

Strategi Menyerang Jarak Dekat Menggunakan Klasifikasi Bayesian Pada NPC (*Non Player Character*)

Siti Asmiatun¹, Latus Hermawan², Tri Daryatni³

Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang 50131
E-mail : ¹ mieafighting@gmail.com, ² fanytiuz@gmail.com

ABSTRAK

Makalah ini membahas tentang strategi menyerang jarak dekat. Dalam game komputer terutama FPS (*First Person Shooter*) sangat penting memperhatikan strategi NPC (*Non Player Character*) untuk membuat game menjadi lebih atraktif dan realistis. Dalam penelitian ini akan membahas tentang strategi menyerang yang dilakukan oleh NPC. Yang dimaksud strategi menyerang dalam penelitian ini adalah dengan mengklasifikasikan cara penyerangan NPC ketika NPC akan menyerang dalam jarak dekat. Penelitian ini menerapkan algoritma bayesian untuk mengklasifikasikan cara penyerangan NPC tersebut. Klasifikasi tersebut untuk diharapkan dapat meningkatkan strategi menyerang melawan musuh. Agar dalam permainan game tersebut, NPC tidak akan mudah terkalahkan. Dalam pengklasifikasian penyerangan NPC akan dibagi menjadi tiga penyerangan yaitu menggigit, menerkam dan mengecoh. Sedangkan untuk variabel yang digunakan dalam pengklasifikasian bayesian adalah *health point*, *defence point* dan jarak dari NPC.

Kata kunci : Strategi, Menyerang, Jarak, Dekat, Klasifikasi, Bayesian

1. PENDAHULUAN

Kecerdasan suatu *game* bergantung pada perilaku dan kemampuan dari *Non Player Character* tanpa campur tangan manusia (*autonomus*) dengan tujuan untuk mewujudkan permainan (*game*) lebih menarik dan realistis. Perilaku suatu agen berupa NPC hingga saat ini terus dikembangkan dengan menggunakan kecerdasan buatan. Dalam suatu *game* tertentu seperti *First Person Shooter* (FPS) terutama untuk serangan jarak dekat (*close combat*), NPC harus dibuat senatural mungkin agar gerakan maupun karakternya seperti halnya manusia. Sehingga *game* tersebut menjadi lebih menarik dan realistis.

AI (*Artificial Intelligence*) pada *game* FPS umumnya terdiri dari perencanaan *path*, mengambil *item*, menggunakan *item*, dan berperang [1]. Khusus untuk berperang NPC juga diharapkan mempunyai strategi-strategi khusus seperti halnya manusia [2]. Dalam penyerangan terlebih untuk serangan jarak dekat menjadi hal yang menarik untuk diteliti. Terutama strategi bagaimana membuat perilaku NPC untuk mengalahkan musuh. Salah satunya adalah dengan mengklasifikasikan perilaku NPC ke dalam beberapa kelompok perilaku untuk menyerang *player* sesuai dengan kondisi dan menggunakan energi minimum.

Strategi menyerang merupakan substansi penting dalam pengembangan *game* untuk mewujudkan *game* menjadi realistis. Beberapa penelitian untuk strategi menyerang telah banyak dilakukan. Model strategi menyerang pada *game* FPS, dimana untuk mendesain perilaku NPC digunakan *Hierarchy Finite State Machine*, dan untuk menentukan respon perilaku yang dihadapi NPC digunakan logika diterapkan oleh mardidkk [3]. Fanani menerapkan algoritma *harmony search* untuk optimasi perilaku perang NPC, algoritma ini menghasilkan pemilihan urutan lawan agar permainan lebih efektif dan lebih cepat mencapai tujuan tim dengan energi minimum [4]. Shu-fai wong telah menerapkan pendekatan klasifikasi bayesian untuk mengenali 10 gerakan dasar pada fitur *motion vector* dengan 628 sampel [5].

Mereferensi dari beberapa penelitian diatas, maka penelitian ini mencoba menggunakan strategi menyerang jarak dekat dengan cara mengklasifikasikan perilaku penyerangan NPC dengan menggunakan algoritma bayesian.

2. TEORI PENUNJANG

Untuk memebrikan gambaran secara umum, pada bab 2 ini akan membahas secara singkat mengenai NPC, Bayesian, AI (*Artificial Intelligence*) dan teori penunjang lainnya yang berkaitan dengan penelitian ini.

2.1. NPC (Non Player Character)

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

2.2. Klasifikasi Bayesian

Bayesian filter atau Naive Bayes Classifier merupakan metode terbaru yang digunakan untuk mengklasifikasikan sekumpulan dokumen. Algoritma ini memanfaatkan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes di masa depan berdasarkan pengalaman di masa sebelumnya.

