

RANCANG BANGUN *MOBILE* ROBOT *MICROMOUSE* UNTUK PENCARIAN RUTE TERPENDEK MENGUNAKAN ALGORITMA *FLOOD FILL*

Rilci Laksmana Putra, Anton Hidayat, MT², Ir. Werman Kasoep, M. Kom³

¹Mahasiswa Sistem Komputer Fakultas Teknologi Informasi Universitas Andalas

²Dosen Teknik Elektro Politeknik Negeri Padang

³Dosen Sistem Komputer Fakultas Teknologi Informasi Universitas Andalas

riellaks@gmail.com, antonramiati@gmail.com, wermankaosep@yahoo.com

ABSTRAK

Mobile robot merupakan salah satu jenis robot yang banyak digunakan dalam membantu pekerjaan manusia. *Mobile* robot *micromouse* adalah salah satu jenis robot yang dibuat agar dapat mencari rute terpendek dalam sebuah labirin atau *maze*. Algoritma *flood fill* merupakan salah satu cara untuk mencari rute terpendek dengan melakukan *maze mapping*, dimana robot akan membandingkan nilai dari setiap *cell* dan bergerak menuju *cell* terkecil. Dalam penelitian yang dilakukan, *mobile* robot *micromouse* dirancang menggunakan Sensor *Ultrasonic* PING untuk mendeteksi halangan dan *Arduino Uno* yang akan mengontrol pergerakan dari motor DC. Robot akan mencari rute terpendek berdasarkan *maze mapping* yang telah dilaluinya. Robot diujikan pada 3 buah labirin atau *maze*, dimana setiap labirin atau *maze* dilakukan 10 kali pengujian dan diperoleh keberhasilan sebesar 80% pada *maze* 1, pada *maze* 2 sebesar 70% dan pada *maze* 3 sebesar 70%.

Kata kunci: *Robot, Micromouse, Mapping, Maze, Arduino Uno, Flood Fill, Ultrasonic, PING*

ABSTRACT

Mobile Robot is one robot that much used to help out man. *Micromouse mobile robot* is one robot that made to can find the route shortest in a maze. *Flood fill algorithm* is one way to find the route shortest by doing *maze mapping*, where the robot will compare the value of each cell and moving toward the smallest cell. In the research, the *micromouse mobile robot* was designed using *Ultrasonic PING* sensor to detect obstacles and *Arduino Uno* that will control the movement of DC motor. The robot will search a route shortest based on *maze mapping* that it has passed. The robot was tested on three mazes, where each maze was tested 10 times and obtained success level on the first maze of 80%, the second maze of 70% and the third maze of 70%.

Keywords: *Robot, Micromouse, Mapping, Maze, Arduino Uno, Flood Fill, Ultrasonic, PING*

I. PENDAHULUAN

Teknologi robotik saat ini mengalami kemajuan yang sangat pesat. Teknologi robotik digunakan untuk menggantikan berbagai pekerjaan manusia, seperti pekerjaan yang membutuhkan ketelitian tinggi dan memiliki resiko tinggi terhadap keselamatan manusia, ataupun pekerjaan yang dilakukan berulang – ulang.

Mobile robot merupakan salah satu jenis robot yang banyak digunakan dalam membantu pekerjaan manusia. *Mobile* robot dapat diterapkan dalam beberapa aplikasi diantaranya: pemadam kebakaran, pengelompokan barang, pengikut objek, robot sepak bola dan robot pencari rute terpendek.

Robot pencari rute terpendek merupakan sebuah robot yang dapat digunakan untuk mencari jalan keluar dalam sebuah labirin atau *maze* yang disebut robot *micromouse*. Dimana labirin atau *maze* merupakan suatu daerah yang memiliki jalan yang berliku-liku. Labirin itu sendiri, banyak kita jumpai

ditempat-tempat yang memiliki gang-gang sempit dipermungkikan padat, dimana banyak terdapat jalan buntu. Oleh karena itu, diperlukan mencari sebuah rute terpendek untuk memilih rute yang bisa digunakan. Prinsip ini biasa diaplikasikan pada sistem *GPS*. Robot pencarian ini sering disebut dengan robot *micromouse*, dimana robot ini dapat bergerak dengan bebas di dalam sebuah area labirin (*maze*) tanpa menyentuh obyek di sekitarnya.

