

AKSI PENYERANGAN *NON-PLAYER CHARACTER* (NPC) MENGGUNAKAN METODE *NAÏVE BAYES* PADA *SHOOTER GAME*

Edi Siswanto^{*1}, Alfa Faridh Suni²

^{1,2} Universitas Negeri Semarang, Semarang
Email: ¹edi99@students.unnes.ac.id, ²alfafs@mail.unnes.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 28 Juli 2020, diterima untuk diterbitkan: 16 November 2021)

Abstrak

Aksi penyerangan pada *non-player character* (NPC) merupakan salah satu substansi penting dalam pembuatan *game*. Dalam melakukan penyerangan diperlukan strategi khusus agar NPC tidak mudah dikalahkan. Salah satunya adalah adanya variasi serangan terhadap pemain. Salah satu metode yang digunakan untuk mengatur penyerangan NPC adalah *rulebase*. Metode *rulebase* dapat memberikan variasi serangan sesuai kondisi NPC, namun metode *rulebase* biasanya menghasilkan *behaviour* yang statis dan tidak adaptif jika terdapat kondisi baru. AI seperti ini akan mudah diprediksi dan repetitif sehingga menurunkan tingkat tantangan bermain *game*. Untuk mengatasi masalah tersebut banyak peneliti yang menggunakan teknik *learning*. Salah satunya menggunakan metode *naïve bayes*. Pada penelitian ini dilakukan penerapan metode *naïve bayes* sebagai strategi penyerangan NPC pada *shooter game*. Metode *naïve bayes* digunakan untuk keputusan serangan yang diambil oleh NPC. Adapun parameter yang digunakan untuk keputusan penyerangan adalah nyawa, jarak, jumlah granat, dan jumlah amunisi yang dimiliki NPC. Sedangkan keputusan penyerangan dibagi menjadi serangan tembak, serangan granat, dan serangan pisau. Hasil penelitian menunjukkan penerapan metode *naïve bayes* membuat NPC mampu melakukan penyerangan secara otonom jika terdapat kondisi baru dengan akurasi 80%. Penerapan metode *naïve bayes* juga lebih unggul dalam persentase kemenangan NPC dibanding metode *rulebase*. Tingkat kemenangan NPC menggunakan metode *naïve bayes* sebesar 60% sedangkan *rulebase* sebesar 16%.

Kata kunci: Aksi Penyerangan, NPC, *Naïve Bayes*, *Rulebase*

ATTACKING BEHAVIOUR OF NON-PLAYER CHARACTER (NPC) USING NAÏVE BAYES METHOD IN SHOOTER GAME

Abstract

Non-Player Character's (NPC) attacks behaviour is one important substance in making games. While NPC attacks needs specific strategy to not get defeated easily. One of the NPC attacks strategy is a variation of offense to player. One of the methods to manage the NPC attack strategy is rulebase. Rulebase method can give variations of the NPC attacks according in conditions, but rulebase method usually producing static behaviour and not adaptive where there is new condition. AI like this would easy too predictive and repetitive so that decrease the challenge of playing games. To overcome these problems, we use naïve bayes method. In this study, naïve bayes method are applied as an NPC's attack strategy to the shooter game. Naïve bayes method used for attack decisions taken by the NPC. The parameters used for the attack's decision are health point, distance, number of grenades, and number of ammunitions owned by the NPC. While attacks decision is divided into firing attacks, grenade attacks, and melee attacks. The results showed that the use naïve bayes method can attack autonomously if there are new condition with an accuracy of 80%. The implementation of naïve bayes method at NPC more superior than rulebase method in percentage of NPC winning. The NPC win rate uses the naïve bayes method is 60% while the rulebase is 16%.

Keywords: *Attacking Behaviour, NPC, Naïve Bayes, Rulebase*

1. PENDAHULUAN

Salah satu produk teknologi komputer yang perkembangannya cukup pesat adalah *game*. Ada banyak *genre game* yang telah dikembangkan salah

satunya *shooter game*. Dalam memainkan *shooter game* pemain diharuskan melakukan aksi cepat dalam merespon keadaan yang terjadi, sehingga memungkinkan pemain dapat melatih ketangkasan mereka (Frosi & Cristina, 2018). Pengembangan

shooter game tidak lepas dari agen NPC. NPC merupakan karakter dalam *game* yang tidak dapat dimainkan oleh pemain dan berperan sebagai musuh atau lawan (Liarokapis, Debattista, Vourvopoulos, Petridis, & Ene, 2014). Agar NPC berperilaku cerdas dibutuhkan kecerdasan buatan dalam pembuatan *game*. Kecerdasan buatan atau *Artificial Intelligence* (AI) adalah model di dalam sistem yang belajar dari pengalaman dan mampu membuat keputusan dari inputan baru (Safrizal & Setiawan, 2019). Pada kasus *shooter game*, perilaku menyerang merupakan substansi penting dalam pengembangan *game* untuk mewujudkan *game* menjadi realistis (Asmiatun, Hermawan, & Daryatni, 2103). Model perilaku penyerangan NPC pada *shooter game* biasanya dimodelkan dengan *finite state machine* (FSM). FSM merupakan alur untuk menggambarkan perpindahan *state* perilaku NPC. Seperti dari mengejar berubah menjadi menyerang karena mendapat kondisi tertentu.

