

PERANCANGAN SISTEM INFERENSI DALAM AI GAME DENGAN LOGIKA FUZZY

Kevin Winata^{#1}, Dr. Ir. Rinaldi Munir, M.T.^{#2}

[#]School of Electrical Engineering and Informatics, Institute Technology of Bandung,
10th Ganeca Street, Bandung, Indonesia

¹kevinwinata@gmail.com

²rinaldi@informatika.org

Abstrak— Penelitian ini membahas mengenai pemanfaatan logika fuzzy pada AI, yaitu pada *bot* dalam *video game*, yaitu *Hedgewars*. Perancangan sistem inferensi fuzzy dilakukan dengan mendefinisikan variabel, himpunan, fungsi keanggotaan dan kaidah fuzzy yang sesuai dengan cara bermain *Hedgewars*. Implementasi dilakukan dengan menambahkan unit yang berisi kode implementasi sistem inferensi fuzzy ke kode program *Hedgewars*. Sistem inferensi ini digunakan untuk memodifikasi sistem skor yang dipakai untuk mengambil keputusan. Hasil dari implementasi tersebut adalah sebuah fitur pada *Hedgewars* untuk mengubah perilaku *bot* dengan sistem inferensi fuzzy tanpa mengganti kode program.

Kata Kunci— logika fuzzy, *bot*, *video game*, *Hedgewars*, sistem inferensi

I. PENDAHULUAN

Video game merupakan *software* permainan yang melibatkan pengguna untuk melakukan interaksi tertentu dan memberikan umpan balik berupa grafik, suara atau hal lainnya. Dalam pengembangannya, *video game* banyak menggunakan *artificial intelligence* (AI). Dengan adanya AI, *video game* tersebut bisa menjadi lebih interaktif sehingga lebih menarik untuk dimainkan. AI pada *video game* dapat muncul dalam berbagai bentuk, mulai dari hantu yang mengejar pemain dalam *video game* Pac-Man hingga *bot* dalam *video game* *Unreal Tournament* [1]. Dalam penelitian ini, penggunaan AI yang diperdalam adalah sebagai *bot*.

Bot masih banyak menggunakan logika tegas (*crisp*), atau disebut juga logika biner dalam pengambilan keputusan. Penggunaan logika tersebut terdapat dalam *finite state machine* atau *behavior tree*. Perluasan dari logika tegas adalah logika fuzzy. Jika pada logika tegas nilai kebenaran hanya berupa 0 dan 1, maka pada logika fuzzy nilai kebenaran terletak di dalam selang [0, 1]. Logika fuzzy lebih jarang digunakan dalam *game* daripada logika tegas, karena kerumitan implementasi logika fuzzy yang lebih besar [2].

Pengambilan keputusan dengan menggunakan logika tegas memiliki beberapa kelemahan. Bila terjadi perubahan keputusan, perubahan tersebut akan terjadi secara tiba-tiba dan tidak alami, misalnya pada kasus mobil yang dikontrol *bot*, mobil tersebut akan berbelok atau berhenti tiba-tiba. Selain itu, keputusan yang diambil juga menjadi lebih mudah diprediksi,

karena probabilitasnya diambil dari semua aturan yang ada, tidak hanya yang paling baik saja [1].

Dengan menggunakan logika fuzzy, keputusan yang diambil menjadi lebih bervariasi, dan keluaran dapat diberikan secara gradual sehingga perubahan keputusan akan terlihat lebih alami. Logika fuzzy juga dapat dibangun dari pengetahuan ahli, sehingga tidak memerlukan perhitungan yang kompleks, memungkinkan untuk peningkatan kinerja. Penggunaan logika fuzzy pada *video game* akan membuat *bot* yang dimilikinya terasa lebih fleksibel, sulit diprediksi, dan lebih menyerupai manusia.

Dalam tulisan ini akan dibahas mengenai aplikasi logika fuzzy dalam *bot* pada sebuah *video game*. Masalah utama yang ingin diselesaikan dalam tulisan ini adalah bagaimana membuat sistem inferensi fuzzy sesuai dengan rancangan keputusan yang perlu diambil oleh *bot*, cara mengembangkan *bot* pada *video game* dengan menggunakan sistem inferensi yang telah dikembangkan, dan menguji tingkat keberhasilan *bot* tersebut dalam mengambil keputusan yang tepat dan sesuai dibanding *bot* asli yang sudah ada sebelumnya.

II. DASAR TEORI

A. Logika Fuzzy

Logika fuzzy modern sendiri pertama kali diperkenalkan oleh Lotfi A. Zadeh, seorang profesor sains komputer di University of California di Berkeley, pada tahun 1965.

1) Himpunan Fuzzy

Himpunan fuzzy memperbolehkan keanggotaan parsial, dimana derajat keanggotaan $\mu_A(x)$ memiliki nilai dalam interval 0 dan 1. Himpunan fuzzy merepresentasikan variabel linguistik dalam bahasa alami, seperti cepat, lambat, kecil, besar, tinggi, pendek, dan sebagainya. Sebuah elemen bisa menjadi anggota dari beberapa himpunan fuzzy sekaligus.

