

PENERAPAN METODE FINITE STATE MACHINE DAN FUZZY PADA GAME “BLACK WARRIOR”

Daniel Ari Setiawan, Suryo Adi Wibowo, Yosep Agus Pranoto
Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
captainzanfer@gmail.com

ABSTRAK

Game adalah salah satu bentuk hiburan yang digemari oleh semua kalangan dari semua rentang usia. Semakin banyak muncul game yang berkualitas, baik dari segi visualisasi maupun dari segi cerita. Salah satu tema game yang digemari adalah tentang petualangan. Berdasarkan kegemaran tersebut, penulis tertarik untuk membuat game “Black Warrior”. Metode FSM yang diterapkan pada level 1 dan level 2 mampu memodelkan respon musuh dengan variasi respon patroli, mengejar, dan menyerang sesuai dengan jarak antara player dan musuh/monster. Pada level 3 penerapan metode *fuzzy* mampu memodelkan respon bos musuh dengan variasi respon patroli, standby, dan menyerang dimana respon tersebut dihasilkan dari kombinasi jarak (antara player dan musuh) dan kecepatan player. Berdasarkan tahap pengujian yang telah dilaksanakan, dapat diketahui bahwa hasil pengujian fungsional dan hasil pengujian kecerdasan buatan yang ada pada game sudah berfungsi dengan baik seluruhnya, sedangkan hasil pengujian user menunjukkan bahwa mayoritas user menilai game yang dikembangkan sudah cukup baik.

Kata Kunci: Game, Black Warrior, Finite State Machine, Fuzzy

1. PENDAHULUAN

Saat ini telah banyak game yang disajikan dengan kualitas visualisasi yang cukup canggih karena didukung oleh teknologi yang semakin maju sehingga permainan menjadi lebih interaktif, kondisi terasa nyata dan pemain merasa hidup di dalam game tersebut. Salah satu tema game yang digemari adalah game petualangan.

Berdasarkan kegemaran tersebut, penulis tertarik untuk membuat game “Black Warrior”. Game “Black Warrior” menyajikan latar belakang tempat yang menarik, terutama bagi anak-anak yang menyukai game bertema petualangan.

Game “Black Warrior” akan menggunakan metode *Finite State Machine* dan metode *Fuzzy* (Tsukamoto). Aplikasi pemrograman yang digunakan untuk mengembangkan game adalah *Unity 3D*.

Game ini bercerita tentang karakter utama “Black” yang terjebak dalam wilayah penuh musuh/monster. Misi yang harus diselesaikan Black adalah mengalahkan musuh/monster untuk keluar dari wilayah tersebut.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Semakin pesatnya perkembangan *game* maka dibutuhkan pengetahuan tentang penggunaan teknologi pada *game* yaitu AI. AI secara garis besar yaitu suatu kreasi manusia dalam sebuah sistem dalam sistem komputer yang meniru kecerdasan pemikiran dan logika manusia (Kristanto, 2004). Perkembangan AI yang telah ada menghasilkan banyak macam AI yang bisa digunakan. Salah satu

AI yang banyak digunakan adalah *Finite State Machine* (FSM) (Wicaksono, Hariadi, dan Mardi, 2013).

FSM merupakan sebuah metode yang umum pada sebuah *Game* untuk menentukan suatu kejadian, aksi, dan kondisi (Herlambang, 2019). FSM diterapkan pada musuh, state (keadaan) awal musuh adalah diam, kemudian ketika terjadi event (peristiwa) yaitu karakter mendekati musuh, maka musuh akan melakukan salah satu dari action (aksi) sesuai dengan yang telah ditentukan, yaitu patrol, mengejar dan menyerang (Kartika, 2017). Sebagai sebuah metodologi perancangan sistem kontrol, penerapan FSM telah banyak diterapkan pada perangkat lunak, khususnya pada *game*. Metode FSM dapat diterapkan untuk nilai-nilai batas perpindahan yang pasti dan sangat mudah diterapkan pada *game* “*The Relationship*” (Rahadian, Suyatno, dan Maharani, 2016). Pengembangan *game* ini menggunakan tools *Unity 3D* dan bahasa pemrograman *C#*. Hasil dari penelitian ini adalah terealisasinya suatu *game* dengan menerapkan metode *Finite State Machine* untuk menentukan respon karakter *non-player character* yang berbeda tergantung dari interaksi yang dilakukan oleh pemain.

Penulis akan menerapkan metode FSM dan *Fuzzy* sehingga bisa memodelkan perilaku musuh/monster. Diterapkannya metode *fuzzy* pada bos enemy maka logika *fuzzy* mampu membuat boss enemy sedikit lebih sulit untuk dikalahkan (Hakiki, 2018). Penulis juga akan menerapkan lingkungan dan tantangan yang bervariasi agar pemain semakin termotivasi untuk menyelesaikan *game* “*Black*

Warrior". Penulis menggunakan software Unity 3D untuk membangun *game* 3D "Black Warrior".

2.2. Pengertian Game

Game merupakan aktifitas terstruktur atau semi terstruktur yang biasanya bertujuan untuk hiburan dan kadang dapat digunakan sebagai sarana pendidikan. Karakteristik *game* yang menyenangkan, memotivasi, membuat kecanduan dan kolaboratif membuat aktifitas ini digemari oleh banyak orang (Wahono, R.S. 2009).

Berdasarkan jenisnya, *game* dapat dibagi menjadi beberapa jenis diantaranya:

1. Fighting (Pertarungan)

Dalam jenis *game* ini, *player* dapat memilih karakter dengan kemampuan berbeda-beda. *Player* juga bisa mengeluarkan jurus dan kemampuan berbeda dari tiap karakter untuk mengalahkan musuh. Biasanya pertarungan hadir secara *one on one* (satu lawan satu) dalam sebuah arena yang terbatas. Contoh : Mortal Kombat, Street fighter dan Tekken.

2. Shooting (Tembak Menembak)

Dalam jenis *game* ini *player* berisi berbagai misi yang harus diselesaikan. Intinya dalam *game* ini, *player* membutuhkan reflex, koordinasi mata dan tangan hingga waktu yang tepat. Contoh : Resident Evil dan Counter Strike.

3. Adventure (Petualangan)

Dalam jenis *game* ini *Player* bisa memainkan karakter untuk menjelajah hutan belantara atau menjelajahi sebuah bangunan. Intinya, *player* akan melalui sebuah misi. Jenis *game* ini akan menekankan jalur cerita dan kemampuan berpikir *player* untuk menemukan petunjuk berikutnya. Contoh : Tomb Raider, Grand Theft Auto dan Dark Souls.

