Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 3, No. 1, Januari 2019, hlm. 643-650 http://j-ptiik.ub.ac.id

Implementasi *Dynamic Difficulty Adjustment* Pada *Racing Game* Menggunakan Metode *Fuzzy*

Reza Saputra[1], Muhammad Aminul Akbar[2], Tri Afirianto[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]rezasaputra.kks@gmail.com, [2]muhammad.aminul@ub.ac.id, [3]tri.afirianto@ub.ac.id

Abstrak

Pada beberapa *racing game* terdapat fitur untuk memainkan permainan secara kompetitif baik melawan pemain lain atau melawan Kecerdasan Buatan (KB) atau yang biasa disebut *bot.* Dalam penerapannya, sering terjadi perbedaan kemampuan yang jauh antara pemain dengan *bot.* Hal ini menyebabkan adanya rasa bosan jika kemampuan pemain jauh lebih baik dibandingkan dengan *bot,* dan akan menimbulkan rasa kegelisahan jika kemampuan pemain jauh lebih rendah jika dibandingkan dengan kemampuan *bot*. Dalam beberapa *racing game* terdapat pilihan untuk memilih tingkat kesulitan sebelum bermain, namun fitur ini dirasa masih kurang dapat mengimbangkan kemampuan pemain dengan *bot* karena seiring berjalannya waktu kemampuan pemain dapat meningkat, dan pemain baru juga cenderung tidak mengetahui tingkat kesulitan yang sesuai dengan kemampuannya. Untuk mengatasi masalah tersebut maka peneliti akan menerapkan *Dynamic Difficulty Adjustment* (DDA) menggunakan metode *Fuzzy* yang mampu menyesuaikan kemampuan *bot* dengan kemampuan pemain seiring berjalannya waktu. Pengujian DDA dilakukan dengan menguji kemampuan *bot* statis dengan *bot* DDA. Hasil pengujian menunjukkan bahwa *bot* DDA mampu menyesuaikan perilakunya dengan kemampuan yang dimiliki oleh pemain, yang mana terjadi perubahan nilai parameter keluaran pada detik ke-35, nilai parameter keluaran yang dihasilkan untuk parameter caution angle, steer sensitivity, max wander distance, dan wander rate secara berurut adalah 41,83, 0,012, 3,57, dan 0,03. Keseluruhan nilai parameter tersebut masuk dalam kategori *HARD.*

Kata kunci: *fuzzy, dynamic difficulty, dynamic difficulty adjustment, DDA, game, racing game*

Abstract

Some racing games have a feature to play the game competitively against other player or against Artificial Intelligence (AI) or commonly called as a bot. In its implementation, we tend to find that the ability between the player and the bot is far. This causes boredom if the player has much higher ability than the bot, and will produce anxiety if the player's ability is much lower when compared with the bot. In some racing games there is an option to choose the difficulty level before starting the game, but this feature is still considered less effective to balance the ability of player and bot, because the ability of players can increase as the time goes by, and new players tend to confused that they don't know what category of difficulty that suits their abitility. To solve the problem, the researcher will implement Dynamic Difficulty Adjustment (DDA) by using Fuzzy method that able to adjust the ability of the bot according to player's ability over time. DDA testing is done by playing and matching static bots with DDA bots. Test results show that DDA bots are able to adjust their behavior with the static bots ability, in which the output parameter value changes at 35 seconds, the output parameter values generated for caution angle, steer sensitivity, max wander distance, and wander rate are 41,83, 0.012, 3,57, and 0,03 respectively. The overall value of the parameter categorized as HARD.

Keywords: *fuzzy, dynamic difficulty, dynamic difficulty adjustment, DDA, game, racing game*

1. PENDAHULUAN

Seiring zaman berkembang perkembangan *video game* terus terjadi, salah satunya pada *racing game*. Menurut data dari bigfish*game*s.com (Big Fish *Game*s, Inc., n.d.), *racing game* memiliki angka keminatan sebanyak 4% untuk wilayah Amerika, 10%

untuk wilayah Eropa, dan 3% pada wilayah Jepang. Meskipun secara angka *racing game* masih diungguli oleh jenis *video game* lainnya, namun *racing game* masih banyak diminati oleh pemain dari kalangan pecinta olahraga dan adrenalin. *Racing game* sendiri merupakan suatu jenis dalam *video game* yang mensimulasikan pertandingan balapan sebagaimana halnya balapan dalam dunia nyata, pemain diharuskan untuk mengendarai kendaraan balap dan berkompetisi dengan pembalap lainnya untuk mencapai garis finish dengan menduduki posisi terdepan.