2) Fungsi keanggotaan

Fungsi keanggotaan adalah fungsi yang menentukan derajat keanggotaan $\mu(x)$ ketika variabel fuzzy x memiliki nilai tertentu. Ada beberapa bentuk fungsi keanggotaan yang umum digunakan, yaitu linier, segitiga, trapesium, atau kurva.

3) Operasi Logika fuzzy

Operasi logika fuzzy sejalan dengan operasi logika biasa. Dalam logika fuzzy, kebenaran dari suatu pernyataan adalah masalah derajat. Untuk mendefinisikan operator logika fuzzy, kita harus mencari operator yang mempertahankan hasil dari operator AND, OR dan NOT pada logika biasa. Operator AND diwakilkan oleh operator min, OR oleh operator max, dan NOT oleh operator komplemen.

4) Kaidah Fuzzy

Kaidah fuzzy, atau disebut juga sebagai implikasi fuzzy, menjelaskan relasi antara himpunan fuzzy masukan dengan himpunan fuzzy keluaran. Sebuah kaidah fuzzy tunggal memiliki bentuk “if x is A, then y is B”. Bagian if dari kaidah tersebut, “x is A”, disebut juga sebagai antesenden atau premis, sedangkan bagian then, “y is B”, disebut juga sebagai konsekwen atau konklusi.

5) Sistem Inferensi Fuzzy

Sistem inferensi fuzzy pada dasarnya mendefinisikan pemetaan nonlinear dari vektor data masukan ke keluaran skalar, dengan menggunakan kaidah fuzzy. Hal ini digunakan untuk penarikan kesimpulan. Proses pemetaan tersebut melibatkan penginterpretasian kaidah fuzzy, yaitu fuzzifikasi, operasi logika fuzzy, dan implikasi, kemudian dilanjutkan dengan agregasi himpunan keluaran dan defuzzifikasi. Model sistem inferensi yang digunakan pada tulisan ini adalah Mamdani.

Fuzzifikasi adalah proses untuk memetakan nilai tegas ke dalam himpunan fuzzy dan menentukan derajat keanggotannya. Jika terdapat operator logika dalam antesenden, maka akan digunakan operasi logika fuzzy. Implikasi adalah proses untuk mendapatkan derajat keluaran dari kaidah if-then. Metode yang digunakan pada implikasi Mamdani adalah dengan menggunakan operator min. Agregasi menyatukan semua keluaran dari kaidah fuzzy menjadi himpunan fuzzy tunggal. Defuzzifikasi dilakukan untuk mendapatkan nilai keluaran tegas dari sistem inferensi. Cara yang paling umum digunakan adalah metode centroid, yaitu menghitung titik berat (center of gravity atau centroid) dan menggunakan nilai tersebut sebagai keluaran.

B. Penggunaan AI pada Video Game

1) Video Game

Menurut [3], definisi *game* adalah sebuah tipe aktivitas bermain yang dilakukan dalam konteks kenyataan yang pura-pura, yang di dalamnya pemain berusaha mencapai tujuan nontrivial tertentu dengan beraksi sesuai dengan aturan. *Video game* adalah bagian dari *game* secara keseluruhan. *Video game* adalah sebuah *game* yang berupa perangkat lunak yang berjalan di atas perangkat keras komputer. Perangkat komputer membuat *video game* dapat meminjam elemen dari berbagai jenis hiburan lainnya, seperti buku, film atau musik.

2) Penggunaan AI pada Video Game

Terdapat bidang keilmuan khusus yang mempelajari penggunaan AI dalam *video game*, yaitu *Game AI*. Menurut [2], *Game AI* merupakan salah satu cabang dari artificial intelligence dasar. Berbeda dengan AI biasa, *Game AI* tidak terlalu memperhatikan kepintaran secara nyata, tetapi hanya berupa ilusi saja. Hal ini dikarenakan sebuah *video game* tidak

memerlukan AI yang benar-benar pintar dan menyerupai manusia untuk sukses dalam menjalankan tugasnya. Perbedaan yang lain adalah fakta bahwa *video game* memerlukan AI yang *real-time*, sehingga perhitungan yang dilakukan tidak boleh lama. Berdasarkan perbedaan ini, bidang *Game AI* terus berkembang secara berbeda namun tetap sejalan dengan AI konvensional, meminjam teknik dan ide dari AI namun tetap mencari pendekatan yang praktis.

Salah satu penggunaan AI pada *video game* adalah *bot*. *Bot* dalam hal ini mengacu kepada agen pintar dalam sebuah *video game* yang dibangun dengan tujuan untuk membantu manusia dalam mengerjakan sebuah tugas yang spesifik, sehingga tugas tersebut tidak perlu lagi dikerjakan manusia. Referensi [4] membagi *bot* dalam *video game* menjadi dua kategori berdasarkan tujuan dan perilakunya. Kategori pertama adalah *bot* yang bertindak sebagai *non-player characters* (NPC), dan yang kedua adalah *bot* yang bertindak sebagai pengganti pemain manusia.

