

PENGEMBANGAN SISTEM KENDALI PERGERAKAN *AUTONOMOUS MOBILE ROBOT* UNTUK MENDAPATKAN JALUR BEBAS HAMBATAN MENGGUNAKAN *FUZZY LOGIC CONTROLLER*

Faikul Umam

Program Studi Mekatronika, Fakultas Teknik, Universitas Trunojoyo Madura
Jl. Raya Telang, PO BOX 2 Kamal, Bangkalan
E-mail : faikul@yahoo.com, faycool.befreesource@gmail.com

ABSTRAK

Autonomous mobile robot merupakan sebuah *robot* yang dapat bernavigasi secara mandiri tanpa bantuan manusia. *Robot* jenis ini dapat melakukan tugasnya dengan baik karena telah memiliki kecerdasan yang menyerupai kecerdasan manusia. Kecerdasan buatan yang ditanamkan pada *autonomous mobile robot* akan membantu *robot* untuk mengambil keputusan serta beradaptasi pada area kerjanya, misalnya untuk menemukan jalur bebas hambatan. Penelitian ini membahas pengembangan sistem pengendalian pergerakan *robot* untuk mencari jalur bebas hambatan, artinya *robot* harus dapat dikendalikan agar tidak mengalami benturan terhadap beberapa halangan (*multiple obstacle*) yang ada didepannya. Untuk dapat melakukan hal tersebut, *robot* diberi kecerdasan buatan menggunakan *Fuzzy Logic Controller*. Dalam simulasi, *robot* 100% berhasil mendapatkan jalur bebas hambatan dengan cara menghindari beberapa halangan yang bergerak.

Kata kunci: Jalur Bebas Hambatan, Halangan bergerak, *Fuzzy Logic Controller*.

ABSTRACT

Autonomous mobile robot is a robot that can navigate autonomously without human assistance. The robot can finished his job do a good job because it has intelligence that ressembler human intelligence. Artificial intelligence is embedded in an autonomous mobile robot will help the robot to make decisions and adapt to the work area, for example to find a freeway. In this study discusses the development of a robot motion control system to search a freeway, it means the robot must be controlled to avoid many obstacle. To do it, the robot was given the artificial intelligence using *Fuzzy Logic Controller*. In simulation, the robot 100% managed to get a freeway by avoiding several moving obstacles.

Keywords: Freeway, Multiple Obstacle, *Fuzzy Logic Controller*

1. Pendahuluan

Robotika merupakan sebuah bidang multidisiplin ilmu yang mempelajari tentang pembuatan sebuah mesin untuk meringankan pekerjaan manusia. Beberapa bidang ilmu yang terkait dalam pembuatan robot antara lain ilmu mekanik yang mempelajari tentang konstruksi robot, ilmu elektronika untuk sistem elektronik robot, dan kecerdasan buatan sebagai otak robot.

Sebenarnya istilah robot berasal dari kata “robota” yang berarti tenaga kasar atau pelayan. Awalnya robot hanya digunakan pada aplikasi industri sehingga muncul istilah robot industri atau *robot manipulator (robot arm)*. Robot ini memiliki konstruksi mirip seperti lengan manusia. Pada saat itu, *robot arm* hanya terbatas pada sebuah mesin dengan konstruksi lengan-lengan kaku yang terhubung secara seri yang dapat bergerak memutar (rotasi) dan memanjang atau memendek (translasi atau prismatik). Robot jenis ini berfungsi untuk melakukan pekerjaan berat yang membutuhkan tingkat ketelitian tinggi dengan tujuan meningkatkan proses produksi industri, seperti merakit mobil, memindahkan bahan yang berat, dan sebagainya. Namun seiring perkembangan jaman kemudian dikenal istilah *mobile robot, humanoid, animaloid*, dan sebagainya, fungsinya juga mengalami perubahan yang tadinya untuk keperluan industri sekarang juga untuk bidang kedokteran, militer bahkan untuk hiburan [1].

