

DESAIN DAN IMPLEMENTASI VIRTUAL REALITY FIRST PERSON MILITARY SIMULATION GAME

SKRIPSI

Diajukan untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Komputer

Disusun oleh :

PRASETYA ARDHANA SARWONO

NIM. 115060800111032



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG

2016

PENGESAHAN

DESAIN DAN IMPLEMENTASI VIRTUAL REALITY
FIRST PERSON MILITARY SIMULATION GAME

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Prasetya Ardhana Sarwono
NIM: 115060800111032

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Erig Muhammad Adams J, ST., M.Kom
NIP: 19850410 2012121001

Wibisono Sukmo Wardhono, ST.,
NIK: 201008 820404 1 001

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Marji, Drs., M.T
NIP : 196708011992031001



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 25 Desember 2015

Mahasiswa,

Prasetya Ardhana Sarwono

NIM. 115060800111032



KATA PENGANTAR

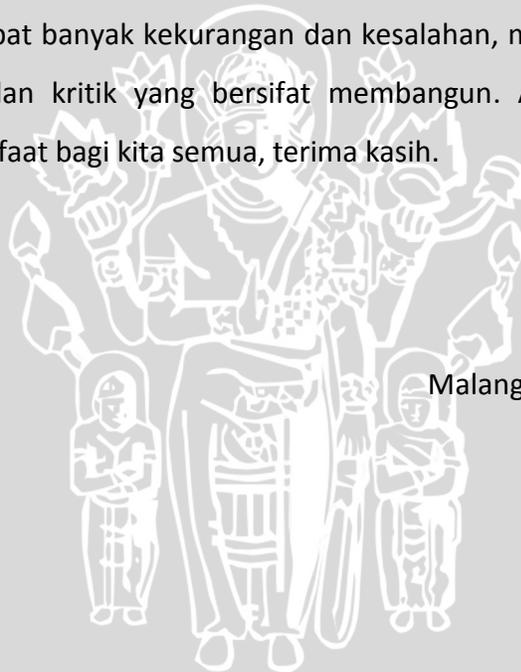
Dengan ucapan syukur Alhamdulillah, penulis panjatkan ke hadirat Allah SWT, atas limpahan rahmat, taufik serta karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Desain dan Implementasi Virtual Reality First Person Military Simulation Game” dengan baik.

Penulis menyadari dalam penyelesaian skripsi ini banyak pihak yang membantu baik secara moril, materil maupun do’a. Oleh karena itu, pada kesempatan ini penulis menyampaikan terimakasih kepada :

1. Bapak Eriq M. Adams J, S.T., M.Kom dan Bapak Wibisono Sukmo Wardhono, S.T., M.T. selaku dosen pembimbing penulis. Terima Kasih atas semua bimbingan dan dorongan semangatnya.
2. Bapak Drs. Marji, M.Si dan Issa Arwani, S.Kom,.M.Sc selaku Ketua dan Sekertaris Program Studi Informatika serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Program Studi Teknik Informatika Universitas Brawijaya.
3. Seluruh dosen Informatika/Illmu Komputer Universitas Brawijaya atas kesediaan membagi ilmu kepada penulis.
4. Seluruh civitas akademika Informatika/ilmu komputer Universitas Brawijaya terutama yang telah banyak membantu dan memberi dukungan selama penulisan skripsi ini.
5. Nenek penulis, Ibu Suyati Dhadi, yang selalu menyayangi dan mendoakan penulis dengan tulus hingga akhirnya penulis dapat menyelesaikan skripsi ini.
6. Kedua Orang Tua penulis, Ibu Prihyanti Lumbanraja dan Bapak Adji Sarwono Adjer Soeripno, yang telah memberikan dukungan secara motivasi dan material, dan semua doa dan kasih sayang yang telah diberikan kepada penulis. Terima kasih kepada Asti Lukita Wardani dan Nasytha Tristie Wardhani, kedua saudara saya yang selalu memberikan kasih sayang dan dorongan semangat untuk saya.

7. Seluruh sahabat yang selalu menemani penulis selama masa perkuliahan dan mendukung pengerjaan skripsi, Tyo, Adit, Zakki, Loka, Deli, Dio, Brian, Faisol, Harir, Dimas, dan kawan-kawan lainnya.
8. Teman-teman terdekat penulis, Fia, Hestia, Caco, Irna, Dayen, Tyok, Erwin, Arief, Fefe, Ugi, Topher, Suroso, Satria, Radit, Wibi, Way, Geky, Aan, Fauzy, Reza, Dicky, Mirza, dan masih banyak lainnya.
9. Semua pihak yang telah membantu penulis dalam pengerjaan skripsi yang tidak bisa disebutkan satu persatu.

Dengan segala kerendahan hati, dalam penyusunan skripsi ini penulis menyadari masih terdapat banyak kekurangan dan kesalahan, maka dari itu penulis mengharapkan saran dan kritik yang bersifat membangun. Akhir kata, semoga skripsi ini dapat bermanfaat bagi kita semua, terima kasih.



Malang, 25 Desember 2015

Penulis



ABSTRAK

Prasetya Ardhana Sarwono. 2016. Desain dan Implementasi *Virtual Reality First Person Military Simulation Game*. Skripsi Program Studi Informatika/ Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Pembimbing : Eriq M. Adams J, S.T., M.Kom dan Wibisono Sukmo Wardhono, S.T., M.T.

Pada masa kini, *video game* tidak hanya digunakan sebagai media hiburan, melainkan banyak juga yang memanfaatkan *video game* sebagai sarana edukasi dan simulasi. Simulasi merupakan suatu proses peniruan dari sesuatu yang nyata beserta keadaan sekelilingnya. Oleh karena itu, pemanfaatan *video game* sebagai media simulasi banyak dimanfaatkan dalam melakukan beragam aktivitas. Salah satu pemanfaatan *video game* sebagai media simulasi dilakukan pada bidang militer. Media *simulator* dan *video game* yang diimplementasikan menggunakan *virtual reality* dimanfaatkan oleh rekrutan militer Amerika Serikat untuk melatih dan mengasah keterampilan tempur mereka. Dengan *virtual reality video game* bergenre *first person shooter* menggunakan *Head Mounted Display* Google Cardboard, akan membawa pemain ke dalam dunia *virtual reality* seolah berada dalam medan perang.

Kata Kunci : Desain dan Implementasi, *First Person Shooter*, *Virtual Reality* HMD, Google Cardboard, Permainan Video, Simulasi, Simulasi Militer

ABSTRACT

Prasetya Ardhana Sarwono. 2016. *Design and Implementation of Virtual Reality First Person Military Simulation Game. Undergraduate Thesis of Informatic Study Program, Faculty of Computer Science, Brawijaya University. Advisor: Eriq M. Adams J, ST, M.Kom and Wibisono Sukmo Wardhono, S.T., M.T.*

Nowadays, video game isn't just being used as an entertainment media, but there are lots who utilizes video game as a tool of education and simulation. Simulation is a process of copying something which is real and everything around it. Therefore, utilization of video game as a simulation media were used to do a lot of activities. One of the utilization of video game as a simulation media were done in the military field. Simulator and video game media which was implemented using virtual reality were used by the U.S. military recruits to train and hone their combat skills. With First Person Shooter virtual reality video game using Head Mounted Display Google Cardboard, wil take the players into the world of virtual reality as if they were in a war zone.

Keywords : *Design and Implementation, First Person Shooter, Virtual Reality HMD, Google Cardboard, Video Game, Simulation, Military Simulation*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Manfaat	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN KEPUSTAKAAN	5
2.1 <i>Game Design</i>	5
2.2 <i>First Person Shooter</i>	5
2.3 <i>Virtual Reality</i>	6
2.4 Trinus VR.....	7
2.5 Google Cardboard.....	7
2.6 <i>Virtual Reality Military Simulation</i>	7
2.7 <i>Game Testing</i>	8
1. <i>Combinatorial Testing</i>	8
2. <i>Test Flow Diagram</i>	8

3. <i>Play Testing</i>	9
BAB III METODE PENELITIAN	12
3.1 Studi Literatur	13
3.2 Perancangan <i>Game</i>	13
1. Deskripsi <i>Game</i>	13
2. <i>Gameplay</i>	13
3. <i>Paper Prototyping</i>	13
3.3 Implementasi <i>Game</i>	13
3.4 Pengujian dan Analisis	14
BAB IV PERANCANGAN DAN IMPLEMENTASI	15
4.1 Perancangan <i>Game</i>	15
4.1.1 Deskripsi <i>Game</i>	15
4.1.2 <i>Gameplay</i>	16
4.1.3 <i>Paper Prototyping</i>	18
4.2 Implementasi <i>Game</i>	25
4.2.1 Penentuan Spesifikasi Sistem	25
4.2.2 Implementasi <i>Gameplay</i>	26
4.2.3 Implementasi VR	33
4.2.4 Implementasi <i>Art dan User Interface</i>	34
BAB V PENGUJIAN DAN EVALUASI	42
5.1 <i>Combinatorial Testing</i>	42
5.2 <i>Test Flow Diagram</i>	44
5.2.1 TFD <i>Game Flow</i>	44
5.2.2 TFD <i>Weapon Ammo</i>	48
5.3 <i>Play Testing</i>	50
BAB VI PENUTUP	55
6.1 Kesimpulan	55
6.2 Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN A	58



DAFTAR GAMBAR

Gambar 2.1 Virtual Reality HMD.....	7
Gambar 3.1 Blok Diagram Runtutan Pengerjaan Skripsi	12
Gambar 4.1 Tampilan pada Sudut Pandang Orang Pertama.....	16
Gambar 4.2 Denah Rancangan Level	19
Gambar 4.3 Perancangan HUD pada <i>Game</i>	20
Gambar 4.4 <i>Game Screen Flow</i> Keseluruhan.....	20
Gambar 4.5 Rancangan <i>Title Screen</i>	21
Gambar 4.6 Rancangan <i>Difficulty Select Screen</i>	22
Gambar 4.7 Rancangan <i>Developers Screen</i>	22
Gambar 4.8 Rancangan <i>Instructions and Controls Screen</i>	23
Gambar 4.9 Rancangan <i>Exit Menu Screen</i>	23
Gambar 4.10 Rancangan <i>Mission Failed Screen</i>	24
Gambar 4.11 Rancangan <i>Complete Screen</i>	24
Gambar 4.12 Rancangan <i>Pause Screen</i>	25
Gambar 4.13 Tampilan Implementasi VR.....	34
Gambar 4.14 Implementasi Level	34
Gambar 4.15 <i>Art Health Bar</i>	35
Gambar 4.16 <i>Art Crosshair</i>	35
Gambar 4.17 <i>Title Screen</i>	36
Gambar 4.18 <i>Difficulty Select Screen</i>	36
Gambar 4.19 <i>Developers Screen</i>	37
Gambar 4.20 <i>Instructions and Controls Screen</i>	37
Gambar 4.21 <i>Exit Menu Screen</i>	38
Gambar 4.22 <i>Mission Failed Screen</i>	38
Gambar 4.23 <i>Mission Complete Screen</i>	39
Gambar 4.24 <i>Pause Screen</i>	39
Gambar 4.25 Tampilan Karakter Pemain.....	40
Gambar 4.26 Tampilan Anggota Regu Polisi.....	40

Gambar 4.27 Tampilan Tentara Musuh 41

Gambar 4.28 Tampilan Sandera 41

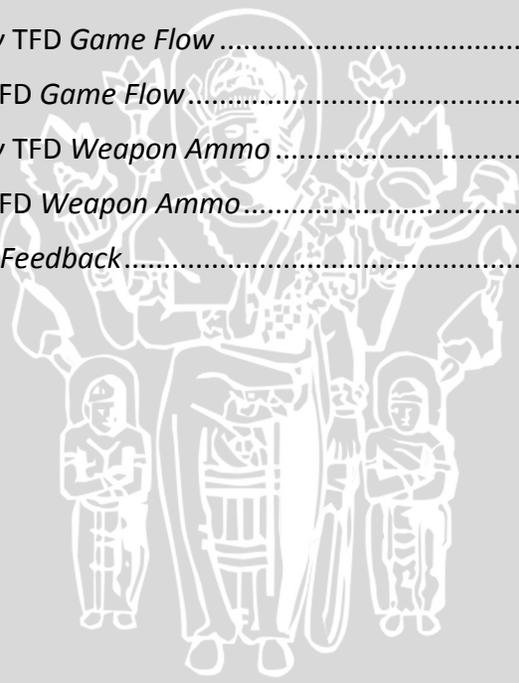
Gambar 5.1 *Test Flow Diagram Game Flow* 44

Gambar 5.2 *Test Flow Diagram Weapon Ammo* 48



DAFTAR TABEL

Tabel 4.1 Spesifikasi Perangkat Keras	26
Tabel 4.2 Spesifikasi Perangkat Lunak	26
Tabel 4.3 <i>Script Code</i> Kontrol Pemain	27
Tabel 4.4 <i>Script Code</i> Mekanik Menembak	28
Tabel 4.5 <i>Script Code</i> Situasi Sandera	30
Tabel 4.6 <i>Script Code</i> Mekanik Nyawa	31
Tabel 4.7 <i>Script Code</i> Situasi Menang	32
Tabel 4.8 <i>Script Code</i> Situasi Kalah	33
Tabel 5.1 Hasil Kombinasi Parameter	42
Tabel 5.2 <i>Data Dictionary</i> TFD <i>Game Flow</i>	45
Tabel 5.3 Hasil Uji <i>Flow</i> TFD <i>Game Flow</i>	46
Tabel 5.4 <i>Data Dictionary</i> TFD <i>Weapon Ammo</i>	49
Tabel 5.5 Hasil Uji <i>Flow</i> TFD <i>Weapon Ammo</i>	50
Tabel 5.6 Hasil Kuisisioner <i>Feedback</i>	51



DAFTAR LAMPIRAN

LAMPIRAN A Kuisiner *Feedback* Pengujian *Play Testing* Game “Save Me Now!!” .. 58



BAB I

PENDAHULUAN

1.1 Latar Belakang

Video game sebagai sebuah sarana permainan memiliki fungsi dan manfaat selain sebagai media hiburan yaitu sebagai sarana edukasi dan simulasi (Trybus, 2014). Media *simulator* dan *video game* juga dimanfaatkan oleh rekrutan militer Amerika Serikat untuk melatih dan mengasah keterampilan tempur mereka (Hsu, 2010). Simulasi dalam bidang militer menggunakan *video game* ataupun VR (*Virtual Reality*) merupakan hal yang perlu dikembangkan untuk meningkatkan keterampilan bertempur tanpa menyebabkan kerusakan yang tidak perlu. Memanfaatkan media simulasi juga merupakan suatu upaya yang dapat menghemat amunisi dan menghindari kerusakan yang tidak diharapkan (VPL Research, 2015). Salah satu fokus yang dapat diambil untuk dijadikan sebuah simulasi yang masih satu lingkup ruang dengan pelatihan militer adalah situasi penyelamatan sandera.