4. Simulasi

Dalam jenis *game* ini menghadirkan konsep permainan yang meniru keadaan di dunia nyata. Semua faktor dalam *game* akan dibuat semirip dunia nyata, ada yang menceritakan kehidupan manusia atau kehidupan menjadi seorang walikota membangun sebuah kota. Dalam *game* ini semua keputusan *player* berpengaruh pada karakter yang dimainkan. Inti dari *game* ini membuat *player* harus berpikir untuk mendirikan, membangun hingga mengatasi masalah dengan dana terbatas. Contoh : The Sims, dan Sim City.

5. Strategi

Dalam jenis *game* ini memerlukan kemampuan *player* untuk memimpin sebuah pasukan, kemudian mengelola sumber daya hingga membangun peradaban. Setelah pembangunan selesai, *player* bisa berperang dengan pasukan lain untuk merebut kekuasaan. Maka dibutuhkan strategi yang tepat agar pasukan bisa menang dan wilayah *player* tidak diserang. Contoh :

Stronghold Crusader, Age of Empires dan Clash of Clans.

6. Sport (Olahraga)

Dalam *game* jenis ini memiliki *gameplay* berbagai jenis olahraga di dunia. Intinya, *player* akan melakukan pertandingan olahraga secara virtual. Contoh : Pro Evolution Soccer dan NBA.

7. Racing (Balapan)

Dalam jenis *game* ini memerlukan kemampuan mengemudi motor, mobil, maupun kendaraan lainnya. *Game* ini akan memacu adrenalin *player* untuk menjadi yang tercepat di sirkuit. *Game* jenis ini contohnya, Forza dan Need for Speed.

8. Role Playing Game

Dalam jenis *game* ini mengandung unsur leveling dan *player* akan bebas menjelajah dunia *game*. Setiap karakter memiliki kemampuan dan kekuatan yang berbeda dan dapat berkembang sesuai yang diinginkan *player*. Beberapa *Game* RPG bahkan membuat *player* bisa menentukan ending dari jalan cerita *game* tersebut. *Game* RPG terbagi menjadi dua jenis yakni *action* RPG dan *turn based* RPG. Contoh : Dragon Quest dan Final Fantasy.

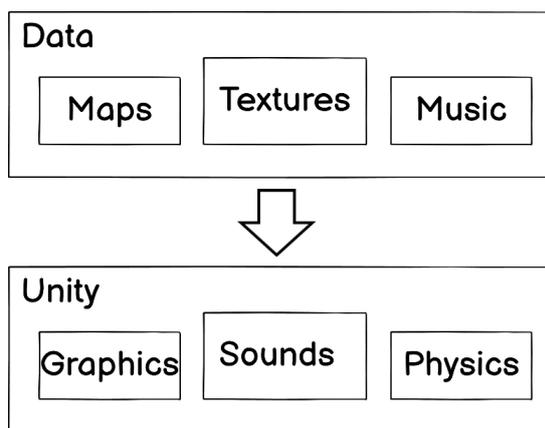
2.3. Game "Black Warrior"

Genre Adventure memiliki *gameplay* dan cerita yang panjang namun masih banyak memiliki celah kekurangan. Jenis musuh/monster dan environment yang hampir sama dalam *gameplay* menjadi kekurangan *game adventure* pada umumnya (Marzuki, 2014). Kekurangan lain yang juga penting adalah jenis interaksi karakter utama dan musuh/monster dengan model yang tidak jauh berbeda antar *game adventure* lainnya membuat perkembangan *game adventure* tidak mengalami kemajuan yang signifikan. Pemain merasa tertantang bila ada inovasi *game adventure* dengan interaksi antar objek dalam *game* yang berbeda dari *game adventure* pada umumnya. Pemain mulai beralih pada *game* pertarungan *player vs player* karena lebih menantang dan tidak monoton.

Game "Black Warrior" adalah *game* dengan *genre adventure* yang berpusat pada karakter yang menghancurkan musuh/monster dan akhirnya menghancurkan *Boss* pada level 3. Penulis akan menerapkan metode FSM dan *Fuzzy* sehingga bisa memodelkan perilaku musuh/monster. Pada level 1 dan level 2 diterapkan metode FSM yang mampu memodelkan respon musuh dengan variasi respon patroli, mengejar, dan memukul sesuai dengan jarak antara *player* dan musuh/monster. Pada level 3 penerapan metode *fuzzy* mampu memodelkan respon bos musuh dengan variasi respon patroli, standby, dan menyerang. Diterapkannya metode *fuzzy* pada bos enemy maka logika *fuzzy* mampu membuat boss enemy sedikit lebih sulit untuk dikalahkan (Hakiki, 2018).

2.4. Unity 3D

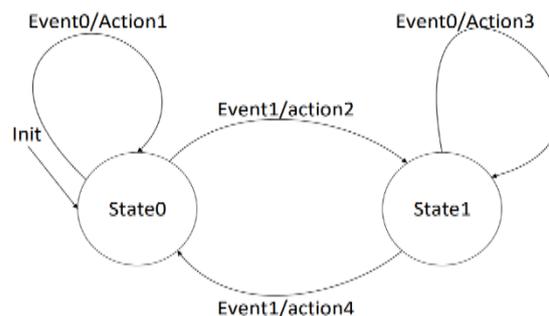
Unity 3D merupakan salah satu teknologi yang digunakan untuk membangun *game* pada berbagai *platform* yang dikembangkan oleh *Unity Technologies*. Unity 3D umumnya digunakan untuk mengembangkan *game* pada *platform* PC. Unity 3D diprogram dalam Bahasa C/C++ dan mendukung script dalam bahasa C#, JavaScript atau Boo. Unity 3D berkembang dari alat pengembangan *game* khusus untuk *game* Mac OS X pada tahun 2005 menjadi alat pengembangan *game* pada berbagai *platform* saat ini (Blackman, 2013). Unity 3D menggunakan konsep aliran data seperti gambar 1.



Gambar 1. Konsep aliran data dalam Unity 3D (Oktora dan Samopa, 2013)

2.5. Finite State Machine (FSM)

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) (Millington dan Funge, 2009). Dalam *state machine* sistem menempati satu *state*(keadaan). Sistem akan beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Sistem akan tetap melakukan aksi yang sama pada suatu *state* sampai sistem menerima *event* tertentu baik yang berasal dari perangkat luar atau komponen dalam sistem itu sendiri. Setiap *state* terhubung oleh transisi dan setiap transisinya mengarah ke satu *state* lainnya. 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 rumit (Yulsilviana dan Ekawati, 2019).



Gambar 2. Contoh Diagram State Sederhana (Yulsilviana dan Ekawati, 2019).

Diagram tersebut memperlihatkan FSM dengan dua buah *state* dan dua buah *input* serta empat buah aksi 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 (Yulsilviana dan Ekawati, 2019).

