

## **PATHFINDING DAN PERILAKU PERANG NON PLAYING CHARACTER (NPC) MENGGUNAKAN a\* dan FUZZY LOGIC PADA GAME SIMULATOR TANK**

Wali Muhammad  
Program Studi Teknik Informatika Politeknik TEDC Bandung  
E-mail: [walim\\_2005@yahoo.com](mailto:walim_2005@yahoo.com)

### **Abstrak**

Simulasi adalah proses implementasi model ke dalam program komputer (software) dengan / tanpa sirkuit elektronik, seperti menjalankan software yang meniru atau menyerupai perilaku sistem nyata tertentu. Game Simulator Tank adalah salah satu jenis simulasi yang menjadikan tank sebagai kendaraan para kru/awak tank dalam operasi dan praktek perang dengan menggunakan kendaraan tempur tank. Game Simulator Tank dijalankan untuk meniru gerakan tank dan atau berbagai jenis kendaraan yang terlibat dalam simulasi ini, dijalankan di medan yang menyerupai situasi aktual dan dibantu dengan software visualisasi. Game Simulator Tank akan baik bila diterapkan mesin Kecerdasan Buatan (AI) dengan baik pada permainannya. AI dalam game simulator umumnya terdiri dari perhitungan pemilihan jalur gerakan, mendeteksi dan bergerak sesuai dengan kehadiran lawan, dan melakukan taktik pertempuran sesuai dengan situasi dan kondisi yang sebenarnya. Hal yang sangat penting dalam Game Simulator Tank adalah aplikasi AI dalam karakter bukan pemain atau *non playing character* (NPC). Untuk menentukan pemilihan jalur tank bukan pemain akan menggunakan kondisi nyata seperti latihan di lapangan yang sebenarnya seperti yang digunakan di medan pertempuran. Dalam hal taktik berperilaku perang, tank bukan pemain memiliki strategi tertentu sebagaimana perilaku tank apabila dikemudikan oleh manusia. Strategi yang dimaksud adalah strategi untuk mengejar lawan, berpatroli di area sendiri, diam, menyerang, bertahan dan lari.

Kata kunci : Game Simulator Tank, A \*, Logika Fuzzy, NPC.

### **Abstract**

*Simulation is the process of implementation of the model into a computer program (software) with / without electronic circuits, as well as executes the software that imitates or resembles the behavior of real systems specific. Tank Simulator games are one type of simulation is to be a vehicle for the train crew in the operation and practice of war using tanks combat vehicles. Tank Simulator game is enabled to mimic the movement of tanks and or any type of vehicle involved in this simulation is run on terrain that resembles the actual situation and assisted with visualization software. Tank Simulator Game will be good when it is applied Artificial Intelligence (AI) engine well on his game. AI in a game simulator generally consists of the calculation of the movement path selection detects and moves according to the presence of an opponent and do battle tactics in accordance with the existing circumstances and conditions. It is very important in Tank Simulator Game is the application of AI in the tank instead of the player character or a non-playing character (NPC). To determine the selection of tank track instead of the player will use the path is as real as the actual field exercises that are used in the terrain. In terms of tactics behave war, tank instead of the player having a specific strategy such as that tank piloted by humans. Intended strategy is a strategy for the opponent, patrolling the area alone, silent, attack, defend, and run.*

*Key words: Tank Simulator Game, A \*, Fuzzy Logic, NPC.*

## **1. Pendahuluan**

### **1.1 Latar Belakang**

*Game Simulator Tank* (GST) adalah salah satu teknik pembelajaran yang menggabungkan beberapa unsur yang ada pada *game* perang dan sistem simulasi untuk mendapatkan hasil yang

lebih baik. GST ini dapat berfungsi untuk menirukan gerakan tank asli dengan peralatan, tombol dan komponen didalamnya yang menyerupai barang aslinya dan dapat digunakan untuk pembelajaran

pengemudian dan pengoperasian tank baik untuk operasi biasa maupun untuk pengoperasian dengan setrategi perang. Disamping itu pada GST dalam melakukan peperangan menggunakan strategi perang dan pola tempur yang baku, seperti dalam melakukan peperangan perlu diperhitungkan jalur terbaik yang harus dilalui, juga kondisi fisik tank, jumlah amunisi yang tersedia, jumlah lawan dan jumlah kawan. Algoritma A\* adalah algoritma pencarian pada graf yang mencari jalur dari satu verteks (simpul) awal ke sebuah verteks akhir yang telah ditentukan. Algoritma A\* menggunakan pendekatan heuristik yang memberikan peringkat ke tiap-tiap verteks yang direpresentasikan dengan titik, dengan cara memperkirakan rute terbaik yang dapat dilalui dari verteks tersebut. Setelah itu tiap-tiap verteks tersebut dicek satu-persatu berdasarkan urutan yang dibuat dengan pendekatan heuristik tersebut [Millington, 2009]. Algoritma A\* sukses besar dalam menemukan jalur jalan pada sebuah *game*, karena itu banyak peneliti yang menaruh harapan untuk mengoptimalkan proses pencarian menggunakan algoritma A\* untuk memenuhi perubahan kebutuhan *game* [Cui, 2011].

