

TUGAS AKHIR

PERANCANGAN GAME HORROR 3D “THE UNREAD” DENGAN METODE FINITE STATE MACHINE DAN FUZZY SUGENO



Disusun oleh:

ABDUL WAHID

21.18.077

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2025**

LEMBAR PERSETUJUAN

PERANCANGAN GAME HORROR 3D “THE UNREAD” DENGAN METODE FINITE STATE MACHINE DAN FUZZY SUGENO

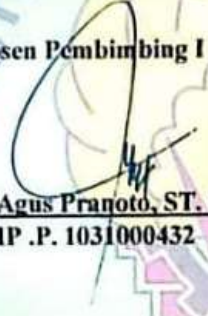
TUGAS AKHIR

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



Disusun Oleh :
Abdul Wahid
21.18.077

Diperiksa dan Disetujui

Dosen Pembimbing I

Yosep Agus Pranoto, ST, MT
NIP .P. 1031000432

Dosen Pembimbing II

Karina Aulia Sari, ST, M Eng
NIP .P. 1031000426

Mengetahui,
Wakil Dekan I

Dr. Ir. Badi Sulistyawati, ST, MT
NIP. 1977061520005012002

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

2025

LEMBAR KEASLIAN
PERNYATAAN KEASLIAN SKRIPSI

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

Nama : Abdul Wahid
NIM : 2118077
Program Studi : Teknik Informatika S-1
Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa skripsi saya dengan judul "PERANCANGAN GAME HORROR 3D "THE UNREAD" DENGAN METODE FINITE STATE MACHINE DAN FUZZY SUGENO" merupakan karya asli dan bukan merupakan duplikat dan mengutip seluruh karya oranglain. Apabila di kemudian hari, karya asli saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima segala konsekuensi apapun yang diberikan Program Studi Teknik Informatika S-1 Fakultas Teknologi Industri Institut Teknologi Nasional Malang.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Malang, 11 juli 2025

Yang membuat pernyataan



Abdul Wahid
NIM. 2118077

PERANCANGAN GAME HORROR 3D “THE UNREAD” DENGAN METODE FINITE STATE MACHINE DAN FUZZY SUGENO

Abdul Wahid, Yosep Agus Pranoto, Karina Auliasari

Teknik Informatika, Institut Teknologi Nasional Malang

Jalan Raya Karanglo km 2 Malang, Indonesia

2118077@scholar.itn.ac.id

ABSTRAK

Game horror merupakan salah satu genre yang digemari karena mampu menghadirkan ketegangan melalui atmosfer, jumpscare, dan alur cerita yang misterius. Penelitian ini bertujuan untuk mengembangkan game horror 3D berjudul "The Unread", yang berfokus pada eksplorasi dan investigasi dalam rumah kosong yang penuh misteri. Game ini dibangun menggunakan metode *Finite State Machine* (FSM) untuk mengatur perilaku musuh (AI) dalam tiga state utama, yaitu *wander*, *chase*, dan *jumpscare*. Untuk membuat perilakuf AI lebih adaptif, FSM dikombinasikan dengan logika *fuzzy Sugeno* yang memproses input berupa jarak ke pemain dan tingkat *sanity* untuk menghasilkan output berupa kecepatan serta transisi perilaku secara dinamis. Game ini dikembangkan dalam *perspektif First Person* (FPP) dengan model 3D *semi-realistis* dan atmosfer audio horror yang imersif. Hasil pengujian menunjukkan bahwa sistem FSM dan *fuzzy Sugeno* berjalan efektif dengan akurasi yang tinggi dan respons yang sesuai terhadap kondisi pemain. Seluruh fitur eksplorasi, interaksi objek, dan transisi antar level berfungsi dengan baik, mendukung alur cerita secara menyeluruh. Berdasarkan uji coba pengguna, sebanyak 76,88% responden memberikan penilaian positif terhadap kontrol, atmosfer horror, efek suara, dan kecerdasan musuh, membuktikan bahwa game "The Unread" berhasil memberikan pengalaman bermain yang menegangkan dan imersif.

Kata kunci : *Game Horror, 3D, FSM, Fuzzy, Sugeno, Eksplorasi, First Person Perspective*

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan kasih-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“Perancangan Game Horror 3d “The Unread” Dengan Metode Finite State Machine Dan Fuzzy Sugeno”**. Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan program pendidikan Strata Satu (S-1) Teknik Informatika di Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

Terselesaikannya laporan ini tentunya tidak lepas dari bantuan, dukungan, serta doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan terima kasih kepada:

1. Tuhan Yang Maha Esa atas kesehatan, kelancaran, dan kekuatan yang diberikan selama proses penyusunan Tugas Akhir ini.
2. Kedua orang tua dan keluarga tercinta yang senantiasa memberikan doa, semangat, dan dukungan secara moril maupun materil.
3. Bapak Yosep Agus Pranoto, S.T., M.T., selaku dosen Pembimbing Utama Skripsi yang telah memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
4. Ibu Karina Aulia Sari, ST., M Eng., selaku dosen Pembimbing Pendamping Skripsi yang telah memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
5. Teman-teman seperjuangan yang tidak dapat disebutkan satu per satu, terima kasih atas kebersamaan, bantuan, serta dukungan selama proses penyusunan Tugas Akhir ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk kritik dan saran yang bersifat membangun demi perbaikan di masa yang akan datang.

Malang, Juli 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB I LATAR BELAKANG	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Manfaat	3
1.6 Sistematika Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu	5
2.2 Definisi Game	6
2.3 Game Horror	6
2.4 Genre Game	6
2.5 Finite State Machine	9
2.6 Fuzzy Logic	10
2.7 Fuzzy Sugeno.....	10
2.8 Fuzzy Mamdani	11
2.9 Kecerdasan Buatan	11
BAB III ANALISIS DAN PERANCANGAN.....	12
3.1 Analisis Kebutuhan Fungsional	12
3.2 Analisis Kebutuhan Non Fungsional.....	12
3.3 Storyline	13
3.3 Storyboard	13
3.5 Level Game	18
3.6 Alur Game	20
3.7 Struktur Menu	22
3.8 Perancangan Metode Finite State Machine	22

3.9 Fuzzy Sugeno Pada AI Enemy	22
3.10 Fungsi Keanggotaan (Membership Function).....	23
3.11 Aturan Fuzzy	26
3.12 Contoh Perhitungan Logika Fuzzy Sugeno.....	26
3.13 Perancangan Antarmuka (User Interface)	29
3.14 Perancangan Kontrol	31
3.15 Desain Karakter.....	32
BAB IV IMPLEMENTASI DAN PENGUJIAN	34
4. 1 Hasil Implementasi.....	34
4.1.1 Hasil Halaman Utama Game	34
4.1.5 Hasil Game Level 1	36
4.1.6 Hasil Game Level 2.....	41
4.1.7 Hasil Game Level 3.....	43
4.1.8 Hasil Game Level 4.....	45
4.1.9 Hasil Game Level 5.....	47
4.2 Pengujian	49
4.2.1 Pengujian Sistem Finite State Machine (FSM)	49
4.2.2 Pengujian Fuzzy	50
4.2.3 Pengujian Fungsional	54
4.2.3 Pengujian User	56
4.2.4 Spesifikasi Perangkat yang Digunakan	58
4.2.5 Rekomendasi Spesifikasi Sistem.....	60
BAB V KESIMPULAN	62
5.1 Kesimpulan.....	62
5.2 Saran.....	63
DAFTAR PUSTAKA	64
LAMPIRAN	66

DAFTAR GAMBAR

Gambar 2. 2 Diagram state sederhana [6]	10
Gambar 3. 1 Flowchart Alur Game The Unread.....	21
Gambar 3. 2 Flowchart Struktur Menu Game The Unread.....	22
Gambar 3. 3 Finite State Machine Pada Musuh Game The Unread	22
Gambar 3. 4 Grafik Fungsi Keanggotaan Distance.....	23
Gambar 3. 5 Grafik Fungsi Keanggotaan Sanity	24
Gambar 3. 6 Output Keanggotaan State Enemy	25
Gambar 3. 7 Pengujian Defuzzifikasi Pada Matlab	28
Gambar 3. 8 Grafik Permukaan Pada Matlab	29
Gambar 4. 1 Tampilan Menu	34
Gambar 4. 2 Tampilan Kontrol Karakter	35
Gambar 4. 3 Tampilan Cut Scane	35
Gambar 4. 4 Tampilan Pause Game	35
Gambar 4. 5 Tampilan Loading Scareen.....	36
Gambar 4. 6 Tampilan Envirovment Level 1, 2 dan 5	36
Gambar 4. 7 Tampilan Pintu depan.....	37
Gambar 4. 8 Tampilan Jumpscare Tuyul	37
Gambar 4. 9 Tampilan Pintu Gudang.....	38
Gambar 4. 10 Tampilan Pintu Belakang	38
Gambar 4. 11 Tampilan Saklar Lampu	38
Gambar 4. 12 Tampilan Inspect Item Catatan.....	39
Gambar 4. 13 Tampilan Item Kunci.....	40
Gambar 4. 14 Tampilan Membuka Pintu Belakang	40
Gambar 4. 15 Tampilan Box Collider Trigger Pintu Belakang	40
Gambar 4. 16 Tampilan Ending Netral	41
Gambar 4. 17 Tampilan CutScane Level 2	41
Gambar 4. 18 Tampilan Catatan di kamar ibu	42
Gambar 4. 19 Tampilan Coretan Kode	42
Gambar 4. 20 Tampilan Keypad	42

Gambar 4. 21 Tampilan Jumpscare Game Over 1 level 2	43
Gambar 4. 22 Tampilan Pintu Bawah Tanah	43
Gambar 4. 23 Envirovment Ruang Bawah Tanah.....	44
Gambar 4. 24 Tampilan pintu ke level 4.....	44
Gambar 4. 25 Tampilan Item Baterai	44
Gambar 4. 26 Tampilan Jumpscare Game Over 1 Level 3	45
Gambar 4. 27 Tampilan Jumpscare Kunti Game Over 2 Level 3.....	45
Gambar 4. 28 Tampilan Envirovment Labirin	46
Gambar 4. 29 Tampilan Jumpscare Game Over Level 4	46
Gambar 4. 30 Tampilan Kunci level 4	46
Gambar 4. 31 Tampilan Pintu Ke Level 5	47
Gambar 4. 32 Tampilan Cut Scane Level 5	47
Gambar 4. 33 Tampilan Jumpscare Game Over Level 5	48
Gambar 4. 34 Tampilan Penentu Ending	48
Gambar 4. 35 Tampilan Good Ending	48
Gambar 4. 36 Tampilan Bad Ending.....	49
Gambar 4. 37 Kondisi Wander.....	49
Gambar 4. 38 Kondisi Chase.....	50
Gambar 4. 39 Kondisi Jumpscare	50
Gambar 4. 40 Pengujian Fuzzy di Unity	51
Gambar 4. 41 Pengujian Fuzzy di Matlab.....	51

DAFTAR TABEL

Tabel 3. 1 Storyboard Game The Unread	13
Tabel 3. 2 Perancangan Antarmuka (User Interface)	29
Tabel 3. 3 Perancangan Kontrol	31
Tabel 3. 4 Desain Karakter Game The Unread	32
Tabel 4. 1 Pengujian Fuzzy	52
Tabel 4. 2 Pengujian Fungsional	54
Tabel 4. 3 Pengujian user	56
Tabel 4. 4 Persentase Pengujian User	57
Tabel 4. 5 Spesifikasi perangkat yang digunakan	58

BAB I

LATAR BELAKANG

1.1 Latar Belakang

Game digital telah menjadi bagian penting dalam kehidupan masyarakat modern dan berkembang pesat sebagai industri yang signifikan di berbagai negara. Lembaga akademik pun mulai menyediakan program studi terkait pengembangan game. Meski minat terhadap penelitian game meningkat dalam dua dekade terakhir, sebagian besar penelitian masih berfokus pada aspek sosial, analisis pemain, dan aplikasi game serius, sementara studi tentang proses pengembangan game itu sendiri masih terbatas. [1].

Game bergenre horror menjadi salah satu genre yang memiliki banyak peminat karena mampu memberikan pengalaman bermain yang mendebarkan melalui elemen ketegangan, *jumpscare*, dan alur cerita yang misterius. Salah satu aspek terpenting dalam game horror adalah atmosfer yang mampu membuat pemain merasa terlibat langsung dalam dunia game. Penggunaan teknologi 3D dalam game horror dapat meningkatkan pengalaman bermain dengan menghadirkan lingkungan yang lebih realistis dan imersif.

Berdasarkan hal tersebut, maka dibuatlah sebuah game horror 3D berjudul "The Unread", di mana pemain akan berperan sebagai karakter utama yang berusaha mengungkap misteri di rumah nya yang lama ditinggalkan. Game ini mengusung konsep eksplorasi dan investigasi yang menuntut pemain untuk mencari petunjuk, menyelesaikan teka-teki, serta menghadapi berbagai kejadian menyeramkan sepanjang permainan.

Untuk mendukung dinamika dalam permainan, metode *Finite State Machine (FSM)* juga digunakan dalam pengembangan game ini, khususnya untuk membangun kecerdasan buatan (*AI*) pada karakter musuh. *FSM* merupakan metode yang memodelkan perilaku entitas berdasarkan keadaan (*state*) yang saling berhubungan. Setiap *state* merepresentasikan kondisi tertentu, dan transisi antar *state* ditentukan oleh kondisi atau peristiwa tertentu. Dalam game "The Unread", *FSM* digunakan untuk mengatur perilaku *AI Enemy*, seperti patroli, mengejar pemain, dan kembali ke jalur patroli. Dengan

menerapkan *FSM*, AI dapat beradaptasi secara dinamis terhadap aksi pemain, menciptakan tantangan yang lebih menegangkan dan realistis selama permainan berlangsung.

Selain *FSM*, game ini juga mengintegrasikan logika *fuzzy Sugeno* untuk membuat transisi antar state lebih adaptif. *Fuzzy Sugeno* memungkinkan AI musuh mengambil keputusan berdasarkan input kontinu seperti jarak ke pemain dan tingkat *sanity*, lalu menentukan kecepatan dan perilaku secara dinamis. Integrasi ini membuat perilaku AI lebih realistis dan menciptakan tantangan yang lebih menegangkan bagi pemain.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang teridentifikasi diatas, maka dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan game horor The Unread sebagai game berbasis desktop?
2. Bagaimana menerapkan metode *Finite State Machine* dalam pembuatan pergerakan *Enemy*?
3. Bagaimana mengintegrasikan logika *fuzzy Sugeno* ke dalam sistem *FSM* untuk menghasilkan perilaku musuh yang lebih adaptif dan dinamis terhadap kondisi pemain?

1.3 Tujuan

Terdapat beberapa tujuan dari penelitian ini sebagai berikut :

1. Mengimplementasikan game horor The Unread sebagai game berbasis desktop.
2. Menerapkan metode *Finite State Machine* dalam pengembangan game "The Unread" untuk membuat NPC bergerak secara otomatis sesuai dengan perilaku yang diharapkan.
3. Mengintegrasikan logika fuzzy Sugeno ke dalam sistem FSM untuk menghasilkan perilaku musuh yang lebih responsif terhadap kondisi pemain secara real-time.

1.4 Batasan Masalah

Agar identifikasi masalah yang dibahas lebih jelas dan mudah dipahami, maka dibuatlah Sbatasan-batasan masalah diantaranya sebagai berikut:

1. Game dikembangkan untuk *PC* (Windows) dan tidak dioptimalkan untuk mobile atau konsol.
2. *Gameplay* berfokus pada eksplorasi dan investigasi, tanpa sistem pertarungan atau senjata.
3. Menggunakan model 3D semi-realistis dengan efek suara untuk atmosfer horror.
4. Kecerdasan Buatan (AI) untuk Enemy menggunakan Finite State Machine yang dipadukan dengan logika fuzzy Sugeno untuk pengambilan keputusan yang lebih adaptif.
5. Game menggunakan perspektif *First Person Perspective (FPP)* untuk meningkatkan imersi dan pengalaman pemain dalam eksplorasi.
6. Game The Unread ditujukan untuk pemain dengan usia minimal 13 tahun.