Dalam pembuatannya, *video game* memerlukan dukungan dari *game environment* yang baik untuk membuat permainan lebih realistis. Beberapa hal yang dapat mendukung hal tersebut yaitu grafis, agen cerdas, dan perancangan *game* itu sendiri. Dengan membuat sebuah rancangan *game* yang baik dan benar, maka nantinya pembuatan *video game* tersebut akan menjadi lebih terstruktur dan teratur. Terdapat beberapa metode yang dapat digunakan dalam membuat perancangan *video game*, beberapa diantaranya yaitu metode *Paper Prototyping* yang merupakan sebuah metode perancangan *game* yang sering digunakan. Hal ini dikarenakan dengan menggunakan metode ini, kita dapat menghemat biaya dan waktu dengan menggambarkan rancangan dari *game* yang akan kita buat di atas kertas dan mendiskusikannya dengan anggota tim lainnya (Snyder, 2003). Pendukung lain untuk membuat suatu permainan yang realistis adalah VR (Halarnkar, 2012). Salah satu media VR yang sudah dikenal di kalangan umum adalah Google Cardboard yang merupakan sebuah proyek VR dari perusahaan Google yang berbentuk *Head Mounted Device* (HMD) (Google Developers, 2014).

Berdasarkan hal-hal tersebut, dalam skripsi ini akan dibuat Desain dan Implementasi *Virtual Reality First Person Military Simulation Game* yang dapat dimanfaatkan untuk melakukan simulasi pelatihan militer. Fokus penulis dalam pengembangan dari *game* ini adalah perancangan dan implementasi *gameplay*, *art*, dan *user interface*. Sedangkan pemodelan pergerakan *Non Player Character* (NPC) dan *Artificial Intelligence* (AI) dikerjakan pada skripsi berjudul “Implementasi *Squad Formation Behavior* untuk *Non Player Character* pada *First Person Military Simulation Game* Menggunakan Algoritma *Flocking*” oleh Achmad Zakki Ridwan,

kemudian “Implementasi Perilaku Taktis dan Strategis *Non Player Character* pada *First Person Military Simulation Game* Menggunakan *Finite State Machine*” oleh Adhitya Wiratama, dan “Implementasi *Pathfinding* pada *First Person Military Simulation Game* dengan algoritma HPA*” oleh Tyo Prasetyo Wibowo.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang, maka dapat dirumuskan permasalahan pada penelitian ini yaitu :

1. Bagaimana *gameplay* dan aturan dari *Virtual Reality First Person Military Simulation Game*?
2. Bagaimana mengimplementasikan hasil perancangan *gameplay* dan aturan dari *Virtual Reality First Person Military Simulation Game*?
3. Bagaimana fungsional implementasi dari *Virtual Reality First Person Military Simulation Game* terhadap *gameplay*?
4. Bagaimana tingkat kepuasan para pemain sebagai *play tester* setelah memainkan *Virtual Reality First Person Military Simulation Game*?

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan lebih terfokus, maka penelitian ini dibatasi oleh hal-hal sebagai berikut :

1. Lingkungan setting dari permainan merupakan map yang terbuka.
2. Skenario dari permainan adalah situasi penyelamatan sandera.
3. *Game* akan diimplementasikan pada *Virtual Reality Head Mounted Device* Google Cardboard menggunakan media implementasi berupa aplikasi TrinusVR.
4. Media Google Cardboard yang diimplementasikan pada *game* hanya berdampak pada sudut pandang pemain.
5. Skripsi ini berfokus kepada perancangan desain dan implementasi dari *game*.

1.4 Tujuan

Tujuan yang ingin dicapai dalam implementasi skripsi ini adalah

1. Untuk merancang sebuah *Virtual Reality First Person Military Simulation Game*.
2. Untuk mengimplementasikan perancangan yang sudah dibuat menjadi *game* yang utuh.

3. Mengetahui apakah fungsional implementasi *gameplay* yang telah dibuat dapat berjalan dengan baik.
4. Mengetahui bagaimana tingkat kepuasan pemain sebagai *play tester* setelah memainkan *game*.

1.5 Manfaat

Manfaat yang diharapkan dari implementasi skripsi ini adalah :

1. Agar perancangan yang dibuat dengan metode *Paper Prototyping* dapat membuat *First Person Military Simulation Game* menjadi lebih realistis.
2. Dapat dijadikan sebagai pembanding atau literatur dalam penelitian yang menerapkan *Paper Prototyping* dan Google Cardboard dimasa yang akan datang.

1.6 Sistematika Penulisan

Sistematika isi dan penulisan dalam skripsi ini antara lain:

BAB I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat dari penelitian serta sistematika penulisan.

BAB II Landasan Kepustakaan

Menguraikan tentang dasar teori dan teori penunjang yang berkaitan dengan *Game Design*, *First Person Shooter*, *Virtual Reality*, *TrinusVR*, *Google Cardboard*, *Virtual Reality Military Simulation*, dan *Game Testing*.

BAB III Metode Penelitian

Berisi tentang gambaran umum langkah-langkah pembuatan *game* dimulai dari tahap perancangan, implementasi serta pengujian secara umum.

BAB IV Rancangan dan Implementasi

Memuat perancangan dari *game* dan bagaimana mengimplementasikan perancangan tersebut ke dalam bentuk nyata dari *game*.

BAB V Pengujian dan Evaluasi

Memuat proses pengujian dan evaluasi dari hasil implementasi perancangan *game* yang telah dibuat dengan metode *paper prototyping*.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan *game design*, pengimplementasian, dan pengujian dari *game*, serta saran-saran untuk pengembangan lebih lanjut.



BAB II

LANDASAN KEPUSTAKAAN

Bab ini membahas dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai Desain dan Implementasi *Virtual Reality First Person Military Simulation Game*.

2.1 Game Design

Game design merupakan salah satu proses penting yang harus kita lakukan sebelum kita memulai untuk membangun sebuah *game* dimana kita membuat dan merancang hal-hal yang menjadi unsur utama *game* seperti tujuan dari *game*, peraturan yang berlaku pada *game*, karakter, tema, interaksi dengan pemain, dan sistem dari *game*. Dalam *game design* kita tidak perlu melakukan *programming*, pembuatan *art* dan animasi, maupun *marketing* dari *game* (Schreiber, 2009).

Salah satu metode yang dapat diaplikasikan dalam pembuatan *game design*, antara lain adalah metode *Paper Prototyping*. *Paper Prototyping* merupakan suatu metode yang telah digunakan secara umum dalam hal membuat desain, melakukan pengujian, dan menentukan *user interface* dengan cara membuat sketsa maupun replika sederhana menggunakan kertas dan gambar yang nantinya akan didiskusikan bersama tim pengembang *game* dalam pengambilan keputusannya (Snyder, 2003).

Biasanya *paper prototyping* jarang digunakan pada genre *first person shooter*, namun hal ini bukan berarti *paper prototyping* tidak dapat dimanfaatkan dalam pengembangan *game first person shooter*. *Paper prototyping* pada pengembangan *game FPS* dapat diaplikasikan dalam hal evaluasi tata letak peta, logika *flow* antar ruang, perancangan *user interface* dan *head up display*, ataupun rincian-rincian lainnya (Grace, 2009).

Salah satu contoh penerapan *paper prototyping* pada *game FPS* dilakukan pada *game Doom*. Dibutuhkan beberapa orang yang menggerakkan karakter *artificial intelligence* dan juga karakter pemain. Peta level digambarkan pada sebuah kertas besar dan diletakkan beberapa kepingan yang mewakili karakter-karakter yang akan digerakkan. Karakter akan digerakkan dengan menggunakan aturan pergiliran seakan-akan bagaimana *game* akan berjalan nantinya (Grace, 2009).

2.2 First Person Shooter

First Person Shooter (FPS) merupakan sebuah genre dari permainan menembak tiga dimensi yang menggunakan sudut pandang orang pertama dimana pemain melihat aksi melalui sudut pandang karakter pemain (Rollings, 2009). Genre FPS berpusat pada perasaan dan pandangan pemain sebagai orang pertama, sama seperti jika pemain berada di dunia 3D dengan membawa senapan maupun senjata

yang digunakan untuk menembak (Yeruva, 2011). Oleh karena itu, banyak *game* bergenre FPS yang dijadikan media simulasi militer atau peperangan.

Dalam genre FPS, *Artificial Intelligence* (AI) merupakan faktor penting yang dapat mempengaruhi kesuksesan genre ini pada dunia *industry game* (Yeruva, 2011). AI pada FPS kebanyakan digunakan untuk mengatur pergerakan dan kecerdasan *bot* musuh sebagai lawan main para pemain maupun *bot* kawan dari pemain (McPartland & Gallagher, 2013).

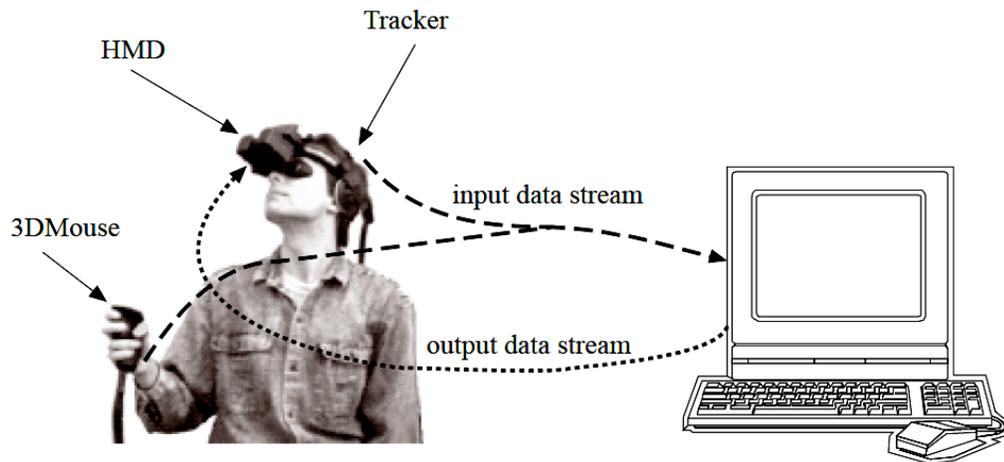
2.3 Virtual Reality

Virtual reality (VR) merupakan teknologi yang dapat membuat pengguna berinteraksi dengan sebuah lingkungan yang disimulasikan oleh komputer (*computer-simulated environment*), sebuah lingkungan sebenarnya yang ditiru atau benar-benar sebuah lingkungan yang hanya ada dalam imajinasi (Sihite, 2013). VR dapat diaplikasikan melalui beberapa media seperti komputer, HMD (*Head-Mounted Display*), *headphone*, dan *motion-sensing gloves* (Steuer, 1993). Media VR yang saat ini merupakan media terpopuler adalah *Virtual Reality Head-Mounted Display* (VR HMD).

Menurut Pallavi Halarnkar dkk dalam jurnal "A Review on Virtual Reality" pada tahun 2012, menjelaskan definisi dari VR HMD. Berikut penjelasan dari jurnal tersebut.

Virtual Reality merupakan lingkungan dari simulasi komputer yang mensimulasikan kehadiran fisik seseorang di dunia nyata maupun maya. Pengguna dapat merasakan dan menyentuh benda simulasi dalam lingkungan yang dibuat dalam simulasi, memberikan persepsi bahwa benda-benda dalam lingkungan tersebut benar-benar ada. Mereka dapat berinteraksi dengan lingkungan virtual baik melalui penggunaan perangkat input standar seperti *keyboard* dan *mouse*, atau media perangkat keras lain. Dengan demikian, pengguna berinteraksi dengan objek virtual 3D dalam kondisi *real-time* dengan cara yang intuitif & alami, memahami mereka sebagai sesuatu yang nyata dan dapat memahami, menganalisis dan berkomunikasi dalam lingkungan simulasi.

Morton Heilig, pencipta Sensorama pada tahun 1956 telah menemukan dan mematenkan sebuah perangkat yang dinamakan sebagai *Head-Mounted Display* (HMD). Pada gambar 2.1 merupakan komponen *virtual reality head mounted display* (VR HMD).



Gambar 2.1 *Virtual Reality HMD*

Sumber : Pallavi (2012)

2.4 Trinus VR

Trinus VR merupakan sebuah *software* yang dibuat untuk para pemain *game* yang ingin memainkan *game* dengan mode *virtual reality* yang menggunakan tampilan dan sensor dari sebuah perangkat *smartphone* dan mengubahnya menjadi sebuah portal untuk *game* yang ingin dimainkan (Odd Sheep, 2015). *Software* ini digunakan dengan cara dikombinasikan dengan *headmount* seperti Google Cardboard, Homido, Durovis Dive, ColorCross, VR One, dan lain-lain dan dapat mengkonversikan sebagian besar *software* komputer ke dalam mode *virtual reality* (Odd Sheep, 2015).

2.5 Google Cardboard

Google Cardboard merupakan sebuah media VR yang dikembangkan oleh perusahaan mesin pencari terbesar dunia Google yang memanfaatkan sebuah perangkat *smartphone* yang dipasangkan pada sebuah kertas karton (*cardboard*) yang membentuk sebuah *head-mounted display* (HMD). Google Cardboard sendiri ditujukan sebagai sebuah sistem yang murah dan diharapkan dapat mendorong minat dalam pembangunan VR dan aplikasi VR (Google Developers, 2014).

2.6 *Virtual Reality Military Simulation*

Virtual Reality telah diadopsi oleh pihak militer, hal ini mencakup angkatan darat, angkatan laut, serta angkatan udara yang digunakan sebagai media pelatihan (VPL Research, 2015). Media VR sangat berguna bagi para prajurit untuk berlatih mengenai bagaimana mereka harus melakukan reaksi saat berada dalam situasi perang maupun situasi yang berbahaya. Mereka dapat mensimulasikan skenario-skenario perang tersebut tanpa mereka harus mengalami resiko kematian maupun

cedera yang serius (VPL Research, 2015). Hal ini membuktikan bahwa simulasi *virtual reality* pada militer membuat pelatihan lebih aman dan lebih tidak memakan biaya dibandingkan dengan metode pelatihan tradisional.

Beberapa contoh penggunaan *virtual reality simulation* pada militer antara lain:

- *Flight Simulation*
- *Battlefield Simulation*
- *Medic training on battlefield*
- *Vehicle Simulation*
- *Virtual boot camp*

Virtual reality juga digunakan untuk terapi *post-traumatic stress disorder*. Para prajurit yang menderita trauma setelah bertempur di medan perang ataupun yang mengalami gangguan psikologis lain dapat melakukan terapi di lingkungan yang “aman”. Mereka dapat melakukan berada dalam kondisi yang biasa mereka alami dan secara perlahan beradaptasi dengan kondisi baru untuk menghilangkan stress ataupun trauma (VPL Research, 2015).

2.7 Game Testing

Game testing adalah sebuah proses pengujian dari sebuah *game* untuk menentukan apakah *game* tersebut dapat berjalan sesuai dengan fungsionalitasnya. *Game testing* juga dapat dilakukan untuk menguji apakah *game* sudah diimplementasikan dengan benar atau mencari *bug* maupun kekurangan yang masih terdapat pada *game* (Schultz, 2005).

Dalam *game testing* terdapat beberapa metode yang dapat diaplikasikan agar memaksimalkan pengujian *game*, yaitu:

1. Combinatorial Testing

Metode *combinatorial testing* merupakan metode yang dilakukan dengan mengkombinasikan satu parameter dengan beberapa parameter lain pada *game* (Schultz, 2005). Setelah kombinasi-kombinasi parameter dibuat, maka selanjutnya *game* akan diuji dengan cara menjalankan kombinasi parameter tersebut pada saat bermain *game* (Schultz, 2005). Metode ini digunakan untuk meminimalisir waktu yang dibutuhkan dalam melakukan tes pada sekian banyaknya parameter yang ada pada *game*, sehingga kita tidak perlu melakukan pengujian parameter-parameter tersebut satu persatu (Schultz, 2005).

