

# IMPLEMENTASI KLASIFIKASI BAYESIAN UNTUK STRATEGI MENYERANG JARAK DEKAT PADA NPC (*NON PLAYER CHARACTER*) MENGGUNAKAN UNITY 3D

Siti Asmiatun, M.Kom<sup>1</sup>

Fakultas Teknologi Informasi dan Komunikasi  
Universitas Semarang

Aria Hendrawan, M.Kom<sup>2</sup>

Fakultas Teknologi Informasi dan Komunikasi  
Universitas Semarang

**Abstract**— *In a game against the enemy attack strategy should be applied, making it more appealing game players to finish the game up with the objectives to be achieved. This study discusses the strategy of attacking at close range to the NPC (Non Player Character). In a game, especially for FPS (First Person Shooter), we need a strategy NPC, with the aim to make the game more attractive and realistic. Strike strategy in this research is to divide some of the behavior of NPCs attack when in the position closest to the enemy. This research applies Bayesian algorithm for classifying the behavior of the NPC attack. The classification is expected to improve the strategy against an enemy attack. NPC assault classification is divided into two offensive behavior is the behavior of hitting and biting behavior. As for the variables used in Bayesian classification is health points, attack points player and a distance obtained from NPC conditions. From the test results using testing method of Bayesian classification confusion matrix, with trial games 10 times, has resulted in the presentation level of accuracy in the confusion matrix reach a percentage of 80%. This proves that the behavior classification melee attack with Bayesian classification method can be applied with good results. Dalam sebuah game strategi penyerangan untuk melawan musuh harus diterapkan, sehingga lebih menarik pemain game untuk menyelesaikan permainan sampai dengan tujuan yang akan dicapai. Penelitian ini membahas tentang strategi menyerang jarak dekat untuk NPC (Non Player Character).*

**Keywords**— *game; classification; naive; bayes; strategi; menyerang; fps*

## I. PENDAHULUAN

Artificial Intelligence (AI) pada game FPS umumnya terdiri dari perencanaan *path*, mengambil *item*, menggunakan *item*, dan berperang [1]. Khusus untuk berperang NPC diharapkan mempunyai strategi-strategi khusus seperti halnya manusia [2][5]. Strategi yang dimaksud bisa berupa strategi mengejar lawan, menyerang lawan maupun menghindari lawan.

Model strategi menyerang bisa bermacam-macam, misalnya strategi menyerang dengan memancing serangan musuh, kemudian pada kondisi tertentu perilaku berubah menjadi menghindar. Jika NPC bergerak menghindar ke arah tim, maka NPC akan mendapatkan bantuan serangan dari rekan satu tim. Bagaimanapun model strategi menyerang, umumnya mempunyai tujuan akhir mengalahkan musuh [3]. Model strategi menyerang dalam sebuah game harus direncanakan untuk membuat game yang tidak membosankan bagi pemain.

Banyak penelitian yang membahas tentang strategi penyerangan salah satunya penelitian dari Yunifa dkk tentang strategi menyerang menggunakan logika fuzzy telah menghasilkan tingkat kemenangan hingga 80% ketika diujicobakan dengan musuh yang mempunyai perilaku umum yaitu menyerang dan menghindari tembakan [6]. Penelitian strategi menyerang juga dilakukan rianto dan norman menggunakan optimasi dengan Simulated Annealing telah menghasilkan kinerja strategi menyerang yang lebih baik dari segi waktu proses [4].

Meninjau penelitian dari siti asmiatun dkk tentang strategi menyerang jarak dekat menggunakan klasifikasi bayesian [9], terdapat beberapa kekurangan salah satunya belum diimplementasikan ke dalam tools game. Dari beberapa penelitian tersebut, penelitian ini difokuskan untuk implementasi strategi menyerang menggunakan klasifikasi dengan algoritma naive bayes menggunakan tools game engine unity. Maka dari itu, dengan pengembangan penelitian ini diharapkan nanti dapat menghasilkan sebuah game yang menarik dengan akurasi yang lebih baik lagi dari penelitian sebelumnya.