Permasalahan yang muncul pada saat menyerang adalah bagaimana membuat NPC agar tidak mudah dikalahkan. NPC yang hanya dapat menyerang saja mudah dikalahkan oleh pemain. Beberapa inovasi telah dilakukan pada agen NPC dari segi *behaviour* sebagai solusi untuk menciptakan NPC yang lebih agresif. Salah satunya adalah memberikan variasi serangan pada agen NPC (Wicaksono, Hariadi, & Mardi, 2013).

Salah satu metode yang digunakan untuk merancang variasi serangan NPC yaitu menggunakan *rulebase* (Yunanta, 2017). Penerapan *rulebase* menghasilkan sebuah kecerdasan buatan yang dapat membuat variasi penyerangan NPC sesuai kondisi yang telah ditentukan. *Rulebase* merupakan sistem yang berisi aturan-aturan yang disimpan dalam bentuk pengetahuan untuk dipelajari oleh AI (Roastiningsih, Gregorius, & Wijaya, 2013). Namun metode *rulebase* biasanya menghasilkan *behaviour* yang statis dan tidak adaptif jika terdapat kondisi baru. AI seperti ini akan mudah diprediksi dan repetitif sehingga menurunkan tingkat tantangan bermain *game* (Wang & Tang, 2015). Selain itu, terdapat penelitian yang menunjukkan penerapan *rulebase* sebagai kecerdasan buatan memberikan tingkat kemenangan NPC sebesar 22% sampai 25% (Abdi, et al., 2017; Putera dan Herumurti, 2018). Berdasarkan hal tersebut banyak peneliti yang memanfaatkan teknik *learning*. Teknik *learning* tersebut diterapkan menggunakan algoritma klasifikasi seperti *k-nearest neighbor* (K-NN), *decision tree*, *support vector machine* (SVM), dan *naïve bayes*. Dari beberapa algoritma klasifikasi tersebut metode *naïve bayes* relatif lebih unggul (Ashari, Paryudi, & Tjoa, 2013; Osisanwo, Akinsola, Awolede, & Himnikaiye, 2017). Seperti penelitian Sanjaya, et. al. (2019) yang menerapkan metode *naïve bayes* sebagai AI pada *racing game* untuk keputusan pengereman (*braking decision*). Hasil penelitian menunjukkan bahwa metode *naïve*

bayes mampu memberikan keputusan pengereman dengan baik tanpa menabrak dinding hasil dari pembelajaran data *training*. Metode *naïve bayes* memiliki kemampuan klasifikasi data dengan akurasi yang tinggi (Gata, et al., 2019). Selain itu metode *naïve bayes* dapat dilatih dengan *dataset* kecil (Kaviani & Dhotre, 2017). Asmiatun dan Hendrawan (2016) juga menerapkan metode *naïve bayes* untuk strategi menyerang NPC. Parameter yang digunakan pada penelitian tersebut adalah kekuatan serang, nyawa, dan jarak. Sedangkan klasifikasi perilaku penyerangan terdiri dari memukul dan menggigit. Pada penelitian tersebut NPC mampu melakukan penyerangan sesuai kondisi yang dialami. Namun pada penelitian tersebut strategi penyerangan hanya pada jarak dekat, sehingga NPC hanya dapat menyerang dari jarak dekat saja. Selain itu belum diketahui tingkat kemenangan NPC dari penggunaan metode *naïve bayes* pada penelitiannya.

Meninjau latar belakang tersebut, akan dilakukan penelitian dengan menerapkan metode *naïve bayes* untuk aksi penyerangan NPC pada *shooter game* ber-genre *Third Person Shooter* (TPS). Metode *naïve bayes* digunakan untuk pengambilan keputusan penyerangan baik dari jarak dekat maupun dari jarak jauh. Serta akan dilakukan uji coba pada simulasi *game* untuk mengetahui tingkat kemenangan NPC dibandingkan metode sebelumnya yaitu *rulebase*.

2. METODE PENELITIAN

2.1. Penentuan Parameter Naïve Bayes

Dalam perhitungannya metode *naïve bayes* memerlukan parameter. Parameter yang digunakan pada penelitian ini antara lain jarak, nyawa, jumlah granat, dan jumlah amunisi. Parameter tersebut merupakan atribut pada NPC yang sering mengalami perubahan nilai dalam *game* sehingga sangat cocok digunakan untuk pengambilan keputusan serangan NPC. Masing-masing parameter yang telah mengalami perubahan akan dibagi menjadi beberapa kriteria.

Tabel 1. Parameter Jarak

jarak	kriteria
$12,0 < \text{jarak} \leq 23,0$	Jauh
$3,0 < \text{jarak} \leq 12,0$	Sedang
$0,1 \leq \text{jarak} \leq 3,0$	Dekat

Tabel 2. Parameter Nyawa

nyawa	kriteria
$350 < \text{nyawa} \leq 500$	Besar
$150 < \text{nyawa} \leq 350$	Sedang
$1 \leq \text{nyawa} \leq 150$	Kecil