2. Test Flow Diagram

Test flow diagram (TFD) adalah model grafik yang merepresentasikan bagaimana seharusnya *game* berjalan dari perspektif pemain (Schultz, 2005). *Test flow diagram* menggunakan diagram untuk merepresentasikan bagaimana game

berjalan dengan cara yang biasa dan dengan cara yang tidak terduga. Dengan mengeksplorasi model ini, penguji dapat mengetahui koneksi-koneksi yang tidak diantisipasi dan mengetahui kondisi *game* yang tidak terduga. TFD ini sangat mudah untuk dimengerti dan dianalisa oleh penguji, pengembang dan produser *game*, oleh karena itu metode ini cukup tepat digunakan dalam pengujian sebuah *game* (Schultz, 2005).

3. Play Testing

Play testing merupakan metode pencarian kekurangan yang dianggap dilakukan dengan cara yang lebih tidak terstruktur dibandingkan dengan metode lainnya (Schultz, 2005). Metode ini dilakukan dengan memainkan *game* untuk menguji beberapa kualitas pada *game* seperti keseimbangan, tingkat kesusahan, dan juga "*fun factor*" (Schultz, 2005). *Play testing* dilakukan untuk mempelajari bagaimana reaksi dan kesenangan pemain mengenai aktifitas yang dilakukan. Tidak seperti teknik pengujian *game* yang sudah disebutkan sebelumnya, *play testing* mengacu kepada pertanyaan, opini, dan penilaian setiap pemain berdasarkan fakta-fakta yang ada pada *game* tersebut (Schultz, 2005).

Dalam pelaksanaan *play testing* nantinya para penguji yang memainkan *game* akan diberikan kuisioner *feedback* yang akan dijawab sesuai dengan apa yang dirasakan para penguji mengenai *game*. Jawaban-jawaban dari kuisioner tersebut nantinya akan diolah menggunakan perhitungan skala likert. Skala likert merupakan suatu skala psikometrik yang umum digunakan dalam kuisioner dan survey (Budiaji, 2013).

Menurut Budiaji dalam jurnal "Skala Pengukuran dan Jumlah Respon Skala Likert" pada tahun 2013, menjelaskan rumus penghitungan jawaban kuisioner menggunakan skala likert. Berikut penjelasan dari jurnal tersebut.

Dalam penghitungan skala likert, hal pertama yang harus kita lakukan adalah menentukan skor dari tiap jawaban yang akan diberikan. Contohnya, sikap yang akan kita pakai yaitu "Baik". Selanjutnya kita menentukan banyaknya jawaban pada tiap pertanyaan yang akan kita berikan. Misalnya 5 skala, berarti Sangat Kurang, Kurang, Cukup, Baik, dan Sangat Baik. Jika pertanyaan yang diberikan bersifat susah untuk diberikan jawaban, otomatis responden cenderung statik oleh karena itu kita dapat memberikan pilihan jawaban yang banyak, misal 7 atau 9 jawaban dari tiap pertanyaan. Hal ini bertujuan agar responden dapat memberikan penilaian sesuai dengan kriteria mereka berdasarkan pilihan yang ada.

Masing-masing dari jawaban akan diberikan nilai untuk dihitung dengan pembagian skor dengan contoh sebagai berikut:

Sangat Bagus (SB) = 5

Bagus (B) = 4

Cukup (C) = 3

Kurang (K) = 2

Sangat Kurang (SK) = 1

Kemudian hasil jawaban akan dihitung dengan rumus $T \times P_n$, dimana T merupakan jumlah panelis yang memilih dan P_n adalah nilai skor likert yang sudah diberikan untuk tiap jawaban. Setelah dihitung dengan rumus $T \times P_n$, hasil skor dari tiap jawaban akan dijumlahkan untuk mendapatkan total skor. Untuk mendapat hasil interpretasi, harus diketahui dulu nilai skor tertinggi (Y) dan nilai skor terendah (X) untuk item penilaian dengan rumus sebagai berikut:

$Y = \text{skor tertinggi likert (SB)} \times \text{jumlah panelis}$

$X = \text{skor terendah likert (SK)} \times \text{jumlah panelis}$

Setelah diketahui nilai X dan Y , untuk menghitung nilai index akan digunakan rumus:

$\text{index \%} = \text{nilai total skor} / Y \times 100\%$

Sebelum menyelesaikan perhitungan, harus dibuat interval dan interpretasi persen agar mengetahui penilaian dengan metode mencari interval skor persen. Rumus interval sendiri yaitu:

$I = 100 / \text{Jumlah skor (likert)}$, dalam kasus ini $= 100 / 5 = 20$

Berdasarkan interval yang sudah diketahui, maka kriteria interpretasi skornya didapatkan sebagai berikut:

- Angka 0% - 19,99% = Sangat Kurang (SK)
- Angka 20% - 39,99% = Kurang (K)
- Angka 40% - 59,99% = Cukup (C)
- Angka 60% - 79,99% = Bagus (B)
- Angka 80% - 100% = Sangat Bagus (SB)

Setelah diketahui rumus dan nilai-nilai yang dibutuhkan, selanjutnya adalah menghitung nilai jawaban per pertanyaan yang ada pada kuisisioner. Nilai yang didapat setelah perhitungan nanti akan berupa persentase yang akan disesuaikan

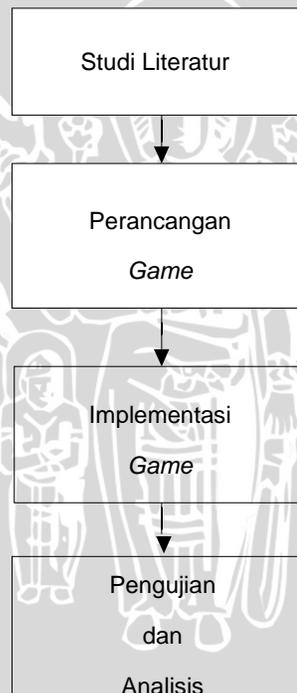
dengan interval kriteria interpretasi skor yang sudah dibuat untuk menarik kesimpulan jawaban dari para panelis.



BAB III

METODE PENELITIAN

Bab ini menjelaskan tentang langkah-langkah yang ditempuh dalam penelitian yang diusulkan. Metodologi yang dilakukan dalam penelitian ini dilakukan melalui beberapa tahapan, yakni studi literatur, perancangan *Virtual Reality First Person Military Simulation Game* dengan metode perancangan *paper prototyping*, implementasi, dan evaluasi. Gambar 3.1 merupakan diagram alir metode yang berisi tahapan-tahapan yang dilakukan dalam penelitian yang diusulkan.



Gambar 3.1 Blok Diagram Runtutan Pengerjaan Skripsi

3.1 Studi Literatur

Studi literatur merupakan kegiatan untuk mempelajari berbagai macam hal yang berkaitan dengan pembuatan *game* ini dalam bentuk referensi atau tulisan. Referensi dapat diambil dari sebuah buku, jurnal, laporan penelitian, maupun artikel yang berhubungan tentang hal-hal yang dapat menunjang pengembangan, sebagai berikut:

1. Perancangan formal elemen dalam *game*
2. *Paper prototyping*
3. *Virtual Reality* Google Cardboard
4. Pengujian *game*

3.2 Perancangan *Game*

Dalam perancangan *game* dilakukan tahap-tahap awal pengembangan dari *game* yang nantinya akan memudahkan implementasi *game*. Perancangan berisi tentang keseluruhan *game concept* dan perancangan teknis dari *game* itu sendiri. Bagian perancangan berisi hal-hal berikut, antara lain:

1. Deskripsi *Game*

Bagian ini membahas tentang semua deskripsi awal *game*. Bagaimana nantinya bentuk *game* yang diinginkan setelah dilakukan survey dan analisa, atau dapat dikatakan, gambaran dari ide-ide yang akan diimplementasikan dalam *game*. Deskripsi *game* meliputi judul *game*, *genre*, *goal*, *platform*, dan peringkat ESRB.

2. *Gameplay*

Dalam sebuah *game* tentu aka nada sistematika perhitungan di dalamnya. Dalam hal ini, desain teknis yang dimaksud merupakan perancangan karakter, kontrol pemain, sistematika *game* serta mekanik *game*.

3. *Paper Prototyping*

Pada sub bab ini, akan dijelaskan mengenai perancangan konsep *art* dari *game* yang dibuat terlebih dahulu di atas kertas. Proses ini dapat dilakukan secara berkelompok dengan melakukan *brainstorming* ide dari masing-masing anggota kelompok.

3.3 Implementasi *Game*

Tahapan implementasi merupakan tahapan pembuatan *game*, yaitu proses penerjemahan perancangan yang telah dibuat sebelumnya ke tampilan yang sebenarnya. Implementasi *game* akan dilakukan dengan menggunakan bahasa pemrograman C# dan Java dalam pembuatan *gameplay* dari *game*. Dalam implementasi VR HMD digunakan aplikasi TrinusVR untuk mengaplikasikan VR pada

device smartphone android yang akan dipasang pada Google Cardboard sebagai media *headmount*.

Pada tahap ini juga menggunakan sistem *Artificial Intelligence* (AI) yang sudah dikembangkan oleh anggota kelompok pengerjaan proyek yang nantinya akan diimplementasikan dalam bentuk *script* dan *library*. Salah satu *library* yang telah dibuat adalah *library* HPA yang mengatur *pathfinding* pada AI dengan menggunakan metode *Near-Optimal Hierarchical Pathfinding* (HPA*).

Bagian implementasi terdiri dari beberapa bagian seperti:

1. Implementasi *gameplay*
2. Implementasi VR HMD
3. Implementasi *art* dan *User Interface*

3.4 Pengujian dan Analisis

Akan dilakukan pengujian berdasarkan implementasi yang telah dibuat berdasarkan pengujian game secara umum. Penulis memilih 3 metode pengujian untuk menguji *game* yaitu metode *Combinatorial Testing*, *Test Flow Diagram* (TFD), dan *Play Testing*.

Skenario pengujian pertama, yaitu *Combinatorial Testing* dilakukan dengan melakukan pengecekan pada *game* dengan mengkombinasikan beberapa pasang parameter dengan parameter lain dalam *game* yang dapat dilakukan saat bermain *game*. Setelah terbentuk kombinasi-kombinasi parameter yang sudah dibuat, maka akan dilakukan pengujian terhadap masing-masing kombinasi pada *game* apakah dapat berjalan dengan baik atau tidak.

Untuk skenario pengujian *Test Flow Diagram*, akan dilakukan dengan membuat TFD dari alur *game* secara umum dan TFD dari *Weapon Ammo*. Setelah itu akan dibuat *data dictionary* dari TFD tersebut dan melakukan pengujian dengan menjalankan *flow* dari masing-masing TFD.

Dalam pengujian *Play Testing*, akan dilakukan terhadap sepuluh orang penguji, dimana sebagian diantaranya pernah dan mengetahui perkembangan *video games* dan sebagian lainnya merupakan orang yang dapat dikatakan awam dalam dunia *video games*. Sepuluh orang penguji tersebut nantinya akan dipersilahkan memainkan *game* dan setelah selesai memainkannya, mereka akan diberikan kuisisioner *feedback* untuk memberikan tanggapan mengenai *game*.

BAB IV

PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan *Game*

Setelah semua analisa kebutuhan dilaksanakan, hal berikutnya merupakan perancangan dari *game* yang ingin dikembangkan. Dalam perancangan ini dilakukan semua tahap-tahap awal pengembangan *game*. Sehingga dapat memudahkan implementasi *game* ini ke depannya. Perancangan berisi tentang keseluruhan *game concept* dan perancangan teknis dari *game* itu sendiri. Bagian perancangan dapat berisi hal-hal berikut, antara lain:

4.1.1 Deskripsi *Game*

Game ini dimainkan oleh satu orang pemain (*single player*). Pemain menggerakkan satu karakter polisi yang bertugas untuk menyelamatkan sandera dari para teroris. Pemain akan ditemani oleh 4 NPC polisi yang berfungsi membantu pemain dalam menyelesaikan misi.

1. Judul *Game*

Game ini ke depannya akan dinamakan "Save me Now!!" Dikarenakan gameplay dari *game* ini mengharuskan pemain untuk melakukan aksi penyelamatan sandera demi menyelesaikan misi utama dari *game*.

2. Genre dan Topik

Genre dari *game* ini adalah *First Person Shooter Simulation* dimana pemain akan menjalankan karakter dengan tampilan *First Person* (orang pertama) dalam sebuah *game* yang mensimulasikan situasi penyelamatan sandera oleh pasukan polisi.

3. Goal (Tujuan *Game*)

Tujuan dari *game* ini cukup sederhana, pemain ditugaskan untuk pergi menyelamatkan sandera dari tempat sandera disekap menuju ke titik aman sembari mengalahkan musuh yang tersebar di *map* permainan dengan cara menembaki musuh tersebut. Misi akan dianggap sukses jika pemain berhasil menyelamatkan dan membawa sandera ke titik aman.

4. Platform

Penggunaan *game* ini akan ditujukan pada perangkat PC yang diintegrasikan dengan *Smartphone* dengan sistem operasi Android sebagai media *display* VR yang dijalankan menggunakan VR HMD Google Cardboard.

5. ESRB Rating

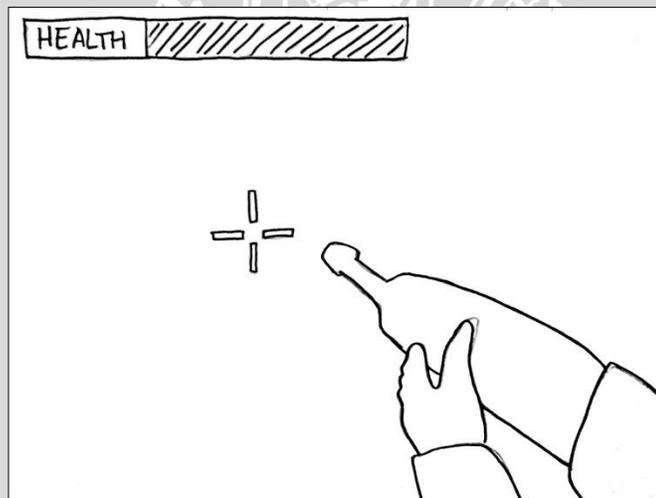
Game ini termasuk dalam peringkat Teen, karena konten dalam *game* ini mengandung sedikit unsur kekerasan seperti peristiwa baku tembak untuk membunuh lawan. Oleh karena itu target pengguna *game* ini sebaiknya merupakan kalangan remaja yang dapat memilah mana hal-hal yang baik dan buruk.

4.1.2 Gameplay

Bagian *gameplay* akan menjelaskan tatanan dan alur dari *game* ini. Mulai dari karakter, kontrol *game*, sistematika *game*, serta mekanik dalam *game*.

4.1.2.1 Karakter

Karakter yang akan dimainkan oleh pemain adalah seorang kepala regu dari tim polisi yang bertugas memimpin regunya dalam misi penyelamatan sandera bernama Sukemi. Dalam *game* ini, karakter nantinya hanya akan nampak bagian tangan dan senjatanya saja, hal ini dikarenakan genre *game First Person Shooter* yang menampilkan *game* dengan sudut pandang orang pertama. Dalam pandangan orang pertama, yang ditampilkan akan berupa seperti pada gambar 4.1.



