

Implementasi *Finite State Machine* dan Algoritma *Naïve Bayes* pada *Game Lord Of Sewandono*

Rian Dwi Susanto¹, Dodik Arwin Dermawan²

^{1,2}Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya

¹rian.17051204020@mhs.unesa.ac.id

²dodikdermawan@unesa.ac.id

Abstrak - Pemanfaatan teknologi untuk mengembangkan suatu kesenian budaya saat ini dirasa sangat tepat karena begitu pesatnya teknologi berkembang, salah satunya di bidang *game*. Di era modern seperti saat ini *game* bukan hanya digunakan sebagai media hiburan tetapi juga dapat dimanfaatkan sebagai media pengenalan suatu budaya. Reog Ponorogo merupakan kesenian budaya khas Jawa Timur yang berasal dari Ponorogo, kesenian ini biasa ditampilkan oleh sekelompok orang yang menari dengan memerankan beberapa tokoh. Klana Sewandono adalah salah satu tokoh yang terkenal dalam kesenian reog ponorogo karakternya digambarkan sebagai seorang raja memakai topeng bermahkota, berwajah merah dan membawa pecut yang dikenal dengan pecut Samandiman. Terinspirasi dari karakter tersebut terciptalah sebuah *game survival* berjudul “*Lord of Sewandono*”, *game* android dengan misi Prabu Sewandono mengalahkan Raja Singabarong untuk menyelamatkan Dwi Sanggalangit yang dibuat menggunakan gabungan antara metode *finite state machine* dan algoritma *naïve bayes*. Gabungan tersebut berfungsi untuk menentukan perilaku dari karakter NPC (*Non Player Character*) atau musuh utama yaitu Raja Singabarong yang terbagi menjadi 4 yaitu maju, mundur, serang, dan bertahan. Variable yang digunakan adalah AP (*Attack Power*), HP (*Health Point*), dan Jarak. Dari pengujian *naïve bayes* sebanyak 25 kali dengan *confusion matrix*, telah didapatkan hasil persentase tingkat akurasi pada *confusion matrix* sebesar 88% *valid* atau sesuai dengan harapan dan 12% *invalid* atau belum sesuai dengan harapan (*error*). Hasil pengujian beta rata-rata penilaian yang diperoleh sebesar 71,67% berdasarkan hasil dari responden terhadap tiap butir pertanyaan kuisioner.

Kata Kunci - reog ponorogo, klana sewandono, *game*, metode *finite state machine*, algoritma *naïve bayes*.

I. PENDAHULUAN

Perkembangan teknologi di Indonesia sangat meningkat. Hal ini dapat dilihat dari salah satu terciptanya teknologi canggih yaitu *game*. Tujuan dari terciptanya *game* adalah sebagai sarana hiburan. Menurut data Pokkt, *Decision Lab and Mobile Marketing Association* (MMA), jumlah *gamer* di Indonesia sudah mencapai 60 juta. Survei Lab Keputusan pada Agustus 2018 menunjukkan bahwa rata-rata pemain berusia 16-24 dan 25-34 sebesar 27%. Penduduk diatas 55 tahun yang juga bermain *game* terhitung sekitar 5% [1]. Menurut data lain yang didapat dari Statista ada 50,8 juta pengguna *game* di tahun 2020 [2].

Pada penelitian terdahulu dengan judul “Pembuatan *game* 2D berjudul si bujang ghanong sebagai sarana edukasi bagi masyarakat menggunakan *construct 2*” telah berhasil

menciptakan *game* yang bertemakan seni reog [3]. *Game* ini memiliki alur cerita seorang anak desa yang diperintah oleh pemimpin desa untuk mencari peralatan kesenian reog yang hilang. Untuk mendukung alur cerita di dalam *game* ditambahkan elemen-elemen kesenian reog. Pada penelitian ini, *game* yang dibuat dapat digunakan sebagai sarana edukasi masyarakat yang belum mengerti akan unsur-unsur Reog, namun pada penelitian ini belum terdapat kecerdasan buatan yang diterapkan, sehingga mengurangi pengalaman *gameplay*. Pengembangan *game* diperlukan kecerdasan buatan atau *Artificial Intelligence* (AI) untuk memberikan pengalaman *game* yang lebih menarik. *Finite State Machine* (FSM) adalah metode perancangan sistem kontrol yang menggunakan tiga hal untuk menggambarkan tingkah lakunya, yaitu *State* (keadaan), *Event* (kejadian) dan *Action* (aksi). Penerapan *Finite State Machine* pada *game* digunakan untuk mengatur respon berbagai NPC (*Non Player Character*) berdasarkan interaksi pemain, karena FSM dapat dirancang untuk merespon gerakan berdasarkan kondisi tertentu [4].