2.6. Fuzzy

Logika *fuzzy* yang pertama kali diperkenalkan oleh Lotfi A. Zadeh pada tahun 1962. Logika *fuzzy* adalah metodologi system control pemecahan masalah, yang cocok untuk diimplementasikan pada system, mulai dari system yang sederhana, system kecil, embedded system, jaringan PC, multi-channel atau workstation berbasis akuisisi data, dan system control. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”, dan lain-lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada diantara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. (Sutojo, Mulyanto, Suhartono, dan Vincent, 2011).

Logika *fuzzy* yang pertama kali diperkenalkan oleh Lotfi A. Zadeh, memiliki derajat keanggotaan dalam rentang 0 (nol) hingga 1 (satu), berbeda dengan logika digital yang hanya memiliki dua nilai yaitu 1 (satu) dan 0 (nol). Proses *fuzzy inference* dapat dibagi dalam lima bagian (Naba, 2009; Kusumadewi dan Purnomo, 2010), yaitu:

1. Fuzzyfikasi Input : FIS mengambil masukan - masukan dan menentukan derajat keanggotaannya dalam semua *fuzzy set*.
2. Operasi Logika *Fuzzy* : Hasil akhir dari operasi ini adalah derajat kebenaran antecedent yang berupa bilangan tunggal.
3. Implikasi : Merupakan proses mendapatkan consequent atau keluaran sebuah IF-THEN rule berdasarkan derajat kebenaran antecedent. Proses ini menggunakan mengambil nilai MIN/terkecil dari dua bilangan : Hasil operasi *fuzzy logic OR* dan *fuzzy set* banyak.
4. Agregasi :Yaitu proses mengkombinasikan keluaran semua IF-THEN rule menjadi sebuah *fuzzy set* tunggal. Pada dasarnya agregasi adalah operasi *fuzzy logic OR* dengan masukannya adalah semua *fuzzy set*.
5. Defuzzyfikasi : Keluaran dari defuzzyfikasi adalah sebuah bilangan tunggal, cara mendapatkannya ada beberapa versi, yaitu centroid, bisector, middle of maximum, largest of maximum dan smallest of maximum

3. METODE PENELITIAN

3.1. Analisis Game

Game yang dibangun merupakan game pertualangan 3 Dimensi yang terdiri dari 3 level yang harus di selesaikan untuk memenangkan game ini. Berikut adalah analisa sistem yang ada pada *game* ini:

1. Sistem *single player*.
2. Grafik *game* 3 Dimensi.
3. Bercerita tentang “Black” yang tersesat di padang rumput dan diserang oleh musuh/monster.
4. Interaksi menggunakan *keyboard*.
5. AI pada level 1 dan 2 menggunakan metode *Finite State Machine* untuk aksinya.
6. Tindakan pada karakter musuh/monster di level 3 menggunakan *Fuzzy Logic* (Tsukamoto).
7. *Game* ini dirancang menggunakan *unity3D* dengan bahasa C#.
8. *Game* ini menggunakan satuan unit dimana 1 unit sama dengan 1 meter.
9. Pada akhir permainan akan ditunjukkan jumlah score level dan score total di setiap level.

3.2. Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan yang berisi proses-proses atau fitur-fitur yang harus disediakan oleh sistem. Berikut adalah analisis kebutuhan fungsional pada *game* ini :

1. Dalam *game Black Warrior*, terdapat *Opening* sebelum memasuki menu utama
2. Dalam *game Black Warrior*, menu utama terdiri dari menu *Start*, menu *Option*, menu *Credit*, dan menu *Exit*
3. Dalam *game Black Warrior*, menu *Start* untuk memulai permainan
4. Dalam *game Black Warrior*, menu *Option* berisi petunjuk *control player*

5. Dalam *game Black Warrior*, menu *Credit* berisi nama pembuat game dan dosen pembimbing
6. Dalam *game Black Warrior*, menu *Exit* untuk keluar dari permainan
7. Dalam *game Black Warrior*, *Player* mempunyai kendali berlari (run) dengan menekan tombol *shift*, menyerang (attack) dengan menekan tombol *left click* pada *mouse*, maju (up) dengan menekan tombol W, mundur (down) dengan menekan tombol S, kanan (right) dengan menekan tombol D, kiri (left) dengan menekan tombol A
8. Dalam *game Black Warrior*, terdapat 3 level
9. Dalam *game Black Warrior*, terdapat *Opening* sebelum memasuki setiap level
10. Dalam *game Black Warrior*, untuk menuju level selanjutnya akan terbuka ketika *Player* berhasil menyelesaikan misi pada setiap level.
11. Dalam *game Black Warrior*, tampil score setelah menyelesaikan level
12. Dalam *game Black Warrior*, terdapat *Ending* setelah menyelesaikan semua level
13. Dalam *game Black Warrior*, akan tampil *final score* setelah *Ending*

3.3. Kebutuhan Nonfungsional

Kebutuhan nonfungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Adapun kebutuhan non fungsional pada *game* meliputi:

1. Sistem pada *Game “Black Warrior”* adalah *single player*
2. *Game “Black Warrior”* mempunyai grafik *game* 3 dimensi
3. Interaksi pada *Game “Black Warrior”* menggunakan *keyboard*
4. Tindakan pada karakter musuh menggunakan metode *finite state machine* dan *fuzzy*
5. *Finite state machine* diimplementasikan pada katakter musuh/monster pada level 1 dan level 2
6. *Fuzzy* diimplementasikan pada karakter Bos musuh/monster pada level 3
7. *Game* ini dibuat dengan menggunakan *software Unity 3D* dan bahasa pemrograman C#

3.4. Analisis Target User

Game “Black Warrior” memiliki target *user* untuk anak sampai remaja dengan usia 15-24 tahun yang dapat mengoperasikan computer. Jika *user* dengan usia dibawah 15 tahun harus dengan bimbingan orang tua dikarenakan didalam *game* terdapat beberapa efek visual yang tidak sesuai bagi remaja dengan usia dibawah 15 tahun. Dikhawatirkan jika tanpa bimbingan orang tua, *user* dibawah usia 15 tahun akan menirukan adegan kekerasan yang ada pada *game “Black Warrior”*.

3.5. Analisis Konsep Game

Game “Black Warrior” ini berjenis *game* petualangan, yang disajikan dalam *view* 3D yang berarti *game* dapat dilihat dari kamera samping dengan grafik dan desain 3D. Game “Black Warrior” ini bercerita tentang “Black” yang tersesat di padang rumput dan diserang oleh para musuh/monster. Game “Black Warrior” memiliki tujuan maupun target yang harus dicapai untuk diselesaikan.