Gambar 4.1 Tampilan pada Sudut Pandang Orang Pertama

4.1.2.2 Kontrol *Game*

Saat memainkan *game*, pemain dapat melakukan beberapa aksi pada karakter dengan menekan tombol sebagai berikut pada *joystick*:

Analog kiri ke atas : pergerakan maju ke depan.

Analog kiri ke kiri : pergerakan ke samping kiri.

Analog kiri ke bawah : pergerakan mundur ke belakang.

Analog kiri ke kanan : pergerakan ke samping kanan.

B : pergerakan melompat.

A : menembakkan peluru.

RB : membuat sandera mengikuti pemain setelah pemain berada dalam radius sandera.

Pemain dapat memberikan perintah kepada ketiga NPC (*Non Player Character*) untuk mengatur formasi dan taktik penyerangan dengan menekan tombol sebagai berikut pada *joystick*:

Klik Analog Kiri : memerintahkan pasukan melakukan HOLD, yaitu berhenti dan menahan posisi mereka dengan posisi mereka berada saat ini.

X : memerintahkan pasukan melakukan ATTACK1, yaitu maju bersama-sama menuju ke area serang 1 yang sudah ditentukan pada peta.

Y : memerintahkan pasukan melakukan ATTACK2, yaitu maju bersama-sama menuju ke area serang 2 yang sudah ditentukan pada peta.

LB : memerintahkan pasukan melakukan FOLLOW, yaitu maju ke arah kapten (karakter pemain) berada dan kemudian berhenti saat target sudah tercapai.

Klik Analog Kanan : memerintahkan pasukan melakukan RETREAT, yaitu pergi menuju area bertahan (area aman) yang sudah ditentukan pada peta.

4.1.2.3 Sistematika *Game*

Inti dari *game* "Save Me Now!!" adalah pemain menggerakkan karakter utama sebagai kapten dari regu polisi. Regu polisi yang ditugaskan pada misi ini terdiri dari 4 orang yaitu 1 kapten (digerakkan oleh pemain) dan 3 orang anggota. Pemain dapat memberikan 5 perintah (HOLD, ATTACK1, ATTACK2, FOLLOW ME, RETREAT) seperti yang sudah dijelaskan pada kontrol *game*, untuk menggerakkan 3 anggota polisi lain.

Pemain memiliki misi untuk menyelamatkan sandera dari area musuh untuk dibawa menuju tempat dimana mobil polisi berada. Dalam menjalankan misi,

pemain maupun anggota polisi lain dapat menembak musuh untuk memberikan *damage* dan membunuh musuh, dan sebaliknya musuh juga dapat menembak pemain untuk melakukan hal yang sama.

Misi akan dianggap sukses setelah pemain berhasil membawa sandera ke tempat dimana mobil polisi berada. Sedangkan misi akan dianggap gagal bila pemain mati dalam pertempuran.

4.1.2.4 Game Mechanics

“Save Me Now!!” merupakan game dengan genre FPS, oleh karena itu game akan ditampilkan dengan perspektif orang pertama. Seperti *game* dengan genre FPS pada umumnya, pemain dapat menjalankan karakter layaknya pemain menjadi karakter tersebut dengan kontrol yang sudah dijelaskan pada bagian kontrol *game*.

Karakter utama pada game memiliki sebuah senjata berupa senapan dengan tipe *Sub-Machine Gun* yang dapat menembakkan peluru secara *auto fire* (peluru dapat ditembakkan secara otomatis dengan menahan *left mouse button*). Peluru akan memberikan *damage* pada musuh yang dapat menyebabkan kematian pada musuh baik dari pihak polisi maupun pihak teroris.

Dalam *game*, akan terdapat *health bar* atau nyawa pemain. Pemain akan memasuki kondisi kalah jika poin *health bar* tersebut kosong akibat terlalu banyak menerima *damage* dari tembakan musuh. Poin awal untuk *health bar* pemain telah ditentukan yaitu 100 poin. Anggota regu polisi serta pasukan teroris musuh juga akan diberikan nyawa sebanyak 100 poin dan dapat terbunuh atau mati jika poin nyawa tersebut habis. Peluru dari masing-masing karakter pada *game* ditentukan memiliki *damage* per tembakan yang mengenai target karakter lawan.

Game akan diberikan 3 macam tingkat kesulitan yaitu *easy*, *normal*, dan *hard*. Tingkat kesulitan pada *game* akan diatur melalui berapa besar *damage* yang dihasilkan oleh tembakan lawan. Pada tingkat kesulitan *easy*, peluru lawan akan memberikan *damage* sebesar 10 poin pada nyawa polisi. Sedangkan pada tingkat kesulitan *normal*, *damage* yang diberikan adalah 15 poin per tembakan dan 20 poin per tembakan pada tingkat kesulitan *hard*.

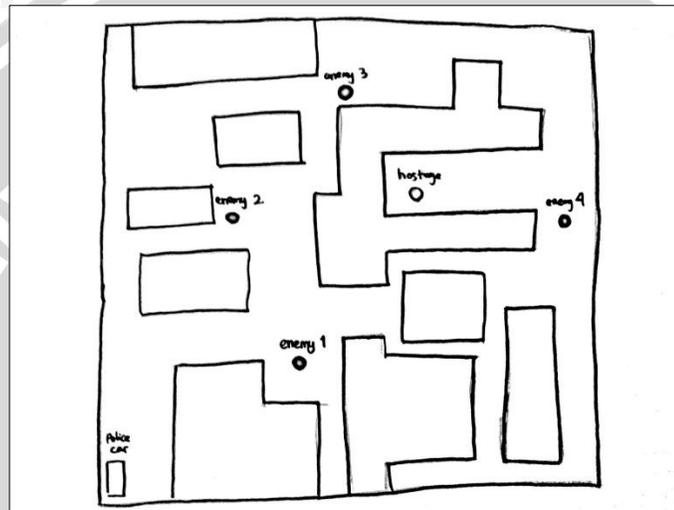
Pemain akan dapat menyelamatkan sandera hanya jika karakter pemain berada di dekat sandera dengan menekan tombol “e” pada *keyboard*. Pemain tidak dapat langsung membuat sandera mengikuti pemain jika karakter pemain berada di lokasi yang jauh dari sandera.

4.1.3 Paper Prototyping

Bagian *paper prototyping* menjelaskan mengenai tahap-tahap perancangan *game* dengan menggunakan metode *paper prototyping*.

4.1.3.1 Perancangan Level

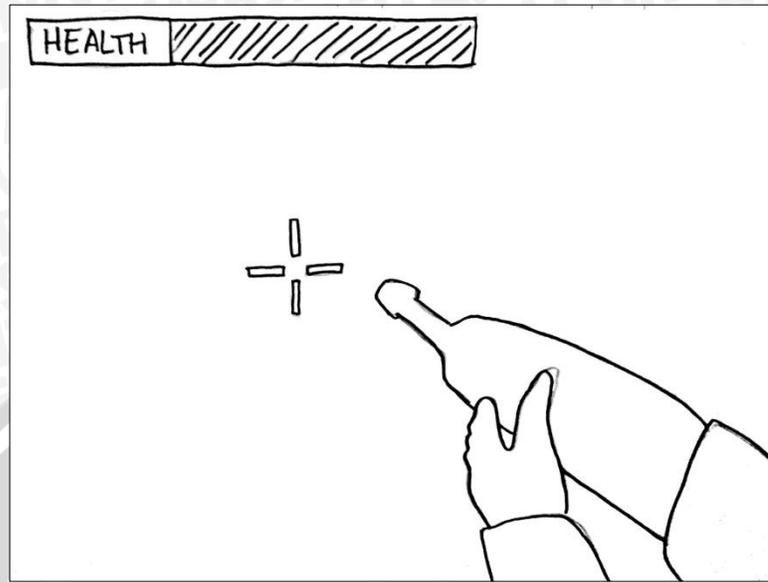
Permainan akan dimulai dengan pemain berada di posisi dimana mobil polisi berada (titik akhir penyelamatan sandera) dengan peta level yang dibentuk menjadi semacam ruangan labirin. Dalam ruangan terdapat 4 orang karakter musuh yang ditugaskan untuk berpatroli di sekitar ruangan untuk menjaga sandera. Terdapat juga mobil polisi di tempat awal karakter pemain memulai permainan. Perancangan bentuk ruangan, posisi karakter musuh, posisi mobil polisi serta posisi sandera dapat dilihat pada gambar 4.2.



Gambar 4.2 Denah rancangan level

4.1.3.2 Perancangan HUD

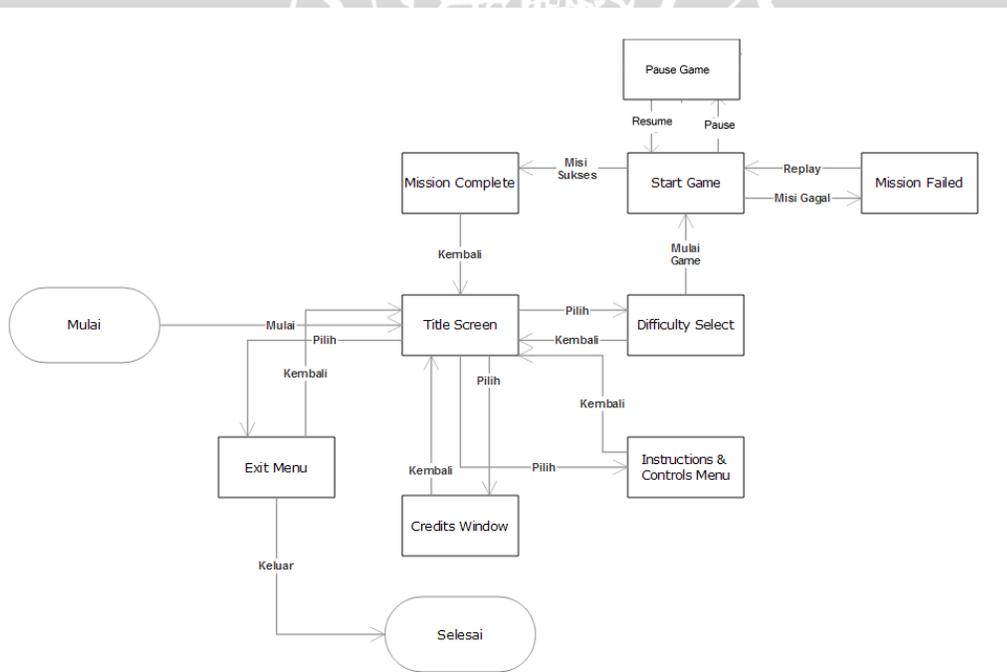
Dalam tampilan *game*, terdapat *Head-up Display* (HUD), dimana HUD terdiri atas *health bar* yang merupakan nyawa dari pemain yang terletak di bagian kiri atas tampilan *game*. Terdapat juga tampilan *crosshair* berbentuk simbol plus berwarna putih di bagian tengah layar sebagai penanda target tembakan. Tampilan HUD tersebut dirancang seperti pada gambar 4.3.



Gambar 4.3 Perancangan HUD pada Game.

4.1.3.3 Perancangan Game Screen Flow

Pada game “Save Me Now!!”, masing-masing dari game scene memiliki hubungan terhadap game scene yang lainnya seperti alur yang ditunjukkan pada gambar 4.4.



Gambar 4.4 Game Screen Flow Keseluruhan.

4.1.3.4 Perancangan GUI

Dalam memudahkan pemain untuk melakukan navigasi pada game, dibuat perancangan *Game User Interface* (GUI) untuk tiap *game scene*.

1. Title Screen

Pada *Title Screen* atau menu utama, terdapat beberapa pilihan tombol yang dapat dipilih oleh pemain untuk menuju *scene* lain, antara lain:

- START GAME : membuka jendela menu *difficulty*.
- INSTRUCTIONS : masuk ke *scene instructions and controls*.
- CREDITS : menampilkan jendela *credits*.
- QUIT : menampilkan jendela *exit menu*.

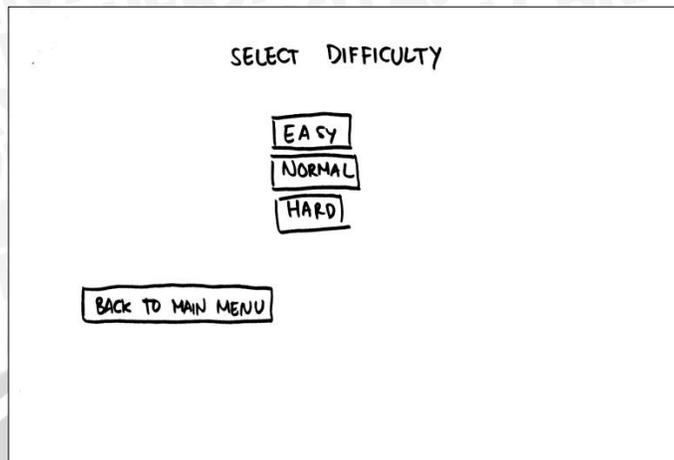
Tampilan *title screen* diilustrasikan pada gambar 4.5.



Gambar 4.5 Rancangan *Title Screen*.

2. Difficulty Select Screen

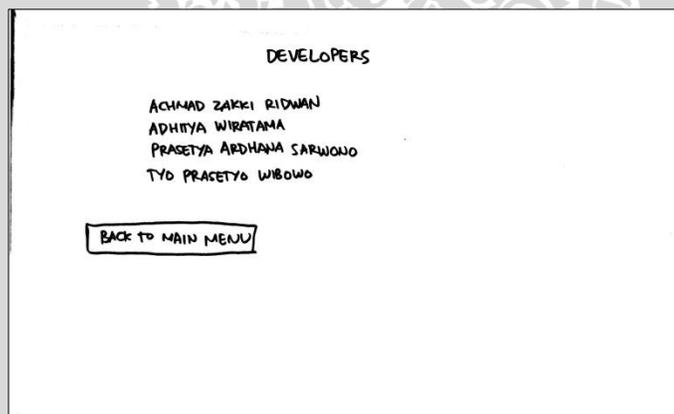
Dalam *Difficulty Select Screen* dapat ditemukan tombol yang dapat digunakan untuk mulai bermain sesuai dengan tingkat kesulitan yang diinginkan yaitu *easy*, *normal*, atau *hard*. Terdapat juga tombol untuk kembali ke menu pada bagian kiri bawah *scene*. Tampilan *difficulty select screen* ditunjukkan pada gambar 4.6.



Gambar 4.6 Rancangan *Difficulty Select Screen*

3. *Developers Screen*

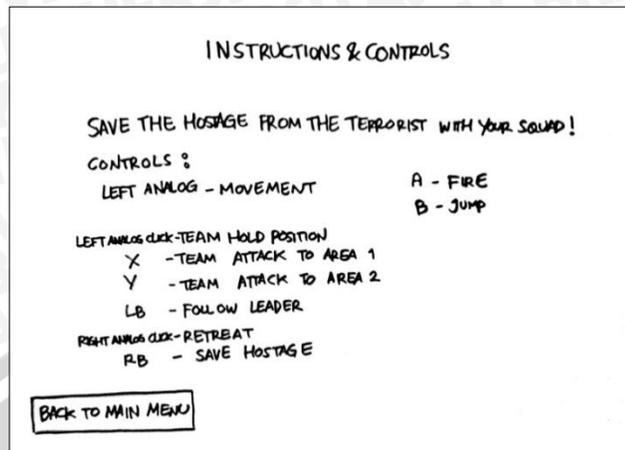
Developers Screen berisi tentang siapa saja *developer* yang terlibat pada pembuatan *game*. Pada *scene* ini terdapat sebuah tombol untuk kembali ke *title screen*. Tampilan rancangan *scene* ini diilustrasikan pada gambar 4.7.