Penelitian dan pengembangan robot pertama kali dilakukan oleh Argonne National Laboratories di Oak Ridge Amerika pada tahun 1940-an untuk menangani material radioaktif bernama *master-slave manipulator*. Saat ini teknologi robot industri mulai tertinggal perkembangannya dengan teknologi *mobile robot*. Teknologi *mobile robot* digunakan secara luas sejak tahun 80-an, aplikasinya tidak lagi berkaitan dengan proses produksi pada industri, tetapi lebih diarahkan untuk membantu tugas manusia diberbagai bidang, misalnya *Robot Pioneer* digunakan untuk misi ke Chernobyl, *Robot Sojourner* digunakan untuk misi ke Mars, *Tour guide robot* yang merupakan jenis robot yang berfungsi untuk menghindari halangan [2].

Robot jalur bebas hambatan merupakan sebuah robot yang diberi kecerdasan agar dapat memilih jalur aman dalam melakukan pergerakan. Dalam proses pergerakannya, robot harus dapat mendeteksi adanya beberapa halangan dan menghindari halangan tersebut. Kecerdasan yang ditanamkan pada robot, harus dapat memilih jalur mana yang akan di tempuh agar robot tidak mengalami benturan. Pada tahun 1994, algoritma *Fuzzy* digunakan sebagai sistem kendali untuk menghindari halangan, tetapi halangan yang digunakan masih bersifat statis (diam) dan masih terpaku pada satu halangan saja [3].

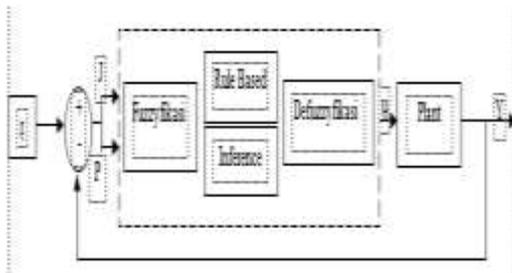
Sistem kendali menggunakan *Fuzzy* juga digunakan untuk navigasi *mobile robot* berbasis ponsel [4], sayangnya dalam penelitian tersebut juga masih menggunakan satu halangan saja. kombinasi antara *Behavior Based Control* dengan pembelajaran *Fuzzy Q-learning* digunakan untuk menyelesaikan permasalahan navigasi pada lingkungan yang tidak terstruktur dan kompleks dengan posisi target yang tidak diketahui [5].

Berdasarkan hal tersebut, pada penelitian ini dikembangkan sebuah sistem kendali menggunakan *Fuzzy Logic Controller* untuk menghasilkan jalur bebas hambatan, dimana jalur tersebut terdapat halangan bergerak dan halangan diam. Metode kecerdasan sebagai sistem kendalinya harus mampu menentukan pergerakan untuk menghindari beberapa halangan bergerak (*multiple obstacle*) pada area kerja dan mampu mencapai target yang sudah ditentukan. Alasan pemilihan *Fuzzy Logic Controller* sebagai sistem kendali karena *Fuzzy* dapat memodelkan fungsi-fungsi *nonlinier* yang sangat kompleks menjadi bahasa yang mudah dimengerti, selain itu *Fuzzy* juga memiliki proses komputasi yang sangat cepat [6] [7].

2. Metode

Fuzzy Logic Controller merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaan antara benar dan salah. Pada teori logika *Fuzzy*, sebuah nilai bisa berilai benar dan salah secara bersamaan, namun beberapa besar kebenaran dan kesalahan suatu nilai tergantung pada bobot keanggotaan yang dimilikinya [8].

Seperti pada Gambar 1, proses pemecahan masalah menggunakan *Fuzzy* diawali dengan membuat *membership function* (fungsi keanggotaan) dan kemudian dilakukan *fuzzyfikasi*. *Fuzzyfikasi* adalah proses perubahan nilai *real* yang ada kedalam *membership function*. Proses selanjutnya membuat tabel *rule base* yakni proses yang merelasikan (implikasi) antara *membership function* A dan *membership function* B hingga memiliki sebuah *output*. Proses relasi secara umum menggunakan aturan *if-then*. Model penalaran pada proses implikasi yang banyak digunakan adalah model min-max. Proses terakhir adalah *defuzzyfikasi*, dalam proses ini metode yang banyak digunakan adalah metode Max, metode Titik Tengah, metode rata-rata dan sebagainya.