Logika fuzzy merupakan salah satu dari *Artificial Intelligent (AI)* yang memiliki kemampuan untuk menyelesaikan masalah perilaku sistem yang kompleks dan memiliki ketidak pastian, sehingga apabila diterapkan pada sebuah sistem, logika fuzzy dapat mengatasi proses dari suatu sistem yang dinamik, kompleks dan memiliki ketidak pastian yang disebabkan oleh kemampuan mengolah informasi numerik dari variabel yang diukur. Dalam hal perilaku perang, Karakter Bukan Pemain (KBP) harus mempunyai strategi khusus seperti halnya para ahli perang bertempur secara riil di medan perang. Strategi yang dimaksud adalah berupa strategi mencari lawan, menyerang lawan, menghindari lawan, bertahan dalam arti berpatroli di area sendiri dan baru membalas bila diserang, diam, dan lari. Dalam pengambilan keputusan untuk berperilaku perang ini digunakan metode logika fuzzy.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah disebutkan dalam uraian sebelumnya maka dapat dirumuskan beberapa permasalahan sebagai berikut:

a. Beberapa teknik pembelajaran perang khususnya dengan menggunakan kendaran tempur tank yang ada banyak yang belum memenuhi sasaran, dimana lokasi dan situasi

latihan tidak menyerupai dengan keadaan sebenarnya seperti bila peperangan dilakukan di medan tempur nyata.

b. Dalam menentukan pemilihan jalur (path) pada pergerakan KBP belum memperhitungkan jalur jalan riil sesuai pada medan latihan sebenarnya, seperti jalan berlumpur, ada genangan air/sungai.

c. Kurang akuratnya penerapan strategi perang yang dipakai, dimana untuk menentukan respon perilaku terhadap setiap perubahan dalam memberikan rekomendasi langkah apa yang harus dilakukan bagi Karakter Bukan Pemain menghadapi situasi dan kondisi yang sedang terjadi.

### 1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari adanya hasil penelitian ini adalah:

a. Untuk lebih mempertajam hasil dari sebuah pembelajaran di bidang Militer khususnya dalam hal belajar mengoperasikan dan latihan perang menggunakan tank tiruan, dilakukan di medan perang yang menyerupai keadaan riil dan dioperasikan secara virtual.

b. Agar dalam melakukan pemilihan jalur pergerakan, Karakter Bukan Pemain sudah memperhitungkan semua ruang keadaan (state) jalur jalan dan medan tempur yang ada.

c. Menentukan algoritma yang paling optimal untuk diterapkan pada *Game Simulator Tank* dalam hal penentuan jalur pergerakan dan perilaku taktis peperangan.

### 1.4 Manfaat Penelitian

a. Manfaat Praktis

Penerapan dari hasil penelitian ini nantinya adalah untuk Simulator Terintegrasi Tank yang akan digunakan oleh TNI angkatan darat dalam melatih para kru kendaraan tempur tank untuk meningkatkan kemampuannya dalam mengoperasikan tank baik untuk situasi damai maupun situasi perang dalam rangka menjaga keamanan dan kesatuan wilayah Negara Kesatuan Republik Indonesia.

b. Manfaat Teoritis

Manfaat teoritis dari hasil penelitian ini diharapkan dapat memberikan sumbangan pada ilmu pengetahuan dalam hal berikut:

b1. Untuk pengembangan teori yang berkaitan dengan perhitungan pemilihan jalur pergerakan (pathfinding) karakter (kendaraan tempur) bukan pemain menggunakan algoritma A\*.

b2. Penentuan perilaku perang Karakter Bukan Pemain menggunakan metode Logika Fuzzy.

**2. Landasan Teori**

**2.1 Algoritma A\* Pathfinding**

Algoritma A\* merupakan salah satu algoritma pencarian *graph* terbaik yang mampu menemukan jalur dengan biaya pengeluaran paling sedikit dari titik permulaan yang diberikan sampai ke titik tujuan yang diharapkan. Algoritma A\* dapat diimplementasikan lebih efisien karena tidak ada simpul yang diproses berulang-ulang. Algoritma ini memeriksa simpul dengan menggabungkan  $g(n)$ , yaitu *cost* (biaya) yang dibutuhkan untuk mencapai sebuah simpul dan  $h(n)$  yaitu *cost* yang didapat dari simpul ke tujuan. Sehingga dapat dirumuskan sebagai :

$$f(n) = g(n) + h(n) \tag{2.1}$$

Keterangan :

$f(n)$  adalah total biaya (*cost*) sebuah jalur (*path*) dari simpul awal ke simpul tujuan (*goal*) melalui simpul  $n$ .

$g(n)$  adalah biaya (*cost*) yang dibutuhkan oleh sebuah jalur untuk mencapai simpul  $n$  dari simpul awal.

$h(n)$  adalah biaya (*cost*) sebuah jalur (*path*) .

**2.2 Fungsi Heuristic**

Ada beberapa fungsi heuristic umum yang bisa dipakai untuk algoritma A\* ini. Salah satunya adalah yang dikenal dengan istilah *Manhattan Distance*. Fungsi heuristic ini digunakan untuk kasus dimana pergerakan pada peta hanya lurus (horizontal atau vertikal), tidak diperbolehkan pergerakan diagonal [Adipranata, 2007].