Tujuan dari *game* ini adalah lolos dari serangan monster-monster yang menyerang, musuh/monster yang menyerang dari setiap level berbeda-beda. Sementara harus mengalahkan para musuh/monster, “Black” akan diberikan kesempatan mengobati/menambah nyawa (*health*) yang sudah berkurang dengan objek item di level 3 di area padang rumput. Target dalam *game* ini adalah *player* harus mampu membunuh semua musuh/monster yang menghalangi pada setiap level dan tetap memiliki nyawa yang tersisa.

Setelah pemain berhasil bertahan sampai akhir *game*, maka *game* akan menampilkan “Black” keluar dari padang rumput dan berhasil bebas dari serangan monster-monster.

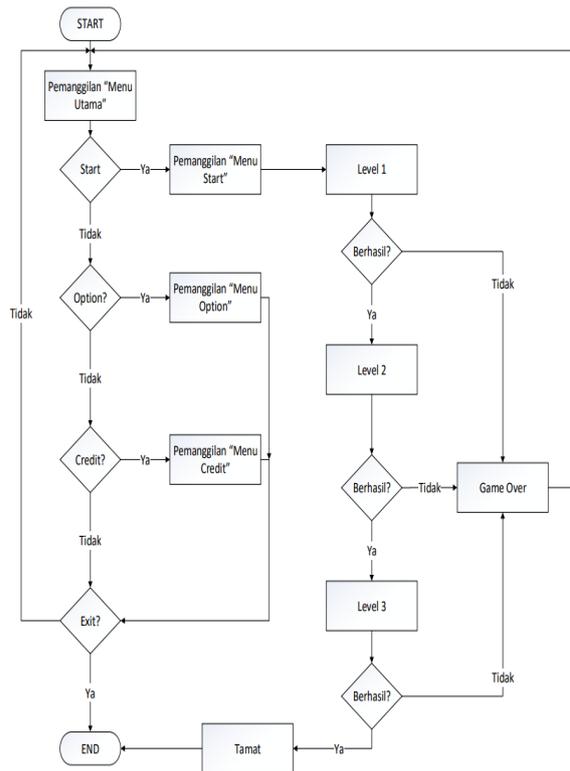
3.6. Perancangan Storyline

Jalan cerita dari *game* “Black Warrior” adalah sebagai berikut:

1. Pada suatu zaman hiduplah “Black” seorang prajurit yang pemberani.
2. Pada suatu ketika “Black” sedang menyusuri padang rumput.
3. Tiba-tiba “Black” menemukan dirinya tersesat di padang rumput.
4. “Black” menemukan beberapa musuh/monster di dekat padang rumput tersebut.
5. Monster-monster itu menyerang “Black”.
6. “Black” pun tidak mempunyai pilihan lain selain harus melawan monster-monster untuk keluar dari padang rumput tersebut.

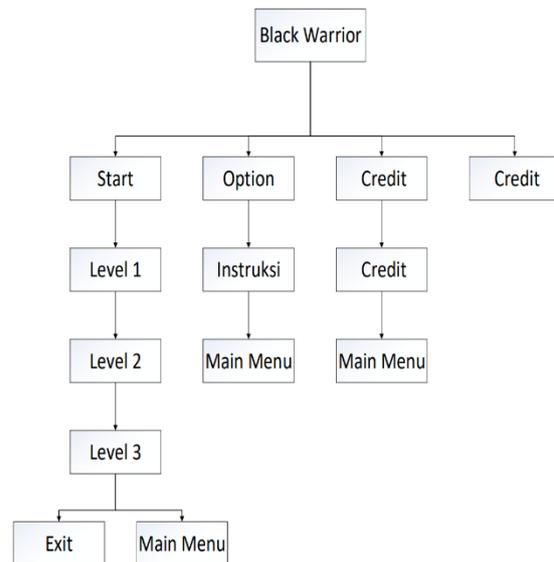
3.7. Flowchart Game

Berikut rancangan flowchart *game* yang ditunjukkan pada Gambar 3. Rancangan flowchart yang didesain adalah rancangan flowchart untuk menjelaskan apa saja yang bisa dilakukan oleh user yang dalam hal ini adalah gamer pada menu – menu yang tersedia.



Gambar 3. Flowchart Game

3.8. Struktur Menu Game



Gambar 4. Struktur Menu

Didalam *game* “Black Warrior”, terdapat beberapa pilihan menu yang dimuat dalam menu utama. Menu-menu tersebut terdiri dari menu *Single Player* untuk memulai *game*, menu *Option* untuk melihat informasi seputar *game*, *Credit* untuk melihat *credit* *game*, dan *Exit* untuk keluar dari *game*.

3.9. Karakter dan Item Game

Karakter pada game “Black Warrior” meliputi karakter utama dan musuh/monster pada level 1, 2 dan 3. Pembuatan karakter menggunakan *blender*. Animasi pada game dikerjakan menggunakan tool yang ada pada *Unity 3D*. Berikut gambar tabel desain karakter:

Tabel 1. Tabel Desain Karakter Utama

Nama Karakter	Karakter	Penjelasan
Black Warrior		Merupakan karakter utama yang ada pada game. Karakter ini menggunakan senjata pedang untuk menyerang dengan damage 10.

Tabel 2. Tabel Desain Karakter Musuh/Monster

Nama Karakter	Karakter	Penjelasan
Musuh Level 1		Merupakan karakter musuh/monster yang ada pada level 1. Karakter ini muncul di titik letak spawn yang telah ditentukan. Di level 1 mempunyai damage 10 dan health 100.
Musuh Level 2		Merupakan karakter musuh/monster yang ada pada level 2. Karakter ini muncul di titik letak spawn yang telah ditentukan. Di level 2 mempunyai damage 10 dan health 100.
Musuh Level 3		Merupakan karakter bos musuh/monster yang ada pada level 3. Karakter ini muncul di titik letak spawn yang telah ditentukan. Di level 3 mempunyai damage 10 dan health 100.

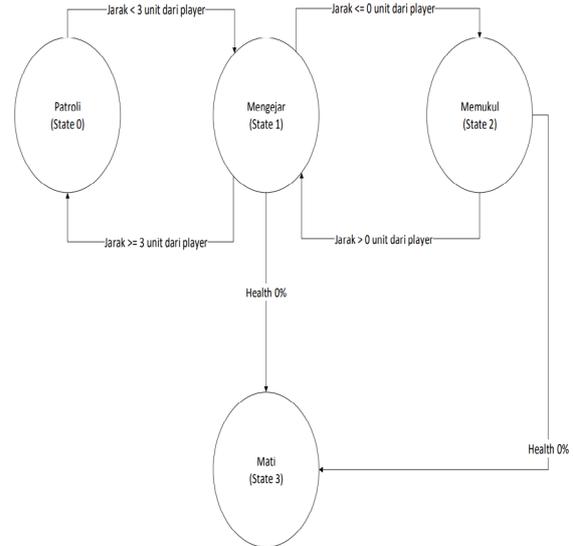
Tabel 3. Tabel Desain Environment dan Item Game

