

Penerapan *Fuzzy Control* untuk Pengereman Kereta dalam *Simulation Game*

Reinaldo Julaidi¹, Rolly Intan², Rudy Adipranata³

Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: reinaldojul12@gmail.com¹, rintan@petra.ac.id², rudya@petra.ac.id³

ABSTRAK

Penelitian mengenai pengereman kereta telah banyak dilakukan. Sebagai contoh, Wahyudi melakukan penelitian perhitungan pengereman kereta berupa kecepatan dengan *Fuzzy*. Yulianti melakukan penelitian pengereman kereta berupa kecepatan dan jarak dengan *Fuzzy*, dan di simulasikan. Namun, penelitian yang dilakukan masih belum menyesuaikan dengan keadaan kereta sesungguhnya saat dilakukan pengereman.

Karena itu, dalam skripsi ini akan dibuat sebuah *game* dengan *genre Simulation Game* dengan AI untuk mensimulasikan pengereman kereta. Pengereman kereta yang dilakukan menggunakan *Fuzzy Control* dimana akan melakukan pengereman berdasarkan setiap *Fuzzy rules* dengan menggunakan metode Mamdani dengan kecepatan, massa, dan jarak sebagai *Fuzzy Input*. Hasil pengujian menunjukkan bahwa kereta dapat berhenti di tujuan dengan melakukan pengereman yang dilakukan oleh *Fuzzy Control* dengan perlahan hingga berhenti di tujuan dengan menggunakan *Fuzzy Inference Engine* yang melakukan proses untuk menghasilkan setiap tindakan keputusan kekuatan pengereman kereta.

Kata Kunci: Fuzzy Logic Controller, Pengereman Kereta, C#, Unity

ABSTRACT

There are a lot of research about Train braking has been done. For example, Wahyudi research calculation of train braking with speed with *Fuzzy*. Yulianti research train braking with speed and distance with *Fuzzy*, and simulated it. However, researchers did not adjust with the real situation of train when it does braking.

Because of that, this thesis will create a game with *Simulation Game* genre with AI to simulate train braking. *Fuzzy Control* will do train braking which will do the braking based on each *Fuzzy rules* using the Mamdani method with speed, mass, and distance as *Fuzzy input*.

The test results show that the train can stop at the destination by braking that is done by *Fuzzy Control* slowly until it stops at the destination using the *Fuzzy Inference Engine* that performs the process to produce every decision for every action on the braking force of the train.

Keywords: Fuzzy Logic Controller, Train brake, C#, Unity

1. PENDAHULUAN

Simulation game adalah *game* yang di desain berdasarkan dunia nyata dengan kemiripan yang sesuai dengan apa yang disimulasikan. *Simulation game* bukan hanya untuk menyenangkan

saat dimainkan, namun juga memiliki banyak manfaat lain dari simulasi tersebut. Manfaat dari simulasi adalah untuk pelatihan, studi perilaku sistem dan mengurangi resiko kegagalan sistem. Maka dengan itu dibuatlah simulasi untuk pengereman kereta api. Paper dengan judul Pengembangan Aplikasi Fuzzy Logic Controller untuk Pengereman Kereta Api di Stasiun dan Simulasinya menggunakan *Fuzzy Control* yang didasarkan menggunakan *fuzzy* metode Mamdani dan *centroid* untuk melakukan perhitungan *fuzzy* dan keputusan hasil untuk pengereman kereta [2]. Paper ini hanya membahas pengereman 1 kereta api yang diuji cobakan dengan menggunakan 1 rel kereta yang hanya bergerak dan berhenti sampai di stasiun saja. Namun kenyataannya, terdapat 2 atau 3 rel kereta dan bisa juga terdapat 2 kereta yang akan menuju ke persimpangan rel kereta sehingga kasus penelitian yang telah dipakai sebelumnya belum mencerminkan yang ada di dunia nyata.

Terdapat juga *paper* penelitian dengan judul Implementasi Fuzzy Logic Controller Pada Sistem Pengereman Kereta Api dimana penelitian ini menggunakan hukum gaya yaitu hukum Newton dan juga menggunakan *fuzzy* metode Mamdani dan *centroid* untuk melakukan perhitungan *fuzzy* dan keputusan hasil oleh *fuzzy* [7]. Tetapi di penelitian ini hanya berupa kalkulasi pengereman kereta sehingga belum mendapatkan gambaran hasil apabila *fuzzy control* yang digunakan untuk pengereman kereta dengan pertimbangan gaya Newton sukses saat di uji coba dengan simulasi.

Oleh karena itu, skripsi ini bertujuan untuk menerapkan *Fuzzy Control* pada pengereman kereta api menggunakan hukum gaya dalam *simulation game* untuk lebih realistis. Selain itu juga dilakukan simulasi pengereman kereta agar yang diuji coba pada simulasi dapat digunakan saat di coba ke dunia nyata.

