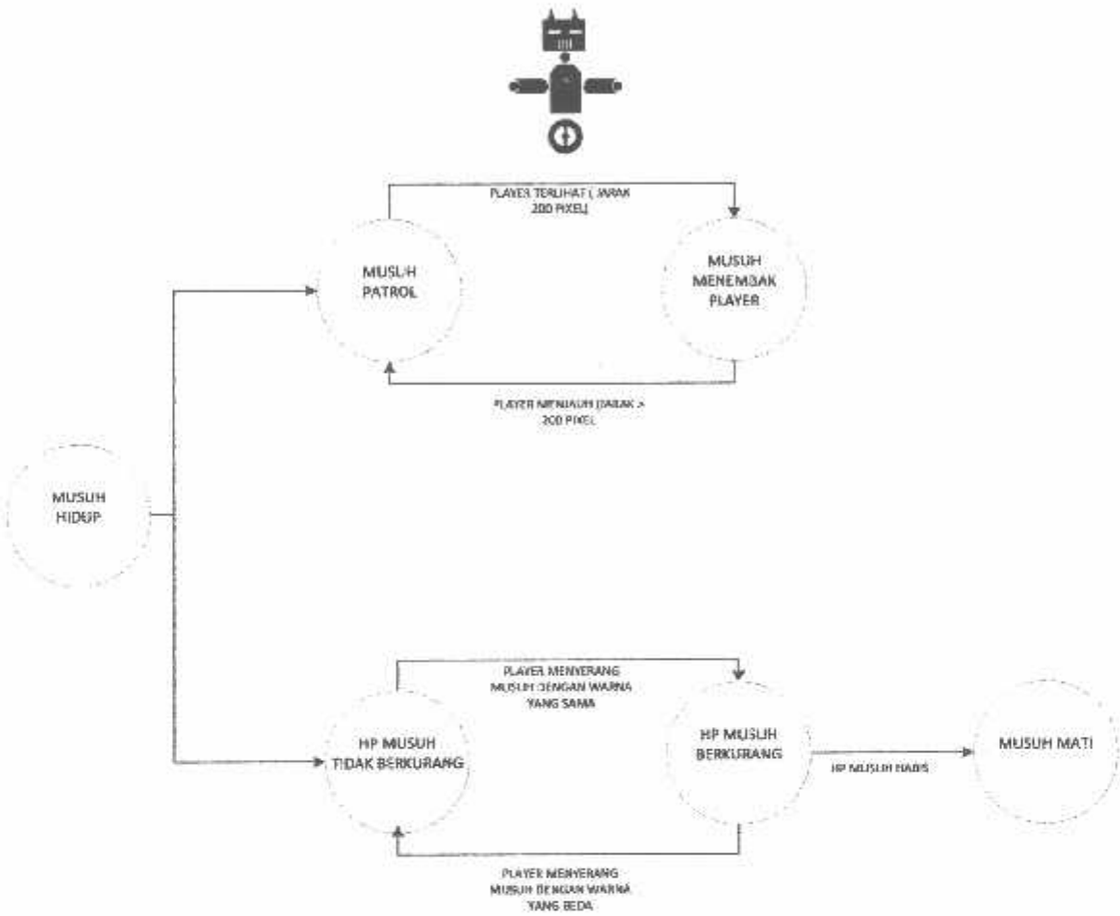
sifat jika karakter player mendekat pada jarak 301-400 pixel dari musuh, maka musuh akan mencurigai keberadaan karakter player, dan ketika player mendekat lagi pada jarak 300 pixel dari musuh, maka musuh akan mengejar dan menyerang player, jika player menjauh dari musuh dengan jarak lebih dari 400 pixel dari musuh, maka musuh akan kembali patrol. Pada *game* ini setiap musuh memiliki 6 warna yang berbeda yaitu hitam, merah, kuning, hijau, biru, dan oranye. Musuh hanya bisa dibunuh jika warna player sama dengan warna musuh, jika karakter player menyerang musuh dengan warna yang berbeda dengan warna musuh maka Health Poin musuh tidak akan berkurang.



Gambar 3.4. Alur diagram *Finite State Machine* Pada Karakter Musuh *Ghost*

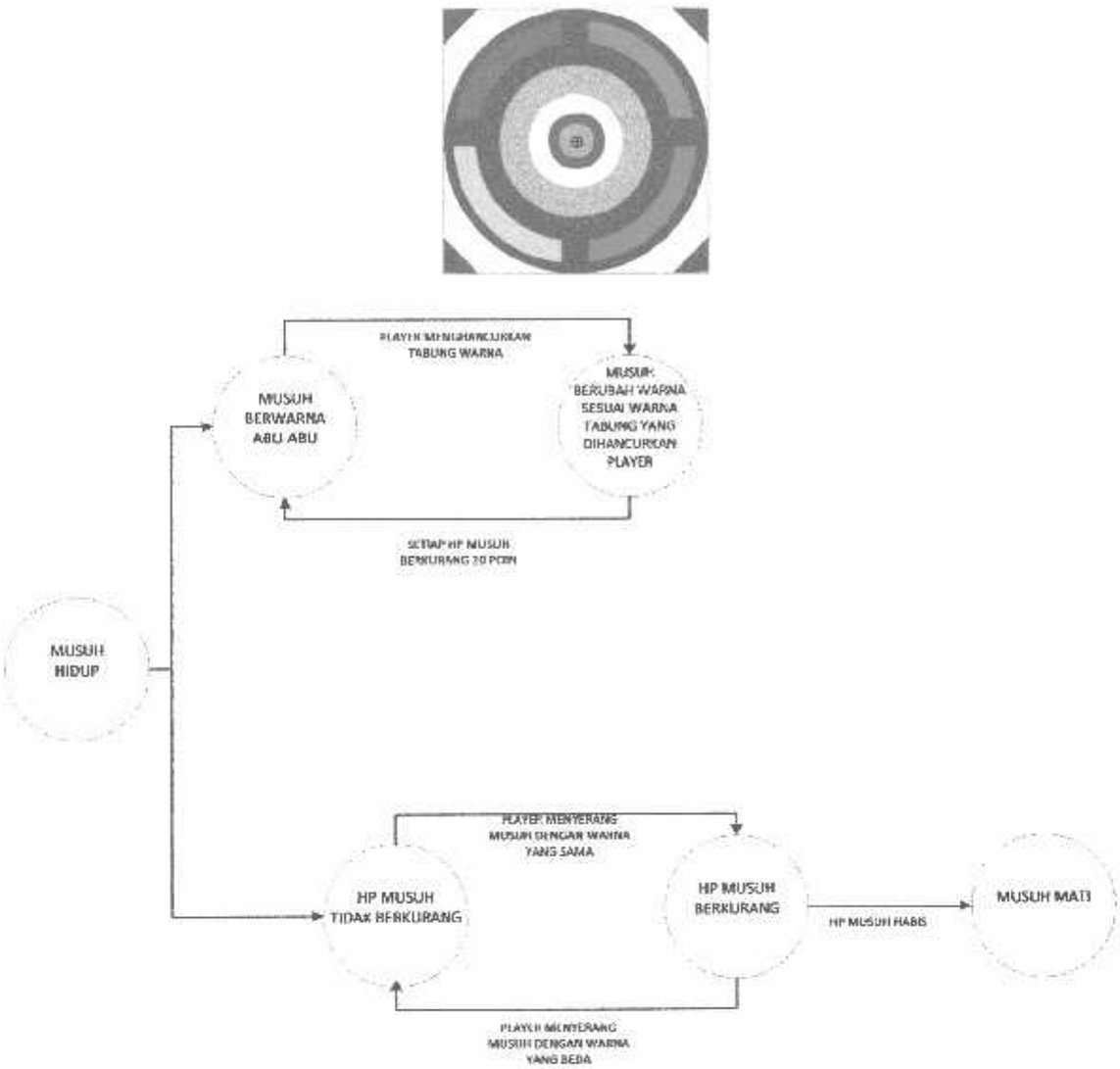
Pada Gambar 3.4 memperlihatkan alur FSM pada karakter musuh *ghost*. Sesuai dengan alur diagram diatas, karakter musuh *ghost* memiliki

sifat jika karakter player mendekat pada jarak 400 pixel dari musuh, maka musuh akan mengejar dan menyerang player, jika player menjauh dari musuh dengan jarak lebih dari 400 pixel dari musuh, maka musuh akan diam pada posisi di mana musuh berhenti mengejar karakter player.