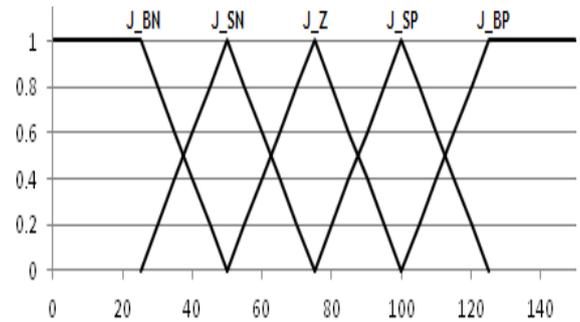
Gambar 1. Diagram blok *fuzzy logic controller* penghindar halangan

Komponen *fuzifikasi* berfungsi untuk memetakan masukan data tegas atau nilai riil ke dalam himpunan *Fuzzy* atau lebih dikenal dengan nama *membership function*, misalnya sangat dekat, dekat dan jauh. Setiap variabel yang menjadi acuan dalam suatu proses harus dibuat *membership function*, misalnya variabel yang digunakan untuk menghindari halangan adalah jarak dan posisi halangan, maka jarak dan posisi harus dipetakan menjadi *membership function*. Ada beberapa tipe *membership function* yang sering digunakan, yakni tipe trapezium atau trapezoidal, tipe segitiga, dan tipe Gaussian.

A. Sistem kendali untuk menemukan jalur bebas hambatan

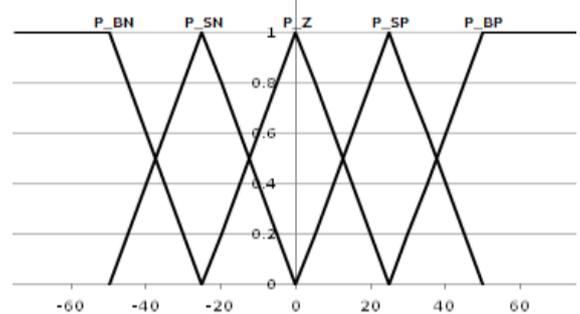
Pada penelitian ini, untuk menghasilkan jalur bebas hambatan, robot harus dapat menghindari beberapa halangan. variabel yang digunakan untuk menghindari halangan adalah jarak dan posisi halangan. Untuk itu dibuat *membership function* jarak dan posisi menggunakan tipe segitiga.

Seperti pada Gambar 2, variabel Jarak dipetakan menjadi 5 *membership function* yakni J_BN yaitu *Big Negatif*, ini berarti jarak halangan sangat dekat, J_SN yaitu *Small Negatif* yang berarti jarak halangan dekat, J_Z yaitu *Zero*, berarti jarak halangan sedang, J_SP yaitu *Small Positif*, berarti jarak halangan jauh, dan J_BP yaitu *Big Positif*, berarti jarak halangan sangat jauh.



Gambar 2. *Membership function* jarak

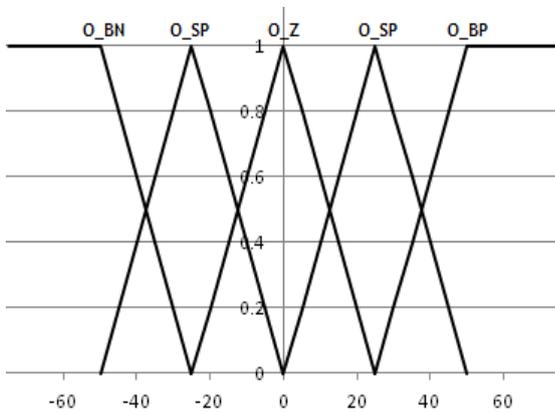
Variabel Posisi juga dipetakan menjadi 5 *membership function* Seperti pada Gambar 3, antara lain P_BN yaitu *Big Negatif*, ini berarti posisi halangan berada di sangat kiri robot, P_SN yaitu *Small Negatif* yang berarti posisi halangan berada di sebelah kiri robot, P_Z yaitu *Zero*, berarti posisi halangan tepat di depan robot, P_SP yaitu *Small Positif*, berarti posisi halangan berada di sebelah kanan robot, dan J_BP yaitu *Big Positif*, berarti posisi halangan berada di sangat kanan robot.



