

# ALGORTIMA PENGHINDARAN TABRAKAN PADA MULTI-ROBOT BERBASIS POSISI DAN PEMBAGIAN ORIENTASI

## *COLLISION AVOIDANCE ALGORITHM ON MULTI-ROBOT BASED POSISITION AND ORIENTATION SHARING*

Joao Amaral de Fatima Pereira<sup>1</sup>, Agung Nugroho Jati, S.T., M.T.<sup>2</sup>, Randy Erfa Saputra, S.T., M.T.<sup>3</sup>

Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[carlosdomingas21@gmail.com](mailto:carlosdomingas21@gmail.com), <sup>2</sup>[agungnj@telkomuniversity.ac.id](mailto:agungnj@telkomuniversity.ac.id), <sup>3</sup>[resaputra@telkomuniversity.ac.id](mailto:resaputra@telkomuniversity.ac.id)

### Abstrak

Algoritma penghindaran tabrakan pada multi-robot merupakan hal yang paling penting pada multi-robot untuk penyelesaian suatu tugas yang telah ditanamkan ke dalam robot masing-masing, sesuai alokasi tugas dan pembagian orientasi antara setiap multi-robot. Dengan adanya penghindaran robot dapat berjalan menuju titik tujuan tanpa adanya tabrakan. Ada beberapa algoritma yang sering digunakan untuk menjalankan penghindaran tabrakan. Maze adalah salah satu masalah yang harus dipecahkan oleh robot ketika berada di dalamnya, untuk menyelesaikan maze dapat menggunakan banyak algoritma, seperti flood fill algorithm, Mouse Finder dan wall follower.

Penghindaran tabrakan juga adalah salah satu masalah mendasar dalam robotika. Masalah umumnya dapat didefinisikan dalam konteks single robot otonom yang bernavigasi di lingkungan dengan hambatan dan / atau entitas seluler lainnya, di mana robot menggunakan penginderaan berkelanjutan dan siklus tindakan. memecahkan labirin dengan lebih dari 2 robot, masalah serius yang perlu dilihat, karena menghindari tabrakan mungkin terjadi selama robot berjalan di dalamnya. robot memiliki tujuan untuk mencapai titik akhir, maka kemungkinan tabrakan akan terjadi jika robot menghadapi dinding labirin atau dengan robot lain. Dalam masalah ini penulis mengetahui bahwa surutnya suatu algoritma untuk setiap robot adalah kunci untuk menghindari masalah tersebut.

Pada penelitian ini dibuat agar dapat menyelesaikan masalah tabrakan pada robot ketika berada di maze dengan menggunakan algoritma Right Wall Follower agar robot dapat menuju titik tujuan dengan tanpa adanya tabrakan.

**Kata kunci:** Penghindaran tabrakan, *Multi-Robot*, *Right Wall Follower*

### Abstract

Multi-robot collision avoidance algorithm is the most important thing in multi-robot for completion of a task that has been implanted into each robot, according to task allocation and orientation division between each multi-robot. With the avoidance of the robot can walk toward the point without collision. There are several algorithms that are often used to run collision avoidance. Maze is one of the problems that robots must solve when in it, to complete the maze can use many algorithms, such as flood fill algorithm, Mouse Finder and wall follower.

Collision avoidance is a fundamental problem in robotics. Problems can generally be defined in the context of autonomous mobile robots that navigate in environments with barriers and/or other mobile entities, where robots use continuous sensing and action cycles. solving a maze with more than 2 robots, such a serious problem that we need to take a look, because collision avoidance may occur during the robots are running inside it. robots have a goal it is to reach the endpoint, then probably collision will happen if the robots face the wall of a maze or with another robot. In this problem we find out that ebbed an algorithm for each robot is the key to avoiding those problems.

In this research is made in order to solve the problem of collision on the robot when in the maze by using the Right Wall Follower algorithm so that the robot can reach the destination point with no collision.

**Keywords:** *Multi-Robot*, *Right Wall Follower*, *Collision Avoidance*

### 1. Pendahuluan

Tujuan penelitian robot saat ini adalah untuk melakukan berbagai macam tugas fisik baik robot yang dikontrol manusia ataupun robot yang diprogram untuk melakukan berbagai tugas. Penelitian yang banyak dilakukan dalam bidang robot yaitu pembuatan *single robot*. Penelitian dalam bidang robot ini sudah semakin berkembang karena adanya penelitian pada robot otonom yang dapat bergerak sendiri karena memiliki kecerdasan buatan untuk dapat menghindari tabrakan, berkomunikasi dan berkoordinasi dalam melakukan pengambilan keputusan.