2. LANDASAN TEORI

2.1 Fuzzy Control

Fuzzy Control sangat sederhana secara konseptual. *Fuzzy Control* terdiri dari *input stage*, *processing stage*, dan *output stage*. *Input stage* merupakan sensor atau input lain seperti tombol, roda, dan seterusnya, sesuai dengan *membership function* beserta *value* tersebut. *Processing stage* mengambil setiap *rule*, dan menghasilkan jawaban dari tiap *rule* dan jawaban dari tiap *rule* digabungkan jadi satu jawaban. Dan yang terakhir, *Output stage* mengkonversi hasil gabungan jawaban tersebut kembali menjadi sebuah kontrol output yang spesifik. *Processing stage* merupakan koleksi dari *logic rules* dengan bentuk IF-THEN, dimana IF dinamakan *Antecedent* dan THEN dinamakan *Consequent* [6].

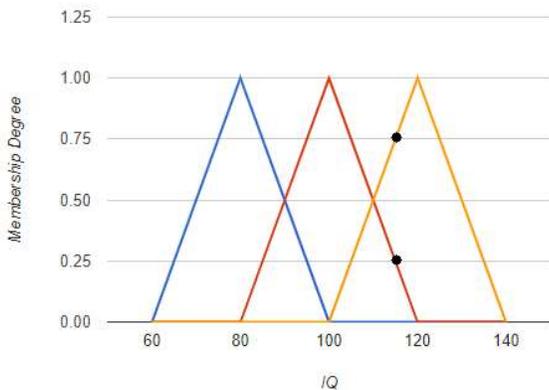
2.2 Fuzzy Logic

Fuzzy logic atau logika *fuzzy* adalah sebuah perkembangan dari logika Boolean oleh Lotfi Zadeh pada tahun 1965 berdasarkan teori

matematis *fuzzy set*, yang merupakan generalisasi dari teori himpunan klasik. Dengan memperkenalkan konsep sederajat dalam verifikasi suatu kondisi, yang memungkinkan kondisi berada di keadaan selain benar atau salah, logika *fuzzy* memberi fleksibilitas penalaran yang sangat berharga, memungkinkan untuk memperhitungkan ketidakakuratan dan ketidakpastian. Salah satu keuntungan logika fuzzy adalah kondisi-kondisi yang ada dalam seperangkat aturan ditetapkan dalam *Natural Language* [4]. Berikut merupakan beberapa contoh *Fuzzy Rules*:

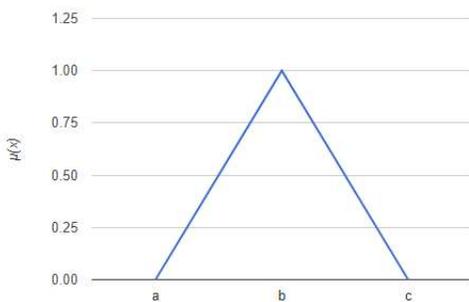
- IF lampu is merah AND kecepatan is tinggi AND jarak is dekat THEN rem is keras
- IF lampu is kuning AND kecepatan is rata-rata AND jarak is jauh THEN rem is halus

Dari contoh peraturan tersebut memang mudah diartikan oleh manusia, tetapi akan bagi komputer untuk mengartikan kata seperti “tinggi” atau “rata-rata” karena tidak memiliki batasan nilai yang pasti. Dari situlah *Fuzzy Logic* diperlukan untuk menempatkan suatu nilai lebih dari satu kategori dimana proses tersebut dilakukan oleh *Fuzzification*. Dari nilai yang telah difuzzifikasi, komputer dapat memproses nilai tersebut untuk mengambil keputusan pada peraturan – peraturan fuzzy (*Fuzzy Rules*) yang sudah ditentukan melalui proses untuk menghasilkan nilai *output* dalam bentuk *fuzzy* melalui proses *defuzzification*, untuk mengembalikan nilai tersebut dalam bentuk *crisp* [3].



Gambar 1. Contoh Fuzzy Set pembagian kategori Dumb, Average atau Clever berdasarkan IQ [3]

Gambar 1 merupakan contoh *fuzzy variable* yang terdiri dari 3 *fuzzy sets*, yaitu *Dumb*, *Average*, dan *Clever*. Titik tersebut mengartikan bahwa seseorang yang mempunyai IQ 115, adalah sebuah *member* dari 2 *fuzzy sets*. Bentuk *Fuzzy Sets* yang biasanya digunakan adalah *triangular* dan *trapezoidal*. *Trapezoidal* dibagi lagi menjadi bentuk *left-shoulder* dan *right-shoulder* [5].



Gambar 2. Triangular Fuzzy Set [5]

Gambar 2 merupakan *triangular fuzzy set* yang memiliki 3 titik koordinat untuk menunjukkan daerah hasil dari sebuah *Membership Function* [5].