3) Penerapan Logika Fuzzy pada Video Game

Berdasarkan [1], logika *fuzzy* bisa diterapkan pada *game* dalam berbagai cara. Yang pertama adalah penerapannya dalam hal yang terkait kontrol pada *game*. Logika *fuzzy* sudah banyak diaplikasikan dalam dunia nyata terkait masalah kontrol, seperti kereta api dan *air conditioning*, sebagaimana telah dijelaskan sebelumnya. *Game* sendiri juga memberikan banyak kesempatan untuk menggunakan *fuzzy* pada sistem kontrol.

Contoh yang paling umum adalah navigasi, misalnya navigasi pada pesawat atau kendaraan yang dikontrol oleh *bot*. Atau lebih jauh lagi, pada kasus kejar-mengejar antar karakter atau objek juga bisa diterapkan sistem kontrol *fuzzy*. Metode deterministik biasa sebenarnya bisa menyelesaikan masalah tersebut, tetapi akan memberikan hasil yang lebih tidak alami. Pada masalah navigasi kejar-mengejar, objek yang digerakkan akan berubah arah secara tiba-tiba dan tidak mulus. Dengan menggunakan *fuzzy*, kontrollernya dapat diberi masukan seperti “Target berada di kiri jauh” atau “Target berada agak kanan di depan”, sehingga controller tersebut bisa menghitung arah dan sudut gerakannya dengan lebih mulus dan alami.

Penerapan selanjutnya adalah pengambilan keputusan. Logika *fuzzy* akan berguna dalam memodelkan komputer dengan pengetahuan yang terbatas dan membuatnya lebih bervariasi serta tidak mudah diprediksi. Hal ini akan diteliti lebih jauh dalam tulisan ini.

III. ANALISIS DAN PERANCANGAN

A. Hedgewars

Video game yang dipilih untuk dijadikan bahan pengembangan AI dengan logika *fuzzy* adalah *Hedgewars*. *Hedgewars* pertama kali diluncurkan pada tahun 2006, dan dapat dijalankan pada berbagai platform dan sistem operasi.

Hedgewars dapat dimainkan dengan memilih sebuah pertandingan. Tiap pertandingan dimainkan oleh 2-4 tim yang terdiri dari beberapa landak. Setiap tim dapat dikendalikan oleh pemain manusia atau komputer. Setiap landak dalam suatu tim akan berusaha mengalahkan landak pihak lawan dengan

menggunakan berbagai senjata. Ada 3 cara yang dapat dilakukan untuk mengalahkan landak lain, yaitu menghabiskan nyawa, menenggelamkannya ke air atau mengeluarkan landak tersebut dari arena pertandingan.



Gbr. 1. Screenshot Video Game Hedgewars

Masing-masing pemain diberikan giliran secara teratur. Pada gilirannya, pemain diberi kontrol atas salah satu landak dalam batas waktu tertentu. Landak tersebut dapat bergerak ke kiri atau ke kanan, melompat, atau menyerang dengan menggunakan senjata sebelum gilirannya berakhir. Landak hanya dapat menyerang satu kali dalam satu giliran. Senjata yang dapat digunakan muncul dalam banyak variasi, masing-masing dengan parameter amunisi, *damage* (seberapa banyak nilai nyawa yang berkurang dari karakter yang terkena serangan), jarak kena, dan kecepatan proyektil tersendiri.

B. Bot Hedgewars

Proses kerja *bot* ketika mendapatkan giliran dapat dibagi menjadi dua fase utama, yaitu fase persiapan dan fase penembakan.

1) Fase persiapan

Pada fase ini, *bot* akan mengisi daftar target untuk diserang. Target dapat berupa landak, baik pihak lawan maupun kawan, ranjau atau tong. Setelah itu, *bot* perlu memilih salah satu senjata untuk ditembakkan pada target. Setiap pasang target dan senjata tersebut diberi skor masing-masing dengan cara perhitungan yang berbeda-beda. Sepasang target dan senjata yang memiliki skor tertinggi akan dipilih untuk dieksekusi pada fase berikutnya.

2) Fase penembakan

Pada fase ini akan ditentukan aksi-aksi apa saja yang perlu dilakukan dimulai dari selesainya memilih senjata hingga penembakan senjata tersebut. Berbeda dengan fase sebelumnya, aksi yang dipilih tidak berdasarkan pada skor yang terbesar. *Bot* melakukan aksi-aksi yang diperlukan secara prosedural.

Masalah yang ditemukan pada *bot* asli Hedgewars berdasarkan pengamatan yang dilakukan adalah sebagai berikut.

- Pengukuran yang terlalu akurat

- Aksi tidak bervariasi dan mudah diprediksi
- Tindakan yang konyol
- Tidak adanya rasa kewaspadaan
- Kurang agresif

C. Analisis Solusi

Solusi yang diusulkan untuk permasalahan tersebut adalah menambahkan sistem inferensi *fuzzy* pada kode *bot Hedgewars*. Tipe sistem inferensi yang digunakan adalah sistem inferensi Mamdani. Logika *fuzzy* digunakan untuk memberikan fleksibilitas dan perilaku yang menyerupai manusia pada tiap aksi yang dilakukan *bot*, bukan untuk membuat *bot* tak terkalahkan oleh pemain lain.