Nama Karakter	Objek	Penjelasan
Environment		Merupakan padang rumput untuk level 1 sampai dengan level 3

3.10. Penerapan Metode Finite State Machine (FSM)

Penerapan Metode FSM pada musuh/monster level 1

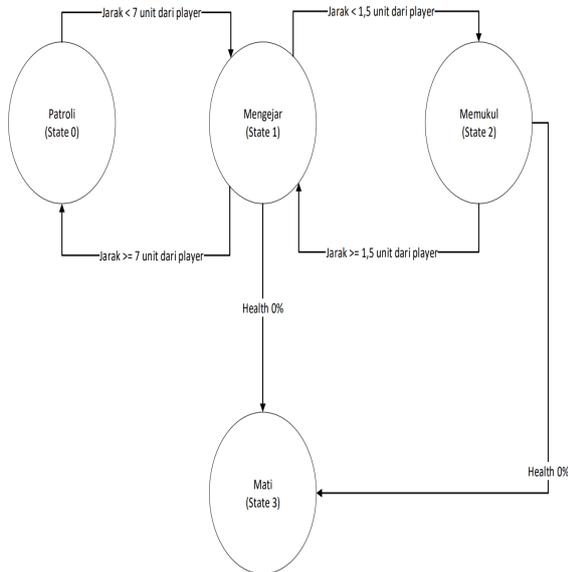
1



Gambar 5. Diagram FSM pada musuh/monster level 1

Gambar 5 menunjukkan ketika sistem dijalankan maka sistem mula-mula berada pada *state 0* (bergerak atau patroli), ketika player berada pada jarak jangkauan musuh/monster (jarak player < 3 *unit*) maka *state 1* (mengejar) akan dijalankan. Jika berada pada jangkauan serang monster (jarak player <= 0 *unit*) maka *state 2* (memukul) akan dijalankan dan ketika *health bar* dari musuh/monster mencapai 0% maka *state 3* (mati) akan dijalankan.

3.11. Penerapan Metode FSM pada musuh/monster level 2



Gambar 6. Diagram FSM pada musuh/monster level 2

Gambar 6 menunjukkan ketika sistem dijalankan maka sistem mula-mula berada pada state 0 (bergerak atau patroli) ketika player berada pada jarak jangkauan musuh/monster (jarak *player* < 7 unit) maka state 1 (mengejar) akan dijalankan. Jika berada pada jangkauan serang musuh/monster (jarak *player* < 1,5 unit) maka state 2 (memukul) akan dijalankan dan ketika *health bar* dari musuh/monster mencapai 0% maka state 3 (mati) akan dijalankan.

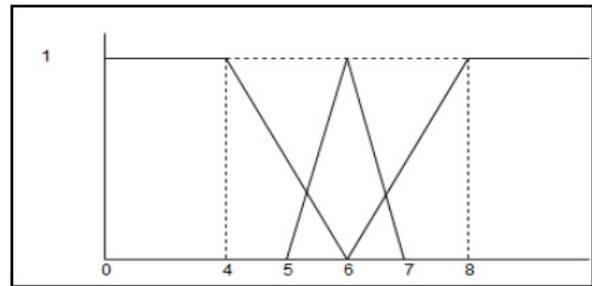
3.12. Penerapan Metode Fuzzy

Dalam game “Back Warrior”, *Fuzzy Logic* digunakan untuk menentukan tindakan yang dilakukan oleh karakter musuh/monster level 3 terhadap *player*. Terdapat 3 tipe tindakan pada musuh/monster level 3 yaitu menyerang, berjaga dan bertahan. Terdapat 2 masukan yaitu jarak dan kecepatan.

Masukan jarak dalam himpunan *fuzzy* dibedakan menjadi 3 jenis yaitu dekat, sedang dan jauh. Dan masukan kecepatan dalam himpunan *fuzzy* juga dibedakan menjadi 3 jenis yaitu pelan, sedang dan cepat. Jenis dari himpunan *fuzzy* lah yang akan menjadi putusan aksi yang akan dilakukan oleh musuh/monster level 3.

1. Fungsi jarak

Fungsi jarak adalah fungsi yang menentukan aksi yang dilakukan oleh Musuh/Monster ataupun *Player*, Gambar 7 adalah bentuk keputusan hasil perhitungan *Fuzzy Logic*.



Gambar 7. Parameter Jarak

Berikut adalah aksi yang dilakukan dari nilai – nilai yang sudah didapatkan dari perhitungan *fuzzy*

Tabel 4. Fungsi Keanggotaan Jarak

Fungsi	Domain	Parameter
Dekat	0 – 6	[0;4;5;6]
Sedang	5 – 7	[5;6;7]
Jauh	6 - 8	[6;7;8]

Berikut ini adalah rumus yang digunakan untuk menentukan aksi pada *fuzzy logic* (Tsukamoto) menentukan aksi sesuai dengan jarak Musuh/monster dan *Player*.

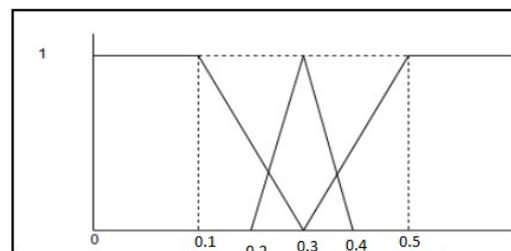
$$\mu \text{ Jarak Aksi Dekat } [x] = \begin{cases} (6-x)/(6-4), & 4 \leq x < 6 \\ 0, & x \geq 6 \end{cases}$$

$$\mu \text{ Jarak Aksi Sedang } [x] = \begin{cases} 0, & x \leq 5 \\ (x-5)/(6-5), & 5 \leq x < 6 \\ (7-x)/(7-6), & 6 \leq x < 7 \\ 0, & 7 \leq x \end{cases}$$

$$\mu \text{ Jarak Aksi Jauh } [x] = \begin{cases} 0, & x \geq 6 \text{ atau } 10 \leq x \\ (x-6)/(8-6), & 6 \leq x < 8 \\ 0, & x \geq 8 \end{cases}$$

2. Fungsi Kecepatan

Fungsi kecepatan menentukan kecepatan aksi yang dilakukan oleh Musuh/Monster dan *Player*, Gambar 8 adalah bentuk keputusan hasil perhitungan *Fuzzy Logic*.



Gambar 8. Parameter Kecepatan

Berikut adalah aksi yang dilakukan dari nilai – nilai yang sudah didapatkan dari perhitungan *fuzzy*.