Algoritma *Naïve Bayes* adalah pengkategorian statistik berdasarkan teorema bayes. Teorema Bayes merupakan metode statistik dasar untuk pengenalan pola (*pattern recognition*), untuk menggunakan algoritma ini dalam pengkategorian, suatu permasalahan harus dapat divisualkan datanya [5]. Algoritma ini banyak digunakan dalam pengembangan penelitian *game*.

Meninjau dari penelitian terdahulu tentang implementasi *Artificial Intelligence* (AI) dalam *game*. Pada penelitian dengan judul “Simulasi Standar *Operational Procedure* Laboratorium Komputer Dengan Pemodelan *Finite State Machine*” telah mengaplikasikan FSM kedalam bentuk *role playing game* dengan keluaran simulasi SOP yang menarik dan mudah dipahami [6]. Penelitian lainya dengan judul “Rancang Bangun *Game* untuk Mengenalkan SOP Pendakian Gunung Semeru Menggunakan Pemodelan FSM” juga mengaplikasikan FSM kedalam bentuk *game* dengan keluaran berupa pengenalan aturan pendakian gunung lebih menarik dan mudah dipahami [7]. Kemudian penelitian dengan judul “Rancang Bangun *Game* Edukasi Media Pembelajaran Bahasa Inggris Menggunakan Pemodelan *Finite State Machine*”, *game* edukasi Bahasa Inggris yang dapat menarik minat anak-anak, hal terbukti dengan meningkatnya skor rata – rata sebesar 81,9, sedangkan skor rata-rata untuk edukasi bahasa inggris secara tradisional adalah 67.1 [8]. Penelitian terakhir lainnya tentang implementasi *finite state machine* dengan judul “*Application of the Finite State Machine Algorithm on 2D Platformer Rabbit*

Games vs Zombies” berhasil menerapkan metode *FSM* pada NPC, dimana NPC dapat bergerak mendekat atau menjauh menyesuaikan jarak dengan pemain [9].

Peninjauan mengenai implementasi algoritma *naïve bayes* dengan judul “Implementasi Klasifikasi Bayesian Untuk Strategi Menyerang Jarak Dekat Pada NPC (*Non Player Character*) Menggunakan Unity 3D” dimana dalam penelitian ini dilakukan pengujian dengan *confusion matrix* mendapatkan presentase 80% pada 10 kali percobaan pengujian [10]. Penelitian lainnya tentang implementasi algoritma *naïve bayes* pada dengan judul “*Determining NPC Behavior in Maze Chase Game using Naive Bayes Algorithm*” memberikan hasil sistem dengan tingkat kesalahan 0.5%, dimana pada penelitian ini penerapan *naïve bayes* digunakan untuk memprediksi posisi zona pemain dan mengarahkan NPC ke zona tempat pemain tersebut berada [11].

Dengan jumlah pengguna *game* yang tinggi maka *game* cocok digunakan sebagai sarana edukasi unsur – unsur kesenian reog. Peneliti mengembangkan perpaduan antara metode *finite state machine* dan *algoritma naïve bayes* untuk memberikan pengalaman lebih menarik pada *game* kesenian reog dengan judul *Lord Of Sewandono*. *Finite state machine* akan diterapkan pada pemodelan perilaku NPC. Metode ini cocok digunakan karena cukup sederhana namun sangat baik dan menarik dalam memodelkan perilaku. Algoritma *naïve bayes* akan diterapkan sebagai algoritma utama untuk pengambilan keputusan perilaku NPC. *Naïve bayes* dipilih karena memiliki tingkat kesalahan rendah serta sudah banyak berhasil diterapkan pada NPC *game*.

II. METODE PENELITIAN

Dalam pelaksanaan penelitian diperlukan sebuah diagram alur yang sebagai acuan untuk mendapat hasil sesuai dengan yang ingin dicapai. Berikut rancangan diagram alur penelitian terdapat pada Gbr. 1 dibawah ini.



Gbr. 1 Diagram Alur Metode Penelitian

Penelitian ini menggunakan dua metode gabungan yaitu metode *finite state machine* dan algoritma *naïve bayes*, dimana metode *finite state machine* digunakan sebagai algoritma pendukung pemodelan perilaku sedangkan algoritma *naïve bayes* digunakan sebagai algoritma utama untuk pengambilan keputusan pada perilaku NPC.

A. Perancangan Game

Game yang dibangun ini bergenre *survival*. Terdapat karakter utama bernama Prabu Klana Sewandana yang melakukan misi penyelamatan Dewi Sanggalangit dari penjara Raja Singabarong. Perilaku dari Raja Singabarong menjadi *object* utama dalam penelitian ini. Berikut karakter – karakter yang akan muncul pada *game* terdapat pada Tabel I.

