

Adaptive AI Pada Survival Horror Game Menggunakan Fuzzy

Leonard Evan Widodo¹, Rolly Intan², Hans Juwiantho³
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121-131. Surabaya 60236
Telp (031) – 2983455, Fax. (031) 8417658

E-mail: leonardevan07@gmail.com¹, rintan@petra.ac.id², hans.juwiantho@petra.ac.id³

ABSTRAK

Ada banyak sekali metode yang digunakan oleh *developer* untuk membuat *artificial intelligence* di dalam *game* mereka. Metode yang paling sering dipakai oleh *developer* karena oleh fleksibilitasnya adalah metode *scripting*. *Scripting* dilakukan secara menulis beberapa *rule* yang ditulis dalam *source code* dalam bentuk *if-then*. Kelemahan dari metode ini adalah NPC *artificial intelligence* gampang untuk dieksploitasi oleh *gamer* yang telah memahami pola dari *behaviour* NPC tersebut.

Konsep *fuzzy logic* digunakan untuk mengatasi masalah tersebut. Konsep *fuzzy* digunakan untuk lebih memberikan AI memiliki *decision making* yang lebih adaptif, dimana kelakuan AI berubah tergantung kondisi yang ada. Konsep *fuzzy* yang digunakan dalam hubungan dengan *Fuzzy Inference* adalah metode *Fuzzy Tsukamoto* dimana semua *rule* memiliki nilai masing - masing yang nantinya akan di cari rata - rata dari semua *rule* untuk menentukan *decision* yang tepat untuk dilakukan. Dalam pembuatan *fuzzy rules*, penentuan desain dari *game* yang dibuat dan juga desain dari *map* harus dibuat terlebih dahulu. Setelah semua jadi, maka pembuatan *fuzzy rules* yang sesuai dapat dibuat.

Hasil penelitian melalui *survey* yang di adakan menunjukkan bahwa *game* yang menggunakan AI metode *fuzzy Tsukamoto* memberikan *decision* yang adaptif terhadap kondisi yang ada pada saat sesi permainan tersebut. Dari 9 orang yang di-survey mengatakan bahwa *Game A* dan *Game B* memiliki *decision* yang berbeda meskipun menggunakan AI yang sama sebanyak 77.8%, sedangkan yang mengatakan bahwa perilaku AI sama sebanyak 22.2%. Dari hasil *survey* ini dapat dikatakan bahwa metode *fuzzy Tsukamoto* dapat menghasilkan *decision* yang adaptif berdasarkan kondisi yang ada.

Kata Kunci: *Fuzzy Logic*, Metode *Tsukamoto*

ABSTRACT

There are many methods used by developers to create artificial intelligence in their games. The method most often used by developers because of it's flexibility is the scripting method. Scripting method is done by writing several rules in the source code in the form of if - then. The problem with this method is that the NPC's artificial intelligence is easily exploited by gamer who have understood the pattern of the NPC's behavior.

Therefore the fuzzy logic is used to overcome these problems. The use of the fuzzy inference method is to make the AI to be adaptive to certain situation where depending on the situation the AI will give different decision making. In this case, Tsukamoto Fuzzy inference method is used in this design. In the method, all rules have their respective values which then will be calculated to look for the average score which then was used to determine which decision to make. Before constructing the fuzzy rules, determining

the design of the game and the design of the map must be made first. After all are done, the appropriate fuzzy rules can be constructed.

The results of the survey showed that the game that uses the Tsukamoto fuzzy inference method provides an adaptive decision according to the conditions at a moment in a game. Ninth of the players who played the game agreed that game A and game B had different decisions despite using the same AI as much as 77.2%, while those who said that AI was the same as much as 22.8%. From the results of this survey it can be said that the Tsukamoto fuzzy inference method can produce adaptive decisions based on the condition of the game at a certain moments.

Keywords: *Fuzzy Logic*, *Tsukamoto method*

1. PENDAHULUAN

Di dalam *video game*, biasanya akan terdapat sebuah *artificial intelligence* yang berupa *bot*, *computer* atau yang lebih dikenal dengan sebutan NPC (*Non-playable character*) yang bertujuan untuk menjadi lawan *player* dan memberikan tantangan pada *player* untuk mencapai *goal* atau *objective* yang diberikan oleh *game* untuk mereka. Ada banyak sekali metode yang digunakan oleh *developer* untuk membuat *artificial intelligence* di dalam *game* mereka. Metode yang paling sering dipakai oleh *developer* karena oleh fleksibilitasnya adalah metode *scripting*. *Scripting* dilakukan secara menulis beberapa *rule* yang ditulis dalam *source code* dalam bentuk *if-then*. Kelemahan dari metode ini adalah NPC *artificial intelligence* gampang untuk dieksploitasi oleh *gamer* yang telah memahami pola dari *behaviour* NPC tersebut.