Tabel 5. Fungsi Keanggotaan Kecepatan

Fungsi	Domain	Parameter
Pelan	0 – 0.3	[0;0.1;0.2;0.3]
Sedang	0.2 – 0.4	[0.2;0.3;0.4]
Cepat	0.3 - 0.5	[0.3;0.4;0.5]

Berikut ini adalah rumus yang digunakan untuk menentukan aksi pada *fuzzy logic* (Tsukamoto) menentukan aksi sesuai dengan kecepatan Musuh/monster dan Player.

$$\mu_{\text{Jarak Aksi Pelan}} [x] = \begin{cases} (0.3 - x)/(0.3 - 0.1), & 0.1 \leq x \leq 0.3 \\ 0, & x \geq 0.3 \end{cases}$$

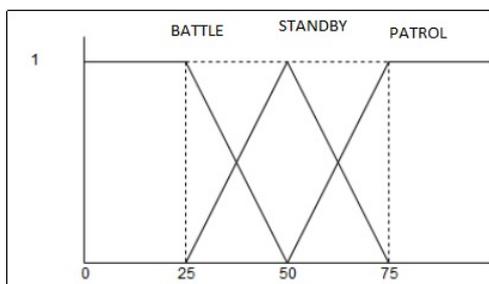
$$\mu_{\text{Jarak Aksi Sedang}} [x] = \begin{cases} 0, & x \leq 0.2 \\ (x - 0.2)/(0.3 - 0.2), & 0.2 \leq x \leq 0.3 \\ (0.4 - x)/(0.4 - 0.3), & 0.3 \leq x \leq 0.4 \\ 0, & 0.4 \leq x \end{cases}$$

$$\mu_{\text{Jarak Aksi Cepat}} [x] = \begin{cases} 0, & x \geq 0.3 \\ (x - 0.3)/(0.5 - 0.3), & 0.3 \leq x \leq 0.5 \\ 0, & x \geq 0.5 \end{cases}$$

3. Hasil Keputusan

Dengan menggunakan logika *fuzzy* maka didapatkan tingkat aturan *fuzzy* hasil keputusan dari 2 parameter yang masing-masingnya memiliki 3 variabel. Sedangkan untuk fungsi implikasi, fungsi yang digunakan adalah fungsi AND (fungsi MIN).

Setelah derajat keanggotaan masing-masing dihitung, proses selanjutnya adalah menghitung defuzzifikasi, dengan rumus sebagai berikut :



Gambar 9. Hasil Kesimpulan

Hasil defuzzifikasi berikut adalah aksi yang akan dilakukan oleh musuh/monster level 3, Tabel 6. menerangkan aksi yang akan dilakukan.

Tabel 6. Hasil Kesimpulan

Fungsi	Domain	Parameter
Battle	0 – 50	[0;25;50]
Standby	25 – 75	[25;50;75]
Patrol	50 - 100	[50;75;100]

Sehingga didapatkan *rule base system* kombinasi sebanyak 9 *rule* tersaji lengkap dalam Tabel 7

Tabel 7. Fuzzy Rule

ANTESEDEN				KONSEKUEN	
Operator	Jarak	Operator	Kecepatan	Operator	tindakan
IF	Dekat	AND	Pelan	THEN	Battle
			Sedang		Battle
			Cepat		Standby
	Sedang		Pelan		Patrol
			Sedang		Patrol
			Cepat		Standby
	Jauh		Pelan		Patrol
			Sedang		Patrol
			Cepat		Patrol

Berikut ini adalah rumus yang digunakan untuk menarik kesimpulan dari kedua nilai input jarak dan kecepatan.

$$\mu_{\text{Battle}} [z] = \begin{cases} 0, & x \geq 25 \\ (50 - z)/(50 - 25), & 25 \leq x \leq 50 \\ 0, & x \geq 50 \end{cases}$$

$$\mu_{\text{Patrol}} [z] = \begin{cases} 0, & x \geq 50 \\ (z - 50)/(75 - 50), & 50 \leq x \leq 75 \\ 0, & x \geq 75 \end{cases}$$

$$\mu_{\text{Stand By}} [z] = \begin{cases} 0, & x \geq 25 \\ (z - 25)/(50 - 25), & 25 \leq x \leq 50 \\ (75 - z)/(75 - 50), & 50 \leq x \leq 75 \\ 0, & x \geq 75 \end{cases}$$

Setelah didapatkan nilai hasil, maka akan didapatkan nilai akhir yang menentukan aksi yang dilakukan.

Tabel 8. Hasil Keputusan

Nilai z	Keputusan
0 ≤ z ≤ 25	Battle
26 ≤ z ≤ 50	Stand by
51 ≤ z ≤ 75	Patrol

3.13. Fungsi Implikasi

Berikut ini adalah pembentukan keputusan dari hasil 2 parameter tersebut:

- IF jarak dekat AND kecepatan sedang THEN *battle*
- IF jarak dekat AND kecepatan cepat THEN *stand by*
- IF jarak sedang AND kecepatan sedang THEN *patrol*
- IF jarak sedang AND kecepatan cepat THEN *stand by*

Setelah aturan dibentuk, maka lakukan aplikasi fungsi implikasi. Fungsi implikasi yang digunakan dalam proses ini adalah fungsi MIN, yaitu dengan mengambil tingkat keanggotaan yang minimum dari variable *input* sebagai *output*-nya. Berdasarkan aturan-aturan yang sesuai dengan kondisi tersebut.

- IF jarak dekat AND kecepatan sedang THEN *battle*

- $a - Predikat_1 = \mu \text{ Jarak Dekat}_{[5]} \cap \mu \text{ Kecepatan Sedang}_{[5]}$
 $= \text{MIN} (\mu \text{ Jarak Dekat}_{[5]}; \mu \text{ Kecepatan Sedang}_{[5]})$
 $= \text{MIN} (0.5; 0) = 0$
2. IF jarak dekat AND kecepatan cepat THEN stand by
 $a - Predikat_2 = \mu \text{ Jarak Dekat}_{[5]} \cap \mu \text{ Kecepatan Cepat}_{[5]}$
 $= \text{MIN} (\mu \text{ Jarak Dekat}_{[5]}; \mu \text{ Kecepatan Cepat}_{[5]})$
 $= \text{MIN} (0.5; 0.5) = 0.5$
3. IF jarak sedang AND kecepatan sedang THEN patrol
 $a - Predikat_3 = \mu \text{ Jarak Sedang}_{[5]} \cap \mu \text{ Kecepatan Sedang}_{[5]}$
 $= \text{MIN} (\mu \text{ Jarak Sedang}_{[5]}; \mu \text{ Kecepatan Sedang}_{[5]})$
 $= \text{MIN} (0; 0) = 0$
4. IF jarak sedang AND kecepatan cepat THEN stand by
 $a - Predikat_4 = \mu \text{ Jarak Sedang}_{[5]} \cap \mu \text{ Kecepatan Cepat}_{[5]}$
 $= \text{MIN} (\mu \text{ Jarak Sedang}_{[5]}; \mu \text{ Kecepatan Cepat}_{[5]})$
 $= \text{MIN} (0; 0.5) = 0$