Perhitungan nilai heuristic untuk node ke- $n$  menggunakan Manhattan Distance adalah sebagai berikut :

$$h(n) = (abs(n.x - goal.x) + abs(n.y - goal.y)) \tag{2.2}$$

Dimana  $h(n)$  adalah nilai *heuristic* untuk *node*  $n$ , dan *goal* adalah *node* tujuan.

Jika pergerakan diagonal pada peta diperbolehkan, maka digunakan fungsi heuristic selain Manhattan Distance. Untuk mendekati kenyataan, *cost* untuk perpindahan node secara diagonal dan orthogonal dibedakan. *Cost* diagonal adalah 1,4 kali *cost* perpindahan secara orthogonal.

Maka fungsi heuristic yang digunakan adalah sebagai berikut: .....

$$\begin{aligned} h\_diagonal(n) &= \min(abs(n.x - goal.x) + abs(n.y - goal.y)) \\ h\_orthogonal(n) &= (abs(n.x - goal.x) + abs(n.y - goal.y)) \\ h(n) &= d2 * h\_diagonal(n) + d1 * (h\_orthogonal(n) - (2 * h\_diagonal(n))) \end{aligned}$$

Keterangan:

$h(n)$  : beaya (*cost*) heuristic untuk node  $n$

$h\_diagonal(n)$ : banyaknya langkah diagonal yang bisa diambil untuk mencapai goal dari node  $n$

$h\_orthogonal(n)$ : banyaknya langkah lurus yang bisa diambil untuk mencapai goal dari node  $n$ .

$n.x$  &  $n.y$ : beaya koordinat  $x$  &  $y$  pada node  $n$

$goal.x$  : beaya koordinat  $x$  pada node tujuan

$goal.y$  : beaya koordinat  $y$  pada node tujuan

$d1$  : beaya perpindahan node secara orthogonal

$d2$  : beaya perpindahan node secara diagonal

**2.2.1 Jarak Euclid dalam Algoritma A\***

Dalam mengaplikasikan Algoritma A\* pada penelitian ini, fungsi heuristic yang digunakan adalah jarak Euclid. Heuristik ini akan menghitung jarak berdasarkan panjang garis yang dapat ditarik dari dua buah titik. Perhitungannya dapat dituliskan sebagai berikut :

$$h(n) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

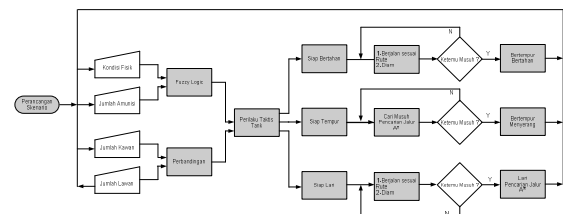
**2.2.2 Logika Fuzzy (Fuzzy Logic)**

Logika Fuzzy bukan merupakan suatu ilmu yang baru, konsep tentang fuzzy itu sendiri sudah ada sejak lama namun ilmu tentang logika fuzzy modern dan metodis baru ditemukan beberapa tahun yang lalu. Logika Fuzzy adalah metodologi pemecahan masalah dengan beribu-ribu aplikasi dalam pengendali yang tersimpan dan pemrosesan informasi. Logika Fuzzy menyediakan cara sederhana untuk menggambarkan kesimpulan pasti dari informasi yang ambigu, samar-samar, atau tidak tepat. Dalam sistem Logika Fuzzy terdapat beberapa tahapan operasional yang meliputi:

1. Fuzzyfikasi
2. Aturan Dasar (*Rule Based*)
3. Komposisi Aturan
4. Defuzzyfikasi

**2.3 Kerangka Pemikiran**

Perancangan sistem yang dipakai secara umum adalah berawal dari permasalahan dimana pengembangan perangkat lunak khususnya untuk *Game Simulator Tank* masih belum optimal dalam penerapan algoritma sistem cerdas. Kerangka pemikiran dalam hal penggunaan algoritma pada pengembangan perangkat lunak secara garis besar konsep desainnya dapat ditunjukkan pada gambar 2.1.

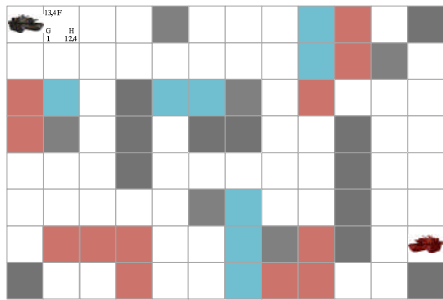


Gambar 2.1: Diagram penerapan algoritma pada *Game Simulator Tank*

**3. Analisis**

**3.1 Analisis pergerakan A\* dengan memperhatikan Kondisi Lingkungan**

Pada kasus ini akan dianalisis penerapan Algoritma A\* pada jalan dengan memperhatikan kondisi lingkungan seperti lumpur, sungai, bangunan dan pohon untuk pencarian jalan terbaik dan/atau waktu tercepat menuju tujuan dengan menghitung jarak langkah dan simpul yang diperiksa. Ada berbagai macam rintangan yang ditemukan dalam sebuah medan peperangan yang digambarkan pada gambar 3.1. Sebagai contoh, pada rancangan peta berbentuk kotak berikut mempunyai berbagai rintangan yang berbeda-beda yang akan dianalisis bagaimana pengaruh pergerakan KBP dalam memilih jalur yang akan dilewati. Peta berikut mempunyai ordo 8 x 12, dimana pergerakan sumbu X adalah 8 dan pergerakan sumbu Y adalah 12. Untuk lebih jelasnya, gambaran peta yang terdapat beberapa macam lingkungan dapat dilihat pada gambar 3.1 berikut:



Gambar 3.1: Peta dengan berbagai kondisi lingkungan

- Kotak : Tanpa Heuristic = 1
  - Kotak : Lumpur Heuristic = 1,5
  - Kotak : Sungai Heuristic = 2
  - Kotak : Bangunan, Tidak dapat dilewati
- Keterangan gambar:



Pada gambar 3.1 diatas, kotak berisi tank warna hitam di sebelah kiri atas adalah titik awal, dan kotak berisi tank warna merah di sebelah kanan bawah adalah titik tujuan. Angka-angka dalam kotak masing-masing adalah nilai f (kiri atas), g (kiri bawah) dan h (kanan bawah). Untuk pergerakan horizontal dan vertikal, beayanya (cost) adalah 1, sedangkan untuk pergerakan

diagonal beayanya adalah 1,4 yang berasal dari  $\sqrt{2}$  dengan pembulatan 1 angka di belakang desimal. Pada kotak di sebelah kanan simpul awal, g bernilai 1 dan h bernilai 12,4. Perhitungannya cukup sederhana, yaitu g bernilai 1 sebab dari simpul awal ke kotak tersebut hanya perlu bergerak horizontal satu kali. H bernilai 12,4 didapat dari banyaknya langkah yang diperlukan untuk mencapai tujuan dari kotak tersebut. F bernilai 13,4 adalah merupakan jumlah dari g(1) dan h(12,4). Selanjutnya, algoritma ini akan memeriksa kotak mana yang bernilai f-nya terkecil dan mengembangkan jalan di sekitar kotak tersebut.

Pada iterasi pertama ada tiga jalur yang ditemukan, yaitu jalur pada koordinat (1,7), (2,7) dan (2,8). Disini akan dihitung nilai f(n), g(n) dan h(n) untuk masing-masing jalur. Namun disini hanya akan dibahas cukup satu jalur saja, sedangkan perhitungan jalur lainnya tidak dibahas karena dianggap penghitungannya similar.

**Koordinat (2, 8)**

Mekanisme perhitungan h(n):

$$h_{diagonal}(n) = \min(\text{abs}(n.x-\text{tujuan}.x), \text{abs}(n.y-\text{tujuan}.y)) = \min(\text{abs}(2-12), \text{abs}(8-2)) = \min(10, 6) = 6$$

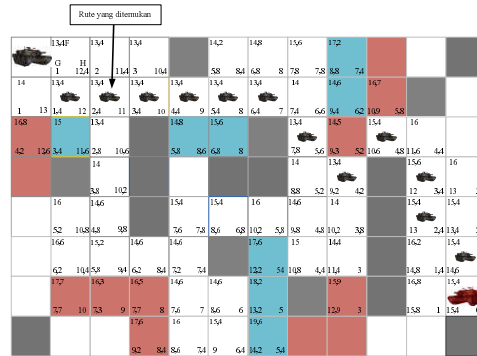
$$h_{orthogonal}(n) = (\text{abs}(n.x-\text{tujuan}.x) + \text{abs}(n.y-\text{tujuan}.y)) = \text{abs}(2-12) + \text{abs}(8-2) = 10 + 6 = 16$$

$$h(n) = d2 * h_{diagonal}(n) + d1 * (h_{orthogonal}(n) - 2 * h_{diagonal}(n)) = 1,4 * 6 + 1 * (16 - 2 * 6) = 8,4 + 4 = 12,4$$

Mekanisme Perhitungan f(n):

$$f(n) = g(n) + h(n) = 1 + 12,4 = 13,4$$

Dengan cara seperti ini, setelah melakukan iterasi ke tigapuluh dua proses pencarian jalur selesai karena telah ditemukan jalur menuju ke target dengan biaya 15,4. Karena jalur paling optimal dari awal (1, 8) dan tujuan (12, 2) sudah ditemukan, KBP akan bergerak menuju tujuan seperti terlihat pada gambar 3.2.



Gambar 3.2 Proses hasil akhir pencarian A\*

**3.2 Analisis dan Perancangan Status Kesehatan Tank dengan Logika Fuzzy**

Status Kesehatan Tank (SKT) merupakan cara untuk memantau status berdasarkan kesiapan tank untuk terus melakukan peperangan, khususnya yang berkaitan dengan kondisi fisik tank dan jumlah amunisi yang masih tersedia. Penggunaan Status kesehatan tank hanya berlaku untuk tank dan tidak untuk kendaraan tempur lain seperti truk pengangkut pasukan, ranpur, dan kendaraan selain tank lainnya. Kategori status kesehatan tank bisa dilihat pada tabel 3.1 berikut.