Gambar 3.5. Alur diagram *Finite State Machine* Pada Karakter Musuh *Mecha*

Pada Gambar 3.5 memperlihatkan alur FSM pada karakter musuh *mecha*. Sesuai dengan alur diagram diatas, karakter musuh *mecha* memiliki sifat jika karakter player mendekat pada jarak 200 pixel dari musuh, maka musuh akan menembak player, jika player menjauh dari musuh dengan jarak lebih dari 200 pixel dari musuh, maka musuh akan kembali patrol.



Gambar 3.6. Alur diagram *Finite State Machine* Pada Karakter Musuh *Boss*

Pada Gambar 3.6 memperlihatkan alur FSM pada karakter musuh Boss. Sesuai dengan alur diagram diatas, karakter musuh boss memiliki sifat jika karakter player menghancurkan tabung warna tertentu, maka warna karakter boss akan berubah warna sesuai warna tabung yang dihancurkan player, karena karakter boss pertama kali muncul memiliki warna abu-abu sehingga player tidak bisa mengurangi Health Point karakter boss karena player tidak memiliki perubahan warna abu-abu maka player harus mencari tabung warna dan menghancurkannya agar bisa mengubah warna karakter

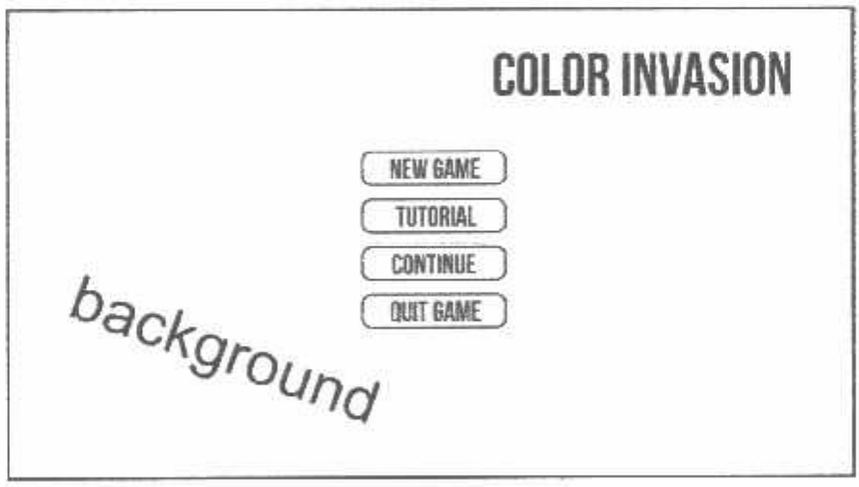
boss. Setiap Health Point karakter boss berkurang sebanyak 20 poin karakter boss akan merubah warnanya lagi menjadi warna abu-abu.

3.2.5 Perancangan Atarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran bagaimana tampilan keseluruhan *Game Color Invasion.*

1. Perancangan Tampilan Menu Utama

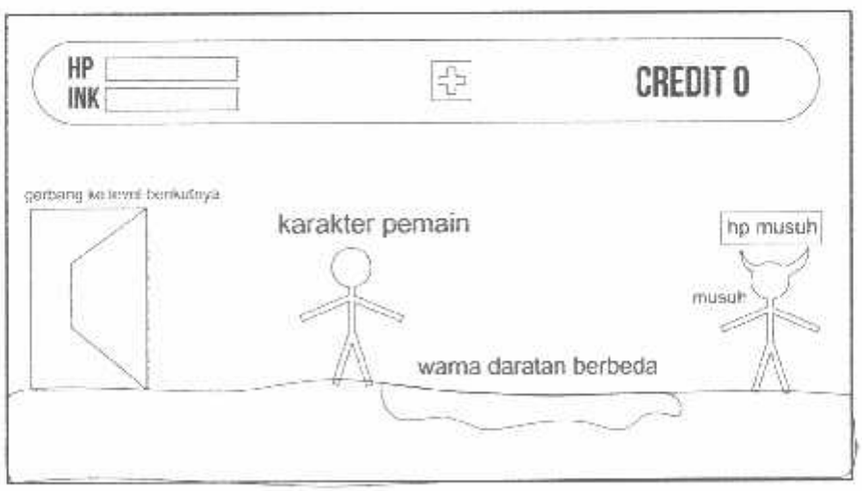
Perancangan tampilan menu utama dari *Game Color Invasion* dimana terdapat tombol antara lain *New Game, Tutorial, Continue, Quit Game*. Jika pemain memilih tombol *New Game*maka pemain akan diarahkan menuju permainan level 1. Jika pemain memilih tombol *Tutorial*, maka pemain akan diarahkan menuju level *game* yang menunjukkan bagaimana cara memainkan *game Color Invasion*. Jika Pemain memilih tombol *Continue* maka pemain akan diarahkan menuju menu untuk memilih level yang diinginkan dengan memasukkan password level. Jika pemain memilih *Quit Game*maka pemain akan keluar dari *game*. Rancangan menu utama dari *Game*Gunslinger ditunjukkan pada Gambar 3.7.



Gambar 3.7. Rancangan Tampilan Menu Utama

2. Perancangan antarmuka *level game*

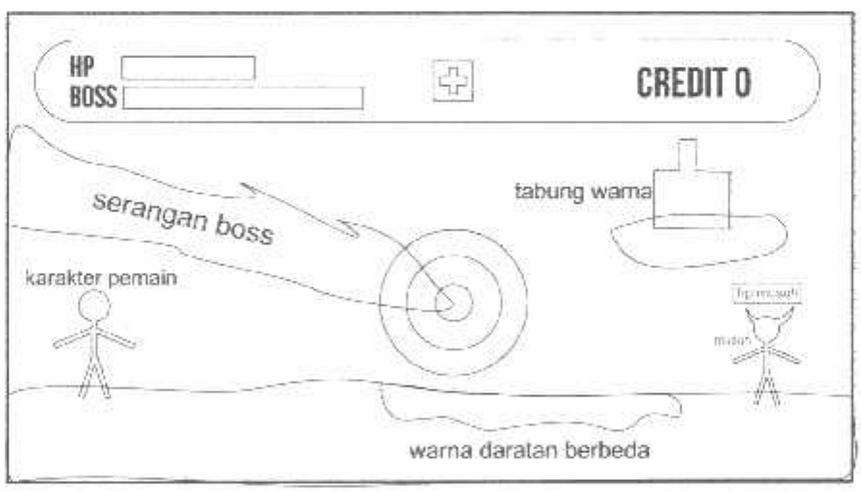
Perancangan antarmuka *level game* merupakan perancangan tampilan level dari *Game Color Invasion.* PadaSetiap level terdapat user interface dari karakter pemain yaitu *health point bar,ink bar, buy health,* dan *credit. Health point bar* bertujuan mengetahui darah dari pemain, jika darah pemain mencapai 0 maka pemain akan diarahkan menuju tampilan *Game over. Ink bar*berfungsi untuk menampilkan banyaknya cairan *ink* yang sudah didapat pemain, jika indikator *ink bar* belum penuh maka karakter tidak bisa membuka gerbang untuk ke level selanjutnya. Tombol symbol "+" atau buy health befungsi untuk membeli darah seharga 10 credit, jika credit pemain kurang dari 10 poin maka karakter tidak bisa membeli *health poin (HP). Credit* berfungsi untuk menampilkan banyaknya *credit*yang didapat pemain, jika pemain membunuh musuh dalam *game* maka pemain akan memperoleh *credit point.* credit digunakan untuk membeli *health poin.* Setiap daratan pada *game* juga memiliki warna yang berbeda, karakter pemain harus mencocokkan warna karakter dengan warna daratan agar *health point bar* pemain tidak berkurang. Musuh dalam *game* juga memiliki *health point bar* , jika *health point bar*musuh mencapai 0 maka musuh mati.Rancangan antarmuka *level game* dari *Game Color Invasion* ditunjukkan pada Gambar 3.8.