3.14. Hasil Keputusan

Setelah mengetahui derajat keanggotaan masing-masing dihitung, proses selanjutnya adalah menghitung hasil keputusan dengan rumus untuk mengambil kesimpulan, berikut ini contoh hasil perhitungan:

1. IF jarak dekat AND kecepatan sedang THEN battle.
 $a - Predikat_1 = 0$
 Hasil keputusan battle.
 Sehingga ketika nilai himpunan Battle = 0 didapatkan fungsi keanggotaan berikut ini:
 $\mu \text{ Battle} [z] = \frac{50 - z}{50 - 25} = 0$
 $z = 50 - 0(25) = 50$
 maka didapatkan hasil z adalah 50.
2. IF jarak dekat AND kecepatan cepat THEN stand by
 $a - Predikat_2 = 0.5$
 Hasil Keputusan stand by
 untuk himpunan stand by = 0.5 didapatkan fungsi keanggotaan sebagai berikut:
 $\mu \text{ Stand By} [z] = \frac{75 - z}{75 - 50} = 0.5$
 $z = 75 - 0.5(25) = 62.5$
 dan

$$\mu \text{ Stand By} [z] = \frac{z - 25}{50 - 25} = 0.5$$

$$z = 0.5(25) + 25 = 37.5$$

maka didapatkan hasil z adalah 62.5 dan 37.5.

3. IF jarak sedang AND kecepatan sedang THEN patrol
 $a - Predikat_3 = 0$
 Hasil keputusan patrol
 untuk himpunan patrol = 0 didapatkan fungsi keanggotaan sebagai berikut ini:
 $\mu \text{ Patrol} [z] = \frac{z - 50}{75 - 50} = 0.5$
 $z = 0(25) + 50 = 50$
 maka didapatkan hasil z adalah 50.
4. IF jarak sedang AND kecepatan cepat THEN stand by
 $a - Predikat_4 = 0$
 Hasil keputusan stand by
 $\mu \text{ Stand By} [z] = \frac{75 - z}{75 - 50} = 0$
 $z = 75 - 0(25) = 75$
 dan
 $\mu \text{ Stand By} [z] = \frac{z - 25}{50 - 25} = 0$
 $z = 0(25) + 25 = 25$
 maka didapatkan hasil z adalah 75 dan 25.

3.15. Penegasan (Defuzzy)

Penegasan atau defuzzyfikasi dikerjakan menggunakan metode centroid average.

$$z^* = \frac{\sum(\mu(z) \times a - \text{predikat})}{\sum a - \text{predikat}}$$

$$= \frac{(0 \times 0.5) + (0.5 \times 62.5) + (0.5 \times 37.5) + (0 \times 50) + (0 \times 75) + (0 \times 25)}{0 + 0.5 + 0.5 + 0 + 0 + 0}$$

$$= \frac{50}{1} = 50$$

Hasil perhitungan dari defuzzyfikasi adalah 50, yang berarti sesuai dengan Tabel 3.11 hasil keputusan akhir yang dilakukan adalah stand by.

4. HASIL DAN PEMBAHASAN

4.1. Pengujian Gameplay

Pengujian gameplay merupakan pengujian bagaimana game berjalan sesuai dengan rancangan sistem yang telah dibuat. Tampilan pengujian game menggambarkan interaksi antara karakter dengan environment di dalam game.



Gambar 10. Pengujian Gameplay

4.2. Tampilan Level 1 Pada Game

Pada Gambar 11 merupakan perancangan level 1 pada game. Pada level 1, terdapat musuh yang berupa tengkorak sebanyak 3 tengkorak. Musuh tersebut memiliki nilai healthpoint masing – masing sebesar 100. Jika musuh/monster terakhir berhasil dikalahkan, maka akan mengantar pemain menuju ke level 2.



Gambar 11. Tampilan Level 1

4.3. Tampilan Level 2 Pada Game

Pada Gambar 12 merupakan perancangan level 2 pada game. Pada level 2, terdapat musuh/monster yang berupa manusia batu sebanyak 3 manusia batu. Musuh/Monster tersebut memiliki nilai healthpoint masing – masing sebesar 100. Jika musuh/monster terakhir berhasil dikalahkan, maka akan mengantar pemain menuju ke level 3.



Gambar 12. Tampilan Level 2

4.4. Tampilan Level 3 Pada Game

Pada Gambar 13 merupakan perancangan level 3 pada game. Dalam level 3 game inilah terdapat

musuh/monster yang berupa bos sebanyak 1 unit. Dimana musuh/monster tersebut memiliki nilai healthpoint sebesar 200. Dan ketika musuh/monster berhasil dikalahkan, maka akan muncul tampilan game berhasil diselesaikan.



Gambar 13. Tampilan Level 3

4.5. Pengujian Kecerdasan Buatan

Tabel 9. Pengujian Kecerdasan Buatan

NO	FUNGSI	OUTPUT	HASIL
1	AI Finite State Machine pada musuh/monster level 1	Musuh/Monster mengejar player jika jarak player <3 units	Sesuai
2	AI Finite State Machine pada musuh/monster level 1	Musuh/Monster akan memukul jika jarak player <=0 units	Sesuai
3	AI Finite State Machine pada musuh/monster level 2	Musuh/Monster mengejar player jika jarak player <7 units	Sesuai
4	AI Finite State Machine pada musuh/monster level 2	Musuh/Monster akan memukul jika jarak player <1.5 units	Sesuai
5	AI Fuzzy pada musuh/monster level 3	Musuh/Monster akan battle jika nilai defusifikasi $0 \leq z \leq 25$ (jarak player dekat dengan kecepatan pelan & sedang)	Sesuai
6	AI Fuzzy pada musuh/monster level 3	Musuh/Monster akan standby jika nilai defusifikasi $26 \leq z \leq 50$ (jarak player dekat dengan kecepatan cepat atau jarak player sedang dengan kecepatan cepat)	Sesuai
7	AI Fuzzy pada musuh/monster level 3	Musuh/Monster akan patrol jika nilai defusifikasi $51 \leq z \leq 75$ (jarak player sedang dengan kecepatan pelan & sedang atau jarak player jauh dengan kecepatan pelan, sedang & cepat)	Sesuai

Berdasarkan Tabel 9, AI yang ada pada game sudah berfungsi dengan baik dengan prosentase 100%. Ketika permainan dimulai dan player mendekati musuh/monster, musuh/monster dapat menyerang player. *Health Bar* dari musuh/monster dan juga player dapat berkurang ketika mendapat serangan. Dengan demikian dapat disimpulkan bahwa semua AI (*Artificial Intelligence*) yang ada didalam game “Black Warrior” dapat berjalan sesuai dengan yang diharapkan.