1.5 Manfaat

1. Untuk dapat melatih keterampilan berpikir kritis, analitis, dan kemampuan memecahkan masalah di bawah tekanan
2. Melatih fokus dan kemampuan observasi pemain selama permainan berlangsung.
3. Membangun keberanian dan kepercayaan diri.

1.6 Sistematika Penelitian

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

BAB I : Pendahuluan berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II : Tinjauan Pustaka berisi dasar teori mengenai permasalahan yang berhubungan dengan penelitian game ini.

BAB III : Analisis dan perncangan berisi metode pengembangan game dan penerapan AI menggunakan FSM dan fuzzy Sugeno.

BAB IV : Implementasi dan pengujian berisi proses pembuatan game dan hasil pengujian AI serta gameplay.

BAB V : Kesimpulan dan saran berisi kesimpulan dari hasil penelitian dan pengembangan game, serta saran untuk pengembangan lebih lanjut

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Menurut Anang Habibi dan Muhammad Ibnu Athoillah (2025) yang berjudul "Pengembangan Game *Survival Horror Unity* 3D dengan Menerapkan *AI* pada *NPC*" membahas perancangan game *Survival Horror* yang mengutamakan pengalaman imersif dengan elemen strategi, aksi, dan manajemen sumber daya. Terinspirasi dari *Resident Evil* dan *Silent Hill*, game ini memanfaatkan atmosfer menegangkan dan ketakutan berbasis psikologi sesuai studi Norton & Gino (2014). Penelitian ini diharapkan dapat meningkatkan pengalaman bermain serta berkontribusi pada industri game lokal[2].

Menurut Febrina Ade Susianti, Anik Vega Vitianingsih, Achmad Choiron, Dwi Cahyono, Anggit Wikaningrum (2024) dengan judul "Aplikasi Game *Survival Horror 3d Pencegahan Penculikan Anak Remaja Berbasis Dekstop*" Penelitian tentang peningkatan kasus penculikan anak di Indonesia menyoroti kurangnya kesadaran orang tua terhadap aktivitas bermain anak serta kebutuhan remaja akan interaksi sosial dalam permainan daring. Untuk mengatasi hal ini, dikembangkan media edukasi berbasis permainan menggunakan finite state machines (FSM) untuk kontrol pemain dan forward chaining dalam sistem teka-teki[3].

Menurut Zefid Frenda Fathoni, Moh Ahsan, Muhammad Priyono Tri Sulistyanto dengan judul "Penerapan Fuzzy Sugeno Pada Npc Rintangan Dalam Game “Ana Cerita Extreme” Menggunakan Unity" (2023) Penelitian sebelumnya dalam pengembangan game Ana Cerita Extreme menunjukkan bahwa ketiadaan NPC rintangan membuat gameplay terlalu mudah. Untuk meningkatkan tantangan, diterapkan logika fuzzy Sugeno agar NPC dapat menyesuaikan perilakunya secara dinamis berdasarkan aturan tertentu. Hasil penelitian menunjukkan bahwa metode ini berhasil meningkatkan kompleksitas permainan, membuatnya lebih menarik dan menantang bagi pemain[4].

2.2 Definisi Game

Game pada hakikatnya merupakan aktivitas bermain yang dilakukan untuk memperoleh kesenangan, hiburan, maupun tantangan. Game dapat dipahami sebagai sebuah sistem yang menghadirkan konflik buatan, diatur oleh seperangkat aturan, serta menghasilkan suatu hasil yang dapat diukur. Game juga merupakan bentuk permainan interaktif yang memiliki tujuan, aturan, umpan balik, dan mendorong keterlibatan emosional pemain. Selain itu, game dapat dipandang sebagai kegiatan yang dilakukan secara sukarela untuk memperoleh kepuasan sekaligus memberikan pengalaman belajar. Dengan demikian, game tidak hanya berfungsi sebagai media hiburan, tetapi juga dapat dimanfaatkan sebagai sarana pembelajaran yang interaktif dan menyenangkan. [5].

2.3 Game Horor

Game merupakan suatu kegiatan yang bertujuan untuk bersenang-senang, olahraga ringan, mengisi waktu luang, dan terkadang digunakan sebagai sarana pendidikan. Salah satu game desktop yang paling banyak diunduh adalah game buatan Indonesia dengan genre *survival horror* yang dibuat dengan *Unity*. *Unity* merupakan game engine yang *user-friendly* dan mendukung lebih dari 25 platform serta menyediakan banyak aset siap pakai. Penulis membuat game 3d berbasis desktop dengan genre *survival horror* yang dibuat dengan *Unity*[6].

2.4 Genre Game

Game memiliki berbagai *genre* dengan karakteristik dan mekanisme permainan yang berbeda, mulai dari aksi cepat, eksplorasi cerita, hingga simulasi realistis. Pemahaman genre membantu pengembang menentukan konsep, desain, dan target audiens yang tepat. Adapun beberapa genre game yang umum dijumpai adalah sebagai berikut:

1. Action

Game bergenre *action* merupakan jenis permainan yang menekankan pada kecepatan, ketepatan, dan keterampilan pemain dalam mengendalikan karakter atau objek di dalam permainan. Umumnya, game bergenre ini menuntut pemain untuk merespons secara cepat terhadap situasi yang berubah secara dinamis, seperti menghindari rintangan,

menyerang musuh, atau mempertahankan suatu area dari ancaman tertentu. Unsur utama dalam game action adalah tantangan fisik dan koordinasi tangan-mata (*hand-eye coordination*) yang tinggi[7].

2. Adventure

Game bergenre adventure merupakan jenis permainan yang menekankan pada eksplorasi, alur cerita, serta pemecahan teka-teki atau tantangan tertentu yang menjadi inti dari pengalaman bermain. Genre ini sering kali memadukan elemen naratif yang kuat dengan interaksi langsung pemain terhadap lingkungan permainan. Pemain dihadapkan pada serangkaian misi atau tujuan yang harus diselesaikan untuk melanjutkan jalannya cerita, sehingga keterlibatan emosional dan rasa penasaran menjadi faktor pendorong utama[8].

3. Role-Playing Game (RPG)

Role Playing Game (RPG) merupakan salah satu genre game yang paling populer dan banyak diminati oleh berbagai kalangan. Dalam game bergenre *RPG*, pemain diberikan kesempatan untuk memainkan peran karakter fiksi atau tokoh khayalan yang memiliki latar cerita tertentu, kemampuan khusus, dan tujuan spesifik. Pemain biasanya diharuskan menjelajahi dunia permainan, berinteraksi dengan karakter lain, menyelesaikan misi, serta mengembangkan kemampuan karakter yang diperankan melalui sistem *leveling* atau peningkatan atribut. [9].

4. Simulation

Game simulasi merupakan salah satu *genre* permainan yang dirancang untuk meniru atau merepresentasikan aktivitas dan situasi nyata dalam bentuk virtual. *Genre* ini bertujuan memberikan pengalaman bermain yang menyerupai kondisi sebenarnya, baik dari segi lingkungan, interaksi, maupun aturan yang berlaku di dunia nyata. Pemain dapat mempelajari dan mengasah keterampilan tertentu melalui simulasi yang dikemas dalam bentuk permainan, sehingga selain memberikan hiburan, juga memiliki nilai edukatif dan praktis[10].

5. Horor

Genre Horror dirancang untuk menciptakan ketegangan, ketakutan, dan suasana mencekam bagi pemain. *Game horror* sering menggabungkan elemen *psikologis*, visual gelap, suara menyeramkan, dan ancaman tak terduga. Banyak *game horror* juga memanfaatkan keterbatasan pemain seperti peluru terbatas atau pencahayaan redup untuk meningkatkan rasa takut dan imersi[11].

6. Sports

Genre sports adalah jenis permainan yang mengadopsi kegiatan olahraga nyata dan menyajikannya dalam bentuk simulasi digital. Game dengan *genre* ini dirancang untuk meniru aturan, teknik, dan dinamika dari berbagai cabang olahraga seperti sepak bola, bola basket, tenis, balap, hingga *e-sport* yang berbasis kompetisi. Tujuan dari game sports tidak hanya memberikan hiburan, tetapi juga dapat digunakan sebagai media pembelajaran, pelatihan strategi, serta peningkatan keterampilan motorik dan kognitif[12].

7. Racing

Genre racing merupakan salah satu jenis permainan yang berfokus pada kecepatan, keterampilan mengendalikan kendaraan, dan pencapaian garis akhir dalam waktu sesingkat mungkin. Permainan ini biasanya mengharuskan pemain untuk mengendalikan objek tertentu seperti mobil, motor, perahu, atau kendaraan lainnya melalui lintasan yang telah ditentukan, sambil menghindari rintangan dan memanfaatkan peluang untuk meningkatkan performa. Karakteristik utama dari *genre* ini meliputi adanya lintasan atau jalur balap, sistem waktu, indikator kecepatan, serta tantangan berupa hambatan fisik maupun kondisi lingkungan yang memengaruhi jalannya permainan[13].

8. Fighting

Game bergenre *fighting* merupakan jenis permainan yang berfokus pada pertarungan antara dua karakter atau lebih dalam sebuah arena tertentu. Pertarungan biasanya dilakukan dalam bentuk *one-on-one combat* atau *team battle*, di mana pemain atau kecerdasan buatan (AI)

saling berhadapan untuk mengalahkan lawan melalui kombinasi serangan, pertahanan, dan strategi. Genre ini mengandalkan mekanisme *real-time combat*, *hit points (HP)* sebagai indikator kesehatan karakter, serta sistem kontrol yang memungkinkan pemain melakukan berbagai gerakan seperti serangan cepat, serangan spesial, dan teknik bertahan[14].

9. Platformer

Game platformer adalah salah satu *genre* permainan yang menitikberatkan pada navigasi karakter melalui lingkungan permainan yang terdiri dari berbagai rintangan, permukaan, atau platform yang harus dilompati atau dilewati untuk mencapai tujuan tertentu. Pada umumnya, *platformer* mengandalkan keterampilan pemain dalam mengendalikan karakter, seperti melompat, memanjat, atau menghindari hambatan, sambil menjaga ketepatan waktu dan koordinasi gerakan[15].

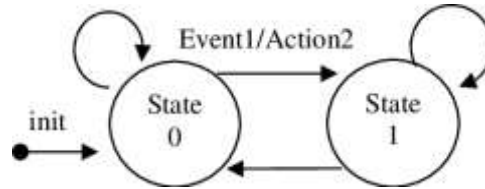
10. Shooter

Shooter merupakan salah satu *genre* permainan yang menitikberatkan pada penggunaan senjata api atau senjata jarak jauh sebagai mekanisme utama dalam *gameplay*. *Genre* ini terbagi menjadi beberapa subkategori, salah satunya adalah *First-Person Shooter (FPS)*, di mana pemain melihat lingkungan permainan dari sudut pandang karakter yang dikendalikan, seolah-olah mereka berada di dalam dunia game tersebut. Fokus utama dari *FPS* adalah keterampilan membidik, kecepatan bereaksi, strategi pertempuran, serta koordinasi tangan dan mata (*hand-eye coordination*)[16].

2.5 Finite State Machine

Finite state machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau event tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika

menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks [17].



Gambar 2. 1 Diagram state sederhana [6]

Gambar 2.2 memperlihatkan *FSM* dengan dua buah state dan dua buah input serta empat buah output yang berbeda seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *State0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

2.6 Fuzzy Logic

Metode *fuzzy Sugeno* memiliki prinsip yang serupa dengan metode *Mamdani*, tetapi perbedaannya terletak pada output yang dihasilkan. Pada metode *Sugeno*, output berupa konstanta, bukan himpunan *fuzzy*. Fungsi keanggotaan dalam metode ini dikenal sebagai fungsi singleton, yang memiliki nilai keanggotaan 1 pada titik tertentu dan 0 di tempat lain[18].

Keunggulan metode *fuzzy Sugeno* adalah kemampuannya menangani berbagai kebutuhan pemodelan dengan pendekatan orde nol, yang seringkali lebih sederhana dan efisien[19].

2.7 Fuzzy Sugeno

Fuzzy Sugeno merupakan salah satu metode dalam logika *fuzzy* yang digunakan untuk pengambilan keputusan berbasis aturan *IF-THEN*. Metode ini memiliki keunggulan dalam menghasilkan output numerik melalui kombinasi linier dari input yang diberikan. Dalam prosesnya, *Fuzzy Sugeno* memanfaatkan fungsi keanggotaan *fuzzy* untuk memodelkan hubungan antara variabel input dan output, serta menggunakan proses implikasi untuk menentukan hasil dari setiap aturan[20].

Keunggulan utama dari metode ini terletak pada kemampuannya dalam memberikan hasil yang presisi dan dapat diimplementasikan secara efisien

dalam berbagai sistem kontrol maupun sistem pendukung keputusan. Oleh karena itu, *Fuzzy Sugeno* banyak diterapkan di berbagai bidang seperti industri, sistem keuangan, hingga kecerdasan buatan[20].

2.8 Fuzzy Mamdani

Fuzzy Mamdani merupakan salah satu metode dalam logika *fuzzy* yang digunakan untuk menyelesaikan permasalahan yang mengandung ketidakpastian, terutama dalam sistem prediksi dan pengambilan keputusan. Metode ini sering dikenal sebagai metode *Min-Max* dan menjadi salah satu pendekatan populer karena kemampuannya dalam menangani data linguistik dan menghasilkan output yang dapat diinterpretasikan secara logis.

Proses dalam metode *Fuzzy Mamdani* melibatkan beberapa tahapan, yaitu: (a) pembentukan himpunan *fuzzy* untuk setiap variabel input dan output, (b) penerapan fungsi implikasi untuk setiap aturan *fuzzy*, (c) pembentukan aturan-aturan *IF-THEN*, serta (d) proses penegasan (*defuzzifikasi*) untuk mendapatkan nilai output yang bersifat pasti. Dalam banyak kasus, proses defuzzifikasi dilakukan dengan metode centroid, di mana hasil akhirnya diperoleh melalui rata-rata tertimbang dari semua output yang dihasilkan oleh aturan *fuzzy*. Metode *centroid* ini menghasilkan output yang halus dan konsisten, sehingga perubahan pada input akan menghasilkan perubahan output yang proporsional[21].

2.9 Kecerdasan Buatan

Kecerdasan Buatan (*Artificial Intelligence / AI*) merupakan teknologi yang dirancang untuk meniru kemampuan berpikir, belajar, dan mengambil keputusan layaknya manusia. Dalam industri video game, *AI* memiliki peran yang semakin signifikan dalam menciptakan pengalaman bermain yang dinamis, realistis, dan imersif. Perkembangan *AI* dalam game telah mengalami kemajuan pesat, mulai dari penggunaan skrip sederhana hingga penerapan algoritma canggih berbasis *machine learning* dan *neural network* yang memungkinkan sistem belajar serta beradaptasi terhadap perilaku pemain[22].

BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional mencakup fitur utama yang harus tersedia dalam game “The Unread”, antara lain:

1. Sistem Navigasi Pemain – Menggunakan *Character Controller* atau *Rigidbody* untuk pergerakan dalam lingkungan 3D.
2. Interaksi Objek – Implementasi *Raycasting* atau *Trigger Collider* untuk mendeteksi interaksi pemain dengan objek.
3. Sistem Puzzle – Menggunakan skrip berbasis C# untuk logika teka-teki dan pemicu event.
4. Efek Suara & Visual – Menggunakan *AudioSource*, *Post Processing*, dan pencahayaan dinamis (*Light Baking* atau *Realtime Lighting*).
5. AI Musuh – Memanfaatkan *NavMesh* untuk navigasi musuh serta *State Machine* untuk perilaku adaptif.