Dua kelompok peneliti, satu oleh Pantel dan Lin, dan yang lain oleh Microsoft Research memperkenalkan metode statistik Bayesian ini pada teknologi anti spam filter. Algoritma Bayesian filtering ini populer adalah pendekatan yang dilakukan oleh Paul Graham [6].

$$P(x_i | V_j) = \frac{n_k + 1}{N + |\text{variabel}|} \quad (1)$$

Keterangan :

|docs j| = jumlah dokumen setiap kategori j

|contoh| = jumlah dokumen dari semua kategori

n_k = jumlah frekuensi setiap variabel

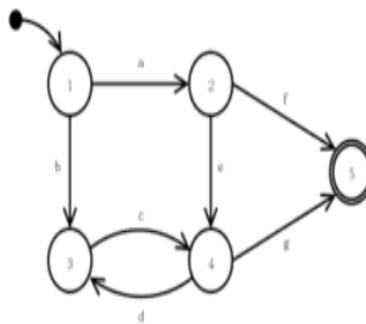
n = jumlah frekuensi variabel dari setiap kategori

|variabel| = jumlah semua variabel dari semua kategori

2.3. FSM (Finite State Machine)

FSM adalah salah satu metode yang paling dikenal untuk memodelkan perilaku agen/NPC dalam sebuah game. Hal ini dikarenakan kesederhanaan dan kemudahan FSM untuk diimplementasikan [4].

Struktur FSM terdiri dari dua komponen utama yaitu keadaan (state) dan transisi (transition). State merupakan keadaan objek saat ini, sedangkan transisi adalah hal yang dilakukan untuk dapat berpindah dari satu state ke state yang lain (Gambar 2)



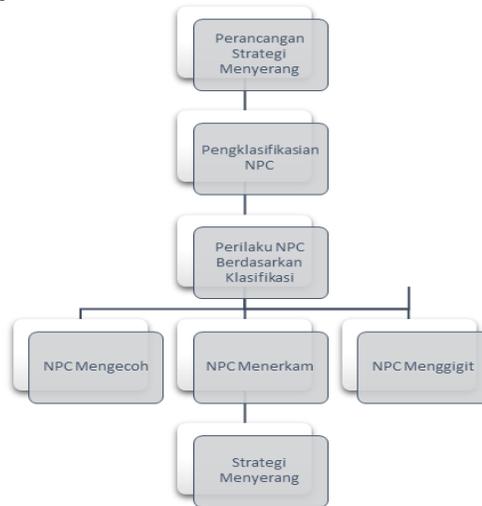
Gambar 2: Diagram FSM

Simbol State digambarkan dengan lingkaran, sedangkan transisi disimbolkan dengan anak panah dengan arah tertentu. Pada Gambar 2, angka 1-5 adalah state sedangkan huruf a-g adalah transisi. Kelebihan dari FSM adalah sederhana dan mudah diimplementasikan. Sedangkan kekurangannya, dalam sistem yang besar FSM akan sulit dipelihara. Implementasikan FSM dalam bahasa pemrograman dapat dilakukan dengan berbagai cara, yaitu: cara tradisional (menggunakan switch-case), look-up tabel (menggunakan matriks untuk menyimpan state), dan dengan paradigma Object Oriented. FSM berkembang menjadi beberapa variasi bila digabungkan dengan metode lain, antara lain: Fuzzy State Machine (FuSM), Probabilistic FSM (PFSM), Hierarchical FSM (HFSM), dan lainnya.