Beberapa *game* memang ada yang cocok untuk menggunakan AI yang menggunakan metode *scripting* seperti contohnya adalah *game* ber-genre *platformer* seperti contohnya adalah *Mario*. *Game* yang ber-genre yang seperti *survival horror game* tidak bisa menggunakan AI yang menggunakan metode *scripting*. AI yang ada di dalam *game* ber-genre ini harus memiliki *decision making* yang terlihat seperti sebuah makhluk yang memiliki akal sendiri. Untuk mengatasi permasalahan ini, maka diperlukan sebuah metode yang digunakan untuk membuat NPC semakin menjadi lebih *adaptive* dan tidak tertebak oleh para *player*.

Banyak *game AI* telah menggunakan banyak konsep, salah satu dari konsep yang digunakan adalah *fuzzy*. *Fuzzy logic* merupakan generalisasi dari *binary logic*. Hal ini dikarenakan *binary* merupakan bagian dari *fuzzy logic* karena *fuzzy logic* memiliki *logic value* antara 0 sampai 1. Dengan kata lain, *binary logic* yang hanya memiliki hanya memiliki dua values, 0 dan 1 merupakan bagian *fuzzy logic*. Di dalam konsep *Fuzzy*, terdapat banyak metoda mengenai *fuzzy inference*, salah satunya adalah metode *Fuzzy Inference Tsukamoto*. Metode *Fuzzy Inference Tsukamoto* pernah diimplementasikan oleh *Ratnajaya* [1] untuk menentukan

pergerakan karakter pada game yang memiliki *genre role playing game*, sehingga menghasilkan sebuah sistem cerdas yang dapat menghasilkan *behaviour* sesuai dengan peran masing-masing.

Pada Game yang memiliki Genre *Survival Horror*, Wonady [3] telah berhasil membuat *AI Companion* menggunakan logika *fuzzy* untuk menentukan *decision making* dengan baik. Menggabungkan apa yang dibuat oleh Ratnajaya [1] dan Wonady [3], dalam penelitian ini saya akan mencoba untuk mengimplementasikan metode *fuzzy inference Tsukamoto* tersebut pada *game* yang memiliki *ber-genre survival horror game* untuk menentukan aksi sang *monster/ hantu* pada saat mereka mengejar *player* yang berusaha untuk bertahan hidup. Selain itu, akan diimplementasikan *neural network* pada *AI* agar dapat lebih adaptif dari pada perilaku *player*. Pada *game survival horror game*, akan lebih baik jika menggunakan metode *fuzzy* dari pada metode *scripting*. Hal ini dikarenakan penggunaan konsep *fuzzy* akan membuat *NPC* akan lebih susah ditebak oleh *player* dan membuat *game* lebih menantang untuk *player* dan juga lebih menarik untuk dimainkan, dimana jika menggunakan *scripting* akan membuat musuh lebih mudah untuk ditebak dan membuat *game* terasa monoton dan membosankan. Ada beberapa *game* yang sudah terkenal memiliki *AI* yang menggunakan metode *fuzzy*, sebagai contohnya adalah *game left 4 dead* [2]. Logika *fuzzy* digunakan karena mudah untuk didesain dan diimplementasikan. *Output* tidak berbeda dengan sistem pengambilan keputusan berbasis data *crisp*. Perbedaannya hanyalah pada parameter atau *input* yang diterima dan bagaimana cara mengolah *input-input* tersebut. Selain itu logika *fuzzy* ringan untuk digunakan secara komputasi.

2. TINJAUAN STUDI

Ada banyak metode yang dapat digunakan untuk membuat *AI* yang dapat digunakan di dalam *game*. Salah satu dari metode yang dapat digunakan adalah metode *fuzzy logic*. Untuk penelitian ini, metode *fuzzy logic* yang digunakan adalah metode *Tsukamoto*. Metode ini dihitung dengan cara mencari minimal dari setiap *rule* dan kemudian di masukan rumus untuk mencari rata-rata dari semua *rule* yang ada untuk mencari *decision* apa yang harus dilakukan oleh *AI*. Metode *Tsukamoto* ini merupakan metode yang pernah digunakan oleh Ratnajaya [1] untuk membuat *AI* pada *game* yang memiliki *genre role playing game*. Dalam penelitian yang dilakukan oleh Ratnajaya [1] dapat menghasilkan *behaviour AI* yang sesuai dengan peran yang diberikan, sebagai contoh karakter penyerang dapat melakukan tugasnya untuk menyerang dan karakter *medic* juga lebih cenderung melakukan aksi *guard*. Perbedaan dari penelitian yang dilakukan oleh Ratnajaya [1] dan penelitian ini adalah *genre game* yang ingin dibuat, dimana penelitian yang dilakukan adalah *survival horror*. *Survival Horror* merupakan sebuah *genre* dimana pemain di tugaskan untuk bertahan hidup dalam kondisi yang sesuai dengan *film horror*. *Game* yang memiliki *genre survival horror* sudah berevolusi banyak sekali dan sudah merupakan *game genre* yang sudah banyak sukses[4].