Kemampuan robot juga memiliki keterbatasan seperti halnya manusia yang dapat melakukan suatu hal dengan lebih cepat dan mudah apabila melakukannya bersama-sama layaknya kehidupan semut yang

menghabiskan waktunya untuk berkoloni dalam kegiatan apapun. Dengan kemungkinan kedepannya menggunakan ultrasonic akan lebih baik dan efisien untuk mengaplikasikan strategi tersebut untuk menghindari tabrakan antara satu robot dengan robot yang lain. Maka dengan itu sebelum penulis membahas tentang algoritma/strategi tersebut, harus memahami terlebih dahulu bagaimana komunikasi antara manusia untuk tidak menabrak satu sama lain.

Penghindaran tabrakan merupakan masalah mendasar dalam robotika. Masalah umumnya dapat didefinisikan dalam konteks robot mobile otonom yang menavigasi di lingkungan dengan hambatan dan / atau entitas bergerak lainnya, dimana robot menggunakan siklus penginderaan dan aksi yang berkesinambungan. Di setiap Siklus, sebuah tindakan untuk robot harus dihitung berdasarkan pengamatan lokal terhadap lingkungan, seperti robot tetap bebas dari tabrakan dengan rintangan dan lainnya bergerak, sambil membuat kemajuan menuju sebuah goal. Penghindaran tabrakan dan masalah perencanaan jalan, suatu mayoritas, hanya berurusan dengan hambatan statis. Menghindari hambatan yang bergerak cukup menjadi tugas, dengan petugas perhitungan matematis yang harus dilakukan di jalan/arena. Komplikasi berkembang biak saat bergerak hambatannya adalah robot dengan kecerdasan mereka sendiri, yang mana hanya menjamin bahwa mereka tidak akan menempel pada jalur tetap apapun. Namun, permasalahan yang muncul pada sistem *Multi-Robot* yang didasarkan pada kemampuan khusus masing-masing robot adalah adanya kemungkinan terjadi tabrakan antara robot maupun halangan yang akan ditempuh oleh robot. Membuat sistem untuk penghindaran tabrakan sangatlah penting, agar permasalahan yang berkaitan dengan tabrakan tidak akan terjadi selama masing-masing robot bekerja untuk mencari goal tertentu.

## 2. Dasar Teori

### 2.1 Algoritma Penyelesaian Maze

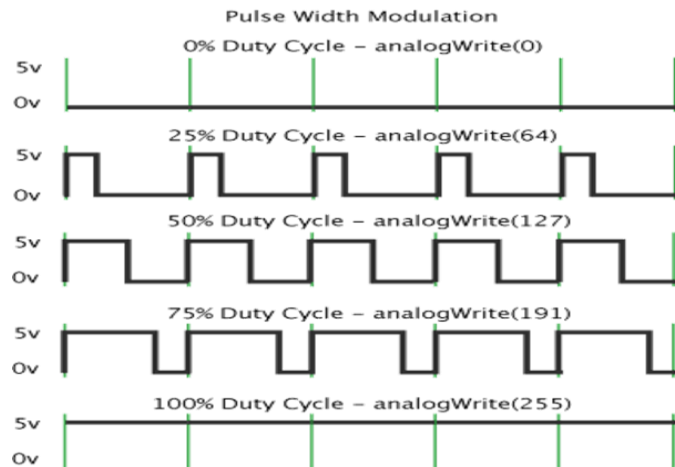
Ada banyak algoritma yang bisa digunakan untuk menyelesaikan labirin seperti random mouse, wall follower, dan algoritma flood fill [2] [13]. Algoritma wall follower umumnya digunakan ketika posisi titik target tidak diketahui. Itu target biasanya diidentifikasi dengan tanda yang unik. Di sisi lain, algoritma pengisian banjir biasanya digunakan saat posisi titik target secara geometris diketahui tetapi robot perlu menemukan rute terpendek [2].