TABEL I
KARAKTER GAME

No	Nama	Karakter	Keterangan
1	Prabu Klana Sewandono		Karakter Utama
2	Warok		Karakter Penunjuk
3	Bujang Ganong		Pasukan Prabu
4	Jathil		Pasukan Prabu
5	Dewi Sanggalangit		Karakter yang akan diselamatkan
6	Raja Singabarong		NPC Utama
7	Hewan		NPC Pendukung

B. Alur Game Lord Of Sewandono

Dalam sebuah *game* diperlukan alur cerita atau urutan dalam menyelesaikannya untuk memberikan pengalaman *game* yang menarik. Pada *game* "Lord Of Sewandono" ini urutan dari konsep sebagai berikut.

1. Pemain memiliki 3 nyawa dengan 100 *Health Point* pada tiap nyawa.
2. Permainan terdiri dari 3 *stage*.
3. Pemain berurutan masuk pada *stage* dan menyelesaikannya untuk bisa lanjut ke *stage* berikutnya.
4. Kondisi *stage* :
 - a. *Stage 1*, misi menemukan Jathil dan Bujang Ganong untuk membantu menyerang NPC serta mengumpulkan koin. Terdapat 2 musuh dengan wujud dan perilaku berbeda yaitu 2 ular dan 1 singa.
 - b. *Stage 2*, misi menemukan alat musik *gamelan* dan mengumpulkan koin untuk ditukar menuju *stage 3*. Terdapat 2 musuh dengan wujud dan perilaku berbeda yaitu 3 ular dan 3 singa.
 - c. *Stage 3*, misi mengalahkan Raja Singabarong untuk menyelamatkan Dewi Sanggalangit.
5. Jika nyawa pemain habis maka *Game Over* dan permainan dimulai kembali dari *stage 1*.

C. Perancangan Strategi NPC

Fokus dari penelitian ini adalah penerapan algoritma *naïve bayes*, penelitian ini akan menerapkan berbagai *action* diantaranya maju, mundur, serang, dan bertahan. Dimana parameter yang digunakan untuk pengambilan keputusan perilaku adalah AP (*Attack Point*), HP (*Health Point*), dan Jarak dari NPC. Pengelompokan parameter dapat dilihat pada Tabel II, Tabel III, dan Tabel IV.

TABEL II
KATEGORI *ATTACK POINT*

Input	Rentang Nilai	Klasifikasi
AP	AP >= 26	Kuat
	AP <= 25	Lemah

TABEL III
KATEGORI *HEALTH POINT*

Input	Rentang Nilai	Klasifikasi
HP	HP >= 51	Tinggi
	HP <= 50	Rendah

TABEL IV
KATEGORI *JARAK*

Input	Rentang Nilai	Klasifikasi
Jarak	Jarak > 7	Jauh
	Jarak >= 4 dan	Sedang

	jarak <= 7	
	Jarak < 4	Dekat

Tabel-tabel diatas (Tabel II, III, IV) merupakan parameter yang akan digunakan untuk pengklasifikasian perilaku NPC utama, yaitu *Attack Power (AP)*, *Health Point (HP)*, dan Jarak. Berikut merupakan *rule base* perilaku dari karakter NPC yang dapat dilihat pada Tabel V.

TABEL V
RULE BASE PERILAKU NPC

No.	Rule
1	Jika AP kuat dan HP tinggi dan jarak jauh maka perilaku maju
2	Jika AP kuat dan HP tinggi dan jarak sedang maka perilaku maju
3	Jika AP kuat dan HP tinggi dan jarak dekat maka perilaku serang
4	Jika AP lemah dan HP tinggi dan jarak jauh maka perilaku maju
5	Jika AP lemah dan HP tinggi dan jarak sedang maka perilaku bertahan
6	Jika AP lemah dan HP tinggi dan jarak dekat maka perilaku mundur
7	Jika AP kuat dan HP rendah dan jarak jauh maka perilaku maju
8	Jika AP kuat dan HP rendah dan jarak sedang maka perilaku bertahan
9	Jika AP kuat dan HP rendah dan jarak dekat maka perilaku serang
10	Jika AP lemah dan HP rendah dan jarak jauh maka perilaku maju
11	Jika AP lemah dan HP rendah dan jarak sedang maka perilaku bertahan
12	Jika AP lemah dan HP rendah dan jarak dekat maka perilaku mundur

D. Klasifikasi *Naïve Bayes* pada NPC

Setelah menentukan perancangan strategi, dilanjutkan menghitung klasifikasi *naïve bayes* untuk dapat menentukan perilaku yang sesuai dengan klasifikasi tersebut. Klasifikasi *naïve bayes* memiliki langkah – langkah untuk pengerjaannya. Berikut langkah – langkah pengerjaannya :

1. Menghitung probabilitas tiap label
2. Menghitung probabilitas kasus per label
3. Menghitung probabilitas label pada data *testing*

4. Membandingkan hasil probabilitas

Pada penelitian ini dataset yang akan dipakai merupakan dataset acak dari perilaku NPC utama. Berikut adalah dataset yang dipakai di dalam penelitian yang dibutuhkan untuk proses klasifikasi dapat dilihat pada tabel VI dibawah.