Penelitian implementasi *fuzzy* di *game* yang memiliki *genre survival horror* pernah dilakukan oleh Wonady [3], dimana *fuzzy* digunakan untuk menentukan *decision* yang dibuat oleh *AI* yang merupakan teman dari pemain. Wonady [3] telah berhasil membuat *npc companion* bekerja dengan baik. Wonady [3] dapat menggabungkan *fuzzy logic* dan *Ellie: buddy system* dalam pembentukan *npc companion*. Meskipun metode yang digunakan dan juga *genre game* dari penelitian yang dilakukan oleh Wonady [3] dan penelitian yang dilakukan sama, cara kedua *game* yang dibuat berbeda. *Game* yang dibuat oleh Wonady [3] memiliki tujuan dimana pemain ditugaskan untuk membunuh musuh

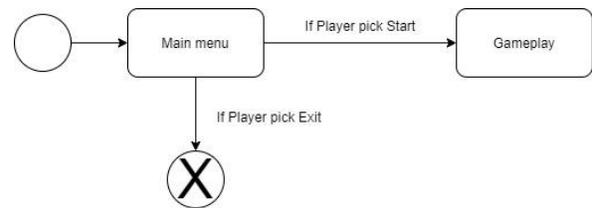
dengan senjata berbeda yang disertai dengan *companion AI* yang menggunakan *Ellie: buddy system*. Dimana *game* yang dibuat dalam penelitian ini menugaskan pemain untuk menghindari konflik dari musuh dan pemain juga tidak dapat membunuh musuh, hanya bisa menghentikan musuh sementara saja. Selain dari itu, penelitian yang dilakukan oleh Wonady [3] menggunakan *fuzzy* untuk menentukan *decision* yang dibuat oleh *companion* untuk membantu pemain sedangkan dalam penelitian ini *fuzzy* digunakan untuk menentukan *decision* yang dibuat oleh musuh untuk mencari pemain dengan mempertimbangkan berapa banyak *item* tersisa, *health* musuh dan juga *ammo* yang dimiliki pemain.

3. DESAIN SISTEM

Game yang dibuat berjudul “*Cursed Temple*” dimana pemain diminta untuk menjelajah kuil kuno untuk mencari harta yang terpendam dan juga menghindari musuh - musuh yang menjaga kuburan kuno itu untuk mencari kekayaan. Tujuan dari *game* ini adalah untuk mengumpulkan beberapa “artefak” di dalam kuil dan juga untuk menghindari musuh yang ada di dalam kuil kuno tersebut. Permainan selesai apabila kedua kondisi ini tercapai, antara pemain mati ataupun pemain berhasil untuk menemukan semua artefak dan keluar dengan selamat. *Game* ini hanya dapat dimainkan secara *single player* ataupun dimainkan secara satu orang saja.

3.1 Main Menu

Pada halaman *main menu*, pemain diberikan 2 pilihan, yaitu *Play game* dan *exit*. Untuk memulai permainan *player* diminta untuk menekan tombol *play game* dan sebaliknya untuk keluar dari aplikasi, pemain diminta untuk menekan tombol *exit*. Diagram alur *main menu* dapat dilihat pada Gambar 1.



Gambar 1. Diagram alur main menu

3.2 Gameplay

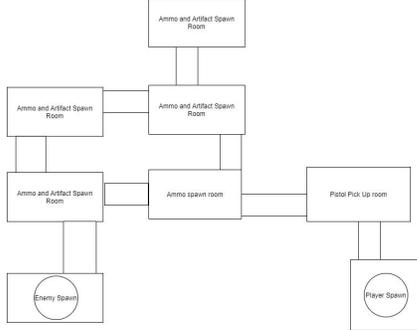
Pada awal *gameplay*, *player* akan di *spawn* di tempat yang sudah ditentukan. Setelah itu *player* harus menelusuri peta yang untuk mencari “artefak” yang sudah di *generate* secara *random* di dalam map. Pada awal mulai *gameplay*, karakter dari pemain akan memberikan monolog dimana *pemain* akan ditunjukkan untuk mengambil senjata untuk menjadi perlindungan dari musuh dan juga *ammo* sebanyak 7 biji. Kemudian pemain akan ditunjukkan untuk melakukan eksplorasi dari map untuk mencari “artefak” yang ada di dalam map tersebut. Apabila pemain bertemu dengan musuh maka pemain disarankan untuk menembak musuh untuk mengusir atau bahkan mengalahkan musuh. Ketika pemain sudah menemukan semua “artefak” maka pemain akan ditugaskan untuk kembali ke pintu awal untuk menyelesaikan *level* tersebut. Berikut merupakan tombol-tombol yang dapat digunakan oleh *player* untuk memainkan *game* ini:

Table 1. Kontrol dari Game

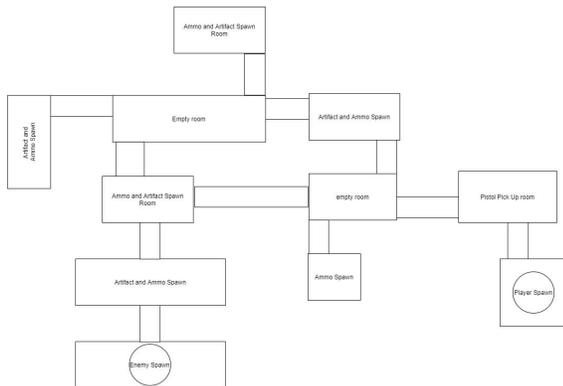
Movements	Controls
Move forward	W
Move backward	S
Move left	A
Move right	D
Fire Gun	Left Mouse Button
Trigger Flash Light	F
Trigger Item / Door	E

3.3 Map , Enemy , Artefak dan Fuzzy Design

Pada subbab berikut dibahas desain dari map yang dibuat yang digunakan di dalam game. *Player* pertama kali akan di *spawn* di dalam ruangan yang memiliki tanda *player spawn*, untuk *enemy* akan di *spawn* pertama kali di ruangan yang memiliki tanda *enemy spawn*. *Artefact* akan di letakan secara *random* di *area* yang tidak memiliki tanda. Pengaturan gerakan dari *enemy* akan dibuat menggunakan *fuzzy logic*. Untuk lebih jelasnya desain dapat dilihat pada Gambar 2 dan Gambar 3.



Gambar 2. Denah level 1



Gambar 3. Denah level 2

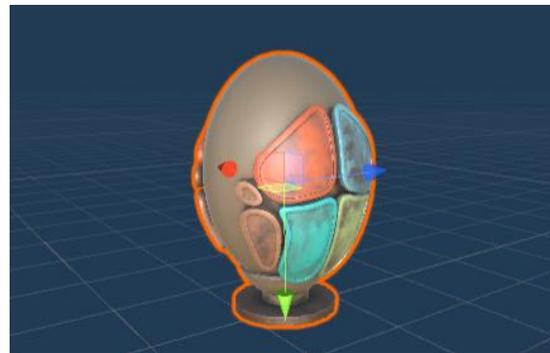
Seperti Gambar 2 dan Gambar 3 di atas, pemain akan memulai pada petak *player spawn* sedangkan musuh akan mulai dari *enemy spawn*. Tujuan dari pemain adalah menemukan artefak yang akan di letakan secara *random* pada petak lainnya yang bukan merupakan *player spawn* ataupun *enemy spawn*. Selain dari artefak pemain juga memiliki kemungkinan untuk menemukan

amunisi yang akan dapat digunakan untuk melawan musuh dengan terbatas. Untuk memenangkan permainan ini , pemain harus menemukan artefak (minimal 1) dan kembali ke tempat *player spawn* untuk memenangkan permainan. *Score* yang akan di dapatkan berbeda tergantung banyak artefak yang dikumpulkan.



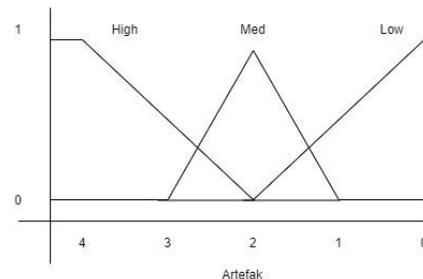
Gambar 4. model musuh

Model dari Gambar 4 merupakan model yang digunakan di dalam game. *Asset* tersebut dapat ditemukan di *Unity Asset Store*. Model tersebut digunakan karena sesuai dengan tema yang diinginkan di dalam game. Tema dari game adalah menjelajahi kuil yang sudah kuno yang memiliki banyak kutukan yang tersembunyi di dalamnya. Model pada Gambar 3.4 bisa digambarkan sebagai orang yang terkena kutukan untuk menjaga kuil tersebut dari pengacau kuil, yaitu dari pemain.



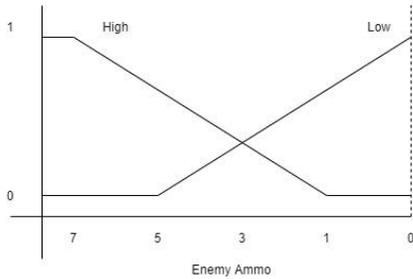
Gambar 5. model artefak

Gambar 5 merupakan model artefak yang akan digunakan dalam Game. Model tersebut digunakan karena sesuai dengan tema yang digunakan di dalam Game, sama halnya dengan *Enemy Model*. Model diastis digunakan karena memiliki bentuk yang mistis dan misterius sehingga cocok digunakan di tempat yang bertemakan kuil kuno. Barang tersebut bisa di anggap sebagai harta dari kuil tersebut yang harus di peroleh oleh *player* untuk memenangkan game.