## II. LANDASAN TEORI

### A. Game

*Game* berasal dari kata bahasa Inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*). Game juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai

pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana game itu menarik untuk dimainkan secara maksimal.

Mengacu pada Adams (2010) game harus menghibur pemain, yaitu:

1. *Game* harus menghadirkan imajinatif, pengalaman koheren, sehingga desainer harus memiliki visi.
2. *Game* harus menjual dengan baik, sehingga desainer harus mempertimbangkan selera konsumen.
3. *Game* dengan lisensi harus bayar kembali biaya lisensi, sehingga desainer harus mengerti keuntungan lisensi dan memanfaatkannya sebagai keunggulan.

*Game* harus menawarkan tantangan kecerdasan dan pengalaman yang mulus, sehingga desainer harus mengerti teknologi.[7]

### B. NPC (Non Player Character)

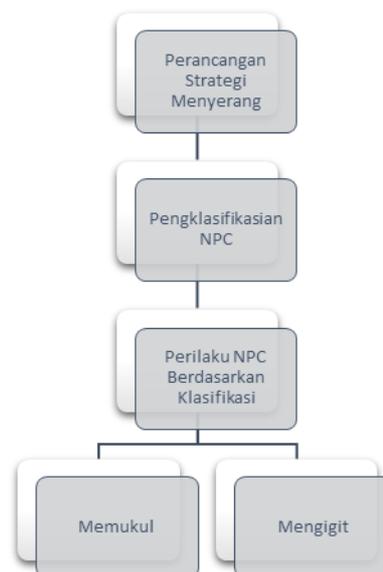
Autonomous character adalah jenis otonomous agent yang ditujukan untuk penggunaan komputer animasi dan media interaktif seperti games dan virtual reality. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk “*avatar*” dalam sebuah permainan atau virtual reality, tindakan yang diarahkan secara real time oleh pemain. Dalam permainan, karakter otonom biasanya disebut NPC (*Non-Player Character*) [3].

### C. Naive Bayesian Classification

*Naive Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai *Teorema Bayes*. Menurut Olson dan Delen (2008:102) menjelaskan *Naive bayes* untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam memproduksi perkiraan akhir dihitung sebagai jumlah frekuensi dari “master” tabel keputusan [8].

## III. METODE PENELITIAN

Metode penelitian yang digunakan untuk penelitian ini adalah dengan menggunakan pendekatan metode *Bayesian*. Seperti yang diilustrasikan dalam roadmap dibawah ini menjelaskan tentang tahap – tahap yang akan dilakukan dalam penelitian ini.



Gambar 3.1 Roadmap Penelitian

### A. Concept

Define Pada tahapan ini dilakukan analisis dari beberapa NPC untuk mendapatkan data yang menjadi variabel atau dataset dalam perhitungan bayesian. Tahapan ini terdiri dari :

#### 1. Perancangan Strategi Menyerang

Dalam menentukan strategi penyerangan diperlukan beberapa perancangan untuk persiapan sebelum menentukan strategi penyerangan. Penelitian ini akan menggunakan beberapa perilaku untuk strategi penyerangan yaitu perilaku Memukul dan Mengigit. Sedangkan parameter yang digunakan untuk menentukan perilaku adalah jarak, *Health Point* (HP) dan *Attack Power player* (APP) dari NPC, parameter dikelompokkan menjadi tiga kategori sebagai berikut:

Tabel 3.1 Kategori APP

APP < 5500	Lemah
APP > 5500	Kuat

Tabel 3.2 Kategori HP

HP < 55000	Besar
HP > 55000	Kecil

Tabel 3.3 Kategori Jarak

Jarak > 3.0	Jauh
Jarak < 1.5	Dekat

Tabel 3.4 Rule penentuan perilaku

Jika APP kuat dan jarak jauh dan HP jauh maka memukul
Jika APP kuat dan jarak jauh dan HP sedang maka memukul
Jika APP kuat dan jarak dekat dan HP dekat maka menggigit
Jika APP kuat dan jarak dekat dan HP jauh maka menggigit
Jika APP lemah dan jarak jauh dan HP sedang maka memukul
Jika APP lemah dan jarak jauh dan HP dekat maka memukul
Jika APP lemah dan jarak dekat dan HP jauh maka menggigit
Jika APP lemah dan jarak dekat dan HP sedang maka menggigit

