

Meningkatkan Behaviour Npc Pada Game 3d Survival Menggunakan Metode Markov

Derry Gunawan¹, Atthariq², Aswandi³

^{1,3}Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.ACEH Medan Km.280 Buketrata 24301 Indonesia
Email: derigunawan28@gmail.com

Abstrak--- *Game Survival Shooter* adalah *game* yang pemainnya mengalahkan semua musuh yang menghadangnya untuk bertahan hidup agar mencapai titik puncak keberhasilan dari suatu misi. Di dalam permainan untuk mengumpulkan *score* yang dicapai berdasarkan misi yang ada dengan mengalahkan semua musuh yang menghampirinya, tetapi musuh pun ikut menyerang sampai darah pemain habis. Penelitian ini bertujuan menentukan hasil perhitungan aksi *NPC* yang sering terjadi. Metode yang digunakan untuk menentukan perilaku *NPC* adalah Markov Chain yang dimasukkan di *Finite State Machine*. Uji coba implementasi *Markov chain* ke *Finite State Machine* pada perilaku *NPC* sesuai dengan desain output perilaku yang didesain sebelumnya, yaitu patrol, mengejar dan menyerang *player*. Hasil Pengujian dari 10 data kondisi *player* yang berbeda perilaku *NPC* adalah *NPC* Patrol 60%, *NPC* Mengejar 20% dan *NPC* 20% Menyerang.

Kata kunci: *Game*, Perilaku *NPC* (*Non Player Character*), *AI*, *Algoritma Markov Chain*, *Finite State Machine*

Abstract--- *Game Survival Shooter* is a game that players defeat all the enemies that block them to survive in order to reach the peak of the success of a mission. In the game to collect the score achieved based on the existing mission by defeating all the enemies who approached it, but the enemy was attacked until the blood of the player runs out. This study aims to determine the results of *NPC* action calculations that often occur. The method used to determine the *NPC* behavior is the Markov Chain included in the *Finite State Machine*. Testing of the implementation of Markov chain to *Finite State Machine* on *NPC* behavior in accordance with the design output design of the previously designed, that is patrol, chase and attack the *player*. Test results from 10 different *player* condition data of *NPC* behavior are *NPC* Patrol 60%, *NPC* Pursuing 20% and *NPC* 20% Attacking.

Keywords: *Game*, Behaviour *NPC* (*Non Player Character*), *AI*, *Algoritma Markov Chain*, *Finite State Machine*

I. PENDAHULUAN

Perkembangan *game* ini saat ini sudah sangat pesat. Setiap hari bermunculan banyak *game* baru baik itu merupakan *game* Android, PC, maupun *Game Console*. Perkembangan *game* didukung oleh para user yang tidak hanya dari golongan anak – anak dan remaja, bahkan sampai orang dewasa juga memainkannya. Seiring perjalanan waktu sudah banyak macam – macam genre *game* yang bermunculan seperti *RPG* (*Role Playing Game*), *FPS* (*First Person Shooter*), *TPS* (*Third Person Shooter*), dan lain sebagainya.

Game merupakan arena keputusan dari aksi pemainnya dan terdapat target tertentu yang ingin dicapai oleh pemainnya. *Game* berdasarkan dimensinya terbagi menjadi dua, yaitu dua dimensi (2D) dan tiga dimensi (3D). Saat ini *Game* banyak menarik perhatian dan digemari oleh pemain *game* tiga dimensi (3D) dengan genre *Third Person Shooter* (*TPS*) [1].