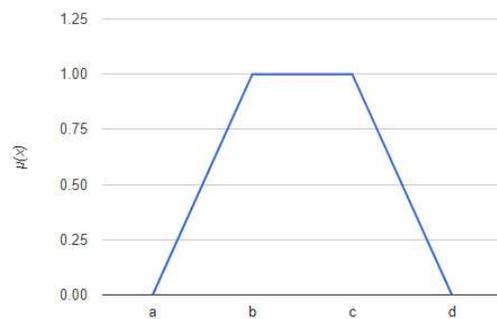
$$\mu(x) = 0, \text{ untuk } x < a \quad (1)$$

$$\mu(x) = \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b \quad (2)$$

$$\mu(x) = \frac{c-x}{c-b}, \text{ untuk } b < x \leq c \quad (3)$$

$$\mu(x) = 0, \text{ untuk } x > c \quad (4)$$

Rumus (1), (2), (3), dan (4) merupakan persamaan untuk mencari *value* dari *triangular fuzzy set* tersebut [5].



Gambar 3. Trapezoidal Fuzzy Set [5]

Gambar 3 merupakan *trapezoid fuzzy set* yang memiliki 4 titik koordinat untuk menunjukkan daerah hasil dari sebuah *Membership Function* [5].

$$\mu(x) = 0, \text{ untuk } x < a \quad (5)$$

$$\mu(x) = \frac{x-a}{b-a}, \text{ untuk } a \leq x \leq b \quad (6)$$

$$\mu(x) = 1, \text{ untuk } b < x \leq c \quad (7)$$

$$\mu(x) = \frac{d-x}{d-c}, \text{ untuk } c < x \leq d \quad (8)$$

$$\mu(x) = 0, \text{ untuk } x > d \quad (9)$$

Rumus (5), (6), (7), (8), dan (9) merupakan persamaan untuk mencari *value* dari *trapezoidal fuzzy set* tersebut [5].

2.3 Simulation Game

Simulation game atau *game* simulasi merupakan permainan berdasarkan segala suatu hal di dalam kehidupan nyata, sehingga hal yang bersifat realistik lebih banyak dijumpai di *game* tipe simulasi. *Game* dengan tipe simulasi menggambarkan tentang kehidupan sehari-hari, kehidupan pekerjaan serta pengaturan dan pengoperasian alat tertentu. Terciptanya ide *game* simulasi biasanya dilihat dari kehidupan sehari-hari [8].

2.4 Metode Mamdani

Metode Mamdani dikenal sebagai metode max-min. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975 [1].

Untuk mendapatkan output diperlukan 4 tahapan:

1. Pembentukan himpunan *fuzzy*: variabel *input* maupun *output* dibagi menjadi satu atau lebih himpunan
2. Aplikasi fungsi implikasi: fungsi implikasi yang digunakan adalah Min
3. Komposisi aturan: ada tiga metode yang digunakan dalam melakukan inferensi sistem *fuzzy*:
 - a. Metode Max
 - b. Metode *Additive* (SUM)
 - c. Metode Probabilistik OR
4. Penegasan (defuzzifikasi): *Input* dari defuzzifikasi adalah suatu himpunan yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Defuzzifikasi terdapat berbagai macam metode tetapi yang paling sering digunakan adalah metode *centroid* [4].

3. DESAIN SISTEM

3.1 Desain Game

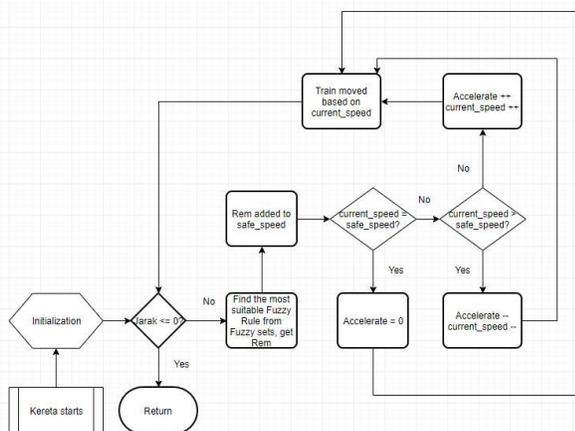
Judul *game* yang akan dibuat adalah “Train Simulator” dimana *game* ini dibuat menggunakan *Fuzzy Control* untuk mengatur pergerakan kereta. Hal ini memungkinkan kereta untuk dapat mengambil keputusan pengereman saat mencapai kondisi tertentu dengan *Fuzzy Rules* yang telah disesuaikan dengan keadaan yang nyata.

Game ini merupakan *simulation game* dimana pemain dapat melihat objek yang disimulasikan. Pemain juga dapat menggunakan *mouse* untuk pemilihan kereta yang diinginkan beserta menambah jumlah orang untuk memasuki kereta tersebut dan juga memulai kereta saat menekan tombol *start*.