2. Klasifikasi NPC (Non Player Character)

Dalam perhitungan algoritma Bayesian, terlebih dahulu menentukan jenis atribut yang akan dipakai untuk perhitungan Bayesian. Dalam kasus klasifikasi NPC, atribut yang digunakan adalah Jarak, Health Point (HP) dan Attack Power Player (APP) yang ada pada masing – masing NPC.

Tabel 3.5 Sambilan atribut NPC 1

Attack Power Player (DPP)	Jarak NPC	Health Point (HP)	Perilaku
kuat	jauh	Besar	memukul
kuat	dekat	Besar	menggigit
kuat	jauh	Kecil	memukul
lemah	jauh	Besar	Memukul
kuat	dekat	Besar	Menggigit
kuat	dekat	Kecil	menggigit
lemah	jauh	Besar	?

Dari dataset diatas, dapat dihitung dengan menggunakan teorema Bayes sebagai berikut :

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

Keterangan :

$P(C_i|X)$  = Probabilitas hipotesis  $C_i$  jika diberikan fakta atau record  $X$  (Posterior probability)

$P(X|C_i)$  = mencari nilai parameter yang memberi kemungkinan yang paling besar (likelihood)

$P(C_i)$  = Prior probability dari  $X$  (Prior probability)

$P(X)$  = Jumlah probability tuple yg muncul

Jadi perhitungan yang dihasilkan dari teorema bayes diatas adalah :

Asumsi :

$Y$  = Perilaku

$X_1$  = APP

$X_2$  = Jarak

$X_3$  = HP

Dengan metode bayes dapat dihitung :

$$P(X_1 = Lemah, X_2 = Jauh, X_3 = Besar | Memukul) =$$

$$\{1/3.3/3.2/3\}. 3/6 = 1$$

$$P(X_1 = Lemah, X_2 = Jauh, X_3 = Kuat | Mengigit) = \{0/3.0/1.2/1\}. 3/6 = 0$$

Sehingga menghasilkan table setiap perilaku sebagai berikut :

Tabel 3.6 Hasil Naïve Bayes Per perilaku

Perilaku	Hasil
Memukul	1
Menggigit	0

3. Penentuan Perilaku NPC (Non Player Character)

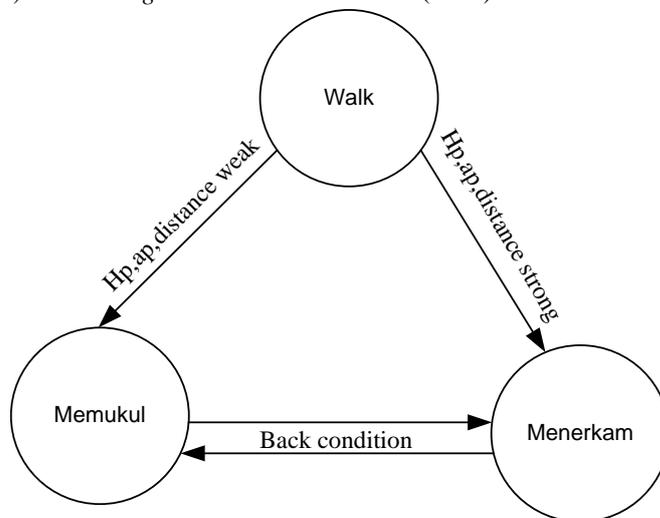
Berdasarkan dari tabel 3.6 nilai probabilitas yang paling tinggi dimiliki oleh kategori Memukul. Jadi perilaku yang akan digunakan NPC adalah perilaku Memukul.