Perkembangan dalam *racing game* juga terus terjadi baik dalam hal grafis, suasana permainan yang lebih realistis, dan juga dalam hal kompetitif. Pada beberapa *racing game* terdapat fitur untuk memainkan permainan secara kompetitif baik melawan pemain lain ataupun melawan Kecerdasan Buatan (KB) atau yang biasanya disebut *bot*. Pengembangan permainan yang kompetitif tidak hanya terjadi pada *racing game* tetapi juga pada *video game* dengan jenis lainnya. Tujuan dibuatkannya permainan yang kompetitif ini adalah untuk mencapai unsur *fun* dan *challenging* yang harus dimiliki oleh suatu *video game*. Dalam *video game*, banyak hal yang dapat mempengaruhi tingkat kepuasan yang diperoleh pemain. Salah satunya adalah tingkat kesulitan yang dialami oleh pemain baik dalam melawan pemain yang lain maupun melawan kecerdasan buatan. Csikszentmihalyi (Silva, et al., 2016), mengungkapkan bahwa Ketika seseorang sedang melakukan suatu aktivitas dan menghayatinya, orang tersebut akan fokus kepada hal yang dikerjakan dan akan memengaruhi keadaan mentalnya. Hal inilah yang terjadi ketika seseorang sedang bermain *game*. Pemain yang memiliki tingkat kemampuan yang tinggi akan cenderung bosan atau tidak merasakan tantangan dalam bermain ketika dihadapkan pada lawan yang memiliki kemampuan yang rendah, sebaliknya ketika pemain yang kemampuannya rendah akan merasa stres dan frustasi ketika dihadapkan pada pemain yang memiliki kemampuan yang tinggi dikarenakan adanya perbedaan tingkat kemampuan yang jauh yang membuat pemain merasa jauh tertinggal atau dikalahkan dengan mudah.

Untuk menjawab permasalahan di atas, terdapat beberapa solusi yang ada seperti menambahkan fitur yang secara eksplisit memperbolehkan pemain untuk memilih tingkat kesulitanan permainan yang sesuai dengan kemampuannya sebelum memulai permainan tersebut. Namun solusi ini dirasa masih kurang karena pemain diharuskan untuk memilih tingkat kesulitanannya sebelum bermain dikarenakan seiring berjalannya permainan tingkat kemampuan bermain dapat meningkat, dan pemain baru juga cenderung tidak mengetahui tingkat kesulitan yang sesuai dengan kemampuannya. Solusi lainnya adalah menggunakan ilmu Kecerdasan Buatan yaitu *Dynamic Difficulty Adjustment* (DDA). DDA dapat mengimbangkan kemampuan pemain dengan lawannya secara otomatis berdasarkan kemampuan pemain melalui komputasi cerdas yang dirancang oleh desainer dan pengembang. DDA banyak digunakan dalam mengatur tingkat kesulitan permainan dalam banyak *video game* seperti pada *Half-Life 2*, *Tower Defense Games*, dan lain-lain.

Sebelumnya terdapat penelitian yang mengangkat topik implementasi DDA menggunakan *Behavior Trees* (BT) pada Fighting *Game* dengan jarak dan delay sebagai parameternya (Jacobsen, et al., 2011). BT merupakan suatu teknik Kecerdasan Buatan dalam *video game* yang mengatur tingkah laku yang dijalankan oleh suatu agent Kecerdasan Buatan berdasarkan beberapa action yang direpresentasikan menggunakan Tree. Kekurangan dari implementasi DDA menggunakan *Behavior Trees* pada penelitian tersebut terdapat kesulitan dalam mengimplementasikannya karena menambahkan utility kedalam framework BT terbukti merupakan hal yang hard untuk dilakukan sebagaimana yang dialami oleh peneliti sebelumnya.

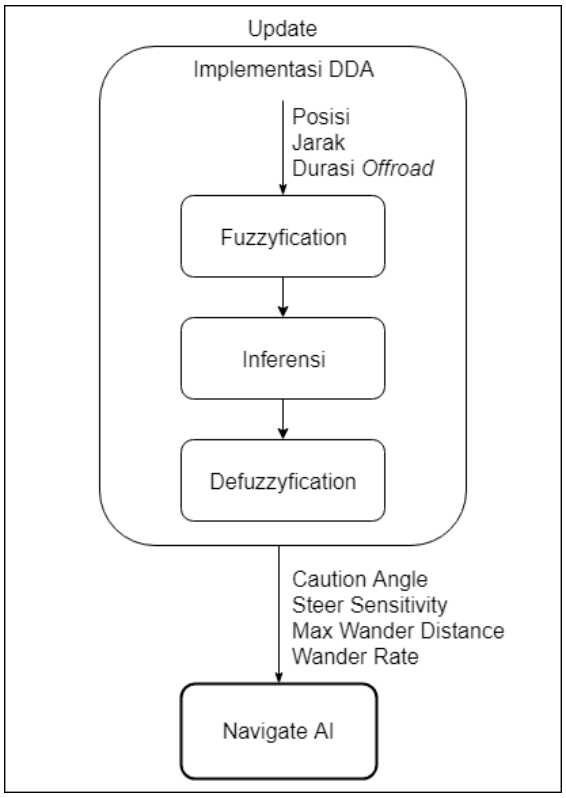
Penelitian lainnya yang diteliti oleh Rietveld yaitu circuit-adaptive rubber banding for *racing game*s (Rietveld, 2014). Penelitian ini memfokuskan untuk mengimbangkan kemampuan antar pemain dengan menyesuaikan jalur balap yang berbeda sesuai dengan kemampuan yang dimiliki oleh tiap pemain berdasarkan tingkat kemampuan easy dan hard. Kelebihan dari penelitian tersebut adalah hasilnya mampu meningkatkan keseimbangan pemain menjadi lebih kompetitif, namun memiliki kekurangan yang mana keluaran merupakan penyesuaian jalur yang berbeda untuk tiap pemain dirasa masih kurang adil.

