

**RANCANG BANGUN GAME SURVIVAL ARMYBOT
MENGUNAKAN METODE FINITE STATE MACHINE
(FSM)**

SKRIPSI



**Disusun Oleh :
RUDY SETYO UTOMO
12.18.003**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2016**

LEMBAR PERSETUJUAN

**RANCANG BANGUN GAME SURVIVAL ARMYBOT 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 :

RUDY SETYO UTOMO

NIM : 12.18.003

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

Joseph Dedy Irawan, ST. MT
NIP. 197404162005011002

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

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
2016**

LEMBAR KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Rudy Setyo Utomo
NIM : 1218003
Program Studi : Teknik Informatika S-1
Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

**“Rancang Bangun Game Survival Armybot Menggunakan Metode Finite
State Machine (FSM)”**

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 2016

Yang membuat Pernyataan



Rudy Setyo Utomo

Abstrak

Game merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaa animasi haruslah memahami pembuatan Game. Atau jika ingin membuat Game, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan. Dalam bahasa Indonesia "Game" dapat diartikan sebagai permainan. Permainan adalah suatu kegiatan yang di dalamnya terdapat peraturan-peraturan dan sebuah sistem yang mana pemain terlibat dalam suatu konflik buatan. Agar tidak terjadi kecurangan di dalam sebuah permainan, maka peraturan dibuat membatasi perilaku pemain dan menentukan jalannya permainan. Sehingga setiap pemain memiliki peluang yang sama untuk memenangkan permainan.

Survival merupakan Game petualangan (adventure Game) dimana pemain diuntut kemampuan berfikirnya untuk menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter, menggunakan benda- benda yang tepat dan diletakkan di tempat yang tepat. Gameplay jenis ini adalah keharusan player memecahkan bermacam- macam misi melalui interaksi dengan karakter lain dalam game tersebut

Algoritma Finite State Machine (FSM) diterapkan kepada musuh, dengan memberikan sensor sebagai indicator respon terhadap gerakan pemain. Game adventure ini dibangun menggunakan aplikasi Unity dan bahasa pemrograman JavaScript dan CSharp, sehingga proses perancangan karakter, animasi, dan algoritma FSM dapat dicapai. Dalam game ini terdapat dua jenis musuh, yaitu hewan buas dan yang terakhir boss berbentuk monster. Perilaku yang dilakukan oleh musuh meliputi melihat pemain, mengejar pemain, dan menyerang pemain.

Dari hasil pengujian fungsional, pengujian AI(Artificial Intelligence), dan control berjalan sesuai. Dari pengujian performa game dapat dijalankan di komputer dengan spesifikasi RAM 1-4GB, VGA minimal 512MB, dengan OS Windows XP, 7, dan 8.1. Dari hasil pengujian pada pengguna dapat disimpulkan bahwa Hasil pengujian user dari 20 responden gameplay dari game ini sebesar 15 responden menyatakan menarik, desain karakter musuh sebesar 16 responden menyatakan menarik, persentasi pengujian user dari 20 responden 53,12% menyatakan menarik, 28,75% menyatakan cukup menarik, dan 18,12% menyatakan kurang menarik.

Kata kunci : *Game, FSM, Survival, Armybot, Adventure*

KATA PENGANTAR

Puji syukur kehadirat Allah yang maha kuasa, karena telah memberikan rahmat dan hidayah-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul “RANCANG BANGUN GAME SURVIVAL ARMYBOT MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)” 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 kesempatan ini dengan segala kerendahan hati, perkenankanlah penyusun mengucapkan terima kasih 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 S-1, Institut Teknologi Nasional Malang.
4. Sonny Prasetyo, ST.MT, selaku Sekretaris Program Studi Teknik Informatika S-1, Institut Teknologi Nasional Malang.
5. Joseph Dedy Irawan, ST.MT, selaku Dosen Pembimbing I.
6. Febriana Santi W, S.Kom.M.Kom, selaku Dosen Pembimbing II.
7. Serta semua pihak yang telah membantu secara langsung maupun tidak langsung sehingga penulis dapat menyelesaikan laporan ini.

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

Malang, Januari 2016

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR KEASLIAN	iii
ABSTRAK.....	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
BAB I. PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metode Penelitian	3
1.6 Sistematika Penulisan	4
BAB II. LANDASAN TEORI	5
2.1 Game	5
2.2 Sejarah Game	5
2.3 Syarat Perancangan Game	10
2.4 Jenis Game	11
2.5 Kecerdasan Buatan.....	13
2.6 Finite State Machine	15
BAB III. ANALISA DAN PERANCANGAN	16
3.1 Analisis Kebutuhan	16
3.1.1 Kebutuhan Game	16
3.1.2 Kebutuhan Non Fungsional	16
3.2 Perancangan Game.....	17
3.2.1 <i>Gameplay</i>	18

3.2.2	Karakter	19
3.2.3	Terrain	22
3.2.4	Perancangan Struktur Menu	24
3.2.5	Alur Game	24
3.2.6	Perancangan Finite State Machine (FSM).....	26
3.2.7	Perancangan Aksi Karakter	28
3.2.8	Perancangan Antarmuka.....	29
BAB IV. IMPLEMENTASI DAN PENGUJIAN		35
4.1	Implementasi Hasil	35
4.1.1	Tampilan Menu Utama.....	35
4.1.2	Tampilan <i>How Play</i>	35
4.1.3	Tampilan <i>About</i>	36
4.1.4	Tampilan <i>Gallery</i>	36
4.1.5	Tampilan <i>Storyboard</i>	37
4.1.6	Tampilan Level 1.....	37
4.1.7	Tampilan Level 2.....	38
4.1.8	Tampilan Level 3.....	38
4.1.9	Tampilan Level 4.....	39
4.1.10	Tampilan <i>Game Over</i>	39
4.2	Pengujian Fungsional	40
4.2.1	Pengujian AI.....	41
4.2.2	Pengujian <i>Control Player</i>	42
4.2.3	Pengujian Performa	43
4.2.4	Pengujian pada Pengguna.....	44
BAB V. PENUTUP		46
5.1	Kesimpulan	46
5.2	Saran	46
DAFTAR PUSTAKA		47
LAMPIRAN		48

DAFTAR GAMBAR

Gambar 2.1	FSM <i>Survival Armybot</i>	15
Gambar 3.1	Struktur Menu.....	24
Gambar 3.2	Flowchart Alur Game.....	25
Gambar 3.3	Penerapan FSM pada <i>Monster Clynk</i>	26
Gambar 3.4	Penerapan FSM <i>Trap</i>	27
Gambar 3.5	Penerapan FSM <i>Fountain</i>	27
Gambar 3.6	Penerapan FSM <i>Enemy Smoe</i>	28
Gambar 3.7	Perancangan Aksi pada Player.....	28
Gambar 3.8	Perancangan Antarmuka Menu Utama.....	28
Gambar 3.9	Perancangan Antarmuka Menu <i>How to Play</i>	30
Gambar 3.10	Perancangan Antarmuka <i>Option</i>	30
Gambar 3.11	Perancangan Antarmuka <i>About</i>	31
Gambar 3.12	Perancangan Antarmuka <i>Level 1</i>	32
Gambar 3.13	Perancangan Antarmuka <i>Level 2</i>	32
Gambar 3.14	Perancangan Antarmuka <i>Level 3</i>	33
Gambar 3.15	Perancangan Antarmuka <i>Level 4</i>	34
Gambar 3.16	Perancangan Antarmuka <i>Respawn</i>	34
Gambar 4.1	Tampilan Menu Utama.....	35
Gambar 4.2	Tampilan Menu <i>How Play</i>	36
Gambar 4.3	Tampilan Menu <i>About</i>	36
Gambar 4.4	Tampilan Menu <i>Gallery</i>	36
Gambar 4.5	Tampilan <i>Storyboard</i>	37
Gambar 4.6	Tampilan Level 1.....	37
Gambar 4.7	Tampilan Level 2.....	38
Gambar 4.8	Tampilan Level 3.....	38
Gambar 4.9	Tampilan Level 4.....	39
Gambar 4.10	Tampilan <i>Game Over</i>	39

DAFTAR TABEL

Tabel 3.1	Karakter <i>Game</i>	19
Tabel 3.2	<i>Terrain Game</i>	22

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan Game di dunia semakin pesat, tidak terkecuali di Indonesia. Game saat ini sudah menjadi alternatif hiburan bagi semua kalangan. Industri dan bisnis pengembangan Game juga sudah menjadi suatu hal yang menjanjikan, terbukti dengan banyaknya perusahaan pengembang Game di Amerika, Eropa dan Asia. Negara Indonesia masih terhitung sebagai konsumen Game, dilihat dari tingkat konsumsi game yang sangat tinggi. Banyak perusahaan-perusahaan yang membawa game-game bagus dari luar negeri untuk dimainkan di Indonesia. Di balik semua itu, tersirat keinginan dari putra putri Indonesia untuk membuat Game mereka sendiri, tetapi masih saja terbentur dengan masalah ilmu, biaya dan tingkat kesulitan pembuatan game yang memang cukup tinggi, padahal kreativitas, inovasi dan imajinasi mereka tidak kalah dengan pengembang game luar negeri. Untuk itu penulis berusaha membuat Game Survival Armybot ini dari Unity dan Maya menggunakan metode Finite State Machines (FSM).

Dalam Game jenis petualangan (adventure Game) pemain dituntut kemampuan berfikirnya untuk menganalisa situasi secara visual, memecahkan rangkaian peristiwa, dan menggunakan senjata sebagai alat pertarungan. Dengan ini secara tidak langsung game jenis petualangan bermanfaat karena sebenarnya membantu untuk perkembangan otak, untuk meningkatkan konsentrasi dan melatih untuk memecahkan masalah dengan tepat dan cepat karena dalam game terdapat berbagai konflik atau masalah yang menuntut kita untuk menyelesaikannya dengan cepat dan tepat. Oleh karena itu penulis membuat game jenis petualangan.

Jenis metode Finite State Machine (FSM) sangat sesuai digunakan dalam pembuatan jenis petualangan karena FSM sendiri merupakan metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal yaitu State (Keadaan), Event (kejadian) dan action (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem

akan berada pada salah satu state yang aktif. Sistem dapat beralih atau bertransisi menuju state lain jika mendapatkan masukan atau event tertentu. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.

1.2. Rumusan Masalah

Adapun rumusan masalah yang akan dibahas dalam skripsi ini adalah

1. Bagaimana merancang game *Survival Armybot*?
2. Bagaimana implementasi kecerdasan buatan *Finite State Machine* pada musuh dalam game *Survival Armybot*?

1.3. Tujuan Penelitian

Adapun yang menjadi tujuan penulisan dalam penyusunan Skripsi adalah sebagai berikut :

1. Membangun Game *Survival Armybot* menggunakan perangkat lunak Unity
2. Mengimplementasikan kecerdasan buatan *Finite State Machine* musuh dalam game.

1.4. Batasan Masalah

Batasan masalah dalam membangun game *Survival Armybot* Menggunakan Metode *Finite State Machine* (FSM) antara lain:

1. Platform yang digunakan dalam membangun Game *Survival Armybot* adalah Unity dengan bahasa pemrograman JavaScript dan CSharp (C#.)
 2. Metode yang digunakan adalah Metode *Finite State Machine* (FSM) untuk menentukan perilaku lawan kepada pemain.
 3. Game yang dibangun adalah jenis *Survival Game*.
 4. Adanya lawan bertarung pemain yaitu berupa robot dan musuh boss yaitu Monster misterius yang harus dikalahkan oleh pemain.
 5. Software yang digunakan untuk melakukan desain model dan objek adalah Autodesk Maya 2013.
 6. Game dibuat hanya untuk *Single Player* atau hanya untuk 1 orang.
 7. Option pada main menu game tidak berfungsi.
-

8. Efek pada game berjumlah 3
9. Alur FSM tidak diproses dalam script.
10. Sound pada game hanya terdapat pada trigger dan main sound
11. Storyboard masih dalam bentuk tulisan.
12. Tidak terdapat animasi peluru pada serangan *player*

1.5. Metodologi

Tahapan-tahapan pada metode penelitian yang digunakan dalam pelaksanaan penelitian ini, meliputi :

1. Studi literature.

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

2. Analisa kebutuhan.

Dengan informasi yang telah diperoleh akan dianalisa agar diperoleh suatu kerangka sebagai sumber acuan perancangan game.

3. Perancangan.

Setelah pengumpulan data selesai, berlanjut pada tahap perancangan game.

4. Implementasi

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

5. Pengujian

Jika aplikasi telah selesai selanjutnya ke tahap pengujian, hal ini dilakukan untuk mengetahui apakah game yang dirancang sudah benar atau mendapat kesalahan.

1.6. Sistematika Penulisan

Sistematika penyusunan proposal skripsi ini untuk memberikan gambaran secara umum dari proposal skripsi yang dibuat yang meliputi bab-bab sebagai berikut :

BAB I : PENDAHULUAN

Berisi Latar Belakang, Rumusan Masalah, Tujuan, Manfaat, Batasan Masalah, Metode Penelitian, dan Sistematika Penulisan

BAB II : LANDASAN TEORI

Membahas tentang software yang akan digunakan, bahasa pemrograman yang digunakan, metode yang digunakan serta berbagai macam software pendukung yang akan digunakan dalam pembuatan aplikasi ini.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Membahas tentang gambaran umum dari tata letak menu dan tampilan yang akan dibuat. Dalam perancangan sistem akan ditentukan rancangan struktur menu, Flowchart dan Layout Prototype.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Membahas tentang hasil implementasi dan analisis hasil uji coba program. Serta memaparkan hasil-hasil dari tahap penelitian, tahap analisis, desain, implementasi desain, hasil testing dan implementasinya, berupa penjelasan teoritik, baik secara kualitatif, kuantitatif, atau secara statistik.