IV. HASIL DAN PEMBAHASAN

Hasil implementasi yang dibahas dalam bab ini dideskripsikan melalui gambar tampilan print screen pada halaman tertentu sesuai dengan langkah-langkah pengembangan metode penelitian yang digunakan yaitu meliputi :

A. Pengumpulan Kebutuhan Game

1) Perancangan Finite State Machine (FSM)



Gambar 4.1 Finite State Machine penyerangan

2) Analisa Tingkat Kesulitan

Tingkat kesulitan akan berbeda di setiap levelnya dilihat dari jumlah musuh, kekuatan musuh, dan selang waktu setiap tembakan yang dilakukan musuh. Di bawah ini adalah perbandingan tingkat kesulitan tiap *level*, dapat dilihat pada tabel 4.1.

Tabel 4.1 Analisa kesulitan *Scoring Non Player Character*

Keterangan	Level 1	Level 2
<b>Jumlah Enemy</b>	12 Zombie, 1 Zombie Bos	20 Zombie, 1 Zombie Bos
<b>Jumlah Tembakan</b>	5 Kali	10 Kali
<b>Strategi Penyerangan Enemy</b>	Memukul dan Menerkam Player	
<b>Karakter Player</b>	Berjalan, Melompat, Menembak, Mengambil Darah, Membuang Senjata, Mengambil Senjata, Mengambil Obat.	

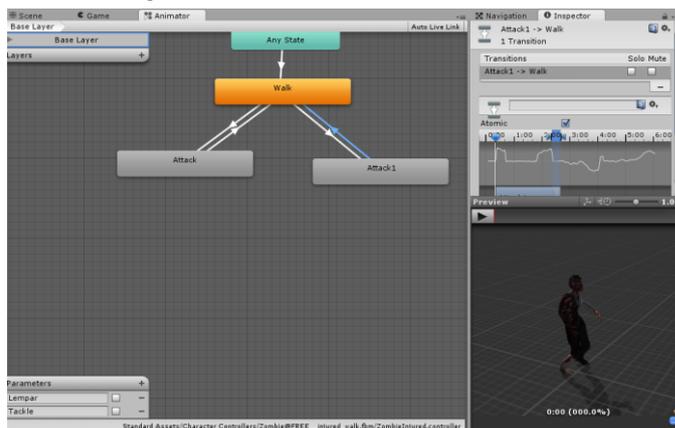
Tabel 4.2 Data training untuk klasifikasi bayes

HP	Jarak	AP	Perilaku
kuat	jauh	besar	memukul
kuat	dekat	Besar	mengigit
kuat	jauh	kecil	memukul
lemah	jauh	besar	Memukul
kuat	dekat	besar	Menggigit
kuat	dekat	kecil	menggigit
lemah	jauh	besar	Memukul
kuat	jauh	besar	memukul
kuat	dekat	Besar	mengigit
kuat	jauh	Kecil	memukul
kuat	dekat	Besar	mengigit
kuat	dekat	Kecil	mengigit

2) Modul *AI Enemy*

Pada tahapan ini *game* dibuat dengan menggunakan *Unity 3D v4.3.4f2*. Tahap ini akan dikembangkan modul-modul program untuk program utama serta program untuk masing-masing *agent* yang akan digunakan dalam *game* sesuai dengan perilaku zombie berdasarkan *naive Bayes Classifier* yang menggunakan data training pada tabel 4.5 diatas. Berikut flowchart script modul *naive Bayes Classifier* :

