

# Pergerakan Non-Playable Character dalam Game Stealth Menggunakan Finite State Machine

Angga Adetiya, Hanny Haryanto\*, Erlin Dolphina, Erna Zuni Astuti, Muljono

Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro  
Semarang, Indonesia

111201811230@mhs.dinus.ac.id, \*hanny.haryanto@dsn.dinus.ac.id, erlin.dolphina@dsn.dinus.ac.id,  
erna.zuni.astuti@dsn.dinus.ac.id, muljono@dsn.dinus.ac.id

**ABSTRACT** – Stealth is a game genre that relies on hiding skills and strategizing to complete missions. Challenges in the form of enemy Non-Playable Characters (NPC) are one of the vital elements of this genre game. However, NPCs generally only move according to a predetermined line or path of movement, so that their movements can be estimated easily after the player has seen them several times. This causes the movement of NPCs to look monotonous and not challenging. Finite State Machine (FSM) is an artificial intelligence method that can simply model enemy behavior. FSM consists of states and transitions where the state is the current state of the NPC, while the transition is the state of a state. This study models the behavior of enemy NPCs in stealth games using FSM so that it is not easy to predict their movements by adjusting idle time, walking, and moving speed. The results of this study indicate that NPC can have nine variations of behavior that can change according to these conditions.

**Keywords:** Artificial Intelligence; Finite State Machine; Game; Non-Playable Character (NPC); Stealth.

**ABSTRAK** – Stealth adalah genre game yang mengandalkan keahlian bersembunyi dan menyusun strategi untuk menyelesaikan misi. Tantangan dalam bentuk Non-Playable Character (NPC) musuh adalah salah satu elemen vital dari game genre ini. Namun NPC umumnya hanya bergerak menyesuaikan line atau jalur pergerakan yang sudah ditentukan sebelumnya, sehingga pergerakannya dapat diperkirakan dengan mudah setelah pemain melihatnya beberapa kali. Hal ini mengakibatkan pergerakan NPC akan terlihat monoton dan tidak menantang. Finite State Machine (FSM) adalah metode kecerdasan buatan yang dapat memodelkan perilaku musuh secara sederhana. FSM terdiri dari state dan transisi dimana state adalah status NPC saat ini, sedangkan transisi adalah kondisi dari suatu status. Penelitian ini memodelkan perilaku NPC musuh pada game stealth menggunakan FSM sehingga tidak mudah diprediksi pergerakannya dengan mengatur waktu idle, berjalan, dan kecepatan bergerak. Hasil dari penelitian ini menunjukkan NPC dapat mempunyai sembilan variasi perilaku yang dapat berubah sesuai dengan kondisi tersebut.

**Kata Kunci:** Finite State Machine; Game; Kecerdasan Buatan; Non-Playable Character (NPC); Stealth.

## 1. PENDAHULUAN

Tidak seperti umumnya game aksi, *game stealth* memaksa pemain untuk menghindari semua musuh, dan jika tidak memungkinkan, pemain diharuskan melumpuhkan musuh secara diam-diam dalam kegelapan atau tidak diketahui oleh musuh yang lain dengan menyerangnya dari belakang [1]. Cara menyelesaikan *game stealth* cukup sulit karena diharuskan membuat sedikit suara saat melewati musuh, menyerang musuh secara diam-diam, dan menyelesaikan misi tanpa diketahui musuh. Umumnya ada beberapa cara untuk menyelesaikan suatu misi dengan berbeda jalur atau cara bermain yang berbeda. Beberapa *game stealth* memberikan karakter mereka sebuah kemampuan spesial jika mereka diketahui oleh musuh. Kelebihan memainkan game ini adalah membuat pemain melatih kesabaran mereka dalam melakukan tindakan, dan memaksa mereka berpikir untuk membuat trik atau taktik untuk melewati musuh selama menyelesaikan *stage* yang dijalani. Hal ini membuat pemain lebih aktif berpikir dalam memainkan sebuah game agar kinerja otak meningkat dan menemukan hal

baru. Kekurangannya adalah *game play* yang masih sama dari generasi sebelumnya yaitu karakteristik *Non-Playable Character (NPC)* dari musuh yang masih berperilaku monoton yang mudah ditebak. Hal ini memudahkan pemain mengerti arah NPC sehingga mudah dikalahkan. Biasanya karakter utama dibuat lemah atau mudah dikalahkan jika pemain ketahuan oleh musuh. Contoh permainan tersebut adalah *Hitman 3* yang mana game tersebut adalah pemain berperan sebagai *spy* dan melakukan tugas dengan hati-hati dan juga menyamar untuk menyelesaikan tugas yang diberikan. Dalam *game* ini jika beberapa kali memainkannya maka pemain akan tahu pola pergerakan musuh yang sudah diatur oleh developernya sehingga pemain mudah menyelesaikan tugas yang diberikan karena sebelumnya sudah mengamati pola pergerakan musuh dengan seksama.