Gambar 3.8. Rancangan Antarmuka Level *Game*

3. Perancangan antarmuka *level boss game*

Perancangan antarmuka *level boss game* merupakan perancangan tampilan level terakhir dari *GameColor Invasion.* Pada level terakhir *game*, pemain akan melawan musuh *boss* yang lebih susah dan lebih kuat daripada musuh-musuh yang lain. Pada level boss ini *ink bar*akan digantikan dengan*Health point bar*dari boss, jika *Health point bar boss* mencapai 0 maka boss akan mati. Pada level *boss* ini terdapat tabung warna yang berfungsi untuk merubah warna dari musuh *boss*, karena warna *boss* pertama kali berwarna abu-abu pemain tidak bisa mengurangi *Health point bar* dari*boss*karena pemain tidak memiliki perubahan warna abu-abu, jadi agar pemain bisa mengurangi *Health point bar* dari*boss*pemain harus mecari dan menghancurkan tabung warna, warna dari *boss*akan berubah sesuai dengan warna dari tabung warna yang dihancurkan pemain. Serangan dari *boss*berupa laser besar yang memutar sepanjang area level *game*, pemain harus mencocokkan warna karakter dengan warna laser agar *Health point bar* pemain tidak berkurang.Rancangan antarmuka *level boss game* dari *Game Color Invasion* ditunjukkan pada Gambar 3.9.



Gambar 3.9. Rancangan Antarmuka Level Boss *Game*

BAB I
PENDAHULUAN

1.1. Latar Belakang

Permainan atau sering disebut dengan *game* merupakan suatu sarana hiburan yang diminati dan dimainkan oleh banyak orang baik dari kalangan anak-anak, remaja maupun orang dewasa. *Game* ini terdiri dari *game* tradisional dan *game* moderen. *Game* tradisional merupakan segala bentuk permainan yang telah ada sejak zaman dahulu dan diwariskan secara turun-menurun dari generasi ke generasi. Biasanya *game* tradisional yang tumbuh dan berkembang dalam suatu masyarakat mencerminkan warna kebudayaan setempat. Sedangkan *game* modern merupakan *game* yang disajikan pada suatu piranti atau perangkat teknologi dan dimainkan secara virtual. Media teknologi yang digunakan, yaitu console, komputer, handphone dan elektronik lainnya. Seiring perkembangan teknologi, *game* tradisional ini sudah jarang dimainkan dan kalah bersaing dari *game* modern.

Dalam setiap *game* tentunya tidak akan terasa menarik jika tidak ada lawan,baik pemain lain ataupun musuh-musuh yang sudah diprogram untuk melawanpemain yang biasanya disebut *Artficial Inteligence* (AI). Dalam *Game* Adventure,AI biasanya digunakan sebagai musuh pemain dan teman yang mengarahkanpemain untuk menyelesaikan alur cerita dalam *game*. Metode AI yang biasanyadigunakan dalam sebuah *game* adventure adalah metode *Finite State Machine*.

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.

Dari uraian diatas membuat penulis tertarik untuk membuat sebuah *game*dengan AI berjenis *Finite State Macine* untuk mengetahui bagaimana AI inibekerjapadasistem *game*.

1.2. Rumusan Masalah

Berdasarkan pada latar belakang masalah diatas, maka didapatkan rumusan masalah yang akan dibahas yaitu :

1. Bagaimana mengimplementasikan metode *Finite State Machine* sebagai kecerdasan buatan pada *game Color Invasion*?
2. Bagaimana membuat *Color Invasion* dengan Adobe Flash?

1.3. Batasan Masalah

Agar pembahasan dalam skripsi ini tidak meluas, maka ditentukan beberapa batasan masalah diantaranya, yaitu :

1. *Game* dibuat menggunakan Adobe Flash 5.
2. Grafik dalam *game* ini adalah 2D.
3. *Game* hanya dapat dimainkan pada PC atau berbasis *Desktop*.
4. *Game* ini dibuat untuk berjalan pada sistem operasi Windows.

1.4. Tujuan

Adapun tujuan dari pembuatan *game* ini yaitu :

1. Menerapkan metode *Finite State Machine* untuk menghasilkan suatu kecerdasan buatan pada *game Color Invasion*.
2. Untuk mengimplementasikan *game Color Invasion* ini dalam bentuk *game*komputer.

1.5. Manfaat

Adapun manfaat dari pembuatan *game* ini yaitu :

1. Mengasah kemampuan strategi dan kecepatan pemain dalam bermain *game*.
2. Sebagai refrensi untuk membangun sebuah *game* 2D, terutama untuk yang ber-genre 2d *Platform*.

1.6. Metode Penelitian

Untuk dapat mencapai keinginan dalam pembuatan *game* adventure 2D *Color Invasion*, dengan menggunakan metode Finite State Machine ini, maka perlu dilakukan dengan langkah-langkah sebagai berikut:

1. Studi Literatur

   Penelitian ini dimulai dengan studi literatur yaitu pengumpulan data yang berhubungan dengan permasalahan yang dibahas sehingga dapat membantu penyelesaian masalah dalam perancangan *game* dari sumber-sumber bacaan seperti, buku, jurnal, referensi, web page, dan karya tulis ilmiah.

2. Analisa kebutuhan

   Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan suatu kerangka yang digunakan untuk acuan perancangan perangkat lunak

3. Perancangan Sistem

   Data-data yang telah terkumpul diimplementasikan kedalam program bersama dengan pembuatan metode FSM.

4. Pengujian Program

   Pengujian coba ini bertujuan untuk memastikan bahwa masing-masing bagian dari sistem ini dapat bekerja sesuai yang diharapkan.

1.7. Sistematika Penulisan

Dalam penyusunan skripsi ini agar lebih mudah dipahami maka dibuatlah suatu sistematika dalam penulisan sebagai berikut :

BAB I : PENDAHULUAN

Pada Bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metode penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Pada Bab ini membahas tentang Landasan Teori menguraikan teori-teori yang mendukung judul, dan pembahasan secara detail. Landasan tori dapat berupa definisi-definisi atau model yang langsung berkaitan dengan ilmu atau masalah yang diteliti. Pada bab ini juga dituliskan

tentang *software*(komponen) yang digunakan dalam pembuatan program atau keperluan saat penelitian.

BAB III : ANALISA DAN PERANCANGAN SISTEM

Bab ini membahas tentang analisa pada sistem dan perancangan sistem yang dibuat.

BAB IV : IMPLEMENTASI DAN PENGUJIAN PROGRAM

Pada Bab ini membahas paparan implementasi dan analisa hasil pengujian program yang dibuat.

BAB V : PENUTUP

Pada Bab ini berisi kesimpulan dan saran yang didapat dari ulasan data-data penelitihan, menyimpulkan bukti-bukti yang diperoleh dari hasil analisa.

BAB II
LANDASAN TEORI

2.1. *Game*

Kata *game* berasal dari bahasa Inggris. Dalam kamus bahasa Indonesia istilah "*game*"adalah permainan. Permainan merupakan bagian dari bermain dan bermain juga bagian dari permainan keduanya saling berhubungan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*Intellectual Playability Game*) yang juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Dalam *game*, ada target - target yang ingin dicapai pemainnya. Permainan adalah kegiatan yang kompleks yang didalamnya terdapat peraturan, bermain dan budaya. Sebuah permainan adalahsebuah sistem dimana pemain terlibat dalam konflik buatan. Disini pemain berinteraksi dengan sistemdan konflik dalam permainan merupakan rekayasa atau buatan. Dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Mendefinisikan apakah yang dimaksud dengan *game*, tidak cukup dengan hanya melihat kamus bahasa. Terdapat banyak makna dalam kata "*game*□. Yang jelas *game* secara naluri adalah merupakan bagian dari kehidupan manusia. Makna sekilas dari *game* memberikan pengertian bahwa *Game* merupakan suatu aktifitas yang tidak dilakukan dengan sungguh-sungguh. Untuk mengetahui apa yang sesungguhnya disebut dengan *game*, maka paling tidak kita dapat memahaminya dari adanya sejumlah pengertian *game* yang biasa kita alami dalam kehidupan. (Zamroni dkk., 2013)

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. Berikut iniberbagai macam jenis *game*

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 kemampun 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, PES, 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 (Anneahira.com).

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 sebagai berikut.

1. Finite State Machines (FSM)

Finite State Machines (FSM) 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 (misal interupsi timer). 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 relative kompleks.Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan real time. Salah satu keutungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relative besar dengan hanya menggunakan sejumlah kecil item state. Selain untuk bidang kontrol, Penggunaan

metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak *game*, aplikasi WEB dan sebagainya.Dalam bahasa pemrograman prosedural seperti bahasa C, FSM ini umumnya direalisasikan dengan menggunakan statemen kontrol switch case atau/dan if..then. Dengan menggunakan statemen-statemen kontrol ini, aliran program secara praktis akan mudah dipahami dan dilacak jika terjadi kesalahan logika (Setiawan, 2006).