3.2 Analisis Kebutuhan Non Fungsional

Kebutuhan non-fungsional dalam pengembangan game “The Unread” mencakup aspek teknis dan pengalaman pengguna untuk memastikan performa, kualitas, dan keandalan game. Berikut adalah beberapa kebutuhan non-fungsional yang harus dipenuhi:

1. Kinerja – Game harus stabil di 30+ *FPS* dengan optimisasi *LOD*, *Occlusion Culling*, dan *light baking*.
2. Keamanan – Proteksi data dengan enkripsi dan validasi input untuk mencegah eksploitasi.
3. Kompatibilitas – Mendukung penuh sistem operasi Windows untuk memastikan performa dan stabilitas yang optimal.
4. *User Experience* – *UI intuitif* dengan desain mendukung atmosfer horror.
5. Stabilitas – Bebas dari *crash* dan *bug fatal* melalui stress test dan *playtesting*.
6. Ukuran File – Optimasi *texture compression* dan penghapusan aset tak terpakai.

7. Audio & Visual – *Spatial audio* dan pencahayaan dinamis untuk meningkatkan kenyamanan *player*.

3.3 Storyline

Setelah bertahun-tahun merantau, seorang pria kembali ke kampung halamannya yang kini tampak asing dan sunyi. Saat mencoba masuk ke rumah tuanya, ia mendapati pintu terkunci dan mulai mencari kunci di gudang, namun sejak saat itu, hal-hal aneh mulai terjadi. Penampakan makhluk menyeramkan muncul tiba-tiba, rumah dalam kondisi berantakan, dan berbagai petunjuk seperti catatan serta coretan kode mulai membimbingnya pada serangkaian teka-teki yang harus dipecahkan untuk bisa melangkah lebih jauh.

Semakin dalam ia menjelajah, teror semakin nyata. Entitas seperti pocong dan kuntilanak memburunya, sementara lorong bawah tanah memperlihatkan sisi tergelap dari rumah itu. Setiap kesalahan bisa berakhir dengan jumpscare mematikan atau kematian. Di akhir perjalanan, ia dihadapkan pada pilihan penting ketika sosok ibunya muncul: tetap tinggal atau meninggalkan semuanya. Keputusan itu akan menentukan akhir dari kisah ini, karena tidak semua rahasia layak untuk dibuka.

3.3 Storyboard

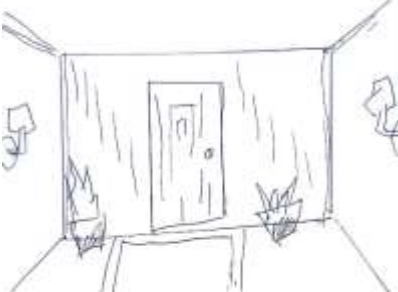
Storyboard yang merepresentasikan setiap tahapan penting dalam permainan. Storyboard ini berfungsi sebagai panduan visual untuk menggambarkan rangkaian peristiwa, interaksi, dan suasana yang dihadapi oleh karakter utama.

Adapun storyboard dari game The Unread dapat dilihat pada Tabel 3.1 berikut:

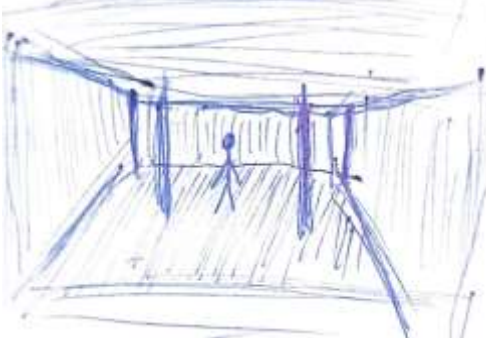
Tabel 3. 1 Storyboard Game The Unread

No	Sketsa	Keterangan
1		Pemain tiba di rumah, tetapi suasananya terasa kosong, dan pintu depan terkunci.

No	Sketsa	Keterangan
2		Pintu belakang juga terkunci. cari cara lain untuk bisa masuk.
3		sosok tuyul kecil muncul dengan gerakan cepat dan menjeramkan—sebuah animasi jumpscare aktif
4		pemain menemukan pintu yang terkunci dan harus memindahkan sebuah blok terlebih dahulu agar pintu bisa terbuka dan bisa dilalui.
5		Dalam alur cerita, pocong akan bereaksi saat pemain mendekat; jika sangat dekat pocong akan menghilang.
6		Pada Level 1, jika pemain memilih keluar lewat pintu depan tanpa melanjutkan ke Level 2, maka akan mendapatkan ending alternatif berupa Ending Netral atau Ending Kabur.

No	Sketsa	Keterangan
7		Kamar terlihat berantakan, ada sebuah catatan di atas meja.
8		Player mengambil dan membaca catatan tersebut, Membaca dan ikuti informasi yang diberikan.
9		Setelah cutscene Level 2 berakhir, pemain menemukan coretan kode di ujung tangga yang berisi petunjuk untuk menyelesaikan puzzle di bawahnya.
10		Lanjutkan pencarian dengan memeriksa semua kamar dan lantai.

No	Sketsa	Keterangan
11		Pada tangga ada tempat tersembunyi, dan ada sebuah pintu yang harus dibuka dengan kode, keypad disini digunakan untuk membuka pintu ke level selanjutnya, kode nya adalah semua coretan yang ada di dinding.
12		Player mencoba membuka pintu yang terkunci, namun justru memicu jumpscare mengejutkan yang langsung mengakhiri permainan dengan game over.
13		Setelah memecahkan puzzle di level sebelumnya, pemain memasuki Rumah Bawah Tanah Level 3 yang gelap dan tegang, dengan musuh berkeliaran. Tertangkap musuh berarti game berakhir.
14		Pemain mencoba kabur ke pintu sebelumnya tanpa menyelesaikan puzzle level 3. Ia tertangkap oleh entitas, ditampilkan animasi penangkapan, lalu layar Game Over muncul dan permainan kembali dimulai dari awal level 3.

No	Sketsa	Keterangan
15		Pemain harus mengumpulkan 2 baterai yang tersebar di ruangan dan sembari menghindari kejaran dari kunti.
16		Di level 4, saat pemain tertangkap oleh Kunti, akan diputar animasi penangkapan sebelum layar menampilkan Game Over dan permainan mengulang kembali dari level ini.
17		Level 5 dimulai dengan cutscene saat pemain terbangun dalam keadaan bingung, lalu segera turun ke lantai bawah untuk mencari tahu apa yang sedang terjadi.
18		Ada hal aneh yang mulai mendekat dan menyeramkan, Kabur dan jangan sampai tertangkap.
19		Ending cerita ditentukan saat pemain mendekati pintu keluar; ibu muncul dan memberikan dua pilihan: tetap tinggal atau meninggalkan rumah.

3.5 Level Game

1. Level 1: Kembali ke Rumah

Lokasi: Halaman depan rumah dan gudang

Objektif: Masuk ke rumah, mencari keluarga

Detail Gameplay:

- Pemain tiba di kampungnya yang terasa lebih sunyi dari biasanya.
- Rumah tampak tua dan berdebu, menunjukkan sudah lama tidak dihuni.
- Tampak seperti orang dari jauh, dan tiba-tiba hilang saat di dekati.
- Pintu utama tiba-tiba tertutup sendiri setelah masuk.
- Pemain mulai merasakan kejanggalan bayangan di sudut ruangan.

Mekanisme :

- Pintu terkunci sendiri, suara bisikan dari ruang keluarga.
- Objek-objek mulai berpindah tempat sendiri saat pemain mendekat.

Jumpscare:

- Sosok hitam terlihat melintas di lorong saat pemain.

Exit ke Level 2:

- Menyelesaikan puzzle membuka pintu belakang rumah.
- Netral Ending:
- Kabur dari rumah karna merasa kejanggalan.

2. Level 2: Jejak Keluarga

Lokasi: Kamar

Objektif: Mencari petunjuk

Detail Gameplay:

- Pemain mulai menemukan barang-barang yang bergeletakan.
- Kamar terkunci, harus mencari kunci.
- Menemukan diary ibu yang mengandung beberapa catatan tentang kejadian aneh sebelum mereka menghilang.
- Mekanisme :
- Jika pemain membuka lemari pakaian, terdengar suara nafas berat dari dalam, tapi tidak ada apa-apa saat diperiksa.
- Lampu berkedip kedip
- Pemain bisa menggunakan senter.

Exit ke Level 3:

- Menemukan kunci keruangan ibu dan Menemukan catatan.

3. Level 3: Kembali ke Rumah

Lokasi: Lorong Bawah Tanah

Objektif: Mengungkap rahasia keluarga

Detail Gameplay:

- Gudang berisi barang-barang lama milik keluarga.
- Sebuah sosok membantu memberi petunjuk.
- Mekanisme :
- Mengumpulkan item untuk membuka lorong.
- Jika player kembali tanpa menyelesaikan puzzle maka sosok entitas akan mengejar.

Jumpscare:

- Saat pemain mendekati lemari muncul entitas.
- Sosok entitas yang selama ini muncul samar-samar akhirnya terlihat jelas.
- Exit ke Level 4:
- Pemain menemukan pintu rahasia yang mengarah ke lorong bawah tanah.

4. Level 4: Lorong Bawah Tanah

Lokasi: Lorong Bawah Tanah

Objektif: Jangan tertangkap kuntilanak.

Detail Gameplay:

- Pemain menemukan ruang bawah tanah tersembunyi, dengan dinding penuh coretan.
- Mencari kunci pintu utama rumah, tapi terjebak dalam kotak dengan puzzle.
- Kabur dari kejaran entitas.

Mekanisme :

- Efek sanity pada pemain ketika lama diruangan.
- Saat masuk lemari maka pemain tidak akan di target.

Jumpscare:

- Terlihat dan menghilang

Exit ke Level 5:

- Pemain berhasil mengambil kunci dan kembali ke lantai atas.

5. Level 5: Kembali ke Ruang Tamu

Lokasi: Ruang Tamu

Objektif: Memutuskan akhir permainan.

Detail Gameplay:

- Membuka pintu keluar (Good Ending).
- Tetap tinggal dan menghadapi sosok yang muncul di belakangnya (Bad Ending).
- Mekanisme :
- Ketika efek sanity tinggi maka entitas akan berlari lebih cepat.
- Jumpscare:
- Terlihat dan menghilang
- Jika salah masuk ruangan, layar berubah hitam dan terdengar suara “Kenapa kamu nggak pulang lebih cepat?”

Exit ke Level 5:

- Pemain berhasil mengambil kunci dan kembali ke lantai atas.

6. Ending:

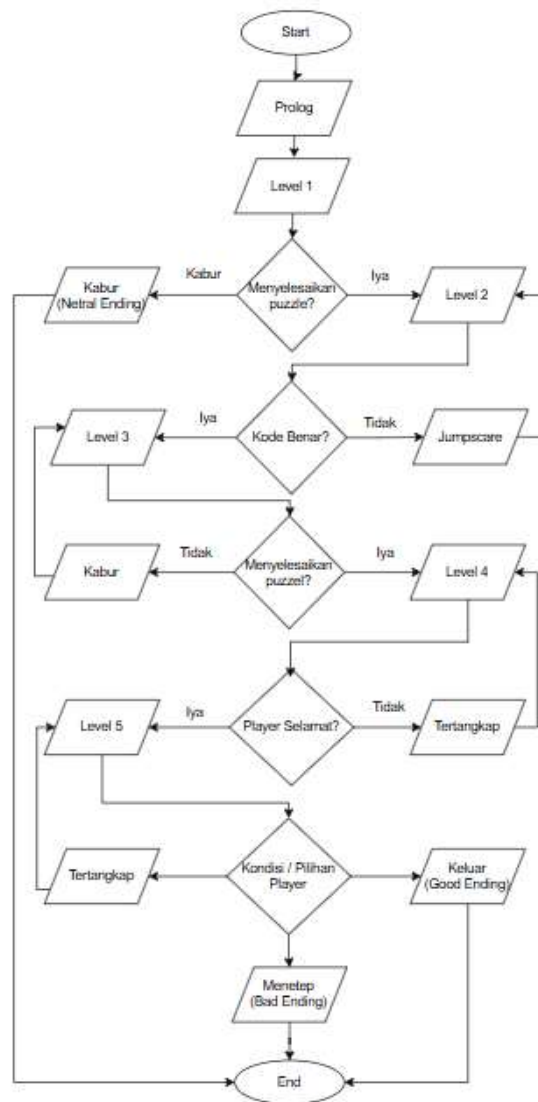
Netral Ending : Pemain keluar rumah dan mengabaikan semuanya.

Good Ending: Pemain keluar dari rumah, dan mengetahui semuanya.

Bad Ending: Pemain terperangkap di rumah selamanya.

3.6 Alur Game

Perancangan alur game bertujuan untuk memetakan proses permainan dari awal hingga akhir. Dengan perancangan ini, pengembang dapat memahami struktur, memastikan alur logis, dan menghindari kesalahan. Flowchart sering digunakan untuk memvisualisasikan alur permainan agar lebih mudah dipahami.

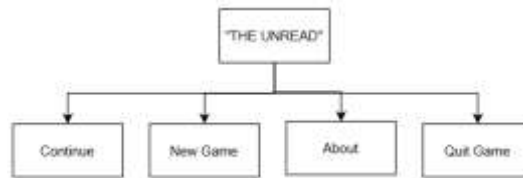


Gambar 3. 1 Flowchart Alur Game The Unread

Flowchart tersebut menggambarkan alur cerita dalam sebuah game yang dimulai dari tahap prolog hingga berbagai kemungkinan akhir permainan. Pemain memulai di Level 1 apabila bisa menyelesaikan puzzle maka lanjut ke level 2, tetapi player bisa mendapatkan ending di level 1 yaitu langsung keluar, jika menemukan kunci sebelum waktu habis maka lanjut ke level 3 jika player menemukan jalan keluar sebelum waktu habis maka akan ke level 4, dan saat level ini player tidak boleh tertangkap, jika tertangkap akan mengulang, dan jika selamat maka lanjut level 5, pada level 5 ini karakter akan memilih ending antara good atau bad ending.

3.7 Struktur Menu

Merupakan rancangan dari struktur menu awal membuka game yang dimana ada pilihan continue, new game, about, dan quit game, seperti pada flowchart pada gambar 3.2

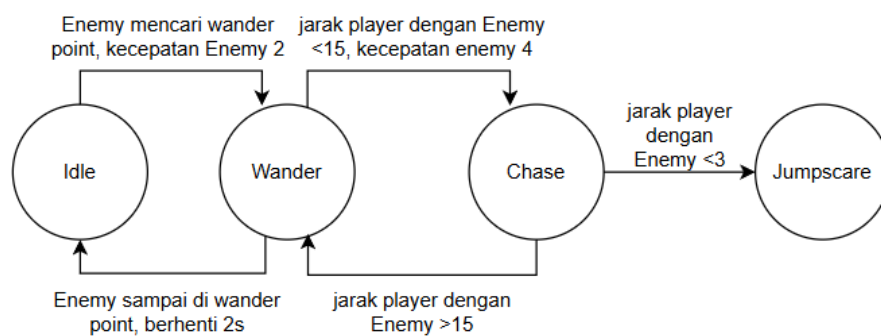


Gambar 3. 2 Flowchart Struktur Menu Game The Unread

3.8 Perancangan Metode Finite State Machine

Metode Finite State Machine (FSM) digunakan untuk mengatur perilaku AI karakter musuh dalam game horor 3D. FSM memungkinkan peralihan antara berbagai keadaan (state) berdasarkan kondisi tertentu yang telah ditentukan. Pada sistem ini, setiap state merepresentasikan tindakan spesifik yang dilakukan oleh Enemy, seperti idle, wander, chase atau kembali ke state wander.