Gambar 4.7 Rancangan *Developers Screen*

4. *Instructions and Controls Screen*

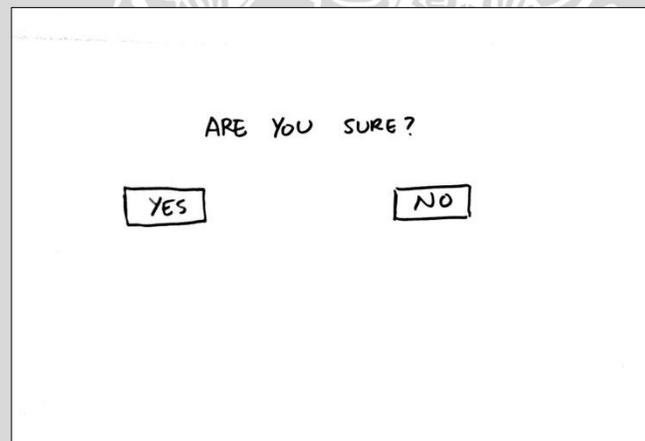
Instructions and Controls Screen menampilkan arahan untuk para pemain mengenai misi yang akan dijalankan pada *game* serta kontrol apa saja yang dapat digunakan oleh pemain saat memainkan *game*. Pada *scene* ini terdapat tombol *back to menu* yang dapat digunakan pemain untuk kembali ke *Title Screen*. Gambar 4.8 menampilkan rancangan dari *Instructions and Controls Screen*.



Gambar 4.8 Rancangan *Instructions and Controls Screen*.

5. Exit Menu Screen

Exit Menu Screen berisi menu yang menanyakan apakah pemain benar-benar ingin keluar dari permainan. Jika pemain memilih *yes*, maka permainan akan tertutup. Ketika pemain memilih *no*, pemain akan kembali ke *title screen*. Tampilan *exit menu screen* ditampilkan pada gambar 4.9.



Gambar 4.9 Rancangan *Exit Menu Screen*

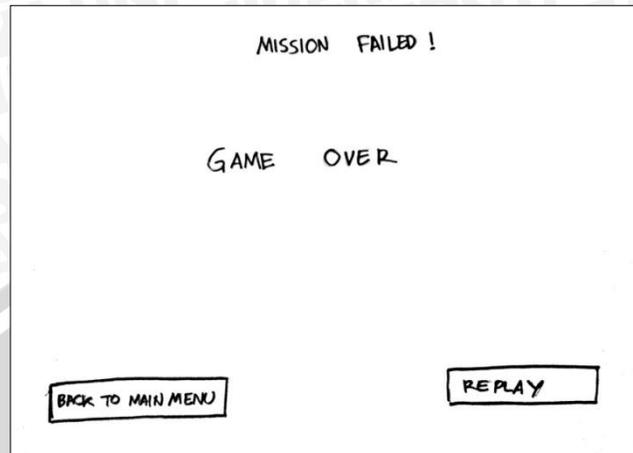
6. Mission Failed Screen

Mission Failed Screen berisi tentang informasi bahwa pemain gagal dalam menjalankan misi dan mengalami *game over*. Dalam *scene* ini terdapat dua tombol menu:

Back to Main Menu : kembali ke *Title Screen*.

Replay : mengulang level kembali.

Gambar 4.10 menunjukkan rancangan dari *Mission Failed Screen*.



Gambar 4.10 Rancangan *Mission Failed Screen*.

7. *Mission Complete Screen*

Mission Complete Screen menunjukkan bahwa pemain telah berhasil menyelesaikan misi membawa sandera ke tempat yang aman. Pada *scene* ini terdapat tombol *Back to Main Menu* yang berfungsi untuk kembali ke *Title Screen*. Gambar 4.11 merupakan sketsa rancangan *Mission Complete Screen*.

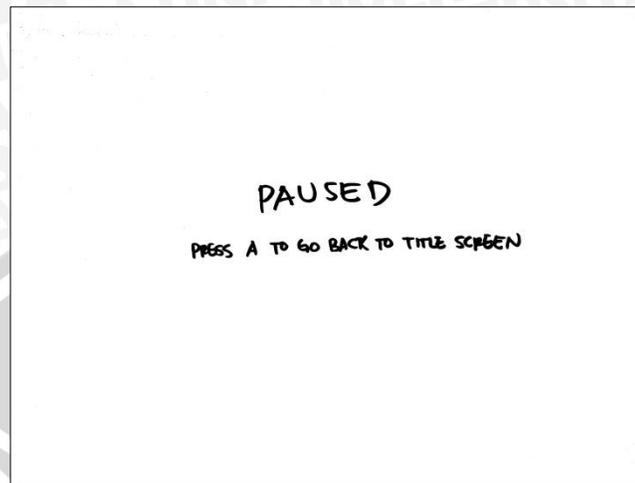


Gambar 4.11 Rancangan *Mission Complete Screen*.

8. *Pause Screen*

Pause Screen hanya akan muncul jika pemain menekan tombol *pause* saat permainan berlangsung. Pemain dapat melanjutkan permainan dengan menekan

tombol *pause* kembali atau kembali ke *title screen* dengan menekan tombol A pada *joystick*. Tampilan rancangan *pause screen* ditunjukkan pada gambar 4.12.



Gambar 4.12 Rancangan *Pause Screen*

4.2 Implementasi *Game*

Bagian ini akan menjelaskan bagaimana proses implementasi dari perancangan *game* yang telah dibuat pada sub bab sebelumnya, serta beberapa teknologi pendukung pengembangan *game*. Terdapat beberapa implementasi yang dilakukan dalam pengembangan *game* ini, diantaranya adalah implementasi *gameplay*, implementasi *art* dan *user interface*, serta implementasi ke dalam VR HMD.

4.2.1 Penentuan Spesifikasi Sistem

Dalam pengembangan implementasi *game* "Save Me Now!!" dibutuhkan spesifikasi sistem yang mendukung agar implementasi dapat berjalan dengan baik. Spesifikasi sistem untuk implementasi *game* dijelaskan pada spesifikasi perangkat keras dan perangkat lunak.

4.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang dipakai dalam proses pengembangan dijelaskan dalam tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
<i>Memory</i>	4.00 GB RAM
<i>Hardisk</i>	500GB
<i>Graphic Card</i>	NVIDIA GeForce GT 540M, 1GB

4.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang dipakai dalam proses pengembangan dijelaskan dalam tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
<i>Operating System</i>	Windows 7.0 64-bit
<i>DirectX Version</i>	DirectX 11
<i>Programing Language</i>	C#/Java
<i>Software Development Kit</i>	Unity Version 5.0.1f3

4.2.2 Implementasi *Gameplay*

Dalam pengimplementasian *gameplay* dari *game* "Save Me Now!", terdapat beberapa *script code* yang berperan penting dalam inti dari permainan. Diantaranya yaitu *script code* yang mengatur kontrol pemain, mekanik menembak, mekanik nyawa, situasi sandera, situasi menang, serta situasi kalah.

4.2.2.1 Implementasi Kontrol Pemain

Untuk menggerakkan karakter pemain, dibutuhkan *script code* yang ditunjukkan pada tabel 4.3.

Tabel 4.3 *Script Code* Kontrol Pemain

<i>Script code</i> untuk Kontrol Pemain	
<p><u>Deklarasi</u></p> <pre>public float speed = 6.0F; public float jumpSpeed = 8.0F; public float gravity = 20.0F; private Vector3 moveDirection = Vector3.zero;</pre>	
<pre>1 2 3 4 5 6 7 8 9 10 11 12</pre>	<pre>void Update() { CharacterController controller = GetComponent<CharacterController>(); if (controller.isGrounded) { moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical")); moveDirection = transform.TransformDirection(moveDirection); moveDirection *= speed; if (Input.GetButton("Jump")) moveDirection.y = jumpSpeed;} moveDirection.y -= gravity * Time.deltaTime; controller.Move(moveDirection * Time.deltaTime);</pre>

Pada deklarasi terdapat deklarasi variabel *speed* bertipe *float* yang mengatur kecepatan gerakan karakter dengan nilai 6.0, variabel *jumpSpeed* bertipe *float* yang mengatur kecepatan lompatan karakter dengan nilai 8.0, variabel *gravity* bertipe *float* yang nantinya akan digunakan untuk mengatur gravitasi dengan nilai 20.0, dan variabel *moveDirection* bertipe *Vector3* yang bernilai (0,0,0) dikarenakan variabel *moveDirection* dideklarasikan dengan nilai *Vector3.zero*.

Baris 1 merupakan pemanggilan metode *Update* yang selalu dijalankan tiap detik selama permainan dijalankan. Baris 2 merupakan metode pengambilan komponen dari *script* pada *class* *CharacterController*. Baris 3 merupakan logika jika *controller* dalam posisi *grounded* maka perintah-perintah dalam tanda kurung akan dilakukan. Baris 4-5 merupakan perintah untuk membaca *input* dari luar dan menggerakkan pemain dengan kecepatan yang sudah ditentukan. Baris 7-12 merupakan logika sederhana yang mengatur lompatan dari karakter jika terdapat *input* dari luar berupa tombol "jump" yang ditekan.

4.2.2.2 Implementasi Mekanik Menembak

Berikut *script* yang mengatur mekanik menembak pada *game* untuk karakter utama ditunjukkan pada tabel 4.4.

Tabel 4.4 Script Code Mekanik Menembak

Script code untuk Mekanik Menembak	
Deklarasi	
<pre> var Bullet : Rigidbody; var Spawn : Transform; var BulletSpeed : float = 1000; var ReloadTime : float = 2; var AmmoInMag : float = 30; var IsFullAuto = true; static var AmmoLeft : float; static var IsReloading = false; private var CanFire = true; var FireRate = 0.1; </pre>	
1	<code>function Start() {</code>
2	<code> AmmoLeft = AmmoInMag;</code>
3	<code>}</code>
4	<code>function Update () {</code>
5	<code> if(IsFullAuto == false){</code>
6	<code> if(!Input.GetButtonDown("Fire2")){</code>
7	<code> if(Input.GetButtonDown("Fire1")){</code>
8	<code> if(AmmoLeft > 0){</code>
9	<code> Fire();</code>
10	<code> }</code>
11	<code> }</code>
12	<code> }</code>
13	<code>}</code>
14	<code>else {</code>
15	<code> if(!Input.GetButtonDown("Fire2")){</code>
16	<code> if(Input.GetButtonDown("Fire1")){</code>
17	<code> if(AmmoLeft > 0){</code>
18	<code> Fire();</code>
19	<code> }</code>
20	<code> }</code>
21	<code>}</code>
22	<code>}</code>
23	<code>if(AmmoLeft == 0){</code>
24	<code> Reload();</code>
25	<code>}</code>
26	<code></code>
27	<code> if(AmmoLeft < 0){</code>
28	<code> AmmoLeft = 0;</code>
29	<code> }</code>
30	<code>}</code>
31	<code>function Fire() {</code>
32	<code> if(CanFire == true && IsReloading == false){</code>
33	<code> var bullet1 : Rigidbody =</code>
34	<code> Instantiate(Bullet, Spawn.position, Spawn.rotation);</code>
35	<code> bullet1.AddForce(transform.forward * BulletSpeed);</code>

Tabel 4.4 (Lanjutan)

```

36     CanFire = false;
37     BroadcastMessage("FireAnima");
38     yield WaitForSeconds(FireRate);
39     CanFire = true;
40     AmmoLeft -= 1;
41     GetComponent.<AudioSource>().Play();
42 }
43 }
44 function Reload(){
45     CanFire = false;
46     IsReloading = true;
47     BroadcastMessage("ReloadAnim");
48     yield WaitForSeconds(ReloadTime);
49     IsReloading = false;
50     CanFire = true;
51     AmmoLeft = AmmoInMag;
52 }

```

Pada deklarasi terdapat deklarasi variabel *Bullet* dengan tipe *Rigidbody* yang nantinya akan diisi dengan objek peluru dan variabel *Spawn* dengan tipe *transform* yang nantinya akan menjadi sumber munculnya peluru. Deklarasi variabel *BulletSpeed* dan variabel *FireRate* adalah deklarasi yang menentukan kecepatan peluru dan kecepatan tembakan. Sedangkan deklarasi variabel *AmmoInMag* dan variabel *AmmoLeft* merupakan deklarasi yang menentukan jumlah peluru pada senapan dan deklarasi variabel *ReloadTime* yang nantinya menentukan berapa lama yang dibutuhkan untuk mengisi kembali peluru pada senapan.

Baris 1 sampai 3 merupakan pemanggilan fungsi *start* yang dijalankan pada awal permainan dimulai dan berisi logika yang mengatur jumlah peluru saat ini. Baris 4 sampai 30 berisi fungsi *update* yang berisi logika yang akan memanggil fungsi *fire* yang berisi mekanik menembak saat pemain meng-klik *left mouse button* seperti yang tertulis pada *code* `if(Input.GetButtonDown("Fire1"))` dan logika yang mengatur kemampuan senjata untuk menembakkan peluru secara otomatis jika pemain menahan *left mouse button* saat menembak, serta logika yang akan memanggil fungsi *reload* yang akan membuat peluru senapan terisi secara otomatis saat jumlah peluru tersebut habis atau bernilai 0.

Baris 31 sampai 42 berisi fungsi *fire* yang mengatur mekanik menembak pada karakter utama. Fungsi akan memunculkan peluru dengan *code* `var bullet1 : Rigidbody = Instantiate (Bullet,Spawn.position,Spawn.rotation);` dan *code* `bullet1.AddForce(transform.forward * BulletSpeed);` akan memberikan gaya pada peluru agar dapat melaju ke depan layaknya peluru yang ditembakkan. Kemudian *code* `BroadcastMessage("FireAnima");` akan memanggil animasi menembak dan

jumlah peluru akan berkurang 1 setiap fungsi ini berjalan seperti yang tertulis pada *code* AmmoLeft -= 1;

Baris 43 sampai 52 berisi fungsi reload yang mengatur mekanik pengisian peluru pada senapan saat jumlah peluru mencapai 0 dan membuat jumlah peluru ke jumlah awal yaitu 30 seperti yang sudah dideklarasikan pada awal *script code* dengan *code* AmmoLeft = AmmoInMag;

4.2.2.3 Implementasi Situasi Sandera

Misi utama dari "Save Me Now!!" adalah menyelamatkan sandera. Berikut *script code* yang mengatur mekanik situasi dari sandera tersebut pada tabel 4.5.