2. Algoritma A *Star*(A*)

Algoritma A *Star*(A*) adalah algoritma pencarian terbaik dalam mencari jalur terpendek dengan perhitungan terkecil pada jalur dengan simpul awal menuju simpul akhir. Algoritma ini pertama kali dideskripsikan pada tahun 1968 oleh Peter Hart, Nils Nilsson, dan Bertram Raphael. Dalam makalah mereka, disebut dengan Algoritma A. Lalu dengan optimasi heuristik, disebut dengan A *Star* (A*).

Perhitungan pada Algoritma A *Star* (A*) dapat ditentukan sebagai berikut :

$$F(x) = G(x) + H(x)$$

Dimana :

a. G(x) adalah nilai pada pergerakan simpul awal menuju simpul berikutnya.

b. H(x) adalah perkiraan nilai pergerakan simpul awal menuju tujuan akhir simpul. Fungsi ini seringkali disebut dengan fungsi heuristik,dinamakan heuristik karena perhitungan tersebut berdasarkan perkiraan (guess).

c. F(x) adalah jumlah nilai dari fungsi G(x) danH(x). dengan nilai terkecil F(x) adalah jalur terpendek menuju tujuan akhir.

Terdapat ketentuan pada grafik agar algoritma A Star (A*) ini bila diterapkan akan selalu mendapatkan jalan yang terpendek. ketentuan tersebut yang harus dipenuhi pada grafik yaitu.

1) Setiap simpul (node) dalam grafik memiliki jumlah terbatas pada area pencariannya.

2) Pada pencarian terdapat jalan yang dilalui untuk mencapai tujuan.

3) Fungsi F(x) pada grafik bernilai rendah daripada fungsi F(x) pada pencarian sebelumnya (Sofwan dkk.).

3. Algoritma *Minmax*

Algoritma *Minimax* (juga sering disebut *Minmax*)adalah sebuah algoritma yang mendasari pola pikir langkah penyelesaian masalah dalam beberapa jenis permainan komputer, seperti *tic-tac-toe*, *othello,checkers*, catur, dll. Pada dasarnya, algoritma *Minimax* sangat andaluntuk menyelesaikan segala masalah dalam pencarian langkah untuk permainan komputer dengan jumlah kemungkinan penyelesaian yang kecil, seperti padapermainan *tic-tac-toe*. Tetapi, jika algoritma *Minimax* digunakan pada permainan dengan jumlah kemungkinan penyelesaian yang besar seperti pada permainan catur, algoritma *Minimax* ini memerlukan waktu yang sangat lama untuk membangun pohon penyelesaian. Oleh karena itu, beberapa metode optimasi telah dikembangkan untuk membatasi melonjaknya jumlah simpul dalam pembangunan pohon penyelesaian.Berbagai jenis metode telah ditemukan untukmengoptimasi algoritma *Minimax*, seperti *alpha-betasearch*, *NegaScout*, *Null Move Search*, MTD(f), dll.Dengan menggunakan metode optimasi inilah proseskomputasi penyelesaian masalah pada permaian catur dapat diminimalisasi (Ilman, 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, Hidayah, 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 cepat1 (Prasetyo, Hidayah, 2014).

6. 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, Oktoviana).

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 kenyataanya. 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 Color Invasion*. Pada tampilan awal ini berisikan tombol *New Game* untuk memulai *game* baru,tombol *Tutorial* untukbelajar cara bermain,tombol *Continue*untuk memainkan lagi level yang diinginkan, dan tombol *quit game* untuk keluar dari *game*.Adapun desain menu utama seperti pada Gambar 4.1.

Gambar 4.1. Tampilan menu utama *Game Color Invasion*

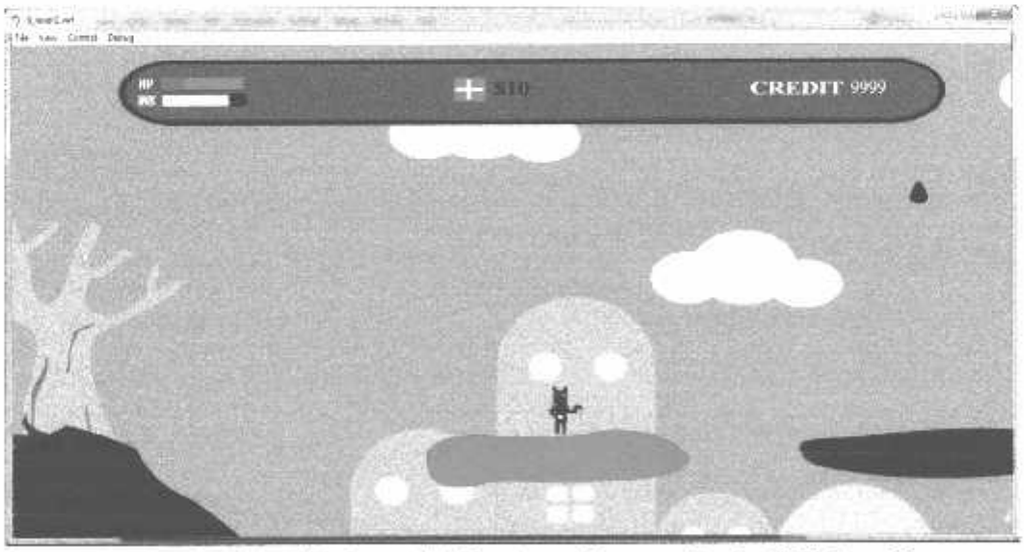
4.1.2 Tampilan Menu Tutorial

Tampilan menu *Tutorial* adalah tampilan menu untuk belajar dan mengetahui fungsi tombol yang digunakan pada *game Color Invasion.* Adapun tampilan dari menu *Tutorial* seperti padaGambar 4.2.

Gambar 4.2 Tampilan menu *Tutorial*

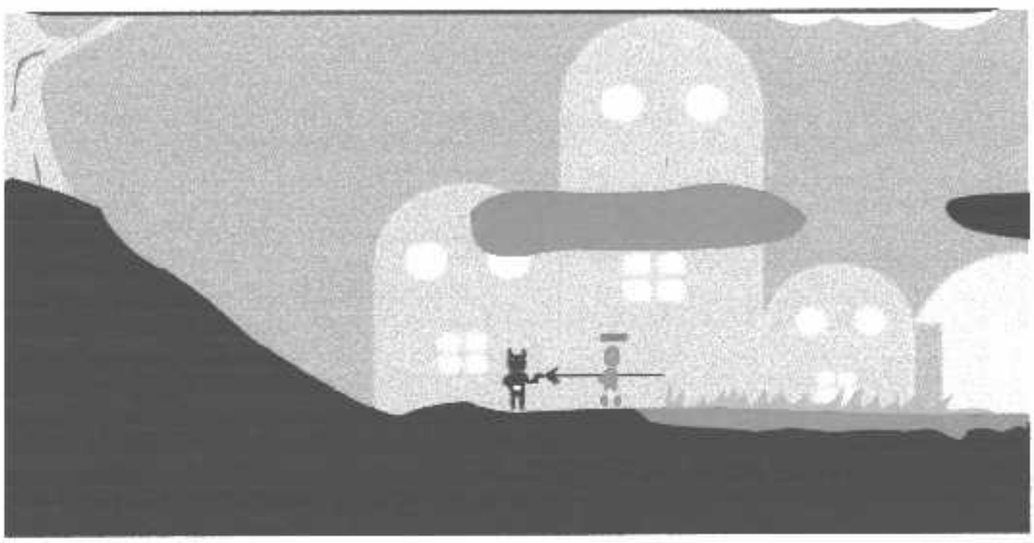
4.2. Pengujian *Game*play

Pengujian *game*play adalah pengujian bagaimana *game* tersebut berjalan sesuai dengan rancangan sistem yang telah dibuat, seperti pada Gambar 4.3.