NPC (*Non-Playable Character*) merupakan semua obyek atau karakter dalam *game* yang tidak dimainkan oleh

player, merupakan obyek bergerak atau karakter pada dunia *game* yang dijalankan oleh komputer dan bisa berinteraksi dengan pemain. Dalam *game TPS genre Shooter game*, salah satu respon umum musuh (*NPC*) adalah memberikan direct feedback pada pemain jika diserang [2]. Seringkali terjadi kasus ketika *NPC* sudah mengejar dan kehilangan jejak *player*, *NPC* tidak kembali ke posisi awal dan patrol. Hal ini akan menyebabkan tingkah laku *NPC* tidak bersifat adaptif *NPC* salah satu komponen *game* sebagai pengganggu untuk mempersulit permainan, sehingga *game* akan terasa lebih menantang [3].

Dalam hal ini dimasukkan konsep *Markov Chain* yang berfungsi menentukan kemungkinan perilaku *NPC* yang terjadi dengan melakukan perhitungan aksi *NPC* ditargetkan kepada karakter atau *Player* saat memasuki area *NPC* [4]. *NPC* akan dipasang di beberapa titik dan memiliki area masing-masing. Jika karakter *player* memasuki area *NPC*, secara otomatis *NPC* akan aktif dan melakukan aksi yang telah dipasangkan sebelumnya dan sesuai dengan keadaan *player* saat itu [5].

1.1 Third Person Shooters

Third Person Shooter (TPS) yaitu jenis *game* yang mengutamakan kecepatan gerakan dalam permainan. Dalam jenis ini akan lebih banyak baku tembak serta mengharuskan untuk bertahan hidup selama mungkin hingga permainan berakhir. dan jenis permainan dengan karakter pemain yang menggunakan sudut pandang orang ketiga yang dapat melihat keseluruhan karakter pemain dari belakang (punggung).

1.2 Finite State Machine

FSM (*Finite State Machine*) adalah sebuah metodologi perancangan istem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *state* (keadaan), *event* (kejadian) dan *action* (aksi). *Artificial Intelligence for Games* menyebutkan bahwa *Finite State Machines* (FSM) masuk dalam ranah *Decision Making* (pembuat keputusan) pada *Artificial Intelligence* (AI) [6] [7]. Teknik ini teknik yang paling banyak dipergunakan untuk permasalahan “*decision making*” dan, sekaligus dengan *scriptingnya*, dipergunakan secara luas untuk merancang sistem *decision making* dalam *game*. *State machine* dikenal secara luas sebagai teknik untuk pemodelan fenomena atau kondisi berbasis *event*, termasuk penguraiannya, serta desain *interface*. *Finite State Machine* (FSM) atau juga disebut sebagai *Finite State Automata*, dianggap sebagai teknik yang secara luas dipergunakan dalam merancang *AI* dalam *game* [8].

1.3 Markov Chain

Rantai *Markov* (*markov chain*) suatu metode yang mempelajari sifat sifat suatu variabel pada masa sekarang yang didasarkan pada sifat- sifatnya di masa lalu dalam usaha menaksir sifat- sifat variabel tersebut di masa yang akan datang [9]. Proses *Markov* adalah proses stokastik masa lalu tidak mempunyai pengaruh pada masa yang akan datang bila masa sekarang diketahui. Dalam analisis *Markov* yang dihasilkan adalah suatu informasi probabilistik yang dapat digunakan untuk membantu pembuatan keputusan. Jadi, analisis ini bukan suatu teknik optimisasi melainkan suatu teknik deskriptif. Analisis *Markov* merupakan suatu bentuk khusus dari model probabilistik yang lebih umum dikenal sebagai proses stokastik (*Stochastic process*) [10].

II. METODOLOGI PENELITIAN

2.1 Analisis Metode Markov Chain

Pada Penelitian ini sebagai metode yang akan digunakan yang memiliki 3 tahapan dari analisis data kemudian dilanjutkan dengan perancangan sistem.