TABEL VI
DATASET PERILAKU NPC

No	AP	HP	Jarak	Perilaku
1	Lemah	Tinggi	Dekat	Mundur
2	Lemah	Tinggi	Sedang	Bertahan
3	Lemah	Rendah	Jauh	Maju
4	Kuat	Tinggi	Jauh	Maju
5	Lemah	Tinggi	Sedang	Bertahan
6	Lemah	Rendah	Sedang	Bertahan
7	Kuat	Rendah	Dekat	Serang
8	Lemah	Tinggi	Jauh	Maju
9	Lemah	Rendah	Dekat	Mundur
10	Lemah	Rendah	Sedang	Bertahan
11	Kuat	Rendah	Dekat	Serang
12	Lemah	Tinggi	Dekat	Mundur
13	Lemah	Tinggi	Jauh	Maju
14	Kuat	Tinggi	Sedang	Maju
15	Lemah	Tinggi	Dekat	Mundur
16	Kuat	Rendag	Dekat	Serang
17	Kuat	Rendah	Sedang	Bertahan
18	Kuat	Rendag	Dekat	Serang
19	Lemah	Rendah	Dekat	Mundur
20	Lemah	Tinggi	Dekat	Mundur

- Menghitung probabilitas tiap label
Pada label menjelaskan data yang terjadi sebanyak 12 data *training*. Dapat disimpulkan:

$$\begin{aligned}
 P(\text{Perilaku} | \text{Maju}) &= \frac{5}{20} \\
 P(\text{Perilaku} | \text{Mundur}) &= \frac{6}{20} \\
 P(\text{Perilaku} | \text{Serang}) &= \frac{4}{20} \\
 P(\text{Perilaku} | \text{Bertahan}) &= \frac{5}{20}
 \end{aligned}$$

- Menghitung probabilitas kasus per label

TABEL VII
LABEL ATTACK POINT

AP	P = Maju	P = Mundur	P = Serang	P = Bertahan
----	-------------	---------------	---------------	-----------------

Kuat	$\frac{2}{5}$	0	$\frac{4}{4}$	$\frac{1}{5}$
Lemah	$\frac{3}{5}$	$\frac{6}{6}$	0	$\frac{4}{5}$

TABEL VIII
LABEL HEALTH POINT

HP	P = Maju	P = Mundur	P = Serang	P = Bertahan
Tinggi	$\frac{4}{5}$	$\frac{4}{6}$	0	$\frac{2}{5}$
Rendah	$\frac{1}{5}$	$\frac{2}{6}$	$\frac{4}{4}$	$\frac{3}{5}$

TABEL IX
LABEL JARAK

Jarak	P = Maju	P = Mundur	P = Serang	P = Bertahan
Jauh	$\frac{4}{5}$	0	0	0
Sedang	$\frac{1}{5}$	0	0	$\frac{5}{5}$
Dekat	0	$\frac{6}{6}$	$\frac{4}{4}$	0

- Menghitung probabilitas label data *testing*
Jika nilai input AP = 35, HP = 55, jarak = 3.5, maka misal :

$$X' = (\text{AP} : \text{Kuat}, \text{HP} : \text{Tinggi}, \text{Jarak} : \text{Sedang})$$

$$\begin{aligned}
 P(\text{Maju}|X') &= P(\text{Kuat}|\text{Maju}) \cdot P(\text{Tinggi}|\text{Maju}) \cdot P(\text{Sedang}|\text{Maju}) \cdot P(\text{Perilaku}|\text{Maju}) \\
 &= \frac{3}{5} \cdot \frac{4}{5} \cdot \frac{1}{5} \cdot \frac{5}{20} \\
 &= 0.024
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Mundur}|X') &= P(\text{Kuat}|\text{Mundur}) \cdot P(\text{Tinggi}|\text{Mundur}) \cdot P(\text{Sedang}|\text{Mundur}) \cdot P(\text{Perilaku}|\text{Mundur}) \\
 &= 0 \cdot \frac{4}{6} \cdot 0 \cdot \frac{6}{20} \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Serang}|X') &= P(\text{Kuat}|\text{Serang}) \cdot P(\text{Tinggi}|\text{Serang}) \cdot P(\text{Sedang}|\text{Serang}) \cdot P(\text{Perilaku}|\text{Serang}) \\
 &= \frac{4}{4} \cdot 0 \cdot 0 \cdot \frac{4}{20} \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Bertahan}|X') &= P(\text{Kuat}|\text{Bertahan}) \cdot P(\text{Tinggi}|\text{Bertahan}) \cdot P(\text{Sedang}|\text{Bertahan}) \cdot P(\text{Perilaku}|\text{Bertahan}) \\
 &= \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{5}{5} \cdot \frac{5}{20} \\
 &= 0.020
 \end{aligned}$$