Kereta akan memutuskan pengereman kereta beserta kecepatan kereta dimana saat mencapai titik tertentu akan memulai melakukan pengereman dengan mengurangi kecepatan hingga berhenti sampai di tujuan. Keputusan yang dilakukan telah sesuai dengan *rules* yang telah ditentukan untuk setiap keputusan pengereman

3.2 Alur Program

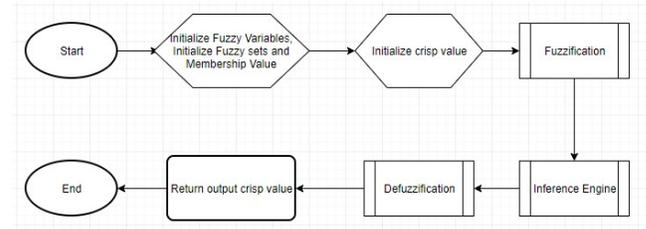
3.2.1 Alur Kereta



Gambar 4. Alur Kereta

Gambar 4 merupakan alur dari kereta yang mengatur kecepatan kereta tersebut. Program akan selalu melakukan pengecekan terhadap jarak kereta dengan tujuan hingga mencapai tujuan. Apabila kereta belum mencapai tujuan, maka program akan melakukan perhitungan untuk mencari rem di setiap titik menggunakan *Fuzzy Inference* untuk mengetahui kapan untuk melakukan rem saat diperlukan, dimana akan berpengaruh ke maksimum kecepatan kereta. Program juga akan melakukan untuk mengatur kecepatan kereta apabila melewati batas maksimal kecepatan kereta dengan memainkan percepatan. Kecepatan kereta akan selalu berubah saat memulai pengereman hingga sampai di tujuan.

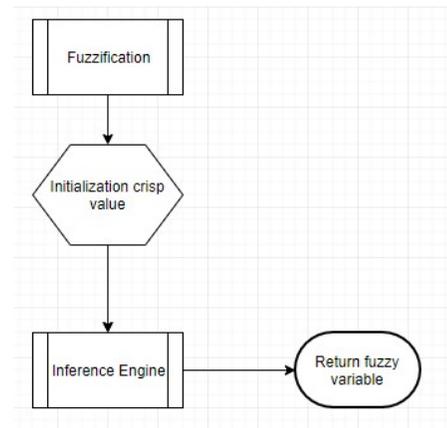
3.2.2 Alur Fuzzy Inference



Gambar 5. Alur Fuzzy Inference

Gambar 5 merupakan alur dari proses perhitungan nilai rem menggunakan *Fuzzy Inference*. Pada awalnya, program akan menginisialisasi *Fuzzy Variable* yang digunakan, termasuk *Fuzzy Variable* rem sebagai *variable output* yang menjadi tolok ukur nilai rem. Semua *Fuzzy Sets* beserta *Membership Value*-nya juga telah diinisialisasi oleh program. Setelah itu, program akan menerima input berupa *crisp value*. *Crisp value* akan dimasukkan ke dalam proses *Fuzzification* dimana menggunakan *Fuzzy Inference Engine* untuk menghasilkan *Fuzzy Variable* baru yang *Fuzzy Sets*-nya sesuai dengan masing-masing value *Fuzzy Sets*. Setelah proses *Fuzzification*, *Fuzzy Variable* tersebut akan dimasukkan kedalam proses *Defuzzification* dimana akan melakukan proses hingga menghasilkan sebuah *output crisp value* berupa nilai rem.

3.2.3 Alur Fuzzification

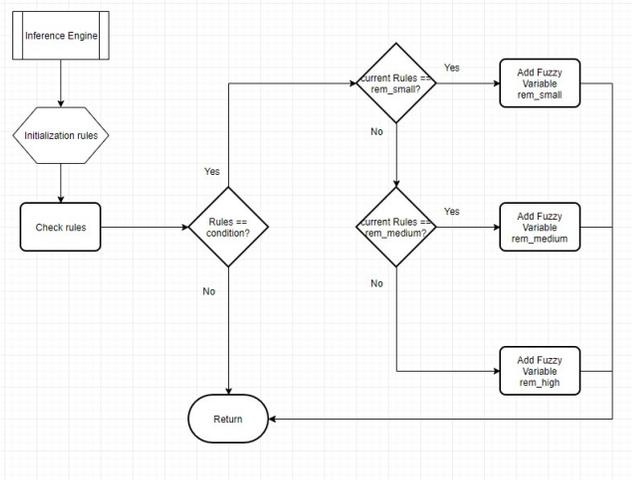


Gambar 6. Alur Fuzzification

Gambar 6 merupakan alur dari proses *Fuzzification*. Proses *Fuzzification* akan sangat berpengaruh untuk perhitungan nilai rem. Fungsinya adalah mengubah input *Crisp Value* menjadi *Fuzzy*

Value, lalu melakukan *Fuzzy Inference Engine* dengan mencari *Fuzzy Rules* yang paling cocok atau terdekat untuk menghasilkan *Fuzzy Variable* yang baru yang akan dipakai pada proses *Defuzzification*.

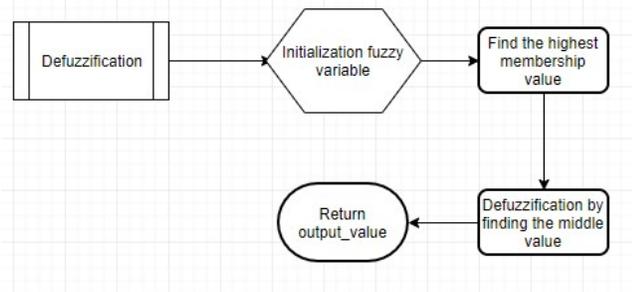
3.2.4 Alur Fuzzy Inference Engine



Gambar 7. Alur Fuzzy Inference Engine

Gambar 7 merupakan alur dari proses *Fuzzy Inference Engine*. Proses ini sangat penting dimana akan berpengaruh sekali terhadap hasil yang dilakukan oleh kereta. Proses ini akan memulai dengan menginisialisasi semua *rules* lalu akan melakukan pengecekan untuk semua *rules* terhadap kondisi. Apabila kondisi kereta cocok dengan kondisi *rules* tersebut maka akan memasukkan *Fuzzy Variable* sesuai dengan *rules* yang telah dibuat.