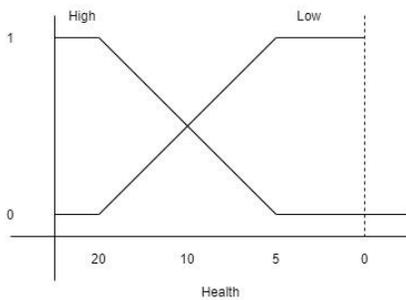
Gambar 6. model fuzzy set jumlah artefak

Model *fuzzy set* pada Gambar 6 merupakan total dari jumlah artefak yang dapat diperoleh dari satu sesi permainan. Setiap kali *game* dimulai akan terdapat 3 artefak yang diletakkan secara random di peta. Ketika masih terdapat 3 artefak yang belum didapatkan oleh pemain maka artefak yang masih ada dikategorikan sebagai *High*, sebaliknya apabila artefak yang tersedia sudah tinggal 1 maka dikategorikan sebagai *Low*.



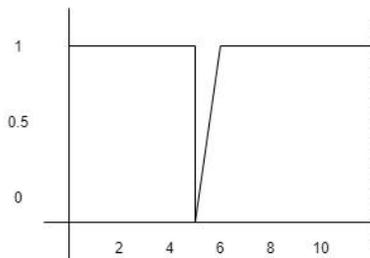
Gambar 7. model fuzzy set enemy ammo

Fuzzy set enemy Ammo seperti Gambar 7 adalah fungsi keanggotaan yang digunakan *AI* untuk memprediksi berapa banyak jumlah amunisi yang dimiliki oleh pemain pada *frame* tertentu. Pemain pertama kali akan memiliki *ammo* sebanyak 7 ketika pertama kali menemukan senjata. *Ammo* berkurang setiap kali saat pemain menembakan peluru dengan menekan tombol *fire1* (left mouse button).



Gambar 8. fuzzy set health dari AI

Fuzzy set health seperti Gambar 8 merupakan health yang dimiliki oleh *AI*. Tiap *AI* memiliki hp sebanyak 20 yang nantinya terkena damage sebanyak 5 apabila musuh terkena serangan oleh pemain.



Gambar 9. Fuzzy set action

Action fuzzy set akan digunakan untuk mengatur dari gerakan yang dilakukan oleh *AI*. *Fuzzy set* pada Gambar 9 digunakan setelah melakukan perhitungan fuzzy Tsukamoto. Nilai 5 kebawah merupakan *Retreat* sedangkan di atas 5 merupakan *wander*.

Dalam fuzzy ini awalnya akan diatur oleh sebuah rule yang akan menentukan pola pengambilan keputusan dari *NPC*. *Rule*-nya adalah sebagai berikut.

1. If (Artefak is high) and (EnemyAmmo is high) and (Health is high) then (output is Action Wander).
2. If (Artefak is high) and (EnemyAmmo is high) and (Health is low) then (output is Action Retreat).
3. If (Artefak is high) and (EnemyAmmo is low) and (Health is high) then (output is Action Wander).
4. If (Artefak is high) and (EnemyAmmo is low) and (Health is low) then (output is Action Retreat).
5. If (Artefak is med) and (EnemyAmmo is high) and (Health is high) then (output is Action Wander).
6. If (Artefak is med) and (EnemyAmmo is high) and (Health is low) then (output is Action Retreat).
7. If (Artefak is med) and (EnemyAmmo is low) and (Health is high) then (output is Action Wander).
8. If (Artefak is med) and (EnemyAmmo is low) and (Health is low) then (output is Action Retreat).
9. If (Artefak is low) and (EnemyAmmo is high) and (Health is high) then (output is Action Wander).
10. If (Artefak is low) and (EnemyAmmo is high) and (Health is low) then (output is Action Wander).
11. If (Artefak is low) and (EnemyAmmo is low) and (Health is high) then (output is Action Wander).
12. If (Artefak is low) and (EnemyAmmo is low) and (Health is low) then (output is Action Wander).

3.4 Desain dari Game

Game yang dibuat memiliki tema kastil lama yang terinspirasi dari *game* yang berjudul *Silent Hill*, dimana tema yang digunakan di *game* adalah sebuah kota tua yang tertinggal dan berkabut. *Lightning* yang digunakan memiliki tema yang gelap dan hanya bisa terlihat jelas apabila terkena pemain menggunakan *flashlight* yang sudah merupakan salah satu aspek dari *game* yang ber-genre *survival horror game*. Dalam beberapa tempat di map juga terdapat banyak kabut yang membuat suasana dari *game* semakin misterius dan menyeramkan. Selain itu, desain dari *gameplay* yang digunakan pada *game* ini juga mirip dengan *game* yang bernama *slenderman* dimana satu bertujuan mencari "note" dan *game* yang dibuat ini bertujuan untuk mencari artefak yang tersebar di dalam map untuk menyelesaikan permainan.