3. METODE PENELITIAN

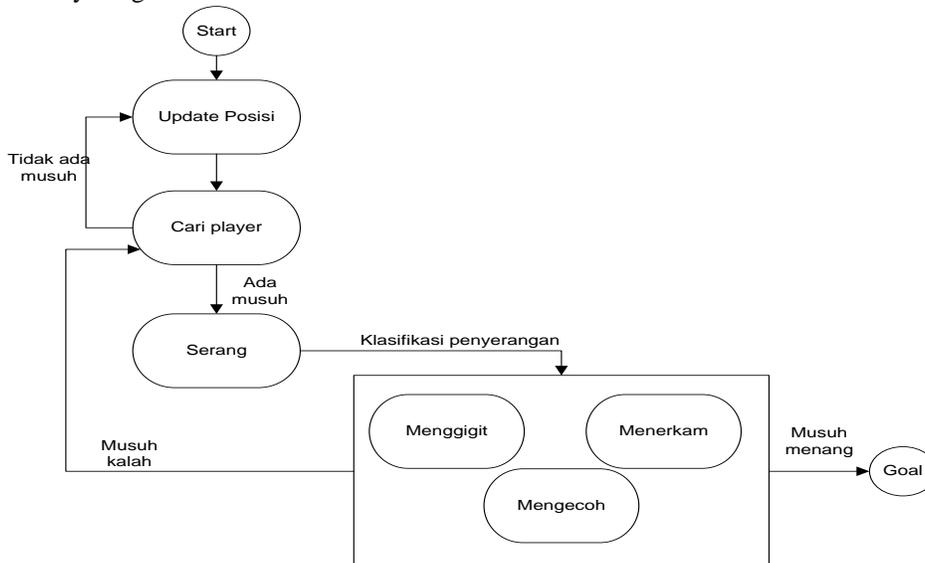
Metode penelitian yang digunakan untuk mendesain strategi menyerang jarak dekat adalah menggunakan *Finite State Machine* (FSM), sedangkan untuk menentukan klasifikasi perilaku NPC berdasarkan digunakan algoritma bayesian.

Pada penelitian ini dilakukan pengklasifikasian untuk strategi penyerangan jarak dekat. Kalasifikasi penyerangan NPC terdiri dari menggigit, menerkam dan mengecoh. Hasil dari klasifikasi bayesian tersebut nantinya berupa nilai yang menyatakan kondisi dari NPC, dan nilai tersebut nantinya yang akan dijadikan acuan untuk menentukan perilaku penyerangan NPC yang sesuai dengan kondisi NPC. Gambar 3 dibawah ini merupakan penjelasan alur dari penelitian ini.



Gambar 3 Blok diagram penelitian

Sedangkan gambar 4 dibawah ini menunjukkan desain FSM untuk penelitian ini. Dalam penelitian ini hanya terfokus pada pengklasifikasian NPC sebelum NPC tersebut menunjukkan perilaku seperti menggigit, menerkam dan mengecoh untuk menyerang musuh.



Gambar 4 FSM Strategi Menyerang

Tabel 1 dibawah ini adalah data *training* yang akan digunakan untuk pengklasifikasian NPC dengan kondisi HP, jarak, dan DP yang berbeda dalam satu NPC.

Tabel 1. Data Training

Nama NPC	Health Point (HP)	Jarak NPC	Defence Point (DP)
NPC 1	75	8,7	80
NPC 1	85	6,5	90
NPC 1	50	8,4	70
NPC 1	75	6,8	85
NPC 1	70	4,3	90

4. HASIL DAN PEMBAHASAN

4.1. Perancangan Strategi Penyerangan

Dalam penelitian ini akan digunakan strategi penyerangan dengan beberapa perilaku penyerangan yang akan diklasifikasikan sesuai dengan kondisi NPC. Data *Training* pada tabel 1 diatas nantinya akan digunakan acuan untuk menghitung nilai NPC dengan kondisi HP, jarak dan DP yang berbeda dengan algoritma bayesian. Setelah mendapatkan nilai dari hasil perhitungan bayesian, metode *fuzzy* akan digunakan untuk menentukan perilaku penyerangan NPC. Tabel 2 dibawah ini merupakan rule yang digunakan untuk penentuan perilaku NPC sesuai dengan hasil dari pengklasifikasi bayesian.

Tabel 2. Rule Penentuan Perilaku

Nama Perilaku	Nilai
Menggigit	0.6 - 1.0
Menerkam	0.3 - 0.7
Mengecoh	0.0 - 0.4

4.2. Pengklasifikasian NPC dengan Bayesian

4.2.1 Perhitungan Bayesian

Contoh perhitungan bayes untuk NPC 1 dengan kondisi HP 75, jarak 8,7 dan DP 80. Rumus :

$$P(x_i | V_j) = \frac{n_k + 1}{N + |\text{variabel}|} \quad (2)$$

Nk1 = 75 (jumlah dari variabel hp)

Nk2 = 8,7 (jumlah dari variabel jarak)

Nk3 = 80 (jumlah dari variabel dp)

N = 75 + 8,7 + 80 = 163,7

N didapatkan dari total masing - masing jumlah variabel yang digunakan.