1. Variabel Penentu Perilaku

Dengan menggunakan Algoritma *Markov Chain* maka eksekusi setiap *state* akan dipengaruhi oleh 3

variabel input yaitu Jarak(Jr), Kesehatan(Ks), Kecepatan(Kc) untuk *Player*. Dari Ketiga input yang akan memutuskan perilaku Variabel *Output NPC* terhadap *Player* berupa patrol, kejar, dan serang. *Markov Chain* di implementasikan ke dalam *NPC* bertujuan untuk menentukan perilakunya terhadap *Player*. Maka Perilaku *NPC* tergantung semua pada kondisi *Player*. jika *Player* berada pada jarak yang jauh dari *NPC* maka *NPC* akan melakukan patrol jika tidak *NPC* akan mengejar atau menyerangnya.

2. Rule Markov Chain Untuk NPC

Dalam rule markov chain perilaku *NPC* menggunakan batas minimal dan batas maksimal nilai parameter :

- Perilaku *NPC* Patrol dengan range 50 – 100
- Perilaku *NPC* Kejar dengan range 30 – 50
- Perilaku *NPC* Serang dengan range 0 – 30

3. Proses Markov Chain Perilaku NPC

Operasi *Markov Chain* dalam pada penelitian ini adalah menggunakan matriks probabilitas transisi yang akan menghitung kemungkinan yang terjadi di masa mendatang ketika *Player* dalam kondisi saat ini. Sehingga dalam menghitung kemungkinan aksi *NPC* sebagai berikut :

$$[Jr Ks Kc] = [Jr Ks Kc] \times \text{Matriks Probabilitas Transisi}$$

Keterangan :

Jr = Jarak

Ks = Kesehatan

Kc = Kecepatan

2.2 Perancangan Sistem

Perancangan sistem ini merupakan tahapan dalam pembuatan *game*. Dalam pembuatan *game* membutuhkan adanya skenario dari permainan yang dilakukan hingga akhir dari *game*. Pada penelitian ini yang paling utama adalah *NPC* sebagai pengganggu terhadap *Player* sehingga di dalam permainan memiliki tingkat kesulitan. Di dalam *NPC* diterapkan suatu kemungkinan perilaku saat *NPC* melakukan aksi seperti patrol, Kejar, dan Serang. Tahapan perancangan tersebut berupa perancangan blok diagram perilaku *NPC* secara umum, perancangan tabel dan proses metode markov pada *NPC*.

2.3 Blok Diagram Perilaku NPC

Berikut ini adalah blok diagram perilaku *NPC* pada tahap – tahap terjadinya proses kemunculan monster di beberapa koordinat sesuai periode waktu yang telah ditetapkan yang bertindak sebagai *NPC*. Didalam *NPC* ini memiliki 3 *State Aksi* yang akan dilakukan seperti keadaan bersiaga, mengejar, dan menyerang. Dapat dilihat di gambar 1.

atas parameter yang menjelaskan bahwa pada jarak, kesehatan berapa NPC akan melakukan aksi patrol, kejar dan serang. Karena syarat nilai untuk perhitungan rantai markov adalah harus berjumlah satu.

Tabel 2 Matriks Probabilitas Transisi Parameter

Variabel Input	Variabel Output		
	Patrol	Kejar	Serang
Jr	65/100 = 0.65	30/100 = 0.3	10/100 = 0.1
Ks	50/100 = 0.5	30/100 = 0.3	20/100 = 0.2
Kc	54/100 = 0.54	25/100 = 0.25	21/100 = 0.21

Value atau nilai dari tabel tersebut merupakan hasil keseluruhan matriks probabilitas transisi sehingga dapat dituliskan sebagai berikut:

$$\text{Matriks Probabilitas Transisi} = \begin{bmatrix} 0.65 & 0.3 & 0.1 \\ 0.5 & 0.3 & 0.2 \\ 0.54 & 0.25 & 0.21 \end{bmatrix}$$

Perhitungan tersebut untuk menentukan kemungkinan perilaku NPC dengan kondisi player saat ini. Perhitungan tersebut akan dikalikan dengan jumlah keseluruhan dari nilai kondisi awal player yaitu 100. Dari hasil kondisi Player yang akan memutuskan aksi NPC saat Player memasuki jangkauan berdasarkan parameter yang sudah ditetapkan.