BAB V : PENUTUP

Berisi kesimpulan dan saran. Kesimpulan didapat dari ulasan data-data penelitian, menyimpulkan bukti-bukti yang dapat diperoleh dan akhirnya menarik intisari apakah hasil yang didapat layak untuk digunakan dan diimplementasikan.

BAB II

LANDASAN TEORI

2.1 Game

Game merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaa animasi haruslah memahami pembuatan Game. Atau jika ingin membuat Game, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.

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

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

2.2 Sejarah Game

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

1. Game Generasi Pertama.

Pada tahun 1972, saat itu banyak orang yang masih belum mengenal *concole Game* dan *computer Game*. Namun sebuah perusahaan bernama Magnavox meluncurkan sebuah video Game pertama yaitu *Odyssey*. Magnavox *Odyssey*, *console Game* pertama di dunia yang

mengoperasikan *Pong*. Tidak lama setelah itu sebuah *Game arcade* legendaris *Atari* berjudul "*Pong*" muncul. *Pong* merupakan sebuah *Game* sederhana yang mengambil konsep permainan tenis, dengan 1 bola dan 2 papan di kiri dan di kanan. Pada tahun 1975 akhirnya *Magnavox* menyerah dan menghentikan produksi *Odyssey*. Sebagai gantinya mereka mengikuti jejak *Atari* dengan memproduksi mesin dingdong bernama *Odyssey 100*, yang khusus menyajikan *Game Pong*.

2. *Game* Generasi Kedua.

Pada tahun 1976, *Fairchild* mencoba menghidupkan kembalidunia *video Game* dengan menciptakan *VES (Video Entertainment System)*. *VES* adalah mesin pertama yang disebut *console*. *Concole* ini menggunakan kaset *magnetic* yang disebut *cartridge*. Konsep ini kemudian diikuti oleh beberapa produsen lain termasuk *Atari*, *Magnavox*, dan *RCA*, ketiga persahaan tersebut juga merilis *console* serupa.

Pada tahun 1977, dunia *Game console* menjadi tidak populer, *Game* yang tidak berhasil menarik minat. *Fairchild* dan *RCA* mengalami kebangkrutan. Praktis, hanya ada *Atari* dan *Magnavox* yang masih bertahan di dunia *video Game*. Kemudian pada tahun 1978, *Magnavox* meluncurkan *Odyssey 2*. Seperti halnya *Odyssey* pertama, *console* ini pun gagal di pasaran. Tak lama berselang *Atari* memunculkan *console* legendaris, *Atari 2600*, yang terkenal dengan *Game Space Invaders*.

Pada tahun 1980, berbagai produsen *console* muncul dan mereka mengambil *Atari 2600* menjadi konsep dan perkembangan dunia *Game* pun semakin pesat. Pada tahun 1983, dunia *video Game* kembali ambruk. *Game* yang kurang kreatif membuat *conclose* kembali mendapat sambutan dingin, apalagi *personal computer* saat itu semakin canggih. Masyarakat lebih memilih membeli *PC*, karena selain untuk bermain *PC* juga produktif untuk bekerja.

3. *Game* Generasi Ketiga.

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

4. *Game* Generasi Keempat.

Pada tahun 1988, NES mendapat sambutan hangat di seluruh dunia. Sampai akhirnya sebuah perusahaan bernama *Sega* mencoba menyaingi produk *Nintendo*, *Sega* merilis *console next-generation* mereka, yaitu *Sega Mega Drive* (yang juga dikenal dengan *Sega Genesis*). Console ini menyajikan gambar yang lebih tajam dan animasi yang lebih halus dibandingkan dengan NES. Console ini berhasil memberi tekanan, tetapi NES tetap bertahan dengan angka penjualan tinggi. Selanjutnya pada tahun 1990, *Nintendo* kembali mengebrak dengan *console next-generation* mereka dengan produk *NESES* (*Super Nintendo Entertainment System*). Selama empat tahun *Nintendo* dan *Sega* menjadi musuh bebuyutan, meskipun ada beberapa produsen seperti *SNK*, dengan *NeoGeo*-nya, *NEC* dengan *TurboGrafx*-dan *Philips CD-i*. Rivalitas yang legendaris, *Super NES* dan *Mario Bross* sebagai ikonnya melawan *Sega Mega Drive* dengan *Sonic The Hedgehog* sebagai ikonnya.

5. *Game* Generasi Kelima.

Pada tahun 1990-1994, *Sega* dan *Nintendo* kembali bersaing. Berbagai *Game* fenomenal dirilis. SNES menyertakan chip *Super FX* pada cartridge mereka, dan *Sega* menggunakan *Sega Virtual Processor*.

Keduanya bertujuan untuk meningkatkan kualitas grafis dari *Game*. Alhasil, SNES dan *Sega* saling beradu dengan *Game*-nya seperti *Donkey Kong Country* (SNES) dan *Vectorman* (*vega*). Tetapi, pada tahun 1993 sebuah perusahaan bernama Panasonic merilis console-nya yaitu Panasonic 3DO. Ini adalah console pertama yang menggunakan CD sebagai pengganti cartridge. Harganya sangat mahal membuat console ini tidak populer. 3DO tidak bertahan lama dan harus segera menghentikan produksinya. 3DO adalah console pertama yang menggunakan media CD. Akhirnya, pada tahun 1994 Atari kembali memunculkan console barunya untuk menandingi Nintendo dan *Sega*. Atari Jaguar jelas jauh lebih canggih ketimbang NES maupun *Sega Drive*, tetapi penggunaannya yang sulit menjadi batu sandungan. Pada tahun yang sama Sony merilis console super canggih, yaitu PlayStation. Atari pun bangkrut dan akhirnya melakukan merger. Console Jepang ini segera mendapat sambutan hangat hingga saat ini.

6. Game Generasi Keenam.

Pada tahun 1998 setelah jatuhnya Nintendo dan *Sega*, kini dunia console menjadi milik Sony. PlayStation menjadi raja dan bisa dibilang tidak memiliki pesaing, *Sega* mencoba meluncurkan *Sega Dreamcast* untuk mematahkan dominasi Sony PlayStation, tetapi kembali gagal dan akhirnya tahun itu juga *Sega* mengundurkan diri dari dunia produsen console. Di tahun 2000, Sony semakin merajalela ketika mereka berhasil merilis console barunya, bernama PlayStation2 yang telah berbasis DVD. Nintendo mencoba bertahan di dunia console dengan merilis GameCube. Console ini tidak menggunakan DVD 12cm biasa, melainkan DVD yang berukuran lebih kecil, yaitu 8cm. Ukuran medianya yang tidak umum menjadi GameCube kurang populer. Satu-satunya pesaing serius PlayStation2 adalah Xbox. Sebuah console keluaran Microsoft yang menggebrak dengan tampilan visual yang sangat tajam dan berkekuaitas. Sayangnya Game Xbox tidak sepopuler PlayStation2.

7. Game Generasi Ketujuh.

Tahun 2005, Xbox teramat meluncur ke pasaran dibanding PlayStation2, dan support Game tenar juga sangat minim. Tetapi, Microsoft seolah beajar dari kesalahan. Pada saat Sony masih melakukan riset untuk console PlayStation3 yang menggunakan Blu-Ray, Microsoft kali ini telah mengambil langkah inifatif dengan mengeluarkan Xbox 360, sebuah console generasi terkini yang memanfaatkan media HD-DVD. Xbox 360 hadir dengan segudang fitur istimewa, mulai dari grafis hingga kecerdasan yang terdapat dalam permainan (Artificial Intelligence). Yang akhirnya memperoleh Best Game Of Years 2006, selang seminggu sebelum Nitendo meluncurkan terobosannya, yaitu Nitendo Wii. Posisi PlayStation3 kurang menuntungkan. Selain harganya mahal, console ini juga harus didukung dengan kualitas device yang canggih.

8. Game Generasi Ke Delapan.

Handheld Merebaknya popularitas Game membuat berbagai perusahaan elektronik berusaha membuat terobosan baru. Di antaranya adalah membuat sebuah mesin Game berukuran kecil, yang bisa dibawa kemanapun. Sekitar tahun 1976-1979, sejarah video Game saku ini bermula, beberapa piranti dan mattel dirilis ke pasaran, tetapi tidak begitu populer. Demikian dengan handheld buatan Milton Bradley yang dilempar ke pasaran.

Pada tahun 1980-1984 perusahaan-perusahaan Jepang mulai merambah pasar Handheld, tetapi tetap saja hasilnya. Berlanjut hingga 1984, hingga Gameboy pun muncul. Handheld buatan Nitendo ini begitu diminati dan dinobatkan sebagai Handheld pertama di dunia yang angka penjualannya dikatakan sukses.

Pada tahun 1989, Atari mengakhiri era Handheld hitam putih. Produk andalannya adalah Atari Lynx yang membawa dimensi baru. Ini adalah Handheld pertama yang mampu menampilkan warna, sekaligus animasi 3 dimensi (3D) yang sederhana. Pada awal 1990 dunia Handheld semakin menggila, NEC, perusahaan elektronik terkemuka di Jepang

membuat Handheld yang mampu membuat animasi 3D lebih kompleks, karena menerapkan konsep grafis 3D untuk PC.

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

Akhirnya pada tahun 2000-2006, Sony merilis handheld pertama yang menggunakan cakram bernama PSP (PlayStation Potable) dan dibarengi dengan hadirnya Nitendo DS, yang menggunakan konsep dual screen. Lalu disusul oleh Game Boy Micro dan Game Park XGP. Nitendo DS Lite dan Peitcan VG Pocket Caplet menjadi handheld terbaru yang diempas ke pasaran.

2.3 Syarat Perancangan Game

Berikut adalah syarat utama dalam merancang sebuah game, yaitu:

1. Game are not linear

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

2. Game have a goal

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

3. Game must be winnable

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

4. Start, Middle and Ending Of The Game

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

2.4 Jenis Game

Berikut ini adalah jenis game yang ada hingga saat ini:

1. Action Game

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

2. Adventure Game

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

3. Casual Game

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

4. Educational Game

Merupakan game menggunakan media pembelajaran interaktif sebagai intinya. Yang mana game ini untuk membrikan pembelajaran, berkaitan dengan penerapan ilmu pengetahuan, ataupun memberi stimulus pada kinerja saraf dan otak pemain.

Contoh: Barbie Secret Agent, Blue's Clues Learning Time, Zuma, Angry Bird, dan sebagainya.

5. Role Playing Game

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

6. Simulation Game

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

7. Sport Game

Adalah jenis game dengan mengadopsi pada kegiatan olah raga di kehidupan nyata serta dilengkapi dengan aturan mainnya. Game jenis ini merupakan game yang menbatkan kecerdasan yang diterapkan pada musuh, kecerdasan lainnya juga diterapkan pada wasit, juri, dan peraturan pertandingan (penentuan pelanggaran). Contoh: Pro Evolution Soccer (PES), Winning Eleven, Street Ball, WWE, dan sebagainya.

8. Strategy Game

Merupakan jenis game yang pada umumnya identik dengan situasi peperangan, biasanya menampilkan pemandangan dari atas (angel from top), arena tempur digambarkan sebagai peta yang bersegmen, pemain dimulai dengan membangun sendiri markas dan perlengkapan pemain. Game ini tidak memainkan tensi yang tinggi, tapi menuntun pemain untuk perfikir logis, cepat dan tepat dalam memenangkan pertempuran. Contoh: Red Alert, Vietnam Squad Battles, Resign Of Chaos, Soccer Manager, dan sebagainya.

2.5 Kecerdasan Buatan

Kecerdasan buatan di definisikan sebagai kecerdasan yang ditujukan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

Berikut adalah contoh AI yang ada pada game :

a. Pathfinding

proses pencarian path tercepat dari titik asal ke titik tujuan dengan menghindari berbagai halangan sepanjang path yang ditempuh. Secara umum pathfinding dapat dibedakan menjadipathfinding statik dan dinamik.

Ada beberapa algoritma yang dapat digunakan untuk mencari rute terpendek, misalnya Best First Search, Greedy Search, Dijkstra, A* (A-Star) Search. Pada penelitian yang dilakukan oleh Michael Alexander Diojo yang berjudul analisis perbandingan komputasi algoritma Dijkstra, A* (A-Star), dan Floyd-Warshall (Michael Alexander Diojo, 2013), didapat kesimpulan bahwa algoritma A* (A-Star) memiliki beban komputasi dan waktu simulasi yang paling kecil dibandingkan dengan algoritma Dijkstra dan Floyd-Warshall.

b. Fuzzy Logic

Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Sebagai contoh:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Pelayan restoran memberikan pelayanan terhadap tamu, kemudian tamu akan memberikan tip yang sesuai atas baik tidak nya pelayanan yang diberikan.
3. Anda mengatakan pada saya beberapa sejuk ruangan yang anda inginkan, saya akan mengatur putaran kipas yang ada pada ruangan ini.
4. Penumpang taksi berkata pada supir taksi seberapa laju kendaraan yang diinginkan, supir taksi akan mengatur pijakan gas taksinya. (Kusumadewi, Sri, 2003)

c. Neural Network

Adalah suatu model dari sistem syaraf biologis yang disederhanakan sebagai suatu alternatif sistem komputasi. (Penelitian Jaringan Syaraf Tiruan,1993). Meskipun kecepatan proses yang telah dicapai komputer dewasa ini lebih cepat jika dibandingkan dengan proses yang dilakukan otak manusia, komputer masih belum bisa menandingi universalitas kemampuan pada otak manusia.