Tabel 3. Parameter Jumlah Granat

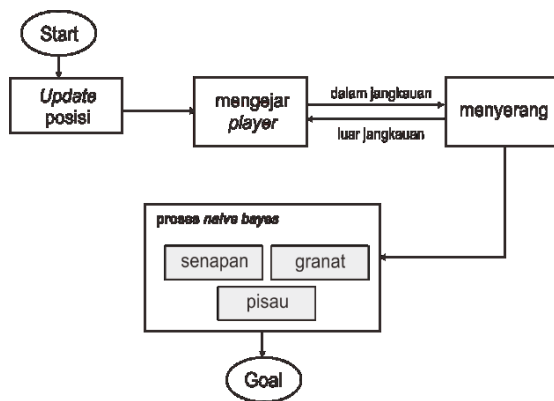
Jumlah granat	kriteria
$10 < \text{Jumlah granat} \leq 15$	Banyak
$5 < \text{Jumlah granat} \leq 10$	Sedang
$1 \leq \text{Jumlah granat} \leq 5$	Sedikit
Jumlah granat = 0	Habis

Tabel 4. Parameter Jumlah Amunisi

Jumlah amunisi	kriteria
$30 < \text{Jumlah amunisi} \leq 50$	Banyak
$15 < \text{Jumlah amunisi} \leq 30$	Sedang
$1 \leq \text{Jumlah amunisi} \leq 15$	Sedikit
Jumlah amunisi = 0	Habis

2.2. Perancangan Perilaku NPC

Perancangan perilaku NPC digambarkan dalam bentuk blok diagram menggunakan metode *finite state machine* (FSM). Blok diagram digambarkan berdasarkan prinsip kerja sistem mulai dari sistem dijalankan sampai terjadi penyerangan NPC terhadap *player*.



Gambar 1. Blok diagram perilaku NPC

Gambar 1 merupakan desain perilaku NPC. Pada saat sistem dijalankan terjadi perubahan posisi NPC (*update posisi*) karena terjadi *state* NPC bergerak mengejar *player*. NPC akan bertransisi ke *state* menyerang apabila sudah berada di radius serang atau kondisi dalam jangkauan serang. NPC akan bertransisi kembali ke *state* mengejar *player* apabila di luar jangkauan serang.

2.3. Pembuatan Dataset

Tahapan awal dalam perhitungan *naïve bayes* diawali dengan pengambilan data dari kondisi NPC. Data tersebut berisi nilai-nilai parameter. Pengambilan data dilakukan dengan teknik *sampling*. Setelah pengambilan data kemudian akan diberikan klasifikasi serangan NPC secara manual oleh *user* yang memiliki *user experience* dalam *game*. Data yang telah diklasifikasi secara manual akan digunakan sebagai *dataset*. Saleh (2015) menggunakan *dataset* sebanyak 60 data dalam penelitiannya untuk perhitungan *naïve bayes*, maka penelitian ini akan digunakan *dataset* sebanyak 60. Kemudian dari 60 *dataset* akan dibagi menjadi 2 *subset* data yaitu data *training* dan data *testing* menggunakan metode *simple split*. Metode *simple split* memilih dua per tiga sebagai data *training* dan satu per tiga sebagai data *testing* (Talib, Mayadewi, & Rosely, 2016). Data *training* dapat dilihat pada Tabel 5 sedangkan data *testing* dapat dilihat pada Tabel 6. Data *training* yang telah ditentukan akan

diterapkan dalam sistem sebagai data pembelajaran NPC dalam memprediksi serangan yang akan diambil. Sedangkan data *testing* akan digunakan sebagai data uji ketika NPC dalam kondisi baru.

Tabel 5. Data *Training*

No	Nyawa	Jarak	Jumlah Granat	Jumlah Amunisi	Serangan
1	Besar	Jauh	Sedikit	Sedang	Tembak
2	Besar	Sedang	Sedang	Sedang	Granat
3	Besar	Sedang	Habis	Habis	Pisau
4	Sedang	Sedang	Sedang	Banyak	Tembak
5	Kecil	Jauh	Banyak	Sedikit	Granat
6	Sedang	Dekat	Habis	Banyak	Pisau
7	Sedang	Sedang	Habis	Sedikit	Tembak
8	Kecil	Jauh	Banyak	Sedang	Granat
9	Kecil	Jauh	Habis	Habis	Pisau
10	Kecil	Sedang	Habis	Sedikit	Tembak
11	Besar	Jauh	Sedang	Habis	Granat
12	Besar	Dekat	Sedikit	Sedikit	Pisau
13	Besar	Sedang	Habis	Sedikit	Tembak
14	Sedang	Jauh	Banyak	Habis	Granat
15	Sedang	Dekat	Habis	Sedikit	Pisau
16	Sedang	Jauh	Habis	Sedang	Tembak
17	Besar	Jauh	Banyak	Sedang	Tembak
18	Kecil	Dekat	Sedikit	Sedang	Pisau
19	Kecil	Dekat	Sedikit	Sedikit	Pisau
20	Kecil	Sedang	Habis	Sedang	Tembak
21	Sedang	Sedang	Banyak	Sedang	Granat
22	Besar	Jauh	Sedikit	Banyak	Tembak
23	Besar	Sedang	Banyak	Sedang	Granat
24	Besar	Dekat	Sedang	Habis	Pisau
25	Kecil	Sedang	Sedikit	Banyak	Tembak
26	Kecil	Jauh	Sedikit	Sedikit	Granat
27	Sedang	Dekat	Sedikit	Sedang	Pisau
28	Kecil	Jauh	Sedang	Banyak	Tembak
29	Besar	Sedang	Banyak	Sedikit	Granat
30	Besar	Jauh	Habis	Habis	Pisau
31	Sedang	Jauh	Sedikit	Banyak	Tembak
32	Sedang	Sedang	Sedang	Sedikit	Granat
33	Kecil	Dekat	Habis	Banyak	Pisau
34	Kecil	Jauh	Banyak	Banyak	Tembak
35	Besar	Sedang	Sedang	Habis	Granat
36	Besar	Dekat	Sedikit	Sedang	Pisau
37	Kecil	Jauh	Habis	Sedikit	Tembak
38	Kecil	Sedang	Sedang	Banyak	Granat
39	Sedang	Sedang	Sedang	Sedang	Granat
40	Besar	Dekat	Banyak	Sedang	Pisau