### 2.2 Algoritma Flood Fill

Algoritma Flood Fill adalah salah satu yang paling efisien algoritma yang akan digunakan [1] [2]. Algoritma ini dapat memecahkan labirin yang rumit dan sulit tanpa memiliki masalah / tabrakan dengan menetapkan nilai semua sel dalam labirin, yang nilai-nilai itu menunjukkan langkah-langkah dari setiap sel ke mencapai titik tujuan. Pada saat yang sama sebagian besar ini Algoritma digunakan dengan menghitung setiap grid yang diteruskan dan yang berikutnya grid, berarti memecahkan labirin dengan mencatat nomor dari grid.

### 2.3 Pulse Width Modulation (PWM)

Modulasi Lebar Pulsa "PWM" adalah metode sederhana oleh mengendalikan nilai analog melalui sinyal digital dengan mengubah atau memodulasi lebar pulsa. Ini adalah sumbu algoritma yang berlaku menggunakan listrik [1]. Dengan menyetel suatu perangkat analog dengan gelombang pulsa berubah antara HIGH dengan "5V" dan LOW dengan "0V" dengan kecepatan yang cepat berperilaku seperti mendapatkan tegangan stabil di suatu tempat antara 0V dan 5V. Gambar-1 menunjukkan contoh PWM dengan tugas siklus 0% - 100 %, yang memberikan nilai tegangan untuk 0V sampai 5V. Jangka waktu siklus menjelaskan jumlah waktu dalam periode ketika pulsa menyala atau HIGH dan mati atau LOW.  $t$  biasanya ditentukan sebagai persentase dari periode penuh. 0% siklus tugas berarti gelombang PWM mati atau LOW dan 100% berarti gelombang sepenuhnya aktif atau HIGH [1].



Gambar 1 *duty cycle* PWM dari 0% - 100%

### 2.4 Algoritma Wall Follower

Algoritma berikut dinding adalah salah satu yang paling sederhana teknik yang dapat digunakan untuk memecahkan labirin. Pada dasarnya, iturobot akan mengambil arahnya dengan mengikuti kiri atau kanan dinding [1] [12] [4]. Setiap kali robot mencapai persimpangan, itu akan terjadi akal untuk dinding pembukaan dan pilih arah memberikan prioritas ke dinding yang dipilih. Dengan mengambil dinding sebagai panduan, strategi ini mampu membuat robot mencapai finish titik labirin tanpa benar-benar menyelesaikannya. Itu instruksi yang terlibat dalam dinding berikut untuk kiri dan kanan hak diberikan dalam tabel-1 di bawah ini. Namun, algoritma ini tidak lagi dianggap sebagai metode yang efisien untuk memecahkan labirin. Ini karena algoritma wall follower akan gagal menyelesaikan beberapa konstruksi labirin, seperti labirin dengan tertutup daerah lingkaran [1].

LEFT WALL FOLLOWER ROUTES	RIGHT WALL FOLLOWERS ROUTES
If NO wall at LEFT TURN LEFT	If NO wall at RIGH TURN RIGHT
Else if NO wall in front FORWARD	Else if NO wall in front FORWARD
Else if NO wall at RIGHT TURN RIGHT	Else if NO wall at LEFT TURN LEFT
Else TURN AROUND	Else TURN AROUND

Gambar *Right dan Left Wall Follower Routine*

### 2.5 Algoritma *Righ Wall Follower*(RWF)

Dari semua karya terkait itu, kami mengambil Dinding Kanan Algoritma Follower sebagai bagian utama, tetapi di dalam Kanan Algoritma Wall Follower itu sendiri, ada keberadaan PWM yang akan benar-benar mengendalikan roda untuk menjelajahi dan menunjukkan bagaimana logika fuzzy bekerja [3] [6] [9].

Algoritma untuk *Right Wall Follower* dapat dimengerti dengan melihat di *pseuocode* berikut:

```

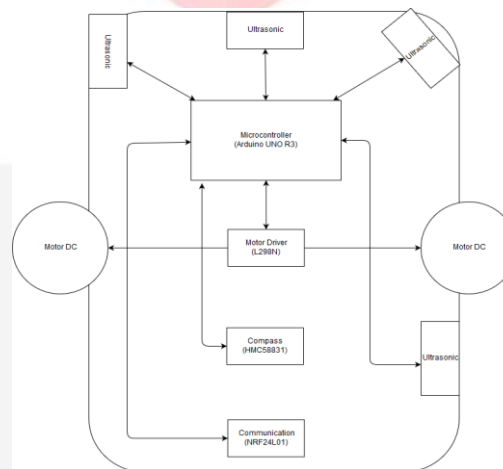
While not Reach Goal
  If frontside is open
    If Right1 > 10
      Turn_Right Slowly
    If Right2 > 10
      Turn_Left Slowly
    Else if right1 is open then
      Turn_right
    Else if left side is open then
      Turn_left
  Else
    Turn_around
Loop