Tabel 4.5 Script Code Situasi Sandera

Script code untuk Situasi Sandera	
Deklarasi float nRadius = 5f; public bool DrawGizmos = false; public Transform player; public Canvas pressMenu; public enum State { SaveClick, Save, Null }; State state = State.Null;	
1	void Start ()
2	{
3	pressMenu.enabled = false;
4	}
5	
6	void Update () {
7	handleInput ();
8	float distance = Vector3.Distance
9	(player.transform.position, transform.position);
10	if (distance < nRadius) {
11	Debug.Log ("in Radius");
12	state_ = State.SaveClick;
13	}
14	if (distance > nRadius) {
15	Debug.Log ("out of Radius");
16	state_ = State.Save;
17	}
18	}
19	public void handleInput () {
20	switch (state_) {
21	case State.SaveClick:
22	Debug.Log ("Save Click State");
23	pressMenu.enabled = true;
24	if (Input.GetButtonDown ("Save")) {
25	Debug.Log ("E KLIK");
26	state = State.Save;

Tabel 4.5 (Lanjutan)

```

27         Follow86.seek86 = true; }
28         break;
29     case State.Save:
30         Debug.Log ("in Save");
31         pressMenu.enabled = false;
32         break;
33     case State.Null:
34         Debug.Log ("nul klik");
35         break;
36     }
37 }
38 void OnDrawGizmos ()
39 {
40     if (DrawGizmos) {
41         Gizmos.color = Color.green;
42         Gizmos.DrawWireSphere (transform.position, nRadius);
43     }
44 }
45 }
    
```

Pada fungsi start berisi pernyataan bahwa variabel pressMenu bersifat *false*. Sedangkan fungsi update berisi logika yang mengatur tentang posisi apakah pemain berada dalam radius sandera yang dapat dilihat pada baris 8 sampai baris 15. Baris 18 sampai 36 berisi fungsi handleInput yang mengatur mekanik penyelamatan sandera jika ditekan tombol “Save” maka sandera akan mengikuti pemain. Fungsi onDrawGizmos pada baris 37 sampai 45 mengatur penampakan radius penyelamatan sandera.

4.2.2.4 Implementasi Mekanik Nyawa

Script code yang mengatur mekanik nyawa dari karakter pemain ditunjukkan pada tabel 4.6.

Tabel 4.6 Script Code Mekanik Nyawa

Script code untuk Mekanik Nyawa	
Deklarasi public float health;	
1	void Start ()
2	{
3	health = 100;
4	}
5	void Update ()
6	{
7	public void OnTriggerEnter (Collider other)

Tabel 4.6 (Lanjutan)

8	{
9	if (other.GetComponent<MoveBullet> ())
10	health -= 10;
11	}
12	}

Deklarasi berisi variabel *health* dengan tipe *float* yang bersifat *public* yang nantinya akan diisi nilai untuk jumlah *health* maksimal dari karakter utama. Baris 1 sampai 4 berisi fungsi *start* yang berisi *code* *health = 100;* yang berarti jumlah nyawa maksimal dari karakter pemain adalah 100. Baris 5 sampai 12 berisi fungsi *update* yang diisi dengan logika sederhana yang mengatur jika *collider* dari karakter utama berbenturan dengan *collider* dari peluru maka nyawa akan berkurang sejumlah 10 poin.

4.2.2.5 Implementasi Situasi Menang

Kondisi untuk situasi menang adalah jika sandera mencapai radius dari tempat mobil polisi berada. Tabel 4.7 berisi *script code* yang mengatur kondisi tersebut.

Tabel 4.7 Script Code Situasi Menang

Script code untuk Situasi Menang	
Deklarasi float nRadius = 10f; public bool DrawGizmos = false; public Transform player;	
1	void Update ()
2	{
3	float distance = Vector3.Distance
4	(player.transform.position, transform.position);
5	if (distance < nRadius) {
6	Debug.Log ("in Radius finish");
7	Application.LoadLevel (4);
8	}
9	}
10	void OnDrawGizmos ()
11	{
12	if (DrawGizmos) {
13	Gizmos.color = Color.green;
14	Gizmos.DrawWireSphere (transform.position, nRadius);}
15	}

Pada deklarasi terdapat variabel `nRadius` dengan tipe `float` bernilai 10 dan objek `player` dengan tipe `transform` yang nantinya akan diisi dengan objek sandera. Terdapat juga deklarasi `bool DrawGizmos` yang bernilai `false`. Fungsi `update` berisi logika yang menghitung jarak dari objek `player` terhadap `nRadius` dan jika jarak objek `player` kurang dari `nRadius`, maka akan dilakukan `load level 4` yang berisi `scene Mission Complete`. Baris 9 sampai 15 berisi fungsi `OnDrawGizmos` yang menunjukkan bentuk dan radius dari `nRadius` pada `scene` perancangan `game`.

4.2.2.6 Implementasi Situasi Kalah

Pemain dinyatakan kalah jika nyawa karakter mencapai nilai 0. Pada tabel 4.8 dituliskan `script code` yang mengatur kondisi kalah pemain.

Tabel 4.8 Script Code Situasi Kalah

Script code untuk Situasi Menang	
Deklarasi <pre>public float health;</pre>	
1	<pre>void Update ()</pre>
2	<pre>{</pre>
3	<pre> if (health <= 0) {</pre>
4	<pre> Application.LoadLevel (3);</pre>
5	<pre> }</pre>
6	<pre>}</pre>

Deklarasi berisi variabel `health` dengan tipe `float`. Pada baris 1 sampai 6 terdapat fungsi `update` yang berisi logika jika `health` lebih kecil atau sama dengan 0, maka akan dilakukan `load level 3` yang berisi `scene Mission Failed`.

4.2.3 Implementasi VR

Dalam pengembangan `game "Save Me Now!!"` menggunakan `Virtual Reality Head Mounted Device` dari `Google Cardboard`, implementasi akan memanfaatkan `software TrinusVR` untuk mengkonversikan `game` dalam mode `virtual reality`. Langkah awal implementasi adalah dengan melakukan instalasi `software TrinusVR` pada perangkat komputer dan perangkat `smartphone`. Setelah instalasi selesai, koneksikan kedua perangkat tersebut melalui kabel atau `wireless connection`. Kemudian jalankan aplikasi `TrinusVR` pada kedua perangkat, lalu jalankan `game` pada perangkat komputer, maka pada perangkat `smartphone` akan muncul tampilan `game` seperti pada gambar 4.13.



Gambar 4.13 Tampilan Implementasi VR

4.2.4 Implementasi Art dan User Interface

Hasil perancangan level, HUD, dan GUI yang telah dibuat pada bagian perancangan sebelumnya diberikan implementasi *art* dan *user interface* untuk membuat *game* lebih menarik. Bagian ini akan menjelaskan tentang implementasi *art* dan *user interface*.

4.2.4.1 Implementasi Level

Implementasi level untuk *game* “Save Me Now!!” dirancang dalam Unity3D pada *editor scene view*. Objek-objek yang akan digunakan antara lain adalah *plane* dan *cube*. *Plane* akan digunakan sebagai lantai dan *Cube* akan digunakan sebagai alas an statis. Hasil implementasi dari peta level ditunjukkan pada gambar 4.14.



Gambar 4.14 Implementasi Level

4.2.4.2 Implementasi HUD

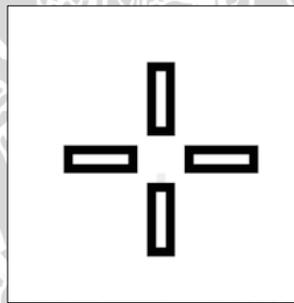
Beberapa HUD yang sudah dirancang dan dibuat konsepnya sebelumnya diimplementasikan secara keseluruhan pada *game* dengan deskripsi sebagai berikut.

Health bar : berbentuk persegi panjang berwarna hijau yang dapat berkurang ukurannya jika nyawa karakter pemain berkurang. Persegi hijau tersebut berada di dalam kotak hitam yang berfungsi sebagai wadah yang diilustrasikan pada gambar 4.15.



Gambar 4.15 Art Health Bar

Crosshair : Berbentuk tanda plus berwarna putih di bagian tengah layar yang berfungsi sebagai penanda target tembakan. *Art crosshair* ditunjukkan pada gambar 4.16.



Gambar 4.16 Art Crosshair

4.2.4.3 Implementasi GUI

Game User Interface yang berupa *game screen*, terdiri dari beberapa *scene* yang memiliki tujuan saling mengarahkan. Beberapa *scene* tersebut diantaranya adalah sebagai berikut. Terdapat sedikit perbedaan pada *paper prototyping* dengan hasil implementasi. Perbedaan tersebut terletak pada menu *difficulty*, *exit*, dan *credits* yang sebelumnya dirancang sebagai jendela *pop up* namun diimplementasikan dalam bentuk *scene*.

1. Title Screen

Tampilan *title screen* sama seperti yang telah dirancang pada *paper prototyping*. Terdapat 4 tombol yang dapat digunakan pemain untuk menavigasi menuju *scene* yang lain. Tombol A untuk memilih START GAME dan membuka *difficulty select screen* untuk memulai game. Tombol B untuk memilih INSTRUCTIONS untuk menuju *scene Instructions and Controls*. Tombol X untuk memilih CREDITS dan membuka *developers screen*. Kemudian tombol Y untuk

memilih QUIT dan membuka *exit menu screen* untuk keluar dari *game*. Implementasi *art* dari *title screen* dapat dilihat pada gambar 4.17.



Gambar 4.17 Title Screen

2. Difficulty Select Screen

Pada *scene* ini, pemain dapat memilih tingkat kesulitan permainan yang ingin dimainkan. Terdapat tiga pilihan level yaitu *easy* yang dapat dipilih dengan menekan tombol A pada *joystick*, *normal* yang dipilih dengan tombol X, dan *hard* yang dipilih dengan menekan tombol Y. Terdapat pula menu untuk kembali ke *title screen* dengan cara menekan tombol B pada *joystick*. Tampilan *difficulty select screen* dapat dilihat pada gambar 4.18.



Gambar 4.18 Difficulty Select Screen

3. Developers Screen

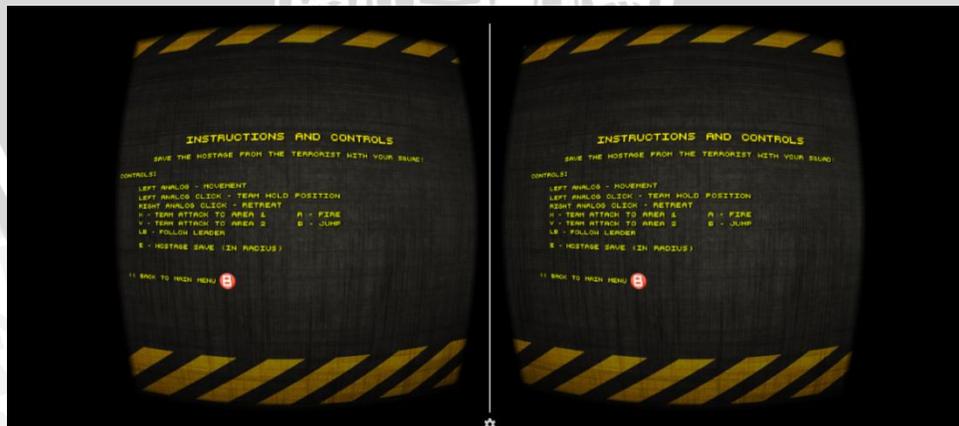
Developers screen menampilkan nama-nama pengembang yang terlibat dalam pembuatan *game* "Save Me Now!". Terdapat satu menu tombol untuk kembali ke *title screen* dengan menekan tombol B pada *joystick*. Tampilan *developers screen* diilustrasikan pada gambar 4.19.



Gambar 4.19 Developers Screen

4. Instructions and Controls Screen

Instructions and Controls Screen diimplementasikan sedikit lebih sederhana dibandingkan pada rancangan *paper prototyping*. Hal ini disebabkan untuk meminimalkan desain agar instruksi lebih mudah dibaca oleh pemain seperti ditunjukkan pada gambar 4.20. Pada *scene* ini terdapat satu menu tombol untuk kembali ke *title screen* dengan menekan tombol B pada *joystick*.



Gambar 4.20 Instructions and Controls Screen

5. Exit Menu Screen

Dalam *exit menu screen* terdapat menu yang dapat dipilih oleh pemain untuk keluar dari permainan. Pemain dapat menekan tombol A pada *joystick* untuk keluar dari permainan dan tombol B untuk kembali ke *title screen*. Tampilan *exit menu screen* dapat dilihat pada gambar 4.21.



Gambar 4.21 Exit Menu Screen

6. Mission Failed Screen

Implementasi *mission failed screen* dibuat sama seperti pada perancangan *paper prototyping*. Terdapat dua tombol pada *scene* ini yaitu tombol *Back to Main Menu* yang dapat dipilih dengan menekan tombol B pada *joystick* dan tombol A untuk memilih *Replay*. Implementasi *scene* ini diilustrasikan pada gambar 4.22.



Gambar 4.22 Mission Failed Screen

7. Mission Complete Screen

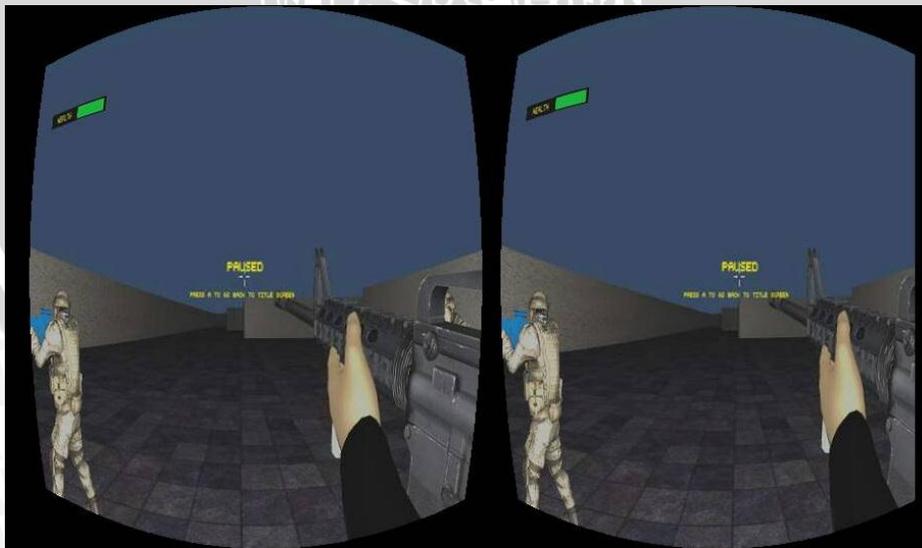
Pada *mission complete screen* terdapat satu tombol *Back to Main Menu* untuk kembali menuju *title screen*. Tampilan implementasi *art mission complete screen* dapat dilihat pada gambar 4.23.



Gambar 4.23 Mission Complete Screen

8. Pause Screen

Pause Screen hanya akan muncul jika pemain menekan tombol *pause* saat permainan berlangsung. Pemain dapat melanjutkan permainan dengan menekan tombol *pause* kembali atau kembali ke *title screen* dengan menekan tombol A pada *joystick*. Tampilan rancangan *pause screen* ditunjukkan pada gambar 4.24.



Gambar 4.24 Pause Screen

4.2.4.4 Implementasi Karakter

Implementasi *art* pada karakter pemain, anggota regu polisi, tentara musuh, dan sandera merupakan elemen penting dalam *game*. Sub bab ini akan menampilkan implementasi *art* dari karakter-karakter tersebut.

1. Karakter Pemain

Karakter pemain hanya ditampilkan bagian tangan dan senjatanya saja karena *game* akan ditampilkan dengan sudut pandang orang pertama. Tampilan karakter pemain ditunjukkan pada gambar 4.25. Penulis mengambil model karakter pemain dari *Asset Store Unity* dengan *package modern weapon pack*.