- Membandingkan hasil probabilitas

Langkah terakhir adalah membandingkan hasil dari probabilitas yang telah dihitung pada tahap 3. Berikut perbandingan hasil probabilitas dapat dilihat pada Tabel X.

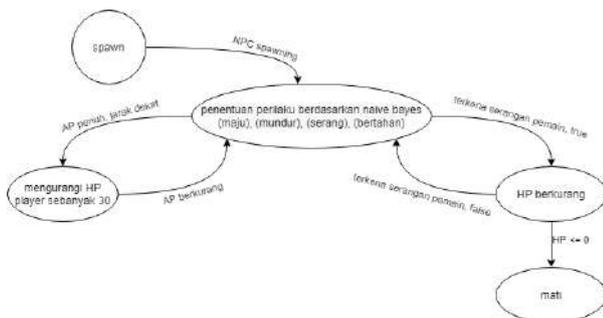
TABEL X
HASIL KLASIFIKASI *NAÏVE BAYES*

Perilaku	Hasil
Maju	0.024
Mundur	0
Serang	0
Bertahan	0.020

Perilaku NPC ditentukan dari nilai probabilitas perhitungan *naive bayes* yang tertinggi yaitu perilaku Maju. Dari X' dengan nilai input AP = 35, HP = 55, jarak = 3.5, maka perilaku NPC adalah Maju.

E. Pemodelan FSM pada NPC

Menggunakan metode *Finite State Machine* (FSM), penelitian ini menerapkan FSM untuk model perilaku NPC.



Gbr. 2 *Finite State Machine* Pada NPC

Penjelasan dari Gbr. 2 *state* pertama *spawn*, pada *state* ini NPC muncul pertama kali. Kemudian berpindah ke penentuan perilaku berdasarkan *naive bayes*. Ketika AP penuh dan jarak menempel pada pemain maka perilaku adalah memberikan serangan dan mengakibatkan berkurangnya HP pemain sebanyak 30 poin, setelah memberikan penyerangan maka AP berkurang dan kembali pada penentuan perilaku berdasarkan *naive bayes*. Saat pemain memberikan serangan (terkena serangan pemain, *true*) maka HP dari NPC akan berkurang, setelah HP kurang dari sama dengan 0 poin maka NPC mati. Jika pemain tidak memberi serangan (terkena serangan pemain, *false*) maka kembali pada penentuan perilaku berdasarkan *naive bayes*.

III. HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan akan dibahas mengenai implementasi dari perancangan *game* yang telah dijelaskan serta implementasi metode *finite state machine* dan algoritma *naive bayes* pada *game*.

A. Implementasi *Game*.

Pada tahap implementasi ini yaitu pembuatan *game* dari *asset* (karakter, alur, lainnya) yang telah disiapkan dengan menggunakan aplikasi Unity. Pada tahap ini juga dilakukan pengimplementasian dari FSM dan *Naive Bayes* pada perilaku NPC.

Berikut tampilan implementasi dari *game Lord Of Sewandono*.



Gbr. 3 Tampilan Awal *Game*

Pada Gbr. 3 terlihat tampilan awal dari *game Lord Of Sewandono*. *Game* bergenre *survival* bertema kebudayaan. Misi dari *game* ini adalah Prabu Klana Sewandono (karakter utama) menyelamatkan Dewi Sanggalangit dari Raja Singabarong (karakter NPC utama).



Gbr. 4 Tampilan Stage 1

Pada *stage 1* (Gbr. 4) seperti pada perancangan alur *game* yaitu misi menemukan Jathil dan Bujangganong untuk dapat membantu menyerang NPC. Dalam *stage* ini pemain akan bertemu NPC hewan yaitu ular dan singa.



Gbr. 5 Tampilan Stage 2

Stage 2 (Gbr. 5) memiliki misi mengumpulkan alat musik gamelan yang merupakan alat musik daerah yang mengiringi kesenian dari reog.