```

### 3. Pembahasan

#### 3.1 Gambaran Umum Sistem

Sebelum kita jelaskan lebih jauh tentang algoritma, pertama-tama kita perlu tahu di mana sensor tersebut ditempatkan dalam robot, maka akan lebih mudah untuk memahami lebih baik algoritma.

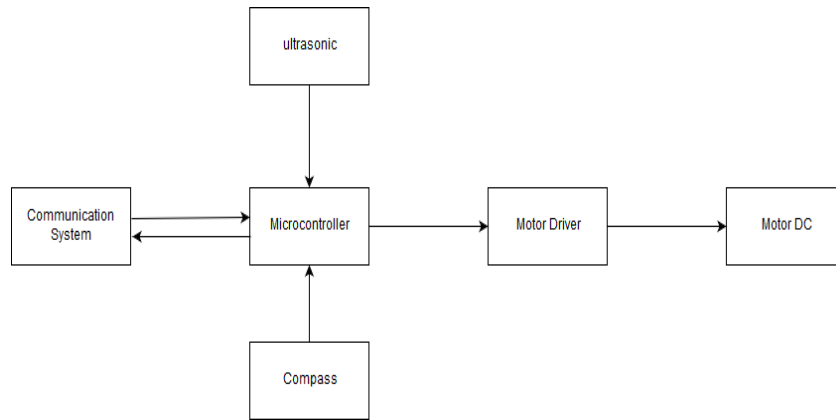


Gambar 3 Gambar design Robot

Pada gambar 3 menunjukan desain pada robot untuk peletakan sensor di bagian robot.

#### 3.2 Perancangan Sistem

Secara garis besar, perancangan Multi-Robot memerlukan beberapa hardware yang diantaranya berupa Mikrokontroler, Sensor ultrasonic, Modul Komunikasi, Motor DC dan Driver Motor. Motor DC diperlukan sebagai motor listrik yang akan digerakkan dengan tegangan arus yang searah dengan kumparan medan agar diubah menjadi energi gerak mekanik. Untuk menjalankan Motor DC, diperlukan Driver motor yang fungsinya mengatur arah dan kecepatan pada motor DC. Sensor ultrasonic diperlukan sebagai penunjang kinerja robot karena perancangan robot yang akan dibuat adalah sebuah sistem Multi-Robot yang membutuhkan fungsi dari sensor dan modul tersebut. Sensor ultrasonic digunakan untuk mendeteksi objek yang berada disekitarnya dan juga untuk menjaga jarak antar robot dan robot dengan objek disebelahnya. Data dari setiap sensor tersebut akan diperoleh dan diproses pada mikrokontroler. Modul komunikasi digunakan agar setiap robot dapat saling bertukar informasi dengan robot lainnya. Sehingga pada saat memiliki informasi yang diperlukan, robot dapat mengambil keputusan dengan mengirimkan sinyal ke robot lainnya untuk berjalan dengan formasi seperti apa dan bergerak maju atau berbelok melintasi objek. Hasil dari komunikasi ini akan diproses di mikrokontroler yang kemudian akan dikirimkan ke tiap-tiap robot melalui modul komunikasi yang sama untuk dikirimkan ke driver motor setiap robot agar dapat menentukan kearah mana robot akan bergerak mengikuti objek.