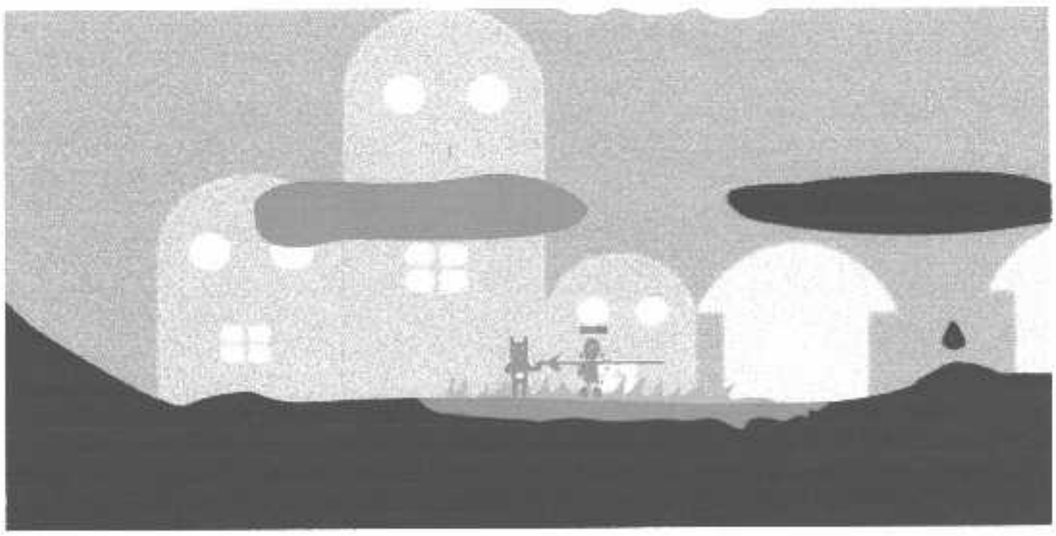
Gambar 4.3. Interaksi Karakter Dengan Dunia di Dalam *Game*

Pada Gambar 4.3 menggambarkan interaksi antara karakter dengan lingkungan di dalam *game*. Pada gambar diatas menunjukkan ketika karakter menyentuh daratan berwarna merah Health Bar karakter akan berkurang, karena warna dari karakter tidak cocok dengan warna daratan. Pada *game* ini karakter bisa berganti warna yaitu warna hitam, kuning, merah, biru, hijau dan oranye.



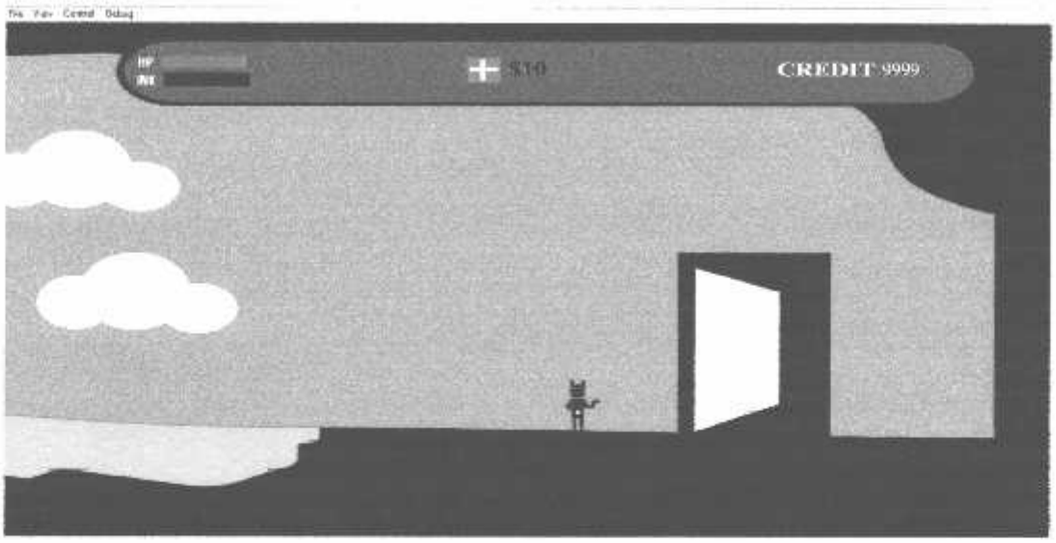
Gambar 4.4. Karakter Menyerang Musuh Dengan Warna Yang Berbeda

Pada Gambar 4.4 menggambarkan ketika karakter menyerang musuh dengan warna yang berbeda, jika warna karakter tidak sama dengan warna musuh yang diserang, maka health bar musuh tidak akan berkurang.



Gambar 4.5. Karakter Menyerang Musuh Dengan Warna Yang Sama

Pada Gambar 4.5menggambarkan ketika karakter menyerang musuh dengan warna yang sama, jika warna karakter sama dengan warna musuh yang diserang, maka *health bar* musuh akan berkurang.



Gambar 4.6. Gerbang terbuka ketika indikator *ink bar* penuh

Pada Gambar 4.6 menggambarkan ketika indikator *ink* penuh maka gerbang menuju level selanjutnya akan terbuka. Jadi karakter harus mengumpulkan cairan *ink* sampai indikator penuh jika ingin membuka gerbang ke level selanjutnya.

4.3. Pengujian *Artificial Intelligence* (AI)

PengujianAI adalah pengujian mengenai fungsi yang berkaitan dengan AI (Artificial Intelligence) yang ada dalam *GameColor Invasion*. Hasil pengujian dari AI *Finite State Machine* dapat dilihat pada Tabel 4.1.

Tabel 4.1. Pengujian AI (Artificial Intelligence)

No	Fungsi	Output	Hasil
1	AI Finite State Machine pada musuh Walker	musuh akan curiga jika pemain berada pada jarak 301-400 pixel dari musuh, jika jarak pemain 300 pixel dari musuh maka musuh akan menyerang dan mengejar pemain, musuh akan kembali patrol ketika jarak pemain >400 pixel dari musuh	Sesuai

2	AI Finite State Machine pada musuh Ghost	Musuh akan menyerang dan mengejar pemain jika jarak pemain berada pada jarak 400 pixel dari musuh, musuh akan berhenti mengejar pemain ketika jarak pemain >400 pixel dari musuh	Sesuai
3	AI Finite State Machine pada musuh Mecha	Musuh akan menembak pemain jika jarak pemain berada pada jarak 200 pixel dari musuh, musuh akan kembali patrol ketika jarak pemain >200 pixel dari musuh	Sesuai
4	AI Finite State Machine pada musuh Boss	Musuh akan berubah warna sesuai dengan tabung warna yang dihancurkan pemain, musuh akan berubah warna menjadi warna abu-abu ketika HP musuh berkurang sebanyak 20 point	Sesuai

Berdasarkan Tabel 4.1. disimpulkan bahwasemua AI (*Artificial Intelligence*) yang ada dalam *Game Color Invasion* berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.4. Pengujian Fungsional

Pengujian fungsional adalah pengujian mengenai proses fungsional yang terjadi dalam *game*. Hasil dari pengujian dapat dilihat pada Tabel 4.2.

Tabel 4.2. Pengujian Fungsional

No	Fungsi	Hasil
1	Tombol pada menu utama berjalan sesuai dengan fungsi	Sesuai
2	Musuh dapat Menyerang karakter player ketika dalam jarak serang	Sesuai
3	Indikator *Health Bar* pada karakter player dan karakter	Sesuai

	musuh berjalan dengan fungsinya	
4	Indikator *Ink Bar* pada karakter player berjalan dengan fungsinya	Sesuai

Berdasarkan Tabel 4.2, disimpulkan bahwasemua fungsiberjalan dengan tingkat keberhasilan 100%sesuai denganyang diharapkan.

4.5. 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 Color Invasion* terdapat pada Tabel 4.3.

Tabel 4.3. Pengujian*Performance*

No	*Processor*	RAM	VGA	OS	Keterangan
1	Intel Pentium 4	512	256MB	WinXP	Berjalan lancar
2	Core 2 Duo	1GB	512MB	Win7	Berjalan lancar
3	Intel Corei3	2GB	1GB	Win8	Berjalan lancar
4	AMD A6	4GB	1GB	Win10	Berjalan lancar
5	Intel Core i5	4GB	1GB	Win8.1	Berjalan lancar

Dari Tabel 4.3, dapat disimpulkan bahwa *Game Color Invasion* dapat dijalankan pada komputer dengan RAM 512MB hingga 4GB dan monitor beresolusi 1366 x 768 pixel, juga dengan menggunakan Windows XP, 7, 8, 8.1 dan Windows 10

4.6. 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 4.4.