Penelitian sebelumnya telah dibahas mengenai *mobile* robot *micromouse* ini dengan menggunakan algoritma *backtracking*. Dimana, robot *micromouse* ini dibuat dengan menggunakan *ATMega 8535* sebagai kontroler, menggunakan sensor *sharp* sebagai indera penglihat dan menggunakan algoritma *backtracking* sebagai metode untuk memecahkan masalah^[1].

Mobile robot yang akan dirancang pada penelitian ini berguna untuk pencarian rute terpendek, menggunakan mikrokontroler *ATMega 328* sebagai kontroler dan menggunakan sensor *ultrasonic* sebagai indera penglihat. Berdasarkan latar belakang tersebut, maka penulis ingin

membuat sebuah rancangan *mobile* robot dengan judul “**Rancang Bangun Mobile Robot Micromouse untuk Pencarian Rute Terpendek Menggunakan Algoritma Flood Fill.**” Robot ini dapat dimanfaatkan untuk membaca sebuah lokasi atau denah yang berguna untuk menuju lokasi tujuan yang diinginkan melalui rute terpendek

II. LANDASAN TEORI

A. Robot

Definisi Robot Robot berasal dari kata “*robota*” yang dalam bahasa Ceko yang berarti budak, pekerja atau kuli. Pertama kali kata “*robota*” diperkenalkan oleh Karel Capek dalam sebuah pentas sandiwara pada tahun 1921 yang berjudul RUR (*Rossum’s Universal Robot*)^[2]. Robot dapat didefinisikan sebagai sebuah alat mekanik yang dapat diprogram yang dapat bekerja secara otomatis.

B. Robot Line Follower

Mobile robot merupakan sebuah robot yang dapat bergerak dengan leluasa karena memiliki alat gerak untuk berpindah posisi. Secara umum dan mendasar sebuah mobile robot dibedakan oleh locomotion system atau sistem penggerak. Locomotion merupakan gerakan melintasi permukaan datar. Semua ini disesuaikan dengan medan yang akan dilalui dan juga oleh tugas yang diberikan kepada robot. Berikut adalah klasifikasi robot menurut jenis locomotion^[3].

C. Robot Micromouse

Micromouse adalah sebuah robot yang dapat bergerak dengan bebas di dalam sebuah area labirin (*maze*) tanpa menyentuh obyek di sekitarnya. Pada akhirnya robot mengetahui ke arah mana harus bergerak, berapa derajat harus berputar, dan bagaimana sebenarnya jika menemui jalan buntu pada area labirin (*maze*) tersebut^[4].

D. Komponen Robot

Komponen robot terdiri dari^[5]: Sensor, Sistem Kecerdasan (Kontrol), Peralatan mekanik (Aktuator), dan Sumber daya (*Power*).

E. Sensor Ultrasonic

Ultrasonic, sering digunakan untuk keperluan mengukur jarak sebuah benda atau untuk mendeteksi rintangan. Teknik mengukur menggunakan ultrasonic ini meniru cara yang digunakan kelelawar atau lumba-lumba yang secara alami menggunakan sonar (*sound navigation and ranging*) untuk keperluan mengukur jarak dan navigasi^[6].

F. Mikrokontroler

Mikrokontroler adalah piranti elektronik berupa IC (Integrated Circuit) yang memiliki kemampuan memanipulasi data (informasi) berdasarkan suatu urutan instruksi (program) yang dibuat oleh programmer. Mikrokontroler merupakan

contoh suatu sistem komputer sederhana yang masuk dalam kategori embedded komputer. Dalam sebuah struktur mikrokontroler akan ditemukan juga komponen-komponen seperti: processor, memory, clock dan lain-lain^[8].

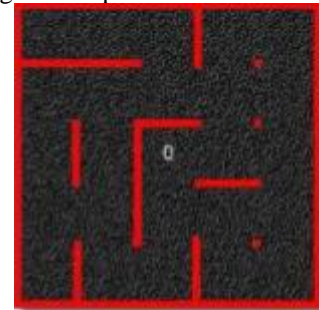
G. Arduino Uno

Arduino Uno adalah *board* berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital input/output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, *jack* listrik tombol *reset*. Arduino Uno merupakan salah satu jenis arduino yang menggunakan USB sebagai antar muka pemrograman atau komunikasi komputer^[11].