Tabel 6. Data *Testing*

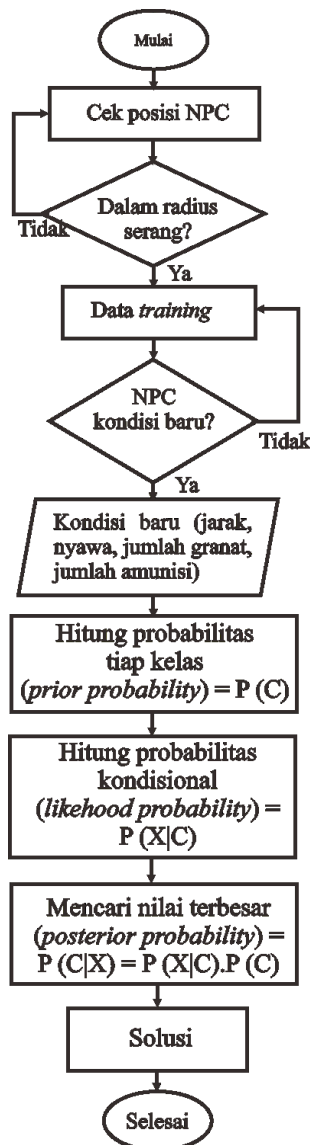
No	Nyawa	Jarak	Jumlah granat	Jumlah amunisi	Serangan
1	Besar	Jauh	Banyak	Banyak	Tembak
2	Besar	Sedang	Sedikit	Sedikit	Granat
3	Besar	Sedang	Sedikit	Banyak	Tembak
4	Besar	Sedang	Sedang	Sedikit	Granat
5	Besar	Dekat	Sedikit	Banyak	Tembak
6	Besar	Dekat	Banyak	Sedikit	Pisau
7	Sedang	Jauh	Banyak	Banyak	Tembak
8	Sedang	Jauh	Sedikit	Sedang	Tembak
9	Sedang	Jauh	Sedikit	Habis	Granat
10	Sedang	Sedang	Banyak	Sedikit	Granat
11	Sedang	Dekat	Banyak	Banyak	Pisau
12	Sedang	Dekat	Sedang	Sedikit	Pisau
13	Sedang	Dekat	Sedikit	Habis	Pisau
14	Kecil	Jauh	Sedikit	Sedang	Tembak
15	Kecil	Sedang	Banyak	Banyak	Tembak
16	Kecil	Sedang	Sedang	Sedang	Granat
17	Kecil	Dekat	Banyak	Banyak	Tembak
18	Kecil	Dekat	Sedang	Banyak	Pisau
19	Kecil	Dekat	Sedang	Sedikit	Pisau
20	Besar	Jauh	Banyak	Banyak	Pisau

2.4. Implementasi *Naïve Bayes* pada NPC

Ketika menyerang maka terjadi perubahan kondisi baik jarak pemain dengan NPC, nyawa, jumlah granat, dan jumlah amunisi yang dimiliki NPC. Perubahan tersebut dijadikan parameter sebagai masukan perhitungan proses *naïve bayes*.

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (1)$$

Persamaan 1 merupakan persamaan *naïve bayes*. $P(C|X)$ merupakan probabilitas *posterior* yaitu nilai yang akan dijadikan keputusan serangan NPC. $P(C)$ merupakan probabilitas *prior*, $P(X|C)$ merupakan probabilitas *likelihood* yaitu peluang kemunculan nilai parameter X pada kelas tertentu C , sedangkan $P(X)$ merupakan *evidence* yaitu peluang munculnya parameter X dari seluruh pengamatan. Langkah kerja proses *naïve bayes* secara detail dapat dilihat dalam bentuk *flowchart* pada Gambar 2.



Gambar 2. Proses *naïve bayes* pada NPC

Pada Gambar 2 dapat dilihat bahwa sistem akan mengecek posisi NPC apakah berada pada radius serang atau diluar radius serang. Ketika NPC berada pada radius serang maka NPC akan menyerang *player* sesuai kondisi pada saat itu.

3. HASIL DAN PEMBAHASAN

Pada tahap hasil dan analisis akan dipaparkan hasil simulasi *game*, hasil pengujian akurasi, dan hasil pengujian tingkat kemenangan NPC.