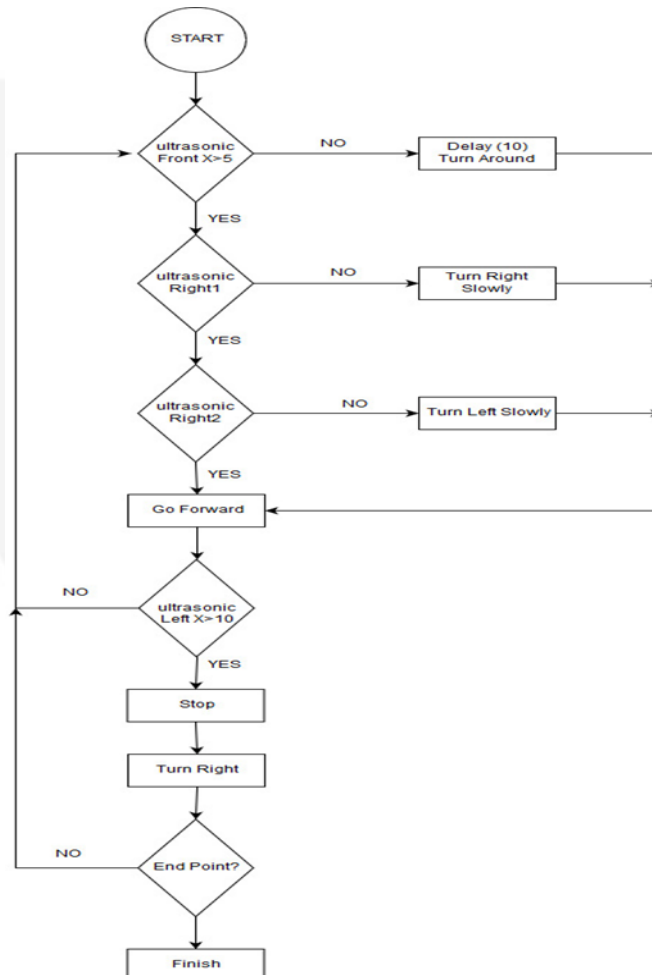


Gambar 4 Diagram Blok Sistem Multi Robot

### 3.3 Perancangan Algoritma Penghindaran Tabrakan

Untuk Algoritma RWF akan bergabung dengan PWM mengendalikan. Tetapi algoritme tersebut digabungkan menggunakan Logika fuzzy. Gambar 4 menunjukkan bahwa RWF berfungsi dengan PWM dan dikendalikan oleh input data yang ditentukan dengan logika fuzzy [6].

Diagram alur di bawah akan menunjukkan, bagaimana kami menerapkan Algoritma Right Wall Follower pada Multi-Robot. Dan ini flowchart juga menunjukkan nilai tertentu dari ultrasonik untuk membuatnya setiap keputusan untuk robot seperti mengikuti dinding, belok kiri, putar kanan atau berbalik. Tujuan dari flowchart ini adalah untuk menemukan titik akhir dengan mengimplementasikan algoritma Right Wall Follower, dan pada saat yang sama tidak ada tabrakan untuk robot dan labirin, juga dengan robot lain.



Gambar 5 Flow-chart Algoritma *Right Wall Follower* untuk Multi robot

Sebelum algoritma di atas berfungsi, dibutuhkan beberapa keputusan membuat untuk menjalankannya, seperti apa yang akan dilakukan sensor jika data mereka miliki dari input, dan bagaimana cara kerjanya. Logika fuzzy bekerja kali ini untuk membiarkan robot membuat keputusan sendiri untuk menjalankannya merancang algoritma [7] [12]. Seperti yang diperlihatkan diagram alur itu, hanya ada 3 ultrasonik yang digunakan untuk menunjukkan bagaimana RWF kerja. Tabel di bawah ini menunjukkan bahwa masukan dari setiap tiga sensor akan memberitahu motor dengan mengendalikan menggunakan PWM. Sebenarnya, menggunakan PWM untuk mengendalikan motor sebaik itu implementasi, karena PWM akan bekerja dengan baik jika suatu mikrokontroler menerima data dari lebih dari 2 sensor.

### 3.4 Perancangan Algoritma Penghindaran Tabrakan

Pada bagian ini sangatlah penting yaitu, akan membahas tentang algoritma yang akan digunakan untuk melakukan penyelesaian labirin pada Multi-Robot tanpa adanya tabrakan antara robot dan labirin.

Alasan menggunakan algoritma *Right Wall Follower* adalah, dikarenakan robot akan di jalankan di dalam maze, dan memiliki tujuan yang harus menyelesaikan maze tersebut. Maka dengan itu, dengan menggunakan algoritma tersebut adalah salah satu pilihan yang baik. Sehingga robot hanya mengikuti dinding labirin dan saling komunikasi antara satu sama lain.