Gambar 3. *Membership function* posisi

Variabel *output* merupakan manuver atau pergerakan robot untuk menghindari halangan. Manuver robot juga dipetakan menjadi 5 *membership function* atau fungsi keanggotaan seperti pada Gambar 4, yakni O_BN yaitu *Big Negatif*, ini berarti robot bergerak sangat ke kiri depan, P_SN yaitu *Small Negatif* yang berarti robot bergerak

sedikit kekiri depan, P_Z yaitu Zero berarti robot bergerak lurus, P_SP yaitu Small Positif, berarti robot bergerak sedikit kanan depan, dan J_BP yaitu Big Positif, robot bergerak sangat kekanan depan.



Gambar 4. Membership Function Output

Setiap membership function harus dicari nilai derajat keanggotannya dengan persamaan (1), dimana μ_A adalah nilai derajat keanggotaan, x adalah :

$$\mu_A(x) = 1 - \frac{\sqrt{(x-a)^2}}{b} \quad (1)$$

Rule base atau basis aturan merupakan kumpulan aturan-aturan kendali Fuzzy yang dibuat berdasarkan pengetahuan manusia untuk menghubungkan antara variabel masukan dan variabel output. Kemudian berdasarkan rule base yang telah dibuat, variabel masukan Fuzzy diolah lebih lanjut untuk mendapatkan sebuah penyelesaian. Dengan demikian dapat diambil suatu keputusan berupa output Fuzzy, yaitu himpunan-himpunan output Fuzzy dengan fungsi keanggotaan yang telah ditetapkan. Pada penelitian ini terdapat 25 rule base atau basis aturan seperti pada Tabel 1.

Berdasarkan Tabel 1, setelah mengimplikasikan masing-masing membership function dihasilkan 25 variasi rule, tetapi pada dasarnya hanya ada 5 variasi Output yang dihasilkan, yaitu BN (Big Negatif) atau robot melakukan manuver sangat ke kiri, SN (Small Negatif) atau robot melakukan manuver ke kiri, Z (Zero) robot melakukan tetap berjalan lurus ke depan, SP (Small Positif) robot melakukan manuver ke kanan, dan BP yaitu (Big Positif) robot melakukan manuver sangat ke kanan.

Tabel 1. Rule Base Untuk Kendali Penghindar Halangan

Rule No.	Jarak	Posisi	Output
1	Sangat Dekat (J_BN)	Sangat Kiri (P_BN)	O_SP
2	Sangat Dekat (J_BN)	Kiri (P_SN)	O_BP
3	Sangat Dekat (J_BN)	Tengah (P_Z)	O_BP
4	Sangat Dekat (J_BN)	Kanan (P_SP)	O_BN
5	Sangat Dekat (J_BN)	Sangat Kanan (P_BN)	O_SN
6	Dekat (J_SN)	Sangat Kiri (P_BN)	O_SP
7	Dekat (J_SN)	Kiri (P_SN)	O_BP
8	Dekat (J_SN)	Tengah (P_Z)	O_BN
9	Dekat (J_SN)	Kanan (P_SP)	O_BN
10	Dekat (J_SN)	Sangat Kanan (P_BN)	O_SN
11	Sedang (J_Z)	Sangat Kiri (P_BN)	O_SP
12	Sedang (J_Z)	Kiri (P_SN)	O_SP
13	Sedang (J_Z)	Tengah (P_Z)	O_SP
14	Sedang (J_Z)	Kanan (P_SP)	O_SN
15	Sedang (J_Z)	Sangat Kanan (P_BN)	O_SN
16	Jauh (J_SP)	Sangat Kiri (P_BN)	O_Z
17	Jauh (J_SP)	Kiri (P_SN)	O_Z
18	Jauh (J_SP)	Tengah (P_Z)	O_SP
19	Jauh (J_SP)	Kanan (P_SP)	O_Z
20	Jauh (J_SP)	Sangat Kanan (P_BN)	O_Z
21	Sangat Jauh (J_BP)	Sangat Kiri (P_BN)	O_Z
22	Sangat Jauh (J_BP)	Kiri (P_SN)	O_Z
23	Sangat Jauh (J_BP)	Tengah (P_Z)	O_Z
24	Sangat Jauh (J_BP)	Kanan (P_SP)	O_Z
25	Sangat Jauh (J_BP)	Sangat Kanan (P_BN)	O_Z