Tabel 4.4. Pengujian *Control Player*

Tombol	Fungsi	Hasil
➡	Menggerakkan player ke kanan	Sesuai
⬅	Menggerakkan player ke kiri	Sesuai
⬆	Player melompat, jika di tekan dua kali akan melakukan *double jump*	Sesuai
Z	Untuk merubah warna karakter menjadi hitam	Sesuai
X	Untuk merubah warna karakter menjadi merah	Sesuai
C	Untuk merubah warna karakter menjadi kuning	Sesuai
V	Untuk merubah warna karakter menjadi biru	Sesuai
X+C	Untuk merubah warna karakter menjadi oranye	Sesuai
C+V	Untuk merubah warna karakter menjadi hijau	Sesuai
Spasi	Untuk Menembak	Sesuai

Dari Tabel 4.4 menunjukkan bahwa semua fungsi dari *control player* Berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

BAB V
PENUTUP

5.1. Kesimpulan

Setelah pembuatan *Game Color Invasion*, maka penulis dapat mengambil kesimpulan :

1. Implementasi *Finite State Machine* dapat diterapkan pada *game* 2D bergenre adventure dengan indikasi musuh dapat mengejar dan meyerang player dengan kondisi tertentu.
2. Semua fungsi dari menu, pergerakan Unit pemain dan musuh berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.
3. *Game* dapat dijalankan pada komputer dengan RAM 512MB hingga 4GB juga dengan menggunakan Windows XP, 7, 8, 8.1, dan Windows 10.

5.2. Saran

Setelah dilakukan pengujian terhadap *GameColor Invasion* maka masih ada kekurangan sehingga untuk pengembangan lebih lanjut disarankan:

1. Objek, karakter musuh, karakter pemain grafiknya diperbagus biar lebih menarik.
2. Penambahan level pada *game* karena hanya terdapat 4 level pada *game Color Invasion* ini.
3. Menambahkan fitur save *game* agar pemain bisa melanjutkan *game* nya lagi ketika *game* di keluarkan.

DAFTAR PUSTAKA

Anggra, 2008. *Memahami Teknik Dasar Pembuatan Game Berbasis Flash.* Yogyakarta: Gava Media

Anneahira.com. *Mengenal Macam-Macam Games.* http://www.anneahira.com /macam-macam-*games*.htm, (Diakses pada tanggal 22 Januari 2015, jam 15.00.)

Dahria Muhammad. Agustus 2008. *Kecerdasan Buatan (Artificial Intelligence).* Volume 5, No. 2

Ilman Anwari. 2008. *Penerapan Algoritma Minimax Dengan Optimasi MTD(f) Pada Permainan Catur*

Prasetiyo Budi, Hidayah Rahmah Maulidia. November 2014. *Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3.* Volume 1, No. 2

Setiawan Iwan. 2006. *Perancangan Software Embedded System Berbasis FSM*

Sholichin Riyadhush, Oktoviana Tri Lucky Yasindan Mohamad. *Implementasi Algoritma Dijkstra Dalam Pencarian Lintasan Terpendek Lokasi Rumah Sakit, Hotel Dan Terminal Kota Malang Berbasis Web*

Sofwan Agus, Isnanto Rizal, Maulana Pasca. *Kecerdasan Buatan Dalam Permainan 3D Menggunakan Visual Basic .NET Dan DirectX*

Wandah. *Dasar Pemrograman Flash Game*

Zamroni Rosidi, Suryaman Nizar, Jalaluddin Ahmad. September 2013. *Rancang Bangun Aplikasi Permainan Untuk Pembelajaran Anak Menggunakan HTML 5.* Volume 5, No. 2

LAMPIRAN

BERITA ACARA UJIAN KOMPRE
FAKULTAS TEKNOLOGI INDUSTRI

NAMA	: Edwin Yunanto
NIM	: 12.18.113
JURUSAN	: Teknik Informatika S-1
JUDUL	: Pembuatan Game Color Invasion Menggunakan Metode Finite State Machine (FSM)

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S1) pada:

Hari	: Sabtu
Tanggal	: 16 Januari 2016
Nilai	: 83.77 (A)

Panitia Ujian Skripsi

Ketua Majelis Penguji

Joseph Dedy Irawan, ST. MT.
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I

Febriana Santy W, S. Kom, M. Kom
NIP.P. 1031000425

Dosen Penguji II

Agung P. Sasmito, S. Pd, M. Pd
NIP.P. 1031500499

FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Edwin Yunanto
NIM : 12.18.113
JURUSAN : Teknik Informatika S-1
JUDUL : Pembuatan Game Color Invasion Menggunakan Metode Finite State Machine (FSM)

Tanggal	Penguji	Uraian	Paraf
21 Januari 2016	I	- Revisi Program - Revisi Laporan - Landasan Teori	
21 Januari 2016	II	- Revisi Laporan - Tata Tulis - Komponen Game 1. Goal 2. Rule 3. Storyboard	

Dosen Penguji I

Febriana Santy W, S. Kom, M. Kom
NIP. 1031000425

Dosen Penguji II

Agung P. Sasmito, S. Pd, M. Pd
NIP.P. 1031500499

Dosen Pembimbing I

Joseph Dedy Irawan, ST. MT.
NIP. 19740416 200501 1 002

Dosen Pembimbing II

Sonny Prasetio, ST. MT.
NIP.P. 1031000433

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

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

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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, 23 Oktober 2015

Nomor : ITN-593/I.INF/TA/2015
Lampiran : —
Perihal : *Bimbingan Skripsi*

Kepada : Yth. Bpk/Ibu Joseph Dedy Irawan, ST, MT
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 : EDWIN YUNANTO
Nim : 1218113
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 :

23 Oktober 2015 S/D 23 Maret 2016

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
K e t u a,

Joseph Dedy Irawan, ST., MT.
NIP.: 197404162003021002

Form S-4a

FORMULIR BIMBINGAN SKRIPSI

Nama : Edwin Yunanto

NIM : 12.18.113

Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016

Judul Skripsi : Pembuatan Game Color Invasion Menggunakan Metode Finite State Machine (FSM)

No	Tanggal	Uraian	Paraf Pembimbing
1	19-11-2015	Revisi Desain Sistem	
2	20-11-2015	Revisi BAB I - III	
3	21-11-2015	ACC BAB I - III	
4	23-12-2015	Revisi BAB IV	
5	26-11-2015	Makalah Seminar Progres	
6	19-12-2015	Makalah Seminar Hasil	
7	13-01-2016	ACC BAB IV	
8	14-01-2016	ACC BAB V	

Malang, 25 Januari 2016

Dosen Pembimbing I

Joseph Dedy Irawan, ST. MT.

NIP. P. 19740416 200501 1 002

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

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

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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, 23 Oktober 2015

Nomor : ITN-593/I.INF/TA/2015
Lampiran : —
Perihal : *Bimbingan Skripsi*

Kepada : Yth. Bpk/Ibu Sonny Prasetio, ST, MT
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 : EDWIN YUNANTO
Nim : 1218113
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 :

23 Oktober 2015 S/D 23 Maret 2016

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
K e t u a,

Joseph Dedy Irawan,ST., MT.
NIP.: 197404162003021002

Form S-4a

FORMULIR BIMBINGAN SKRIPSI

Nama : Edwin Yunanto

NIM : 12.18.113

Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016

Judul Skripsi : Pembuatan Game Color Invasion Menggunakan Metode Finite State Machine (FSM)

No	Tanggal	Uraian	Paraf Pembimbing
1	20-11-2015	Revisi BAB I	
2	21-11-2015	Revisi BAB II	
3	23-11-2015	Revisi BAB III, Penulisan dan Perancangan AI Dalam Game	
4	26-11-2015	Makalah Seminar Progres	
5	19-12-2015	Makalah Seminar Hasil	
6	09-01-2016	ACC BAB I - III	
7	11-01-2016	Revisi BAB IV Pengujian Diberi Persentase	
8	12-01-2016	Revisi BAB V Kesimpulan	
9	13-01-2016	ACC BAB IV & BAB V	
10	14-01-2016	ACC Kompre	

Malang, 25 Januari 2016

Dosen Pembimbing II

Sonny Prasetio, ST. MT.