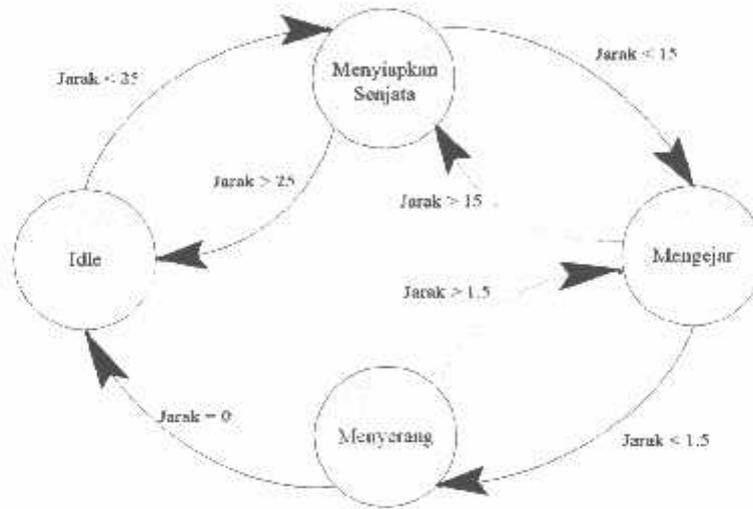
Otak manusia berisi berjuta-juta sel syaraf yang bertugas untuk memproses informasi. Tiap-tiap sel bekerja seperti suatu prosesor sederhana. Masing-masing sel tersebut saling berinteraksi sehingga mendukung kemampuan kerja otak manusia.

Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan hubungan antara neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. (Kusumadewi, Sri, 2003)

2.6 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. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks.



Gambar 2.1 FSM Survival Army Bot

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisis Kebutuhan

Kebutuhan di bagi menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Diuraikan sebagai berikut :

3.1.1 Kebutuhan Game

Dalam pembuatan *Game* ini membutuhkan perancangan konsep yang berguna untuk membantu dalam pembuatan *Game*. Tahapan ini diawali dengan mengenal dan menganalisa spesifikasi dari *Game* yang akan dibuat. Hasil analisis spesifikasi *Game* tersebut kemudian dilanjutkan dengan menentukan struktur alur dari *Game* yang akan dibuat seperti *level* dari *Game* tersebut.

Dalam tahap ini konsep rancangan *Game Survival Armybot* adalah:

- a. Menggunakan model AI FSM (*Finite State Machine*) untuk penerapan sifat musuhnya.
- b. Dalam *Game* ini *Player* mempunyai fitur untuk *Respawn* secara manual.
- c. *Player* di *Game* ini juga dapat bergerak ke kanan dan kiri dengan menekan tombol arah dan bisa melompat.
- d. *Player* juga dapat menembak kan peluru tidak terlihat dengan menekan *Mouse0* atau klik kiri.
- e. Fitur *level* dalam *Game* tersebut memiliki 4 *level* yang memiliki tingkat kesulitan yang berbeda.

3.1.2 Kebutuhan Non Fungsional

A. Analisa Kebutuhan Perangkat Keras

1. Spesifikasi minimum perangkat keras yang dibutuhkan oleh pengembang adalah:
 - a. Prosesor dengan kecepatan 1.6 Ghz.
 - b. RAM 1 GB.
 - c. Hardisk 20 GB
 - d. Monitor
 - e. VGA Card 256 MB
 - f. *Mouse* dan *Keyboard*

2. Spesifikasi minimum perangkat keras yang dibutuhkan oleh pemain adalah:
 - a. Prosesor dengan kecepatan 1.6 Ghz ke atas.
 - b. RAM 1 Gb.
 - c. Hardisk 700 Mb.
 - d. Monitor.
 - e. VGA Card 128 Mb.
 - f. *Mouse* dan *Keyboard*
 - g. *Speaker*

B. Analisa Kebutuhan Perangkat Lunak

1. Spesifikasi perangkat lunak bagi pengembang yang digunakan dalam membangun aplikasi *Game Survival Armybot* adalah:
 - a. Sistem Operasi *Windows*
 - b. Unity Engine 4.6
 - c. Autodesk Maya 2013
2. Perangkat lunak yang digunakan oleh pemain untuk menjalankan *Game Survival Armybot* adalah sistem operasi *Windows (Windows XP, Windows Vista, Windows 7, Windows 8, Linux)*.

3.2 Perancangan Game

Perancangan *Game* adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci. Pada *Game Survival Armybot*, metode FSM diterapkan dalam setiap musuh dalam *Game*. Terdapat dua susunan aksi yang akan dilakukan oleh musuh dalam *Game Survival Armybot* yaitu pemain berada dalam jarak jangkauan – melihat pemain pemain dan dalam jangkauan serang – mendekati pemain – menyerang pemain.

3.2.1 Game Play

Pada *Game Survival Armybot* ini terdapat 4 *level* yang harus dilalui oleh pemain. Apabila ingin melewati suatu *level*, Pemain diharuskan untuk menyelesaikan *Quest / Misi* yang terdapat pada setiap *level*. Sebelum melewati

level pertama pemain akan menjumpai cerita perjalanan seorang Professor yang membuat program berbahaya di dunia maya maupun dunia , hal tersebut mengundang perhatian banyak pihak. Pada *level* pertama pemain hanya melewati satu misi yang harus diselesaikan, yaitu menolong NPC yang terkepung oleh 4 musuh kerucut/ monster kerucut.

Kemudian pada *level* berikutnya yaitu *level* dua, Pemain akan dihadapi musuh yang berjumlah 30 musuh yang akan *respawn* dalam tiap detiknya, khususnya akan *respawn* dalam waktu yang tidak tetap. Jika pemain berhasil menemukan jalan keluar dan menemukan fountain sebagaimana yang diperintahkan oleh *Game* maka akan melanjutkan ke *level* berikutnya yaitu *level* tiga.

Setelah berhasil melewati *level* 2 pemain akan sampai pada *level* 3. Disini pemain berada pada suasana tengah malam. Pemain telah dibekali senter depan, kecepatan musuh bertambah dan jebakan semakin besar *Damagenya*, yaitu sekitar 50% dari jumlah *Healthy* pemain. hanya saja di mountain pemain akan dituntut untuk memecahkan teka-teki untuk melanjutkan ke *level* berikutnya

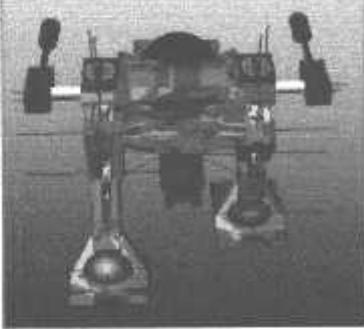
Level 4 ini adalah *level* terakhir dalam *Game* ini. pemain akan berada pada situasi yang menghendaki pemain untuk bersabar karena di *level* 4 pemain harus bertahan selama 3 menit dari kepungan musuh serta bos monster. Jika pada *level* 4 ini pemain berhasil melewatinya maka pemain akan berakhir dan kembali ke menu awal. Pada permainan *Game Survival Armybot* terdapat empat *level* yang harus dilewati pemain agar dapat menyelesaikan *Game Survival Armybot* ini. Langkah-langkah bermain pada *Game Survival Armybot* antara lain:

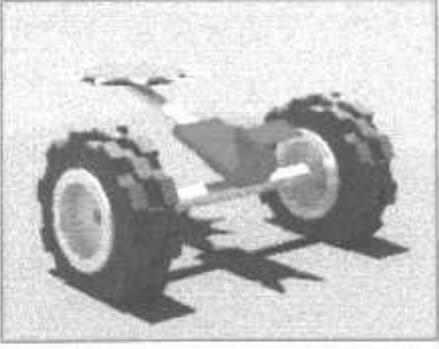
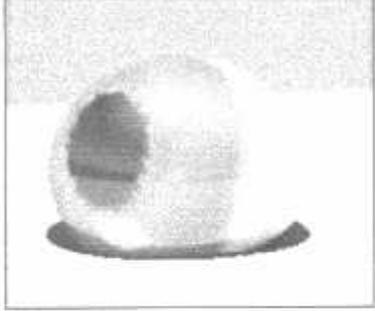
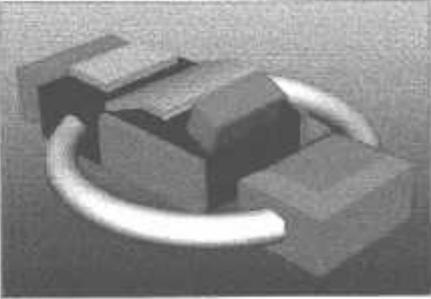
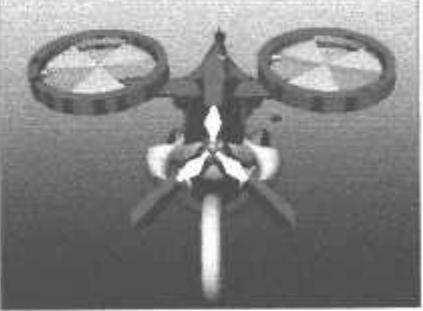
1. Pada *level* satu pemain harus mencari sebuah senjata yang berada di stage untuk menyelesaikan *level*.
 2. Dengan mengalahkan para musuh seperti hewan buas, pemain dapat menyelesaikan misi.
 3. Pemain diberikan kemampuan untuk menyerang musuh dengan berbagai cara, serangan akan mengakibatkan kematian pada musuh.
 4. Jika pemain terkena serangan musuh maka *Health* pemain akan berkurang.
-

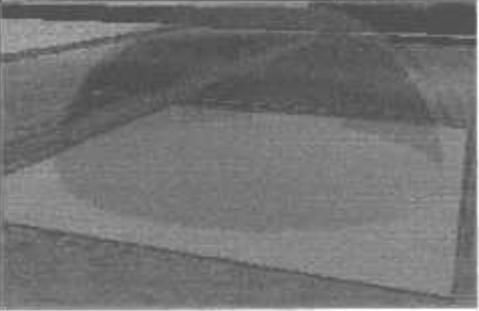
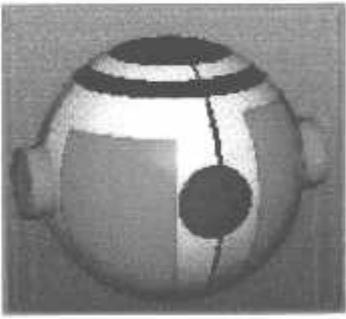
3.2.2 Karakter

Terdapat beberapa karakter dalam *Game Survival Armybot* ini. dikarenakan *Game* ini bersifat First person Controller (Sudut pandang orang pertama) maka pemain/*Player* tidak memiliki tampilan secara menyeluruh. Beberapa karakter yang akan muncul di *Game* diantaranya :

Tabel 3.1 Karakter *Game*

Karakter	Tampilan	Keterangan
<p>Pemain</p>		<p>Karakter ini akan Digunakan oleh pemain/user sebagai <i>Player</i>.</p>
<p>Enemy Clynk</p>		<p>Karakter <i>Enemy/</i> lawan dari <i>Player</i> dimana <i>enemy</i> ini akan mengejar <i>Player</i> dengan menembus berbagai tembok tanpa menciderai <i>Player</i>.</p>

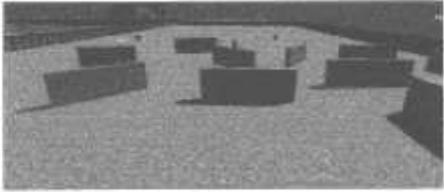
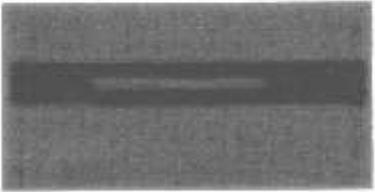
Karakter	Tampilan	Keterangan
Enemy Smoe		<p><i>Enemy</i> ini akan muncul secara acak di berbagai tempat yang ada dalam jangkauan <i>Player</i>. Mengejar <i>Player</i> dan menyerang sampai <i>Health</i> 0.</p>
Enemy OneEye		<p><i>Enemy</i> ini dapat mengejar <i>Player</i>. Jika jarak <i>Player</i> mendekati <i>enemy</i> ini</p>
Trap1		<p>Jebakan ini mengurangi <i>Health Player</i> sampai mati dan <i>Game Over</i></p>
Boss Enemy		<p><i>Enemy</i> ini dapat mengejar sesuai FSM dan menyerang.</p>

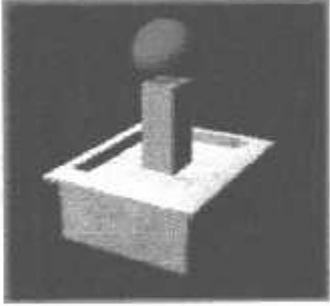
Karakter	Tampilan	Keterangan
Fountain		Fountain adalah tempat untuk menambah <i>Health Player</i> dan menyelesaikan misi <i>level 2</i>
Trap2		Jebakan yang hanya mengurangi $\frac{1}{2}$ darah <i>Player</i> dari total <i>Health Player</i>

3.2.3 Terrain

Medan atau Terrain yang ada pada Game adalah sebagai berikut :

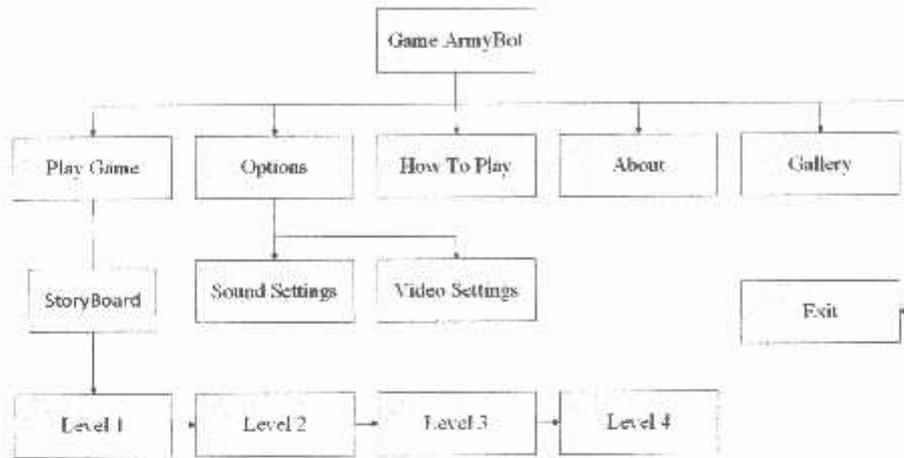
Tabel 3.2 Terrain Game

Karakter	Tampilan	Keterangan
<i>Day Ground</i>		<i>Ground</i> atau dasar lantai yang dapat dilewati oleh pemain
<i>Night Ground</i>		<i>Ground</i> yang dilewati oleh <i>Player</i> pada saat malam hari
<i>Big Lamp</i>		<i>Environment</i> ini dapat menyala di malam hari
<i>Grass Wall</i>		Dinding yang terbuat dari rumput tidak dapat ditembus oleh <i>Player</i> , dapat ditembus oleh <i>enemy Clynk</i>