Setelah mekanisme pertimbangan dilakukan, langkah terakhir adalah defuzzyfikasi. Defuzzyfikasi merupakan proses perubahan nilai Fuzzy ke sinyal yang bersifat bukan Fuzzy atau nilai riil. Beberapa metode defuzzyfikasi yang sering digunakan antara lain Center Of Area (COA) dan Maximum Of Mean (MOM). Pada penelitian ini metode yang digunakan adalah Center Of Area (COA).

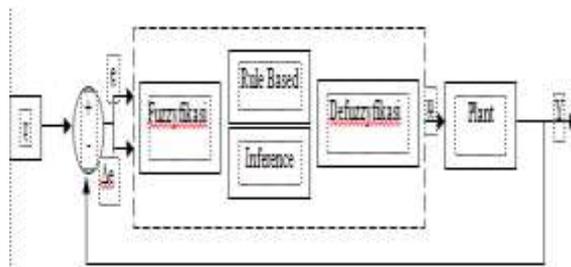
$$v_o = \frac{\sum_{k=1}^m v_k \mu_k(v_k)}{\sum_{k=1}^m \mu_k(v_k)} \quad (2)$$

dimana :

- v_o : nilai keluaran
- m : tingkat kuantisasi
- v_k : elemen ke k
- $\mu_v(v_k)$: derajat keanggotaan elemen pada fuzzy set v
- v : semesta pembicaraan

B. Sistem kendali untuk mencapai ketarget

Saat awal robot berjalan, yang dihitung pertama kali adalah posisi (sudut) target sebagai tujuan utama. Robot akan selalu menghitung agar arah hadap robot terhadap target atau sudut target sama dengan 0. Algoritma yang digunakan untuk melakukan hal tersebut juga menggunakan *Fuzzy*.

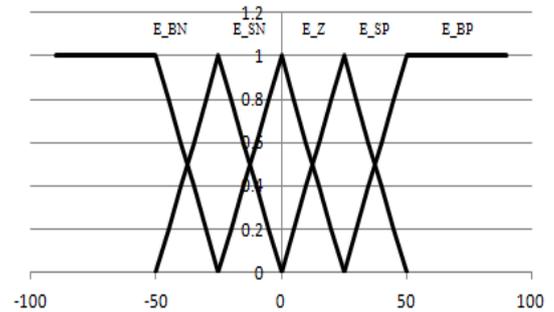


Gambar 5. Diagram blok *fuzzy logic controller* mencapai target

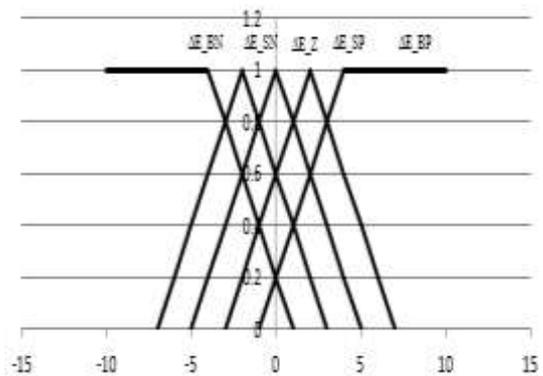
Seperti pada Gambar 5, untuk menuju target, robot akan selalu menyesuaikan arah hadap robot agar selalu tegak lurus terhadap target. Dalam pergerakan menuju target, robot mulai menemukan hambatan, tentu saja robot harus bergerak untuk menghindarinya, saat itu pula sudah pasti arah hadap robot mulai berubah terhadap posisi target. Sistem kendali ini berfungsi untuk memperbaiki posisi target utama agar sama dengan 0, dalam hal ini, sistem kendali menggunakan *error* sudut dan *delta error* sebagai variabel utama untuk memperbaiki nilai sudut target agar sama dengan 0. Untuk itu, *error* sudut dan *delta error* harus diterjemahkan terlebih dahulu menjadi *membership function*.