Penulis mencoba untuk meningkatkan kekurangan yang terdapat pada DDA dalam *racing game* dengan mengimplementasikan

metode *Fuzzy*. Kelebihan dari metode *Fuzzy*. Adapun kelebihan dari metode *Fuzzy* adalah sangat fleksibel, konsep dasar yang mudah dimengerti, memiliki toleransi tinggi terhadap kebenaran suatu data, dan mampu membuat variasi dan mengambil langkah optimal dari suatu kondisi (Kusumadewi & Purnomo, 2010). Sebelumnya terdapat penelitian yang menggunakan metode *Fuzzy* untuk mengatur perilaku musuh dalam *Action-RPG Game* (Purba, et al., 2013). Pada hasilnya, penelitian mampu menerapkan *Fuzzy* kedalam *Action-RPG Game* dan mampu menghasilkan keluaran sesuai dengan aturan yang ada dengan baik.

Dengan adanya penelitian DDA menggunakan metode Fuzzy ini, diharapkan mampu meningkatkan keseimbangan antara pemain dengan *bot*, yang mana secara tidak langsung akan meningkatkan pengalaman pemain baik itu merasa lebih menyenangkan maupun menantang.

2. IMPLEMENTASI



Gambar *1*. Alur Kerja Proses Implementasi DDA

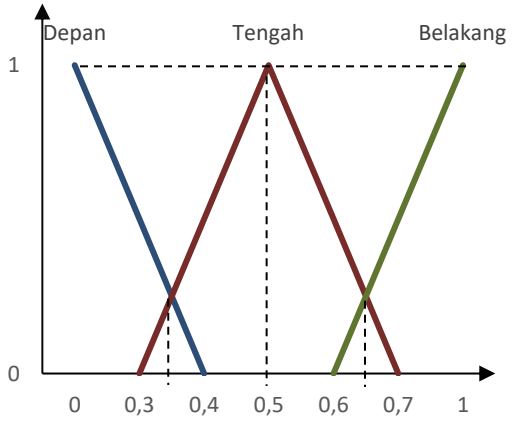
Implementasi DDA dijalankan dalam fungsi *Update* yang terdapat pada Unity *framework*, fungsi *Update* akan dijalankan secara terus menerus dalam delta waktu. Dalam alur yang telah digambarkan pada **Error! Reference source not found.**, dalam fungsi *Update* dijalankan fungsi Impementasi DDA dan fungsi *Navigate* AI. Dalam fungsi Implementasi DDA, dijalankan proses logika *fuzzy* yang terdiri dari *fuzzyfication*, inferensi, dan *defuzzyfication*. Pada tahap *fuzzyfication* akan dicari derajat keanggotaan untuk parameter posisi, jarak, dan durasi *offroad* yang telah dimasukkan sebelumnya. Kemudian dilanjutkan dengan proses Inferensi untuk mencari nilai α-predikat dan nilai z (nilai tegas) untuk setiap parameter keluaran, kemudian diakhiri dengan tahap *defuzzyfication* untuk mencari nilai Z (nilai keluaran) untuk setiap parameter keluaran. Hasil dari proses logika *fuzzy* ini adalah nilai keluaran berupa *caution angle*, *steer sensitivity*, *max wander distance*, dan *wander rate* yang telah menyesuaikan nilai dari parameter masukan.

Setelah fungsi Implementasi DDA selesai dijalankan, selanjutnya akan dijalankan fungsi *Navigate AI*, yang mana di dalamnya terdapat proses untuk menghitung kecepatan yang dihasilkan oleh mobil menggunakan bayak parameter yang beberapa di antaranya adalah *caution angle*, *steer sensitivity*, *max wander distance*, dan *wander rate*, yang mana nilai keempat parameter ini telah diperoleh dari proses DDA menggunakan *fuzzy* sebelumnya.