Karakter	Tampilan	Keterangan
<p>Dinding Kayu</p>		<p>Dinding yang terbuat dari kayu untuk menahan <i>Player</i> agar tidak terkena serangan <i>enemy</i></p>
<p>Gunung Hiasan</p>		<p>Gunung Hiasan yang dapat menambah <i>Healthy Player</i></p>
<p>Grass Wall2</p>		<p>Dinding rumput yang terdapat pada <i>level 2</i></p>
<p>Health Lamp</p>		<p>Lampu ini terdapat di berbagai <i>level</i>, pada <i>level 3</i> lampu ini dapat menambah <i>Health Player</i></p>

3.2.4 Perancangan Struktur Menu

Perancangan struktur menu adalah perancangan tata urutan menu dari *Game Survival Armybot*. Perancangan struktur menu pada *Game Survival Armybot* dapat dilihat pada gambar 3.1



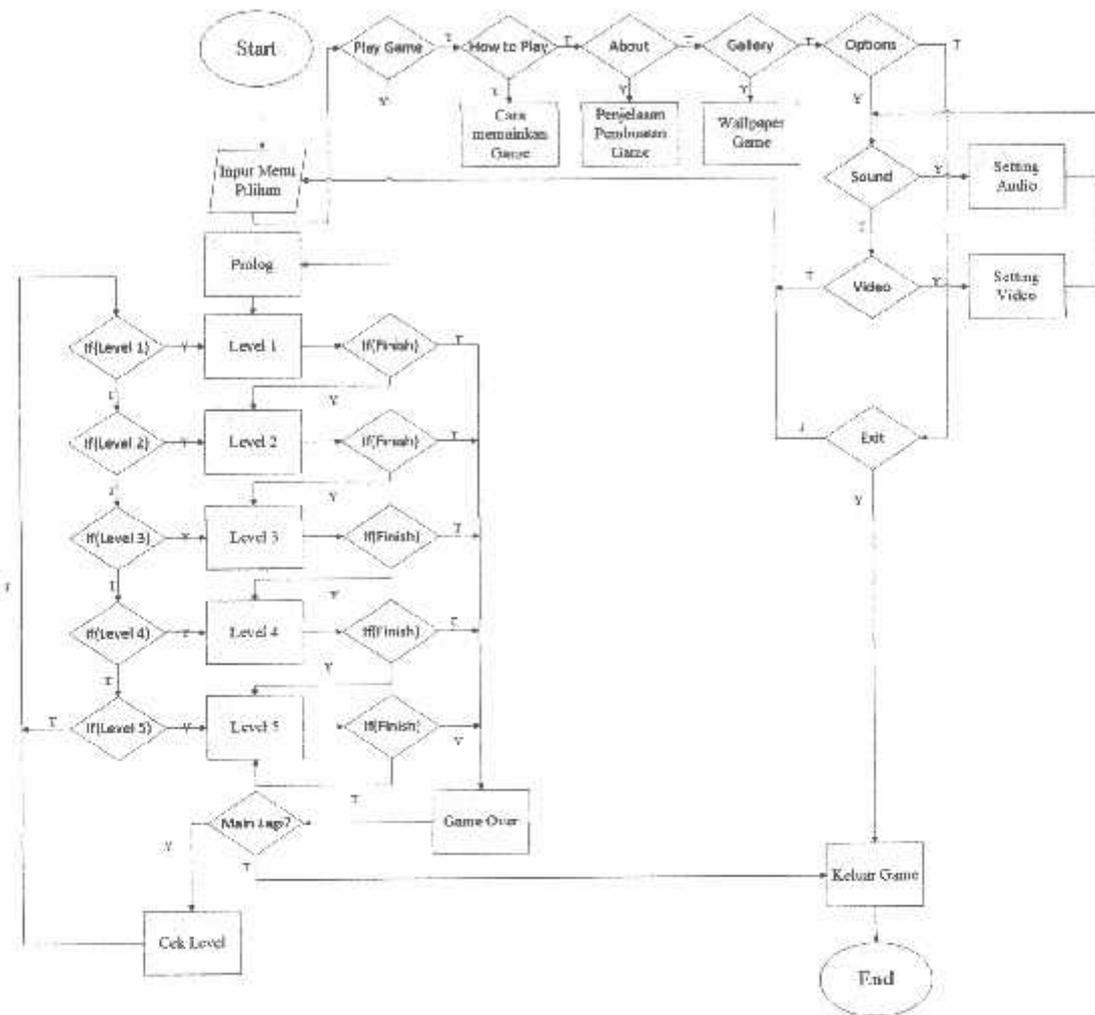
Gambar.3.1 Struktur Menu

Pada gambar 3.1, *Game* dimulai dari menu utama dimana pada tampilan menu utama terdapat beberapa pilihan menu yaitu *Play Game*, *Options*, *How to Play*, *About*, serta menu untuk keluar dari *Game*. Pilihan pertama pada menu utama adalah *Play Game*, menu ini digunakan untuk memulai permainan dari *Game Survival Armybot*, apabila pemain telah memilih menu mulai main pemain akan dihadapkan pada menu tampilan dari *level* satu yang kemudian dapat dilanjutkan ke *level* selanjutnya sampai *level* empat apabila pemain dapat menyelesaikan *level*.

Selanjutnya adalah pilihan menu kedua pada menu utama yaitu *Option*, yang memiliki fungsi untuk mengatur sound dan video. Berikutnya adalah menu *How to Play* yang berisi penjelasan tentang cara untuk memainkan *Game*. Selanjutnya adalah menu *About* yang berisi tentang *Game* yang dibuat. Dan yang terakhir menu keluar digunakan apabila pemain ingin keluar dari permainan, menu pilihan keluar akan mengeluarkan pemain dari permainan serta menutup program.

3.2.5 Alur Game

Perancangan Alur berfungsi untuk mengetahui alur proses dari program dimulai dari *start* program sampai pada *end* program. Alur dari *Game Survival Armybot* ditunjukkan dengan *Flowchart* pada gambar 3.2



Gambar.3.2 *Flowchart* Alur Game

Seperti yang terlihat pada gambar 3.2, program dimulai dari start program yang kemudian dilanjutkan pada menu utama yaitu menu tampilan pilihan untuk memulai permainan, pengaturan, petunjuk, tentang *Game*, serta terdapat opsi untuk keluar. Pada opsi pengaturan akan terdapat beberapa pengaturan sebelum permainan dimulai seperti pengaturan *audio* dan video grafik. Menu *How to Play* untuk mengetahui cara permainan dan menu *About* untuk penjelasan *Game*

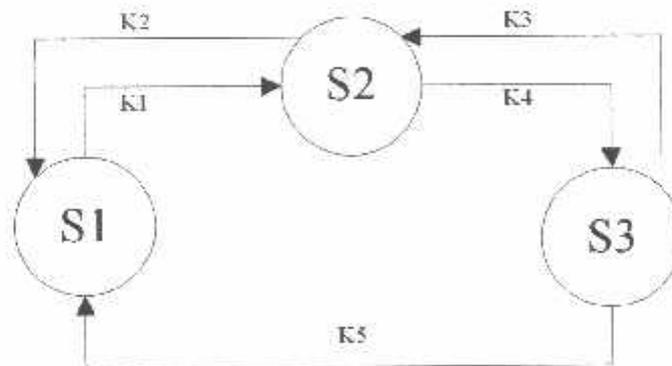
Kemudian opsi keluar digunakan apabila pemain tidak ingin untuk memulai permainan, opsi ini berfungsi untuk keluar dari *Game* dan mengakhiri program.

Apabila pemain memilih untuk mulai main, pemain akan dihadapkan pada cerita prolog yang kemudian dilanjutkan ke *level* satu sampai *level* empat, di sini pemain akan memulai permainan dan melewati rintangan pada setiap *level* selanjutnya. Akan tetapi jika pemain gagal untuk melewati rintangan atau kehabisan helath maka akan ada pilihan *respawn* (main lagi) atau main menu. Apabila pemain memilih *respawn* maka pemain akan memainkan *level* yang sama dan jika milih main menu *Game* akan berakhir atau *Game Over* dan dikembalikan ke tampilan awal.

3.2.6 Perancangan *Finite State Machine* (FSM)

Finite State Machine (FSM) terdiri dari serangkaian *State* yang menentukan pengambilan keputusan. Setiap *State* dapat berpindah ke *State* lainnya jika memenuhi kondisi yang telah ditentukan sebelumnya.

1. Perancangan AI pada monster Clynk



Gambar 3.3 Penerapan FSM pada *Monster Clynk*

State 1 : monster *Idle*.

State 2 : monster menghadapkan wajahnya pada *Player*.

State 3 : monster mengejar *Player*.

Kondisi K1 : Pemain memasuki jarak 50 – 100 *pixel*.

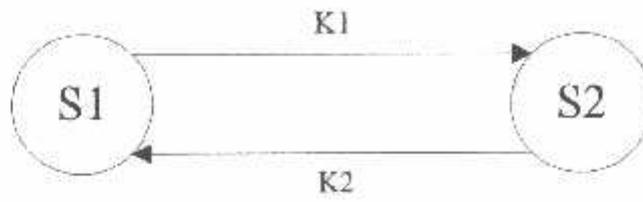
Kondisi K2 : Pemain berada di luar area monster 50 - 100 *pixel*.

Kondisi K3 : Pemain berada di luar area monster 20 – 49 *pixel*.

Kondisi K4 : Pemain memasuki area monster 20 – 49 *pixel*

Kondisi K5 : Pemain berada di luar area monster 50 – 100 *pixel*.

2. *Finite State Machine pada Trap*



Gambar 3.4 Penerapan FSM *Trap*

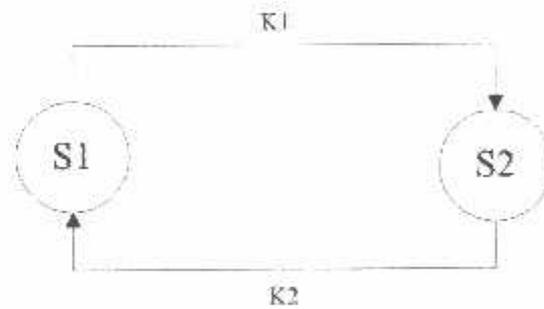
State 1 : *Trap* berada di suatu tempat

State 2 : *Trap* memberikan damage pada *Player*

Kondisi K1 : *Player* memasuki daerah *Trap*

Kondisi K2 : *Player* berada di luar daerah *Trap*

3. *Finite State Machine pada fountain*



Gambar 3.5 Penerapan FSM *Fountain*

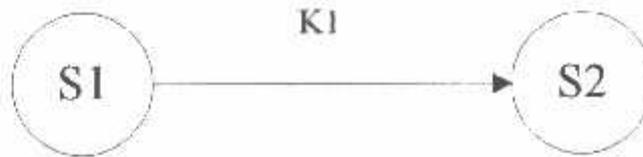
State 1 : *Fountain Idle*

State 2 : *Fountain* menambah health *Player*

Kondisi K1 : *Player* memasuki daerah *Fountain*

Kondisi K2 : *Player* berada di luar daerah *Fountain*

4. *Finite State Machine* pada *Enemy Smoe*



Gambar 3.6 Penerapan FSM *Enemy Smoe*

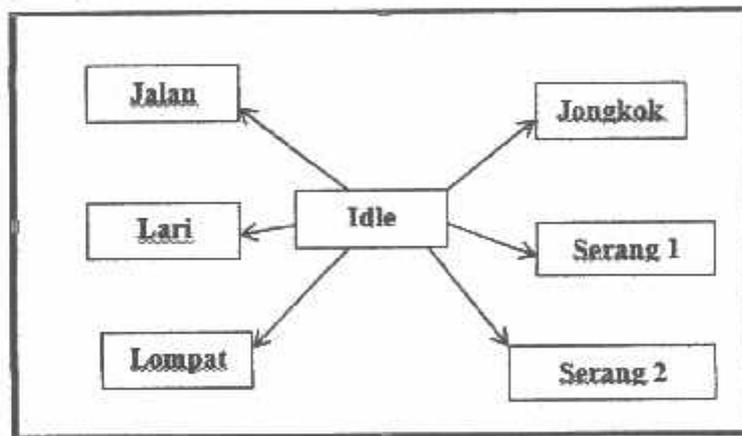
State 1 : *Enemy Smoe* sebelum Respawn

State 2 : *Enemy Smoe* Mengejar *Player*

Kondisi K1 : *Enemy Smoe* Spuwn

3.2.7 Perancangan Aksi Karakter

Perancangan Aksi karakter adalah perancangan reaksi tindakan karakter terhadap suatu kondisi tertentu, karakter pemain memiliki beberapa aksi yaitu Idle, berjalan, berlari, merunduk, melompat, dan menyerang. Perancangan aksi karakter ditunjukkan pada gambar 3.7



Gambar 3.7 Perancangan Aksi Karakter pada *Player*

3.2.8 Perancangan Antar Muka

Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi *Game* yang akan dibangun.