2.5 Perancangan Proses Metode Markov

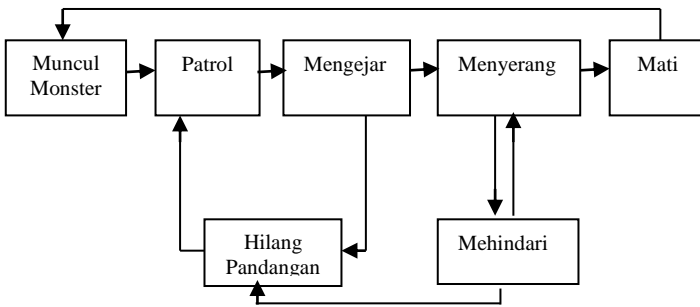
Tahap perhitungan perilaku NPC yang terjadi suatu proses dimana NPC akan mencari Player dengan memiliki 3 variabel inputan Kesehatan, Jarak dan Kecepatan yang dimiliki oleh Player untuk menghitung probabilitas Player. Ketika Sudah didapatkan terjadi proses pencocokan dengan rules nilai parameter yang telah ditentukan tetapi ketika kondisi status player Berubah maka kembali menghitung nilai probabilitas lagi, Jika Status player tidak berubah Maka Musuh akan mengambil Sebuah keputusan untuk melakukan aksi. Flowchart metode markov perilaku NPC dapat dilihat di gambar 2

III. HASIL DAN PEMBAHASAN

Berikut adalah tujuan atau hasil akhir gambar yang digunakan dalam tampilan permainan game 3 dimensi "Survival Shooter Third Person Shooter".

3.1 Implementasi Game

Implementasi merupakan suatu tahapan dalam perancangan. Tahapan implemntasi dilakukan setelah proses analisa dan perancangan selesai dilakukan.



Gambar 1 Block Diagram Perilaku NPC

2.4 Perancangan Tabel

Rancangan tabel ini tahapan untuk menentukan aksi dari perilaku NPC dengan kondisi player saat ini. Perancangan tersebut berupa nilai awal dari kondisi player yang berbeda, nilai variabel parameter NPC, dan matriks probabilitas transisi yang akan menghitung kemungkinan yang terjadi di masa mendatang.

1. Tabel Nilai Awal kondisi Player

Pada tabel ini berupa kondisi pada player setiap waktu berubah sesuai dengan perjalanan permainan, sehingga perilaku NPC terhadap player juga akan berubah.

Tabel 1 Nilai Awal Kondisi Player

No	Variabel Input			Jumlah
	Jr	Ks	Kc	
1	20	60	20	100
2	60	40	20	120
3	50	40	30	130
4	40	30	40	110
5	80	50	10	140
6	40	30	40	90
7	40	30	10	80
8	20	20	20	60
9	10	20	10	40
10	10	10	10	30

2. Tabel Variabel Parameter

Pada Tabel ini merupakan nilai acak dari suatu variabel yang sudah ditentukan berdasarkan parameter perilaku NPC. Jumlah dari ketiga baris adalah 100 tetapi jumlah kolom tidak. Informasi ini digunakan untuk membuat matrik probabilitas transisi. Nilai dari tabel tersebut di

Dalam hasil pengujian ini adalah proses game kemungkinan perilaku NPC yang terjadi berdasarkan aksi yang dilakukan pada *Player*. Analisa data yang diperoleh berdasarkan hasil perhitungan pada kondisi NPC dapat dilihat pada tabel – tabel sebagai berikut.