2.1 Parameter Masukan

2.1.1 Posisi

Nilai parameter posisi diperoleh dengan cara mengambil nilai posisi sementara pemain pada saat waktu DDA *fuzzy* akan diimplementasikan. Parameter posisi dibagikan menjadi tiga kategori yaitu depan, tengah dan belakang. Pemberian posisi direpresentasikan dengan nilai dengan skala 0-1 diperoleh dengan cara membagikan nilai posisi pemain dengan jumlah total dari pebalap. Fungsi keanggotaan parameter posisi dapat dilihat pada Gambar 2.



Gambar 2 Fungsi Keanggotaan Parameter Posisi

Berdasarkan Gambar 2, suatu pembalap dikatakan berada pada kategori depan jika berada pada posisi dengan skala di bawah 0,4, sedangkan pembalap yang berada pada posisi dengan skala di antara 0,3 sampai 0,7 bisa dikatakan berada pada kategori tengah, dan pembalap dengan skala posisi di atas 0,6 bisa dikatakan berada pada kategori belakang. Fungsi persamaan keanggotaan untuk himpunan depan, tengah, dan belakang dari variabel posisi dimodelkan sebagai berikut :

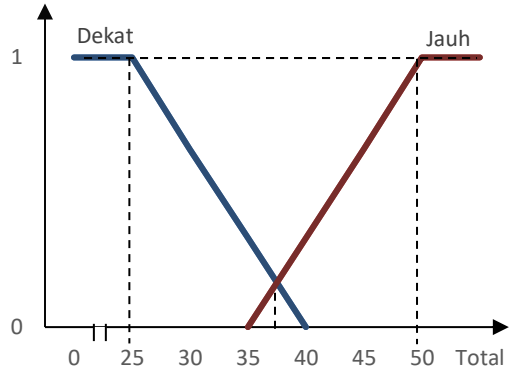
$$\mu PosisiDEPAN(x) = \begin{cases} 0; & x \geq 0{,}4 \\ \frac{0{,}4 - x}{0{,}4 - 0}; & 0 \leq x \leq 0{,}4 \end{cases} \quad (1)$$

$$\mu PosisiTENGAH(x) = \begin{cases} 0; & x \leq 0{,}3 \; atau \; x \geq 0{,}7 \\ \frac{x - 0{,}3}{0{,}5 - 0{,}3}; & 0{,}3 \leq x \leq 0{,}5 \\ \frac{0{,}7 - x}{0{,}7 - 0{,}5}; & 0{,}5 \leq x \leq 0{,}7 \end{cases} \quad (2)$$

$$\mu PosisiBelakang(x) = \begin{cases} 0; & x \leq 0{,}6 \\ \frac{x - 0{,}6}{1 - 0{,}6}; & 0{,}6 \leq x \leq 1 \end{cases} \quad (3)$$

2.1.2 Jarak

Nilai parameter jarak diperoleh dengan cara menghitung nilai absolut dari selisih jarak tempuh antara *bot* dengan pemain. Parameter jarak dibagikan menjadi dua kategori yaitu dekat dan jauh. Fungsi persamaan keanggotaan untuk parameter jarak dapat dilihat pada Gambar 3.



Gambar 3 Fungsi Keanggotaan Parameter Jarak

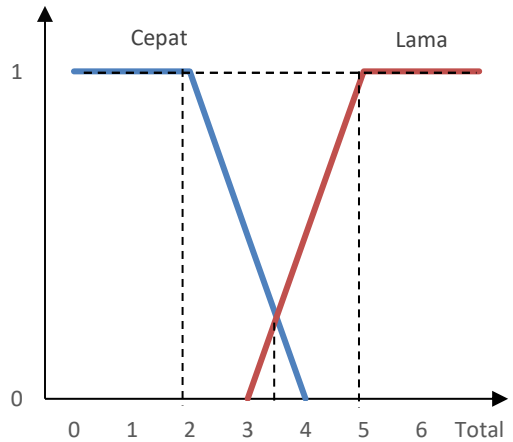
Berdasarkan Gambar 3, nilai jarak dimasukkan ke dalam kategori dekat jika bernilai di antara 0 meter sampai dengan 40 meter, sedangkan perbedaan dimasukkan kategori jauh jika jaraknya bernilai lebih dari 35 meter. Fungsi persamaan keanggotaan untuk himpunan dekat dan jauh dari variabel jarak dimodelkan sebagai berikut :

$$\mu JarakDEKAT(x) = \begin{cases} 0; & x \geq 40 \\ \frac{40 - x}{40 - 25}; & 25 \leq x \leq 40 \\ 1; & x \leq 25 \end{cases} \quad (4)$$

$$\mu JarakJAUH(x) = \begin{cases} 0; & x \leq 28 \\ \frac{x - 35}{50 - 35}; & 35 \leq x \leq 50 \\ 1; & x \geq 1 \end{cases}$$

2.1.3 *Offroad*

Nilai parameter *offroad* diperoleh dengan cara menghitung lamanya suatu bagian dari mobil *bot* ketika berada diluar jalur. Parameter *offroad* dibagikan menjadi dua kategori yaitu cepat dan lama. Fungsi keanggotaan parameter durasi *offroad* Gambar 4.