Tabel 3.1: Kategori Status Kesehatan Tank

Nilai SKT	Kategori	Status Kesehatan Tank
80 - 100	Kesehatan tank sangat baik	Baik
60 - 80	Kesehatan tank cukup baik	Cukup Baik
40 - 60	Kesehatan tank masih standar/normal	Normal
20 - 40	Kesehatan tank rusak tingkat ringan	Agak Rusak
1 - 20	Kesehatan tank rusak tingkat berat	Rusak

Untuk mendapatkan *output* berupa Status Kesehatan Tank, diperlukan empat tahapan sebagai berikut:

**3.2.1 Pembentukan Himpunan Fuzzy**

Pada Metode Mamdani, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan fuzzy. Dalam penentuan Status Kesehatan Tank (SKT), variabel *input* dibagi menjadi dua yaitu variabel Kondisi Fisik dan Jumlah Amunisi. Himpunan fuzzy beserta fungsi keanggotaan dari variabel Kondisi Fisik, Jumlah Amunisi, dan hasil keluaran Status Kesehatan Tank direpresentasikan sebagai berikut:

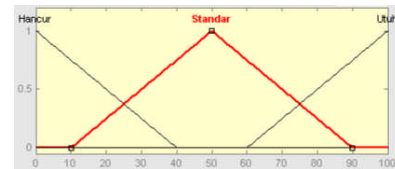
**a. Himpunan Fuzzy Variabel Kondisi Fisik**

Pada variabel Kondisi Fisik didefinisikan tiga himpunan fuzzy, yaitu HANCUR, STANDAR, dan UTUH. Untuk merepresentasikan variabel Kondisi Fisik digunakan bentuk kurva bahu kiri untuk himpunan fuzzy HANCUR, bentuk kurva segitiga untuk himpunan fuzzy STANDAR, dan bentuk kurva bahu kanan untuk himpunan fuzzy UTUH. Gambar himpunan fuzzy untuk variabel Kondisi Fisik ditunjukkan pada gambar 3.3

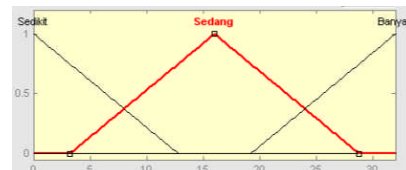
**b. Himpunan Fuzzy Variabel Jumlah Amunisi**

Pada variabel Jumlah Amunisi didefinisikan tiga himpunan fuzzy, yaitu SEDIKIT, SEDANG, dan BANYAK. Untuk merepresentasikan variabel Jumlah Amunisi digunakan bentuk kurva bahu kiri

untuk himpunan fuzzy SEDIKIT, bentuk kurva segitiga untuk himpunan fuzzy SEDANG, dan bentuk kurva bahu kanan untuk himpunan fuzzy BANYAK. Representasi himpunan fuzzy untuk variabel Jumlah Amunisi ditunjukkan pada gambar 3. 4.

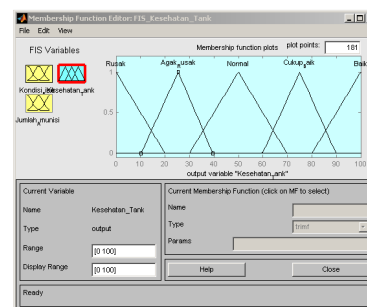


Gambar 3.2 : Himpunan fuzzy Kondisi Fisik



Gambar 3.3: Himpunan fuzzy: Jumlah Amunisi

c. Himpunan Fuzzy Variabel Status Kesehatan Tank Pada variabel Status Kesehatan Tank didefinisikan lima himpunan fuzzy, yaitu RUSAK, AGAK RUSAK, NORMAL, CUKUP BAIK dan BAIK. Untuk merepresentasikan variabel Status Kesehatan Tank digunakan bentuk kurva bahu kiri untuk himpunan fuzzy RUSAK, bentuk kurva segitiga untuk himpunan fuzzy AGAK RUSAK, NORMAL dan CUKUP, dan bentuk kurva bahu kanan untuk himpunan fuzzy BAIK. Representasi himpunan fuzzy untuk variabel Status Kesehatan Tank ditunjukkan pada gambar 3.5.



Gambar 3.4: Representasi Variable Status Kesehatan Tank

Setelah pembentukan fungsi keanggotaan pada masing-masing variabel, *input* yang berupa nilai crisp akan diubah ke dalam *fuzzy input* yaitu dengan menentukan derajat keanggotaan nilai *input* pada sebuah himpunan fuzzy, proses ini disebut fuzzyfikasi.

**3.2.2 Aplikasi Implikasi Fungsi**

Setelah pembentukan himpunan fuzzy, maka dilakukan pembentukan aturan fuzzy. Aturan-aturan dibentuk untuk menyatakan relasi antara *input* dan *output*. Tiap aturan merupakan suatu implikasi.

Operator yang digunakan untuk menghubungkan antara dua *input* adalah operator AND, dan yang memetakan antara *input-output* adalah IF-THEN. Proposisi yang mengikuti IF disebut anteseden, sedangkan proposisi yang mengikuti THEN disebut konsekuen. Berdasarkan kategori dalam SKT, maka dapat dibentuk aturan-aturan seperti terlihat pada Tabel 3.2 berikut.