4.6. Pengujian Fungsional

Tabel 10. Pengujian Fungsional

No	Item Uji	Keterangan
1	Main Menu	Berhasil
2	Start	Berhasil
3	Pause	Berhasil
4	Resume	Berhasil
5	Option	Berhasil
6	Credit	Berhasil
7	Exit	Berhasil
8	Backsound Effect Main Menu	Berhasil
9	Backsound Level 1	Berhasil
10	Backsound Level 2	Berhasil
11	Backsound Level 3	Berhasil
12	Backsound Musuh/Monster 1	Berhasil
13	Backsound Musuh/Monster 2	Berhasil
14	Backsound Musuh/Monster 3	Berhasil
15	Player Sound Effect	Berhasil
16	Enemy Sound Effect	Berhasil
17	Masuk Level 1	Berhasil
18	Masuk Level 2	Berhasil
19	Masuk Level 3	Berhasil
20	Tampilan Nyawa Player	Berhasil
21	Tampilan Nyawa Musuh/Monster	Berhasil
22	Item Level 3	Berhasil
23	Next Level	Berhasil
24	Control Maju (Up)/W	Berhasil
25	Control Mundur (Down)/S	Berhasil
26	Control Kiri (Left)/A	Berhasil
27	Control Kanan (Right)/D	Berhasil
28	Control Berlari (Run)/Shift	Berhasil
29	Control Memukul (Attack)/Left click Mouse	Berhasil

Hasil dari pengujian fungsional yang ada pada game sudah berfungsi dengan baik dan sesuai dengan tingkat prosentase 100% .

4.7. Pengujian User

Tabel 11. Pengujian User

No	Pertanyaan	Penilaian		
		Kurang	Cukup	Baik
1	Bagaimanakah tampilan game adventure “Black Warrior”?	-	8	2
2	Bagaimanakah pemahaman user saat memainkan game	-	4	6

	adventure “Black Warrior”?			
3	Bagaimanakah kemudahan pengoperasian kontrol Player pada game adventure “Black Warrior”?	2	4	4
4	Bagaimanakah kecerdasan buatan musuh/monster dalam mengejar saat player mendekat dengan jarak tertentu?	1	7	2
5	Bagaimanakah respon musuh/monster ketika menyerang player saat berada di jarak tertentu?	3	3	4
Jumlah		6 (12%)	26(52%)	18(36%)

Berdasarkan Tabel 11, didapatkan hasil pengujian user sebagai berikut, Kurang sebesar 12%, Cukup sebesar 52%, Baik sebesar 36%. Maka hasil pengujian terhadap 10 orang responden cenderung menilai game “Black Warrior” dengan nilai cukup dengan nilai prosentase 52%.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian dan beberapa pengujian yang telah dilakukan terhadap game “Black Warrior” didapatkan kesimpulan:

1. Implementasi game 3D bergenre adventure menggunakan metode *Finite State Machine* dan *Fuzzy Logic* di desktop dengan hasil fungsi dari menu, pergerakan unit pemain, dan musuh/monster seluruhnya berjalan sesuai dengan yang diharapkan.
2. Implementasi *Finite State Machine* dapat diterapkan dalam game level 1 dan 2 untuk player dan juga musuh/monster, seluruhnya sesuai dengan yang diharapkan.
3. Implementasi *Fuzzy Logic* dapat diterapkan dalam game level 3 untuk menggerakkan musuh/monster level 3 dengan aksi yang berbeda sesuai dengan kecepatan dan jarak player, seluruhnya sesuai dengan yang diharapkan.
4. Hasil dari pengujian user menunjukkan bahwa mayoritas user menilai game yang dikembangkan sudah cukup baik.

5.2. Saran

Berdasarkan penelitian dan pengujian terhadap game “Black Warrior”, maka masih ada banyak

kekurangan sehingga untuk pengembangan lebih lanjut disarankan:

1. *Game* dapat dikembangkan kembali menjadi *game* berbasis mobile sehingga dapat dimainkan di *smartphone*.
2. *Game* dapat ditambahkan dengan level – level baru dengan tambahan musuh/monster yang baru lagi sesuai dengan keinginan.

DAFTAR PUSTAKA

- [1] Blackman, S., 2013. *Beginning 3D Game Development with Unity4 : All-In-One, Multi-Platform Game Development*, Apress Media LLC, New York.
- [2] Hakiki, I., 2018, *Penerapan Metode Fuzzy Pada Game 3d "Simple Way"*. Jurnal Mahasiswa Teknik Informatika, Institut Teknologi Nasional Malang, Vol 3 No.1
- [3] Herlambang, M., 2019, *Penerapan Metode Finite State Machine Pada Game Dreadman*. Jurnal Mahasiswa Teknik Informatika, Institut Teknologi Nasional Malang, Vol 3 No.2
- [4] Kartika, R. D., 2017, *Pembuatan Game Adventure Kumachi No Shima Dengan Menggunakan Metode FSM (Finite State Machine)*. Jurnal Mahasiswi Teknik Informatika, Institut Teknologi Nasional Malang, Vol 1 No.1
- [5] Kristanto, Andri, 2004, *Kecerdasan Buatan*, Graha Ilmu, Yogyakarta.
- [6] Marzuki, & Chandra, F., 2014. *Game Berbasis Adventure Sebagai Pendukung Pembelajaran Pengenalan Kata Bahasa Inggris Untuk Anak Usia Dini*. Jurnal Sistem Informasi dan Telematika, 1-11.
- [7] Millington, I., & Funge, J. 2009. *Artificial Intelligence for Games*. CRC Press.
- [8] Naba, A., *Belajar Cepat Fuzzy Logic Menggunakan Matlab*. 2009. Penerbit Andi, Yogyakarta.
- [9] Rahadian, M. F., Suyatno, A., & Maharani, S. 2016. *Penerapan Metode Finite State Machine Pada Game "The Relationship"*. Jurnal Informatika Mulawarman, 14-22.
- [10] Wicaksono, A., Hariadi, M., & Mardi, S., 2013. *Strategi Menyerang Npc Game Fps Menggunakan Fuzzy Finite State Machine*. Seminar Nasional Teknologi Informasi dan Multimedia 2013, 25-30.
- [11] Wahono, R. S. (2006). Aspek dan Kriteria Penilaian Media Pembelajaran. [Online]. Diakses dari <http://romisatriawahono.net/2006/06/21/aspekdan-kriteria-penilaian-media-pembelajaran/>.
- [12] Yulsilviana, E., & Ekawati, H., 2019. *Penerapan Metode Finite State Machine (Fsm) Pada Game Agent Legenda Anak Borneo*, STMIK Widya Cipta Dharma Samarinda, Sebatik 2621-069X, 116-123