Gambar 4 Fungsi Keanggotaan Parameter *Offroad*

Berdasarkan Gambar 4, durasi *offroad* dimasukkan ke dalam kategori cepat jika bernilai lebih kecil dari 4 detik dan kategori lama jika bernilai lebih dari 3 detik. Fungsi persamaan keanggotaannya dimodelkan sebagai berikut :

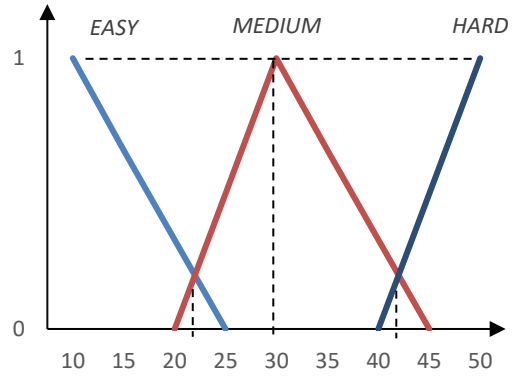
$$JarakDEKAT(x) = \begin{cases} 0; & x \geq 40 \\ \frac{40-x}{40-25}; & 25 \leq x \leq 40 \\ 1; & x \leq 25 \end{cases}$$

$$\mu JarakJAUH(x) = \begin{cases} 0; & x \leq 28 \\ \frac{x-35}{50-35}; & 35 \leq x \leq 50 \\ 1; & x \geq 1 \end{cases}$$

2.2 Parameter Keluaran

2.2.1 *Caution Angle*

Parameter *caution angle* dibagikan menjadi tiga kategori yaitu *easy*, *medium* dan *hard*. Fungsi keanggotaan parameter *caution angle* dapat dilihat pada Gambar 5.



Gambar 5 Fungsi Keanggotaan Parameter *Caution Angle*

Berdasarkan Gambar 5, *caution angle* dimasukkan ke dalam kategori *easy* jika bernilai antara 15 sampai 25, kategori *medium* jika bernilai di antara 20 hingga 45, dan kategori *hard* jika bernilai di antara 40 hingga 50. Fungsi persamaan keanggotaannya dimodelkan sebagai berikut :

$$\mu caEASY(o1) = \begin{cases} 0; & x \geq 25 \\ \frac{25-x}{25-10}; & 10 \leq x \leq 25 \end{cases} \quad (8)$$

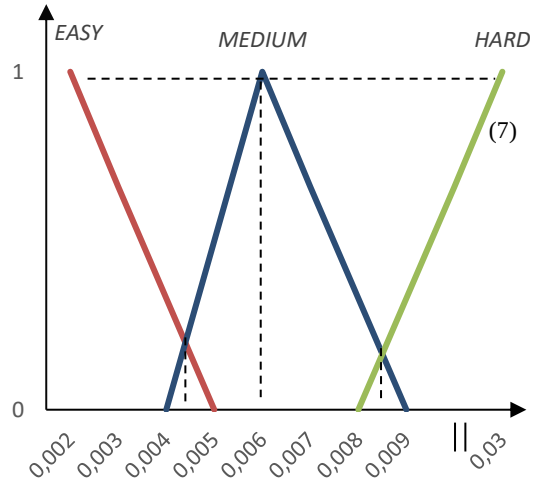
$$\mu caMEDIUM(o1) = \begin{cases} 0; & x \leq 20 \; atau \; x \geq 45 \\ \frac{x-20}{30-20}; & 20 \leq x \leq 30 \\ \frac{45-x}{45-30}; & 30 \leq x \leq 45 \end{cases} \quad (9)$$

(10)

$$\mu caHARD(o1) = \begin{cases} 0; & x \leq 40 \\ \frac{x-40}{50-40}; & 40 \leq x \leq 50 \end{cases}$$

2.2.2 *Steer Sensitivity*

Parameter *steer sensitivity* dibagikan menjadi tiga kategori yaitu *easy*, *medium* dan *hard*. Fungsi keanggotaan parameter *steer sensitivity* dapat dilihat pada Gambar 6. (6)



(7)