H. Algoritma Flood Fill

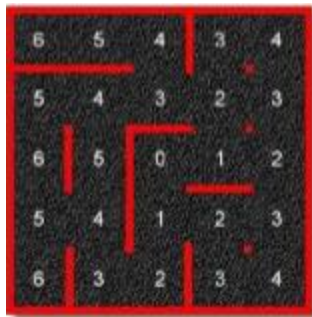
Algoritma *flood fill* dapat dianalogikan menggunakan air yang ditumpahakan pada sebuah *maze*. Berikut ini penjelasannya^[13].

- Proses penumpahan air terpusat hanya pada satu titik (*center*, selanjutnya titik ini akan dikenal sebagai *destination* atau tujuan)
- Air akan membanjiri titik *center* ini, kemudian mulai mengalir ke area di sekitarnya, yang tidak terhalang oleh dinding (dapat diakses secara langsung)
- Secara *virtual*, *maze* dibagi menjadi beberapa kotak kecil (*array*)
- Kotak dimana titik *center* berada, diberi nilai ‘ 0 ’, seperti yang terlihat pada Gambar 1.



Gbr 1. Kondisi Awal *Maze*^[13]

- Kotak yang terisi air setelah *center*, akan diberi nilai 0
- Kotak yang terisi air setelah golongan 1, akan diberi nilai 2
- Kotak yang terisi air setelah golongan 2, akan diberi nilai 3
- Dan begitu pula untuk kotak yang terisi air selanjutnya, seperti terlihat pada Gambar 2.



Gbr 2. Kondisi Maze Setelah Dipenuhi Air^[13]

Arti dari nilai di dalam masing-masing kotak adalah jumlah kotak yang harus ditempuh dari kotak tersebut untuk mencapai *center* (tujuan). Asumsikan kotak yang berada pada bagian bawah sebelah kiri merupakan *start*, kemudian ikutilah kotak yang memiliki nilai lebih kecil dari nilai kotak yang sedang ditempati. Rute yang akan terbentuk adalah rute terpendek yang dapat ditempuh dari *start* menuju ke *center*^[13].

I. Motor Shield

Motor DC tidak dapat dikendalikan langsung oleh mikrokontroler, karena kebutuhan arus yang besar sedangkan keluaran arus dari mikrokontroler sangat kecil, maka digunakan sebuah *driver* motor. Salah satu *driver* motor yang dapat digunakan yaitu motor *shield* L298. Motor *shield* ini menggunakan sebuah chip L298P yang dapat menghasilkan output hingga sebesar 2A pada tiap *channel*. Pengontrolan kecepatan dapat menggunakan PWM yang dapat diperoleh dari keluaran PWM Arduino pada Pin 5 dan 6. Fungsi *enable/disable* dari pengontrolan motor ditunjukkan pada Pin Digital 4 dan 7 Arduino. Motor *shield* ini dapat dialiri arus secara langsung dari Arduino atau dari sumber arus eksternal^[14].

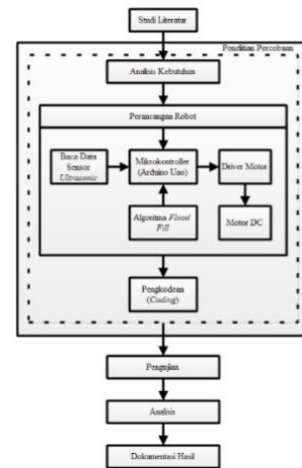
J. Motor DC

Motor DC bekerja bila pada kedua kaki stator diberikan tegangan atau beda potensial sehingga pada rotor terjadi perubahan energi listrik menjadi energi mekanik yang mengakibatkan terjadinya tolak menolak antara rotor dan stator, karena kedua bagian ini dipasang secara berdampingan dengan kutub yang berbeda. Dari tolak menolak yang terjadi pada kedua magnet yang ditimbulkan oleh stator dan rotor maka terjadi suatu pergerakan yang mengakibatkan rotor akan berputar sesuai dengan pemberian tegangan pada kaki stator^[15].