1. Perancangan Antar Muka Menu Utama

Perancangan antar muka menu utama adalah perancangan tampilan menu utama dari *Game Survival Armybot* dimana pada menu utama terdapat tombol

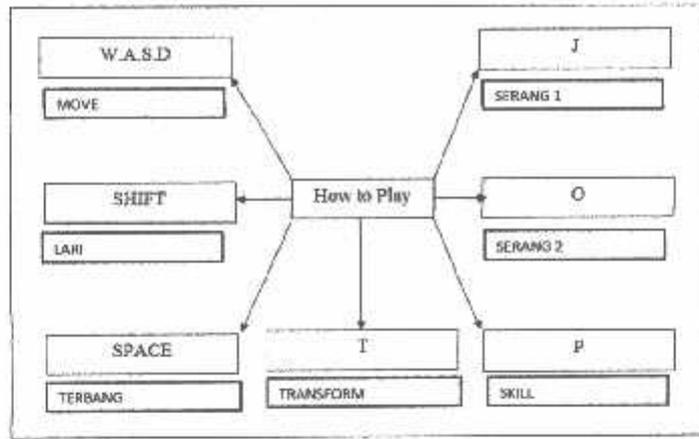
Play Game, *How to Play*, *Options*, *About* serta tombol *Exit*. Tombol *Play Game* berfungsi untuk memulai permainan, jika pemain memilih tombol mulai main maka pemain akan dihadapkan pada antar muka dari *level 1*. Tombol *How to Play* digunakan apabila pemain tidak mengetahui cara bermain dari *Game Survival Armybot*, tombol *About* akan mengarahkan pemain pada antar muka keterangan pembuat *Game*. Kemudian tombol *Options* digunakan apabila ingin mengganti setting awal dari *Game* sebelum memulai permainan dan selanjutnya adalah tombol *Exit* digunakan untuk keluar dari permainan. Rancangan antar muka menu utama dari *Game Survival Armybot* ditunjukkan dalam gambar 3.8



Gambar 3.8 Perancangan Antar Muka Menu Utama.

2. Perancangan Antar Muka *How to Play*

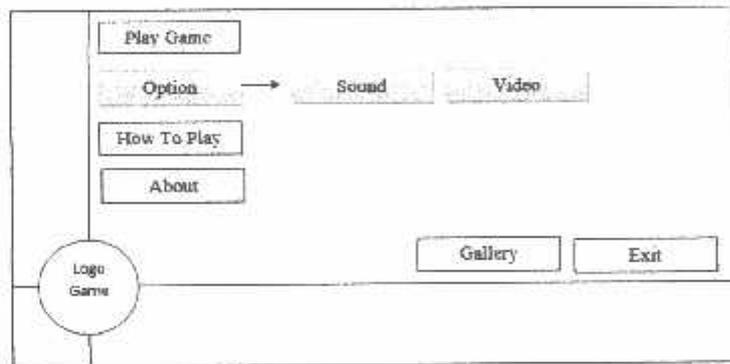
Perancangan antar muka *How to Play* adalah perancangan tampilan menu petunjuk dari *Game Survival Armybot* dimana pada menu *How to Play* terdapat petunjuk serta cara untuk memainkan *Game*. Di antaranya adalah tombol W,A,S,D atau Arah pada *Keyboard* untuk melakukan aksi bergcrak sesuai arah yang diinginkan, tombol space untuk melakukan aksi melompat, tombol C pada *Keyboard* untuk melakukan gerakan jongkok, tombol Shift pada *Keyboard* untuk berlari, tombol I pada *Keyboard* untuk mengecek Inventory / tas *Player* dan juga terdapat gambar *Mouse* dimana klik kanan untuk menyerang. Rancangan antar muka petunjuk dari *Game Survival Armybot* ditunjukkan dalam gambar 3.9.



Gambar 3.9 Perancangan Antar Muka Menu *How to Play*

3. Perancangan Antar Muka *Options*

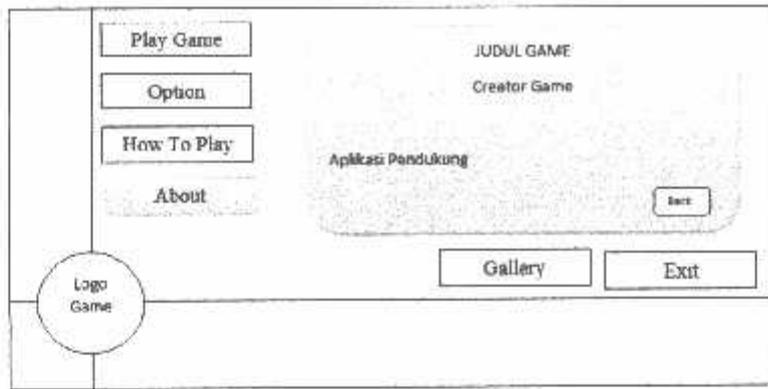
Perancangan Antar muka *Options* adalah perancangan tampilan menu pengaturan dari *Game Survival Armybot* dimana pada menu pengaturan terdapat pilihan pengaturan sebelum memainkan *Game*. Pilihan menu pada pengaturan yaitu pengaturan Sound dan Video di dalam *Game*.



Gambar 3.10 Perancangan Antar Muka *Options*

4. Perancangan Antar Muka *About*

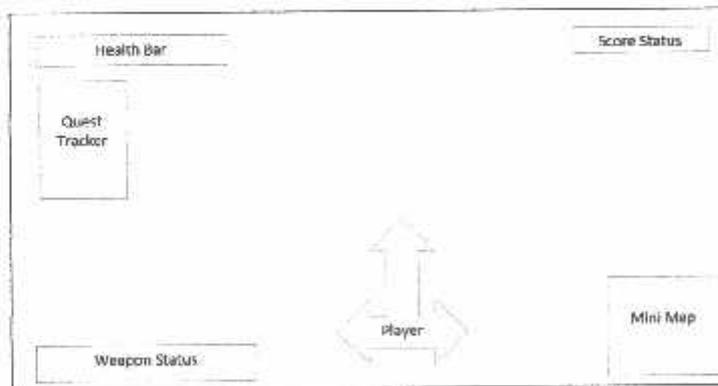
Perancangan Antar muka *About* adalah perancangan tampilan menu tentang *Game Survival Armybot* dimana pada antar muka ini berisi informasi seperti pembuat dari *Game* dan software yang digunakan.



Gambar 3.11 Perancangan Antar Muka *About*

5. Perancangan Antar Muka *Level 1*

Perancangan antar muka *level* satu adalah perancangan tampilan pada *level* satu dari *Game Survival Armybot* dimana pada *level 1* pemain akan memulai permainan pada environment *level 1*. Dalam antar muka *level 1* akan terdapat *Health Status* yang menunjukkan status darah atau kehidupan dari karakter *Player*. Pada gambar 3.12 terdapat *Health Status*, kemudian *Quest* sebagai petunjuk *Player* untuk melakukan misi yang diperintahkan, Pada *level* satu karakter dapat digerakkan oleh pemain untuk berjalan, berlari, melompat, merunduk dengan menekan tombol pada *Keyboard* setra memukul dengan meng-*click* kanan *Mouse* seperti yang ditunjukkan dalam antar muka *How to Play*. Rancangan antar muka *level 1* dari *Game Survival Armybot* ditunjukkan dalam gambar 3.12.

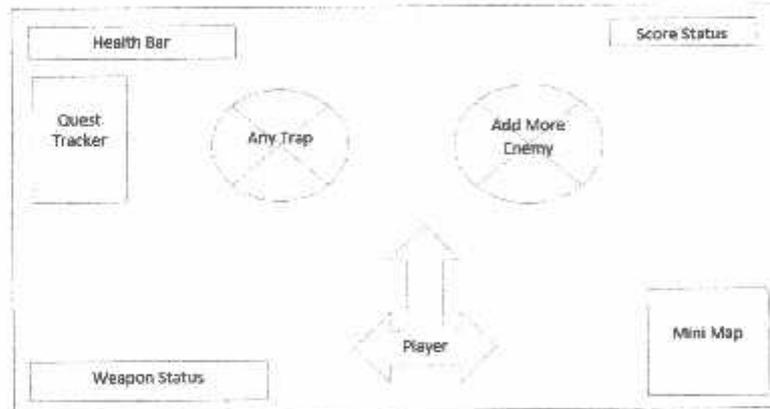


Gambar 3.12 Rancang Antar Muka *Level*

6. Perancangan Antar Muka *Level 2*

Perancangan antar muka *level* dua adalah perancangan tampilan pada *level* dua dari *Game Survival Armybot* dimana pada *level 2* pemain akan memulai

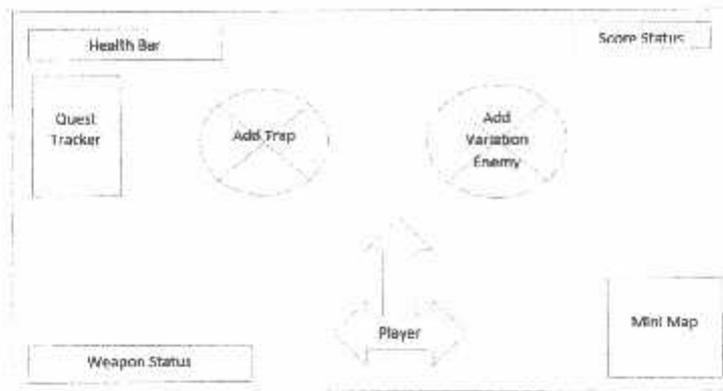
permainan pada environment *level 2*. Dalam antar muka *level 2* akan terdapat *Health Status* yang menunjukkan status darah atau kehidupan dari karakter *Player*. Pada gambar 3.13 terdapat *Health Status*, kemudian *Quest* sebagai petunjuk *Player* untuk melakukan misi yang diperintahkan. Pada *level dua* karakter dapat digerakkan oleh pemain untuk berjalan, berlari, melompat, merunduk dengan menekan tombol pada *Keyboard* setra memukul dengan meng-*click* kanan *Mouse* seperti yang ditunjukkan dalam antar muka *How to Play*. Rancangan antar muka *level 2* dari *Game Survival Armybot* ditunjukkan dalam gambar 3.13.



Gambar 3.13 Rancang Antar Muka *Level 2*

7. Perancangan Antar Muka *Level 3*

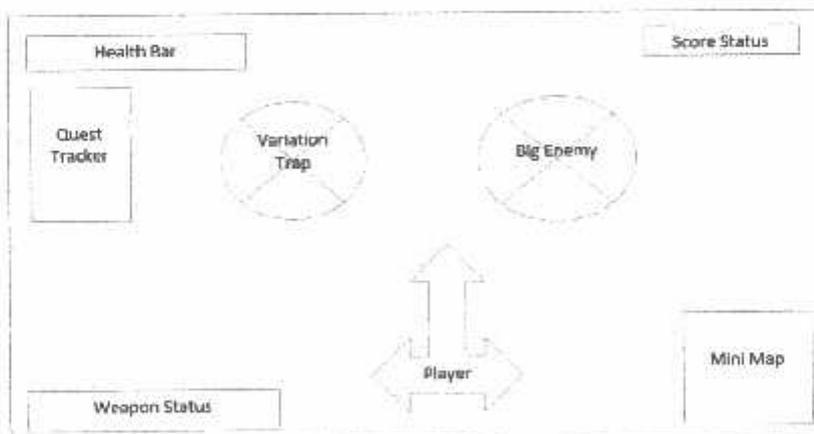
Perancangan antar muka *level tiga* adalah perancangan tampilan pada *level tiga* dari *Game Survival Armybot* dimana pada *level 3* pemain akan memulai permainan pada environment *level 3*. Dalam antar muka *level 3* akan terdapat *Health Status* yang menunjukkan status darah atau kehidupan dari karakter *Player*. Pada gambar 3.14 terdapat *Health Status*, kemudian *Quest* sebagai petunjuk *Player* untuk melakukan misi yang diperintahkan, Pada *level tiga* karakter dapat digerakkan oleh pemain untuk berjalan, berlari, melompat, merunduk dengan menekan tombol pada *Keyboard* setra memukul dengan meng-*click* kanan *Mouse* seperti yang ditunjukkan dalam antar muka *How to Play*. Rancangan antar muka *level 3* dari *Game Survival Armybot* ditunjukkan dalam gambar 3.14.



Gambar 3.14 Rancang Antar Muka *Level 3*

8. Perancangan Antar Muka *Level 4*

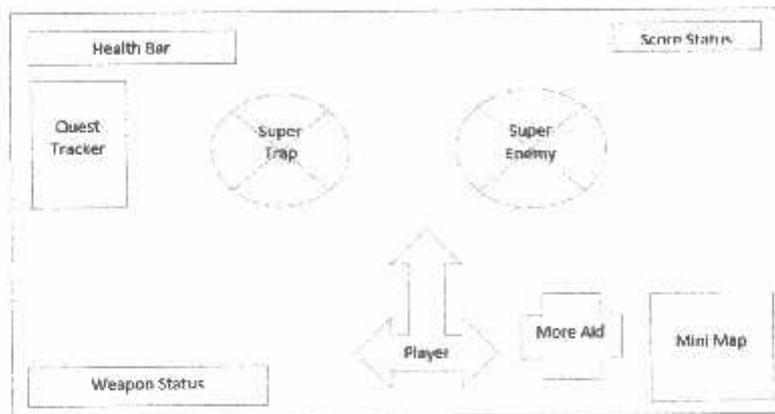
Perancangan antar muka *level* empat adalah perancangan tampilan pada *level* empat dari *Game Survival Armybot* dimana pada *level 4* pemain akan memulai permainan pada environment *level 4*. Dalam antar muka *level 4* akan terdapat *Health Status* yang menunjukkan status darah atau kehidupan dari karakter *Player*. Pada gambar 3.15 terdapat *Health Status*, kemudian *Quest* sebagai petunjuk *Player* untuk melakukan misi yang diperintahkan, Pada *level* empat karakter dapat digerakkan oleh pemain untuk berjalan, berlari, melompat, merunduk dengan menekan tombol pada *Keyboard* setra memukul dengan meng-*click* kanan *Mouse* seperti yang ditunjukkan dalam antar muka *How to Play*. Rancangan antar muka *level 4* dari *Game Survival Armybot* ditunjukkan dalam gambar 3.15



Gambar 3.15 Rancang Antar Muka *Level 4*

9. Perancangan Antar Muka *Respawn*

Perancangan Antar muka *Respawn* adalah perancangan tampilan saat kondisi *Player* mati, yang kemudian muncul menu untuk melanjutkan permainan atau tidak. Jika pemain memilih *Respawn* maka karakter *Player* akan dihidupkan kembali ketempat semula, dan apabila tidak maka pemain akan dilemparkan ke menu utama. Rancangan antar muka *respawn* dari *Game Survival Armybot* ditunjukkan pada gambar 3.16.