Gbr. 6 Tampilan Stage 3

Pada *stage 3* (Gbr. 6) terakhir ini terdapat implementasi *finite state machine* dan *naïve bayes* pada NPC utama (Raja Singabarong) yang merupakan fokus dari penelitian ini. Pemain akan mendapatkan pengalaman bermain yang menarik dengan adanya AI pada NPC tersebut.

B. Implementasi Perilaku NPC

Implementasi pada perilaku NPC ditunjukkan pada Tabel XI berikut :

TABEL XI
IMPLEMENTASI PERILAKU NPC

No	Gambar	Keterangan
1		Perilaku yang dihasilkan NPC adalah maju atau mundur
2		Perilaku yang dihasilkan NPC adalah bertahan (melompat)
3		Perilaku menyerang ketika AP kuat dan jarak sangat dekat dengan pemain

Animasi dan gerak yang bekerja pada perilaku NPC maju atau mundur adalah berjalan yang ditunjukkan pada Tabel XI nomor 1. Pada perilaku bertahan adalah melompat yang ditunjukkan pada Tabel XI nomor 2, dan terakhir animasi serta gerakan menyerang pada perilaku menyerang ke pemain pada Tabel XI nomor 3.

C. Pengujian Algoritma *Naïve Bayes*

Pengujian ini dilakukan untuk mengetahui kemungkinan terjadinya peluang dan kemiripan kondisi penentuan pada dataset dan atribut. Penerimaan kondisi dikatakan berhasil jika ditemukan nilai tertinggi dari hasil probabilitas yang diuji. Berikut pengujian algoritma *naïve bayes* dengan menggunakan 25 data *training* atau data uji yang dapat dilihat pada Tabel XII berikut.

TABEL XII
PENGUJIAN ALGORITMA *NAÏVE BAYES*

No Uji	Perilaku diharapkan	Perilaku hasil	Value
1	Bertahan	Bertahan	Valid
2	Serang	Serang	Valid
3	Mundur	Mundur	Valid
4	Maju	Maju	Valid
5	Serang	Maju	Invalid
6	Maju	Maju	Valid
7	Maju	Bertahan	Invalid
8	Maju	Maju	Valid
9	Bertahan	Bertahan	Valid
10	Bertahan	Bertahan	Valid
11	Bertahan	Bertahan	Valid
12	Bertahan	Bertahan	Valid
13	Bertahan	Bertahan	Valid
14	Mundur	Mundur	Valid
15	Bertahan	Bertahan	Valid
16	Maju	Maju	Valid
17	Maju	Maju	Valid
18	Bertahan	Bertahan	Valid
19	Maju	Maju	Valid
20	Mundur	Mundur	Valid
21	Bertahan	Bertahan	Valid
22	Mundur	Mundur	Valid
23	Mundur	Mundur	Valid
24	Serang	Serang	Valid
25	Serang	Serang	Valid

Dari hasil pengujian *naïve bayes* Tabel XII, terdapat 2 nilai *invalid* atau tidak sesuai dengan data yang diharapkan. Dari pengujian tersebut dapat diambil kesimpulan menggunakan *confusion matrix* dengan *class A* untuk Maju, *class B* untuk Mundur, *class C* untuk Serang, dan *class D* untuk bertahan yang dapat dilihat pada Tabel 13 berikut :

TABEL XIII
CONFUSION MATRIX

Predicted Class			
Class A : Maju, Class B : Mundur, Class C : Serang, Class D : Bertahan			
Actual Class	Class A	6	1
	Class B	5	0
	Class C	3	1
	Class D	9	0

Pada Tabel XIII diatas, *confusion matrix* didapatkan 6 pada *class A* diperkirakan tepat dan sebanyak 1 diperkirakan tidak tepat, kemudian 5 *class B* untuk perkiraan tepat dan 0 perkiraan tidak tepat, 3 *class C* perkiraan tepat dan 1 perkiraan tidak tepat, terakhir 9 *class D* perkiraan tepat dan 0 perkiraan tidak tepat. Selanjutnya menghitung nilai akurasi. Untuk perhitungan nilai akurasi dapat dilakukan berdasarkan hasil yang telah didapat.

$$\text{Akurasi} = \frac{6+5+3+3+9}{6+1+5+0+3+1+9+0} \times 100\%$$

$$= 88\%$$

Dari perhitungan akurasi diatas dapat disimpulkan bahwa tingkat presisi presentase yang dihitung dengan *confusion matrix* adalah 88% akurat atau stabil.