Seperti pada Gambar 6. *Membership function error* sudut menggunakan tipe segitiga dan dipetakan menjadi 5 *Membership Function* antara lain E_{BN} yaitu *Big Negatif*, artinya arah hadap robot sangat ke kanan dari

pandangan semula yang tegak lurus terhadap target, O_{SN} yaitu *Small Negatif* artinya arah hadap robot ke kanan dari pandangan semula, E_Z yaitu *Zero*, artinya arah hadap robot masih sama seperti semula yaitu tegak lurus terhadap target, E_{SP} yaitu *Small Positif*, artinya arah hadap robot ke kiri dari pandangan semula dan E_{BP} yaitu *Big Positif*, artinya arah hadap robot sangat ke kiri dari pandangan semula.



Gambar 6. *Membership function error* sudut



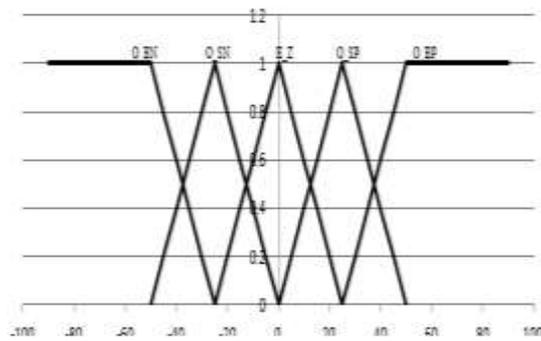
Gambar 7. *Membership function delta error* sudut

Membership function Delta *error* sudut juga menggunakan tipe segitiga dan dipetakan menjadi 5 keanggotaan seperti pada Gambar 5 keanggotaan seperti pada Gambar 7, antara lain DE_{BN} , DE_{SN} , DE_Z , DE_{SP} , dan DE_{BP} . *Membership function Output* juga dipetakan menjadi 5 keanggotaan seperti pada Gambar 8, antara lain O_{BN} artinya jika arah hadap robot sangat ke kanan, maka robot akan diperintahkan untuk mengubah arah hadapnya atau belok kiri sampai arah hadap robot kembali tegak lurus terhadap target, begitu juga dengan O_{SN} . O_Z artinya arah hadap robot masih sama seperti semula, sedangkan untuk O_{SP} dan O_{BP} prinsipnya sama dengan O_{SN} dan O_{BN} . Untuk lebih jelasnya, mekanisme pencarian dan

pencapaian target direpresentasikan dalam bentuk *rule base* seperti pada Tabel 2.

Tabel 2. *Rule base* untuk pencapaian ke target

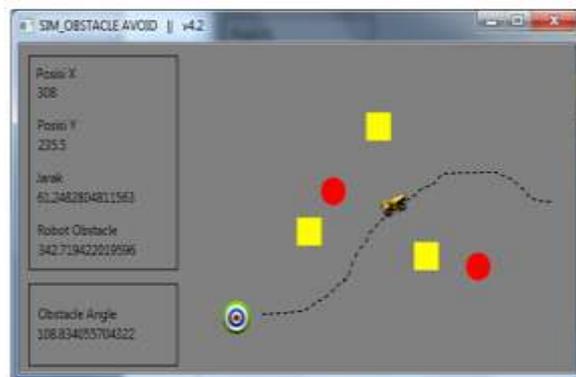
Rule No.	Jarak	Posisi	Output
1	Sangat Kiri (E_BN)	Sangat Besar_Negatif (DE_BN)	O_BP
2	Sangat Kiri (E_BN)	Besar_Negatif (DE_BN)	O_BP
3	Sangat Kiri (E_BN)	Nol (DE_Z)	O_SP
4	Sangat Kiri (E_BN)	Besar Positif (DE_SP)	O_SP
5	Sangat Kiri (E_BN)	Sangat Besar Positif (DE_BP)	O_SP
6	Kiri (E_SN)	Sangat Besar_Negatif (DE_BN)	O_BP
7	Kiri (E_SN)	Besar_Negatif (DE_BN)	O_SP
8	Kiri (E_SN)	Nol (DE_Z)	O_SP
9	Kiri (E_SN)	Besar Positif (DE_SP)	O_SP
10	Kiri (E_SN)	Sangat Besar Positif (DE_BP)	O_SP
11	Tengah (E_Z)	Sangat Besar_Negatif (DE_BN)	O_SP
12	Tengah (E_Z)	Besar_Negatif (DE_BN)	O_Z
13	Tengah (E_Z)	Nol (DE_Z)	O_Z
14	Tengah (E_Z)	Besar Positif (DE_SP)	O_Z
15	Tengah (E_Z)	Sangat Besar Positif (DE_BP)	O_SN
16	Kanan (E_SP)	Sangat Besar_Negatif (DE_BN)	O_BN
17	Kanan (E_SP)	Besar_Negatif (DE_BN)	O_SN
18	Kanan (E_SP)	Nol (DE_Z)	O_SN
19	Kanan (E_SP)	Besar Positif (DE_SP)	O_SN
20	Kanan (E_SP)	Sangat Besar Positif (DE_BP)	O_SN
21	Sangat Kanan (E_BP)	Sangat Besar_Negatif (DE_BN)	O_BN
22	Sangat Kanan (E_BP)	Besar_Negatif (DE_BN)	O_BN
23	Sangat Kanan (E_BP)	Nol (DE_Z)	O_SN
24	Sangat Kanan (E_BP)	Besar Positif (DE_SP)	O_SN
25	Sangat Kanan (E_BP)	Sangat Besar Positif (DE_BP)	O_SN