Tabel 3.2: Aturan-aturan dalam penentuan Status Kesehatan Tank

		Kondisi Fisik		
		Rusak	Standar	Utuh
Jumlah Amunisi	Sedikit	Rusak	Agak Rusak	Normal
	Sedang	Agak Rusak	Normal	Cukup Baik
	Banyak	Normal	Cukup Baik	Baik

Setelah penentuan fungsi keanggotaan variabel, maka dilakukan pembentukan aturan logika fuzzy. Berdasarkan data yang ada, dapat dibentuk aturan-aturan sebagai berikut :

[R1]: Jika Kondisi Fisik adalah HANCUR dan Jumlah Amunisi adalah SEDIKIT maka Status Kesehatan Tank adalah RUSAK.

[R2]: Jika Kondisi Fisik adalah HANCUR dan Jumlah Amunisi adalah SEDANG maka Status Kesehatan Tank adalah AGAK RUSAK.

[R3]: Jika Kondisi Fisik adalah HANCUR dan Jumlah Amunisi adalah BANYAK maka Status Kesehatan Tank adalah NORMAL.

[R4]: Jika Kondisi Fisik adalah STANDAR dan Jumlah Amunisi adalah SEDIKIT maka Status Kesehatan Tank adalah AGAK RUSAK.

[R5]: Jika Kondisi Fisik adalah STANDAR dan Jumlah Amunisi adalah SEDANG maka Status Kesehatan Tank adalah NORMAL.

[R6]: Jika Kondisi Fisik adalah STANDAR dan Jumlah Amunisi adalah BANYAK maka Status Kesehatan Tank adalah CUKUP BAIK.

[R7]: Jika Kondisi Fisik adalah UTUH dan Jumlah Amunisi adalah SEDIKIT maka Status Kesehatan Tank adalah NORMAL.

[R8]: Jika Kondisi Fisik adalah UTUH dan Jumlah Amunisi adalah SEDANG maka Status Kesehatan Tank adalah CUKUP BAIK.

[R9]: Jika Kondisi Fisik adalah UTUH dan Jumlah Amunisi adalah BANYAK maka Status Kesehatan Tank adalah BAIK.

Setelah aturan dibentuk, maka dilakukan aplikasi fungsi implikasi. Pada Metode Mamdani, fungsi implikasi yang digunakan adalah MIN, yang berarti tingkat keanggotaan yang didapat sebagai konsekuen dari proses ini adalah nilai minimum dari variabel Kondisi Fisik dan Jumlah Amunisi. Sehingga didapatkan daerah fuzzy pada variabel Status Kesehatan Tank untuk masing-masing aturan.

### 3.2.3 Komposisi Aturan

Pada metode Mamdani, komposisi antar fungsi implikasi menggunakan fungsi MAX yaitu dengan cara mengambil nilai maksimum dari *output* aturan kemudian menggabungkan daerah fuzzy dari masing-masing aturan dengan operator OR.

$$\mu_{sf}[x]=$$

$$\max(\mu_{kf1}[x], \mu_{kf2}[x], \mu_{kf3}[x], \mu_{kf4}[x], \mu_{kf5}[x], \mu_{kf6}[x], \mu_{kf7}[x], \mu_{kf8}[x], \mu_{kf9}[x])$$

Keterangan:

$\mu_{sf}[x]$  : nilai keanggotaan solusi fuzzy sampai aturan ke-*i*;  $\mu_{kfi}[x]$

$\mu_{kf}[x]$  : nilai keanggotaan konsekuen fuzzy setiap aturan ke-*i*, dimana  $i = 1, 2, \dots, 6$ .

### 3.2.4 Penegasan (Defuzzyfikasi)

*Input* dari proses defuzzyfikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan fuzzy, sedangkan *output* yang dihasilkan merupakan suatu bilangan tegas (*crisp*) pada domain himpunan fuzzy tersebut. Sehingga jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*. Defuzzyfikasi yang digunakan dalam menentukan nilai SKT adalah dengan metode *centroid*. Pada metode ini, solusi *crisp* diperoleh dengan cara mengambil titik pusat ( $Z_0$ ) daerah fuzzy. Secara umum dirumuskan:

$$Z_0 = \frac{\int_a^b Z \mu(z) dz}{\int_a^b \mu(z) dz}$$

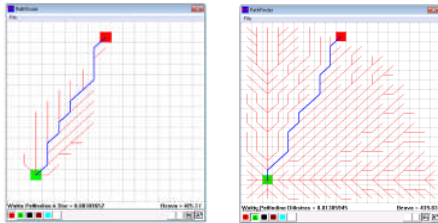
untuk domain kontinyu, dengan  $Z_0$  adalah nilai hasil defuzzyfikasi dan  $\mu(z)$  adalah derajat keanggotaan titik tersebut, sedangkan  $Z$  adalah nilai domain ke-*i*.

## 4. Hasil dan Pembahasan

### 4.1 Pengujian Algoritma A\* untuk Pencarian Jalur (Pathfinding)

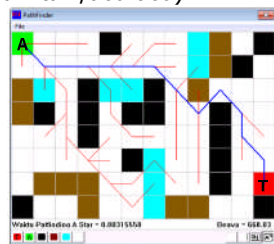
Pengujian yang akan digunakan untuk menemukan jalur yaitu dengan menggunakan penghalang dan tanpa penghalang. Pertama setelah pengguna menempatkan posisi awal dan posisi tujuan yang diinginkan, dan sudah menempatkan penghalang pada medan yang digunakan maka pencarian jalur dapat dilakukan dengan memilih submenu A\* pada

menu Algoritma. Hasil jalan yang ditemukan dengan waktu pencarian dapat dilihat pada gambar 4.1a.