III. METODE PENELITIAN

A. Rancangan Penelitian

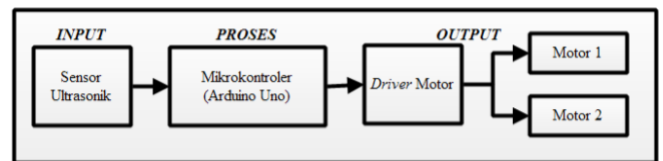
Berikut adalah beberapa tahap penelitian yang dilakukan



Gbr 3. Rancangan Penelitian

B. Blok Diagram

Berikut adalah blok diagram dari sistem yang dibuat.



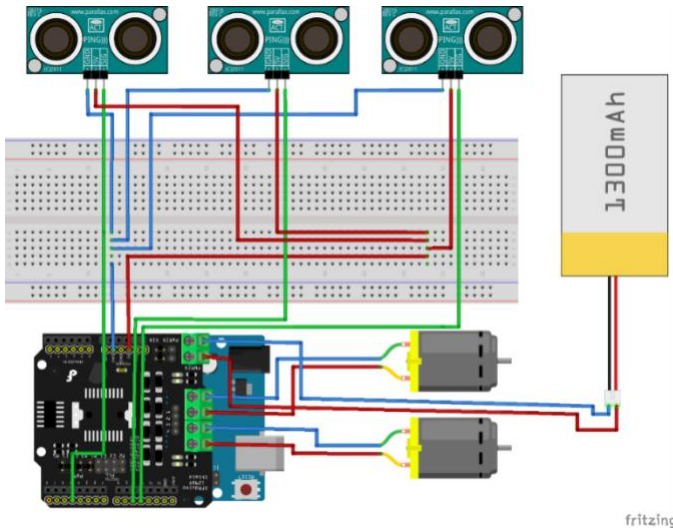
Gbr 4. Blok Diagram Sistem

Sistem yang dibuat ini bekerja dengan adanya sensor *ultrasonic* yang berfungsi untuk membaca halangan yang ada di sekitar robot, jika tidak ada halangan di depan robot maka robot akan bergerak maju sedangkan jika ada halangan maka robot akan melihat sisi kanan atau sisi kiri dari robot untuk dapat bergerak ke arah yang tidak ada halangan. Untuk memproses data yang diperoleh sensor *ultrasonic* digunakan Arduino Uno sebagai *controller* utama.

C. Wiring Diagram Robot

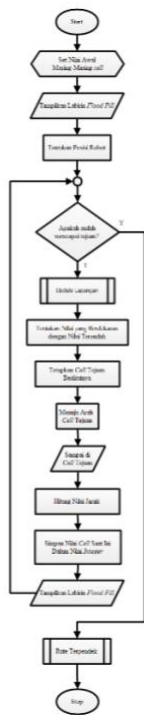
Wiring diagram robot pada Gambar 5. terdiri dari 3 buah sensor *ultrasonic* PING, 2 motor dc dan motor shield L298. Sensor *ultrasonic* ini dihubungkan ke *board* arduino uno, semua pin yang digunakan adalah pin digital (D3, D10 dan D11). Pin D3 digunakan sebagai sensor kiri, pin D10 digunakan sebagai sensor tengah dan pin D11 digunakan sebagai sensor kanan.

Ping yang digunakan untuk menghubungkan antara motor shield L298 dengan board arduino uno adalah ping digital (D4, D5, D6, dan D7). Dimana pin D4 dan D5 (*Enable*) digunakan sebagai *output* motor motor 1 dan ping D6 (*Enable*) dan D7 digunakan sebagai *output* motor 2. Sumber yang digunakan untuk board arduino uno dan motor shield berasal dari baterai 7,4V.



Gbr 5. Skema Rangkaian Keseluruhan

D. Perancangan Software



Gbr 6. Flowchart perancangan software

Penjelasan dari Gambar 6 di atas adalah sebagai berikut, robot berjalan dari posisi start, setiap 20 cm robot akan berhenti, mengecek apakah terdapat persimpangan atau tidak, ada atau tidaknya persimpangan akan dibaca oleh robot sebagai data, data dari lapangan ini akan masuk sebagai data pada virtual array. Data ini selanjutnya akan menentukan nilai-nilai untuk neighbour cell atau dengan kata lain robot melakukan proses update. Nilai-nilai neighbour cell ini kemudian dibandingkan, dan robot akan bergerak ke arah neighbour cell yang memiliki nilai paling kecil. Begitu seterusnya hingga robot menemukan finish atau tujuan.