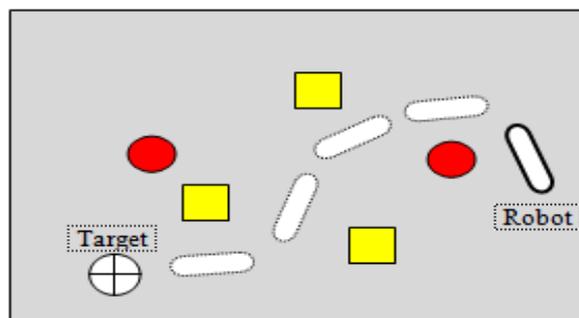
Gambar 8. *Membership function output*

3. Hasil Dan Pembahasan

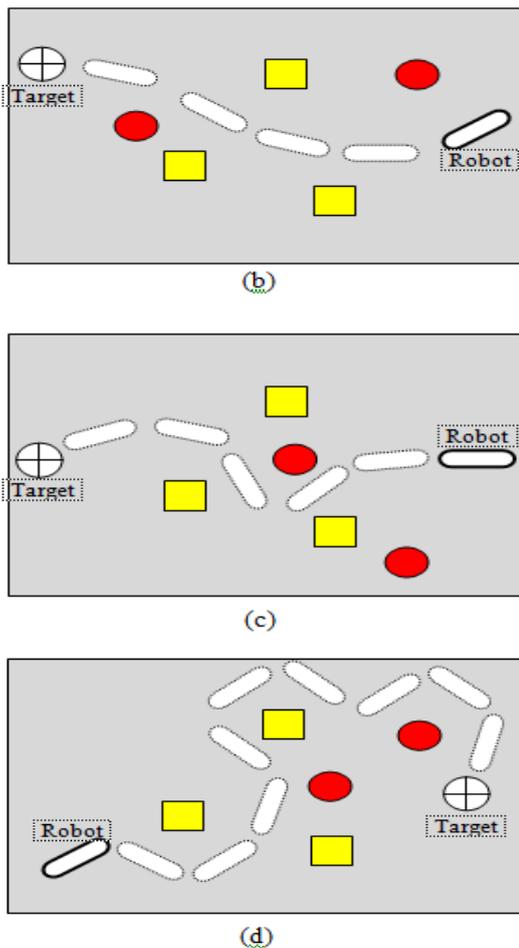
Software simulasi dibuat menyerupai kondisi sebenarnya. Seperti Gambar 9, ada dua jenis halangan yang digunakan, yaitu halangan yang bersifat diam dan halangan yang bergerak. Kedua jenis halangan tersebut masing-masing dipasang 2 buah halangan untuk halangan bergerak dan 3 buah untuk halangan diam. Pada *software* simulasi halangan bergerak diwakili oleh lingkaran yang bergerak secara vertikal atau bergerak keatas kebawah dan horisontal atau bergerak kekanan kekiri secara terus menerus dengan kecepatan tertentu, sedangkan halangan diam di pasang secara acak.