Kecerdasan Buatan dapat digunakan untuk membentuk perilaku dinamis dalam *game*, dalam hal ini adalah *game stealth*. Kecerdasan buatan diaplikasikan pada *game stealth* supaya perilaku menjadi lebih dinamis untuk membuat *game play* menjadi lebih hidup karena tidak mudah diprediksi oleh pemain. Perilaku NPC dapat



dengan acak mencari *player* atau menjaga tempat yang ditugaskan kepadanya. *Finite State Machine (FSM)* adalah salah satu metode dalam kecerdasan buatan yang umum digunakan untuk pembentukan perilaku agen cerdas di *game*.

Penelitian terkait yang dilakukan oleh [2] merancang perilaku NPC pada game *First Person Shooter (FPS)*. Metode yang digunakan penelitian ini adalah *Finite State Machine (FSM)*. Hasil dari penelitian ini adalah FSM dapat membentuk perilaku dinamis musuh. Penelitian yang dilakukan oleh [3] melakukan pemetaan perilaku NPC pada game *Role Playing Game*. Metode yang digunakan juga adalah *Finite State Machine (FSM)*. Penelitian ini menghasilkan perilaku agen cerdas atau NPC yang memberikan variasi respon dinamis. Penelitian yang dilakukan oleh [4] mengatur pergerakan manuver kelompok NPC berbasis Boids pada game RTS. Penelitian selanjutnya dilakukan oleh [5] yang melakukan perancangan perilaku NPC pada game FPS 3D. Metode yang digunakan adalah *Finite State Machine (FSM)* yang menghasilkan berbagai macam respon dari musuh yang berbeda-beda tergantung masukan dari pemain. Penelitian berikutnya dilakukan oleh [6] untuk meningkatkan perilaku NPC pada *game survival*. Dengan menggunakan metode *Finite State Machine (FSM)*. Hasil yang diperoleh adalah probabilitas kemungkinan perilaku yang dilakukan oleh NPC berdasarkan yang dilakukan oleh *player*. Selain FSM, pemodelan perilaku agen cerdas dalam juga sering menggunakan logika Fuzzy, seperti yang dilakukan oleh [7] dalam membentuk tingkat kesulitan otomatis, kemudian [8] juga menggunakan Fuzzy untuk mengatur tingkat kesulitan dari game. Penggunaan fuzzy untuk pengaturan perilaku NPC dilakukan oleh [9]. Fuzzy di dalam penelitian ini digunakan untuk membentuk perilaku penyerangan NPC. Kemudian [10] menggunakan Fuzzy untuk memodelkan perilaku NPC pada game bertipe RPG. Dari penelitian yang sudah dibahas dapat dilihat bahwa jarang atau belum ada pemodelan NPC pada game bertipe *stealth*. Hal ini disebabkan beberapa kesulitan spesifik yang terdapat pada game bertipe *stealth* ini. Menurut penelitian [11], kesulitan yang terjadi adalah pada pembentukan perilaku NPC. Jika perilaku NPC dibuat terlalu bervariasi maka permainan akan menjadi terlalu sulit, namun jika terlalu terpolat maka akan terkesan tidak alami pergerakannya [12].

Dari penelitian di atas, *Finite State Machine (FSM)* banyak digunakan untuk memodelkan perilaku baik di game yang bersifat penelitian maupun game komersial [13]. Hal ini disebabkan kelebihan FSM yang sederhana [14] namun dapat memodelkan perilaku yang cukup kompleks [15]. Penelitian ini berkontribusi pada pembentukan perilaku NPC yang lebih dinamis pada game *stealth* menggunakan FSM.