Penambahan sistem inferensi *fuzzy* akan dibatasi pada fase persiapan. Ada dua buah sistem inferensi *fuzzy* yang akan digunakan, yaitu sistem inferensi penentuan target dan sistem inferensi pemilihan senjata.

1) Penentuan target

Sistem inferensi penentuan target akan menghitung kesesuaian pemilihan masing-masing target dengan situasi pertandingan. Kesesuaian ini dilihat dari nyawa target, jaraknya dengan landak yang mendapat giliran, tertumpuk tidaknya target dengan landak musuh yang lain, dan apakah target pernah menyerang landak yang mendapat giliran. Target yang perlu diprioritaskan adalah yang memiliki nyawa yang kecil, jarak yang dekat, tertumpuk di antara landak di tim lawan. Target yang pernah menyerang landak yang mendapat giliran menunjukkan bahwa posisinya menguntungkan, jadi perlu juga dipertimbangkan.

2) Pemilihan senjata

Sistem inferensi pemilihan senjata akan menghitung kesesuaian penggunaan masing-masing senjata yang ada pada tim dengan suatu target tertentu. Dalam bagian ini, senjata dibagi menjadi empat buah tipe, yaitu tipe peluncur, tipe granat, tipe senapan, dan tipe pukul. Masing-masing tipe memiliki perhitungan tingkat kesesuaian yang berbeda.

Tingkat kesesuaian pada masing-masing bagian akan dihitung dengan kaidah *fuzzy* yang bersesuaian, dan nilainya akan menjadi keluaran dari sistem inferensi *fuzzy* dalam bentuk pengali. Pengali ini akan menjadi faktor untuk memperbesar atau memperkecil skor pasangan target dan senjata, sehingga mempengaruhi keputusan yang diambil *bot* pada giliran.

D. Rancangan Solusi

1) Variabel Fuzzy dan Fungsi Keanggotaan

Fungsi keanggotaan yang digunakan adalah linier dan segitiga. Bentuk ini digunakan untuk menyederhanakan proses inferensi, terutama pada proses defuzzifikasi yang memerlukan perhitungan daerah di bawah kurva untuk menghitung *centroid*.

a) Masukan

- hptarget: Nyawa landak calon target. Dibagi menjadi 2 himpunan *fuzzy*, yaitu RENDAH dan TINGGI.

- b. jarak: Jarak landak yang mendapat giliran dengan target. Dibagi menjadi 3 himpunan *fuzzy*, yaitu BERSENTUHAN, DEKAT dan JAUH.
- c. penumpukkan: Menunjukkan apakah landak calon target tertumpuk dengan landak-landak tim lawan yang lain. Dibagi menjadi 2 himpunan *fuzzy*, yaitu RENGGANG dan PADAT.
- d. ancaman: Nilainya ditentukan oleh berapa kali target berhasil menyerang landak kawan. Dibagi menjadi 2 himpunan *fuzzy*, yaitu RENDAH dan TINGGI.
- e. ketinggian: Jarak landak yang mendapat giliran dengan target secara vertikal. Dibagi menjadi 3 himpunan *fuzzy*, yaitu LEBIHBAWAH, SETARA, dan LEBIHTINGGI.
- f. halangan: Halangan dapat dihitung dengan menarik garis lurus dari landak yang mendapat giliran ke target, kemudian membandingkan jumlah titik kosong dengan titik yang sudah ditempati arena atau landak lain. Dibagi menjadi 3 himpunan *fuzzy*, yaitu MINIMAL, SEDIKIT dan BANYAK.
- g. angin: Nilai angin menunjukkan seberapa besar kekuatan angin pada giliran. Kekuatan dan arah angin akan mempengaruhi jalannya proyektil yang ditembakkan oleh beberapa tipe senjata. Dibagi menjadi 2 himpunan *fuzzy*, yaitu LEMAH dan KUAT.
- h. frekuensi: Frekuensi menunjukkan berapa kali senjata telah dipilih untuk digunakan. Dibagi menjadi 2 himpunan *fuzzy*, yaitu JARANG dan SERING.

b) Keluaran

Variabel keluaran yang digunakan pada kedua sistem inferensi, penentuan target dan pemilihan senjata, adalah pengali. Pengali memiliki nilai di antara 0 dan 2, dan merupakan variabel *fuzzy* yang dibagi menjadi 3 himpunan *fuzzy*, yaitu RENDAH, CUKUP dan TINGGI.

2) Kaidah Fuzzy

a) Penentuan Target

Kaidah *fuzzy* pada sistem inferensi penentuan target menggunakan variabel-variabel *fuzzy* berupa hptarget, penumpukkan, jarak dan ancaman.