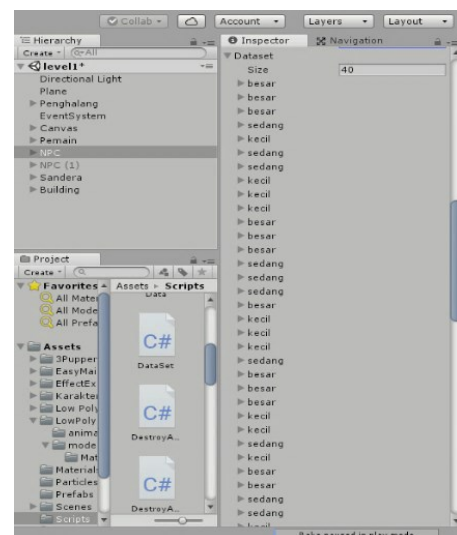
3.1. Hasil Simulasi *Game*

Setelah dilakukan perancangan kemudian akan diimplementasikan kedalam *tools game engine Unity3D*. Tampilan simulasi *game* dapat dilihat pada Gambar 3 sampai 4.



Gambar 3. Tampilan *gameplay*

Gambar 3 merupakan tampilan awal saat *game* dijalankan. Terlihat bahwa hanya ada 1 pemain dan 1 NPC pada simulasi *game* ini. Selanjutnya gambar 4 merupakan tampilan penerapan data *training* pada *tools game*. Data *training* disimpan dalam bentuk *file* yang berekstensi *.txt* sebagai *database* dalam proses perhitungan *naïve bayes*.



Gambar 4. Implementasi data *training*

3.2. Hasil Pengujian Akurasi

Pengujian akurasi berfungsi untuk mengetahui ketepatan NPC dalam memprediksi serangan dengan benar terhadap keseluruhan data. Pengujian akurasi diperoleh dengan melakukan percobaan yaitu

memainkan *game* dimana NPC sudah diberikan data *training*. Kemudian kondisi NPC akan dimanipulasi agar mendapat data baru sesuai data *testing* dan mengamati respon serangan yang dilakukan NPC. Berdasarkan pengujian yang telah dilakukan didapatkan hasil seperti pada Tabel 7.

Tabel 7. Hasil pengujian akurasi

No	Nxyawa	Jarak	Jumlah granat	Jumlah amunisi	Klasifikasi manual	Klasifikasi sistem	Kesimpulan
1	Besar	Jauh	Banyak	Banyak	Tembak	Tembak	Sesuai
2	Besar	Sedang	Sedikit	Sedikit	Granat	Tembak	Tidak sesuai
3	Besar	Sedang	Sedikit	Banyak	Tembak	Tembak	Sesuai
4	Besar	Sedang	Sedikit	Sedikit	Granat	Granat	Sesuai
5	Besar	Dekat	Sedikit	Banyak	Tembak	Pisau	Tidak sesuai
6	Besar	Dekat	Banyak	Sedikit	Pisau	Pisau	Sesuai
7	Sedang	Jauh	Banyak	Banyak	Tembak	Tembak	Sesuai
8	Sedang	Jauh	Sedikit	Sedang	Tembak	Tembak	Sesuai
9	Sedang	Jauh	Sedikit	Habis	Granat	Pisau	Tidak sesuai
10	Sedang	Sedang	Banyak	Sedikit	Granat	Granat	Sesuai
11	Sedang	Dekat	Banyak	Banyak	Pisau	Pisau	Sesuai
12	Sedang	Dekat	Sedang	Sedikit	Pisau	Pisau	Sesuai
13	Sedang	Dekat	Sedikit	Habis	Pisau	Pisau	Sesuai
14	Kecil	Jauh	Sedikit	Sedang	Tembak	Tembak	Sesuai
15	Kecil	Sedang	Banyak	Banyak	Tembak	Tembak	Sesuai
16	Kecil	Sedang	Sedang	Sedang	Granat	Granat	Sesuai
17	Kecil	Dekat	Banyak	Banyak	Tembak	Pisau	Tidak sesuai
18	Kecil	Dekat	Sedang	Banyak	Pisau	Pisau	Sesuai
19	Kecil	Dekat	Sedang	Sedikit	Pisau	Pisau	Sesuai
20	Kecil	Dekat	Sedikit	Habis	Pisau	Pisau	Sesuai

Hasil pengujian akurasi pada Tabel 7 kemudian akan dianalisis menggunakan metode *confussion matrix*. Tujuannya adalah untuk melihat performa metode *naïve bayes* pada pengambilan keputusan serangan NPC. Pada penelitian ini akan digunakan model *confussion matrix* 3 x 3 (*multiclass*) karena klasifikasi serangan ada 3 kelas yaitu serangan tembak, serangan granat, dan serangan pisau. Model *confussion matrix* 3 x 3 dapat dilihat pada Tabel 8.