## 2. METODE PENELITIAN

Gambar 1 adalah alur tahapan penelitian yang dimulai dari pengumpulan data. Data yang dikumpulkan adalah data sekunder yang berupa referensi dari penelitian

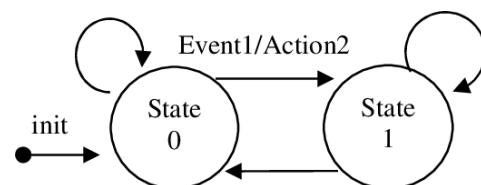
sebelumnya. Data juga berasal dari observasi, yang didapatkan dari memainkan game-game komersial seperti *Assassin's Creed* dan *Hitman 3*.



Gambar 1. Alur tahapan penelitian

Metode yang akan digunakan adalah Finite State Machine (FSM). FSM merupakan model yang umum digunakan untuk merancang perilaku musuh di game yang mempunyai kesederhanaan dan kemudahan dalam pemahaman implementasinya. Finite State Machine (FSM) adalah metodologi merancang system control yang menggambarkan perilaku atau prinsip kerja system dengan menggunakan tiga hal berikut State (Keadaan), Event (Kejadian), dan Action (Aksi) [16]. Finite State Machine (FSM) adalah model umum yang digunakan untuk merancang perilaku agen cerdas dalam game yang memiliki kelebihan dalam kesederhanaan komputasi dan kemudahan pemahaman dan implementasi [17].

Sistem dapat beralih atau beralih ke keadaan lain jika mendapat input atau peristiwa tertentu, baik yang berasal dari perangkat eksternal atau komponen dalam sistem itu sendiri. Transisi keadaan ini umumnya juga disertai dengan tindakan yang diambil oleh sistem ketika menanggapi input yang terjadi. Tindakan ini bisa berupa tindakan sederhana atau melibatkan serangkaian proses relatif [18].



Gambar 2. Contoh Diagram State Sederhana

Diagram di Gambar 2 menunjukkan FSM dengan dua buah state dan dua buah input serta empat tindakan

output yang berbeda seperti yang ditunjukkan pada gambar. Ketika system dihidupkan, system akan bertransisi menuju state0, dalam kondisi ini system akan menghasilkan Action1 jika input Event0, sedangkan jika terjadi Event1 maka Action akan dieksekusi kemudian system transisi ke State1 dan seterusnya [18].

Non-Playable Character (NPC) merupakan karakter yang tergolong tidak dikontrol oleh pemain melainkan dikontrol oleh computer. Dapat dikatakan sebagai karakter yang melakukan aktivitas secara otomatis.

NPC dirancang untuk membuat sebuah game agar tidak membosankan, seperti bermain game pertarungan yang mengharuskan dua orang untuk duel. Karakter pendukung NPC merupakan karakter yang terlibat dalam permainan tetapi tidak memiliki tujuan untuk memenangkan permainan melainkan melakukan peran yang mendukung player/lawan untuk memenangkan permainan. Selain itu NPC sendiri dirancang dengan berbagai perintah program untuk melakukan tugas atau peran tertentu di game. Salah satu jenis NPC adalah musuh, yang fungsinya memberikan tantangan utama dalam game.

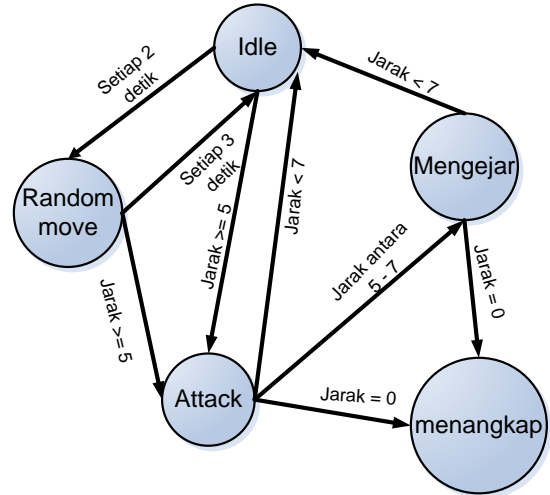
Dalam perancangan FSM ini, musuh akan dikategorikan dalam beberapa perilaku sesuai tingkat kesulitan. Musuh NPC dalam game ini ada 2 macam yaitu NPC Melee dan NPC range. 2 tipe NPC tersebut akan berada di setiap stage untuk menghalangi player menyelesaikan stage yang akan dilalui.