Gambar 10. *Software simulasi*



(a)



Gambar 11. (a), (b), (c), (d) Ilustrasi saat robot menghindari halangan

Software simulasi dibuat menggunakan bahasa pemrograman C# seperti pada Gambar 10. Untuk mengetahui seberapa besar tingkat keberhasilan robot dalam melakukan tugasnya, percobaan dilakukan berulang-ulang dengan posisi target yang berbeda seperti pada Gambar 11. Dari beberapa kali percobaan, robot selalu bisa menghindari halangan dan berhasil mencapai target utama tanpa mengalami benturan. Pada awal pergerakannya, robot akan menyesuaikan arah hadap terhadap target, tetapi ketika ada halangan, robot mulai menghitung jarak (dalam pixel) dan posisi halangan tersebut untuk menentukan kemana robot akan menghindar, jika pada saat menghindar ternyata masih ada halangan lain yang datang, maka robot akan tetap menghitung jarak dan posisi halang tersebut untuk melakukan aksi berikutnya, tetapi pada saat yang sama robot tidak mengabaikan tugas utamanya, oleh karena itu, robot akan selalu memperbaiki posisi target utama dan

menyesuaikan arah hadap terhadap target yang sudah ditentukan.

4. Kesimpulan

Dari penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa trayektori robot mengalami perubahan saat jarak dan posisi halangan dapat teridentifikasi, *Fuzzy Logic Controller* sebagai sistem kendalinya mengolah informasi tersebut untuk menghasilkan keputusan dalam hal pemilihan jalur bebas hambatan. Kecepatan komputasi dan penghitungan menggunakan *Fuzzy* dalam hal mengolah informasi sangat baik, ini terbukti robot selalu dapat menghindari beberapa halangan sekaligus tanpa mengalami benturan. Saat menghindari halangan, sistem kendali lain (sistem kendali untuk mencapai target) yang juga menggunakan *Fuzzy Logic Controller* juga bekerja sangat baik, meskipun banyak halangan yang datang, sistem kendali ini selalu dapat menyesuaikan arah hadap robot terhadap target, sehingga robot tidak pernah kehilangan target dan selalu bisa sampai pada target yang telah ditentukan.

Daftar Pustaka

- [1] Fu K.S, Gonzales R.C, Lee C.S.G.[1987], *Robotics Control, Sensing, Vision, And Intelligence*. McGraw-Hill Book Company.
- [2] Siegwart, R. Nourbakhsh, Illah, R. [2004], *Introduction to Autonomous Mobile Robots*. The MIT Press Cambridge.
- [3] Reignier Patrick.[1994], "Fuzzy Logic Techniques For Mobile Robot Obstacle Avoidance", *Sincedirect Robotics and Autonomous Systems*, 12 143-153.
- [4] Leyden, M. Toal, D. and Flanagan, C. [2006], *A Fuzzy Logic Based Navigation System for a Mobile Robot*, Department of Electronic & Computer Engineering, University of Limerick, Ireland.
- [5] Anam, K.[2008], *Skema Behavior-Based Control Dengan Pembelajaran Fuzzy Q-Learning Untuk Sistem Navigasi Autonomous Mobile Robot*,

Master Theses of Electrical Engineering, Insitut Teknologi Sepuluh Nopember, Surabaya.

- [6] Ovid, F. Yordan, C.[2008], “Optimal Fuzzy Control of Autonomous Robot Car”, *4th IEEE Conference Intelligent Systems*, 4-62 – 4-65.
- [7] Feng, H.M. Chen, C.Y. Horng, J.H.[2010], “Intelligent Omni-Directional Vision-Based Mobile Robot Fuzzy Systems Design And Implementation”, *Sincedirect Expert Systems with Applications* 37 4009–4019.
- [8] Wang X.Li. [1997], *A Course In Fuzzy Systems And Control*, Prentice Hall PTR.