Variabel = 3 n = 163,7

$$\frac{(75 + 1)(8,7 + 1)(75 + 1)}{163,7 + 3}$$

$$\frac{76 \times 9,7 \times 81}{163,7 + 3}$$

$$\frac{59713,2}{166,7 \times 3}$$

$$\frac{59713,2}{4632407,963} = 0,0128$$

4.2.2 Hasil dari Klasifikasi Bayesian

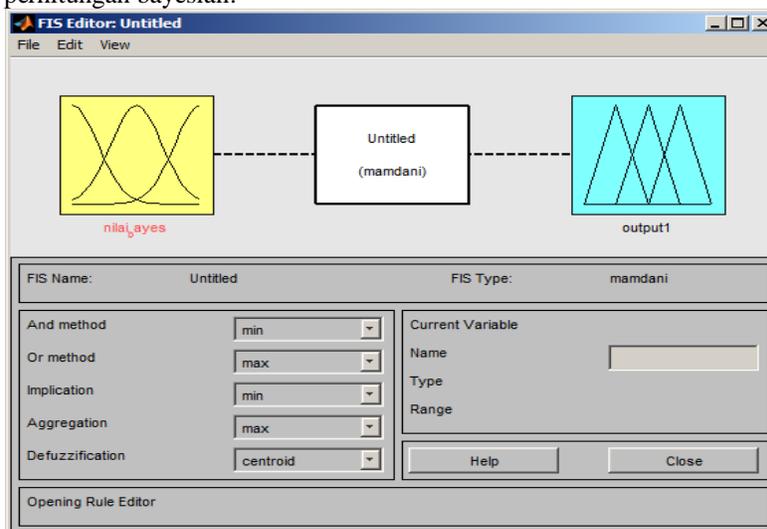
Tabel 3 dibawah ini merupakan hasil dari perhitungan kalsifikasi bayesian dengan kondisi HP, Jarak dan DP NPC yang berbeda.

Tabel 3 Hasil Perhitungan Bayes

Nama NPC	Health Point	Jarak NPC	Defence Point	Hasil Bayes
NPC 1	75	8,7	80	0,9536769
NPC 1	85	6,5	90	1,01910828
NPC 1	50	8,4	70	0,7492762
NPC 1	75	6,8	85	0,9716271
NPC 1	70	4,3	90	0,95715113

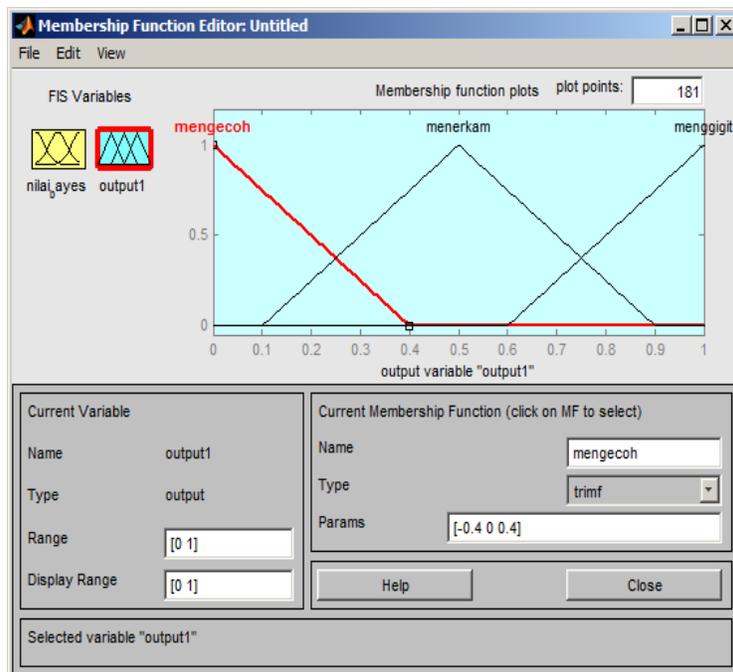
4.3. Perilaku NPC Berdasarkan Klasifikasi Bayesian

Dari hasil perhitungan bayesian seperti tabel 3 diatas, tahap selanjutnya adalah menentukan perilaku penyerangan NPC yang sesuai dengan kondisi NPC. Dalam penelitian ini untuk menentukan perilaku penyerangan NPC digunakan logika *fuzzy*. Gambar 5 dibawah merupakan penerapan logika *fuzzy* dengan menggunakan variabel nilai hasil dari perhitungan bayesian.