Gambar 4.25 Tampilan Karakter Pemain

2. Anggota Regu Polisi

Tampilan anggota regu polisi ditampilkan layaknya anggota polisi pada umumnya dengan seragam berwarna putih dan senapan berwarna biru seperti yang ditunjukkan pada gambar 4.26. Model dari polisi diambil dari *Soldier package* yang diunduh dari *Asset Store Unity*.



Gambar 4.26 Tampilan Anggota Regu Polisi

3. Tentara Musuh

Tentara musuh dirancang seperti tentara polisi namun menggunakan seragam berwarna hitam dan senapan berwarna hitam sebagai pembeda dengan anggota regu polisi. Gambar 4.27 menampilkan implementasi *art* tentara musuh. Model dari tentara musuh diunduh dari *Asset Store Unity* dan berada dalam satu *package* dengan model polisi yaitu *Soldier package*.



Gambar 4.27 Tampilan Tentara Musuh

4. Sandera

Sandera pada *game* “Save Me Now!!” ditampilkan sebagai anak perempuan yang bisa dilihat pada gambar 4.28. Model dari sandera diambil dari *Asset Store Unity* yaitu *Sporty Girl package*.



Gambar 4.28 Tampilan Sandera

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini dilakukan proses pengujian terhadap implementasi *Virtual Reality First Person Military Simulation Game "Save Me Now!!"*. Pengujian dilakukan dengan menguji tiap skenario dan pencatatan hasil berdasarkan parameter pengujian.

5.1 Combinatorial Testing

Skenario pengujian pertama, yaitu *Combinatorial Testing* dilakukan dengan melakukan pengecekan pada *game* dengan mengkombinasikan beberapa pasang parameter dengan parameter lain dalam *game* yang dapat dilakukan saat bermain *game*. Akan digunakan 3 macam parameter dalam permainan, yaitu parameter jumlah anggota polisi, perintah yang digunakan, dan parameter apakah sandera mengikuti pemain atau tidak. Pengujian ini dilakukan untuk mengetahui apakah kombinasi-kombinasi parameter yang sudah dibuat pada *game* dapat berjalan dengan baik atau tidak.

Tabel 5.1 Menunjukkan hasil kombinasi 3 parameter dalam game yaitu parameter jumlah anggota polisi, perintah yang digunakan, dan parameter apakah sandera mengikuti pemain atau tidak dan hasil uji dari kombinasi parameter tersebut.

Tabel 5.1 Hasil Kombinasi Parameter

Sandera Mengikuti Pemain	Jumlah Anggota Polisi	Perintah	Hasil Uji
Ya	3	Hold	Valid
Ya	3	Attack1	Valid
Ya	3	Attack2	Valid
Ya	3	Follow	Valid
Ya	3	Retreat	Valid
Ya	2	Hold	Valid
Ya	2	Attack1	Valid
Ya	2	Attack2	Valid
Ya	2	Follow	Valid
Ya	2	Retreat	Valid
Ya	1	Hold	Valid
Ya	1	Attack1	Valid
Ya	1	Attack2	Valid

Tabel 5.1 (Lanjutan)

Ya	1	Follow	Valid
Ya	1	Retreat	Valid
Ya	0	Hold	Valid
Ya	0	Attack1	Valid
Tidak	3	Hold	Valid
Tidak	3	Attack1	Valid
Tidak	3	Attack2	Valid
Tidak	3	Follow	Valid
Tidak	3	Retreat	Valid
Tidak	2	Hold	Valid
Tidak	2	Attack1	Valid
Tidak	2	Attack2	Valid
Tidak	2	Follow	Valid
Tidak	2	Retreat	Valid
Tidak	1	Hold	Valid
Tidak	1	Attack1	Valid
Tidak	1	Attack2	Valid
Tidak	1	Follow	Valid
Tidak	1	Retreat	Valid
Tidak	0	Hold	Valid
Tidak	0	Attack1	Valid
Tidak	0	Attack2	Valid
Tidak	0	Follow	Valid
Tidak	0	Retreat	Valid

Setelah diketahui hasil kombinasi dari parameter yang telah ditentukan, pengujian dilakukan dengan mencoba kombinasi parameter pada *game*.

Setelah melakukan pengujian dari hasil-hasil kombinasi yang dibuat pada *combinatorial testing*, ditemukan bahwa *game* sudah dapat berjalan dengan baik. Hal ini dibuktikan dengan berhasilnya semua skenario kombinasi dari parameter yang sudah ditentukan. Kombinasi dari parameter jumlah anggota polisi, perintah yang digunakan, dan parameter apakah sandera mengikuti pemain atau tidak secara keseluruhan telah berhasil dijalankan pada *game* dengan baik dan sesuai dengan bagaimana seharusnya.

Tabel 5.2 Data Dictionary TFD Game Flow

Events/Actions	Deskripsi
Enter	Mulai menjalankan "Save Me Now!!"
GoTitleScreen	Masuk ke <i>Title Screen</i>
Click Credits	Meng-klik tombol <i>Credits</i>
OpenCreditsWindow	Mengecek apakah jendela <i>Credits</i> terbuka.
Click Back to Menu	Meng-klik tombol <i>Back to Menu</i>
Click Quit	Meng-klik tombol <i>Quit</i>
OpenQuitWindow	Mengecek apakah jendela <i>Quit</i> terbuka
Click Start Game	Meng-klik tombol <i>Start Game</i>
OpenDifficultySelect	Mengecek apakah jendela <i>Difficulty Select</i> Terbuka
Select Easy	Memilih <i>easy</i> pada jendela <i>Difficulty Select</i>
StartLevelEasy	Memulai permainan dengan level <i>easy</i>
Select Normal	Memilih <i>normal</i> pada jendela <i>Difficulty Select</i>
StartLevelNormal	Memulai permainan dengan level <i>normal</i>
Select Hard	Memilih <i>hard</i> pada jendela <i>Difficulty Select</i>
StartLevelHard	Memulai permainan dengan level <i>hard</i>
Health = 0	Kondisi nyawa pemain mencapai nilai 0
GoMissionFailed	Masuk menuju <i>Mission Failed Screen</i>
Click Replay	Meng-klik tombol <i>Replay</i>
ReplayLevel	Mengulang level kembali
Hostage Saved	Kondisi dimana sandera berhasil diselamatkan
GoMissionComplete	Masuk menuju <i>Mission Complete Screen</i>
Click Pause	Meng-klik tombol pause
PauseGame	Permainan berhenti sementara
Click Unpause	Meng-klik tombol unpause
ContinueGame	Melanjutkan permainan
Exit	Keluar dari permainan

Minimum Path Generation yang dihasilkan:

1,2,3,4,5,6,7,6,8,24,27,12,13,18,21,6,9,23,26,14,15,19,21,6,10,22,25,16,17,20,21,4,29

Eksekusi *Minimum Path Generation* dilakukan setelah ditentukan sesuai dengan path tersebut dengan mengecek *data dictionary* tiap *flow* ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Uji Flow TFD Game Flow

Flow	Event/Action	Hasil yang diharapkan	Hasil yang didapatkan
1	Enter / GoTitleScreen	Mulai menjalankan SaveMeNow.exe dan membuka <i>title screen</i>	Mulai menjalankan SaveMeNow.exe dan membuka <i>title screen</i>
2	Click Credits / OpenCreditsWindow	Meng-klik tombol <i>Credits</i> dan masuk ke <i>credits screen</i>	Meng-klik tombol <i>Credits</i> dan masuk ke <i>credits screen</i>
3	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
4	Click Quit / OpenQuitWindow	Meng-klik tombol <i>Quit</i> dan masuk ke <i>quit menu screen</i>	Meng-klik tombol <i>Quit</i> dan masuk ke <i>quit menu screen</i>
5	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
6	Click Start Game / OpenDifficultySelect	Meng-klik tombol <i>Start Game</i> dan masuk ke <i>difficulty select screen</i>	Meng-klik tombol <i>Start Game</i> dan masuk ke <i>difficulty select screen</i>
7	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
8	Select Easy / StartLevelEasy	Memilih <i>difficulty easy</i> kemudian memulai <i>game</i> dengan level <i>easy</i>	Memilih <i>difficulty easy</i> kemudian memulai <i>game</i> dengan level <i>easy</i>
9	Select Normal / StartLevelNormal	Memilih <i>difficulty normal</i> kemudian memulai <i>game</i> dengan level <i>normal</i>	Memilih <i>difficulty normal</i> kemudian memulai <i>game</i> dengan level <i>normal</i>
10	Select Hard / StartLevelHard	Memilih <i>difficulty hard</i> kemudian memulai <i>game</i> dengan level <i>hard</i>	Memilih <i>difficulty hard</i> kemudian memulai <i>game</i> dengan level <i>hard</i>
11	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
12	Health = 0 / GoMissionFailed	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>

Tabel 5.3 (Lanjutan)

13	Click Replay / ReplayLevel	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama
14	Health = 0 / GoMissionFailed	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>
15	Click Replay / ReplayLevel	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama
16	Health = 0 / GoMissionFailed	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>	Nyawa pemain mencapai nilai 0 dan membuka <i>mission failed screen</i>
17	Click Replay / ReplayLevel	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama	Meng-klik tombol <i>Replay</i> dan mengulang permainan dengan tingkat kesulitan yang sama
18	Hostage Saved / GoMissionComplete	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>
19	Hostage Saved / GoMissionComplete	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>
20	Hostage Saved / GoMissionComplete	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>	Sandera berhasil diselamatkan dan masuk ke <i>mission complete screen</i>
21	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
22	Click Pause / PauseGame	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara
23	Click Pause / PauseGame	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara
24	Click Pause / PauseGame	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara	Meng-klik tombol <i>Pause</i> dan permainan berhenti sementara

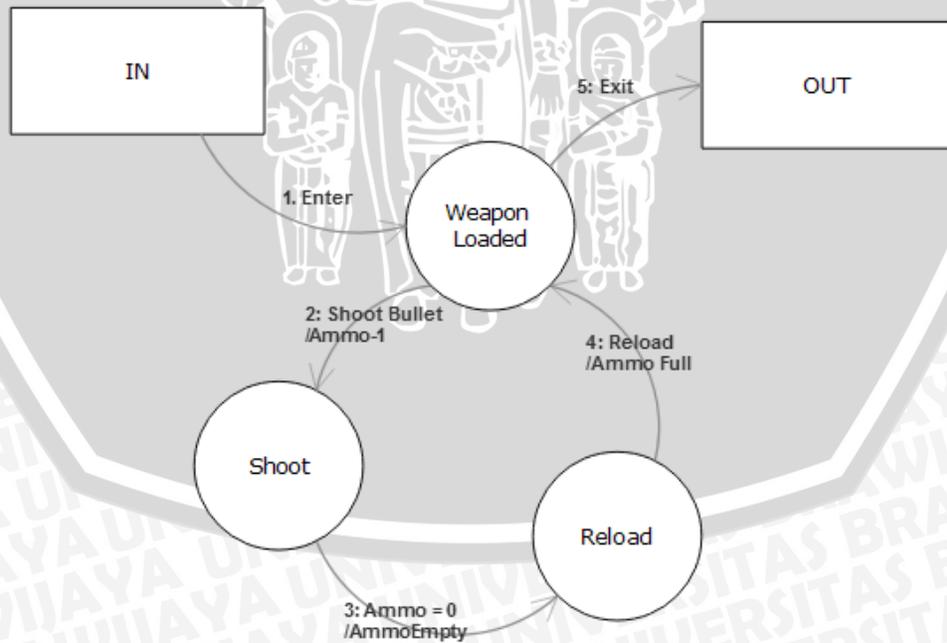
Tabel 5.3 (Lanjutan)

25	Click Unpause / ContinueGame	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan
26	Click Unpause / ContinueGame	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan
27	Click Unpause / ContinueGame	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan	Meng-klik tombol <i>Unpause</i> dan melanjutkan permainan
28	Click Back to Menu / GoTitleScreen	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>	Meng-klik tombol <i>Back to Menu</i> dan masuk kembali ke <i>title screen</i>
29	Exit	Keluar dari permainan dan menutup SaveMeNow.exe	Keluar dari permainan dan menutup SaveMeNow.exe

Setelah pengujian TFD dilakukan, ditemukan bahwa semua *flow* yang dibuat untuk *game flow* dinyatakan valid dan dapat berjalan dengan baik. Hal ini membuktikan bahwa implementasi dari *game flow* berhasil dilakukan dan sesuai dengan hasil yang diharapkan.

5.2.2 TFD Weapon Ammo

Untuk TFD kedua, yaitu TFD *Weapon Ammo*, dibuat untuk mengetahui bagaimana *flow* dari senjata dan peluru yang digunakan karakter pemain pada *game*. Gambar 5.2 menampilkan hasil TFD *Weapon Ammo*.



Gambar 5.2 Test Flow Diagram Weapon Ammo

Data Dictionary dari TFD Weapon Ammo ditunjukkan pada tabel 5.4.

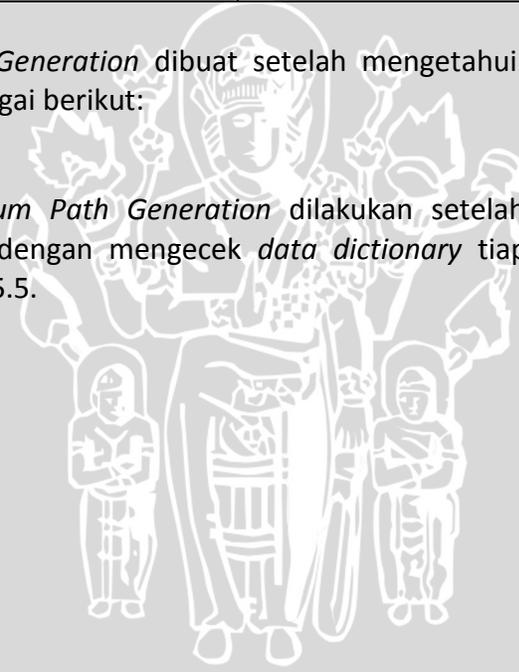
Tabel 5.4 Data Dictionary TFD Weapon Ammo

Events/Actions	Deskripsi
Enter	Mulai <i>flow</i>
Shoot Bullet	Menembakkan peluru
Ammo-1	Mengecek apakah peluru berkurang 1
Ammo = 0	Jumlah peluru mencapai nilai 0
AmmoEmpty	Peluru pada senapan habis
Reload	Mengisi kembali peluru pada senapan
AmmoFull	Peluru sudah terisi kembali
Exit	Akhir dari <i>flow</i>

Minimum Path Generation dibuat setelah mengetahui flow dari TFD dan menghasilkan *path* sebagai berikut:

1,2,3,4,5

Eksekusi *Minimum Path Generation* dilakukan setelah ditentukan sesuai dengan *path* tersebut dengan mengecek *data dictionary* tiap *flow* seperti yang ditunjukkan pada tabel 5.5.