1. Perubahan perilaku NPC dalam keadaan patrol dikarenakan kondisi kesehatan *player* masih penuh dan jarak *player* terhadap NPC masih jauh. Dapat dilihat di console game pada tabel 3

Tabel 3 Tampilan Console NPC Patrol

No	Console Game Aksi Terhadap NPC			Ket
	Jarak	Kecepatan	Kesehatan	
1	53.8	29	10.2	Patrol
2	63.5	37.8	9.18	Patrol
3	69.03	44.98	14.49	Patrol
4	50.9	38.5	18.7	Patrol
5	73.94	47.98	17.88	Patrol
6	58.41	29.60	20.10	Patrol

2. Perubahan perilaku NPC dalam keadaan mengejar dikarenakan jarak *player* terhadap NPC sangat dekat dan NPC dapat menjangkaunya. Selain itu kesehatan *player* juga masih dalam keadaan 100 secara keseluruhan dapat dilihat di tabel 4.

Tabel 4 Tampilan Console NPC Kejar

No	Console Game Aksi Terhadap NPC			Ket
	Jarak	Kecepatan	Kesehatan	
1	53.8	29	10.2	Patrol
2	42	19.4	14.8	Kejar
3	69.03	44.98	14.49	Patrol
4	50.9	38.5	18.7	Patrol
5	33.36	18.3	5.88	Kejar
6	58.41	29.60	20.10	Patrol

3. Perubahan perilaku NPC dalam keadaan mengejar sekaligus menyerang *Player player* dikarenakan *player*

mendekati NPC. Selain itu kesehatan *player* mulai berkurang. Berikut tampilan tersebut dapat dilihat pada tabel 5.

Tabel 5 Tampilan Console NPC Serang

No	Console Game Aksi Terhadap NPC			Ket
	Jarak	Kecepatan	Kesehatan	
1	53.8	29	10.2	Patrol
2	42	19.4	14.8	Kejar
3	69.03	44.98	14.49	Serang
4	50.9	38.5	18.7	Patrol
5	33.36	18.3	5.88	Serang
6	58.41	29.60	20.10	Kejar
7	71.41	55.0	30.9	Kejar
8	48.41	29.60	50.13	Patrol

Dari hasil 10 kali percobaan yang dilakukan untuk menentukan kemungkinan aksi yang terjadi pada *Player* maka didapat tingkat persentase yang sering terjadi adalah NPC dalam keadaan patrol sebanyak 6 kali kemungkinan yang terjadi, dan NPC dalam keadaan mengejar dan menyerang didapatkan 2 kali aksi yang dilakukan NPC. Sehingga Aksi perilaku yang sering terjadi adalah 60%. Secara detail dapat dilihat pada tabel 6.

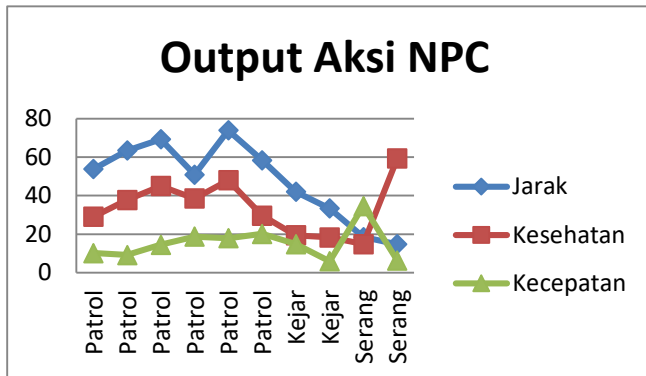
Tabel 6 Output Aksi NPC Terhadap Player

No	Output Aksi NPC			
	Jarak	Kesehatan	Kecepatan	Ket
1	53.8	29	10.2	Patrol
2	63.5	37.8	9.18	Patrol
3	69.03	44.98	14.49	Patrol
4	50.9	38.5	18.7	Patrol
5	73.94	47.98	17.88	Patrol
6	58.41	29.60	20.10	Patrol
7	42	19.4	14.8	Kejar
8	33.36	18.3	5.88	Kejar
9	18.5	14.8	34.6	Serang
10	14.7	59.3	6.34	Serang

Dari tabel 6 dapat dilihat bahwa semua output sudah sesuai dengan rule yang telah ditentukan. Perilaku yang dihasilkan dari output tersebut adalah patrol = 60%,

Mengejar = 20% dan Menyerang = 20% dari data 10 yang ada.