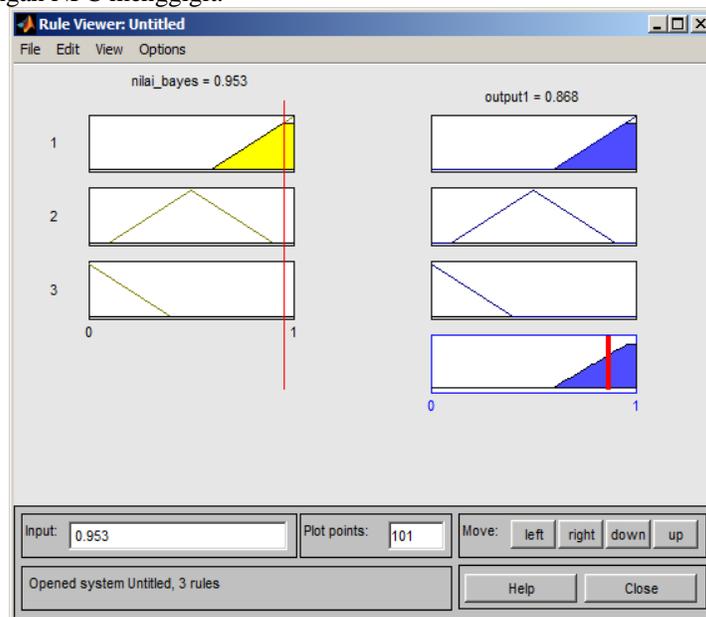
Gambar 5 Penentuan Perilaku Dengan Logika *Fuzzy*

Dan proses selanjutnya yaitu menentukan rule yang terdapat didalam proses mamdani yang tepat dikotak putih pada gambar 5 diatas. Setelah rule ditentukan baru kita menentukan nilai output yang nantinya akan menjadi acuan untuk menentukan perilaku penyerangan NPC seperti pada gambar 6 dibawah ini.



Gambar 6 Nilai Output Untuk Logika Fuzzy

Dan gambar 7 dibawah ini adalah hasil dari *rule viewer*. Sebagai contoh disini menggunakan nilai bayes yang digunakan sebesar 0.953 maka hasil outputnya dapat terlihat dalam area biru yang menunjukkan dengan perilaku penyerangan NPC menggigit.



Gambar 7 Hasil Rule Viewer Logika Fuzzy

5. KESIMPULAN DAN SARAN

Berdasarkan hasil analisa pada algoritma bayesian sebagai salah satu metode yang ditawarkan untuk menyelesaikan klasifikasi NPC yang bertujuan untuk kalsifikasi strategi menyerang jarak dekat seperti FPS, dapat diambil beberapa kesimpulan. Juga ditambahkan saran dalam pengembangan penelitian berikutnya.

5.1. Kesimpulan

- a. Dengan algoritma klasifikasi bayesian, strategi penyerangan pada masing-masing NPC dapat dikelompokkan sesuai dengan kondisi dari masing – masing NPC agar tidak mudah terkalahkan.
- b. *Finite State Machine* digunakan untuk menggambarkan perilaku strategi menyerang pada *game*.
- c. Tingkat kemenangan strategi menyerang ini belum bisa diketahui karena belum disimulasikan dalam *game* FPS.

5.2. Saran

- a. Untuk penelitian berikutnya parameter bisa ditambah dengan membandingkan *health* NPC player dengan NPC lawan.
- b. Untuk mengurangi waktu komputasi jumlah NPC lawan dapat dikurangi dengan hanya memilih NPC lawan yang berada di area serangan.
- c. Metode yang diusulkan dalam menyelesaikan masalah klasifikasi NPC untuk pengklasifikasian strategi menyerang jarak dekat layak untuk dicobakan pada *real game*.

DAFTAR PUSTAKA

- [1] JinHyuk Hong dan Sung-Bae Cho. *Evolving Reactive NPCs for the Real-Time Simulation Game*. CIG, 2005.
- [2] Yunifa Miftachul Arif, Fachrul Kurniawan dan Fresy Nugroho. *Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy*. National Seminar on Electrical, Informatics, and Its Education, 2011.
- [3] Yunita Miftachul Arif, Mochamad Hariadi dan Supeno Mardi S.N. *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite Machine dan Logika Fuzzy*.
- [4] Fanani dan Nurul Zainal. *Optimasi Strategi Menyerang Kelompok NPC pada Game Pertarungan Jarak Dekat Menggunakan Algoritma Harmony Search*. 2012.
- [5] Shu-Fai Wong dan Roberto Cipolla. *Real-time Interpretation of Hand Motions using a Sparse Bayesian Classifier on Motion Gradient Orientation Images*. Department of Engineering, University of Cambridge.2005
- [6] Rachli, Muhamad.2007. *Email Filtering menggunakan Naive Bayesian*. Bandung : Tugas Akhir Jurusan Teknik Elektro Institut Teknologi Bandung.
- [7] JavaMail™ API Design Specification. Sun Microsystems USA, Desember 2005