Pada game “The Unread” ini metode Finite State Machine diterapkan pada karakter musuh. Alur kecerdasan buatan finite state machine seperti gambar 3.3.



Gambar 3. 3 Finite State Machine Pada Musuh Game The Unread

3.9 Fuzzy Sugeno Pada AI Enemy

Fuzzy Sugeno digunakan dalam sistem kecerdasan buatan (AI) untuk menentukan perilaku AI Enemy berdasarkan input yang diterima. Pada game horor 3D ini, Fuzzy Sugeno diterapkan dalam pengambilan keputusan AI Enemy terhadap posisi pemain (Player) dan Sanity. Dengan pendekatan ini, AI dapat secara dinamis beralih di antara beberapa keadaan (state) seperti Idle, Wander, Chase, dan Jumpscare berdasarkan Distance dan status Sanity.

Variabel input dan output dalam sistem *Fuzzy Sugeno* ini terdiri dari:

1. Input

1) *Distance Player* (x)

- Dekat: $x < 3$
- Sedang: $3 \leq x \leq 15$
- Jauh: $x > 15$

2) *Sanity Player* (y)

- Rendah: $x < 250$
- Sedang: $250 \leq x \leq 500$
- Tinggi: $x > 500$

2. Output:

1) *State dan Speed AI Enemy*

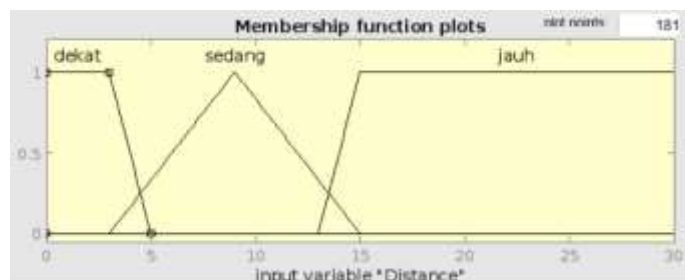
- *Jumpscare*: 0
- *Idle/Wander*: 2
- *Chase* (*Sanity* Tinggi) : 4
- *Chase 2* (*Sanity* Sedang): 5
- *Chase 3* (*Sanity* Rendah): 6

3.10 Fungsi Keanggotaan (Membership Function)

Fungsi keanggotaan digunakan untuk mengonversi nilai input numerik menjadi derajat keanggotaan dalam himpunan fuzzy. Berikut adalah fungsi keanggotaan untuk setiap variabel input:

a) Input 1: *Distance Player* (x)

Berikut merupakan grafik fungsi keanggotaan (*membership function*) untuk *Distance*, terdapat 3 himpunan yaitu dekat, sedang dan jauh dapat dilihat pada gambar 3.4:



Gambar 3. 4 Grafik Fungsi Keanggotaan Distance

Dibawah ini merupakan rumus fungsi keanggotaan (membership function) untuk variabel Distance :

- 1) Dekat (<3) Domain: [0, 0, 3, 5]

Menggunakan fungsi *trapezoidal*:

$$\mu_{Dekat}(x) = \begin{cases} 1, & x \leq 3 \\ \frac{5-x}{5-3}, & 3 < x < 5 \\ 0, & x \geq 5 \end{cases} \quad (3.1)$$

- 2) Sedang (≥ 3 atau ≤ 15) Domain: [3, 9, 15]

Menggunakan fungsi *triangular*:

$$\mu_{Sedang}(x) = \begin{cases} \frac{x-3}{9-3}, & 3 \leq x \leq 9 \\ \frac{15-x}{15-9}, & 9 < x < 15 \\ 0, & \text{lainnya} \end{cases} \quad (3.2)$$

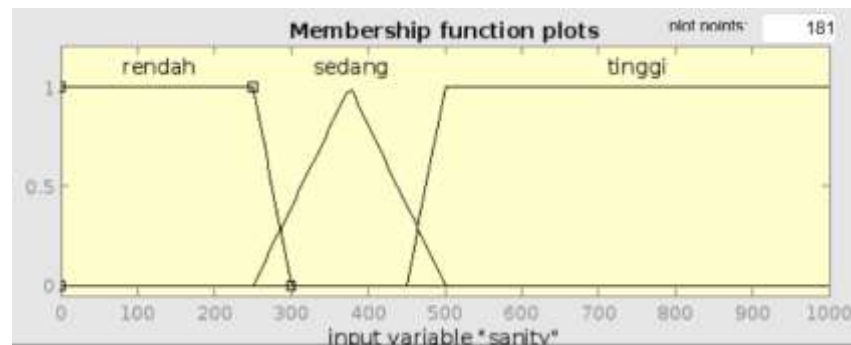
- 3) Jauh (>15) Domain: [13, 15, 30, 30]

Menggunakan fungsi *trapezoidal*:

$$\mu_{Jauh}(x) = \begin{cases} 0, & x \leq 13 \\ \frac{x-13}{15-13}, & 13 < x < 15 \\ 1, & x \geq 15 \end{cases} \quad (3.3)$$

- b) Input 2: Sanity Player (y)

Berikut merupakan fungsi keanggotaan (membership function) untuk Sanity, terdapat 3 himpunan yaitu rendah, sedang dan tinggi dapat dilihat pada gambar 3.5:



Gambar 3. 5 Grafik Fungsi Keanggotaan Sanity

Dibawah ini merupakan rumus fungsi keanggotaan (membership function) untuk variabel Sanity :

1) Rendah (<250) Domain: [0, 0, 250, 300]

Menggunakan fungsi *trapezoidal*:

$$\mu_{Rendah}(y) = \begin{cases} 1, & y \leq 250 \\ \frac{300-y}{300-250}, & 250 < y < 300 \\ 0, & y \geq 300 \end{cases} \quad (3.4)$$

2) Sedang (≥ 250 atau ≤ 500) Domain: [250, 375.5, 500]

Menggunakan fungsi *triangular*:

$$\mu_{Sedang}(y) = \begin{cases} \frac{y-250}{375.5-250}, & 250 \leq y \leq 375.5 \\ \frac{500-y}{500-375.5}, & 375.5 < y < 500 \\ 0, & \text{lainnya} \end{cases} \quad (3.5)$$

3) Tinggi (>500) Domain: [450, 500, 1000, 1000]

Menggunakan fungsi *trapezoidal*:

$$\mu_{tinggi}(y) = \begin{cases} 0, & y \leq 450 \\ \frac{y-450}{500-450}, & 450 < y < 500 \\ 1, & y \geq 500 \end{cases} \quad (3.6)$$

c) Output: Kecepatan Enemy (Speed)

Berikut Merupakan Fungsi keanggotaan (*membership function*) untuk Output *State/Speed* , dapat dilihat pada gambar 3.6:



Gambar 3. 6 Output Keanggotaan State Enemy

- *Jumpscare*: 0
- *Idle/Wander*: 2
- *Chase (Sanity Tinggi)* : 4
- *Chase 2 (Sanity Sedang)*: 5
- *Chase 3(Sanity Rendah)*: 6

3.11 Aturan Fuzzy

Aturan *fuzzy* digunakan untuk menentukan tindakan yang akan diambil AI berdasarkan kombinasi input. Berikut adalah aturan *fuzzy* pada sistem AI *Enemy*:

1. IF Distance Dekat AND Sanity Rendah THEN Jumpscare = 0
2. IF Distance Dekat AND Sanity Sedang THEN Jumpscare = 0
3. IF Distance Dekat AND Sanity Tinggi THEN Jumpscare = 0
4. IF Distance Sedang AND Sanity Rendah THEN Chase3 = 6
5. IF Distance Sedang AND Sanity Sedang THEN Chase2 = 5
6. IF Distance Sedang AND Sanity Tinggi THEN Chase = 4
7. IF Distance Jauh AND Sanity Rendah THEN Wander = 2
8. IF Distance Jauh AND Sanity Sedang THEN Wander = 2
9. IF Distance Jauh AND Sanity Tinggi THEN Wander = 2

3.12 Contoh Perhitungan Logika Fuzzy Sugeno

Berikut rumus menghitung output *Speed* :

Speed Output :

$$Output = \frac{\sum(\alpha_i \times z_i)}{\sum \alpha_i} \quad (3.7)$$

dimana :

α_i : derajat keanggotaan (nilai kekuatan atau bobot) dari aturan ke-i

z_i : output konstan atau linear dari aturan ke-i

Contoh perhitungan Jika *Distance Player* (x) = 7.5

dan *Sanity Player* (y) = 144

1) Tahapan Fuzzifikasi

Hitung Derajat Keanggotaan (μ)

a) *Distance Player* (x = 7.5)

1) Dekat

$$\mu_{Dekat}(x) = \begin{cases} 1, & x \leq 3 \\ \frac{5-x}{5-3}, & 3 < x < 5 \\ 0, & x \geq 5 \end{cases} \quad (3.8)$$

$$\mu_{Dekat}(7.5) = 0 \text{ (Karena } 7.5 > 5 \text{)}$$

2) Sedang

$$\mu_{Sedang}(x) = \begin{cases} \frac{x-3}{9-3}, & 3 \leq x \leq 9 \\ \frac{15-x}{15-9}, & 9 < x < 15 \\ 0, & \text{lainnya} \end{cases} \quad (3.9)$$

$$\mu_{Sedang}(7.5) = \frac{7.5 - 3}{9 - 3} = \frac{1.5}{6} = 0.75$$

3) Jauh

$$\mu_{Jauh}(x) = \begin{cases} 0, & x \leq 13 \\ \frac{x-13}{15-13}, & 13 < x < 15 \\ 1, & x \geq 15 \end{cases} \quad (3.10)$$

$$\mu_{Jauh}(7.5) = 0 \text{ (Karena } 7.5 < 13)$$

b) Sanity Player ($y = 144$)

1) Rendah

$$\mu_{Rendah}(y) = \begin{cases} 1, & y \leq 250 \\ \frac{300-y}{300-250}, & 250 < y < 300 \\ 0, & y \geq 300 \end{cases} \quad (3.11)$$

$$\mu_{Rendah}(144) = 1 \text{ (Karena } 144 \leq 250)$$

2) Sedang

$$\mu_{Sedang}(y) = \begin{cases} \frac{y-250}{375-250}, & 250 \leq y \leq 375.5 \\ \frac{500-y}{500-375.5}, & 375.5 < y < 500 \\ 0, & \text{lainnya} \end{cases} \quad (3.12)$$

$$\mu_{Sedang}(144) = 0 \text{ (Karena } 144 < 250)$$

3) Jauh

$$\mu_{tinggi}(y) = \begin{cases} 0, & y \leq 450 \\ \frac{y-450}{500-450}, & 450 < y < 500 \\ 1, & y \geq 500 \end{cases} \quad (3.13)$$

$$\mu_{tinggi}(144) = 0 \text{ (Karena } 144 \leq 450)$$

2) Evaluasi Aturan dan Inferensi

Identifikasi Aturan yang Aktif

Dari aturan *fuzzy*:

Aturan ke-4 = *IF Distance Sedang AND Sanity Rendah THEN Chase=6*

Maka :

$$\text{Derajat Aktivasi } (\alpha) = \min((0.75, 1) = 0.75 \quad (3.14)$$

3) Perhitungan Output dan Defuzzifikasi

Hitung Output Menggunakan Weighted Average

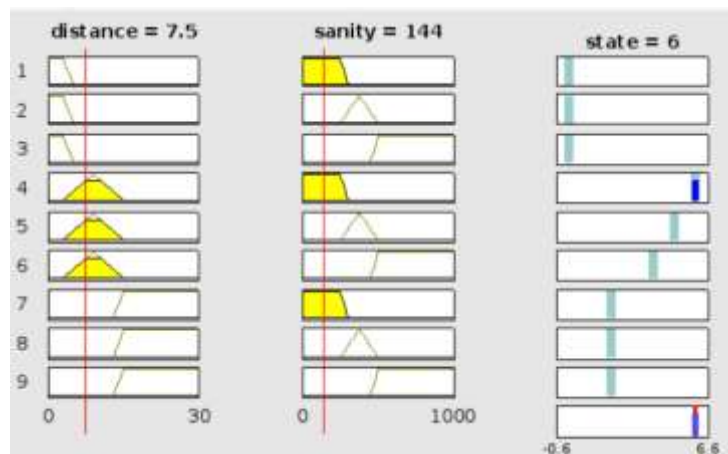
Aturan ke-4 (α) = 0.75 dan $z = 6$

$$\text{Output} = \frac{(0.75 \times 6)}{0.75} = 6 \quad (3.15)$$

Dengan $\text{Distance} = 7.5$ dan $\text{Sanity} = 144$, speed AI Enemy adalah 6 berada di state Chase3 .

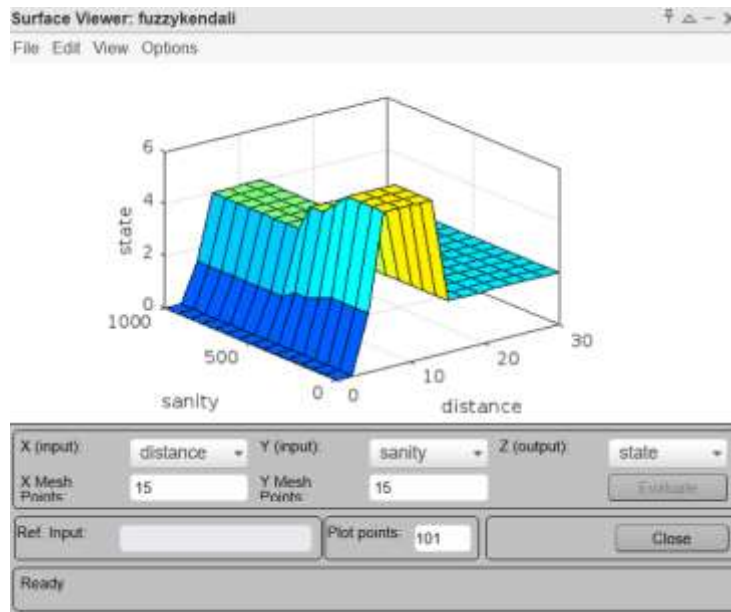
4) Pengujian pada matlab

Berikut gambar hasil pengujian di Matlab jika $\text{distance} = 7.5$ dan $\text{sanity} = 144$ maka $\text{state} = 6$ berada di state Chase3 , dapat dilihat pada gambar 3.7:



Gambar 3. 7 Pengujian Defuzzifikasi Pada Matlab

Berikut tampilan Surface (grafik permukaan) digunakan untuk memvisualisasikan hubungan antara distance, sanity dan state dalam sistem fuzzy.



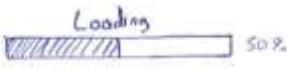
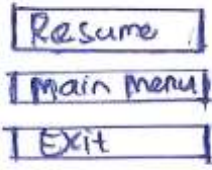
Gambar 3. 8 Grafik Permukaan Pada Matlab

3.13 Perancangan Antarmuka (User Interface)

Perancangan antarmuka bertujuan menggambarkan tampilan game sebelum dibuat. Desain disesuaikan dengan tema *horor The Unread* agar mendukung suasana tegang dan tetap mudah digunakan pemain.

Tabel 3. 2 Perancangan Antarmuka (User Interface)

No	Tampilan	Keterangan
1		Tampilan menu utama menampilkan antarmuka sederhana dengan tiga opsi utama: <i>New Game</i> untuk memulai permainan dari awal, <i>About</i> untuk melihat informasi singkat tentang <i>game</i> , dan <i>Quit Game</i> untuk keluar dari permainan.
2		Tampilan <i>About</i> yang berisi informasi kontrol seperti tombol WASD untuk bergerak, Mouse untuk melihat sekeliling, <i>C</i> untuk berlari, <i>E</i> untuk interaksi, dan <i>g</i> untuk membuang item.