3.2.5 Alur Defuzzification



Gambar 8. Alur Defuzzification

Gambar 8 merupakan alur dari proses *Defuzzification*. Proses *Defuzzification* berguna untuk mengembalikan *Fuzzy Variable* yang telah didapat dari proses *Fuzzification* ke dalam bentuk *Crisp Value*. Dalam proses ini, metode yang digunakan adalah metode *centroid*, dimana akan mencari *membership value* lalu melakukan *defuzzification* menggunakan *centroid* dimana akan mencari nilai tengah dari *Fuzzy Variable* tersebut lalu akan mengembalikan ke dalam bentuk *Crisp Value*.

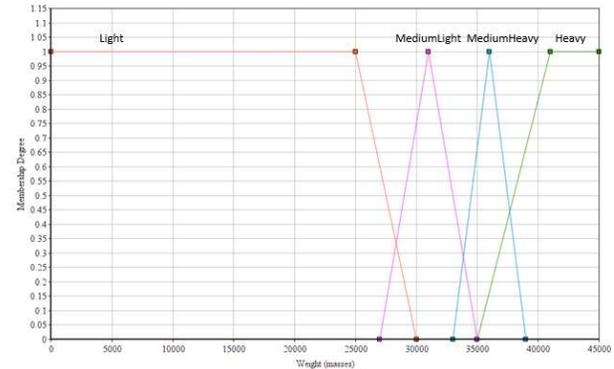
3.3 Desain Initial Membership Function AI

Dalam game ini, *Fuzzy Inference System* digunakan oleh kereta untuk melakukan pengereman kereta. Semua *Initial Membership*

Function yang digunakan oleh kereta untuk kedua hal tersebut berperan untuk mengatur keputusan pada AI.

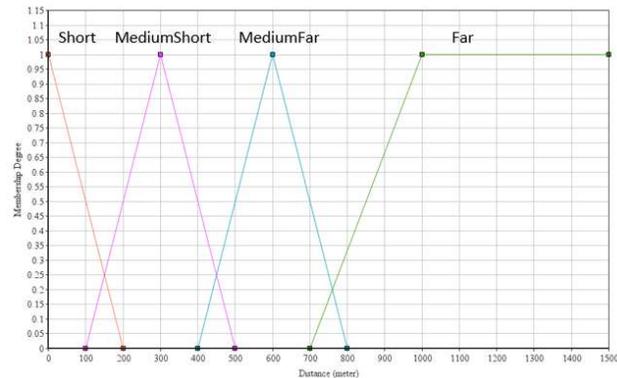
3.3.1 Initial Membership Functions

Fuzzy Variable yang digunakan sebagai *input* oleh kereta untuk menentukan terjadinya pengereman kereta adalah sebagai berikut.



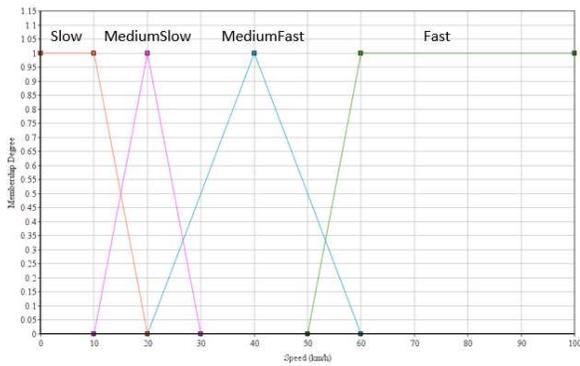
Gambar 9. Membership Function Weight

Gambar 9 merupakan desain dari *Membership Function Weight*. Daerah fungsi dalam *Membership Function* ini terbagi menjadi 4 yaitu *Light*, *Medium*, *MediumHeavy*, dan *Heavy*. Daerah fungsi *Light* berbentuk trapesium *Right Shoulder*, dengan batas nilai fungsi terdapat pada titik 0, 25000, dan 30000 *Masses*. Daerah fungsi *MediumLight* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 27000, 31000, dan 35000 *Masses*. Daerah fungsi *MediumHeavy* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 33000, 36000, dan 39000 *Masses*. Daerah fungsi *Heavy* berbentuk trapesium *Left Shoulder*, dengan batas nilai fungsi terdapat pada titik 35000, 41000, dan 45000 *Masses*.