Gambar 4.1: a. Jalur algoritma A\*  
b. Jalur algoritma Dijkstra

Pengujian pencarian jalur dengan algoritma A\* dapat dicapai dengan waktu 3.896 ms dan biayanya adalah 425.17. Sebagai perbandingan pencarian jalur dengan algoritma Dijkstra dicapai dengan waktu 13.059 ms dan biayanya adalah 439.83, dimana hasil tersebut dapat dilihat pada gambar 4.1b. Gambar 4.2 adalah hasil pencarian jalur dengan algoritma A\* dengan 3 macam penghalang, yaitu lumpur (kotak warna coklat), genangan air/sungai (kotak warna hijau kebiruan) dan bangunan/pohon/kendaraan lain (kotak warna hitam/abu-abu).



Gambar 4.2: Jalur A\* dengan berbagai macam kondisi lingkungan  
Perbandingan hasil pengujian performansi untuk algoritma A\* dan Dijkstra dengan beberapa variasi titik awal dan titik tujuan pada peta dengan ordo (19,19) dapat dilihat pada tabel 4.1.

Tabel 4.1: Perbandingan hasil pengujian antara A\* dan Dijkstra

No.	Posisi Awal	Posisi Tujuan	A*		Dijkstra		Kondisi
			Waktu	Beaya	Waktu	Beaya	
1	(0,0)	(18,18)	1.468	638.36	23.257	638.36	Halangan
2	(0,0)	(0,18)	0.937	431.05	14.909	14.909	Tidak ada
3	(0,0)	(18,0)	0.966	470.84	187.58	187.58	Tidak ada
4	(1,4)	(17,15)	6.121	520.90	20.647	20.647	Tidak ada
5	(1,4)	(17,15)	6.570	554.15	16.741	16.741	Tidak ada
6	(0,0)	(18,18)	9.211	711.56	17.748	17.748	Ada
7	(0,0)	(0,18)	4.849	546.22	20.423	20.423	Ada

Dengan hasil uji yang telah dilakukan dapat dilihat hasil pencarian dengan menggunakan algoritma A\* yaitu waktu pencarian, simpul yang diperiksa dan jarak rute yang dihasilkan. Pada algoritma A\* jarak rute yang dihasilkan pendek, dengan waktu yang jauh lebih cepat dibandingkan Dijkstra.

#### 4.1.2 Metode Logika Fuzzy dalam menentukan keputusan perilaku perang

Proses uji coba untuk metode Logika Fuzzy dilakukan dengan memasukkan sejumlah data kedalam program yang telah dibuat. Hasil keputusan diambil dari perhitungan fuzzy kondisi fisik, jumlah amunisi, jumlah tank kawan dan jumlah tank lawan. Masing-masing kondisi merupakan hasil perhitungan fuzzy dari data yang bersifat dinamis (bisa berubah setiap saat).

##### 4.1.2.1 Status Kesehatan Tank

Hasil dari algoritma pembelajaran menggunakan perangkat lunak untuk pengujian penggunaan metode logika fuzzy sudah dapat menghasilkan sebuah keputusan berdasarkan data *input* yang dimasukkan. Hasil program setelah *input* dimasukkan untuk Status Kesehatan Tank dapat dilihat pada *output* pada tabel 4.2 dibawah ini.

Tabel 3.2: Hasil pengujian Status Kesehatan Tank

Kondisi Fisik	Jumlah Amunisi	Status Kesehatan Tank
Utuh	Sedang	Baik
Utuh	Sedikit	Agak Rusak
Sedang	Sedang	Normal
Sedang	Banyak	Cukup Baik
Hancur	Cukup	Rusak
Hancur	Banyak	Agak Rusak
Utuh	Banyak	Baik

##### 4.1.3 Penentuan Status Perilaku Perang

Dalam menentukan Status Perilaku Perang kapan KBP akan siap tempur, bertahan atau bahkan melarikan diri (menghindar) didapat dari pengolahan Status Kesehatan Tank dengan perbandingan jumlah kawan dan lawan, bila perbandingan jumlah kawan dengan lawan lebih besar atau sama dengan 1.5, maka Status Kesehatan Tanknya semua KBP Status Perilaku Perangnya adalah TEMPUR. Untuk lebih jelasnya kondisi Status Perilaku Perang KBP yang diolah berdasarkan Status Kesehatan Tank dan perbandingan jumlah kawan dan lawan dapat dilihat pada tabel 4.3.

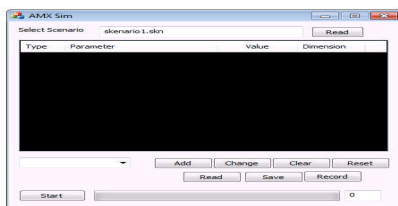
Tabel 4.3: Kondisi Status Perilaku Perang

Perbandingan Jumlah Kawan-Lawan	Status Kesehatan Tank	Status Perilaku Perang
$\geq 2:1$	Semua kondisi	Tempur
1:1 - 1.5:1	Cukup Baik, Baik	Tempur
	Normal, Agak Rusak	Bertahan
	Rusak	Lari
$\leq 1:1$	Cukup Baik, Baik	Bertahan
	Normal, Agak Rusak, Rusak	Lari

**4.2 Pengujian dan Implementasi pada Game Simulator Tank**

Setelah dilakukan pengujian terhadap algoritma-algoritma yang digunakan, yaitu A\* dan logika fuzzy, selanjutnya algoritma tersebut diimplementasikan pada aplikasi perangkat lunak *Game Simulator Tank (GST)*.