Dengan menggunakan algoritma *Right Wall Follower*, yang berarti robot akan berjalan mengikuti dinding bagian kanan, sehingga dengan cara mengikuti, robot dapat menyelesaikan sebuah maze, dengan waktu yang cukup cepat.

Tabel 3.1 berikut ini akan menjelaskan tentang pergerakan motor atau *PWM*, ketika motor mendapatkan inputan dari nilai ultrasonic:

Tabel 1 pwm algoritma RWF

Sensor depan	Sensor Kanan	Motor kiri	Motor kanan	Arah
$X > 10$	$X > 8 \ \& \ X < 10$	Pwm= 90	Pwm= 60	Maju
$X > 10$	$X \geq 10$	Pwm= 90	Pwm= 60	Maju
$X > 10$	$X \leq 8$	Pwm= 60	Pwm= 90	Maju

Tabel 2 berikut akan memperlihatkan kapan robot akan belok kiri dan kanan sesuai arah hadap motor DC dan PWM yang diprogramkan.

Tabel 2 putaran motor pada arah hadap robot

Arah Hadap Robot	Arah motor Kiri	Arah motor Kanan
Belok Kiri	Mundur	Maju
Belok Kanan	Maju	Mundur
Putar Balik	Maju	Mundur

Setelah mengetahui arah hadap robot, maka dari situ kita dapat melakukan keputusan sesuai nilai inputan dari ultrasonic, dan output adalah gerakan motor DC.

Tabel 3 dibawah ini akan menunjukkan putaran motor dan arah motor terhadap arah hadap robot

Tabel 3. 3 PWM untuk arah hadap robot

Arah Robot	Motor Kiri		Motor Kanan	
	Maju	Mundur	Maju	Mundur
Belok Kiri	PWM=0	PWM=90	PWM=90	PWM=0
Belok Kanan	PWM=90	PWM=0	PWM=0	PWM=90
Putar Balik	PWM=90	PWM=0	PWM=0	PWM=90
Berhenti	PWM=0	PWM=0	PWM=0	PWM=0
Mundur	PWM=0	PWM=90	PWM=0	PWM=90

berikut akan menunjukan pengambilan keputusan dan arah robot, untuk PWM pada motor dapat di lihat di tabel 3.3, yang besrta putaran motor DC.

Tabel 3. 4 arah hadap robot berdasarkan nilai input ultrasonic

Sensor Depan	Sensor Kanan	Sensor Kiri	Arah
$X < 10$	$X > 20$	$X < 20$	Belok kanan
$X < 10$	$X < 20$	$X > 20$	Belok kiri
$X < 10$	$X < 10$	$X < 10$	Putar balik

#### 4. Implementasi dan Pengujian Sistem

##### 4.1 Hasil Pengujian pada algoritma Wall follower

Tabel 4.1 berikut memperlihatkan hasil pengujian algoritma RWF robot:

*Tabel 3 Pengujian Algoritma RWF*

Pengujian ke -	Robot 1	Robot 2	Mengenal Junction	Keterangan
	Berjalan Mengikuti Dinding Kanan	Berjalan Mengikuti Dinding Kanan		
1	Ya	Ya	Ya	Akurat
2	Ya	Ya	Ya	Akurat
3	Ya	Ya	Ya	Akurat
4	Ya	Ya	Ya	Akurat
5	Ya	Ya	Ya	Akurat
6	Ya	Ya	Ya	Akurat
7	Ya	Ya	Ya	Akurat
8	Ya	Ya	Ya	Akurat
9	Ya	Tidak	Ya	Tidak Akurat
10	Ya	Ya	Ya	Akurat

Dari hasil di atas menyimpulkan bahwa robot berjalan dengan mengikuti dinding dengan akurat, sehingga tidak ada tabrakan yang terjadi antara robot dan dinding

##### 4.2 Hasil Pengujian robot di junction dan mengambil keputusan

Tabel 4.2 berikut memperlihatkan hasil pengujian eksekusi robot sesuai dengan data yang diterima:



Tabel 3 Pengujian putaran arah hadap robot

Kondisi	Robot 1		Robot 2			Keterangan
	F<10 & L>20 & R < 20	F<10 & L>20 & R < 20	F<10 & L>20 & R < 20	F<10 & L>20 & R < 20	F<10 & L<20 & R < 20	
Pengujian ke -	Belok Kiri	Belok Kanan	Belok Kiri	Belok kanan	Bergerak Sesuai Data Yang Diterima	
1	Ya	Ya	Ya	Ya	Ya	Akurat
2	Ya	Ya	Ya	Ya	Ya	Akurat
3	Ya	Ya	Ya	Ya	Ya	Akurat
4	Ya	Ya	Ya	Ya	Ya	Akurat
5	Ya	Ya	Ya	Ya	Tidak	Tidak Akurat
6	Ya	Ya	Ya	Ya	Ya	Akurat
7	Ya	Ya	Ya	Ya	Ya	Akurat
8	Ya	Ya	Ya	Ya	Ya	Akurat
9	Ya	Ya	Ya	Ya	Ya	Akurat
10	Ya	Ya	Ya	Ya	Ya	Akurat

### Hasil Pengujian 2 robot dalam arena

Tabel 1 Hasil Pengiriman 2 robot dalam arena

Pengujian ke -	Robot 1	Robot 2	2 Robot Berhadapan		Keterangan
	Di Junction Kirim Data	Di Junction Kirim Data	Putar balik	Delay Putar Balik	
1	Ya	Ya	Ya	0.001s	Akurat
2	Tidak	Tidak	Ya	0.021s	Tidak
3	Ya	Ya	Ya	0.022s	Akurat
4	Ya	Ya	Ya	0.001s	Akurat
5	Tidak	Tidak	Ya	0.028s	Tidak Akurat
6	Ya	Ya	Tidak	-	Akurat
7	Ya	Ya	Ya	0.001s	Akurat
8	Ya	Ya	Ya	0.001s	Akurat



9	Ya	Ya	Tidak	-	Akurat
10	Ya	Ya	Ya	0.001s	Akurat

Gambar dibawah ini menunjukkan bagaimana robot berjalan di dalam sebuah labirin:



Gambar 5 arena robot dan robot

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan, maka dapat disimpulkan sebagai berikut :

1. Robot dapat menghindari tabrakan antara robot satu sama lain, maupun robot dengan maze
2. Robot berjalan dengan menggunakan algoritma Right Wall Follower untuk mencari titik akhir *endpoint*
3. Robot mengenal junction dengan nilai inputan ultrasonic depa, kiri dan kanan, sehingga robot dapat mengambil keputusan yang benar.
4. Dan ketika 1 robot mendapatkan titik akhir atau jalan keluar maze, maka Robot tersebut akan mengirim data ke Robot lain

### 5.2 Saran

Saran yang dapat diberikan penulis untuk pengembangan selanjutnya:

1. Dalam studi kasus *multi-robot*, seharusnya robot-robot dapat saling mengirimkan informasi dan menerima informasi secara simultan dan bergerak bersamaan sesuai dengan fungsinya.
2. Dalam melakukan perjalanan melintasi lingkungannya, robot tetap berjalan sesuai apa yang di tanamkan, tetap untuk melihat kondisi robot ketika di dalam maze.
3. Tabrakan antara robot dan maze kadang terjadi dikarenakan nilai inputan ultrasonic yang tidak terlalu akurat.

## Daftar Isi

- [1] Angelo Martinez, Eddie Tunstall, and Mo Jamshidi " Fuzzy Logic Based Collision Avoidance For a Mobile Robot". CAD Laboratory for Intelligent and Robotic Systems, 1994.
- [2] Reinhard Brauningl, Pedro Sanz, Jose Manuel Ezkerra "Fuzzy Logic Wall Following of a Mobile Robot Based on the Concept of General Perception". ICAR '95, 7th International Conference on Advanced Robotics, Sant Feliu de Guixols, Spain, PP.367-376, SEPT., 1995.
- [3] I. Gavrilut, V. Tiponut, A. Gacsadi, L. Tepelea" Wall-following Method for an Autonomous Mobile Robot using Two IR Sensors "University of Timisoara, B-dul Vasile Parvan No. 2, 300223 Timisoara, Romania.
- [4] Michael G. Murphy, Ph.D. "Fuzzy Logic Path Planning system for collision Avoidance by an Autonomous Rover vehicle", NASA/ASEE Summer Faculty Fellowship Program—1991
- [5] C.G. Rusu, I.T. Birou "Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors ",10th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 27-29, 2010