NIP.P. 1031000433

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-Gura No. 2
MALANG

Malang, 23 Oktober 2015

Lampiran : 1(Satu) berkas

Perihal : Kesediaan sebagai Pembimbing Skripsi

Kepada : Yth. Bpk/Ibu **Sonny prasetio ST,MT**
Dosen Pembina Prodi Teknik Informatika S-1
Institut Teknologi Nasional
MALANG

Yang bertanda tangan dibawah ini:

Nama : EDWIN YUNANTO
Nim : 1218113
Prodi : Teknik InformatikaS-1

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing ~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (Proposal Terlampir) :

PEMBUATAN GAME COLOR INVASION MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.
Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami sampaikan terima kasih.

Prodi T. InformatikaS-1
K e t u a,

Joseph Dedy Irawan,ST., MT.
NIP : 197404162005021002

Hormat Kami,

EDWIN YUNANTO

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jln. Bendungan Sigura-gura No. 2
Jln. Raya Karanglo Km2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : EDWIN YUNANTO

Nim : 1218113

Program Studi : Teknik Informatika S1

Dengan ini menyatakan bersedia / ~~tidak bersedia~~ *) membimbing skripsi dari mahasiswa tersebut dengan judul :

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, 28-10-2015

Hormat Kami,

Sonny prasetio ST,MT

Catatan :
Setelah disetujui agar formulir ini diserahkan mahasiswa/i
yang bersangkutan kepada adminstrasi Program Studi untuk diproses lebih lanjut
*) coret yang tidak perlu

Form S-3b

INSTITUT TEKNOLOGI NASIONAL
Jln. Bendungan Sigura-gura No. 2
Jln. Raya Karanglo Km2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : EDWIN YUNANTO

Nim : 1218113

Program Studi : Teknik Informatika S1

Dengan ini menyatakan bersedia / ~~tidak bersedia~~ *) membimbing skripsi dari mahasiswa tersebut dengan judul :

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, _____

Hormat Kami,

Joseph Dedy Irawan, ST, MT

Catatan :
Setelah disetujui agar formulir ini diserahkan mahasiswa/i
yang bersangkutan kepada adminstrasi Program Studi untuk diproses lebih lanjut
*) coret yang tidak perlu

Form S-3b

Lampiran 1. Script Pemain

```
var charSpeed:Number = 10;
var grav:Number = 0;
var gravity:Number = 5;
var maxJump:Number = -35;
var sentuhdaratan:Boolean = false;
var doublejump:Boolean = false;
var jumptime:Number = 5;
var camDesX:Number = char._x;
_root.vcam.credit = 0;

char.onEnterFrame = function()
{
      if (char.attack.pm.lm.hitTest(_root.musuh))
      {
            _root.musuh.hpmusuh.hpm._xscale -= 4;
      }
      if (char.attackb.pmb.lmb.hitTest(_root.musuh))
      {
            _root.musuh.hpmusuh.hpm._xscale -= 4;
      }
      if (char.attack.pm.lm.hitTest(_root.ghost))
      {
            _root.ghost.hudg.hpg._xscale -= 9;
      }
      if (char.attackb.pmb.lmb.hitTest(_root.ghost))
      {
            _root.ghost.hudg.hpg._xscale -= 9;
      }
      if(char.attack.ph.lh.hitTest(_root.robot.rserang.bodi)
      )
      {
```

```
        _root.robot.hudr.hpmr._xscale -= 4;
    }
    If(char.attackb.phb.lhb.hitTest(_root.robot.rserang.bo
    di))
    {
        _root.robot.hudr.hpmr._xscale -= 4;
    }
    if (char.attack.pb.lb.hitTest(_root.boss.blue))
    {
        _root.vcam.hudboss.bosshp._xscale -= 1;
    }
    if (char.attackb.pbb.lbb.hitTest(_root.boss.blue))
    {
        _root.vcam.hudboss.bosshp._xscale -= 1;
    }
    if (char.attack.pk.lk.hitTest(_root.boss.yellow))
    {
        _root.vcam.hudboss.bosshp._xscale -= 1;
    }
    if (char.attackb.pkb.lkb.hitTest(_root.boss.yellow))
    {
        _root.vcam.hudboss.bosshp._xscale -= 1;
    }
    if (char.attack.pg.lg.hitTest(_root.boss.green))
    {
        _root.vcam.hudboss.bosshp._xscale -= 1;
    }
    if (char.attackb.pgb.lgb.hitTest(_root.boss.green))
    {
        _root.vcam.hudboss.bosshp._xscale  = 1;
    }
```

```
if (char.attack.pm.lm.hitTest(_root.boss.red))
{
      _root.vcam.hudboss.bosshp._xscale -= 1;
}
if (char.attackb.pmb.lmb.hitTest(_root.boss.red))
{
      _root.vcam.hudboss.bosshp._xscale -= 1;
}
if (char.attack.po.lo.hitTest(_root.boss.orange))
{
      _root.vcam.hudboss.bosshp._xscale -= 1;
}
if (char.attackb.pob.lob.hitTest(_root.boss.orange))
{
      _root.vcam.hudboss.bosshp._xscale -= 1;
}
char._y += grav;
grav += gravity;
while (daratan.hitTest(char._x, char._y, true))
{
      char._y -= gravity;
      grav = 0;

}
if (daratan.hitTest(char._x, char._y + 5, true))
{
      sentuhdaratan = true;
      jumptime = 5;
}
```

```
else
{
      sentuhdaratan = false;
}
if (Key.isDown(Key.UP) && sentuhdaratan)
{
      grav = maxJump;
      doublejump = true;
}
if(Key.isDown(Key.UP) && doublejump && jumptime > 0)
{
      grav = maxJump;
      jumptime--;
}
else if (Key.isDown(Key.LEFT))
{
      char._x -= charSpeed;
      char._xscale = -_xscale;
      char.body.hitam.play();
      char.body.merah.play();
      char.body.kuning.play();
      char.body.biru.play();
      char.body.hijau.play();
      char.body.oren.play();
      char.attack.ph.gotoAndStop(1);
      char.attack.pm.gotoAndStop(1);
      char.attack.pk.gotoAndStop(1);
      char.attack.pb.gotoAndStop(1);
      char.attack.pg.gotoAndStop(1);
```

```
        char.attack.po.gotoAndStop(1);
    }
    else if (Key.isDown(Key.RIGHT))
    {
        char._x += charSpeed;
        char._xscale = _xscale;
        char.body.hitam.play();
        char.body.merah.play();
        char.body.kuning.play();
        char.body.biru.play();
        char.body.hijau.play();
        char.body.oren.play();
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
    }
    else if (Key.isDown(Key.SPACE))
    {
        char.attack.ph.play();
        char.attack.pm.play();
        char.attack.pk.play();
        char.attack.pb.play();
        char.attack.pg.play();
        char.attack.po.play();
        char.body.hitam.gotoAndStop(1);
        char.body.merah.gotoAndStop(1);
```

```
        char.body.kuning.gotoAndStop(1);
        char.body.biru.gotoAndStop(1);
        char.body.hijau.gotoAndStop(1);
        char.body.oren.gotoAndStop(1);
    }
    else
    {
        char.body.hitam.gotoAndStop(1);
        char.body.merah.gotoAndStop(1);
        char.body.kuning.gotoAndStop(1);
        char.body.biru.gotoAndStop(1);
        char.body.hijau.gotoAndStop(1);
        char.body.oren.gotoAndStop(1);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);

    }
    if (Key.isDown(88))
    {
```

```
        _root.char.body.gotoAndStop(2);
        _root.char.attack.gotoAndStop(2);
        _root.char.attackb.gotoAndStop(2);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(90))
    {
        _root.char.body.gotoAndStop(1);
        _root.char.attack.gotoAndStop(1);
        _root.char.attackb.gotoAndStop(1);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
```

```
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(67))
    {
        _root.char.body.gotoAndStop(3);
        _root.char.attack.gotoAndStop(3);
        _root.char.attackb.gotoAndStop(3);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(86))
    {
        _root.char.body.gotoAndStop(4);
        _root.char.attack.gotoAndStop(4);
        _root.char.attackb.gotoAndStop(4);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
```

```
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(Key.ESCAPE))
    {
        _root.gotoAndStop("menu",1);
    }
    if (Key.isDown(86) && Key.isDown(67))
    {
        _root.char.body.gotoAndStop(5);
        _root.char.attack.gotoAndStop(5);
        _root.char.attackb.gotoAndStop(5);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
```

```
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(88) && Key.isDown(67))
    {
        _root.char.body.gotoAndStop(6);
        _root.char.attack.gotoAndStop(6);
        _root.char.attackb.gotoAndStop(6);
        char.attack.ph.gotoAndStop(1);
        char.attack.pm.gotoAndStop(1);
        char.attack.pk.gotoAndStop(1);
        char.attack.pb.gotoAndStop(1);
        char.attack.pg.gotoAndStop(1);
        char.attack.po.gotoAndStop(1);
        char.attackb.phb.gotoAndStop(1);
        char.attackb.pmb.gotoAndStop(1);
        char.attackb.pkb.gotoAndStop(1);
        char.attackb.pbb.gotoAndStop(1);
        char.attackb.pgb.gotoAndStop(1);
        char.attackb.pob.gotoAndStop(1);
    }
    if (Key.isDown(Key.SPACE) && Key.isDown(Key.LEFT))
    {
        char.attackb.phb.play();
        char.attackb.pmb.play();
        char.attackb.pkb.play();
        char.attackb.pbb.play();
        char.attackb.pgb.play();
        char.attackb.pob.play();
```

```
            char.body.hitam.play(1);
            char.body.merah.play(1);
            char.body.kuning.play(1);
            char.body.biru.play(1);
            char.body.hijau.play(1);
            char.body.oren.play(1);
      }
      elseif(Key.isDown(Key.SPACE)&&Key.isDown(Key.RIGHT))
      {
            char.attackb.phb.play();
            char.attackb.pmb.play();
            char.attackb.pkb.play();
            char.attackb.pbb.play();
            char.attackb.pgb.play();
            char.attackb.pob.play();
            char.body.hitam.play(1);
            char.body.merah.play(1);
            char.body.kuning.play(1);
            char.body.biru.play(1);
            char.body.hijau.play(1);
            char.body.oren.play(1);
      }
      else
      {
            char.attackb.phb.gotoAndStop(1);
            char.attackb.pmb.gotoAndStop(1);
            char.attackb.pkb.gotoAndStop(1);
            char.attackb.pbb.gotoAndStop(1);
            char.attackb.pgb.gotoAndStop(1);
            char.attackb.pob.gotoAndStop(1);
      }
};
```