B. Pengumpulan *Model Game* (Material Collecting)



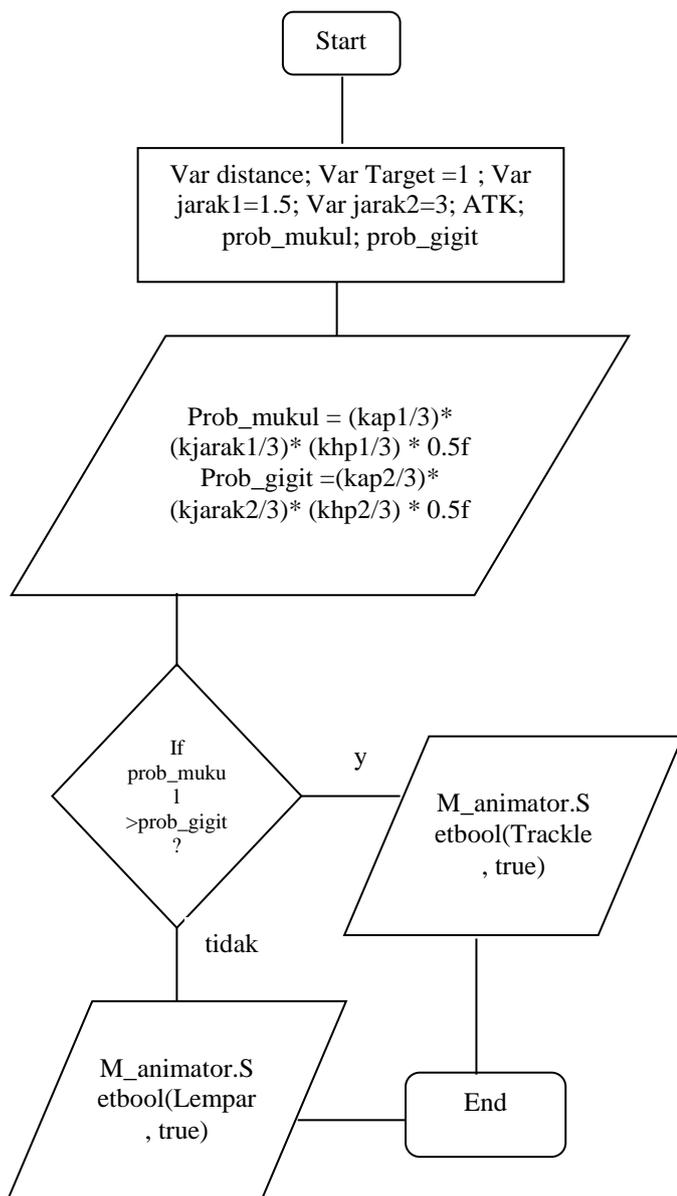
Gambar 4.2 Animator *Attack* ke *Walk*

C. Pembuatan Pengembangan *Game* (Assembly)

Pada tahap ini, ada beberapa data training yang akan di implementasikan dengan formula *naive Bayes Classifier*.

1) Dataset

Tahapan diawali dengan melakukan pengambilan data sample atau conntoh data dari konidisi zombie. Dimana data tersebut dapat dilihat pada tabel 4.3.



Gambar 4.3 Flowchart kalsifikasi naive bayes

#### D. Pengujian Skenario

Pengujian skenario bertujuan untuk mengetahui hasil perbandingan metode bayesian dengan pohon keputusan melalui *interface game*. Untuk menentukan kebenaran uji, diperlukan sebuah *rule* yang menjadi penentuan perilaku. Sebuah *rule* yang ditentukan oleh *developer game* tersebut bisa dilihat ditabel 3.1 sampai dengan tabel 3.4.

Untuk penentuan perilaku pada metode bayesian, dibutuhkan perbandingan nilai probabilistik perilaku satu dengan nilai probabilistik lainnya. Jadi kondisi tersebut bisa dikatakan diterima jika :

$Probabilitas\ mukul > probabilitas\ gigit = memukul$

$Probabilitas\ gigit > probabilitas\ mukul = menggigit$

Sedangkan untuk metode *rule-based* dapat dikatakan diterima dengan kondisi sebagai berikut :