Untuk menjalankan GST ini dilakukan dengan menjalankan file eksekusi yang bernama *ATS\_IGd.exe* terlebih dahulu, lalu kemudian disusul dengan menjalankan file eksekusi *Host\_IGd.exe*. Pada layar akan terlihat tampilan antarmuka sebagaimana terlihat pada gambar 4.3. Untuk memulai melakukan permainan *Game Simulator Tank* tekan tombol *Start*, namun sebelum itu pengguna harus memilih skenario terlebih dahulu tentang medan latihan yang dipakai, macam dan jumlah kendaraan yang akan digunakan, lokasi kendaraan, dan sebagainya. Pemilihan skenario dilakukan dengan memasukkan nama skenario yang akan dipakai setelah itu menekan tombol *read*.



Gambar 4.3: Tampilan awal *Game Simulator Tank*

Setelah beberapa waktu berjalan, perjalanan yang dilalui oleh tank dapat dilihat pada gambar 4.4.



Gambar 4: Jalur yang dilalui tank dengan algoritms A\*

Sebagai perbandingan bila tank tidak menggunakan algoritma A\* untuk pencarian jalurnya, maka Tank berjalan sesuai dengan rute yang telah ditetapkan oleh program seperti terlihat pada gambar 4.5.



Gambar 5: Jalur yang dilalui tank tanpa algoritms A\*

**5. Penutup**

**5.1 Kesimpulan**

Beberapa kesimpulan yang dapat diambil dari proses pembuatan Tesis ini adalah sebagai berikut:

1. Aplikasi Game Simulator Tank ini menggunakan situasi yang menyerupai dengan keadaan sebenarnya, seperti medan latihan perang diambil dari lokasi Riil, yaitu di Padalaran dan jalur jalan dan kontur medan yang ada menyerupai Kondisi di lapangan.
2. Pemilihan jalur sudah memperhitungkan kondisi jalan riil dengan memperhitungkan apakah jalan tersebut berpenghalang tidak dapat dilewati atau dengan nilai heuristic tertentu sehingga kalau dilewati akan memperlambat jalan.
3. Penerapan strategi perang semakin akurat, dimana respon perilaku karakter bukan pemain akan selalu memperhitungkan kondisi kesehatan kendaraan waktu itu.

**5.2 Saran**

Penerapan algoritma A\* dan metode Logika Fuzzy sudah cukup baik dalam membuat sebuah media pembelajaran berbasis simulasi berupa *Game Simulator Tank* untuk memenuhi sasaran sesuai dengan peruntukannya, dimana semua proses bisa dilakukan menyerupai dengan kondisi riil di lapangan. Namun hal yang sudah ada akan bisa menjadi lebih hidup dan lebih baik lagi bila juga menggunakan beberapa metode lain bersamaan dengan penerapan kedua metode yang menjadi obyek penelitian Tesis ini. Metode tersebut adalah penentuan gerak mengikuti pergerakan lawan secara lebih baik yaitu menggunakan *Finite State Machine* dan metode *Reinforcement Learning*.

**Daftar Pustaka**

[1] Xiao Cui and Hao Shi Jurnal : *A\*-based Pathfinding in Modern Computer Games*. School of Engineering and Science, Victoria University, Melbourne, Australia, Januari 2011.

- 
- [2] Yifan Li, et al. Jurnal : *Fuzzy Logic in Agent-Based Game Design*. Department of Electrical and Computer Engineering University of Alberta Edmonton, AB T6G 2V4, Canada, 2004.
- [3] Millington and John Funge - 2009: *Artificial Intelligence for Games Second Edition*. Morgan Kaufmann Publishers is an imprint of Elsevier. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, 2009.
- [4] T. Sutojo, S.Si.,M.Kom., Edy Mulyanto, S.Si.,M.Kom., Dr. Vincent Suhartono – 2011: [Kecerdasan Buatan](#). Andi Publisher, Jogjakarta.
- [5] Yunifa Miftachul Arif, et al. Jurnal : *Strategi Menyerang pada Game FPS Menggunakan Hierarchy Finite State Machine dan Logika Fuzzy*. Pasca Sarjana Teknik Elektro ITS, Surabaya 60111
- [6] Bambang Sridadi, 2009. *Pemodelan dan Simulasi Sistem : Teori, Aplikasi dan Contoh Program dalam Bahasa C*. Informatika, Bandung,
- [7] Yung-Ping Fang, et al. Jurnal: *Applying Reinforcement Learning for the AI in a Tank-Battle Game*. Department of Information Management National University of Kaohsiung, No. 700, Kaohsiung University Road, Kaohsiung, 811 Taiwan, Desember 2010.
- [8] H<http://www.delta3d.org>. diakses 7 Juni 2013.
- [9] Adipranata, Rudy, Andreas Handojo & Happy Setiawan (2007). *Aplikasi Pencari Rute Optimum Pada Peta*. Universitas Kristen Petra, Surabaya