Tugas Utama NPC Melee adalah menjaga dan berpatroli di daerah paling dekat dengan player sehingga mengawali perlawanan player dengan NPC Melee, lalu ada NPC Range yang menjaga di pertengahan dan akhir jika player berhasil melewati NPC Melee. Jika Player berhasil melewati semua NPC musuh dan mengambil barang sesuai misi yang dituju maka player harus kembali ke titik awal untuk bisa menyelesaikan stage yang sedang dijalankan. Tabel 1 menunjukkan daftar perilaku untuk tiap jenis musuh.

**Tabel 1.** Daftar Perilaku Tiap Jenis Musuh

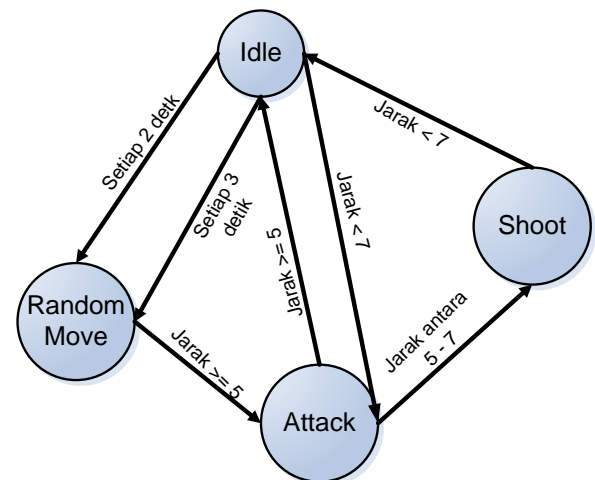
Musuh Melee (Jenis 1)	Musuh Range (Jenis 2)
1. Berjalan Acak	1. Berjalan Acak
2. Mengejar	2. Menembak
3. Menangkap	3. Idle
4. Idle	

Musuh jenis 1 (Melee) akan berjalan secara acak sesuai dengan area yang sudah ditentukan, jika player memasuki area deteksi oleh NPC maka NPC jenis Melee akan mengejar dan menangkap player. Jika berhasil tertangkap maka game over atau player mati. Jika player menjauh saar dikejar oleh NPC Melee dengan jarak < 7 maka NPC akan kembali idle dan kembali seperti sebelum NPC mengetahui player. Alur FSM musuh jenis 1 ditunjukkan pada Gambar 3.



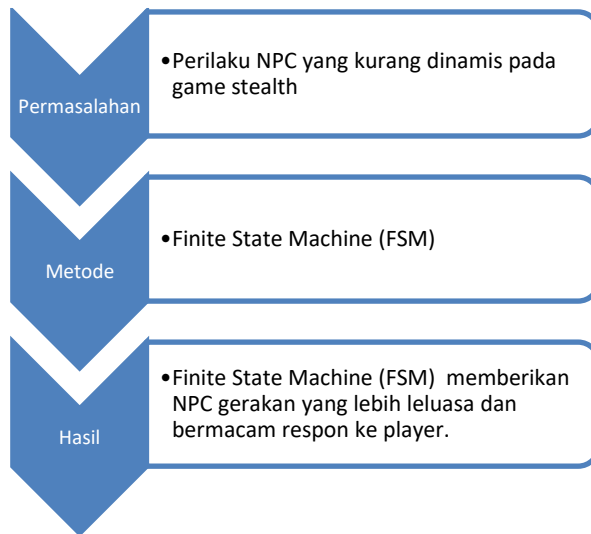
**Gambar 3.** Alur FSM Musuh Jenis 1 (Melee)

Musuh jenis 2 (Range) akan berjalan secara acak sesuai dengan daerah yang sudah ditentukan, jika player memasuki area deteksi NPC maka NPC akan menembak player. Jika player terkena tembakan maka player akan mati dan mengulang dari awal. Jika player menjauh dari NPC Range yang menembak sejauh < 7 maka NPC Range akan kembali idle dan kembali seperti semula. Alur FSM untuk musuh jenis 2 dapat dilihat pada Gambar 4.



**Gambar 4.** Alur FSM musuh jenis 2 (range)

FSM tersebut akan diimplementasikan ke dalam prototipe game stealth bertipe *top-down* (game yang dilihat dari atas). Evaluasi dilakukan dengan cara menguji tiap kondisi dan perilaku apa yang dihasilkan dari kondisi tersebut. Kerangka pemikiran penelitian ini disajikan pada Gambar 5.