$Nilai\ hp > nilai\ ap \ \&\&\ jarak < 1.5 = menggigit$

$Nilai\ hp < nilai\ ap \ \&\&\ jarak > 3 = memukul$

#### E. Pengujian Metode bayesian

Berdasarkan hasil pengujian skenario, diperoleh hasil bahwa terdapat 8 yang sesuai dengan kelas yang diberikan dan 2 hasil yang tidak sesuai dengan hasil yang diberikan. Berikut ini dijabarkan contoh *confusion matrix* dari hasil pengujian percobaan skenario dengan metode *naive bayes* dapat diambil kesimpulan sebagai berikut :

Tabel 4.3 Hasil validasi

Percobaan ke-	Perilaku yang diharapkan	Hasil perilaku	Kesimpulan
1	Menggigit	Menggigit	Valid
2	Memukul	Memukul	Valid
3	Menggigit	Menggigit dan terbang	Tidak valid
4	Memukul	Memukul	Valid
5	Memukul	Memukul	Valid
6	Menggigit	Menggigit	Valid
7	Memukul	Memukul	Valid
8	Menggigit	Menggigit	Valid
9	Menggigit	Menggigit terbang	Tidak valid
10	Memukul	Memukul	Valid
11	Menggigit	Menggigit	Valid
12	Memukul	Memukul	Valid
13	Menggigit	Menggigit dan terbang	Tidak valid

14	Memukul	Memukul	Valid
15	Memukul	Memukul	Valid
16	Menggigit	Menggigit	Valid
17	Menggigit	Menggigit dan terbang	Tidak valid
18	Menggigit	Menggigit	Valid
19	Memukul	Memukul	Valid
20	Memukul	Memukul	Valid

Dari tabel diatas, dapat disimpulkan dengan tabel *confusion matrix* dengan *Class a* dan *Class b* pada tabel *confusion matrix* berturut-turut menunjukkan class memukul dan menggigit yaitu sebagai berikut :

Tabel 2.4 Contoh Confusion matrix

		Predicted Class	
		Class a = memukul	Class b = menggigit
Actual Class	Class a	10	0
	Class b	6	4

Berdasarkan hasil *confusion matrix* terlihat bahwa 10 record pada class *a* diprediksi tepat sebagai class *a* dan sebanyak 0 record diprediksikan tidak tepat sebagai class *b*. Selanjutnya, seluruh record pada class *b* terdapat 4 record diprediksi tidak tepat pada class *a* dan 6 record diprediksi tepat sebagai class *b*.

Dari hasil tersebut dapat dihitung akurasi sebagai berikut :

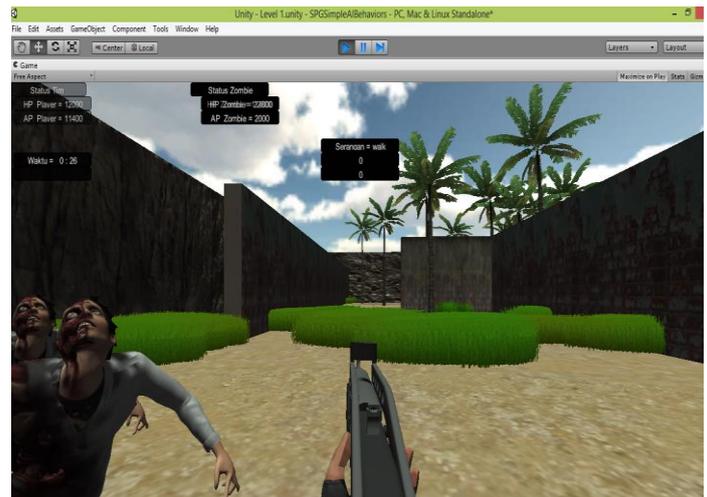
$$\text{Akurasi} = \frac{10 + 6}{10 + 0 + 6 + 4} \times 100 = 80\%$$

Jadi dapat disimpulkan bahwa perhitungan persentase tingkat akurasi pada *confusion matrix* mencapai nilai persentase sebesar 80%.