Gambar 6 Fungsi Keanggotaan Parameter *Steer Sensitivity*

Berdasarkan Gambar 6, *steer sensitivity* dimasukkan ke dalam kategori *easy* jika bernilai antara 0,002 sampai 0,005, kategori *medium* jika bernilai di antara 0,006 hingga 0,09, dan kategori *hard* jika bernilai di antara 0,008 hingga 0,03. Fungsi persamaan keanggotaannya dimodelkan sebagai berikut:

$$\mu ssEASY(o2) = \begin{cases} 0; & x \geq 0{,}05 \\ \frac{0{,}05-x}{0{,}05-0.02}; & 0{,}02 \leq x \leq 0{,}05 \end{cases}$$

(11)

$$\mu ssMEDIUM(o2) = \begin{cases} 0; & x \leq 0{,}004 \; atau \; x \geq 0{,}009 \\ \frac{x-0{,}004}{0{,}006-0{,}004}; & 0{,}004 \leq x \leq 0{,}006 \\ \frac{0{,}009-x}{0{,}009-0{,}006}; & 0{,}006 \leq x \leq 0{,}009 \end{cases}$$

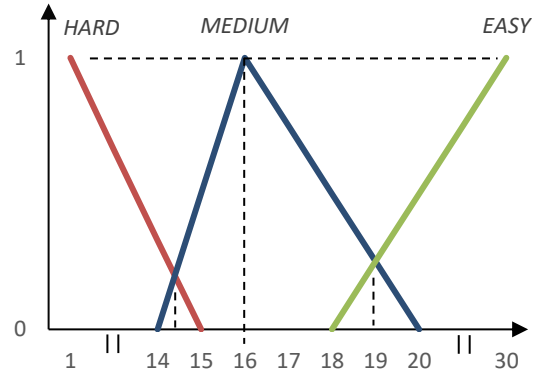
(12)

$$\mu caHARD(o1) = \begin{cases} 0; & x \leq 0{,}008 \\ \frac{x-0{,}008}{0{,}03-0{,}008}; & 0{,}008 \leq x \leq 0{,}03 \end{cases}$$

(13)

2.2.3 *Max Wander Distance*

Parameter *max wander distance* dibagikan menjadi tiga kategori yaitu *easy*, *medium* dan *hard*. Fungsi keanggotaan parameter *max wander distance* dapat dilihat pada Gambar 7.



Gambar 7 Fungsi Keanggotaan Parameter *Max Wander Distance*

Berdasarkan Gambar 7, *max wander distance* dimasukkan ke dalam kategori *easy* jika memiliki nilai jarak di antara 18 hingga 30, kategori *medium* jika memiliki nilai jarak di antara 14 hingga 20, dan kategori *hard* jika memiliki nilai jarak di antara 1 hingga 15. Fungsi persamaan keanggotaannya dimodelkan sebagai berikut:

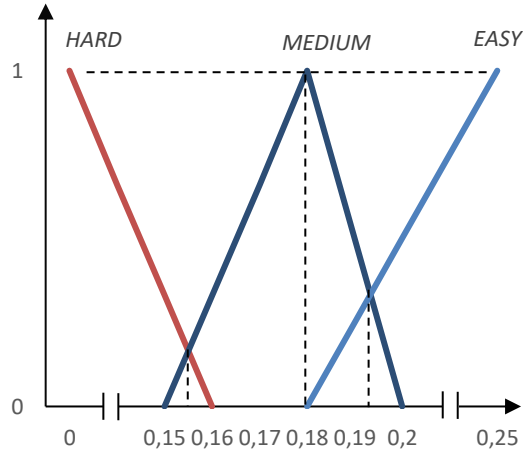
$$\mu mwEASY(o3) = \begin{cases} 0; & x \leq 18 \\ \frac{30-x}{30-18}; & 18 \leq x \leq 30 \end{cases}$$

$$\mu mwMEDIUM(o3) = \begin{cases} 0; & x \leq 14 \; atau \; x \geq 20 \\ \frac{x-14}{16-14}; & 14 \leq x \leq 16 \\ \frac{20-x}{20-16}; & 16 \leq x \leq 20 \end{cases} \quad (15)$$

$$\mu mwHARD(o3) = \begin{cases} 0; & x \geq 15 \\ \frac{x-1}{15-1}; & 1 \leq x \leq 15 \end{cases} \quad (16)$$

2.2.4 *Wander Rate*

Parameter *wander rate* dibagikan menjadi tiga kategori yaitu *easy, medium* dan *hard*. Fungsi keanggotaan parameter *wander rate* dapat dilihat pada Gambar 8.