- *if* hptarget RENDAH *then* pengali TINGGI
- *if* ancaman TINGGI *then* pengali TINGGI
- *if* penumpukkan PADAT *then* pengali TINGGI
- *if* jarak BERSENTUHAN *then* pengali TINGGI
- *if* jarak DEKAT *then* pengali CUKUP
- *if* penumpukkan RENGGANG *then* pengali CUKUP
- *if* hptarget TINGGI *then* pengali CUKUP
- *if* jarak JAUH *then* pengali RENDAH
- *if* ancaman RENDAH *then* pengali RENDAH

b) Pemilihan Senjata

Kaidah *fuzzy* pada sistem inferensi pemilihan senjata menggunakan variabel-variabel *fuzzy* yang digunakan adalah hptarget, jarak, ketinggian, angin, halangan, dan frekuensi.

Kaidah yang digunakan berbeda untuk masing-masing tipe senjata.

Tipe Peluncur

- *if* jarak JAUH *then* pengali TINGGI
- *if* halangan SEDIKIT *then* pengali TINGGI
- *if* hptarget RENDAH *then* pengali TINGGI
- *if* angin LEMAH *then* pengali TINGGI
- *if* frekuensi JARANG *then* pengali TINGGI
- *if* jarak DEKAT *then* pengali CUKUP
- *if* ketinggian LEBIHBAWAH *then* pengali CUKUP
- *if* halangan BANYAK *then* pengali CUKUP
- *if* hptarget TINGGI *then* pengali CUKUP
- *if* angin KUAT *then* pengali RENDAH
- *if* ketinggian LEBIHTINGGI *then* pengali RENDAH
- *if* frekuensi SERING *then* pengali RENDAH

Tipe Granat

- *if* jarak DEKAT *then* pengali TINGGI
- *if* angin KUAT *then* pengali TINGGI
- *if* frekuensi JARANG *then* pengali TINGGI
- *if* jarak JAUH *then* pengali CUKUP
- *if* ketinggian LEBIHTINGGI *then* pengali CUKUP
- *if* hptarget RENDAH *then* pengali CUKUP
- *if* ketinggian LEBIHBAWAH *then* pengali RENDAH
- *if* angin LEMAH *then* pengali RENDAH
- *if* frekuensi SERING *then* pengali RENDAH

Tipe Pukul

- *if* jarak BERSENTUHAN *then* pengali TINGGI
- *if* jarak DEKAT *and* halangan MINIMAL *then* pengali TINGGI
- *if* frekuensi JARANG *then* pengali TINGGI
- *if* jarak DEKAT *and* ketinggian SETARA *then* pengali CUKUP
- *if* jarak JAUH *then* pengali RENDAH
- *if* frekuensi SERING *then* pengali RENDAH

Tipe Senapan

- *if* halangan MINIMAL *then* pengali TINGGI
- *if* hptarget RENDAH *then* pengali TINGGI
- *if* frekuensi JARANG *then* pengali TINGGI
- *if* jarak JAUH *then* pengali TINGGI
- *if* jarak DEKAT *then* pengali CUKUP
- *if* halangan SEDIKIT *then* pengali RENDAH
- *if* halangan BANYAK *then* pengali RENDAH
- *if* frekuensi SERING *then* pengali RENDAH

IV. IMPLEMENTASI

Sistem inferensi *fuzzy* diimplementasi pada unit *uFuzzy* yang ditulis dalam bahasa *Pascal*. Kode ini dibuat berdasarkan

template dasar sistem inferensi *fuzzy* yang ditulis oleh [5]. Unit ini mendefinisikan 4 buah tipe bentukan.

- a. *TInOutput*: Tipe ini mewakili variabel *fuzzy* masukan dan keluaran. Dalam tipe ini disimpan nama dan nilai variabel *fuzzy* beserta fungsi keanggotaan untuk semua himpunan *fuzzynya*.
- b. *TMemberFn*: Tipe ini mewakili fungsi keanggotaan untuk suatu himpunan *fuzzy*. Karena bentuk yang digunakan hanya dua, linear dan segitiga, maka sebuah fungsi keanggotaan dapat diwakilkan oleh 2 buah titik dan 2 nilai kemiringan. Dalam tipe ini juga disimpan nama dan derajat keanggotaan sebuah variabel pada himpunan *fuzzy*.
- c. *TRuleElement*: Tipe ini mewakili antesenden atau konsekwen. Dalam tipe ini disimpan *pointer* menuju nilai derajat keanggotaan yang tersimpan pada *TMemberFn*.
- d. *TRule*: Tipe ini mewakili kaidah *fuzzy*. Dalam tipe ini tersimpan kumpulan *TRuleElement* pada bagian antesenden dan konsekwen.

Unit ini juga memiliki 5 buah fungsi dan prosedur untuk menjalankan sistem inferensi *fuzzy*.

- a. *fuzzification*: Menjalankan *compute_degree_of_membership* untuk setiap variabel *fuzzy* masukan.
- b. *rule_evaluation*: Melakukan proses operasi logika *fuzzy* dan implikasi.
- c. *defuzzification*: Melakukan proses agregasi, lalu mengembalikan nilai variabel *fuzzy* keluaran yang sudah didefuzzifikasi.
- d. *compute_degree_of_membership*: Menghitung derajat keanggotaan sesuai dengan parameter nilai variabel *fuzzy* yang diberikan.
- e. *compute_area_of_trapezoid*: Menghitung luas area di bawah fungsi keanggotaan.