#### F. Distribusi

Pada tahap distribusi merupakan tahapan yang digunakan untuk membangun *game* kedalam sebuah aplikasi *game* yang sesuai dengan perangkat yang dibutuhkan. Untuk membangun game tersebut, langkah pertama yaitu membuka *build setting* yang terdapat dimenu *file*. Kemudian setelah muncul pengaturan *build*, terlebih dahulu kita menambahkan beberapa *file scene game* yang dibutuhkan melalui tombol *add current*. Dibagian bawah *build settings* terdapat pilihan *platform* yang digunakan untuk memilih perangkat distribusi yang

mendukung *game* tersebut. Kemudian yang terakhir pilih tombol *build* dibagian paling bawah.



Gambar 4.4 Display game

## V. KESIMPULAN DAN SARAN

Dari hasil penelitian *game* yang dibuat dan dikembangkan, maka dapat disimpulkan bahwa berdasarkan hasil testing, ada beberapa pengujian yang digunakan dalam *game* ini yaitu pengujian interface yang mnguji fungsi-fungsi *game* secara interface ke pemain berhasil berfungsi dengan baik. Kemudian ada pengujian skenario dari 10 kali percobaan *game* dapat melakukan skenario yang diujikan menggunakan metode bayesian, sedangkan dengan metode *rule-based* ada beberapa yang tidak bisa dilakukan. Untuk pengujian metode klasifikasi bayesian menggunakan pengujian *confusion matrix*. Dari percobaan permainan sebanyak 10 kali, telah menghasilkan persentase tingkat akurasi pada *confusion matrix* mencapai nilai persentase sebesar 80%.

Kedepannya diharapkan klasifikasi perilaku dapat ditambahkan paramater dengan metode lain sehingga ada perbandingan metode. Selain itu dibutuhkan tambahan perilaku menyerang untuk zombie, sehingga permainan yang dihasilkan lebih menarik pemain. Karena kelemahan dipenelitian ini, zombie didalam *game* tidak dapat menyerang musuh ketika posisi paling dekat dengan musuh karena *collider* zombie menyentuh *collider* musuh, sehingga mengakibatkan zombie terbang dan serangan zombie tidak fokus. Hal ini diperlukan penelitian lebih lanjut untuk menentukan jarak antara zombie dengan musuh.

## DAFTAR PUSTAKA

- [1] JinHyuk Hong dan Sung-Bae Cho. "Evolving Reactive NPCs for the Real-Time Simulation Game". CIG, 2005.
- [2] Yunifa Miftachul Arif, Fachrul Kurniawan dan Fresy Nugroho. "Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy". National Seminar on Electrical, Informatics, and Its Education, 2011.
- [3] Yunifa Miftachul Arif, Mochamad Hariadi dan Supeno Mardi S.N. "Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite Machine dan Logika Fuzzy". 2012
- [4] Rianto dan Noorman." Optimasi Strategi Menyerang NPC Group Pada Game Pertarungan Jarak Dekat Dengan Menggunakan Simmulated Annealing". Institut Teknologi Surabaya. 2012
- [5] McPartland M, and Gallagher M. "Creating a Multi-Purpose First Person Shooter Bot With Reinforcement Learning ". IEEE Symposium on Computational Intelligence and Games, pp. 143-150,2008.
- [6] Yunifa Miftachul Arif, Moch.Hariadi, Supeno M. "Integrasi Hierarchy Finite State Machine Dan Logika Fuzzy Untuk Desain Strategi NPC Game". Jurusan Teknik Elektro. Institut Teknologi Surabaya. 2011
- [7] Adams, E. (2010). *Fundamental Of Game Design, 2nd Edition*. Pearson Education, Inc, Berkeley.
- [8] D.L. Olson and D. Delen." Advanced data mining techniques". Springer Verlag. ISBN3540769161. 2008
- [9] Siti Asmiatun, Latius Hermawan dan Tri Daryanti. "Strategi Menyerang Jarak Dekat Menggunakan Klasifikasi Bayesian Pada NPC (*Non Player Character*)". Seminar Nasional Teknologi Informasi dan Komunikasi Terapan. ISBN979-26-0266-6. 2013