Selanjutnya robot akan mencari rute terpendek yang harus ditempuh oleh robot untuk berjalan menuju tujuannya. Seperti yang ditunjukkan oleh Gambar 7, dimana robot sudah mengetahui nilai jarak yang diperoleh dan robot menentukan nilai yang berdekatan dengan nilai yang terendah. Setelah itu, robot akan bergerak menuju lokasi cell berikutnya hingga mencapai finish atau tujuan.



Gbr 7. Flowchart perancangan software

E. Perancangan Labirin atau Maze

Labirin dirancang untuk robot untuk memecahkan adalah dari ukuran 3×3 cell seperti yang ditunjukkan pada Gambar 3.6. labirin yang sebenarnya dibangun, seperti yang ditunjukkan pada Gambar 8, memiliki ukuran fisik 60cm x 60cm. Labirin ini dirancang sehingga akan memiliki beberapa jalur yang harus dipecahkan. Salah satu jalan lebih panjang dari yang lain. Robot harus memutuskan mana salah satu jalur yang lebih pendek dan memecahkan labirin melalui jalan itu.



(a)



(b)

Gbr 8. (a) Design Maze dan (b) Bentuk Maze

F. Perancangan Algoritma Flood Fill

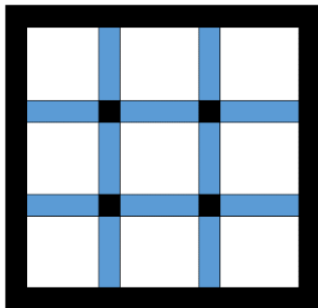
Algoritma *flood fill* digunakan untuk memecahkan labirin karena keseimbangan dalam efisiensi dan kompleksitas. Langkah utama dalam algoritma adalah *Mapping*, *Flooding*, dan *Updating*.

a. Maze Mapping

Untuk robot untuk dapat memecahkan labirin, robot harus mengetahui ukuran labirin dan membagi mereka ke dalam sejumlah *cell* yang dapat digunakan

di kemudian menghitung jalur terpendek ke tujuan. Dalam tugas akhir ini, labirin digunakan 3 x 3 cell. Antara dua sel bisa ada dinding. Dalam deretan 3 cell tersebut terdapat 2 dinding diantara mereka dan 2 dinding disudut labirin dan jumlah dinding dan cell sebanyak 7 buah.

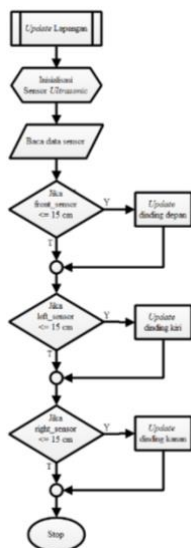
Informasi ini akan disimpan dalam array berukuran 7 x7 seperti yang ditunjukkan pada Gambar 3.8. Cell putih adalah cell yang dapat ditempatkan oleh robot. Dan cell warna biru adalah lokasi untuk dinding. Cell warna hitam merepakan dinding tetap dan persimpangan antara dinding.



Gbr 9. Maze Mapping

b. Update Lapangan

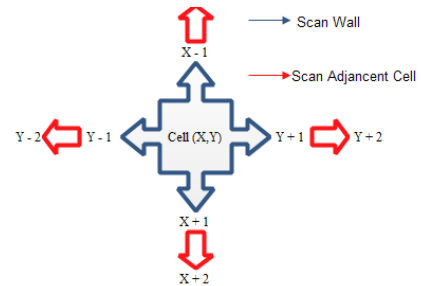
Sebelum robot memutuskan mana yang ingin pindah cell, itu harus memeriksa apakah cell itu dikelilingi oleh dinding, di salah satu dari tiga arah yaitu: kanan, kiri dan depan. Robot membaca jarak rintangan di setiap arah dan memeriksa apakah jarak di masing-masing lebih dari 15 cm. Jika melebihi 15 cm diperbarui sebagai "dinding" di sisi masing-masing. Hal ini digambarkan dengan pada flowchart pada Gambar 10.