No	Tampilan	Keterangan
3		Tampilan <i>loading screen</i> atar <i>scane</i> , jadi sebelum <i>load scane</i> selanjut menjalan <i>loading screen</i> terlebih dahulu.
4		Tampilan <i>Pause game</i> akan muncul saat <i>player</i> menekan <i>ESC</i> saat permainan dimulai, pada pause game ada 3 <i>button</i> yaitu <i>resume</i> untuk melanjutkan permainan , main menu untuk kembali ke menu dan <i>exit</i> untuk keluar dari game.
5		Tampilan <i>UI</i> interaksi seperti pada pintu meja, kertas mau pun item, saat jarak tertentu atau dekat dengan benda yang bisa di interaksi maka akan muncul tampilnya.
6		Tampilan <i>Sanity</i> sebagai indikator rasa ketakutan <i>player</i> semakin kecil <i>sanity</i> yang di miliki maka semakin gelap pandangan <i>player</i> .
7		Tampilan <i>Game Over</i> akan muncul saat <i>player</i> tertangkap atau ke <i>jumpscare</i> maka akan muncul tampilan <i>game over</i> .
8		Tampilan <i>Ending</i> akan muncul apabila menyelesaikan permainan atau menemukan <i>side ending</i> .

3.14 Perancangan Kontrol

Perancangan kontrol bertujuan menjelaskan tata letak dan fungsi tombol yang digunakan pemain untuk mengendalikan karakter dalam game. Karena *The Unread* merupakan game desktop, kontrol dirancang menggunakan kombinasi keyboard dan mouse agar mendukung navigasi dan interaksi secara optimal.

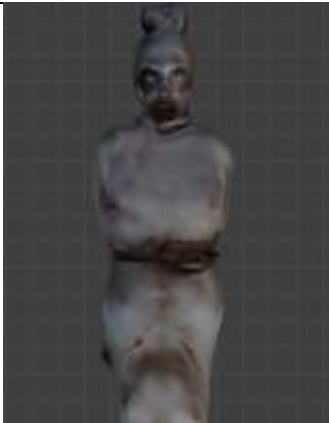
Tabel 3. 3 Perancangan Kontrol

No	Tampilan	Keterangan
1		W, A, S, D digunakan untuk menggerakkan karakter: W untuk maju, A untuk bergerak ke kiri, S untuk mundur, dan D untuk bergerak ke kanan.
2		C digunakan untuk berlari, memungkinkan karakter bergerak lebih cepat terutama saat menghindari musuh atau berpindah lokasi.
3		D digunakan untuk membuang item yang sedang dibawa dari inventori.
4		E digunakan untuk berinteraksi dengan objek atau mengambil item yang ditemukan.
5		Esc digunakan untuk membuka Pause Menu, menghentikan permainan sementara, mengakses pengaturan.
6		Mouse digunakan untuk mengarahkan pandangan kamera sesuai pergerakan mouse, memudahkan pemain mengamati lingkungan sekitar.

3.15 Desain Karakter

Desain karakter merupakan salah satu aspek penting dalam pengembangan game, terutama pada *genre* horor yang di gunakan sebagai aspek *jumpscare* mau pun sebagai *hunter*, Karakter dibuat sendiri menggunakan *addon blender* seperti *MB lab*, *Make Human* dan menggunakan *meshy*, dan untuk *rigging* nya menggunakan *mixamo*.

Tabel 3. 4 Desain Karakter Game The Unread

No	Karakter	Keterangan
1		Tuyul, Muncul pada jumpscare diawal permain pada level 1 dan level 2, karakter tidak menyerang langsung pemain
2		Ibu Player, Muncul pada <i>prolog</i> dan dari level awal hingga ending, karakter mempengaruhi <i>happy</i> atau <i>bad ending</i> di level akhir
3		Pocong, Sering muncul untuk jumpscare ringan, karakter tidak menyerang langsung pemain.

No	Karakter	Keterangan
4		Kuntilanak, muncul pada level 2, 3 dan 5 sebagai game over apabila pemain keluar tanpa menyelesaikan puzzel
5		Kuntilanak 2, Sama seperti sebelumnya akan tetapi kunti ini lebih agresif dari kuntilanak, muncul pada level 3 dan 4

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4. 1 Hasil Implementasi

Pada bab ini dijelaskan implementasi dari game *The Unread* , yaitu sebuah game horor dengan elemen puzzle, *jumpscare*, dan sistem *sanity*. Implementasi dilakukan dengan menggunakan *Unity Engine* serta bahasa pemrograman C# untuk logika permainan.

4.1.1 Hasil Halaman Utama Game

1. Menu awal game

Menu utama terdiri dari tiga pilihan, yaitu *Start* untuk memulai permainan dan langsung masuk ke level pertama, *About* yang berisi informasi kontrol seperti tombol WASD untuk bergerak, Mouse untuk melihat sekeliling, C untuk berlari, E untuk interaksi, dan g untuk membuang item, serta *Exit* yang digunakan untuk keluar dari game dan menutup aplikasi.



Gambar 4. 1 Tampilan Menu

2. Kontrol Karakter

About yang berisi informasi kontrol seperti tombol WASD untuk bergerak, *Mouse* untuk melihat sekeliling, C untuk berlari, E untuk interaksi, dan g untuk membuang item.



Gambar 4. 2 Tampilan Kontrol Karakter

3. Cutscene

Cutscene untuk pengantar cerita lalu akan meng *load scene* selanjutnya, Disini menggunakan activation untuk *TMPro* dan *load scene* berikut nya setelah prolog.



Gambar 4. 3 Tampilan Cut Scene

4. Pause Game

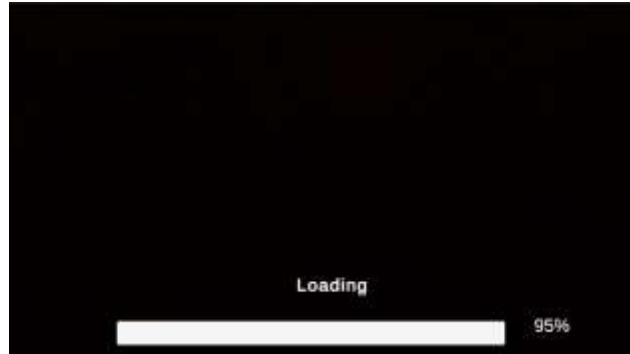
Pause game akan muncul saat player menekan ESC saat permainan dimulai, pada pause game ada 3 button yaitu resume untuk melanjutkan permainan , main menu untuk kembali ke menu dan *exit* untuk keluar dari game.



Gambar 4. 4 Tampilan Pause Game

5. Transisi Level

Transisi level disini berupa loading screen atau scene, jadi sebelum load scene selanjut berjalan *loading screen* terlebih dahulu, disini menggunakan *TMP Pro* dan juga *sellinder* sebagai animasi loading nya.



Gambar 4. 5 Tampilan Loading Scareen

4.1.5 Hasil Game Level 1

1. Envirovement

Envirovment Rumah ada 2 yang pada scene utama yaitu rumah utama dan gudang, scene ini untuk level 1, 2 dan 5, tidak *enemy* pada scene ini, ini sebagai prolog dan penentu ending dalam game.



Gambar 4. 6 Tampilan Envirovment Level 1, 2 dan 5

2. Pintu Depan

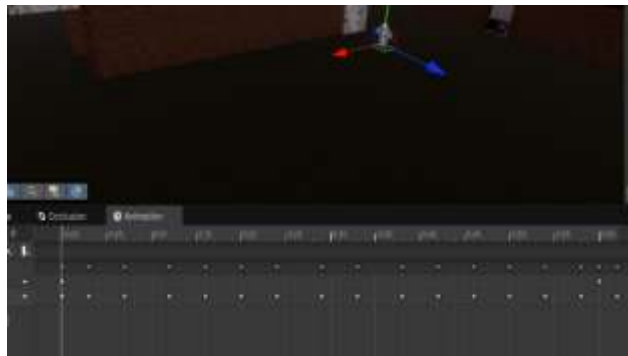
Pintu depan menggunakan interaksi *door lock* ini pintu ini terkunci dan akan menampilkan dialog saat kondisi pintu terkunci dan pintu ini juga di gunakan sebagai kabur dari dalam untuk *trigger neutral ending*.



Gambar 4. 7 Tampilan Pintu depan

3. Jumpscare Tuyul

Jumpscare tuyul menggunakan animation, muncul saat player berada di box trigger maka objek tuyul akan aktif dan langsung menjalankan jumpscare tuyul berupa animasi pergerakan.



Gambar 4. 8 Tampilan Jumpscare Tuyul

4. Pintu Gudang

Pintu menggunakan block, jika *player* berinteraksi dengan pintu maka akan masuk kondisi terkunci maka *player* harus memindahkan block terlebih dahulu untuk merubah status pintu jadi open.



Gambar 4. 9 Tampilan Pintu Gudang

5. Pintu Belakang

Pintu belakang jika di interkasi maka akan muncul status membutuhkan kunci, dan *player* diwajibkan mencari kunci untuk membuka pintu, item kunci berada di gudang, ini satu satunya jalan untuk melanjutkan cerita.



Gambar 4. 10 Tampilan Pintu Belakang

6. Saklar

Saklar digunakan sebagai pemicu untuk menyalakan lampu di area gudang. Saat sudah melewati puzzle pintu, saklar ini sebagai pemuncu objek lampu.



Gambar 4. 11 Tampilan Saklar Lampu

7. Triger Pocong

Objek pocong dalam game akan bereaksi terhadap posisi pemain. Jika jarak antara objek pocong dan pemain terdeteksi kurang dari 5, maka sistem akan memicu perintah untuk menonaktifkan objek pocong tersebut.



Gambar 4. 17 Tampilan Objek Pocong

8. Catatan

Catatan termasuk *inspect* item dan berbeda dengan item, *inspect* item digunakan untuk mengrotate dan zoom suatu objek, catatan ini juga sebagai *clue* untuk jalan cerita.



Gambar 4. 12 Tampilan Inspect Item Catatan

9. Item kunci

Item kunci berfungsi untuk membuka pintu belakang dan melanjutkan permainan. kunci ini disimpan di dalam gudang, tepatnya di laci paling atas sebuah lemari.



Gambar 4. 13 Tampilan Item Kunci

10. Membuka Pintu belakang

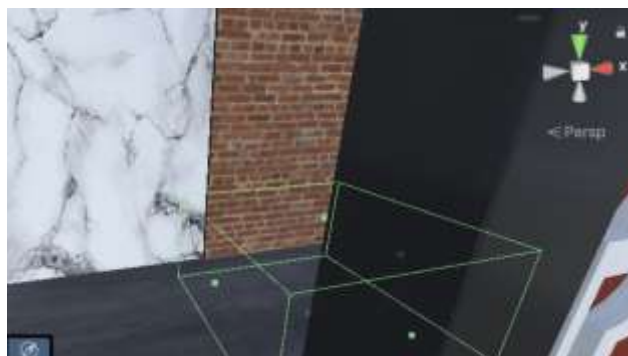
Pintu belakang yang awalnya terkunci hanya bisa dibuka menggunakan item kunci. Kunci tersebut berada di gudang dan disimpan di laci paling atas. player harus menemukannya untuk bisa melanjutkan ke area berikutnya.



Gambar 4. 14 Tampilan Membuka Pintu Belakang

11. Triger Pintu Belakang

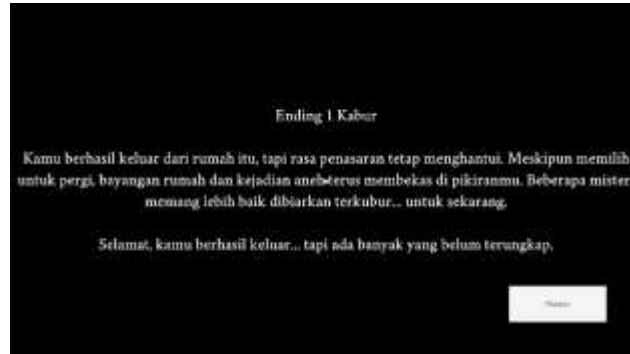
Triger pintu belakang ini digunakan saat *player* sudah memasuki rumah dan saat *player* berada di box maka event akan di jalan kan seperti pintu tertutup.



Gambar 4. 15 Tampilan Box Collider Trigger Pintu Belakang

12. Ending Netral

Pada level 1 jika *player* tidak lanjut ke level 2 bisa mendapatkan ending netral atau ending kabur yang dapat di capai saat *player* keluar menggunakan pintu depan.



Gambar 4. 16 Tampilan Ending Netral

4.1.6 Hasil Game Level 2

1. CutScane Kamar

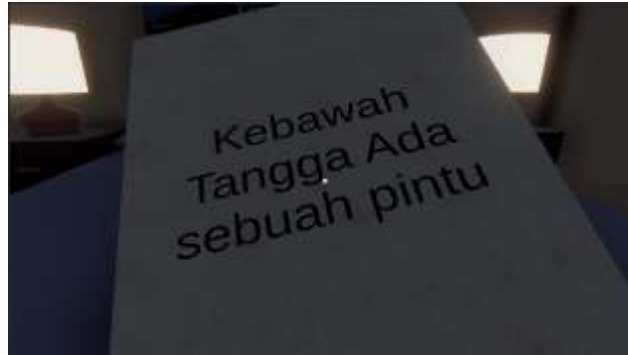
Cut scane level 2 sebagai pemisah antara level 1 dan 2, dan saat level 2 di mulai saat pemain bangun tidur dan melihat sosok ibu, ditambah dengan sedikit percakapan.



Gambar 4. 17 Tampilan CutScane Level 2

2. Catatan ibu

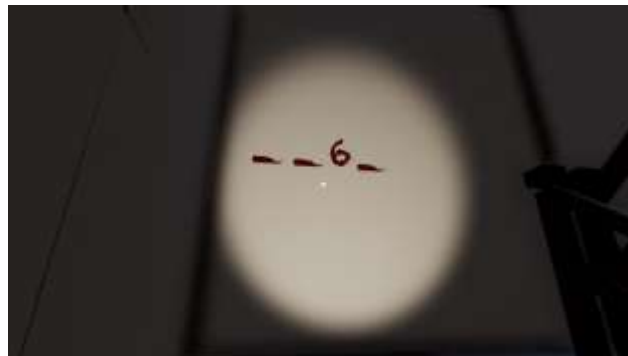
Catatan termasuk inspect item, catatan disini berupa informasi untuk ke level selanjut nya, yaitu berada dibawah tangga ada sebuah pintu pintu tersembunyi untuk lanjut ke level selanjutnya.



Gambar 4. 18 Tampilan Catatan di kamar ibu

3. Coretan kode

Saat sudah melalui *cutscene* level 2 maka di sepanjang ujung tangga ada sebuah coretan kode yang berupa informasi untuk menyelesaikan puzzle dibawah tangga.



Gambar 4. 19 Tampilan Coretan Kode

4. Keypad

Pada tangga ada tempat tersembunyi, dan ada sebuah pintu yang harus dibuka dengan kode, *keypad* disini digunakan untuk membuka pintu ke level selanjutnya, kode nya adalah semua coretan yang ada di dinding.



Gambar 4. 20 Tampilan Keypad

5. Game Over

Saat *player* mencoba membuka pintu dalam kondisi masih terkunci, maka akan langsung memicu adegan jumpscare yang menegangkan, dan mengakhiri permainan dengan kondisi *game over*.



Gambar 4. 21 Tampilan Jumpscare Game Over 1 level 2

6. Pintu next scane

Setelah memasukkan kode yang benar pada keypad, pemain akan dapat membuka pintu yang sebelumnya terkunci, sehingga memungkinkan mereka untuk melanjutkan permainan dan masuk ke level berikutnya.