Lampiran 2. Script Musuh *Walker*

```
var gravitasi:Number = 0;
var gravitys:Number = 5;
var sentuhdaratans:Boolean = false;
var gerakkiri:Boolean = true;
var curiga:Boolean = false;
var enemySpeed:Number = 2;
var serang:Boolean = false;
var kecepatanserang:Number = 7;

musuh.onEnterFrame = function()
{
	if (char.body.hitTest(_root.musuh))
	{
		_root.vcam.hud.hp._xscale -= 0.8;
	}
	musuh._y += gravitasi;
	gravitasi += gravitys;
	while (daratan.hitTest(musuh._x, musuh._y, true))
	{
		musuh._y -= gravitys;
		gravitasi = 0;
	}
	if (daratan.hitTest(musuh._x, musuh._y + 5, true))
	{
		sentuhdaratans = true;
	}
	else
	{
		sentuhdaratans = false;
	}
```

```
if (gerakkiri && !curiga && !serang)
{
      this._x -= enemySpeed;
      this._xscale = _xscale;
}
else if (!curiga && !serang)
{
      this._x += enemySpeed;
      this._xscale = -_xscale;
}
if (musuh._x >= 2570)
{
      gerakkiri = true;
}
else if (musuh._x <= 2051)
{
      gerakkiri = false;
}
it (char._y > this._y && char._x > this._x)
{
      serang = false;
      curiga = false;
}
if (char._x < this._x + 400 && char._x > this._x -
400)
{
      this.gotoAndStop(2);
      curiga = true;
}
else
```

```
{
      curiga = false;
      this.gotoAndStop(1);
}
if (char._x < this._x + 300 && char._x > this._x -
300)
{
      serang = true;
}
if (char.hitTest(_root.musuh))
{
      serang = true;
      this.gotoAndStop(3);
}
if (char._x < this._x && serang)
{
      this._x -= 6;
      this.gotoAndStop(3);
      this._xscale = -_xscale;
}
else if (char._x > this._x + 350)
{
      serang = false;
}
if (char._x > this._x && serang)
{
      this._x += 6;
      this.gotoAndStop(3);
      this._xscale = _xscale;
```

```
        }
        else if (char._x > this._x - 350)
        {
                serang = false;
        }
};
```

Lampiran 3.Script Musuh *Ghost*

```
var speed:Number = 10;
var turnRate:Number = 2.5;
var agroRange:Number = 400;
var moveX:Number = 0;
var moveY:Number = 0;

function doFollow(follower:MovieClip, target:MovieClip)
{
        var distanceX:Number = target._x - follower._x;
        var distanceY:Number = target._y - follower._y;

distanceTotal = Math.sqrt(distanceX * distanceX +
distanceY * distanceY);

        if (distanceTotal <= agroRange)
        {

var moveDistanceX:Number = turnRate * distanceX /
distanceTotal;
var moveDistanceY:Number = turnRate * distanceY /
distanceTotal;

                moveX += moveDistanceX;
                moveY += moveDistanceY;
```

```
var totalmove = Math.sqrt(moveX * moveX + moveY * moveY);

            moveX = speed * moveX / totalmove;
            moveY = speed * moveY / totalmove;

            follower._x += moveX;
            follower._y += moveY;

follower._rotation = 180 * Math.atan2(moveY, moveX) /
Math.PI;
      }
}
_root.onEnterFrame = function()
{
      doFollow(ghost,char);
};
```

Lampiran 4.Script Musuh *Mecha*

```
var gravitasir:Number = 0;
var gravitysr:Number = 5;
var sentuhdaratansr:Boolean = false;
var gerakkirir:Boolean = true;
var enemySpeedr:Number = 3;
var serangr:Boolean = false;
var kecepatanserangr:Number = 7;

robot.onEnterFrame = function()
{
     if (char3.body.hitTest(_root.robot.rgerak))
     {
          _root.vcam3.hud.hp._xscale -= 0.8;
     }
     if (char3.body.hitTest(_root.robot.rserang.bodi))
     {
          _root.vcam3.hud.hp._xscale -= 0.8;
     }
     if(char3.body.hitTest(_root.robot.rserang.bullets))
     {
          _root.vcam3.hud.hp._xscale -= 1;
     }
     robot._y += gravitasir;
     gravitasir += gravitysr;
     while (daratan.hitTest(robot._x, robot._y, true))
     {
          robot._y -= gravitysr;
          gravitasir = 0;
     }
     if (daratan.hitTest(robot._x, robot._y + 5, true))
     {
```

```
            sentuhdaratansr = true;
        }
        else
        {
            sentuhdaratansr = false;
        }
        if (gerakkirir && !serangr)
        {
            this._x -= enemySpeedr;
            this._xscale = _xscale;
        }
        else if (!serangr)
        {
            this._x += enemySpeedr;
            this._xscale = _xscale;
        }
        if (robot._x >= -1150)
        {
            gerakkirir = true;
        }
        else if (robot._x <= -1600)
        {
            gerakkirir = false;
        }
        if (_root.char3.hitTest(_root.robot.area))
        {
            _root.robot.gotoAndStop(2);
            serangr = true;
```

```
            this._xscale = _xscale;
      }
      else
      {
            serangr = false;
      }
};
```

Lampiran 5.Script Musuh *Boss*

```
function onEnterFrame()
{
      if (_root.vcam4.hudboss.bosshp._xscale == 80)
      {
            _root.boss.gotoAndStop(1);
            _root.laserboss.gotoAndStop(1);
            unloadMovie(pintub);

      }
      if(_root.vcam4.hudboss.bosshp._xscale == 59)
      {
            _root.boss.gotoAndStop(1);
            _root.laserboss.gotoAndStop(1);
            unloadMovie(pintuk);

      }
      if (_root.vcam4.hudboss.bosshp._xscale == 39)
      {
            _root.boss.gotoAndStop(1);
            _root.laserboss.gotoAndStop(1);
            unloadMovie(pintuh);
```

```
        }
        if (_root.vcam4.hudboss.bosshp._xscale == 19)
        {
            _root.boss.gotoAndStop(1);
            _root.laserboss.gotoAndStop(1);
            unloadMovie(pintum);
            unloadMovie(pintum1);
        }
        if (_root.vcam4.hudboss.bosshp._xscale <= 0)
        {
            _root.boss.gotoAndStop(7);
        }
}
```