Gbr 10. Update Lapangan

Untuk robot memperbarui data dinding dengan benar, maka robot harus mengetahui arah mana yang dihadapinya. Ada empat arah pergerakan yang akan

dihadapi oleh robot yaitu : utara, selatan, timur dan barat, seperti yang ditunjukkan pada Gambar 11 dan robot terus melakukan pembacaan terhadap perubahan yang terjadi.



Gbr 11. Scan Arah Pergerakan Robot dalam Multi Dimensional Array

c. Melakukan Flooding Pada Maze

Setelah robot memperbarui informasi dinding untuk cell saat ini, robot mulai membanjiri matriks untuk menemukan jalan terpendek ke tujuan. Robot akan melakukan pemeriksaan satu cell dalam satu keadaan sampai robot menemukan tujuannya. Algoritma ini akan memberikan nilai setiap cell berdasarkan seberapa jauh cell itu dari cell tujuan. Berdasarkan hal itu, cell tujuan akan diberikan nilai 0. Jika robot beradapa pada cell yang bernilai 3, maka robot akan melakukan 2 langkah (2 cell) untuk untuk mencapai cell tujuan.

Setelah memperbarui informasi dinding untuk sel saat ini, robot mulai membanjiri matriks untuk menemukan jalan terpendek ke tujuan [6]. Melakukan hal yang sama untuk semua sel dan terus mengulangi beberapa kali sampai jalur antara robot dan tujuannya ditemukan. Algoritma memberikan nilai untuk setiap sel berdasarkan seberapa jauh itu dari sel tujuan. Berdasarkan itu, sel tujuan mendapat nilai 1. Jika robot ini berdiri pada sel dengan nilai 4, berarti itu akan mengambil robot 3 langkah (3 sel) untuk mencapai sel tujuan. Algoritma ini mengasumsikan bahwa robot tidak bisa bergerak secara diagonal dan hanya bisa membuat 90 derajat bergantian.

IV. HASIL DAN ANALISA

A. Implementasi Sistem

Mobile robot micromouse yang dibuat ini bertujuan untuk mencari rute terpendek untuk mencapai tujuan pada tengah labirin. Mobile robot ini micromouse dibuat menggunakan aklirik dengan ukuran 12cm x 10cm, dengan tinggi robot 7cm. Mobile robot micromouse dapat dilihat pada Gambar 4.1.



Gbr 12. Mobile Robot Micromouse

B. Implementasi Alat

Dalam pembuatan *mobile robot micromouse* seperti pada Gambar 4.2, terdiri dari arduino, motor DC, motor *shield* L298P, baterai 8,23 Volt, dan sensor ultrasonik. Pada implementasi alat ini, menggunakan *supply* dari baterai.



Gbr 13. Mobile Robot Micromouse

C. Pengujian Sensor *Ultrasonic*

Sensor *ultrasonic* merupakan sensor yang digunakan untuk mendeteksi halangan yang akan dilewati oleh *mobile robot micromouse*. Sensor *ultrasonic* yang digunakan dalam pembuatan *mobile robot micromouse* ini adalah sensor *PING* yang berjumlah 3 buah. Dimana 1 buah sensor berfungsi sebagai sensor kiri, 1 buah untuk sebagai sensor tengah dan 1 buah sensor berfungsi sebagai sensor kanan. Inputan sensor *ultrasonic* akan dihubungkan ke pin digital yang ada pada arduino.

Dalam pengujian sensor *ultrasonic* yang digunakan pada robot *mobile robot micromouse* dilakukan dengan cara membandingkan antara data yang dibaca oleh sensor dengan data yang sebenarnya yang diukur dengan menggunakan mistar seperti pada Gambar 14. Hasil perbandingan pengukuran dapat dilihat pada Tabel 1:



Gbr 14. (a). Mengukur Jarak dengan Menggunakan Mistar dan (b) Mengukur Jarak dengan Menggunakan Sensor Ping Melalui Serial Monitor

Tbl 1. Tabel Pengujian Sensor *Ultrasonic*

Percobaan ke –	Jarak Sebenarnya	Jarak Yang Terukur Oleh Sensor <i>Ultrasonic</i>	Error
1	5 cm	5 cm	0%
2	10 cm	10 cm	0%
3	15 cm	15 cm	0%
4	20 cm	20 cm	0%
5	25 cm	25 cm	0%
6	30 cm	30 cm	0%
7	35 cm	35 cm	0%
8	40 cm	40 cm	0%
9	45 cm	45 cm	0%
10	50 cm	50 cm	0%