Tabel 5.5 Hasil Uji *Flow TFD Weapon Ammo*

Flow	Event/Action	Hasil yang diharapkan	Hasil yang didapatkan
1	Enter	Mulai <i>flow</i> dengan peluru penuh	Mulai <i>flow</i> dengan peluru penuh
2	ShootBullet/Ammo-1	Menembakkan peluru dan peluru berkurang 1 per tembakan	Menembakkan peluru dan peluru berkurang 1 per tembakan
3	Ammo = 0/AmmoEmpty	Jumlah peluru bernilai 0 / Peluru pada senapan habis	Jumlah peluru bernilai 0 / Peluru pada senapan habis
4	Reload/AmmoFull	Mengisi kembali peluru pada senapan dan peluru pada senapan kembali penuh	Mengisi kembali peluru pada senapan dan peluru pada senapan kembali penuh
5	Exit	<i>Flow</i> selesai	<i>Flow</i> selesai

Setelah pengujian TFD dilakukan, ditemukan bahwa semua *flow* yang dibuat untuk *weapon ammo flow* dinyatakan valid dan dapat berjalan dengan baik. Hal ini membuktikan bahwa implementasi dari *Weapon Ammo* berhasil dilakukan dan sesuai dengan hasil yang diharapkan.

5.3 Play Testing

Dalam pengujian *Play Testing*, akan dilakukan terhadap sepuluh orang penguji, dimana sebagian diantaranya pernah dan mengetahui perkembangan *video games* dan sebagian lainnya merupakan orang yang dapat dikatakan awam dalam dunia *video games*. Sepuluh orang penguji tersebut nantinya akan dipersilahkan memainkan *game* "Save Me Now!!" dan setelah selesai memainkannya, mereka akan diberikan kuisisioner *feedback* untuk memberikan tanggapan mengenai *game* "Save Me Now!!". Pengujian ini dilakukan untuk mengetahui bagaimana tanggapan para pemain yang memainkan *game* "Save Me Now!!" mengenai *game* itu sendiri.

Play Testing dilakukan setelah *game* dirasa cukup stabil. Fokus yang ditunjukkan dalam permainan ini adalah *feedback* pengguna tentang *gameplay*, tampilan *art*, penyajian, dan lain sebagainya yang dapat dilihat dari *game* secara global. Penghitungan jawaban dari kuisisioner akan dihitung menggunakan skala likert. Tampilan hasil kuisisioner ditunjukkan pada tabel 5.6.

Tabel 5.6 Hasil Kuisioner *Feedback*

No.	Pertanyaan	Sangat Kurang (SK)	Kurang (K)	Cukup (C)	Bagus (B)	Sangat Bagus (SB)
1	Bagaimana dengan gameplay yang disajikan?	0	0	5	4	1
2	Bagaimana tentang efisiensi saat bermain?	0	2	7	1	0
3	Bagaimana tingkat kesulitan <i>game</i> ?	0	0	6	4	0
4	Seberapa " <i>fun</i> " kah <i>game</i> tersebut?	0	1	4	5	0
5	Bagaimana dengan <i>art</i> yang disajikan dalam <i>game</i> ?	3	5	2	0	0

Setelah didapatkan jawaban dari sepuluh panelis, akan dilakukan pengukuran skor dengan menggunakan skala likert. Masing-masing dari jawaban akan diberikan nilai sesuai dengan hasil perhitungan sebagai berikut:

Data jawaban dari pertanyaan pertama didapatkan sebagai berikut:

- Panelis yang menjawab SB = 1
- Panelis yang menjawab B = 4
- Panelis yang menjawab C = 5
- Panelis yang menjawab K = 0
- Panelis yang menjawab SK = 0

Kemudian hasil data dihitung dengan rumus $T \times P_n$ dan seluruh hasilnya dijumlahkan untuk menemukan nilai total skor.

$$\begin{aligned}
 SB &= 1 \times 5 = 5 \\
 B &= 4 \times 4 = 16 \\
 C &= 5 \times 3 = 15 \\
 K &= 0 \times 2 = 0
 \end{aligned}$$

$$SK = 0 \times 1 = 0$$

$$\text{Total skor} = 5 + 16 + 15 + 0 + 0 = 36$$

Setelah nilai total skor didapatkan, selanjutnya dihitung dengan rumus nilai total skor / Y x 100% untuk mencari persentase interpretasi.

$$36 / 50 \times 100\% = 72\%$$

Data jawaban dari pertanyaan kedua didapatkan sebagai berikut:

- Panelis yang menjawab SB = 0
- Panelis yang menjawab B = 1
- Panelis yang menjawab C = 7
- Panelis yang menjawab K = 2
- Panelis yang menjawab SK = 0

Kemudian hasil data dihitung dengan rumus $T \times P_n$ dan seluruh hasilnya dijumlahkan untuk menemukan nilai total skor.

$$SB = 0 \times 5 = 0$$

$$B = 1 \times 4 = 4$$

$$C = 7 \times 3 = 21$$

$$K = 2 \times 2 = 4$$

$$SK = 0 \times 1 = 0$$

$$\text{Total skor} = 0 + 4 + 21 + 4 + 0 = 29$$

Setelah nilai total skor didapatkan, selanjutnya dihitung dengan rumus nilai total skor / Y x 100% untuk mencari persentase interpretasi.

$$29 / 50 \times 100\% = 58\%$$

Data jawaban dari pertanyaan ketiga didapatkan sebagai berikut:

- Panelis yang menjawab SB = 0
- Panelis yang menjawab B = 4
- Panelis yang menjawab C = 6
- Panelis yang menjawab K = 0
- Panelis yang menjawab SK = 0

Kemudian hasil data dihitung dengan rumus $T \times P_n$ dan seluruh hasilnya dijumlahkan untuk menemukan nilai total skor.

$$SB = 0 \times 5 = 0$$

$$B = 4 \times 4 = 16$$

$$C = 6 \times 3 = 18$$

$$\begin{aligned} K &= 0 \times 2 = 0 \\ SK &= 0 \times 1 = 0 \\ \text{Total skor} &= 0 + 16 + 18 + 0 + 0 = 34 \end{aligned}$$

Setelah nilai total skor didapatkan, selanjutnya dihitung dengan rumus nilai total skor / Y x 100% untuk mencari persentase interpretasi.

$$34 / 50 \times 100\% = 68\%$$

Data jawaban dari pertanyaan keempat didapatkan sebagai berikut:

- Panelis yang menjawab SB = 0
- Panelis yang menjawab B = 5
- Panelis yang menjawab C = 4
- Panelis yang menjawab K = 1
- Panelis yang menjawab SK = 0

Kemudian hasil data dihitung dengan rumus $T \times P_n$ dan seluruh hasilnya dijumlahkan untuk menemukan nilai total skor.

$$\begin{aligned} SB &= 0 \times 5 = 0 \\ B &= 5 \times 4 = 20 \\ C &= 4 \times 3 = 12 \\ K &= 1 \times 2 = 2 \\ SK &= 0 \times 1 = 0 \\ \text{Total skor} &= 0 + 20 + 12 + 2 + 0 = 34 \end{aligned}$$

Setelah nilai total skor didapatkan, selanjutnya dihitung dengan rumus nilai total skor / Y x 100% untuk mencari persentase interpretasi.

$$34 / 50 \times 100\% = 68\%$$

Data jawaban dari pertanyaan keempat didapatkan sebagai berikut:

- Panelis yang menjawab SB = 0
- Panelis yang menjawab B = 0
- Panelis yang menjawab C = 2
- Panelis yang menjawab K = 5
- Panelis yang menjawab SK = 3

Kemudian hasil data dihitung dengan rumus $T \times P_n$ dan seluruh hasilnya dijumlahkan untuk menemukan nilai total skor.

$$\begin{aligned} SB &= 0 \times 5 = 0 \\ B &= 0 \times 4 = 0 \end{aligned}$$

$$\begin{aligned}C &= 2 \times 3 = 6 \\K &= 5 \times 2 = 10 \\SK &= 3 \times 1 = 3 \\ \text{Total skor} &= 0 + 0 + 6 + 10 + 3 = 19\end{aligned}$$

Setelah nilai total skor didapatkan, selanjutnya dihitung dengan rumus nilai total skor / Y x 100% untuk mencari persentase interpretasi.

$$19 / 50 \times 100\% = 38\%$$

Setelah dihitung pengukuran skor dengan menggunakan skala likert, didapatkan hasil persentase interpretasi 72%, maka untuk jawaban dari pertanyaan "Bagaimana dengan gameplay yang disajikan?" adalah Bagus (B). Untuk pertanyaan kedua, didapatkan hasil persentase interpretasi 58%, sehingga jawaban dari pertanyaan "Bagaimana tentang efisiensi saat bermain?" adalah Cukup (C). Pada pertanyaan ketiga, didapatkan hasil persentase interpretasi 68%. Maka jawaban dari pertanyaan "Bagaimana tingkat kesulitan *game*?" adalah Bagus (B). Hasil persentase interpretasi pertanyaan keempat adalah 68%. Oleh karena itu jawaban dari pertanyaan "Seberapa *fun*" kah *game* tersebut?" adalah Bagus (B). Dalam penghitungan nilai jawaban dari pertanyaan kelima, didapatkan hasil persentase interpretasi 38%. Maka untuk jawaban dari pertanyaan "Bagaimana dengan *art* yang disajikan dalam *game*?" adalah Kurang (K).

Hasil dari *play testing* menyebutkan bahwa para *game tester* menganggap *gameplay* yang disajikan sudah bagus dan tingkat efisiensi saat bermain juga sudah dianggap bagus. Selain itu juga dapat disimpulkan bahwa tingkat kesulitan pada *game* sudah bagus dan dapat membuat *game* dianggap cukup *fun* untuk dimainkan. Namun *art* yang disajikan dalam *game* masih kurang dikarenakan *art* yang diimplementasikan masih seadanya dan kurang maksimal.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan terhadap Desain dan Implementasi *Virtual Reality First Person Simulation Game* “Save Me Now!!” dapat diambil kesimpulan sebagai berikut:

1. *Gameplay* dan aturan dari *game* yang dibuat adalah *game* dengan genre *first person shooter* dengan skenario permainan penyelamatan sandera. Pemain menggerakkan karakter dengan sudut pandang orang pertama sebagai ketua polisi yang dapat mengarahkan anggota regu polisi dengan misi untuk menyelamatkan sandera. Pemain dapat membunuh musuh dengan cara menembak musuh. Misi dianggap gagal jika pemain mati, dan misi dianggap berhasil jika pemain bisa membawa sandera ke daerah aman di mobil polisi.
2. *Gameplay* dan aturan dari *Virtual Reality First Person Military Simulation Game* diimplementasikan dengan bahasa C# dan Java menggunakan Unity3D dan platform PC yang diintegrasikan dengan VR pada *smartphone* Android yang dijalankan menggunakan TrinusVR.
3. Fungsional dari *virtual reality first person military simulation game* terhadap *gameplay* sudah baik. Hal ini dibuktikan dengan pengujian *combinatorial* yang menghasilkan bahwa semua kombinasi dari parameter jumlah anggota polisi, perintah yang digunakan, dan parameter apakah sandera mengikuti pemain atau tidak bernilai valid dan dapat berjalan dengan baik. Pengujian TFD juga membuktikan bahwa *game flow* dan *weapon ammo flow* yang telah dibuat dapat berjalan sesuai dengan hasil yang diharapkan akan terjadi.
4. Menurut para *play tester*, *gameplay* yang dibuat sudah bagus dan tingkat kesulitannya juga bagus dan *game* dianggap *fun* untuk dimainkan, namun *art* yang disajikan dalam *game* masih kurang maksimal. Hal ini dapat disimpulkan dengan melihat hasil survey yang sudah dilakukan pada pengujian *play testing*.

6.2 Saran

Untuk meningkatkan hasil yang telah dicapai dari penelitian ini dapat dilakukan beberapa perbaikan sebagai berikut:

1. Dibuat variasi level dan peta yang lebih banyak agar *game* menjadi lebih menantang dan tidak monoton.
2. *Art* dari *game* diperbaiki kualitasnya agar *game* lebih nyaman saat dimainkan.

DAFTAR PUSTAKA

- Snyder, Carolyn., 2003, *Paper Prototyping: the fast and easy way to design and refine user interfaces*. San Francisco, CA: Morgan Kaufmann.
- Trybus, Jessica., 2014, Game Based Learning: What it is, Why it Works, and Where it's Going. *New Media Institute*, [online] Tersedia di: <<http://www.newmedia.org/game-based-learning--what-it-is-why-it-works-and-where-its-going.html>> [Diakses 23 Juni 2015].
- Hsu, Jeremy., 2010, For the U.S. Military, Video Games Get Serious. *livescience* [online] Tersedia di: <<http://www.livescience.com/10022-military-video-games.html>> [Diakses 21 Juni 2015].
- Steuer, Jonathan. *Defining Virtual Reality: Dimensions Determning Telepresence*. Department of Communication, Stanford University. Stanford, California. 1993.
- Google Developers, 2014. *Google Cardboard* [online] <<http://developers.google.com/cardboard/>> [Diakses 17 Juli 2015]
- Schreiber, Ian. 2009, *Game Design Concepts*. United State License: Creative Commons Attribution 3.0.
- Schultz, Charles P. & Bryant, Robert., 2005. *Game Testing: All in One*. The Premier Press: Thomson Course Technology PTR.
- Halarnkar, Pallavi, Sahil Shah, Harsh Shah, Hardik Shah, dan Anuj Shah. A Review on Virtual Reality. Department of Computer Engineering, NMIMS University Mumbai, Maharashtra 400056, India. 2012.
- Budiaji, Weksi. Skala Pengukuran dan Jumlah Respon Skala Likert. Dosen Fakultas Pertanian Universitas Sultan Ageng Tirtayasa, Serang, Banten. 2013.
- Sihite, Berta., Samopa., Febriliyan, & Sani., Nisfu Asrul. Pembuatan Aplikasi 3D Viewer Mobile dengan Menggunakan Teknologi Virtual Reality (Studi Kasus: Perobekan Bendera Belanda di Hotel Majapahit). Jurusan Sistem

Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember. Surabaya, Indonesia. 2013.

Rollings, Andrew & Adams, Ernest. 2009, *Fundamentals of Game Design*. 2nd ed. New Jersey: Prentice Hall

Odd Sheep, 2015. *Your PC Games in Virtual Reality on Your Smartphone Today*. [online] <<http://trinusr.com/>> [Diakses 29 Desember 2015]

VPL Research, 2015. *Virtual Reality in Military*. [online] Tersedia di: <<http://www.vrs.org.uk/virtual-reality-military/>> [Diakses 25 Desember 2015]

Yeruva, Abhisek R. *Gamebots Evolution for First Person Shooter (FPS) Games*. University of New Mexico. Albuquerque, New Mexico. 2011.

McPartland, Michelle & Gallagher, Marcus. *Game Designers Training First Person Shooter Bots*. University of Queensland. Brisbane, Queensland. 2013.

Grace, Lindsay. *The Design of Play*. Miami University. Oxford, Ohio. 2009.

LAMPIRAN A

Kuisiner *Feedback Pengujian Play Testing*

Game "Save Me Now!!"

No.	Pertanyaan	Sangat Kurang (SK)	Kurang (K)	Cukup (C)	Bagus (B)	Sangat Bagus (SB)
1	Bagaimana dengan gameplay yang disajikan?					
2	Bagaimana tentang efisiensi saat bermain?					
3	Bagaimana tingkat kesulitan <i>game</i> ?					
4	Seberapa "fun" kah <i>game</i> tersebut?					
5	Bagaimana dengan <i>art</i> yang disajikan dalam <i>game</i> ?					