Gambar 10. Membership Function Distance

Gambar 10 merupakan desain dari *Membership Function Distance*. Daerah fungsi dalam *Membership Function* ini terbagi menjadi 4 yaitu *Short*, *MediumShort*, *MediumFar*, dan *Far*. Daerah fungsi *Short* berbentuk trapesium *Right Shoulder*, dengan batas nilai fungsi terdapat pada titik 0, 0, dan 200 meter. Daerah fungsi *Medium Short* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 100, 300, dan 500 meter. Daerah fungsi *MediumFar* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 400, 600, dan 800 meter. Daerah fungsi *Far* berbentuk trapesium *Left Shoulder*, dengan batas nilai fungsi terdapat pada titik 700, 1000, dan 1500 meter.

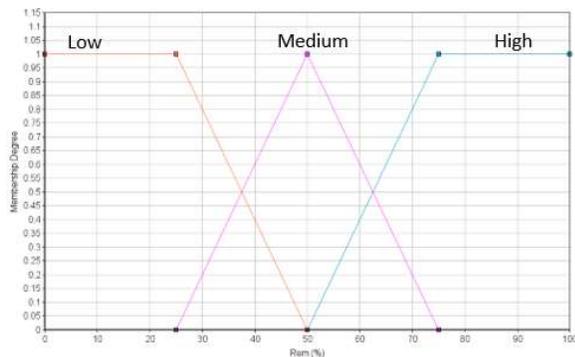


Gambar 11. Membership Function Speed

Gambar 11 merupakan desain dari *Membership Function Speed* Daerah fungsi dalam *Membership Function* ini terbagi menjadi 4 yaitu *Slow*, *MediumSlow*, *MediumFast*, dan *Fast*. Daerah fungsi *Slow* berbentuk trapesium *Right Shoulder*, dengan batas nilai fungsi terdapat pada titik 0, 10, dan 20 km/h. Daerah fungsi *MediumSlow* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 10, 20, dan 30 km/h. Daerah fungsi *MediumFast* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 20, 40, dan 60 km/h. Daerah fungsi *Fast* berbentuk trapesium *Left Shoulder*, dengan batas nilai fungsi terdapat pada titik 50, 60, dan 100 km/h.

3.3.2 Output Membership Function AI

Fuzzy Variable yang merupakan *output* dari *Fuzzy Inference Engine* adalah *Fuzzy Variable Rem*.



Gambar 12. Membership Function Rem

Gambar 12 merupakan desain dari *Membership Function Rem*. Daerah fungsi dalam *Membership Function* ini terbagi menjadi 3 yaitu *Small*, *Medium*, dan *High*. Daerah fungsi *small* berbentuk trapesium *Right Shoulder*, dengan batas nilai fungsi terdapat pada titik 0, 25, 50%. Daerah fungsi *Medium* berbentuk segitiga, dengan batas nilai fungsi terdapat pada titik 25, 50, 75%. Daerah fungsi *High* berbentuk trapesium *Left Shoulder*, dengan batas nilai fungsi terdapat pada titik 50, 75, 100%.

3.4 Desain Fuzzy Rules

Berikut ini merupakan *rules* yang diterapkan pada kereta untuk menentukan nilai Rem dari setiap titik tersebut:

- If Distance is Far then Brake is Low
- If Distance is MediumFar and Speed is Slow and Mass is Light then Brake is Low
- If Distance is MediumFar and Speed is Slow and Mass is MediumLight then Brake is Low

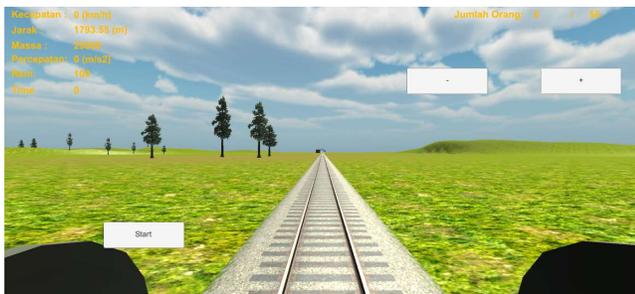
- If Distance is MediumFar and Speed is Slow and Mass is MediumHeavy then Brake is Low
- If Distance is MediumFar and Speed is Slow and Mass is Heavy then Brake is Low
- If Distance is MediumFar and Speed is MediumSlow and Mass is Light then Brake is Low
- If Distance is MediumFar and Speed is MediumSlow and Mass is MediumLight then Brake is Low
- If Distance is MediumFar and Speed is MediumSlow and Mass is MediumHeavy then Brake is Low
- If Distance is MediumFar and Speed is MediumSlow and Mass is Heavy then Brake is Low
- If Distance is MediumFar and Speed is MediumFast and Mass is Light then Brake is Low
- If Distance is MediumFar and Speed is MediumFast and Mass is MediumLight then Brake is Low
- If Distance is MediumFar and Speed is MediumFast and Mass is MediumHeavy then Brake is Low
- If Distance is MediumFar and Speed is MediumFast and Mass is Heavy then Brake is Medium
- If Distance is MediumFar and Speed is Fast and Mass is Light then Brake is Low
- If Distance is MediumFar and Speed is Fast and Mass is MediumLight then Brake is Low
- If Distance is MediumFar and Speed is Fast and Mass is MediumHeavy then Brake is Low
- If Distance is MediumFar and Speed is Fast and Mass is Heavy then Brake is Medium
- If Distance is MediumShort and Speed is Slow and Mass is Light then Brake is Low
- If Distance is MediumShort and Speed is Slow and Mass is MediumLight then Brake is Low
- If Distance is MediumShort and Speed is Slow and Mass is MediumHeavy then Brake is Low
- If Distance is MediumShort and Speed is Slow and Mass is Heavy then Brake is Low
- If Distance is MediumShort and Speed is MediumSlow and Mass is Light then Brake is Low
- If Distance is MediumShort and Speed is MediumSlow and Mass is MediumLight then Brake is Low
- If Distance is MediumShort and Speed is MediumSlow and Mass is MediumHeavy then Brake is Low
- If Distance is MediumShort and Speed is MediumSlow and Mass is Heavy then Brake is Medium
- If Distance is MediumShort and Speed is MediumFast and Mass is Light then Brake is Low
- If Distance is MediumShort and Speed is MediumFast and Mass is MediumLight then Brake is Medium
- If Distance is MediumShort and Speed is MediumFast and Mass is MediumHeavy then Brake is Medium
- If Distance is MediumShort and Speed is MediumFast and Mass is Heavy then Brake is Medium
- If Distance is MediumShort and Speed is Fast and Mass is Light then Brake is Medium
- If Distance is MediumShort and Speed is Fast and Mass is MediumLight then Brake is Medium
- If Distance is MediumShort and Speed is Fast and Mass is MediumHeavy then Brake is Medium
- If Distance is MediumShort and Speed is Fast and Mass is Heavy then Brake is Medium
- If Distance is Close then Brake is High