4. PENGUJIAN SISTEM

Pada bab ini akan dijelaskan tentang pengujian terhadap aplikasi dan sistem *AI* yang telah dibuat. Pengujian sistem dilakukan agar dapat memastikan apakah *AI* yang menggunakan metode fuzzy Tsukamoto yang dibantu dengan *neural network* itu lebih baik dari *AI scripting*. Pengujian dilakukan terhadap tampilan serta beberapa fitur dari aplikasi *game*. Selain itu, pengujian juga dilakukan untuk mengetahui apakah metode fuzzy Tsukamoto yang digunakan lebih baik atau tidak.

4.1 Pengujian Aplikasi game

Game didesain untuk memiliki tema kastil tua yang sudah tertinggal. Tema ini diambil mirip dengan *game* yang berjudul *resident evil* dimana temanya adalah gedung yang sudah tertinggal dan memiliki atmosfer yang gelap.

4.1.1 Pengujian Main Menu



Gambar 10. Main Menu

Pada awal mula permainan berjalan pemain akan diarahkan ke *main menu* seperti pada Gambar 10, di sini pemain di berikan 2 pilihan yaitu *Start* atau *Exit*. Pemain dapat memulai permainan dengan memilih pilihan *Start*. Pilihan *Exit* digunakan oleh pemain untuk keluar dari permainan. Jika pemain telah menyelesaikan permainan ini, maka pemain akan dikembalikan ke halaman ini.

4.1.2 Pengujian Level 1



Gambar 11. Level 1 Basement

Setelah pemain memilih *New Game* atau melalui *Load Game*, pemain akan dipindahkan pada level 1 yaitu *Basement*. Pada map ini pemain ditugaskan untuk mencari artefak yang tersebar di dalam map juga menghindari dari musuh *AI* yang ditugaskan untuk menangkap pemain. Apabila pemain telah mengumpulkan semua artefak, maka pemain akan ditugaskan untuk kembali ruang depan. Artefak dan *Ammo* akan di-generate secara *random* di ruangan-ruangan tertentu. *Level 1* dapat dilihat seperti pada Gambar 11.

4.1.3 Pengujian level 2



Gambar 12. Level 2 Rooftop

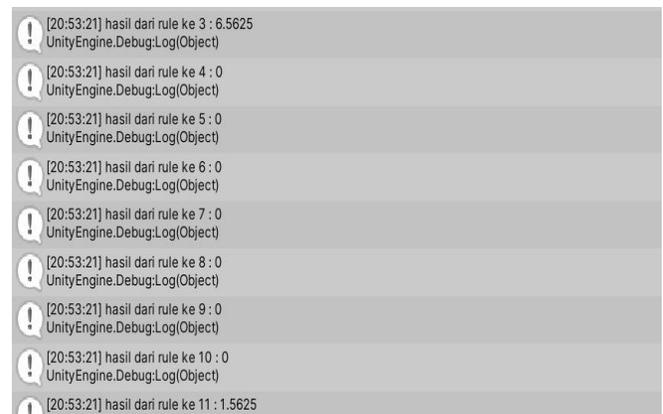
Setelah pemain berhasil menyelesaikan *level 1* maka pemain akan dipindahkan ke *map* lain yang berbeda dan lebih luas. Disini pemain memiliki tugas yang sama seperti *map* yang sebelumnya. Apabila pemain telah menyelesaikan *map* yang ini maka pemain akan berhasil menyelesaikan *game*. *Level 2* dapat dilihat seperti pada Gambar 12.

4.2 Pengujian Terhadap AI

Pengujian dilakukan untuk mengetahui apakah *AI* telah berjalan dengan seharusnya. Pengujian ini dibagi menjadi 2. Pertama adalah pengujian *decision making* yang dilakukan oleh *AI* berdasarkan hasil dari *fuzzy* yang didapat. Yang Kedua adalah Pengujian apakah *AI* akan menyerang *player* apabila *player* sudah dekat dengan *AI*.

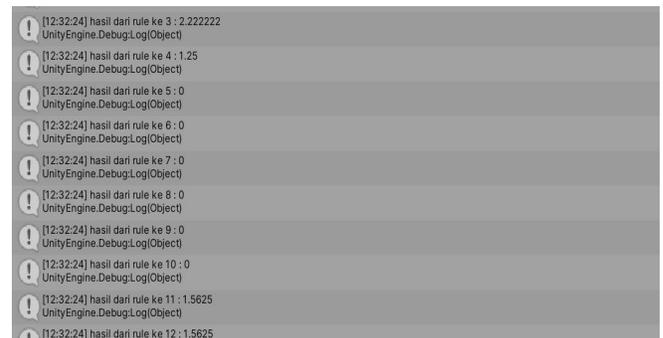
4.2.1 Pengujian Decision making

Ketika *Game* pertama dimulai maka *Ammo* yang dimiliki berupa 0, Artefak akan di-generate secara *random* dan health musuh akan berupa 20. Berikut merupakan hasil dari perhitungan *fuzzy* pada awal *game* dimulai:



Gambar 13. Hasil perhitungan fuzzy rule

Dari perhitungan pada Gambar 13 dapat dilihat bahwa *AI* telah dapat melakukan sebuah *decision* untuk melakukan *wander*. Pengujian kedua dilakukan apabila *AI* memutuskan untuk melakukan aksi *retreat*. Di dalam pengujian ini dilakukan apabila *health* yang dimiliki oleh musuh merupakan 10, artefak di-generate sebanyak 3 kali, dan *ammo* yang dimiliki oleh pemain merupakan 0. Adapun perhitungan sebagai berikut:



Gambar 14. Hasil perhitungan fuzzy retreat

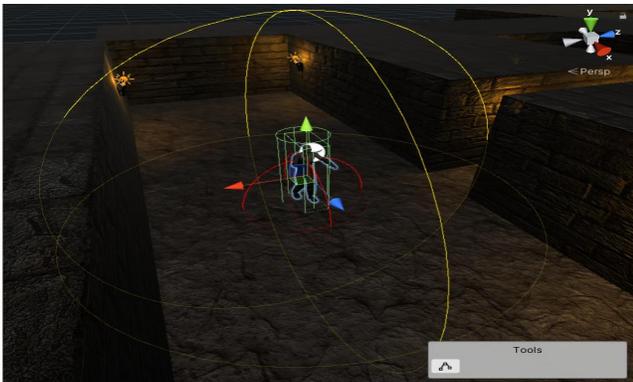
Dari contoh yang tersedia pada Gambar 14 dapat dilihat bahwa *AI* sudah bisa melakukan *decision* untuk melakukan *retreat* pada kondisi demikian. Berdasarkan perhitungan pada Gambar 14 dapat kita lihat bahwa *AI* sudah bisa memberikan pilihan yang tepat berdasarkan kondisi yang ada.

4.2.2 Pengujian Attack dan Chase



Gambar 15. Ai memberikan damage ke pemain

Aksi *attack* yang dapat dilihat pada Gambar 15 akan terjadi apabila pemain telah memasuki pada jarak *chase* dan jarak *attack* dari *AI*. *AI* juga hanya dapat melihat sebesar 180 derajat dari depan *AI*. Jarak *Chase* range dan *Attack* range dapat dilihat pada Gambar 16.



Gambar 16. Jarak Chase dan Attack range AI

4.2.3 Pengujian Room Priority

Pada pengujian ini *AI* melakukan pemilihan ruangan yang dituju dengan bantuan *neural network*. *AI* sudah dapat memilih ruangan yang dituju seperti yang ditunjukkan pada Gambar 17. Penggunaan *neural network* berbeda dengan yang sudah di tulis di ruang lingkup karena *fuzzy logic* yang dibuat sudah berjalan dengan baik dan sudah memberikan *decision* yang sesuai dan membuat *game* lebih susah daripada metode *scripting*, selain itu *fuzzy set* yang digunakan di dalam *game* berbentuk statis jadi tidak bisa untuk di update dengan menggunakan *neural network*. *Neural network* akan di anggap lebih baik apabila *AI* dapat mengalahkan pemain. *Neural network* yang digunakan pada penelitian ini dibuat oleh [5].



Gambar 17. Pemilihan ruangan berdasarkan neural network

4.3 Pengujian terhadap Player

4.3.1 Moving

Pada *game* ini pemain dapat bergerak dengan menggunakan *keypad* WASD. Pemain juga dapat berlari dengan menggunakan tombol *left shift*.

4.3.2 Shooting

Pada *game* ini pemain juga dapat melakukan *shooting* yaitu menembak enemy. *Shooting* dapat dilakukan dengan menekan tombol *left mouse button*. *Pistol* hanya dapat berisi 7 peluru dan *pistol* hanya dapat ditembakkan ketika pemain memiliki peluru.

4.3.3 Taking damage



Gambar 18. pemain terkena damage

Pada *game* ini, apabila musuh *AI* dapat menyerang pemain apabila musuh terlalu dekat dengan pemain. Apabila pemain menerima *damage* dari musuh maka layar akan menjadi lebih merah yang menandakan pemain telah menerima *damage* dari musuh. Pemain hanya memiliki 4 health di *game* ini dan *game* akan selesai apabila pemain telah kehabisan *health* seperti yang terlihat pada Gambar 18.