Tabel 8. *Confussion matrix* (Iskandar & Suprpto, 2015)

		<i>Predicted class</i>		
		<i>Class A</i>	<i>Class B</i>	<i>Class C</i>
<i>Actual class</i>	<i>Class A</i>	AA	AB	AC
	<i>Class B</i>	BA	BB	BC
	<i>Class C</i>	CA	CB	CC

Keterangan:

Class A = Kelas dengan nama A

Class B = Kelas dengan nama B

Class C = Kelas dengan nama C

AA = Jumlah data dengan nilai sebenarnya A dan diprediksi tepat sebagai kelas A

AB = Jumlah data dengan nilai sebenarnya A dan diprediksi sebagai kelas B

AC = Jumlah data dengan nilai sebenarnya A dan diprediksi sebagai kelas C

BA = Jumlah data dengan nilai sebenarnya B dan diprediksi sebagai kelas A

BB = Jumlah data dengan nilai sebenarnya B dan diprediksi tepat sebagai kelas B

BC = Jumlah data dengan nilai sebenarnya B dan diprediksi sebagai kelas C

CA = Jumlah data dengan nilai sebenarnya C dan diprediksi sebagai kelas A

CB = Jumlah data dengan nilai sebenarnya C dan diprediksi sebagai kelas B

CC = Jumlah data dengan nilai sebenarnya C dan diprediksi tepat sebagai kelas C

Adapun hasil analisis menggunakan *confussion matrix* ditunjukkan pada Tabel 9.

Tabel 9. Nilai *confussion matrix*

		<i>Klasifikasi sistem (predicted class)</i>		
		<i>Class A</i> (Tembak)	<i>Class B</i> (Granat)	<i>Class C</i> (Pisau)
<i>Klasifikasi manual (actual class)</i>	<i>Class A</i> (Tembak)	6	0	2
	<i>Class B</i> (Granat)	1	3	1
	<i>Class C</i> (Pisau)	0	0	7

Berdasarkan tabel 9 hasil *confussion matrix* didapatkan bahwa 6 data pada kelas serangan tembak diprediksi tepat sebagai kelas serangan tembak dan sebanyak 3 data pada kelas serangan granat diprediksi tepat sebagai serangan granat serta sebanyak 7 data serangan pisau diprediksi tepat

sebagai serangan pisau. Selain itu terdapat 2 data klasifikasi manual pada serangan granat yang diprediksi tidak tepat sebagai serangan granat yaitu 1 data diprediksi sebagai serangan tembak dan 1 data diprediksi sebagai serangan pisau. Terdapat juga 2 data pada klasifikasi manual serangan tembak yang diprediksi tidak tepat sebagai kelas serangan tembak yaitu 2 data diprediksi sebagai sebagai serangan pisau dan 0 data diprediksi sebagai serangan granat. Sedangkan semua data pada klasifikasi manual serangan pisau diprediksi tepat sebagai serangan pisau. Dari hasil tersebut dapat dihitung akurasi sesuai persamaan 2.

$$\text{Akurasi} = \frac{AA + BB + CC}{AA + AB + AC + BA + BB + BC + CA + CB + CC} \times 100\% \quad (2)$$

$$\begin{aligned} \text{Akurasi} &= \frac{6 + 3 + 7}{6 + 0 + 2 + 1 + 3 + 1 + 0 + 0 + 7} \times 100\% \\ &= \frac{16}{20} \times 100\% \\ &= 80\% \end{aligned}$$

Berdasarkan perhitungan akurasi diperoleh nilai akurasi sebesar 80%. Nilai akurasi ini sama seperti nilai akurasi pada penelitian Asmiatun & Hendrawan (2016) namun pada penelitian tersebut belum diketahui tingkat kemenangan NPC.

3.3. Hasil Pengujian Tingkat Kemenangan NPC

Pengujian tingkat kemenangan NPC dilakukan dengan memainkan *game* oleh pemain (pakar *game*). Selama pengujian akan dicatat hasil percobaan (menang atau kalah) dan sisa nyawa pemain. Uji coba dilakukan oleh 5 orang *gamers*, masing-masing melakukan uji coba sebanyak 10 kali secara bergantian pada tiap metode. Sehingga didapatkan sebanyak 50 kali percobaan untuk setiap metode. Jumlah percobaan ini sudah cukup untuk melihat perbedaan yang signifikan pada tingkat kemenangan NPC dari masing-masing metode uji coba.

Adapun yang akan dijadikan *rulebase* sebagai AI pada NPC adalah data *training* pada metode *naïve bayes*. Sehingga terdapat 40 *rulebase*. Contoh *rulebase* yang digunakan:

1. **If** (nyawa = besar **and** jarak = jauh **and** jumlah granat = sedikit **and** jumlah amunisi = sedang) **then** serangan tembak
2. **If** (nyawa = besar **and** jarak = sedang **and** jumlah granat = sedang **and** jumlah amunisi = sedang) **then** serangan granat
3. **If** (nyawa = besar **and** jarak = sedang **and** jumlah granat = habis **and** jumlah amunisi = habis) **then** serangan pisau
4. **If** (nyawa = sedang **and** jarak = sedang **and** jumlah granat = sedang **and** jumlah amunisi = banyak) **then** serangan tembak

5. **If** (nyawa = besar **and** jarak = sedang **and** jumlah granat = habis **and** jumlah amunisi = habis) **then** serangan pisau
6. **If** (nyawa = sedang **and** jarak = dekat **and** jumlah granat = habis **and** jumlah amunisi = banyak) **then** serangan pisau
7. **If** (nyawa = sedang **and** jarak = sedang **and** jumlah granat = habis **and** jumlah amunisi = sedikit) **then** serangan tembak
8. **If** (nyawa = kecil **and** jarak = jauh **and** jumlah granat = banyak **and** jumlah amunisi = sedang) **then** serangan granat
9. **If** (nyawa = kecil **and** jarak = jauh **and** jumlah granat = habis **and** jumlah amunisi = habis) **then** serangan pisau
10. **If** (nyawa = kecil **and** jarak = sedang **and** jumlah granat = habis **and** jumlah amunisi = sedikit) **then** serangan tembak

Berdasarkan hasil uji coba antara pemain dengan kedua metode, diperoleh jumlah kemenangan NPC sebanyak 30 kali dan jumlah kekalahan sebanyak 20 kali pada metode *naïve bayes* sedangkan *rulebase* NPC mendapat kemenangan sebanyak 8 kali serta 42 kali kalah dari 50 kali percobaan. Sehingga tingkat kemenangan NPC dapat dihitung sesuai persamaan 3 (Abdi, et al., 2017).