Gambar 3.16 Rancangan Antar Muka *Respawn*

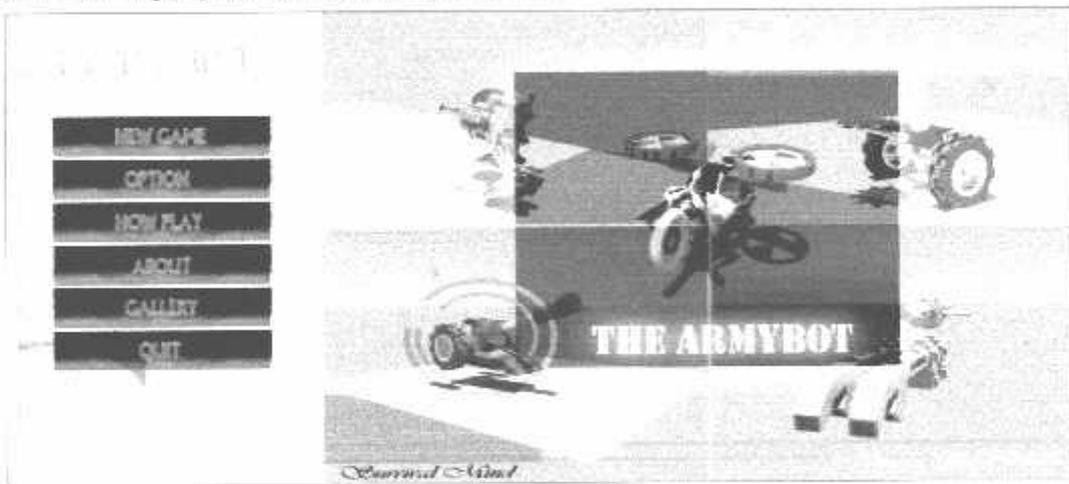
BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Hasil

Berikut tahapan hasil implementasi *Game "Survival Armybot"* menggunakan *Unity Engine*. Tampilan yang terdapat pada *Game* yaitu tampilan *menu* utama, tentang, level 1, level 2, level 3, dan level4 (Boss). Tampilan-tampilan tersebut adalah sebagai berikut :

4.1.1 Tampilan Menu Utama

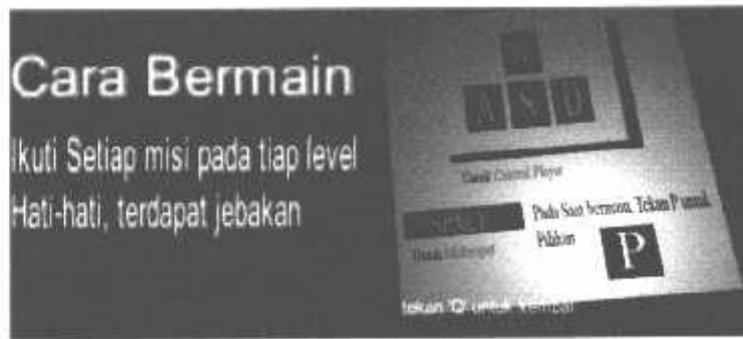
Tampilan *menu* utama adalah tampilan yang muncul pada saat membuka *Game "Survival Armybot"*. Pada saat tampilan terdapat pilihan Mulai, Panduan, dan Tentang saya. Seperti pada gambar 4.1



Gambar 4.1 Tampilan menu utama

4.1.2 Tampilan How Play

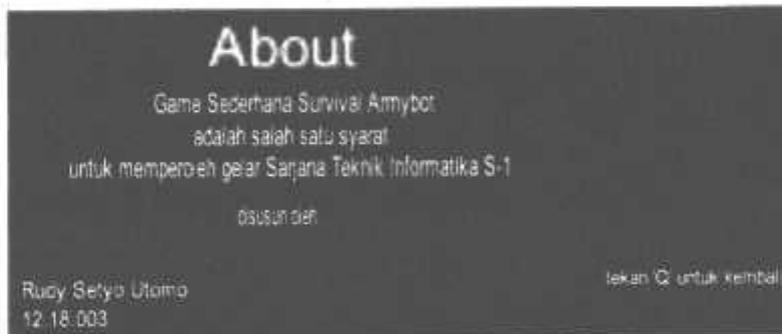
Tampilan *How Play* adalah tampilan yang berisikan tentang informasi tertulis dari *Game "Survival Armybot"*. Informasi tersebut berisi tentang *control Player* yang di gunakan untuk bermain *Game "Survival Armybot"*.



Gambar 4.2 Tampilan menu How Play

4.1.3 Tampilan *About*

Tampilan *About* adalah tampilan yang berisikan tentang informasi tertulis dari pembuat *Game "Survival Armybot"*.



Gambar 4.3 Tampilan menu About

4.1.4 Tampilan *Gallery*

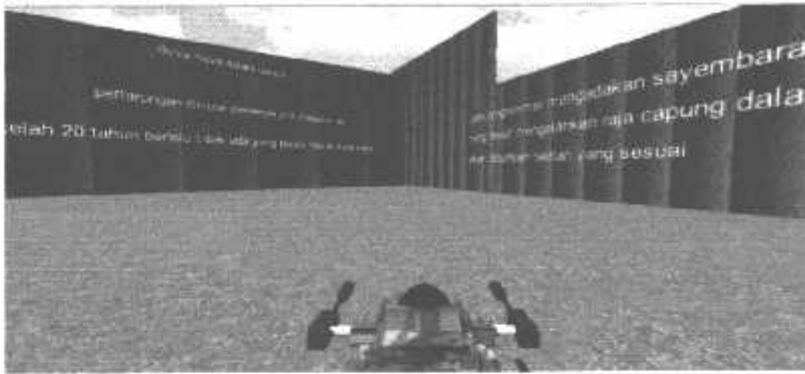
Tampilan *Gallery* adalah tampilan yang berisikan tentang beberapa gambar karakter dalam *Game Survival Armybot*.



Gambar 4.4 Tampilan menu Gallery

4.1.5 Tampilan *Storyboard*

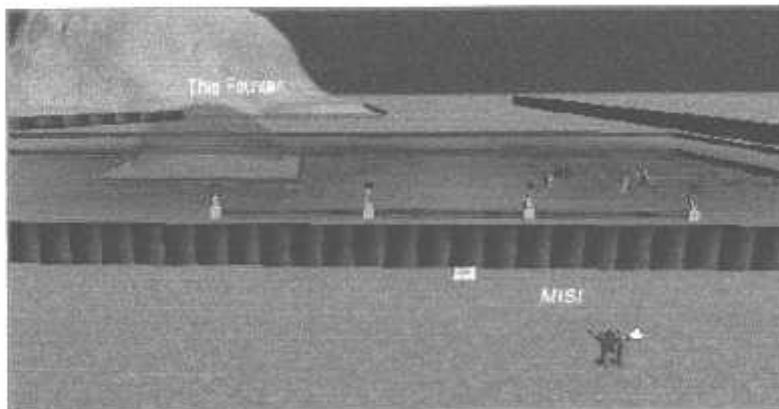
Tampilan *Storyboard* adalah tampilan yang berisikan jalan nya cerita yang berkaitan dengan *Game Survival Armybot*.



Gambar 4.5 Tampilan *Storyboard*

4.1.6 Tampilan *Level 1*

Pada level 1 ini pemain diberi *quest* atau misi untuk menyelamatkan NPC dari kepungan musuh. Terdapat banyak jebakan, di depan *Player* juga telah disediakan text misi yang berisi perintah, terdapat 4 *Enemy* kerucut yang siap menghadang *Player*, namun enemy kerucut tidak memberikan damage, *enemy* kerucut hanya untuk mengelabui *Player* dan keberadaannya membantu *enemy* Smoe untuk menyerang *Player*. Ketika *Player* menyentuh NPC maka *Game* akan berganti scene menuju level 2. Pada saat berada di zona NPC suara win akan terdengar.



Gambar 4.6 Tampilan *Level 1*

4.1.7 Tampilan Level 2

Pada level 2 ini, *Player* dituntut untuk lebih cepat menuju fountain karena musuh akan muncul lebih banyak dari level 1. Jarak menang level 2 lebih jauh dari pada level 1. *Player* pada level ini bertujuan agar *Player* mengerti tentang fungsi fountain pada *Game*, yaitu sebagai penambah health *Player*.



Gambar 4.7 Tampilan Level 2

4.1.8 Tampilan Level 3

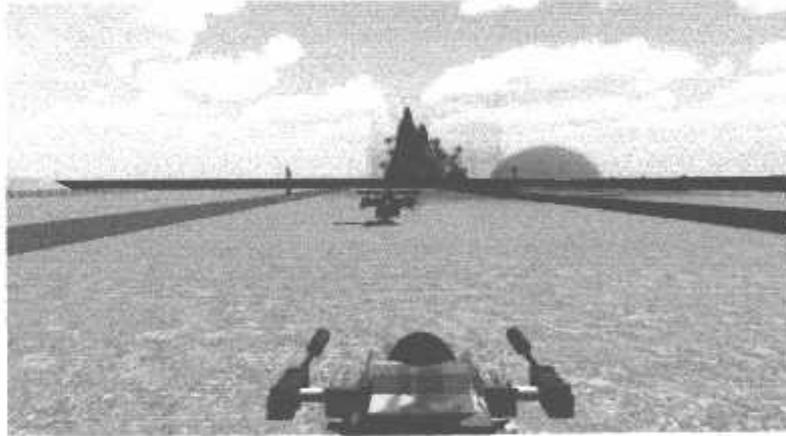
Suasana permainan pada level 3 berbeda dari level sebelumnya, pada level 3 permainan di set pada malam hari, *Player* dituntut untuk lebih cermat dalam menyelesaikan permainan, karena pada malam hari memberikan tantangan tersendiri bagi *Player*. *Trap* atau jebakan berada di dalam rumput dan di balik tembok. Misi pada level ini adalah menuju zona gunung seperti yang telah ditunjukkan dalam *Game*.



Gambar 4.8 Tampilan Level 3

4.1.9 Tampilan Level 4

Level 4 adalah level terakhir pada *Game Survival Armybot*. Pada saat *Player* memulai level 4, boss musuh telah berada di depannya, damage boss musuh lebih besar dari pada musuh-musuh yang ada. *Player* harus bertahan selama 3 menit dalam *Game* untuk menang.



Gambar 4.9 Tampilan Level 4

4.1.10 Tampilan Game Over

Tampilan ini akan muncul apabila *Health Player* mencapai 0. *Health Player* berkurang jika terkena damage musuh, terkena jebakan ataupun karena kesalahan *Player* itu sendiri. *Player* akan selalu *Spawn* atau muncul di level 1 jika *Game over* di *scene* lain atau *scene* 1, untuk menghindari spawn *down level* diharapkan *Player* dapat menekan R sebelum *Health* 0.



Gambar 4.10 Tampilan Game Over

4.2 Pengujian Fungsional

Pengujian fungsional adalah pengujian mengenai proses fungsional yang ada dalam *Game*. Hasil dari pengujian dapat dilihat pada tabel 4.1.

4.1 Tabel pengujian fungsional

No.	Fungsi	Hasil
1.	Karakter musuh dapat mengejar karakter <i>Player</i> .	[√] Sesuai
2.	Karakter musuh dapat menycrang pemain ketika berada pada jarak serang.	[√] Sesuai
3.	Tombol pada menu utama berjalan sesuai dengan fungsi.	[√] Sesuai
4.	Pemberian <i>health</i> pada karakter berjalan normal dengan indikasi karakter pamain mati ketika <i>health</i> poin kurang dari nol.	[√] Sesuai
5.	AI <i>Finite State Machine</i> pada musuh	[√] Sesuai
6.	Karakter Musuh Dapat Melihat Karakter <i>Player</i>	[√] Sesuai
7.	Pilihan Mulai pada menu utama dapat berjalan	[√] Sesuai
8.	Karakter Boss <i>Enemy Capes</i> dapat menyerang Karakter <i>Player</i>	[√] Sesuai
9.	Fountain dapat mengisi <i>Health Player</i> pemain	[√] Sesuai
10.	Trap dapat mengurangi <i>Health Poin</i> Pemain	[√] Sesuai

Pada masing-masing fungsi berjalan sesuai dengan output, tidak ada error ataupun salah fungsi.

4.2.1 Pengujian AI (Artificial Intelligence)

Pengujian AI (*Artificial Intelligence*) adalah pengujian mengenai fungsi yang berkaitan dengan AI (*Artificial Intelligence*) yang ada dalam *Game*. Hasil dari pengujian AI *Finite State Machine* dapat dilihat pada tabel 4.2.