4. PENGUJIAN SISTEM

4.1 Pengujian Aplikasi

Pengujian Aplikasi ini dilakukan untuk mengetahui proses permainan dengan memainkan *game* yang telah dibuat. Pada saat program dijalankan, maka akan muncul tampilan *Main Menu*.

Pada tampilan *Main Menu*, terdapat 2 tombol yang dapat ditekan oleh pemain, yaitu tombol *Start Game* dan tombol *Quit Game*. Tombol *Start Game* berfungsi untuk memulai *game*, sedangkan tombol *Quit Game* berfungsi untuk keluar dari *Game*. Jika pemain memilih untuk memulai *game*, maka akan diarahkan ke tampilan pemilihan kereta dimana memilih kereta yang diinginkan. Setelah pemain telah memilih, pemain akan diarahkan pada Gambar 13.



Gambar 14. Tampilan *game*

Pada Gambar 14, program menampilkan dari sisi kereta dengan menampilkan kondisi kereta seperti kecepatan, jarak sampai di tujuan, massa kereta, percepatan kereta, dan kondisi rem kereta. Pemain dapat menambah jumlah orang dengan menekan tombol “+” dan juga mengurangi jumlah orang dengan menekan tombol “-” dimana jumlah orang akan mempengaruhi total massa kereta dan akan mempengaruhi kekuatan pengereman kereta. Terdapat juga tombol Start yang dapat ditekan oleh pemain untuk memulai kereta berjalan menuju tujuan. Jarak pada tujuan akan selalu berkurang hingga sampai di tujuan. Kecepatan, percepatan akan berubah saat terjadinya pengereman. Rem akan terjadi apabila kereta telah mendekati tujuan. Semakin besar nilai rem maka akan semakin kuat kekuatan pengereman dimana kecepatan akan semakin berkurang.

Saat kereta mulai mendekati di tujuan, maka kereta akan melakukan pengereman dengan mengurangi kecepatan yang dipengaruhi oleh percepatan dan rem seperti pada Gambar 14.



Gambar 15. Tampilan *game* dekat tujuan

Dapat dilihat pada Gambar 15 bahwa terjadi pengereman dan kereta mengurangi kecepatan sesuai dengan keadaan percepatan, hingga melakukan pengereman yang makin besar saat hampir mencapai tujuan.

Hasil pengujian dari program dapat dilihat pada Gambar 16 dimana kereta berhasil melakukan pengereman kereta berhenti sampai di tujuan.



Gambar 16. Tampilan *game* sampai tujuan

4.2 Pengujian *Fuzzy Control*

Pengujian ini dilakukan untuk mengetahui apakah *Fuzzy Control* yang dilakukan pada *game* ini sudah tepat atau tidak. Pengujian ini dilakukan dengan memainkan *game* yang telah dibuat, lalu menganalisa *output* yang didapat dari *Fuzzy Control* apakah *Fuzzy Rules* telah sesuai dipakai. Berikut adalah pengujian penerapan *Fuzzy Control*. Kecepatan, massa, dan jarak merupakan *Fuzzy input*, dimana akan menghasilkan *Fuzzy output* untuk Rem