Gambar 4. 22 Tampilan Pintu Bawah Tanah

4.1.7 Hasil Game Level 3

1. Envirovment Ruang Bawah Tanah

Envirovment Rumah Bawah Tanah adalah scane 2 untuk level 3, akan memasuki apabila sudah menyelesaikan kan *puzzel* pada level 1 dan disini sudah ada *enemny* apabila tertangkap maka *game over*.



Gambar 4. 23 Envirovment Ruang Bawah Tanah

2. Pintu ke level 4

Pintu menuju level selanjutnya, pemain harus menyelesaikan puzzle yaitu mengumpulkan dua item yang tersebar dan sembari menghindari tangkapan kunti.



Gambar 4. 24 Tampilan pintu ke level 4

3. Item Baterai

Item baterai disini digunakan untuk membuka, pemain harus mengumpulkan 2 baterai yang tersebar di ruangan dan sembari menghindari kejaran dari kunti.



Gambar 4. 25 Tampilan Item Baterai

4. Game over 1

Game over 1 pada level 3 akan terjadi apabila pemain tertangkap oleh kunti, saat *game over* akan berjalan animasi tertangkap terlebih dahulu lalu akan menampilkan *game over* lalu mengulang pada level 3.



Gambar 4. 26 Tampilan Jumpscare Game Over 1 Level 3

5. Game Over 2

Game over 2 terjadi apabila pemain kabur kembali ke pintu sebelumnya tanpa menyelesaikan puzzle level 3, saat *game over* akan berjalan animasi tertangkap terlebih dahulu lalu akan menampilkan *game over* lalu mengulang pada level 3.

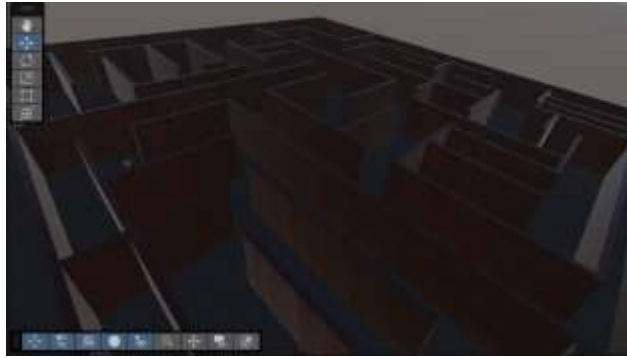


Gambar 4. 27 Tampilan Jumpscare Kunti Game Over 2 Level 3

4.1.8 Hasil Game Level 4

1. Envirovment Labirin

Envirovment labirin adalah scane 3 untuk level 4, dan disini sudah ada *enemny* , akan memasuki apabila *player* sudah menyelesaikan *puzzel* level 3.



Gambar 4. 28 Tampilan Envirovment Labirin

2. Game over

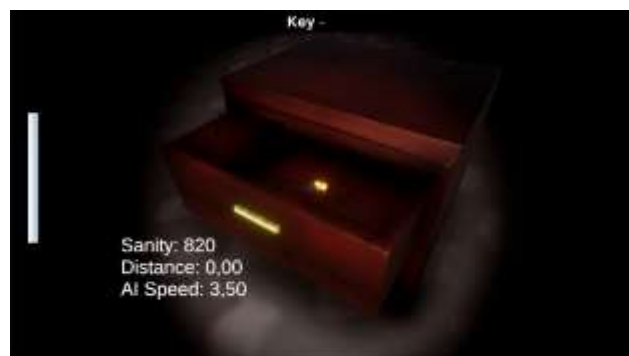
Game over 1 pada level 4 akan terjadi saat pemain tertangkap oleh kunti, saat *game over* akan berjalan animasi tertangkap terlebih dahulu lalu akan menampilkan *game over* lalu mengulang pada level 4.



Gambar 4. 29 Tampilan Jumpscare Game Over Level 4

3. Item Kunci

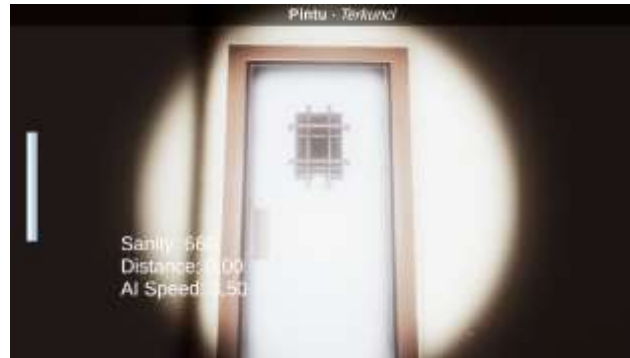
Item kunci digunakan untuk membuka pintu dan kembali ke atas. Item ini merupakan syarat penting yang harus ditemukan oleh pemain agar dapat melanjutkan permainan.



Gambar 4. 30 Tampilan Kunci level 4

4. Kembali ke atas

Setelah mendapat *item* kunci maka pemain harus kembali ke awal masuk level 4 menuju pintu sembari menghindari kejaran kunti, apabila sampai maka pintu akan terbuka dan akan masuk ke level selanjutnya.



Gambar 4. 31 Tampilan Pintu Ke Level 5

4.1.9 Hasil Game Level 5

1. Cutscene Level 5

Awal level dimulai dengan *cut scene* level 5 di mulai dari pemain terbangun dari tidur, pemain yang kebingungan lalu langsung turun ke lantai bawah.



Gambar 4. 32 Tampilan Cut Scane Level 5

2. Game over

Game over pada level terjadi apabila pemain salah memasuki ruangan, saat *game over* akan berjalan animasi tertangkap terlebih dahulu lalu akan menampilkan *game over* lalu mengulang pada level 5.



Gambar 4. 33 Tampilan Jumpscare Game Over Level 5

3. Penentu Ending

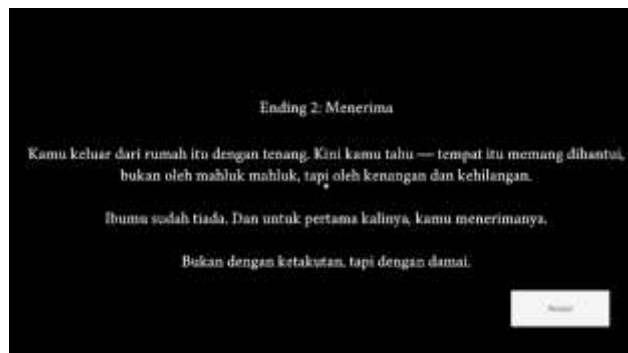
Penentu ending ini akan terjadi apabila pemain mendekati pintu depan dan ingin keluar, maka ibu akan muncul, dan pemain memiliki 2 pilihan, keluar aatau menetap.



Gambar 4. 34 Tampilan Penentu Ending

4. Good Ending

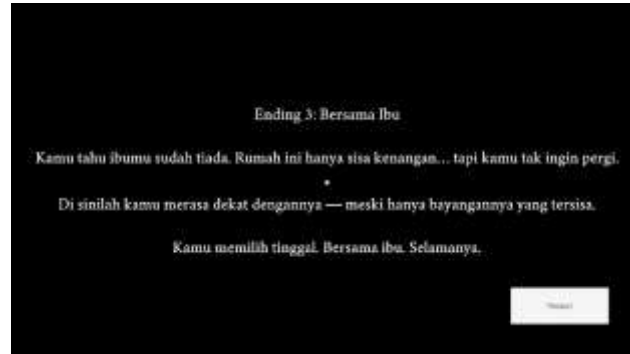
Good ending akan muncul apabila pemain mengabaikan semua termasuk semua kenangan bahkan menginggal ibu sendiri dan tetap keluar dari rumah lewat pintu depan.



Gambar 4. 35 Tampilan Good Ending

5. Bad Ending

Bad ending akan muncul apabila pemain lebih memilih tetap tinggal di rumah dengan cara mendekati ibu, dan akan terjebak selamanya di dalam rumah.



Gambar 4. 36 Tampilan Bad Ending

4.2 Pengujian

Setelah melakukan implementasi sistem, maka sistem selanjutnya akan diuji coba. Uji coba sistem dilakukan dengan menjalankan implementasi sistem yang telah selesai dengan tujuan apakah fungsi sudah sesuai dengan harapan apa tidak.

4.2.1 Pengujian Sistem Finite State Machine (FSM)

FSM digunakan untuk mengatur perilaku AI musuh, memiliki 3 kondisi yaitu *wander*, *case*, *jumpscare*

1. Kondisi Wander

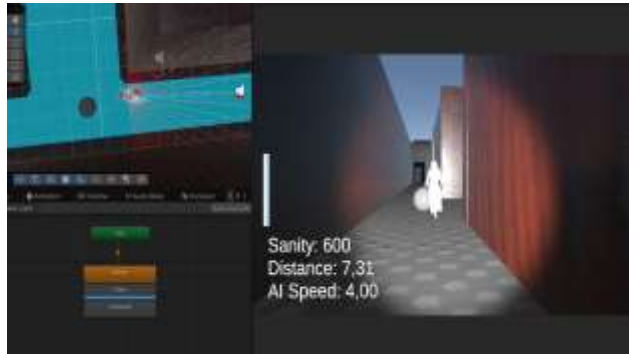
Kondisi *wander* atau patroli akan terjadi apabila tag player tidak berada didalam jangkauan atau tidak terdeteksi maka *AI Hunter* akan ke *Wander Point* hingga *player* terdeteksi.



Gambar 4. 37 Kondisi Wander

2. Kondisi Chase

Kondisi Chase atau mengejar akan terjadi apabila tag player di temukan dan berada di jangkauan penglihatan *AI Hunter* atau terdeteksi, akan lepas apabila jarak *hunter* lebih dari 15.



Gambar 4. 38 Kondisi Chase

3. Kondisi Jumpscare

Kondisi jumpsacre terjadi apabila jarak player dengan AI hunter kurang dari 3 maka akan menjalan animasi tertangkap terlebih dahulu lalu akan menampilkan *game over* lalu mengulang pada level awal.



Gambar 4. 39 Kondisi Jumpscare

Kesimpulan: *FSM* berpindah *state* sesuai logika yang diatur. Perubahan state dapat diamati secara langsung dalam gameplay

4.2.2 Pengujian Fuzzy

Fuzzy Logic digunakan untuk menentukan output kecepatan musuh dan perubahan state chase berdasarkan dua input utama jarak ke pemain dan Sanity pemain. Pengujian dilakukan dengan dua pendekatan yaitu unity dan matlab.

1. Fuzzy pada unity

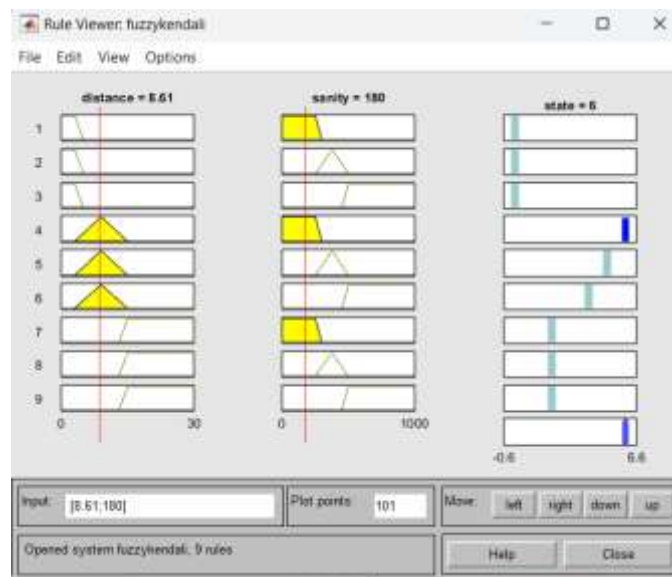
Kondisi *fuzzy* pada unity berupa *TMP pro*, yang menampilkan input dan juga output. Pada tampilan input *sanity* nya : 180 dan *distance* nya : 8,61 menghasilkan output : 6.



Gambar 4. 40 Pengujian Fuzzy di Unity

2. Fuzzy pada Matlab

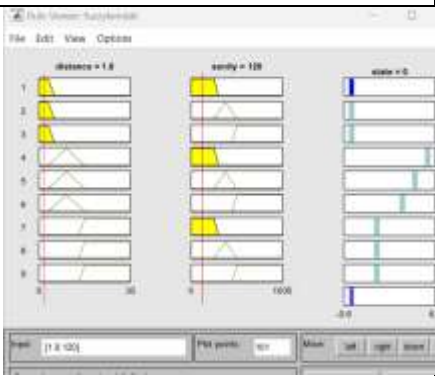
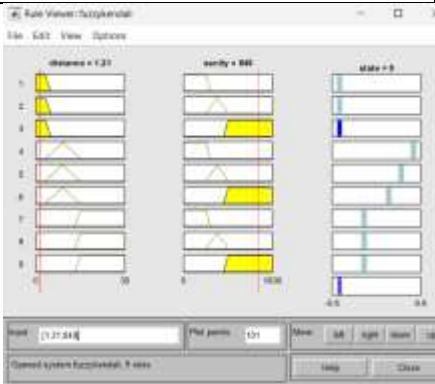
Fuzzy pada matlab berisi input dan output, untuk input *sanity* nya : 180 dan *distance* nya : 8,61 menghasilkan output : 6.

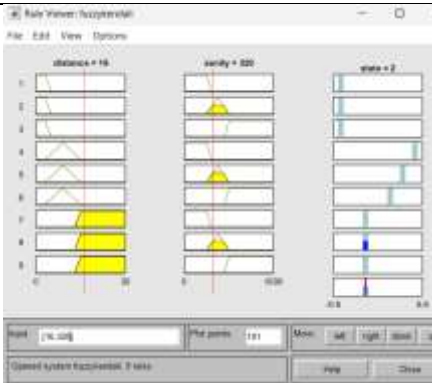


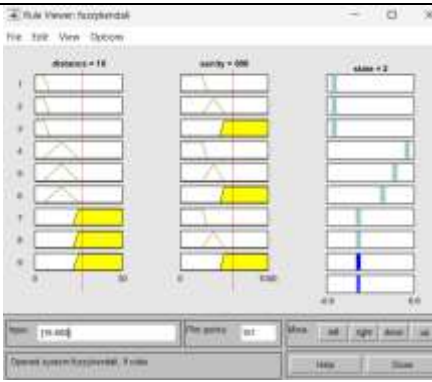
Gambar 4. 41 Pengujian Fuzzy di Matlab

Tabel berikut menunjukkan beberapa skenario pengujian yang dilakukan untuk membandingkan hasil output logika *fuzzy* antara implementasi di *Unity* dan *MATLAB*.