4.2 Tabel Pengujian AI

No.	Fungsi	Output	Hasil
1	AI (<i>Finite State Machine</i>) pada Monster Clynk	Pada saat <i>Player</i> mendekati musuh dalam jarak 50-100, musuh menghadap <i>Player</i> dan mengikuti arah <i>Player</i> sampai mendekati jarak 49 <i>pixel</i> , kemudian musuh mengejar	Sesuai
2	AI (<i>Finite State Machine</i>) pada <i>Trap</i>	Pada saat <i>Player</i> menyentuh <i>Trap</i> atau jebakan, health <i>Player</i> berkurang sesuai dengan jumlah damage pada <i>trap</i>	Sesuai
3	AI (<i>Finite State Machine</i>) pada <i>Fountain</i>	Pada saat <i>Player</i> memasuki zona <i>Fountain</i> akan mendapatkan tambahan <i>Health</i>	Sesuai
4	AI (<i>Finite State Machine</i>) pada <i>Enemy Smoe</i>	Musuh <i>Smoe</i> mengejar <i>Player</i> pada saat muncul atau <i>Spawn</i>	Sesuai

Pada masing-masing fungsi pada AI berjalan sesuai dengan output, tidak ada error ataupun salah fungsi

4.2.2 Pengujian *Control Player*

Pengujian *control Player* adalah pengujian fungsi dari setiap tombol yang diterapkan untuk menggerakkan karakter tabel 4.3.

4.3 Tabel Pengujian *Control Player*

No.	Tombol	Fungsi	Output
1	A	Bergerak ke kiri	Pemain bergerak kekiri(Sesuai)
2	D	Bergerak ke kanan	Pemain bergerak kekanan(Sesuai)
3	W	Bergerak Maju	Pemain bergerak maju (Sesuai)
4	S	Bergerak Mundur	Pemain bergerak mundur (Sesuai)
5	Space	Melompat	Pemain melompat(Sesuai)
6	Mouse 0 / Click Kiri	Menembak	Pemain Menembak (Sesuai)
7	P	<i>Temporary Menu</i>	Tampil <i>Temporary Menu</i> (Sesuai)
8	R	<i>Respawn Player</i>	Pemain Berganti Tempat ke posisi semula dalam level (sesuai)

Tabel diatas menunjukkan bahwa 8 dari 8 tombol pada *control Player* berjalan sesuai dengan fungsi yang diinginkan.

4.2.3 Pengujian Performa

Pengujian performa adalah pengujian yang dilakukan pada kinerja atau respon perangkat keras. Menguji *Game* pada komputer yang berbeda. tabel 4.4.

4.4 Tabel Pengujian Performa

No	Processor	RAM	VGA	SO	Keterangan
1.	Intel core i5	4 GB	3 GB	Win 8	Berjalan lancar
2.	Intel core i3	2 GB	4 GB	Win 8	Berjalan lancar
3.	Core 2 Duo	1GB	512 MB	Win XP	Berjalan lancar
4.	Amd A8	4GB	3 GB	Win 7	Berjalan lancar
5.	Celeron	4GB	2 GB	Win 8.1	Berjalan lancar
6.	Intel core i3	4 GB	3 GB	Win8	Berjalan lancar
7.	Intel Core i3	2 GB	2 GB	Ubuntu 14.0	Berjalan lancar

Dapat disimpulkan bahwa *Game "Survival Armybot"* dapat dimainkan dengan RAM minimal 1 GB.dan dapat dijalankan di OS (*Operating System*) *Windows 8, Windows 8.1, Windows XP, Linux, Windows 7.*

4.2.4 Pengujian pada Pengguna

Pengujian pada pengguna ditujukan kepada *user* yang memainkan *Game "Survival Armybot"*. terdapat kriteria dan penilaian pada pengujian terdapat pada tabel 4.5.

4.5 Tabel Pengujian pada Pengguna

No	Kriteria	Penilaian		
		A	B	C
1	<i>Gameplay</i>	15	3	2
2	Desain karakter musuh	16	3	1
3	Desain karakter player	14	4	2
4	<i>Genre Survival</i>	10	7	3
5	AI (<i>Artificial Intelligence</i>) musuh	7	8	5
6	<i>Control pada player</i>	7	7	6
7	Desain <i>Environment</i>	10	6	4
8	Tampilan Menu Utama	6	8	6

Ket: A : Menarik
 B : Cukup Menarik
 C : Kurang Menarik

Hasil pengujian user dari 20 responden berdasarkan kuisioner di atas *gameplay* dari *game* ini sebesar 15 responden menyatakan menarik, desain karakter musuh pada *game Survival Armybot* ini sebesar 16 responden menyatakan menarik, Artificial Intelligence atau AI pada musuh dalam *game* ini 7 responden menyatakan menarik.

Hasil persentasi pengujian user dari 20 responden 53,12% menyatakan menarik, 28,75% menyatakan cukup menarik, dan 18,12% menyatakan kurang menarik. Hasil tersebut diperoleh:

Menyatakan Menarik = $85/160 \times 100\% = 53,12\%$

Menyatakan Cukup Menarik = $46/160 \times 100\% = 28,75\%$

Menyatakan Kurang Menarik = $29/160 \times 100\% = 18,12\%$

Nilai 160 diperoleh dari jumlah semua penilaian dari A, B, C. Sedangkan 85 diperoleh dari jumlah responden yang memberikan nilai A. Hasil 46 diperoleh dari jumlah responden yang memberikan nilai B. 29 diperoleh dari jumlah responden yang memberikan nilai C.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari perancangan dan implementasi pada *Game Survival Armybot* dengan menggunakan metode Finite State Machine (FSM) dapat diambil beberapa kesimpulan yaitu :

1. Dari hasil pengujian fungsional *Game "Survival Armybot"*, semua output dinyatakan sesuai.
2. Dari hasil pengujian AI output dari metode yang digunakan sudah sesuai.
3. Dari hasil hasil control gerak dari player berdasarkan fungsi sesuai dengan hasil
4. Dari hasil pengujian control gerak menunjukan bahwa 8 dari 8 tombol pada control player berjalan sesuai dengan fungsi yang diinginkan

Dari hasil pengujian fungsional, pengujian AI(Artificial Intelligence), dan control berjalan sesuai. Dari pengujian performa game dapat dijalankan di komputer dengan spesifikasi RAM 1-4GB, VGA minimal 512MB, dengan OS Windows XP, 7, dan 8.1. Dari hasil pengujian pada pengguna dapat disimpulkan bahwa Hasil pengujian user dari 20 responden gameplay dari game ini sebesar 15 responden menyatakan menarik, desain karakter musuh sebesar 16 responden menyatakan menarik,. persentasi pengujian user dari 20 responden 53,12% menyatakan menarik, 28,75% menyatakan cukup menarik, dan 18,12% menyatakan kurang menarik.

5.2 Saran

Dalam pembuatan *game "Survival Armybot"* ini masih memiliki kekurangan-kekurangan. berikut adalah saran yang dikemukakan oleh penulis :

1. *Game Survival Armybot* ditambahkan menu untuk *multiplayer*
2. Game dapat dimainkan di OS Android

DAFTAR PUSTAKA

- Hermawan, Benny. 2004. *Menguasai Java 2 & Object Oriented Programming*. Yogyakarta: Penerbit Andi.
- Komputer, Wahana 2014. *Mudah Membuat game 3 Dimensi Unity 3d*. Yogyakarta: Penerbit Andi.
- Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- Mudah Membuat Game 3 Dimensi Unity 3D (Seno, Ed). 2014. Semarang : Wahana Komputer.
- Nugroho, widhi 2013. *Implementasi Finite State Machine Pada Game Malam Jum'at Kliwon*.
- Pedoman Penulisan Karya Ilmiah Edisi Kelima. 2010. Malang: Penerbit Universitas Negeri Malang.
- Roedavan, Rickman. 2014. *Unity Tutorial Game Engine*. Bandung: Penerbit Informatika.

LAMPIRAN

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Rudy Setyo Utomo
NIM : 1218003
JURUSAN : Teknik Informatika S-1
JUDUL : Rancang Bangun Game Survival Armybot Menggunakan Metode
Finite State Machine(FSM)

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

Hari : Jum'at
Tanggal : 15 Januari 2016
Nilai : 79,25 (B+)

Panitia Ujian Skripsi :

Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT
NIP. 197404162005011002

Anggota Penguji :

Dosen Penguji I



A. Fahrudi S., S.Kom.MT
NIP.P 1031500497

Dosen Penguji II



Moh. Mift. Rokhman, S.Kom.M.Kom
NIP.P. 1031500479

FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Rudy Setyo Utomo
NIM : 1218003
JURUSAN : Teknik Informatika S-1
JUDUL : Rancang Bangun Game Survival Armybot Menggunakan Metode Finite State Machine(FSM)

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	15 Januari 2016	1. Level sampai 5, Trap Effect, Option Menu 2. Lv 1 sentuh, Lv 2 Mindahin ke tempat lain, Lv 3 Perang, Lv 5 senjata Xray	
2.	Penguji II	15 Januari 2016	1. Perbaiki Menu + Option 2. Tambahkan aksi tembak menembak 3. Perjelas Peran Setiap NPC	

Dosen Penguji I



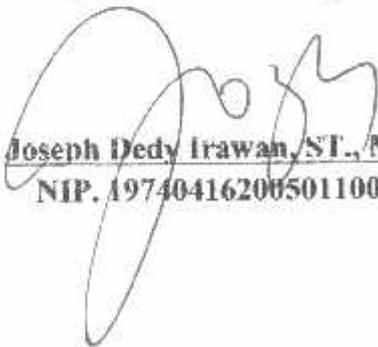
A. Fahrudi S., S.Kom.MT
NIP.P 1031500497

Dosen Penguji II



Moh. Mift. Rokhman, S.Kom.M.Kom
NIP.P 1031500479

Dosen Pembimbing I



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

Dosen Pembimbing II



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



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

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

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 :

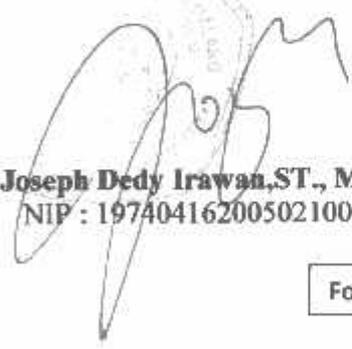
Nama : RUDY SETYO UTOMO
Nim : 1218003
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

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
Ketua,


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

Form S-4a



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

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

*T. 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/LINF/TA/2015
Lampiran : ---
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Febriana Santi W, S.Kom, M.Kom
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 : RUDY SETYO UTOMO
Nim : 1218003
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
Ketua,

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

Form S-4a



INSTITUT TEKNOLOGI NASIONAL
Fakultas Teknologi Industri
Program Studi Teknik Informatika S-1

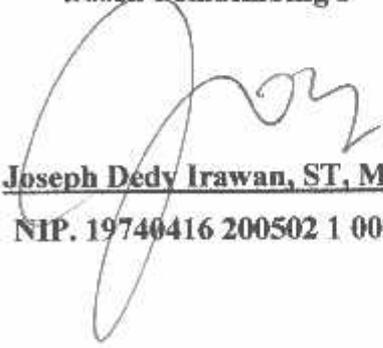
FORMULIR BIMBINGAN SKRIPSI

Nama : Rudy Setyo Utomo
NIM : 1218003
Masa Bimbingan : 23 Oktober 2015 sampai 23 Maret 2016
Judul Skripsi : RANCANG BANGUN GAME SURVIVAL
ARMYBOT MENGGUNAKAN METODE FINITE
STATE MACHINE (FSM)

No	Tanggal	Uraian	Paraf
1	27/10/15	Konsultasi Game Armybot	
2	28/10/15	Desain Sistem Game	
3	05/11/15	Cek BAB II + III	
4	21/11/15	Konsultasi Seminar Progress	
5	03/12/15	Perbaiki Game	
6	06/12/15	Konsultasi BAB IV + BAB V	
7	05/01/15	Konsultasi Laporan + Program	
8	12/01/15	Konsultasi BAB V	
9	13/01/15	ACC Kompre	

Malang, 2016

Dosen Pembimbing I


Joseph Dedy Irawan, ST, MT

NIP. 19740416 200502 1 002



INSTITUT TEKNOLOGI NASIONAL
Fakultas Teknologi Industri
Program Studi Teknik Informatika S-1

FORMULIR BIMBINGAN SKRIPSI

Nama : Rudy Setyo Utomo
NIM : 1218003
Masa Bimbingan : 23 Oktober 2015 sampai 23 Maret 2016
Judul Skripsi : RANCANG BANGUN GAME SURVIVAL
ARMYBOT MENGGUNAKAN METODE FINITE
STATE MACHINE (FSM)

No	Tanggal	Uraian	Paraf
1	27/10/15	Konsultasi Game Armybot	<i>hm</i>
2	05/11/15	Konsultasi BAB III	<i>hm</i>
3	21/11/15	ACC BAB III	<i>hm</i>
4	24/11/15	ACC Seminar Progress	<i>hm</i>
5	04/12/15	Demo Program	<i>hm</i>
6	10/12/15	Perbaiki BAB IV	<i>hm</i>
7	11/01/15	Perbaiki Laporan Semhas	<i>hm</i>
8	13/01/15	Konsultasi BAB IV, BAB V	<i>hm</i>
9	14/01/15	ACC Kompre	<i>hm</i>

Malang, 2016

Dosen Pembimbing II

Febriana Santi W, S.Kom, M.Kom

NIP.P. 1031000425

KUISIONER

Nama : _____

1. Menurut anda bagaimana gameplay dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 2. Menurut anda bagaimana desain karakter musuh dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 3. Menurut anda bagaimana desain karakter player dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 4. Bagaimana menurut anda genre survival yang menjadi genre dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 5. Bagaimana menurut anda AI (Artificial Intelligence) yang diterapkan pada musuh dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 6. Bagaimana menurut anda control player yang diterapkan pada player dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 7. Menurut anda bagaimana desain environment dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 8. Menurut anda bagaimana tampilan menu utama dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
-