Unit ini menyediakan fungsi untuk melakukan seluruh proses sistem inferensi *fuzzy*, yaitu fungsi *inference*.



Gbr. 2. Antarmuka Pemilihan Tingkat Kesulitan

Integrasi akan dilakukan pada dua bagian, yaitu bagian antarmuka pengguna yang menggunakan *framework* Qt dengan bahasa C++ dan bagian *engine* yang ditulis dalam bahasa Pascal. Integrasi antarmuka dilakukan untuk memberikan pilihan pada pemain untuk menggunakan *bot* asli *Hedgewars* atau *bot* yang telah dilengkapi sistem inferensi *fuzzy*. Integrasi *engine* dilakukan dengan memanggil fungsi *inference* untuk memodifikasi skor aksi yang telah dihitung.

V. PENGUJIAN

Ada dua pengujian yang dilakukan dalam tahap ini, yaitu perbandingan kinerja dan pengaruh sistem inferensi *fuzzy* terhadap perilaku *bot*. *Bot* yang sudah ditambahkan sistem inferensi *fuzzy* selanjutnya akan disebut sebagai *bot fuzzy*, dan yang belum disebut sebagai *bot normal*.

A. Perbandingan Kinerja

Pada bagian ini akan dilakukan perbandingan kinerja antara *bot fuzzy* dengan *bot normal*. Hal yang dibandingkan pada pengujian ini adalah waktu berpikir antara kedua *bot* dalam sebuah giliran.

Pengujian ini dilakukan untuk mengukur pengaruh penambahan sistem inferensi *fuzzy* pada *bot Hedgewars* dari segi waktu. Secara logis, penambahan sistem inferensi *fuzzy* akan memperlambat kinerja *bot* karena komputer perlu melakukan perhitungan lebih banyak daripada *bot* yang tidak menggunakan sistem inferensi.

Ada dua kriteria yang didefinisikan pada pengujian ini. Kriteria pertama adalah signifikansi pertambahan waktu berpikir. Oleh karena tujuan utama *bot* adalah untuk digunakan sebagai lawan bermain manusia, maka signifikansi ini akan diuji dari terasa tidaknya perbedaan waktu berpikir *bot fuzzy* dengan *bot normal*. Jika pertambahan waktu berpikir *bot fuzzy* tidak signifikan, maka pengujian dianggap berhasil.

Kriteria kedua adalah pertumbuhan kompleksitas waktu sistem inferensi dengan jumlah objek dalam pertandingan. Kriteria ini diukur dengan menghitung variansi waktu berpikir masing-masing *bot*. Variansi yang besar menunjukkan persebaran data terhadap rata-ratanya lebih jauh. Persebaran ini akan menunjukkan pertambahan waktu secara eksponensial terhadap jumlah objek. Jika variansi waktu berpikir *bot fuzzy* lebih kecil, maka pengujian dianggap berhasil.

TABEL I. HASIL PENGUJIAN KINERJA

Bot	Rata-rata Waktu Eksekusi (ns)	Variansi Waktu Eksekusi
Normal	79626.49733	4159569267
Fuzzy	94132.5877	3281427138
Selisih	14506.09037	878142129

B. Pengujian Pengaruh Sistem Inferensi Fuzzy pada Perilaku Bot

Pada bagian ini akan dilakukan perbandingan perilaku antara *bot fuzzy* dan *bot normal*. Perilaku yang dievaluasi

adalah kesesuaian penentuan target dengan kaidah yang sudah ditentukan dan variasi pemilihan senjata.

Sistem inferensi fuzzy ditambahkan pada *bot* untuk menyesuaikan perilakunya dengan kaidah *fuzzy* yang didefinisikan. Pengujian ini dilakukan untuk melihat seberapa besar sistem inferensi *fuzzy* dapat mengubah perilaku *bot*.

Kriteria pada pengujian penentuan target adalah kesesuaiannya dengan kaidah *fuzzy*. Pada kaidah, target yang diprioritaskan adalah yang memiliki HP kecil, jaraknya dekat, tertumpuk dengan landak lain, atau memiliki nilai ancaman yang besar. Jika *bot fuzzy* terbukti memilih target dengan parameter yang sesuai seperti pada kaidah, maka pengujian dianggap berhasil.

Kriteria pada pengujian pemilihan senjata ada dua, yaitu variasi senjata yang dipilih dan kesesuaiannya dengan kaidah *fuzzy*. Variasi senjata yang dipilih merupakan salah satu faktor *bot* yang tidak mudah diprediksi. Variasi ini akan diukur melalui nilai variansi. Nilai variansi yang lebih kecil menunjukkan bahwa masing-masing senjata dipilih dengan lebih merata satu sama lain, dengan kata lain pemilihan senjatanya yang lebih bervariasi. Kesesuaian pemilihan senjata dengan masing-masing kaidah akan dievaluasi berdasarkan tipe senjata, dengan parameter yang sama seperti pada penentuan target.