Tabel 4. 1 Pengujian Fuzzy

No	Unity	Matlab	Hasil
1			x:1,80 y:120 Output: 0
2			x: 2,40 y: 320 Output: 0
3			x: 1,21 y: 840 Output: 0
4			x: 7,88 y:180 Output: 6

No	Unity	Matlab	Hasil
5			x: 5,43 y: 300 Output: 5
6			x: 7,47 y: 760 Output: 4
7			x: 16 y: 40 Output: 2
8			x: 16 y: 320 Output: 2

No	Unity	Matlab	Hasil
9			x: 16 y: 660 Output: 2

Kesimpulan: Hasil keluaran fuzzy logic pada game sama dengan hasil perhitungan di MATLAB, baik dari segi kecepatan maupun kondisi akhir

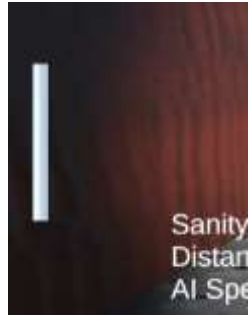
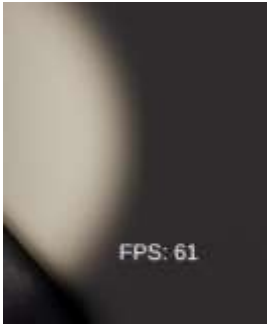
4.2.3 Pengujian Fungsional

Tabel berikut menunjukkan beberapa skenario pengujian yang dilakukan:

Tabel 4. 2 Pengujian Fungsional

NO	Fitur Yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
1.	Kontrol Pemain	Pemain dapat bergerak maju, mundur, kanan, kiri, lompat, dan berinteraksi objek	Semua kontrol dapat dijalankan dengan lancar	Sukses 
2.	AI Hunter (FSM)	Musuh mengejar pemain jika terdeteksi, kembali patroli jika kehilangan jejak	AI berpindah state sesuai kondisi yang ditentukan	Sukses 

NO	Fitur Yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
3.	AI Hunter (Fuzzy)	Saat mengejar player kecepatan enemy terpengaruh terhadap sanity dari player	Speed enemy terhadap sanity sesuai dengan perhitungan fuzzy	Sukses 
4.	Transisi Level	Setelah menyelesaikan puzzle, pemain berpindah ke level berikutnya	Transisi berjalan	Sukses 
5.	Jumpscare	Pemicu jumpscare aktif jika pemain memasuki area tertentu	Efek jumpscare muncul	Sukses 
6.	Interaksi Objek Puzzle	Pemain bisa mengambil item dan menggunakannya untuk menyelesaikan puzzle	Puzzle bisa diselesaikan dan memicu perubahan game state	Sukses 

NO	Fitur Yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Status
7.	UI Sanity	UI bar sanity	UI tampil nilai dengan benar dan responsif	Sukses 
8.	Performa	Game tidak mengalami lag atau crash selama dimainkan di beberapa device uji	FPS stabil, tidak ada freeze, tidak crash	Sukses 

4.2.3 Pengujian User

Pengujian user bertujuan untuk mengetahui sejauh mana game dapat diterima oleh pemain dari dari kontrol, suasana horor, efek suara dan jumpscare dan keseluruhan game.

Tabel 4. 3 Pengujian user

No	Pertanyaan	Skala Jawaban				
		SK	B	C	K	SK
1	Game berjalan lancar tanpa lag berat pada perangkat saya	16	7	4	2	2
2	Kontrol karakter mudah dipahami dan digunakan	20	6	3	2	0
3	Visual dan pencahayaan mendukung suasana horor	14	9	6	1	1

No	Pertanyaan	Skala Jawaban				
		SK	SK	SK	SK	SK
4	Efek suara dan jumpscare menciptakan atmosfer menyeramkan	14	9	6	1	1
5	Pergerakan AI musuh terasa logis berdasarkan posisi dan sanity pemain	18	7	3	3	0
6	Secara keseluruhan, saya menikmati memainkan game ini	15	8	5	2	1
	Total	97	46	27	11	5

Keterangan

Banyak pertanyaan : 6

Banyak user : 31

Pembagi : $6 * 31 = 186$

Tabel 4. 4 Persentase Pengujian User

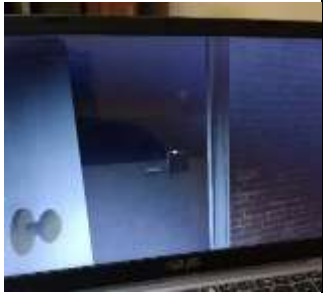
No	Persentase	Nilai
1	Persentase menjawab sangat baik	$97/186 * 100\% = 52.15\%$
2	Persentase menjawab baik	$46/186 * 100\% = 24.73\%$
3	Persentase menjawab cukup	$27/186 * 100\% = 14.52\%$
4	Persentase menjawab kurang	$11/186 * 100\% = 5.91\%$
5	Persentase menjawab sangat kurang	$5/186 * 100\% = 2.69\%$

Berdasarkan hasil pengujian yang dilakukan terhadap 31 user dengan total 186 jawaban, diperoleh bahwa sebagian besar responden memberikan penilaian sangat baik (52,15%) dan baik (24,73%) terhadap game yang diuji. Hal ini menunjukkan bahwa game secara umum telah diterima dengan baik oleh pemain, terutama dari segi performa, kontrol, suasana horor, efek suara, serta logika pergerakan AI. Hanya sebagian kecil yang memberikan penilaian cukup (14,52%), kurang (5,91%), dan sangat kurang (2,69%), yang menunjukkan bahwa masih terdapat beberapa aspek minor yang bisa ditingkatkan dalam pengembangan selanjutnya.

4.2.4 Spesifikasi Perangkat yang Digunakan

Tabel 4. 5 Spesifikasi perangkat yang digunakan

NO	Prosesor	RAM	GPU	OS	Hasil
1	Ryzen 7 5000u	8GB	AMD Radeon ™ Vega 7 Graphics	Windo ws 11 64-bit	Berjalan 30-60 fps 
2	I5- 7200U	8GB	Nvidia GEFOR CE 930mx	Windo ws 10 64-bit	Berjalan 25–40 fps 
3	I5- 8250u	8GB	Intel uhd graphic	Windo ws 11 64 bit	15-30 – aga berat 
4	Ryzen 7 4800H	16GB	NVIDIA GeForce RTX 3050	Windo ws 11 64 bit	60fps – stabil 

NO	Prosesor	RAM	GPU	OS	Hasil
5	AMD Ryzen 5600H	8GB	AMD Radeon Vega 7	Windows 11 64 bit	50-60fps – cukup stabil 
6	i5 11400H	8GB	RTX 3050	Windows 11 64 bit	60fps – sangat stabil 
7	Ryzen 5 5500U	16GB	Radeon Vega 7	Windows 11 64 bit	30–50 fps – Cukup stabil 
8	Ryzen 7 3700X	16GB	Nvidia RTX 3060Ti	Windows 11 64 bit	60 fps – sangat stabil 
9	Intel Celeron N4500	8GB	Intel UHD Graphics	Windows 10 64-bit	10–20 fps sangat buruk 

NO	Prosesor	RAM	GPU	OS	Hasil
10	Intel Core i3-1115G4	8GB	Intel UHD Graphics	Windows 11 64 bit	20–35 fps – agak berat 
11	AMD Ryzen 3 3250U	8GB	AMD Radeon™ Vega 3 Graphics	Windows 11 64 bit	20–30 fps – Agak berat 
12	Intel Celeron N2830	8GB	Intel UHD Graphics	Windows 11 64 bit	5–15 fps – sangat buruk 
13	Ryzen 5 5600	16GB	GTX 1660 Super	Windows 11 64 bit	60 fps – sangat stabil 

4.2.5 Rekomendasi Spesifikasi Sistem

Untuk memastikan game dapat berjalan dengan lancar dan memberikan pengalaman bermain yang optimal, berikut ini adalah rekomendasi spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan. Spesifikasi dibagi menjadi dua kategori: Minimum dan Rekomendasi.

1. Spesifikasi Minimum

Spesifikasi ini memungkinkan game dijalankan dengan kecepatan frame sekitar 25–40 FPS.

- Prosesor: Intel Core i5-7200U / AMD Ryzen 5 2500U atau setara
- RAM: 8 GB
- GPU: NVIDIA GeForce 930MX / AMD Radeon Vega 7 / Intel UHD Graphics
- Penyimpanan: Minimal 1 GB ruang kosong
- Sistem Operasi: Windows 10 64-bit

2. Spesifikasi Rekomendasi

Spesifikasi ini mendukung gameplay stabil tanpa drop dan kecepatan frame stabil di atas 60 FPS..

- Prosesor: Intel Core i5-11400H / AMD Ryzen 7 4800H atau setara
- RAM: 16 GB
- GPU: NVIDIA GeForce RTX 3050 atau lebih tinggi
- Penyimpanan: SSD dengan minimal 2 GB ruang kosong
- Sistem Operasi: Windows 11 64-bit

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang dilakukan, dapat disimpulkan bahwa seluruh tujuan pengembangan game "*The Unread*" telah tercapai dengan baik:

1. Penerapan metode *Finite State Machine (FSM)* berhasil mengatur perilaku musuh dalam tiga kondisi utama yaitu *wander*, *chase*, dan *jumpscare*. Pengujian menunjukkan bahwa transisi antar *state* berlangsung sesuai logika yang diharapkan dan dapat diamati langsung dalam gameplay, sesuai dengan tujuan pertama.
2. Integrasi logika *fuzzy Sugeno* ke dalam *FSM* terbukti berhasil dalam menghasilkan perilaku musuh yang dinamis dan adaptif terhadap kondisi pemain. Hasil pengujian menunjukkan bahwa output *fuzzy* pada Unity dan MATLAB menghasilkan nilai kecepatan dan state yang sama, membuktikan keakuratan serta konsistensinya, sejalan dengan tujuan kedua.
3. Mekanisme eksplorasi dan investigasi yang mendukung alur cerita berhasil dibangun melalui fitur interaksi objek, puzzle, dan sistem transisi antar level. Hal ini terbukti dari pengujian fungsional yang menunjukkan bahwa seluruh fitur berjalan dengan baik dan mendukung keberlangsungan gameplay, mencerminkan tercapainya tujuan ketiga.
4. Optimalisasi elemen visual, audio, dan desain level menghasilkan pengalaman bermain yang imersif dan menegangkan. Berdasarkan pengujian user, sebesar 76,88% responden memberikan penilaian “sangat baik” dan “baik”, khususnya dalam aspek kontrol, atmosfer horor, efek suara, serta logika AI musuh. Hasil ini menunjukkan bahwa tujuan keempat tercapai dengan respons positif dari sebagian besar pengguna akhir, meskipun masih terdapat ruang untuk perbaikan pada beberapa aspek minor.

5.2 Saran

Adapun saran untuk pengembangan lebih lanjut dari game dan penelitian ini adalah sebagai berikut:

1. Optimasi performa perlu dilakukan untuk meningkatkan kelancaran pada perangkat dengan spesifikasi rendah, khususnya pada bagian efek pencahayaan, suara, dan transisi level.
2. Tingkat kesulitan *AI* dapat dikembangkan lebih lanjut dengan menambahkan elemen pembelajaran atau adaptive difficulty berdasarkan progres pemain, serta pengayaan *rule fuzzy* untuk respons yang lebih kompleks.
3. Game dapat dikembangkan menjadi lebih menarik dengan menambahkan fitur *inventory*, *multiple ending*, *voice acting*, dan sistem *save/load* agar pengalaman bermain lebih imersif.
4. Sistem Fuzzy Logic yang digunakan dalam AI dapat ditingkatkan dengan mengimplementasikan metode Mamdani untuk menghasilkan perilaku musuh yang lebih dinamis dan responsif terhadap kondisi permainan secara real-time. Hal ini juga dapat meningkatkan kompleksitas dan realisme dalam interaksi AI terhadap pemain.

DAFTAR PUSTAKA

- [1] Henrik Engstrom "Game Development Research" ISBN 978-91-984918-7-6, 978-91-984918-8-3 (2020)
- [2] Anang Habibi, Muhammad Ibnu Athoillah "Pengembangan Game Survival Horror Unity 3D dengan Menerapkan AI pada NPC" Vol. 4 No. 1 (2025)
- [3] Febrina Ade Susianti, Anik Vega Vitianingsih, Achmad Chiron, Dwi Cahyono, Anggit Wikaningrum "Aplikasi Game Survival Horror 3d Pencegahan Penculikan Anak Remaja Berbasis Dekstop" Vol. 7 No. 2 (2024)
- [4] Zefid Frenda Fathoni, Moh Ahsan, Muhammad Priyono Tri Sulistyanto "Penerapan Fuzzy Sugeno Pada Npc Rintangan Dalam Game “Ana Cerita Extreme” Menggunakan Unity" Vol. 5 No. 2 (2023)
- [5] Muhammad ikmal Ikmal "Game Edukasi Menyusun Kata Untuk Meningkatkan Pemahaman Anak Dengan Menggunakan Metode MDLC" Vol. 4 No. 1 (2024)
- [6] Setta Ramadaniati, Dian Ahkam Sani, Mochammad Firman Arif "Rancang Bangun Mobile Game Adventure Of Studies Sebagai Media Pembelajaran" Vol 6, No 1 (2021)
- [7] Geraldo Clieffer Lawento, Sumenge T.G. Kaunang, Virginia Tulenan "Rancang Bangun Game Edukasi Genre Action Mitigasi Bencana Alam Tsunami" Vol. 16 No. 2 (2021)
- [8] Mohammad Andrean Jaya Aprilianto, Esti Wijayanti, Ahmad Abdul Chamid "Game Adventure of Cakra Versi Cerita Rakyat Indonesia Sebagai Media Pembelajaran Interaktif" Vol. 5 No. 1 (2025)
- [9] Roberto Kaban, Fandy Syahputra, Fajrillah "Perancangan Game RPG (Role Playing Game) “Nusantara Darkness Rises” Volume 2, No. 4, Juli 2021
- [10] Thoriq Ikhwan Alfawas, Abdul Rahim, Rudiman Rudiman "Penerapan Fitur Ekstraksi TF-IDF untuk Analisis Sentimen Ulasan Game Bus Simulator Indonesia dengan Algoritma Naive Bayes" Vol. 4 No. 5 (2024)
- [11] Aris Hadisopiyan, Christian D. Suhendra, Parma Hadi Rantelinggi "Membuat Game 3d Survival Horror “Suanggi Survival Papua” Berbasis Desktop Menggunakan Unity" Vol. 5 No. 3 (2020)
- [12] Sendi Novianto "Game Method Based on Genre Game as Higher Educational

- Learning: Systematic Literature Review with VOSviewer" Vol. 10 No. 1 (2023)
- [13] Fadzlin Ahmadon, Muhammad Zikri Selahuddeen, Elin Eliana Abdul Rahim, Hazlifah Mohd Rusli "Matching Game Genre with Lesson Content – A Development of Blood Circulation Racing Game" Vol. 4 No. 3 (2024)
 - [14] Pujana Paliyawan, Ruck Thawonmas, Kingkarn Sookhanaphibarn, Worawat Choensawat "Audience participation fighting game: Exploring social facilitation for an enhanced APG experience" Volume 10, Issue 2e23967, January 30, 2024
 - [15] Ariesto Hadi "Designing Platformer Game using GameMaker Studio" Vol. 2 No. 1 (2024)
 - [16] Sebastian Lesmana, Octavianus Ariwana, Rudy P Halim, Alexander A S Gunawan "Behavior Correlation between Games in First-Person Shooter Genre based on Personality Traits" Volume 179, (2021)
 - [17] I Made Suandana Astika Pande, Ricky Aurelius Nurtanto Diaz, Rm Addin Ashary Ramadhan "PEMBUATAN HOROR GAME 3D MULTIPLAYER CO-OP “CALONARANG” BERBASIS MOBILE" Vol 5 No 2 (2023)
 - [18] Hafizzudin Sifaulloh, Juniardi Nur Fadila, Fresy Nugroho "Penerapan Metode Finite State Machine pada Game Santri on the Road" Vol. 3 No. 1 (2021)
 - [19] Ahdi Sanjaya, Jusuf Wahyudi, Yode Arliando "Penerapan Logika Fuzzy Sugeno untuk Menentukan Reward pada Game Edukasi Platformer Berbasis Android" Volume 6 Nomor 2 (2022)
 - [20] Danang Rifai, Farid Fitriyadi "Penerapan Logika Fuzzy Sugeno dalam Keputusan Jumlah Produksi Berbasis Website" Vol. 2 No. 2 (2023)
 - [21] Shucy Fadila "Aplikasi Ricezzy Untuk Menghitung Irigasi Tanaman Padi Menggunakan Metode Fuzzy Mamdani" Vol. 2 No. 1 (2022)
 - [22] Muhammad Gausan Izza, Dyah Febria Wardhani "Perkembangan Kecerdasan Buatan Dalam Industri Video Game" Vol. 5 No. 2 (2025)