Gambar 8 Fungsi Keanggotaan Parameter *Wander Rate*

Berdasarkan Gambar 8, *Wander rate* dimasukkan ke dalam kategori *easy* jika bernilai antara 0,18 hingga 0,25, kategori *medium* jika bernilai di antara 0,15 hingga 0,2, dan kategori *hard* jika bernilai di antara 0 hingga 0,16. Fungsi persamaan keanggotaannya dimodelkan sebagai berikut :

$$\mu wrEASY(o4) = \begin{cases} 0; & x \leq 0{,}18 \\ \frac{0{,}25-x}{0{,}25-0{,}18}; & 0{,}18 \leq x \leq 0{,}25 \end{cases} \quad (14)$$

$$\mu wrMEDIUM(o4) = \begin{cases} 0; & x \leq 0{,}15 \; atau \; x \geq 0{,}20 \\ \frac{x-0{,}15}{0{,}18-0{,}15}; & 0{,}15 \leq x \leq 0{,}18 \\ \frac{0{,}20-x}{0{,}20-0{,}18}; & 0{,}18 \leq x \leq 0{,}20 \end{cases} \quad (17)$$

$$\mu mwHARD(o4) = \begin{cases} 0; & x \geq 0{,}16 \\ \frac{x-0}{0{,}16-0}; & 0 \leq x \leq 0{,}16 \end{cases}$$

2.3 Pendefinisian *Rule*

Dalam bagian ini dipaparkan aturan-aturan *fuzzy* yang digunakan dalam penelitian. Aturan-aturan ini digunakan dalam proses inferensi.

Tabel 1 *Rule Fuzzy*

No	Masukan			Keluaran
	Posisi	Jarak	*Offroad*	
1	DEPAN	JAUH	CEPAT	*HARD*
2	DEPAN	JAUH	LAMA	*HARD*
3	DEPAN	DEKAT	CEPAT	*HARD*
4	DEPAN	DEKAT	LAMA	*MEDIUM*
5	TENGAH	JAUH	CEPAT	*MEDIUM*

6	TENGAH	JAUH	LAMA	*MEDIUM*
7	TENGAH	DEKAT	CEPAT	*MEDIUM*
8	TENGAH	DEKAT	LAMA	*MEDIUM*
9	BELAKANG	JAUH	CEPAT	*EASY*
10	BELAKANG	JAUH	LAMA	*EASY*
11	BELAKANG	DEKAT	CEPAT	*EASY*
12	BELAKANG	DEKAT	LAMA	*EASY*

Berdasarkan Tabel 1, jika posisi pemain masuk dalam kategori DEPAN, jarak mobil antara *bot* dan pemain masuk dalam kategori JAUH, dan durasi *offroad* masuk dalam kategori CEPAT, maka keluaran yang dihasilkan berupa *HARD*, yang mana keempat parameter keluaran (*caution angle, steer sensitivity, max wander distance*, dan *wander rate)* masing-masing akan masuk ke dalam kategori *HARD*. Begitupula jika parameter kondisi dari masukan menghasilkan keluaran berupa *MEDIUM*, keempat parameter keluaran akan masuk ke dalam kategori *MEDIUM*, begitu juga untuk keluaran kategori *EASY*.

3. PENGUJIAN DAN ANALISIS

Pada bagian ini akan dipaparkan hasil pengujian-pengujian yang telah dilakukan beserta analisisnya.

Tabel 2 Nilai Parameter Masukan

Detik	Posisi	Jarak	*Offroad*
7	2	5.36 m	2 detik
14	2	5.09 m	1 detik
21	2	19.26 m	2 detik
28	2	1,53 m	3 detik
35	1	27,75 m	1 detik
42	1	37,94 m	0 detik
49	1	30,35 m	2 detik
56	1	28,24 m	0 detik
63	1	26,73 m	4 detik
70	1	25,41 m	2 detik

Tabel 3 Nilai Parameter Keluaran

Detik	*Caution Angle*	*Steer Sensitivity*	*Max Wander*	*Wander rate*
7	22,50	0,03	22,00	0,2,
14	22,50	0,0045	22,00	0,2,
21	22,50	0,0045	22,00	0,2,
28	22,50	0,0045	22,00	0,2,
35	41,83	0,012	3,57	0,03
42	40,00	0,008	1,00	0
49	46,43	0,022	10,01	0,1
56	40,00	0,008	1,00	0
63	40,00	0,008	1,00	0
70	44,28	0,017	6,99	0,07