Gambar 5. Kerangka Pemikiran

### 3. HASIL DAN PEMBAHASAN

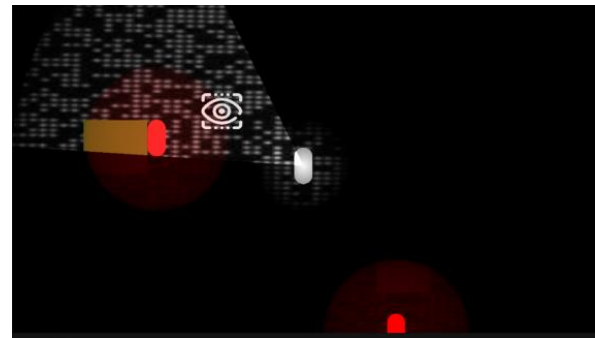
Rancangan FSM yang telah dibuat akan dilakukan pengujian pada NPC musuh. Untuk berpindah dari satu state ke state yang lain ditentukan berdasarkan jarak pixel dengan player. State yang telah dirancang diantaranya yaitu random move, mengejar, menembak.

Pada random move terdapat beberapa aspek seperti idle, pemilihan arah bergerak, lama waktu bergerak, kecepatan bergerak. Yang pertama dilakukan NPC adalah memilih arah pergerakan secara acak (atas, bawah, kiri, kanan) setelah itu lama waktu bergerak bergantung pada berapa second NPC tersebut diatur (Walk Time) dan kecepatan bergerak yang telah diatur (Move Speed), setelah NPC berhenti bergerak maka NPC akan memasuki Idle (Wait Time) sesuai yang telah diatur untuk NPC tersebut, setelah selesai waktu Idle maka NPC akan kembali memilih arah pergerakan secara acak.

Pada mengejar, NPC tipe Melee jika player memasuki area NPC musuh dalam satuan Pixel (Distance Between) yang sudah ditentukan maka fase random move akan diganti ke fase mengejar yang membuat NPC mengejar player atau mendekati player dengan kecepatan yang sudah ditentukan. Pada NPC tipe Range, jika player memasuki jarak dalam satuan pixel (Line of Sight) yang sudah ditentukan maka NPC tersebut akan mengejar player dengan kecepatan (speed) yang sudah ditentukan, setiap NPC dapat berbeda-beda dalam mengisi jarak mengejar dan juga kecepatan setiap NPC tipe Range. Jika player mendekati jangkauan tembak NPC (Shooting Range) yang sudah ditentukan maka NPC akan menembakkan peluru yang kecepatan pelurunya sudah diatur sebelumnya.

Gambar 5(a) menunjukkan pergerakan NPC musuh (warna putih) dengan Line of Sight (simbol mata ke arah kiri) kemudian dilanjutkan pergerakan NPC yang ditunjukkan pada Gambar 5(b). Gambar 5(b) memperlihatkan perilaku musuh yang bergerak acak (dalam hal ini bergerak ke atas) dengan Line of Sight (simbol mata) ke arah kanan. Perilaku musuh jenis Melee

Bergerak secara acak. Dan akan mengejar player jika berada dalam jangkauan NPC, jika player keluar dari jangkauan NPC maka NPC akan kembali bergerak acak.



(a)

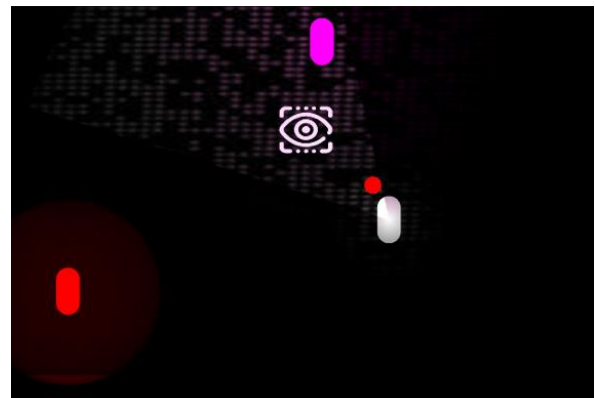


(b)

Gambar 5. Uji Coba Musuh Jenis Melee

- (a) Pergerakan musuh berwarna putih dengan Line of Sight (simbol mata)
- (b) Pergerakan acak musuh dan juga Line of Sight-nya (tanda mata) berubah arah

Dalam jangkauan NPC artinya berada di dalam area Line of Sight, yang disimbolkan dengan tanda mata. Dalam game *stealth*, jika di dalam Line of Sight musuh maka pemain ketahuan. Jadi misi utama pemain adalah sebisa mungkin menghindari masuk ke dalam Line of Sight musuh. NPC akan menyerang jika pemain berada dalam jarak serang. Sebelum sampai jarak serang, NPC akan terus mengejar sampai pemain keluar dari Line of Sight dari musuh.