4.3.4 Ammo pickup



Gambar 19. Player mengambil ammo

Pada *game* ini, *ammo* merupakan bahan yang penting, karena memiliki *ammo* merupakan satu-satunya cara untuk mengusir musuh dari pemain. *Ammo* dapat ditemukan di ruangan - ruangan dan berbentuk seperti kotak kecil seperti pada Gambar 19. Jumlah *ammo* pada satu sesi *gameplay* di *random* jadi ada suatu saat dimana *ammo* banyak sekali di dapatkan dan juga ada saat dimana *ammo* jarang sekali didapatkan. Ketika mengambil *ammo*, *ammo* yang dimiliki pemain akan selalu menjadi 7 saat mengambil *ammo box* meskipun *ammo* yang dimiliki sekarang adalah 6 atau berapapun jumlah dari *ammo* yang dimiliki oleh pemain dalam suatu saat di dalam *game*. Apabila pemain menginjak *ammo* maka jumlah *ammo* akan bertambah seperti Gambar 20. Proses pengambilan *ammo* dan artefak sama, yaitu dengan cara menginjak *ammo* dan artefak yang di temukan di dalam *game*.

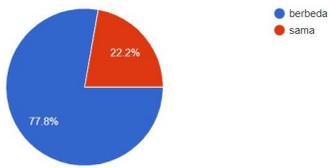


Gambar 20. player telah mengambil ammo

4.4 Pengujian Aplikasi game dengan audience

Setelah memainkan Game A dan Game B apakah ada perbedaan dari perilaku AI yang ada di dalam game?

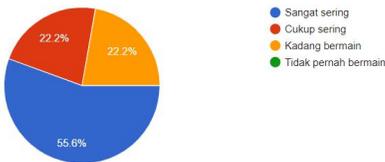
9 responses



Gambar 21. Hasil survey game 1

Apakah Anda sering bermain game?

9 responses

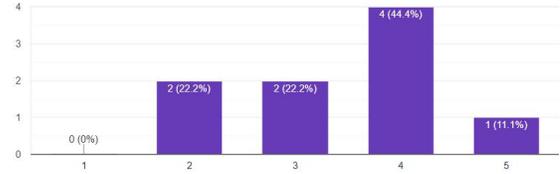


Gambar 22. Hasil survey game 2

Untuk melihat apakah AI yang menggunakan metode *Fuzzy Tsukamoto* dapat memberikan response yang adaptif maka penulis mengadakan suatu *survey* kepada orang - orang untuk mencoba *game*. *Game* di-build 2 kali dengan menggunakan kondisi yang berbeda dimana artifak di *game* A akan di generate sebanyak 4 atau 3 dan *game* B akan digenerate sebanyak 2 kali saja. Dari Hasil *survey* seperti Gambar 21 dapat yang mengatakan bahwa perilaku AI pada *game* A dan B memiliki perilaku yang berbeda meskipun menggunakan AI yang sama adalah sebanyak 77,8% dari 9 orang yang di *survey*. Dari 9 orang yang di *survey* , diketahui bahwa 55.6% dari orang yang di *survey* merupakan orang yang sering bermain *game* jika dilihat pada Gambar 22. Berdasarkan hasil dari *survey*, Dari skala 1 sampai 5 yang mengatakan bahwa AI berperilaku berbeda banyak yang mengatakan 4 dapat dilihat dari Gambar 23.

Dari Skala 1 sampai 5 seberapa berbeda AI yang ada di game A dan B?

9 responses



Gambar 23. Hasil survey game 3

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil perancangan model dan penelitian ini, dapat diambil beberapa kesimpulan antara lain:

- Desain dari *game* sudah memiliki tema *survival horror*.
- AI sudah memiliki *decision making* yang sudah *adaptif*. Berdasarkan *survey*, sebanyak 77.8% dari semua orang yang di *survey* mengatakan bahwa perilaku AI berbeda pada kondisi *game* yang berbeda meskipun menggunakan AI yang sama.
- Berdasarkan hasil pengujian dapat disimpulkan bahwa 44.4% dari orang yang di *survey* mengatakan bahwa *game* sudah dapat memberikan suasana *horror* yang cukup menyeramkan.

5.2 Saran

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut untuk penelitian ini adalah sebagai berikut:

- Fuzzy rules bisa dibuat lebih banyak lagi untuk dapat menghasilkan *decision making* yang lebih adaptif lagi.

6. DAFTAR PUSTAKA

- [1] Ratnajaya, D. and Wibawa H.A.2018.Implementasi Kecerdasan Buatan dalam Menentukan Aksi Karakter pada Game RPG dengan Logika Fuzzy Tsukamoto. Jurnal Ilmu Komputer dan Informatika 2018 82-86.
- [2] The AI Systems of Left 4 Dead. Dari https://steamcdn-a.akamaihd.net/apps/valve/2009/ai_systems_of_l4d_mike_borth.pdf.
- [3] Wonady, W. et al.2017.Implementasi fuzzy logic dan ellie: buddy ai pada companion dalam game survival horror
- [4] Burke, A.2018.*The evolution of survival horror games*. Retrieved from <https://motley.ie/the-evolution-of-the-survival-horror-genre/>.
- [5] Tutorial On Programming An Evolving Neural Network in C# w/ Unity3D.2017. Retrieved from <https://www.youtube.com/watch?v=Yq0SfuiOVYE>