Nilai eror pada pembacaan sensor jarak adalah:

$$\% \text{ Error} = \frac{|\text{Jarak yang diinginkan} - \text{Jarak yang dibaca}|}{\text{Jarak yang diinginkan}} \times 100\%$$

$$\% \text{ Error} = ((10 - 10) / 15) \times 100\%$$

$$= 0\%$$

Pada pengujian sensor jarak ini nilai yang di dapatkan adalah 0 %. Jadi Sensor yang digunakan dapat berfungsi dengan baik dan dapat digunakan dengan pada penelitian ini.

D. Pengujian Motor Shield L298 Terhadap Kecepatan Motor

Pengujian motor shield L298 bertujuan untuk mengetahui hasil kecepatan PWM yang dihasilkan oleh motor DC 12V. Pengujian dilakukan dengan cara menghubungkan motor *shield* L298P dengan supply yang berasal dari baterai 7,4 Volt, dan motor DC. Tabel 4.2 adalah tabel pengujian tegangan keluaran dari kecepatan PWM.

Tbl 2. Pengujian Motor Shield L298

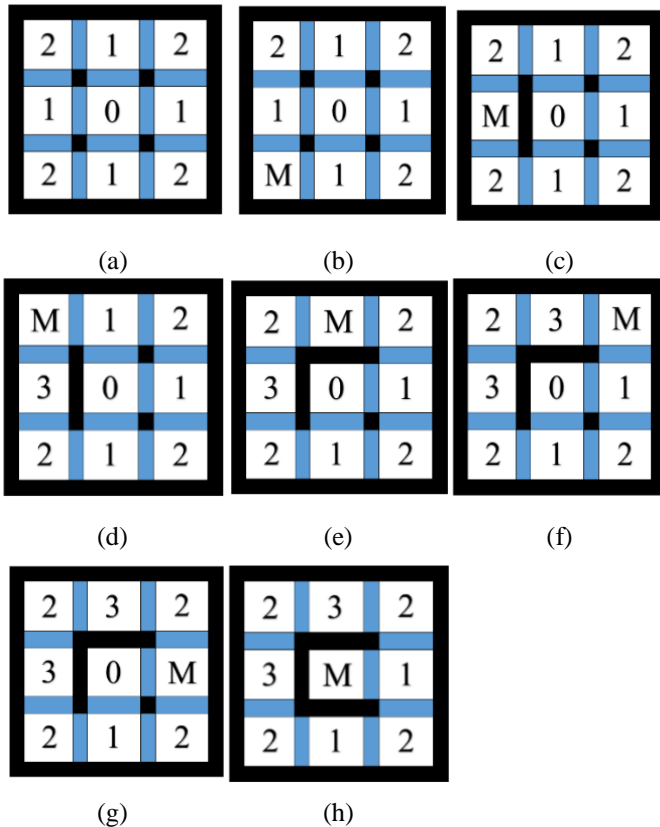
No	Tegangan Masukan (Volt)	Kecepatan PWM	Tegangan keluaran (Volt)	
			Motor 1	Motor 2
1	8,23	0	0	0
2	8,23	50	2,532	2,832
3	8,23	100	5,37	5,57
4	8,23	150	6,07	6,19
5	8,23	200	6,39	6,49
6	8,23	255	6,71	6,77

Pengujian yang dilakukan pada motor *shield* dengan memberikan tegangan masukan yang sama, didapat hasil tegangan keluaran pada motor 1 dan motor 2 berbeda, seperti terlihat pada Tabel 2. Hal ini terjadi karena tegangan keluaran

pada motor mempengaruhi kecepatan PWM, semakin besar tegangan keluaran maka semakin besar pula PWM yang dihasilkan