KUISIONER

Nama : Taufiqunafiman

1. Menurut anda bagaimana gameplay dalam game ini?
 Menarik b. Cukup Menarik c. Kurang menarik
 2. Menurut anda bagaimana desain karakter musuh dalam game ini?
 Menarik b. Cukup Menarik c. Kurang menarik
 3. Menurut anda bagaimana desain karakter player dalam game ini?
a. Menarik Cukup Menarik c. Kurang menarik
 4. Bagaimana menurut anda genre survival yang menjadi genre dalam game ini?
 Menarik b. Cukup Menarik c. Kurang menarik
 5. Bagaimana menurut anda AI (Artificial Intelligence) yang diterapkan pada musuh dalam game ini?
a. Menarik Cukup Menarik c. Kurang menarik
 6. Bagaimana menurut anda control player yang diterapkan pada player dalam game ini?
a. Menarik b. Cukup Menarik Kurang menarik
 7. Menurut anda bagaimana desain environment dalam game ini?
a. Menarik Cukup Menarik c. Kurang menarik
 8. Menurut anda bagaimana tampilan menu utama dalam game ini?
 Menarik b. Cukup Menarik c. Kurang menarik
-

KUISIONER

Nama : Bella Admaja

1. Menurut anda bagaimana gameplay dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 2. Menurut anda bagaimana desain karakter musuh dalam game ini?
 a. Menarik b. Cukup Menarik c. Kurang menarik
 3. Menurut anda bagaimana desain karakter player dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 4. Bagaimana menurut anda genre survival yang menjadi genre dalam game ini?
 a. Menarik b. Cukup Menarik c. Kurang menarik
 5. Bagaimana menurut anda AI (Artificial Intelligence) yang diterapkan pada musuh dalam game ini?
a. Menarik b. Cukup Menarik c. Kurang menarik
 6. Bagaimana menurut anda control player yang diterapkan pada player dalam game ini?
 a. Menarik b. Cukup Menarik c. Kurang menarik
 7. Menurut anda bagaimana desain environment dalam game ini?
 a. Menarik b. Cukup Menarik c. Kurang menarik
 8. Menurut anda bagaimana tampilan menu utama dalam game ini?
 a. Menarik b. Cukup Menarik c. Kurang menarik
-

Lampiran 1. Script Darah

```
using UnityEngine;
using System.Collections;
public class Darah : MonoBehaviour {
    [SerializeField]
    int _maximumHealth = 100;
    int _currentHealth = 0;
    override public string ToString()
    {
        return _currentHealth + "/" + _maximumHealth;
    }
    public bool IsDead{ get { return _currentHealth <= 0; } }
    void Start()
    {
        _currentHealth = _maximumHealth;
    }
    public void Damage (int damageValue)
    {
        _currentHealth -= damageValue;
        if (_currentHealth < 0) {
            _currentHealth = 0;
        }
        if (_currentHealth <= 0) {
            Animation a =
GetComponentInChildren<Animation>();
            a.Stop();
            Destroy(GetComponent<CharacterController>());
                Destroy(GetComponent<PlayerAnimation>());
                Destroy(GetComponent<Senjata>());
        }
        else//its an Enemy
        {
            Destroy(GetComponent<EnemyMovement>());

            Destroy(GetComponentInChildren<EnemyAttack>());
        }
        Destroy(GetComponent<CharacterController>()); }
    }
}
```

Lampiran 2. Script *Enemy Smoe Animation*

```
using UnityEngine;
using System.Collections;

public class EnemyAnimation : MonoBehaviour {
    Animation _animation;
    void Start()
    {
        _animation = GetComponentInChildren<Animation>();

        string animationToPlay = "";
        switch(Random.Range (0, 4))
        {
        default:

        case 0:
            animationToPlay = "move1_some";
            break;
        case 1:
            animationToPlay = "move2_some";
            break;
        case 2:
            animationToPlay = "move3_some";
            break;
        case 3:
            animationToPlay = "attack_some";
            break;
        case 4:
            animationToPlay = "idle_some";
            break;
        }
        _animation [animationToPlay].wrapMode = WrapMode.Loop;
        _animation.Play (animationToPlay);
        _animation[animationToPlay].normalizedTime =
Random.value;
    }
}
```

Lampiran 3. Script Senjata

```
using UnityEngine;
using System.Collections;
public class Senjata : MonoBehaviour {
    [SerializeField]
    int _damageKena = 50;

    AudioSource PlayerAudio;
    void Start()
    {
        Screen.lockCursor = true;
    }

    void Update ()
    {
        if (Input.GetKey (KeyCode.Escape)) {

            Screen.lockCursor = false;
        }
        if (Input.GetButtonDown ("Fire1")){
            Screen.lockCursor = true;
            Ray mouseRay = Camera.main.ViewportPointToRay
(new Vector3 (0.5f,0.5f, 0));
            RaycastHit hitInfo;
            if (Physics.Raycast (mouseRay, out hitInfo)){
                Darah darahMusuh =
hitInfo.transform.GetComponent<Darah>();

                if(darahMusuh != null)
                {
                    darahMusuh.Damage (_damageKena);
                }
            }
        }
    }
}
```

Lampiran 4. Script Item Speed boost

```
using UnityEngine;
using System.Collections;

public class PlayerStats : MonoBehaviour {

    int _jumlahMusuhTerbunuh = 0;
    public int musuhTerbunuh {get {return _jumlahMusuhTerbunuh;
} set {
        _jumlahMusuhTerbunuh = value;}}
}
```

Lampiran 5. Script Menu Play

```
function Start () {
}

function Update ()
{
}

function OnGUI ()
{
    if (Input.GetKey(KeyCode.P))
    {
        Screen.lockCursor = false;
        if (GUI.Button(Rect(Screen.width*0.5-50, 200-
20, 300, 40), "Tekan 'R' untuk Respawn"))
        {
            Application.LoadLevel(1);
        }

        if (GUI.Button(Rect(Screen.width*0.5-50, 240, 300, 40), "Tekan
'X' untuk Exit"))
        {
            Application.Quit();
        }

        if (GUI.Button(Rect(Screen.width*0.5-50, 300, 300, 40), "Tekan
'M' untuk Main Menu"))
        {
            Application.LoadLevel(0);
        }
    }
    else if (Input.GetKey(KeyCode.R))
    {
        Application.LoadLevel(1);
    }
    else if (Input.GetKey(KeyCode.X))
    {
        Application.Quit();
    }
}
```

```
else if (Input.GetKey(KeyCode.M))  
    {  
        Application.LoadLevel(0);  
    }  
}
```

Lampiran 6. Script *Advanced AI*

```
var Distance;

var Target : Transform;

var LookAtDistance = 25.0;

var chaseRange = 15.0;

var AttackRange = 1.5;

var moveSpeed = 5.0;

var Damping = 6.0;

var attackRepeatTime = 1.0;

private var attackTime : float;

var controller : CharacterController;

var gravity : float = 20.0;

private var MoveDirection : Vector3 = Vector3.zero;

function Start()

{

    attackTime = Time.time;

}

function Update ()

{

    Distance = Vector3.Distance(Target.position,
transform.position);

    if (Distance < LookAtDistance)

    {

        lookAt();

    }

}
```

```
    }  
    if(Distance > LookAtDistance)  
    {  
        renderer.material.color = Color.green;  
    }  
    if(Distance < AttackRange)  
    {  
        attack();  
    }  
    else if(Distance < chaseRange)  
    {  
        chase();  
    }  
}  
function lookAt ()  
{  
    renderer.material.color = Color.yellow;  
    var rotation = Quaternion.LookRotation(Target.position -  
transform.position);  
    transform.rotation = Quaternion.Slerp(transform.rotation,  
rotation, Time.deltaTime * Damping);  
}  
  
function chase()  
{  
    renderer.material.color = Color.red;  
    moveDirection = transform.forward;
```

```
    moveDirection *= moveSpeed;

    moveDirection.y -= gravity * Time.deltaTime;

    controller.Move(moveDirection * Time.deltaTime);
}

function attack()
{
    if(Time.time > attackTime)
    {
        attackTime = Time.time + attackRepeatTime;
    }
}

function ApplyDamage ()
{
    chaseRange += 30;

    moveSpeed += 2;

    LookAtDistance += 40;
}
```

Lampiran 7. Script *PlayerGUI*

```
using UnityEngine;
using System.Collections;

public class PlayerGui : MonoBehaviour {

    [SerializeField]
    Texture2D _crosshair;

    Darah _playerHealth;

    void Start()

    {
        _playerHealth = GetComponent<Darah>();
    }

    void OnGUI()

    {
        GUI.Label(new Rect(5,5,100,100), "Health:" +
            _playerHealth.ToString ());

        float x = (Screen.width - _crosshair.width) / 2;
        float y = (Screen.height - _crosshair.height) / 2;

        GUI.DrawTexture (new Rect (x, y, _crosshair.width,
            _crosshair.height),
            _crosshair);
    }
}
```

Lampiran 8. Player Stats

```
using UnityEngine;

using System.Collections;

public class PlayerStats : MonoBehaviour {

    int _jumlahMusuhTerbunuh = 0;

    public int musuhTerbunuh {get {return _jumlahMusuhTerbunuh;
} set {

        _jumlahMusuhTerbunuh = value;}}

}
```

Lampiran 9. *Enemy Attack*

```
using UnityEngine;
using System.Collections;

public class EnemyAttack : MonoBehaviour {

    [SerializeField]
    int _besarDamage = 5;

    float _nextTimeAttackIsAllowed = -1.0f;

    void OnTriggerStay(Collider other)
    {
        if(other.tag == "Player" && Time.time >=
        _nextTimeAttackIsAllowed)
        {
            Darah playerHealth =
other.GetComponent<Darah>();

            playerHealth.Damage (_besarDamage);
            _nextTimeAttackIsAllowed = Time.time + 1.0f;
        }
    }
}
```

Lampiran 10. Tampil Darah

```
using UnityEngine;
using System.Collections;

public class TampilDarah : MonoBehaviour {

    Darah _playerHealth;

    void Start()
    {
        _playerHealth = GetComponent<Darah>();
    }

    // Use this for initialization

    void OnGUI () {

        GUI.Label(new Rect(20,20,500,500), "Health:" +
            _playerHealth.ToString ());

    }

}
```

Lampiran 11. *Script Main Menu*

```
using UnityEngine;
using System.Collections;

public class AMainMenu : MonoBehaviour {

    [SerializeField]
    GUISkin _guiSkin;

    [SerializeField]
    Texture2D _mainMenuBg;

    [SerializeField]
    Texture2D _howToPlay;

    [SerializeField]
    Texture2D _aboutButton;

    [SerializeField]
    Texture2D _galleryButton;

    [SerializeField]
    Texture2D _newGameButton;

    [SerializeField]
    Texture2D _quitButton;

    void OnGUI()
```

```
{  
  
    GUI.skin = _guiSkin;  
  
    GUI.DrawTexture(new Rect(0, 0, Screen.width,  
Screen.height),  
  
        _mainMenuBg);  
  
    if (GUI.Button(new Rect(70, 130, 270,  
_newGameButton.height),  
  
        _newGameButton) && !OptionsGui.IsOpen)  
  
    {  
  
        Application.LoadLevel(5);  
  
    }  
  
    if (GUI.Button(new Rect(70, 130 + 44 + 10, 270,  
_howToPlay.height),  
  
        _howToPlay))  
  
    {  
  
        Application.LoadLevel(6);  
  
    }  
  
    if (GUI.Button (new Rect (70, 130 + (2 * (44 + 10)),  
270, _aboutButton.height),  
  
        _aboutButton))  
  
    {  
  
        Application.LoadLevel(7);  
  
    }  
  
    if (GUI.Button (new Rect (70, 130 + (3 * (44 + 10)),
```

```
270, _galleryButton.height),  
        _galleryButton))  
    {  
        Application.LoadLevel(8);  
    }  
  
    if (GUI.Button (new Rect (70, 130 + (4 * (44 + 10)),  
270,  
        _quitButton.height),  
_quitButton) && !OptionsGui.IsOpen)  
    {  
        Application.Quit();  
    }  
}  
}
```

Lampiran 12. *Script Display Level*

```
var target : Transform;

function Update () {

var other = gameObject.GetComponent("Level2");

if ( Vector3.Distance(target.position, transform.position ) < 3) {
other.enabled = true;
}

if ( Vector3.Distance(target.position, transform.position ) > 3) {
other.enabled = false;
}
}
```

Lampiran 13. *Script Laser*

```
function Start () {  
    this.renderer.enabled = false;  
}  
function Update () {  
    if(Input.GetButton("Fire1"))  
    {  
        if(this.renderer.enabled == false)  
        {  
            this.renderer.enabled = true;  
        }  
    }  
    else  
    {  
        this.renderer.enabled = false;  
    }  
}
```

Lampiran 14. *Script Tembak*

```
var peluru : Transform;
var peluruK : Transform;
var speed =10000;
function Update ()
{
    if(Input.GetButton("Fire1"))
    {
        Tembak();
    }
}
function Tembak ()
{
    var pel = Instantiate(peluru, peluruK.position,
peluruK.rotation);
    pel.rigidbody.AddForce(transform.forward * speed);
    Destroy (pel.gameObject, 2);
}
```

Lampiran 15. *Script Turrent Collision*

```
var explosion : Transform;

function OnTriggerEnter (hit : Collision) {

    if(hit.gameObject.tag == "Player")

    {

        Destroy(hit.gameObject);

        Destroy(gameObject);

    }

}
```

Lampiran 16. *Script* Petunjuk

```
function OnGUI () {  
  
    // Make a background box  
  
    GUI.Box (Rect (10,10,300,90), "Bantu NPC di tengah kepungan  
Musuh,");  
  
  
    // Make the first button. If it is pressed,  
Application.LoadLevel (1) will be executed  
  
    if (GUI.Button (Rect (130,40,80,20), "Yeah")) {  
  
        Application.LoadLevel(1);  
  
    }  
  
}
```