Mengingat kondisi pertandingan selalu diacak pada tiap kali *bot* bertanding, maka perbedaan parameter antara *bot* normal dan *bot fuzzy* harus cukup besar untuk menunjukkan perubahan perilaku. Perbedaan yang kecil dapat dianggap sebagai suatu bentuk kebetulan.

1) Kesesuaian Penentuan Target

TABEL II. HASIL PENGUJIAN PENENTUAN TARGET

bot	hptarget	jarak	penumpukkan	ancaman
Normal	69.65664 16	587.281 2838	0.959899 749	2.0526315 79
Fuzzy	65.81704 261	636.224 4215	1.040100 251	2.1002506 27
Selisih	3.839598 99	48.9431 377	0.080200 502	0.0476190 48

Target yang dipilih oleh *bot fuzzy* relatif lebih sesuai dengan kaidah *fuzzy* yang telah ditentukan daripada *bot* normal. *Bot fuzzy* berhasil memilih target dengan HP yang lebih kecil, lebih tertumpuk dengan landak lawan yang lain, dan nilai ancaman yang lebih besar, tetapi gagal dalam memilih jarak rata-rata target karena yang dipilih *bot fuzzy* lebih besar. Dapat dilihat pula bahwa perbedaan masing-masing parameter antara *bot* normal dan *fuzzy* cukup kecil. Oleh karena itu, dalam penentuan target sistem inferensi *fuzzy* dianggap tidak sepenuhnya berhasil mengubah perilaku *bot*.

2) Variasi Pemilihan Senjata

TABEL III. HASIL PENGUJIAN PENENTUAN TARGET

Bot	Normal	Fuzzy	Selisih
Bazooka	237	200	37
Mortar	8	15	7
Drill	16	15	1
Grenade	18	53	35
Watermelon	1	1	0
Gas Bomb	7	11	4
Shotgun	13	12	1
Deagle	15	20	5
SniperRifle	10	8	2
FirePunch	5	2	3
Hammer	3	31	28
Kamikaze	1	3	2
Whip	17	3	14
BaseballBat	6	1	5
AirAttack	16	20	4
Dynamite	2	2	0
Cake	25	3	22
Variansi	2893.543	2631.716	261.827

Hasil pengamatan terhadap data pemilihan senjata pada masing-masing *bot*, dapat dilihat bahwa variansi pemilihan senjata *bot fuzzy* lebih kecil daripada *bot* normal. Sesuai dengan kriteria yang telah disebutkan, dapat disimpulkan bahwa pemilihan senjata pada *bot fuzzy* lebih bervariasi daripada *bot* normal.

3) Kesesuaian Pemilihan Tipe Senjata Peluncur

TABEL IV. HASIL PENGUJIAN TIPE SENJATA PELUNCUR

Bot	hptarget	jarak	ketinngian	halangan	angin	frekuensi
Normal	71.29 88506	628. 9189 016	60.88 59601 6	310.6 2452 1	1.247 4954 02	5.0114 94253
Fuzzy	63	712. 3682 112	- 44.11 79030	333.9 6521 74	1.207 4783 34	4.4521 73913
Selisih	8.298 851	83.4 493	105.0 039	23.34 07	0.040 017	0.5593 2

Dari hasil tersebut, dapat diamati bahwa semua parameter pemilihan tipe senjata peluncur pada *bot fuzzy* lebih sesuai dengan kaidah daripada *bot* normal, kecuali halangan. Tetapi halangan memang tidak memperbesar skor walaupun nilainya tinggi, oleh karena itu hal ini tidak menyalahi kaidah.

4) Kesesuaian Pemilihan Tipe Senjata Granat

TABEL V. HASIL PENGUJIAN TIPE SENJATA GRANAT

Bot	hptarget	jarak	ketinggian	halangan	angin	frekuensi
Normal	86.46 15385	597. 9981 231	- 40.15 32776 4	237.7 3076 9	1.261 0615 41	1.2307 69231
Fuzzy	72.46 15384 6	461. 4894 386	- 34.29 31056	233.2 9230 77	1.475 4410 55	1.4153 84615
Selisih	14	136. 5087	5.860 17	4.438 461	0.214 38	0.1846 2

Dari hasil tersebut, dapat diamati bahwa semua parameter pemilihan tipe senjata granat pada *bot fuzzy* lebih sesuai dengan kaidah daripada *bot normal*.

5) Kesesuaian Pemilihan Tipe Senjata Senapan

TABEL VI. HASIL PENGUJIAN TIPE SENJATA SENAPAN

Bot	hptarget	jarak	ketinggian	halangan	angin	frekuensi
Normal	62.92 10526	225. 8839 752	- 63.12 58802 5	71.84 2105 3	1.344 4830 03	1.6052 63158
Fuzzy	72.9	348. 3074 945	98.81 40334 5	89.32 5	1.426 8020 75	1.6
Selisih	9.978 95	122. 424	161.9 4	17.48 29	0.082 32	0.0052 63

Dari hasil tersebut, dapat diamati bahwa ada ketidaksesuaian dalam pemilihan tipe senapan *bot fuzzy* dengan kaidahnya, yaitu pada parameter hptarget dan halangan.