LAMPIRAN

Lampiran 1 : Formulir Perbaikan Skripsi



FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Abdul Wahid
 NIM : 2118073
 JURUSAN : Teknik Informatika S-1
 JUDUL : PERANCANGAN GAME HORROR 3D "THE UNREAD"
 DENGAN METODE FINITE STATE MACHINE DAN FUZZY
 SUGENO

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	16 Juli 2025	1. Setiap persamaan agar diberi nomor persamaan. 2. Setiap variabel pada persamaan harus diberi keterangan variabel.	
2.	Penguji II	16 Juli 2025	1. Perbanyak responden dan story line 2. Perbaiki laporan	

Dosen Penguji I

Sentot Achmadi, Ir., M.Si., Dr.
 NIP .P. 1039500281

Dosen Penguji II

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

Dosen Pembimbing I

Yosep Agus Pranoto, ST, MT
 NIP .P. 1031000432

Dosen Pembimbing II

Karina Aulia Sari, ST, M Eng
 NIP .P. 1031000426

Lampiran 2 : Surat Keputusan Dosen Pembimbing 1

	PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI FAKULTAS TEKNIK SIPIL DAN PERENCANAAN PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. INI (PERSERO) MALANG BANK NAGA MALANG	Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65146 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417638 Fax. (0341) 417634 Malang

Malang, 18 Maret 2025

Nomor : ITN-972/IIINF/TA/2025
Lampiran : ---
Perihal : Pembimbing Utama Skripsi

Kepada : **Yth. Bpk/Ibu Yosep Agus Pranoto ST., MT.**
Dosen Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

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

Nama : Abdul Wahid
Nim : 2118077
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

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

27 Februari 2025 s/d 27 Agustus 2025

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


Mengetahui
Program Studi Teknik Informatika S-1
Ketua,
Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432

Form S-4a

Lampiran 3 : Surat Keputusan Dosen Pembimbing 2



Malang, 18 Maret 2025

Nomor ITN-972/III INF/TA/2025

Lampiran ---

Perihal Pembimbing Pendamping Skripsi

Kepada Yth. Bpk/Ibu Karina Auliasari ST., M.Eng.
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

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

Nama : Abdul Wahid
Nim : 2118077
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

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

27 Februari 2025 s/d 27 Agustus 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,

Yosep Agus Pranoto, ST., MT.
NIP P. 1031000432

Form S-4a

Lampiran 4 :Formulir bimbingan skripsi dosen pembimbing 1

FORMULIR BIMBINGAN SKRIPSI

Nama : Abdul Wahid
 NIM : 2118077
 Masa Bimbingan : 27 Februari 2025 s.d 22 Agustus 2025
 Judul Skripsi : Perancangan Game 2D Berbasis "Pac-Man" dengan metode FSM & Fuzzy Sugeno

No.	Tanggal	Uraian	Paraf Pembimbing
1.	21 Maret 2025	Penjelasan flowchart, Revisi flowchart: Permisori "dibuatkan ini"	
2.	22 April 2025	Penjelasan fuzzy pada AI, kuis-kuis dengan Sany, perbaikan flowchart	
3.	23 April 2025	Parameter jarak AI, kuis, Level 3 - 3 batas waktu, kuis-kuis flowchart, fuzzy evil	
4.	23 Mei 2025	Pengujian FSM dan excel masuk ke bab 4	
5.	27 Mei 2025	FSM pada musuh di level 3 - pada karakter kuis	
6.	02 Juni 2025	Jurnal 5 tahun terakhir, Pengujian FSM, Pengujian Fuzzy & Fungsional Game	
7.	05 Juli 2025	Pembahasan pengujian pada bab 4 & pada hasil jurnal	
8.	02 Juli 2025	Pembahasan Revisi: Samudra, karakter masuk ke bab 3 & dilanjutkan revisi	
9.	09 Juli 2025	Menampilkan Hasil Revisi Sebelumnya & perbaikan pada jurnal	
10.	10 Juli 2025	Perbaikan game pada level 1	

Malang, 10/7/ 2025

Dosen Pembimbing 1

Yosep A. P.

Lampiran 5 : Formulir bimbingan skripsi dosen pembimbing 2

FORMULIR BIMBINGAN SKRIPSI

Nama : Abdul Wahid
 Nim : 2102017
 Masa Bimbingan : 27 Februari 2025 s/d 27 Agustus 2025
 Judul Skripsi : Perancangan Game Horror 3D Threaded dengan metode FSO & Puser Sugeno

No.	Tanggal	Uraian	Paraf Pembimbing
1.	14 April 2025	Perancangan Game (Karakter, Asset, Storyboard, Environment)	Li
2.	16 April 2025	Perancangan Alttunten	Li
3.	23 April 2025	Demo Game Level 1	Li
4.	8 Mei 2025	Demo Game Level 2 (keypad)	Li
5.	15 Mei 2025	Demo Game Level 3 (kunti)	Li
6.	22 Mei 2025	Revisi Game Level 4 & 5	Li
7.	28 Mei 2025	Revisi paper bag. pembahasan	Li
8.	26 Juni 2025	Revisi Bab IV	Li
9.	9 Juli 2025	Revisi Game Level 2	Li
10.	9 Juli 2025	Acc Laporan Skripsi x Game	Li

Malang, 14 April - 2025

Dosen Pembimbing 2

Karina Auliasari

Lampiran 6 : Berita Acara



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Abdul Wahid
Nim : 2118077
Jurusan : Teknik Informatika
Judul : PERANCANGAN GAME HORROR 3D "THE UNREAD" DENGAN
METODE FINITE STATE MACHINE DAN FUZZY SUGENO

Dipertahankan Di hadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1)
Pada

Hari : Rabu
Tanggal : 16 Juli 2025
Nilai : 79

Panitia Ujian Skripsi
Penguji I


Sentot Achmadi, Ir., M.Si., Dr.
NIP .P. 1039500281

Panitia Ujian Skripsi
Penguji II


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

Panitia Ujian Skripsi
Ketua Majelis Penguji


Yosep Agus Pranoto, ST, MT.
NIP .P. 1031000432

Lampiran 7 : Uji Turnitin

TUGAS AKHIR PERANCANGAN GAME HORROR 3D "THE UNREAD" DENGAN METODE FINITE STATE MACHINE DAN FUZZY SUGENO

ORIGINALITY REPORT

4%	4%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	eprints.itn.ac.id Internet Source	2%	
2	repo.darmajaya.ac.id Internet Source	1%	
3	jurnal.untan.ac.id Internet Source	1%	



INSTITUT TEKNOLOGI NASIONAL MALANG
PERPUSTAKAAN PUSAT
 Jl. Pembangunan, Gedung 200, 2nd Floor, ITN
 Telp: (041) 444333 Fax: (041) 444333 Email: library@itn.ac.id

FORM UJIPLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang

Nama: Abdul Wahid
 NIM: 2116077
 Fakultas - Jurusan: Industri Teknik Informatika
 Email: 1990.wahid1660@gmail.com
 No Tlp: 0853350332075
 Judul / Judul artikel: Perancangan Game Horror 3D "The Unread" Dengan Metode Finite State Machine dan Fuzzy Sugeno

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiasi menggunakan aplikasi turnitin dengan hasil kemiripan (Similarity) sebesar: 9 %
 Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

(Abdul Wahid)

Malang 16-08-2025
 Pelaksana

(Ratna W. Sph)

Lampiran 8 : Source Code Finite state machine

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

public class AIHunterFSM : MonoBehaviour
{
    [Header("Wander Settings")]
    public List<Transform> wanderPoints;
    public float wanderSpeed = 3.5f;
    public float idleWanderDelay = 2f;
    public float idleDetectionAngle = 360f;
    public bool randomizeWanderPoint = false;

    [Header("Detection Settings")]
    public string targetTag = "Target";
    public float defaultDetectionRadius = 10f;
    public float chaseDetectionRadius = 15f;
    public float defaultDetectionAngle = 45f;
    public float chaseDetectionAngle = 90f;
    public float targetLostDistance = 15f;
    public float chaseSpeed = 4f;
    public int rayDetectCount = 5;
    public float rayHeight = 1.0f;

    [Header("Attack Settings")]
    public float attackDistance = 1.5f;
    public UnityEvent onAttackJumpscareEvent;

    [Header("Distract Settings")]
    public float distractIdleDelay = 3f;

    [Header("Animation Settings")]
    public string moveParameter = "Move";
    public string wanderBlendAnim = "Wander";
    public string chaseBlendAnim = "Chase";
```

```

[Header("References")]
public Sanity sanityScript;
public FuzzySpeedController fuzzyController; // Referensi ke
script fuzzy

private NavMeshAgent agent;
private Animator animator;
private Transform currentTarget;
private bool isDistracted = false;
private Vector3 distractLocation;
private int currentWanderIndex = 0;
private bool isIdle = false;
private float currentDetectionRadius;
private float currentDetectionAngle;
private bool isChasing = false;
private float chaseStartTime;
private bool isAttack = false;
private bool wasChasing = false;

private void Start()
{
    agent = GetComponent<NavMeshAgent>();
    animator = GetComponent<Animator>();

    WanderToNextPoint();
    currentDetectionRadius = defaultDetectionRadius;
    currentDetectionAngle = defaultDetectionAngle;
}

private void Update()
{
    if (isDistracted) return;

    if (currentTarget)
    {
        if (isChasing)
        {
            float distance =
Vector3.Distance(transform.position, currentTarget.position);

```

```

        if (distance > currentDetectionRadius)
            LoseTarget();
        else
        {
            ChaseTarget();
            UpdateSpeedByFuzzy(distance);
            UpdateDetectionRadiusDuringChase();
        }
    }
}
else
{
    DetectTarget();

    if (!isIdle && !agent.pathPending &&
agent.remainingDistance < 0.5f)
        StartCoroutine(HandleIdle());
}

UpdateAnimator();
}

private void UpdateAnimator()
{
    if (isChasing && !wasChasing)
        animator.Play(chaseBlendAnim);
    else if (!isChasing && wasChasing)
        animator.Play(wanderBlendAnim);

    float moveSpeed = Mathf.Clamp01(agent.velocity.magnitude
/ agent.speed);
    animator.SetFloat(moveParameter, moveSpeed);

    wasChasing = isChasing;
}

private void DetectTarget()
{

```

```

        if (!isChasing) RaycastDetection();
    }

    private void RaycastDetection()
    {
        float angleStep = currentDetectionAngle / rayDetectCount;

        for (int i = 0; i < rayDetectCount; i++)
        {
            float angle = -currentDetectionAngle / 2 + (angleStep
* i);

            Vector3 rayDirection = Quaternion.Euler(0, angle, 0)
* transform.forward;

            Ray ray = new Ray(transform.position + Vector3.up *
rayHeight, rayDirection);
            if (Physics.Raycast(ray, out RaycastHit hit,
currentDetectionRadius))
            {
                if (hit.collider.CompareTag(targetTag))
                {
                    currentTarget = hit.transform;
                    isChasing = true;
                    chaseStartTime = Time.time;
                    return;
                }
            }
        }

    }

    private void ChaseTarget()
    {
        if (currentTarget == null) return;

        agent.SetDestination(currentTarget.position);

        float distance = Vector3.Distance(transform.position,
currentTarget.position);

```

```

        if (distance <= attackDistance && !isAttack)
        {
            onAttackJumpscareEvent.Invoke();
            isAttack = true;
        }
    }

    private void LoseTarget()
    {
        currentTarget = null;
        agent.speed = wanderSpeed;
        currentDetectionRadius = defaultDetectionRadius;
        currentDetectionAngle = defaultDetectionAngle;
        isChasing = false;
        isAttack = false;
        WanderToNextPoint();
    }

    private void WanderToNextPoint()
    {
        if (wanderPoints.Count == 0) return;

        int nextIndex = randomizeWanderPoint ? Random.Range(0,
wanderPoints.Count) : currentWanderIndex;
        agent.speed = wanderSpeed;
        agent.SetDestination(wanderPoints[nextIndex].position);

        if (!randomizeWanderPoint)
            currentWanderIndex = (currentWanderIndex + 1) %
wanderPoints.Count;
    }

    private IEnumerator HandleIdle()
    {
        isIdle = true;
        currentDetectionAngle = idleDetectionAngle;
        yield return new WaitForSeconds(idleWanderDelay);
        isIdle = false;
        currentDetectionAngle = defaultDetectionAngle;
    }

```

```

        WanderToNextPoint();
    }

    public void Distract(Vector3 location)
    {
        if (currentTarget != null) return;
        isDistracted = true;
        distractLocation = location;
        agent.SetDestination(distractLocation);
        StartCoroutine(HandleDistract());
    }

    private IEnumerator HandleDistract()
    {
        while (agent.pathPending || agent.remainingDistance >
0.5f)
            yield return null;

        yield return new WaitForSeconds(distractIdleDelay);
        isDistracted = false;
        WanderToNextPoint();
    }

    private void UpdateDetectionRadiusDuringChase()
    {
        if (isChasing && Time.time >= chaseStartTime + 3f &&
currentDetectionRadius < chaseDetectionRadius)
        {
            currentDetectionRadius = chaseDetectionRadius;
        }
    }

    private void UpdateSpeedByFuzzy(float distance)
    {
        if (currentTarget == null || sanityScript == null ||
fuzzyController == null) return;

        float sanity = sanityScript.CurrentSanity;

```

```
float speed =
fuzzyController.CalculateSugenoSpeed(distance, sanity);

    if (isChasing)
    {
        if (distance > 3f)
            speed = Mathf.Max(speed, 4f);
        else
            speed = 0f;
    }

    agent.speed = speed;
    chaseSpeed = speed;
}
}
```

Lampiran 9 : Source Code Fuzzy Sugeno

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

public class FuzzySpeedController : MonoBehaviour
{
    float TrapezoidMF(float x, float a, float b, float c, float
d)
    {
        if (x <= a || x >= d) return 0f;
        else if (x >= b && x <= c) return 1f;
        else if (x > a && x < b) return (x - a) / (b - a);
        else return (d - x) / (d - c);
    }

    float TriangleMF(float x, float a, float b, float c)
    {
        if (x <= a || x >= c) return 0f;
        else if (x == b) return 1f;
        else if (x > a && x < b) return (x - a) / (b - a);
        else return (c - x) / (c - b);
    }

    float DistanceClose(float x) => TrapezoidMF(x, 0, 0, 3, 5);
    float DistanceMedium(float x) => TriangleMF(x, 3, 9, 15);
    float DistanceFar(float x) => TrapezoidMF(x, 13, 15, 30, 30);

    float SanityLow(float y) => TrapezoidMF(y, 0, 0, 250, 300);
    float SanityMedium(float y) => TriangleMF(y, 250, 375.5f,
500);
    float SanityHigh(float y) => TrapezoidMF(y, 450, 500, 1000,
1000);

    public float CalculateSugenoSpeed(float distance, float
sanity)
    {
        float dClose = DistanceClose(distance);
        float dMedium = DistanceMedium(distance);
        float dFar = DistanceFar(distance);
```

```

        float sLow = SanityLow(sanity);
        float sMed = SanityMedium(sanity);
        float sHigh = SanityHigh(sanity);

        List<(float weight, float speed)> rules = new
List<(float, float)>
        {
            (Mathf.Min(dClose, sLow), 0f),
            (Mathf.Min(dClose, sMed), 0f),
            (Mathf.Min(dClose, sHigh), 0f),
            (Mathf.Min(dMedium, sLow), 6f),
            (Mathf.Min(dMedium, sMed), 5f),
            (Mathf.Min(dMedium, sHigh), 4f),
            (Mathf.Min(dFar, sLow), 2f),
            (Mathf.Min(dFar, sMed), 2f),
            (Mathf.Min(dFar, sHigh), 2f)
        };

        float numerator = 0f;
        float denominator = 0f;

        foreach (var rule in rules)
        {
            numerator += rule.weight * rule.speed;
            denominator += rule.weight;
        }

        return denominator == 0 ? 2f : numerator / denominator;
    }
}

```