Berdasarkan Tabel 2 pada detik ke-7, nilai parameter masukan posisi, jarak, dan *offroad* secara berurutan adalah 2, 5,36 meter, dan 3 detik, berdasarkan parameter *fuzzy* masukan yang telah dirancang, maka nilai posisi masuk dalam kategori TENGAH, nilai jarak masuk dalam kategori DEKAT, dan nilai *offroad* masuk dalam kategori CEPAT. Nilai dari ketiga parameter tersebut masuk dalam *rule* nomor 7 seperti yang tercantum pada Tabel 1. Berdasarkan rule nomor 7, maka keluaran yang dihasilkan adalah *EASY*, yang mana hasil keluarannya dapat dilihat pada Tabel 3 pada detik ke-7, nilai dari parameter keluaran *caution angle*, *steer sensitivity*, *max wander distance*, dan *wander rate* secara berurutan nilainya adalah 22,50, 0,03, 22, dan 0,2. Berdasarkan parameter keluaran *fuzzy* yang telah dirancang, maka nilai dari parameter-parameter keluaran tersebut masuk dalam kategori *EASY*. Pada Tabel 3 detik ke-35 terjadi perubahan nilai parameter yang mencolok, hal ini terjadi karena pada nilai parameter-parameter masukan pada detik ke-35 menghasilkan keluaran berupa *HARD*. Bisa dilihat pada Tabel 2 detik ke-35, nilai posisi, jarak, dan *offroad* secara berurutan adalah 1, 27,75 meter, dan 4 detik, yang mana nilai posisi masuk pada kategori DEPAN, nilai jarak masuk dalam kategori JAUH, dan nilai *offroad* masuk dalam kategori LAMA. Nilai dari ketiga parameter tersebut masuk dalam *rule* nomor 2 seperti yang dipaparkan pada Tabel 1. Terjadinya perubahan perilaku dari *MEDIUM* menjadi *HARD* ini mengakibatkan kemampuan dari *bot* DDA meningkat sehingga pada detik ke-35, *bot* DDA mengejar ketertinggalan. Adanya perubahan nilai parameter ini menunjukkan bahwa *bot* DDA sudah mampu melakukan adaptasi terhadap perilakunya berdasarkan kemampuan dari pemain sesuai dengan nilai parameter masukan yang diperoleh selama balapan berlangsung.

4. KESIMPULAN

Kesimpulan-kesimpulan yang diperoleh didasarkan dari hasil pengujian yang diperoleh

dan analisis. Adapun kesimpulan yang dapat diambil adalah :

1. DDA dengan metode *fuzzy* diimplementasikan ke dalam *racing game* menggunakan posisi, jarak, dan durasi *offroad* sebagai parameter masukan, parameter keluaran yang digunakan adalah *caution angle, steer sensitivity, max wander distance*, dan *wander rate.* Kemudian implementasi DDA dengan metode *fuzzy* dilakukan setiap tujuh detik setelah balapan dimulai.
2. Sistem DDA sudah mampu berjalan dengan baik, yang mana pada detik ke-7 nilai parameter keluaran tergolong dalam kategori *MEDIUM*, kemudian pada detik ke-35 nilai parameter keluaran tergolong dalam kategori *HARD* sehingga parameter keluaran yang dihasilkan untuk *caution angle, steer sensitivity, max wander distance*, dan *wander rate secara* berurutan adalah 41,83, 0,012, 3,57, dan 0,03.

5. DAFTAR PUSTAKA

Arsenault, D., 2009. *Video game* Genre, Evolution and Innovation. Journal for Computer *Game* Culture, Volume 3, pp. 149-176.

Big Fish *Game*s, Inc., n.d. *Game* Genres Across the World. [Online] Tersedia di: https://www.bigfish*game*s.com/blog/stats/*game*-genres-across-the-world/ [Diakses 12 December 2017].

Fitri, A. & Mahmudy, W. F., 2017. Optimasi Keanggotaan Fuzzy Tsukamoto Menggunakan AlgoritmaGenetika pada Penentuan Prioritas Penerima Zakat. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Volume 1, pp. 125-138.

Jacobsen, K. S., Olsen, T. & Phan, L. H., 2011. Dynamic Difficulty Adjustment Using Behavior Trees. Master Thesis.

Kusumadewi, S. & Purnomo, H., 2010. Aplikasi logika fuzzy untuk pendukung keputusan. Jakarta: Graha Ilmu.

Porsinelli, P., 2013. Why is Unity so popular for video*game* development?. [Online] Tersedia di: https://designa*game*.eu/2013/12/unity-popular-video*game*-development/ [Diakses 15 Juli 2018].

Purba, K. R., Hasanah, R. N. & Muslim, M. A., 2013. Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam *Game* Bertipe Action-RPG. Jurnal EECCIS (Electrics Electronics Communications Controls Informatics Systems), Volume 7, pp. 15-20.

Rietveld, A., 2014. Circuit-adaptive Rubber Banding. MSc Thesis.

Silva, M. P., Silva, V. d. N. & Chaimowicz, L., 2016. Dynamic Difficulty Adjustment Through an Adaptive AI. Brazilian Symposium on Computer *Game*s and Digital Entertainment, Volume 14, p. 174.

US *Game*r, 2015. The Grandfather of *Racing game*s. [Online] Tersedia di: https://www.us*game*r.net/articles/the-first-overhead-viewed-racer-was-a-classic [Diakses 15 Juli 2018].