Tabel 1. Hasil Pengujian Pertama *Fuzzy Control*

Time (Second)	Input			Output	Match with Fuzzy Rules?
	Kecepatan	Massa	Jarak	Rem	
0	0 Slow	25000 Light	1793.55 Far	100 High	Yes
15.711	100 Fast	25000 Light	1033.303 Far	0 Low	Yes
31.614	57 MediumFast	25000 Light	385.075 MediumShort	42.588 Medium	Yes
44.146	22 Slow	25000 Light	45.147 Short	77.427 High	Yes
63.602	0 Slow	25000 Light	0 Short	100 High	Yes

Tabel 1 merupakan hasil pengujian pertama berupa *input* dan *output* perhitungan nilai rem dengan setiap titik menggunakan *Fuzzy Control*. Data pengujian diambil berdasarkan durasi waktu berjalannya simulasi kereta. Saat kereta belum dijalankan maka kecepatan awal adalah 0. Setelah saat kereta telah berjalan dan berhenti mencapai tujuan, data terakhir pada Tabel 1, menunjukkan bahwa kereta telah berhenti di tujuan.

Tabel 2. Hasil Pengujian Kedua *Fuzzy Control*

Time (Second)	Input			Output	Match with Fuzzy Rules ?
	Kecepatan	Massa	Jarak	Rem	
0	0 Slow	35000 MediumHeavy	1793.55 Far	100 High	Yes
15.711	100 Fast	35000 MediumHeavy	1033.303 Far	0 Low	Yes
32.411	24 MediumSlow	35000 MediumHeavy	384.394 MediumShort	75 Medium	Yes
55.409	22 Slow	35000 MediumHeavy	45.322 Short	77.339 High	Yes
74.714	0 Slow	35000 MediumHeavy	0 Short	100 High	Yes

Tabel 2 merupakan hasil pengujian kedua berupa *input* dan *output* perhitungan nilai rem dengan setiap titik menggunakan *Fuzzy Control*. Data pengujian diambil berdasarkan durasi waktu berjalannya simulasi kereta. Saat kereta belum dijalankan maka kecepatan awal adalah 0. Setelah saat kereta telah berjalan dan berhenti mencapai tujuan, data terakhir pada Tabel 2, menunjukkan bahwa kereta telah berhenti di tujuan. Perbedaan antara Tabel 1 dan Tabel 2 adalah massa kereta pada pengujian pertama lebih ringan dari pada pengujian kedua, dimana akan mempengaruhi kekuatan pengereman kereta, dimana juga akan memakan waktu lebih lama untuk mencapai di tujuan.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil pengujian dari Bab 5, dapat disimpulkan beberapa hal sebagai berikut:

- Kereta dapat berhenti di tujuan dengan melakukan pengereman yang dilakukan oleh *Fuzzy Control* dengan perlahan hingga berhenti di tujuan.
- Setiap keputusan yang diambil oleh kereta dilakukan dengan mengubah seluruh *parameter input* dari bentuk *crisp value* ke *fuzzy value*, lalu diolah oleh *Fuzzy Inference Engine* untuk menghasilkan *output* yang sesuai.

- Kereta belum dapat melakukan perbelokkan ataupun pergantian rel karena kesulitan dalam implementasi perbelokkan rel dimana juga berpengaruh pada pergantian rel.

5.2 Saran

Beberapa saran untuk pengembangan aplikasi selanjutnya adalah:

- *Game* ini dapat dikembangkan lagi untuk beberapa *platform* yang lain seperti pada *mobile device* dengan Android atau iOS dengan menggunakan Oculus Rift sehingga lebih nyata
- Menambah *Fuzzy Inference System* untuk kondisi lain seperti perbelokkan atau pergantian rel.

6. REFERENCES

- [1] Abrori, M. & Prihamayu, M.A. 2015. Aplikasi Logika Fuzzy Metode Mamdani dalam Pengambilan Keputusan Penentuan Jumlah Produksi. Yogyakarta: Universitas Gadjah Mada.
- [2] Bria, Y.P. 2012. Pengembangan Aplikasi Fuzzy Logic Controller Untuk Pengereman Kereta Api di Stasiun dan Simulasinya. Yogyakarta: Universitas Katolik Widya Mandira.
- [3] Buckland, M. 2005. Programming Game AI by Example. Plano: Wordware Publishing, Inc.
- [4] Dernoncourt, F. 2013. Introduction to Fuzzy Logic. Cambridge: Massachusetts Institute of Technology.
- [5] Lidianto, S.V., Budhi, G.S. & Intan, R. 2016. Perancangan dan Pembuatan Action Game dengan Artificial Intelligence dan Machine Learning. Surabaya: Universitas Kristen Petra.
- [6] Vijayaraghavan, G. & Jayalakshmi, M. 2015. A Quick Review on Applications of Fuzzy Logic in Waste Water Treatment. International Journal for Research in Applied Science & Engineering Technology, Vol 3 Issue V, pp 421-425.
- [7] Wahyudi. 2005. Implementasi Fuzzy Logic Controller pada Sistem Pengereman Kereta Api. Semarang: Jurusan Teknik Elektro – FT Undip.
- [8] Wijaya, E.S.K. 2013. Pembuatan Game Simulasi Perkebunan, Peternakan, dan Interaksi Sosial. Surabaya: Universitas Kristen Petra.