6) Kesesuaian Pemilihan Tipe Senjata Pukul

TABEL VII. HASIL PENGUJIAN TIPE SENJATA PUKUL

Bot	hptarget	jarak	ketinggian	halangan	angin	frekuensi
Normal	57.68 75	507. 4556 791	38.27 27062 9	291.8 4375	1.420 6404 01	1.1562 5
Fuzzy	63.65	642. 6722 942	- 24.68 08274	281.3 25	1.100 8995 93	1.025
Selisih	5.962 5	135. 217	62.95 353	10.51 875	0.319 741	0.1312 5

Dari hasil tersebut, dapat diamati bahwa ada ketidaksesuaian dalam pemilihan tipe senapan *bot fuzzy* dengan kaidahnya, yaitu pada parameter jarak.

C. Kesimpulan Hasil Pengujian

Dari hasil pengujian di atas, dapat disimpulkan bahwa penambahan sistem inferensi *fuzzy* pada *bot Hedgewars* menyebabkan adanya perubahan perilaku *bot* pada

pertandingan tanpa terlalu menambah waktu yang diperlukan *bot* untuk berpikir. Namun, dapat dilihat pula bahwa perubahan perilaku yang terjadi relatif kecil. Hal ini dapat disebabkan oleh perbedaan skor masing-masing aksi yang mungkin diambil *bot* terlalu besar. Akibatnya, ketika skor dikalikan dengan pengali keluaran dari sistem inferensi *fuzzy*, aksi dengan skor terbesar akan tetap sama dengan sebelumnya. Pada keadaan seperti itu, *bot fuzzy* akan terlihat berperilaku sama dengan *bot normal*.

Selain itu, tidak semua perubahan perilaku pada *bot* sesuai dengan perubahan yang diharapkan ketika merancang kaidah pada sistem inferensi *fuzzy*. Penyebab utamanya adalah faktor ketidakpastian yang muncul pada saat pengujian dilakukan dalam sekian banyak pertandingan. Ketidakpastian ini menyebabkan *bot* "terpaksa" memilih aksi yang berlawanan dengan kaidah *fuzzy* di antara aksi-aksi lain yang lebih tidak sesuai lagi. Contohnya adalah pada ketidaksesuaian parameter jarak pada penentuan target dan pemilihan senjata tipe pukul. Dari data dapat dilihat bahwa secara kebetulan *bot fuzzy* rata-rata mendapatkan jarak yang lebih jauh daripada *bot normal*, oleh karena itu *bot fuzzy* terkadang harus mengambil target yang jauh atau menggunakan senjata tipe pukul walaupun jaraknya jauh.

VI. KESIMPULAN

1. Sistem inferensi *fuzzy* yang digunakan untuk pengambilan keputusan *bot* telah berhasil dirancang dengan menganalisis perilaku *bot Hedgewars* dan menentukan variabel, himpunan, fungsi keanggotaan, dan kaidah *fuzzy* yang sesuai dengan perubahan perilaku *bot* yang diinginkan.
2. Sistem inferensi yang telah dikembangkan telah berhasil diintegrasikan pada *bot Hedgewars* dengan mengubah sistem skor yang digunakan untuk mengambil keputusan. Perubahan yang dilakukan pada sistem skor adalah mengalikan skor yang dihitung dengan nilai yang dihasilkan oleh sistem inferensi. Penambahan sistem inferensi *fuzzy* memberikan kemudahan untuk mengubah perilaku *bot* dalam batas tertentu dengan memodifikasi variabel, fungsi keanggotaan, atau kaidah *fuzzy* tanpa perlu mengganti kode program.
3. Tingkat keberhasilan *bot* yang sudah menggunakan sistem inferensi *fuzzy* diuji dengan membandingkannya dengan *bot* yang tidak menggunakan sistem inferensi. Hal yang dibandingkan adalah waktu berpikir, kesesuaian penentuan target, variasi pemilihan senjata, dan kesesuaian pemilihan masing-masing tipe senjata. Pengujian waktu berpikir, variasi dan kesesuaian pemilihan sebagian tipe senjata berhasil memenuhi kriteria, sedangkan yang lainnya gagal.

REFERENSI

- [1] Bourg, M. David dan Glenn Seeman (2004). *AI for Game Developers*. O'Reilly, Sebastopol, California.
- [2] Pirovano, Michele. (2009). *The use of Fuzzy Logic for Artificial Intelligence in Games*. Computer Science, University of Milano, Milano, Italy.

- [3] Adams, Ernest. (2009). *Fundamentals of Game Design 2nd Edition*. Pearson Education, Berkeley : New Riders.
- [4] Verweij, Tim. (2007). *Hierarchically-Layered MP Bot System*. Department of Computer Science, Faculty Of Science, Vrije Universiteit, Amsterdam.
- [5] Viot, Greg. (1993). *Fuzzy logic in C*. Dr. Dobb's Journal, February 1993 (special issue on Cognitive Computing), <http://www.chebucto.ns.ca/>, diakses 10 April 2014.