4. Berikut ini adalah hasil grafik dari 10 percobaan dari perhitungan markov berdasarkan aksi NPC yang berbeda. Maka hasil tersebut dilihat pada gambar 3.



Gambar 3 Output Aksi NPC

IV KESIMPULAN

4.1 Kesimpulan

Berdasarkan game yang telah dibuat dan uji coba yang dilakukan, maka dapat disimpulkan bahwa hasil perhitungan output aksi NPC yang terjadi dengan mengalikan hasil dari matriks probabilitas transisi, dikalikan dengan jumlah nilai keseluruhan kondisi Player. Sehingga hasil aksi NPC didapatkan dari kondisi *Player* seperti Jarak, Kesehatan, dan Kecepatannya. Hasil uji coba dari 10 data kondisi player yang berbeda dapat dipersentasekan adalah *NPC* Patrol 60%, lebih banyak kemungkinan yang terjadi aksi *NPC* terhadap *Player*.

4.2 Saran

Dalam membangun sebuah *game* harus lebih memiliki tantangan yang kompleks dan menantang agar melatih kelincahan dan akurasi saat memainkan permainan. Pada bagian *Artifial Intelligence* pada *NPC* masih banyak perilaku berdasarkan aksi yang dilakukan dalam kehidupan sehari-hari untuk pembuatan *game*, dengan penambahan Algoritma yang lebih bagus *game* dapat berjalan dengan lebih baik

REFERENSI

- [1] Costikyan, G (2013). *Uncertainty in Games*. MIT Press,20
- [2] Craig, W. Reynolds. *Steering Behaviors For Autonomous Characters*. Sony Computer Entertainment America.1999
- [3] Miftachul , Kurniawan dan Nugroho. 2011. “*Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy*”. National Seminar on Electrical, Informatics, and Its Education. Online : http://elektro.um.ac.id/ceie/2011/assets/paper/National/Informatics/A218%20%20YUNIFA_MISTACHUL_ARIF%20%20DESAIN%20PERUBAHAN%20PERILAKU.pdf.
- [4] Mardi, Miftahul, Hariadi, dan Mauridhi. 2011. *Perilaku Taktis Untuk Non-Player Characters Di Game Peperangan Meniru Strategi Manusia Menggunakan Fuzzy Logic Dan Hierarchical Finite State Machine*. Onlinehttp://kursor.trunojoyo.ac.id/wpcontent/uploads/2012/03/vol6_no1_p7.pdf.
- [5] Nugroho, Fresy and Arif, Yunifa Miftachul. 2016. *Desain Non Playble Character Sebagai Musuh Pada Game Sepeda Menggunakan Metode Markov State Machine*. Online: <http://etheses.uin.malang.ac.id/id/eprint/3451>
- [6] Millington, Ian. *Artifial Intelligence for Games*. San Francisco, U.S.A.: Morgan Kaufmann Publishers. 2006
- [7] Kristanto, Andri, 2004, *Kecerdasan Buatan*, Yogyakarta: Graha Ilmu
- [8] Srivastava, Mani, 2001, *Modeling Embedded Systems*, Online:http://statistic.gunadarma.ac.id/pub/books/Embedded/nesl.ucla.edu/lectures/L02_2pp.pdf.
- [9] Siagan. 2006. *Pengertian Markov Chain*. dari Online: <https://sutrisnoadityo.wordpress.com/2013/10/12/rantai-markov-markov-chain/>.
- [10] Chen, Jun., Suksompong, Prapun., Berger, Toby. *Communication through a Finite State Machine with Markov Property*. School of Electrical and Computer Engineering. Cornell University. Itacha, New York.