Gambar 6. Perilaku musuh jenis Range (menembak dengan peluru warna merah ke arah pemain)



Gambar 6 memperlihatkan sekilas perilaku NPC Musuh jenis Range. Setelah bergerak secara acak, NPC akan diam sejenak kemudian lanjut bergerak acak. Jika pemain berada dalam Line of Sight atau ketahuan maka NPC akan melakukan pengejaran. Jika pemain masuk dalam jarak serang maka akan menembak ke arah pemain. Perilaku ini yang ditunjukkan pada Gambar 6 di atas. Hasil pengujian perilaku NPC ditunjukkan pada tabel transisi pada Tabel 2 berikut. Baris menunjukkan hasil dari  $\delta(q,a) = p$ , dimana perilaku / state ( $q$ ) dan isi kolom input  $a$  menunjukkan hasil perilaku yang merupakan pemetaan dari state  $q$  yang mendapat input  $a$  menghasilkan  $p$ .

**Tabel 2.** Hasil Pengujian State FSM untuk NPC Musuh Jenis Melee

$\delta$	a1	a2	a3	a4	a5	a6
q0	-	p1	p2	-	-	-
q1	p0	-	p2	-	-	-
q2	-	-	-	p0	p3	p4
q3	-	-	-	p0	-	p4
q4	-	-	-	-	-	-

Dimana q0 = idle, q1 = random move, q2 = random move, q3 = attack, q4 = mengejar, p0 = idle, p1 = random move, p2 = attack, p3 = mengejar, p4 = menangkap, a1 = 3 detik, a2 = 2 detik, a3 = jarak  $\geq 5$ , a4 = jarak  $< 7$ , a5 = jarak 5-7, dan a6 = jarak 0.

**Tabel 3.** Hasil Pengujian State FSM untuk NPC Musuh Jenis Range

$\delta$	a1	a2	a3	a4	a5	a6
q0	-	p1	-	p2	-	-
q1	-	-	p2	-	-	-
q2	-	-	p0	-	p3	-
q3	-	-	-	p0	-	-

Dimana q0 = idle, q1 = random move, q2 = attack, q3 = shoot, p0 = idle, p1 = random move, p2 = attack, p3 = shoot, a1 = 3 detik, a2 = 2 detik, a3 = jarak  $\geq 5$ , a4 = jarak  $< 7$ , a5 = jarak 5-7, dan a6 = jarak 0.

Tabel 2 dan Tabel 3 menunjukkan hasil perilaku NPC yang diujicobakan. Pada tabel tersebut dapat dilihat dari masing-masing input  $a$  menghasilkan antara satu sampai dua jenis perilaku.

#### 4. KESIMPULAN

Finite state machine (FSM) pada penelitian ini menghasilkan sembilan variasi perilaku NPC berdasarkan kondisi pada game stealth. Penggunaan metode Finite State Machine pada game Stealth dapat membuat permodelan pergerakan NPC musuh dan membuatnya tidak bisa diprediksi dengan mudah. Penelitian ini dapat

dikombinasikan dengan algoritma yang lebih akurat seperti Fuzzy. Agar dapat menghasilkan game yang lebih menarik dan menyenangkan.