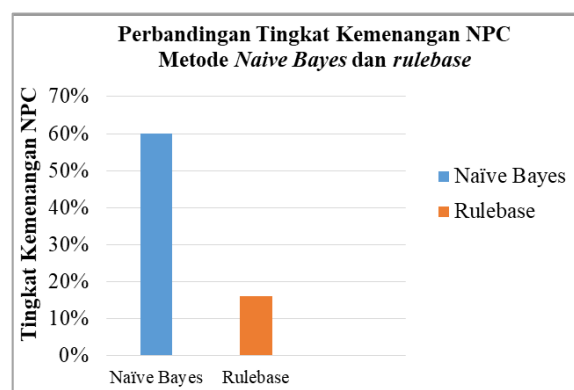
$$\% \text{Kemenangan} = \frac{\text{Jumlah kemenangan NPC}}{\text{Jumlah semua pertandingan}} \times 100\% \quad (3)$$

$$\text{NPC Naïve bayes} = \frac{30}{50} \times 100\% = 60\%$$

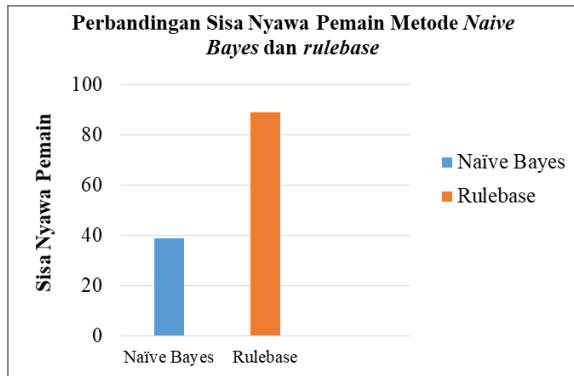
$$\text{NPC Rulebase} = \frac{8}{50} \times 100\% = 16\%$$

Tabel 10. Hasil uji coba pemain

metode	Jumlah NPC menang	Jumlah NPC kalah	Tingkat kemenangan NPC	Jumlah sisa nyawa pemain
<i>Naïve bayes</i>	30	20	60%	38,9
<i>Rulebase</i>	8	42	16%	89,12



Gambar 5. Perbandingan tingkat kemenangan NPC



Gambar 6. Perbandingan sisa nyawa pemain

Berdasarkan pengujian tingkat kemenangan NPC dengan menggunakan metode *naïve bayes* didapatkan tingkat kemenangan NPC yang lebih besar jika dibandingkan metode *rulebase*. Gambar 5 menunjukkan bahwa tingkat kemenangan NPC pada metode *naïve bayes* memiliki tingkat persentase lebih besar yaitu sebesar 60% dibandingkan metode *rulebase*. Hal ini menunjukkan bahwa NPC pada metode *naïve bayes* memiliki jumlah kemenangan yang lebih banyak dari setiap percobaan. Oleh karena itu dapat dikatakan bahwa metode *naïve bayes* memberikan tingkat kemenangan pada NPC yang lebih tinggi dibandingkan metode *rulebase* sebagai kecerdasan buatan pada NPC. Selain menghitung persentase tingkat kemenangan NPC, dilakukan pula menghitung jumlah sisa nyawa dari pemain. Tujuannya adalah mengetahui seberapa agresif NPC dalam melakukan serangan ke pemain. Apabila diakhir pertandingan sisa nyawa pemain sedikit maka NPC lebih agresif dalam melakukan serangan.

Hasil uji coba diperoleh hasil bahwa penerapan metode *naïve bayes* membuat NPC lebih agresif dalam melakukan serangan. Hal ini terbukti dari rata-rata sisa nyawa pemain jika melawan metode *naïve bayes* sisa nyawa pemain lebih sedikit dibandingkan melawan *rulebase*. Sisa nyawa pemain pada metode *naïve bayes* rata-rata sisa 38,9 sedangkan pada *rulebase* rata-rata sisa 89,12 seperti yang ditunjukkan pada Gambar 6. Pada metode *naïve bayes* NPC dapat melakukan serangan secara otonom jika terjadi kondisi baru sehingga NPC lebih agresif dalam menyerang pemain, sedangkan penggunaan *rulebase*, NPC bisa menyerang apabila telah memenuhi *rulebase* yang dibuat. Sehingga NPC belum adaptif jika terjadi kondisi baru.

4. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi metode *naïve bayes* pada *game shooter* maka diperoleh simpulan bahwa:

1. Implementasi metode *naïve bayes* berhasil diterapkan sebagai kecerdasan buatan untuk strategi penyerangan NPC dalam pengambilan keputusan serangan dengan akurasi sebesar 80%.