D. Pengujian Beta

Penelitian ini menggunakan pengujian Beta untuk mengetahui tingkat performa aplikasi dari hasil penilaian pengguna. Pengujian beta dilakukan dengan melakukan survei dengan menyebarkan angket atau kuesioner ke beberapa responden yang akan menilai suatu aplikasi dengan memberikan penilaian di setiap butir pertanyaan dengan menggunakan skala *likert*. [12] Sampel yang diambil adalah 15 orang, nantinya responden dipersilahkan mencoba *game* dan mengisi kuisisioner yang telah disediakan.

Pengujian menggunakan 4 kategori jawaban dengan bobot yang berbeda untuk setiap jawabannya seperti pada tabel XIV berikut.

TABEL XIV
Bobot Jawaban Pertanyaan

Jawaban	Sangat Setuju	Setuju	Kurang Setuju	Tidak Setuju
Bobot	4	3	2	1

Untuk mencari nilai presentasi dari masing-masing pendapat kuisisioner digunakan rumus skala *likert* sebagai berikut :

$$P = S/Q \times 100\%$$

Keterangan :

P : Nilai presentasi yang dicari

S : Jumlah frekuensi dikalikan dengan skor yang ditetapkan tiap jawaban

Q : Skor tertinggi dikalikan dengan jumlah sampel

Kuisisioner ini terdiri dari 5 pertanyaan, dan berikut pertanyaan beserta hasil dengan pengujian beta.

1. Apakah *game* LOS ini berjalan dengan baik ?

TABEL XV
Hasil Pengujian Pertanyaan 1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	4	2	8
	Setuju	3	6	18
	Kurang Setuju	2	7	14

	Tidak Setuju	1	0	0
Jumlah			15	40

$$P = (40 / 60) \times 100\% = 66,67\%$$

2. Apakah tampilan *game* LOS secara umum sudah menarik ?

TABEL XVI
Hasil Pengujian Pertanyaan 2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	4	1	4
	Setuju	3	10	30
	Kurang Setuju	2	4	8
	Tidak Setuju	1	0	0
Jumlah			15	42

$$P = (42 / 60) \times 100\% = 70\%$$

3. Apakah karakter dalam *game* LOS mudah dikenali ?

TABEL XVI
Hasil Pengujian Pertanyaan 3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	4	2	8
	Setuju	3	8	24
	Kurang Setuju	2	5	10
	Tidak Setuju	1	0	0
Jumlah			15	42

$$P = (42 / 60) \times 100\% = 70\%$$

4. *Game* LOS ini memberikan pengalaman *game* yang menarik ?

TABEL XVI
Hasil Pengujian Pertanyaan 4

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	4	2	8
	Setuju	3	11	33
	Kurang Setuju	2	2	4
	Tidak Setuju	1	0	0
Jumlah			15	45

$$P = (45 / 60) \times 100\% = 75\%$$

5. Apakah dengan *game* LOS ini dapat memberikan edukasi tentang unsur kesenian reog ?

TABEL XVI
Hasil Pengujian Pertanyaan 5

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	4	3	12
	Setuju	3	7	21
	Kurang Setuju	2	5	10
	Tidak Setuju	1	0	0
Jumlah			15	43

$$P = (43 / 60) \times 100\% = 71,67\%$$

Dari hasil pengujian beta menunjukkan bahwa *game Lord Of Sewandono* memperoleh presentase rata-rata sebesar 70,66%. Nilai presentase terbesar 75% untuk penilaian pertanyaan ke-4 mengenai pengalaman dalam memainkan

game *Lord Of Sewandono* menarik. Dimana hal ini sesuai pada fokus penelitian ini yaitu menerapkan metode *Finite State Machine* dan Algoritma *Naïve Bayes* untuk memberikan pengalaman bermain lebih menarik.

IV. KESIMPULAN

Hasil penerapan penggabungan metode *finite state machine* dan algoritma *naïve bayes* ini membuktikan bahwa metode *finite state machine* dan algoritma *naïve bayes* menjadikan pergerakan dari NPC lebih *real* dan *gameplay* menjadi lebih *attractive*.

Berdasarkan hasil pengujian dari *game* yang telah dibuat di dapatkan hasil yaitu pengujian perilaku NPC utama berfungsi dengan baik. Perilaku Maju dan Mundur adalah animasi dan gerakan berjalan, perilaku Menyerang dengan animasi dan gerakan menyerang, terakhir perilaku Bertahan yaitu animasi dan gerakan melompat.

Dengan menggunakan 25 data uji scenario untuk menguji perilaku menggunakan algoritma *naïve bayes* pada *game*. Dari pengujian algoritma *naïve bayes* tersebut, tahapan uji yang digunakan adalah *confusion matrix* dengan persentase akurasi *confusion matrix* didapatkan hasil 88% nilai valid, dan 12% tingkat akurasi tidak valid.