#### 5. DAFTAR PUSTAKA

- [1] W. Min et al., "DeepStealth: Game-Based Learning Stealth Assessment With Deep Neural Networks," *IEEE Trans. Learn. Technol.*, vol. 13, no. 2, pp. 312–325, Apr. 2020, doi: 10.1109/TLT.2019.2922356.
- [2] Y. W. Astuti, A. Yunus, and M. Ahsan, "Perilaku Non Player Character (NPC) pada Game FPS 'Zombie Colonial Wars' Menggunakan Finite State Machine (FSM)," *Kurawal - J. Teknol. Inf. Dan Ind.*, vol. 2, no. 1, Art. no. 1, Mar. 2019, doi: 10.33479/kurawal.2019.2.1.53-63.
- [3] M. B. Nendya, S. G. Gunanto, and R. G. Santosa, "Pemetaan Perilaku Non-Playable Character Pada Permainan Berbasis Role Playing Game Menggunakan Metode Finite State Machine," *J. Animat. Games Stud.*, vol. 1, no. 2, Art. no. 2, 2015, doi: 10.24821/jags.v1i2.1304.
- [4] S. D. Ribaupierre, B. Kapralos, F. Haji, E. Stroulia, A. Dubrowski, and R. Eagleson, "Healthcare Training Enhancement Through Virtual Reality and Serious Games," in *Virtual, Augmented Reality and Serious Games for Healthcare 1, Intelligent Systems Reference Library*, vol. 68, 2014, pp. 9–28. doi: 10.1007/978-3-642-54816-1.
- [5] M. A. W. Saputra, J. N. Fadila, and F. Nugroho, "Perancangan Game First Person Shooter 3D 'Saving Islamic Kingdom' dengan Menggunakan Finite State Machine (FSM)," *Walisono J. Inf. Technol.*, vol. 2, no. 2, Art. no. 2, Dec. 2020, doi: 10.21580/wjit.2020.2.2.6981.
- [6] D. Gunawan, A. Atthariq, and A. Aswandi, "Meningkatkan Behaviour Npc Pada Game 3d Survival Menggunakan Metode Markov," *J. Infomedia Teknik Inform. Multimed. Jar.*, vol. 2, no. 1, Art. no. 1, Dec. 2017, doi: 10.30811/v2i1.477.
- [7] I. A. Ahmadi, E. M. A. Jonemaro, and M. A. Akbar, "Penerapan Algoritme Logika Fuzzy Untuk Dynamic Difficulty Scaling Pada Game Labirin," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput. J-PTIHK Univ. Branijaya*, 2018.
- [8] A. B. Harisa, H. Haryanto, and H. A. Santoso, "Model Tingkat Kesulitan Dinamis Berbasis Logika Fuzzy," *Semin. Nas. Teknol. Inf. Dan Multimed.*, 2016.

- [9] M. A. Darmawan, H. Haryanto, and Y. Rahayu, "Perilaku Penyerangan NPC Berbasis Fuzzy Sugeno pada Game Action-RPG Bertema Sejarah Geger Pacinan," *Creat. Inf. Technol. J.*, vol. 4, no. 3, pp. 195–206, 2017.
- [10] D. Ratanajaya and H. A. Wibawa, "Implementasi Kecerdasan Buatan dalam Menentukan Aksi Karakter pada Game RPG dengan Logika Fuzzy Tsukamoto," *Khazanah Inform. J. Ilmu Komput. Dan Inform.*, vol. 4, no. 2, p. 82, 2018, doi: 10.23917/khif.v4i2.6744.
- [11] C. Paixão, E. Vilar, P. Noriega, and F. Rebelo, "Can a Stealth Game Be Used to Learn Interaction Design Concepts?," in *Advances in Ergonomics in Design*, Cham, 2021, pp. 882–889. doi: 10.1007/978-3-030-79760-7\_106.
- [12] E. Byrne, *Game level design*. Hingham, Mass. : Charles River Media, 2005. Accessed: Oct. 31, 2022. [Online]. Available: <http://archive.org/details/gameleveldesign0000byrn>
- [13] M. F. Syahputra, A. Arippa, R. F. Rahmat, and U. Andayani, "Historical Theme Game Using Finite State Machine for Actor Behaviour," *J. Phys. Conf. Ser.*, vol. 1235, p. 12122, Jun. 2019, doi: 10.1088/1742-6596/1235/1/012122.
- [14] S. Lucas, M. Mateas, M. Preuss, P. Spronck, and J. Togelius, "Artificial and Computational Intelligence in Games," 5, 2012. doi: <http://dx.doi.org/10.4230/DagRep.2.5.43>.
- [15] I. Millington and J. Funge, *Artificial Intelligence for Games, Second Edition*. 2009. doi: 10.1017/S0263574700004070.
- [16] A. Badruddin, "Perancangan dan Implementasi Finite State Machine pada Game Castle of Illusion," *Skripsi*, 2019.
- [17] H. Tito, Bimantoro1, Hanny, "Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia," *Appl. Intell. Syst.*, vol. Vol.1, No., no. 3, pp. 210–219, 2016.
- [18] 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.