- [6] David Llorca, Vicente Milanés, Ignacio Parra, Miguel Gavilán, Iván García Daza, Joshua Pérez Rastelli, M.A. Sotelo, "Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller", *IEEE Transactions on Intelligent Transportation Systems*, VOL. 12, No. 2, June 2011.
- [7] Akib Islam, Farogh Ahmad, P.Sathya, "Shortest Distance Maze Solving Robot", *International Journal of Research in Engineering and Technology*, eISSN: 2319-1163 | pISSN: 2321-7308.
- [8] Marco A. C. Simões, Helder Guimarães Aragão, Victor Souza, Simon Viegas, "Using Fuzzy Logic to Build a Heterogeneous Multiagent System for the Robotics Soccer Problem" *Computer Architecture and Operating Systems Group(ACSO) State University of Bahia (UNEB)*
- [9] Yee Mon Nyein, Nu Nu Win, "Path Finding and Turning with Maze Solving Robot", *International Journal of Science, Engineering and Technology Research (IJSETR) Volume 5, Issue 9, September 2016*, ISSN: 2278 – 7798.
- [10] Yuan Zhou, Hesuan Hu, Yang Liu, and Zuohua Ding, "Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach" *IEEE Transaction on Systems, man and Cybernetics Systems*. 2017..
- [11] Dali Sun, Alexander Kleiner, Bernhard Nebel, "Behavior-based Multi-Robot Collision Avoidance", 2014 IEEE International Conference on Robotics & Automation (ICRA), Hong Kong Convention and Exhibition Center May 31 - June 7, 2014. Hong Kong, China
- [12] E. S. Ningrum, R. Susetyo and A. H. Alasiry, "Rancang Bangun Mobile Robot untuk Formasi Follow The Leader dengan Metode Dead Reckoning Based Trajectory Following," Institut Teknologi Sepuluh November, Surabaya
- [13] F. R. Q. Aini, "Perancangan dan Implementasi Localization Menggunakan Algoritma Monte Carlo Localization Pada Multi-Robot Beroda," Universitas Telkom, Bandung, 2016
- [14] S. Carpin and L. E. Parker, "Cooperative Leader Following Distributed Multi-Robot System," in *International Conference on Robotics & Automation*, Washington DC, 2002
- [15] Dali Sun, Alexander Kleiner, Bernhard Nebel, "Behavior-based Multi-Robot Collision Avoidance", 2014 IEEE International Conference on Robotics & Automation (ICRA), Hong Kong Convention and Exhibition Center May 31 - June 7, 2014. Hong Kong, China
- [16] L. A. Arta, Suwandi and A. Qurthobi, "Rancang Bangun Kontrol Fuzzy Logic Pada Metode Leader-Follower Untuk Penjejak Formasi Mobile Robot," Universitas Telkom, Bandung
- [17] C.G. Rusu, I.T. Birou "Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors", 10th International Conference on Development and Application system, Suceava, Romania, May 27-29, 2010.
- [18] B. University, "Fuzzy Logic," *Stanford Encyclopedia of Philosophy*, 2006.
- [19] R. Johnson, "Fuzzy logic and Fuzzy logic Sun Tracking Control," *Calvin College*, 2002.
- [20] S. Kusumadewi, Aplikasi Logika Fuzzy untuk Pendukung Keputusan, Yogyakarta: Graha Ilmu, 2004.
- [21] L.-X. Wang, "A Course in Fuzzy Systems and Control," United States of America, 1997.
- [22] L.-X. Wang, A Course in Fuzzy Systems and Control, United States of America: Prentice-Hall International, 1997.
- [23] S. Kusumadewi and H. Purnomo, Aplikasi Logika Fuzzy untuk Pendukung Keputusan Edisi 2, Yogyakarta: Graha Ilmu, 2013.
- [24] S. Kusumadewi, Analisa dan Desain Sistem Fuzzy Menggunakan Toolbox MATLAB, Yogyakarta: Graha Ilmu, 2002.
- [25] Roland Illah R. ,Siegwart Nourbakhsh Autonomus Mobile Robot. Alessandro Farinelli, Luca Iocchi, Daniele Nardi, Multi-Robot Systems: A classification focused on coordination, IEEE , pp. 2015-2028, 2004
- [26] M.Brambilla, E.Ferrante, M. Birattari, and M. Dorigo, A review from the swarm engineering perspective.
- [27] Gerardo Beni, From Swarm Intelligence to Swarm Robotics, 3342:1-9 · July 2004.