2. Metode *naïve bayes* memberikan tingkat kemenangan NPC yang lebih baik dibanding menggunakan *rulebase*. Metode *naïve bayes* mendapat tingkat kemenangan sebesar 60%, sedangkan *rulebase* sebesar 16%.

Pada penelitian ini dibatasi bahwa metode *naïve bayes* hanya digunakan untuk memilih keputusan serangan yang diambil NPC. Sehingga perlu dilakukan pengembangan lebih lanjut seperti memberikan keputusan perilaku NPC untuk bersembunyi, atau dapat menggunakan metode lain untuk menambah perilaku NPC tersebut.

DAFTAR PUSTAKA

- ABDI, M., HERUMURTI, D., & KUSWARDAYAN, I. (2017). Analisis Perbandingan Kecerdasan Buatan pada Computer Player dalam Mengambil Keputusan pada Game Battle RPG. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 15(2), 226-237. <https://doi.org/10.12962/j24068535.v15i2.a671>.
- GATA, W., BASRI, H., HIDAYAT, R., PATRAS, Y. E., BAHARUDDIN, B., FATMASARI, R., TOHARI, S., & WARDHANI, N. K. (2019). Algorithm Implementations Naïve Bayes, Random Forest. C4.5 on Online Gaming for Learning Achievement Predictions. *International Conference on Research of Educational Administration and Management (ICREAM)*, 258(Icream 2018), 1-9. <https://doi.org/10.2991/icream-18.2019.1>.
- ASMIATUN, S., & HENDRAWAN, A. (2016). Implementasi Klasifikasi Bayesian Untuk Strategi Menyerang Jarak Dekat Pada Npc (Non Player Character) Menggunakan Unity 3D. *Jurnal Transformatika*, 13(2), 42. <https://doi.org/10.26623/transformatika.v13i2.139>.
- ASMIATUN, S., HERMAWAN, L., & DARYATNI, T. (2013). Strategi Menyerang Jarak Dekat Menggunakan Klasifikasi Bayesian Pada NPC (Non Player Character). *Semantik*, 3(1), 351-357.
- F. OSISANWO., J.E.T, AKINSOLA., O, AWODELE., J. O, HIMNIKAIYE., O, O., & J, A. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3), 128-138. <https://doi.org/10.14445/22312803/ijett-v48p126>.
- FROSI, F. O., & CRISTINA, I. (2018). *Building Bots for Shooter Games based on the Bartle's Player Types and Finite State Machines: A Battling Behaviour Analysis*. 631-634.

- ISKANDAR, D., & SUPRAPTO, Y. K. (2015). Perbandingan Akurasi Klasifikasi Tingkat Kemiskinan Antara Algoritma C 4.5 dan Naive Bayes. *Jurnal Ilmiah NERO*, 2(1), 37–43.
- KAVIANI, P., & DHOTRE, S. (2017). Short Survey on Naive Bayes Algorithm. *International Journal of Advance Engineering and Research Development*, 4(11).
- LIAROKAPIS, F., DEBATTISTA, K., VOURVOPOULOS, A., & ENE, A. (2014). Comparing Interaction Techniques for Serious Games through Brain-Computer Interfaces : A User Perception Evaluation Study. *Entertainment Computing*. <https://doi.org/10.1016/j.entcom.2014.10.004>
- PUTERA, M. I. A., & MURTI, D. H. (2018). Peningkatan Kecerdasan Computer Player Pada Game Pertarungan Berbasis K-Nearest Neighbor Berbobot. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 16(1), 90. <https://doi.org/10.12962/j24068535.v16i1.a710>
- ROSTIANINGSIH, S., GREGORIUS, S. B., & WIJAYA, H. K. (2013). Game Simulasi Finite State Machine Untuk Pertanian Dan Peternakan. *Jurnal DKV Adiwarna*, 5, 2–7.
- SAFRIZAL, & SETIAWAN, D. R. (2019). Penerapan Algoritma Minimax Dengan Optimasi Memory Enhanced Test Driver With Value F Pada Game Catur. *Ilmiah Fakultas Teknik*, 1, 1–476. <https://doi.org/10.1017/CBO9781107415324.004>
- SALEH, A. (2015). Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. 2(3),207–217. <https://doi.org/10.24076/citec.2015v2i3.49>
- SANJAYA, S. W., AKBAR, M. A., & AFIRIANTO, T. (2019). Penerapan Naïve Bayes untuk NPC Braking Decision pada Racing Game. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIC) Universitas Brawijaya*, 3(4), 3252–3257.
- TALIB, F. Z., MAYADEWI, R. P., & ROESLY, E. (2016). Aplikasi Prediksi Peminat SMAN 8 Bandung Menggunakan Metode Klasifikasi Dengan Algoritma ID3. 2(3), 1053–1061.
- WANG, D., & TAN, A. H. (2015). Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(2), 123–138. <https://doi.org/10.1109/TCIAIG.2014.2336702>
- WICAKSONO, A., HARIADI, M., & N, S. MARDI. S. (2013). *Strategi Menyerang Npc Game Fps Menggunakan Fuzzy Finite State Machine*. 25–30.
- YUNANTA, D. P. (2017). *Game Rescue Idol Dengan Menggunakan Metode Finite State Machine (Fsm)*. 1(2), 96–103.