Berdasarkan hasil pengujian beta rata-rata penilaian yang diperoleh sebesar 71,67% berdasarkan hasil dari 15 responden terhadap tiap butir pertanyaan kuisioner.

V. SARAN

Penelitian pada perancangan dan implementasi program, *game* yang telah dibuat masih terdapat banyak kekurangan, sehingga dibutuhkan perancangan dan implementasi kembali, diantaranya :

1. Penambahan *stage* atau level permainan agar pemain lebih tertantang dalam menyelesaikan *game* ini.
2. Pembentukan alur *game* agar unsur – unsur kesenian reog dapat dipahami dengan baik.
3. Meningkatkan tampilan desain.

UCAPAN TERIMAKASIH

Ujaran syukur dan terimakasih penulis sampaikan kepada :

1. Allah SWT atas segala rahmat dan pertolongannya, sehingga penulis dapat menyelesaikan penelitian ini.
2. Orang tua yang senantiasa memberikan dukungan dan semangat
3. Dosen pembimbing yang dengan sabar membimbing penelitian ini dari awal hingga akhir
4. Teman-teman dan sahabat yang selalu memberikan energi positif.

DAFTAR PUSTAKA

- [1] Lokadata.beritagar.id, "Pemain *game* online menurut usia, 2018," *Lokadata.Beritagar.Id*, 2018.
<https://lokadata.beritagar.id/chart/preview/pemain-game-online-menurut-usia-2018-1579509362>.
- [2] D. M. Dahwilani, "Survei: 16,5 Persen Masyarakat Habiskan Waktu Main *Game* Online selama Pandemi Covid-19," *Inews.Id*, 2020.
[https://www.inews.id/techno/internet/survei-165-persen-masyarakat-habiskan-waktu-main-game-online-selama-pandemi-](https://www.inews.id/techno/internet/survei-165-persen-masyarakat-habiskan-waktu-main-game-online-selama-pandemi-covid-19)

- [3] W. Kurniadani, A. Fajaryanto, T. Informatika, F. Teknik, and U. M. Ponorogo, "Pembuatan *Game* 2D Berjudul 'Si Bujang Ghanong' Sebagai Sarana Edukasi Bagi Masyarakat Menggunakan Construct 2."
- [4] M. F. Rahadian, A. Suyatno, and S. Maharani, "Penerapan Metode Finite State Machine Pada *Game* ' The Relationship,'" vol. 11, no. 1, pp. 14–22, 2016.
- [5] M. H. Rifqo, A. Wijaya, and J. Pseudocode, "Implementasi Algoritma Naive Bayes Dalam Penentuan Pemberian Kredit," vol. IV, no. September, pp. 120–128, 2017.
- [6] D. A. Dermawan, D. F. Suyatno, and M. S. Hawari, "Simulasi Standard Operational Procedure Laboratorium Komputer Dengan Pemodelan Finite State Machine Pada Perilaku Teknisi," vol. XIV, 2019.
- [7] S. B. Wicaksana, R. S. Nugroho, S. D. Prasetya, and D. A. Dermawan, "Rancang Bangun *Game* untuk Mengenalkan SOP Pendidikan Gunung Semeru Menggunakan Pemodelan FSM," *J. Inf. Eng. Educ. Technol.*, vol. 03, 2019.
- [8] A. P. Yani, M. Y. Charnelia, S. D. R. A. Cahyono, and D. A. Dermawan, "Rancang Bangun *Game* Edukasi Media Pembelajaran Bahasa Inggris Menggunakan Pemodelan Finite State Machine," vol. 5, no. 2, pp. 37–43, 2019.
- [9] A. Solihin, E. W. Hidayat, and A. P. Aldya, "Application of the Finite State Machine Algorithm on 2D Platformer Rabbit *Games* vs Zombies," vol. 4, no. 1, pp. 33–38, 2019, doi: 10.15575/join.v4i1.293.
- [10] S. Asmiatun, M. Kom, and F. Teknologi, "Implementasi Klasifikasi Bayesian Untuk Strategi Menyerang Jarak Dekat Pada NPC (Non Player Character) Menggunakan Unity 3D," vol. 13, pp. 42–48, 2016.
- [11] H. Zohro and M. Chase, "Determining NPC Behavior in Maze Chase *Game* using Naïve Bayes Algorithm," *2019 Int. Semin. Res. Inf. Technol. Intell. Syst.*, pp. 91–96, 2019.
- [12] N. Kadek, D. Rusjyanthi, T. Informasi, U. Udayana, and P. Korespondensi, "Usability Testing Pada Simulator Media Pembelajaran Lalu Usability Testing on Android-Based Traffic Educational Media," vol. 8, no. 2, 2021, doi: 10.25126/jtiik.202184271.