E. Pengujian Robot

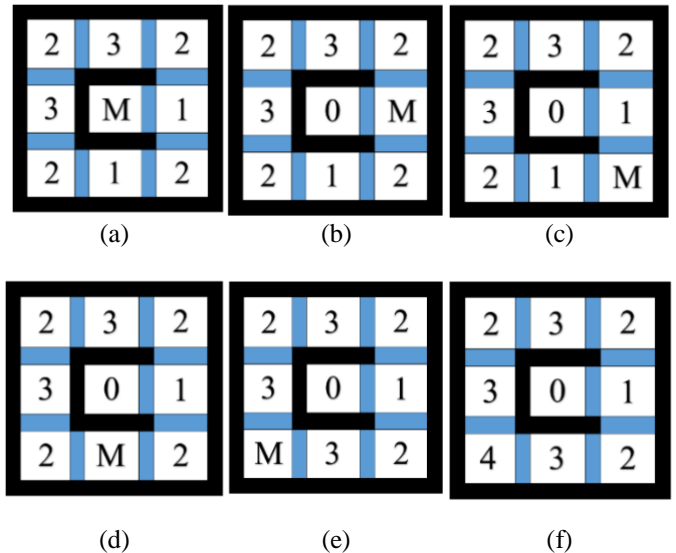
a. Proses Mapping Maze Menuju Pusat Maze



Gbr 15. Proses Mapping Maze Menuju Pusat Pada Maze 1

Pada Gambar 15, robot akan melakukan pemetaan labirin dan menuju titik pusat dalam sebuah labirin. Langkah pertama yang dilakukan oleh robot adalah menampilkan labirin flood fill yang telah dirancang, seperti pada Gambar 15(a). Kemudian robot akan mengidentifikasi posisi awal robot yang telah diberikan dan mengisialisasikan posisi tersebut dengan huruf "M", seperti pada Gambar 15(b). Setelah itu, robot membaca dinding yang ada disekitar robot dan robot bergerak menuju nilai terkecil. Saat robot bergerak, robot akan melakukan update nilai maze yang telah dilaluinya dan melakukan pemetaan labirin tersebut seperti Gambar 15(c). Langkah tersebut dilakukan oleh robot sampai robot mencapai pusat maze atau finish. Seperti yang terlihat pada gambar 15(d, e, f, g, h), dimana pada Gambar 15(h) robot telah mencapai titik pusat atau finish.

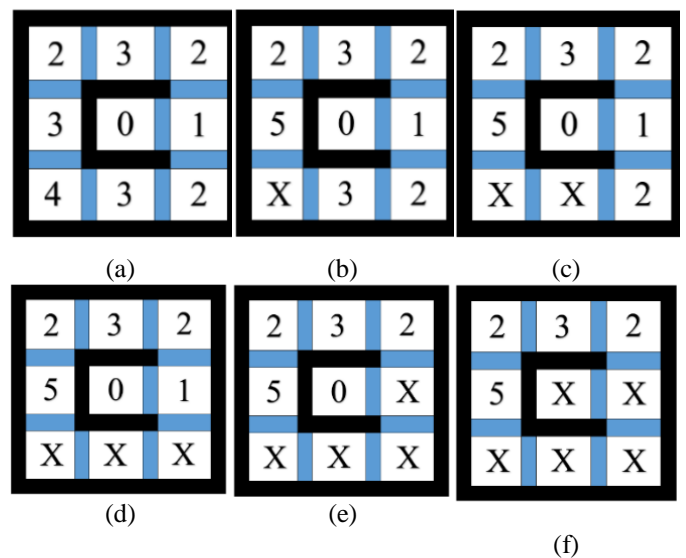
b. Proses Mapping Maze Menuju Lokasi Start



Gbr 16. Proses Mapping Maze Menuju Lokasi Start Pada Maze 1

Pada Gambar 16, robot akan melakukan pemetaan labirin dan menuju lokasi start dalam sebuah labirin. Langkah pertama yang dilakukan oleh robot adalah menampilkan labirin flood fill yang telah dipetakan pada proses menuju titik pusat, seperti pada Gambar 16(a). Setelah itu, robot membaca dinding yang ada disekitar robot dan robot bergerak menuju nilai terkecil. Saat robot bergerak, robot akan melakukan update nilai maze yang telah dilaluinya dan melakukan mapping pada maze tersebut seperti Gambar 16(b). Langkah tersebut dilakukan oleh robot sampai robot mencapai lokasi start. Seperti yang terlihat pada gambar 16(c, d, e), selanjutnya robot akan menampilkan labirin flood fill yang telah dipetakan semuanya, seperti yang terlihat pada Gambar 16(f).

c. Proses Rute Terpendek



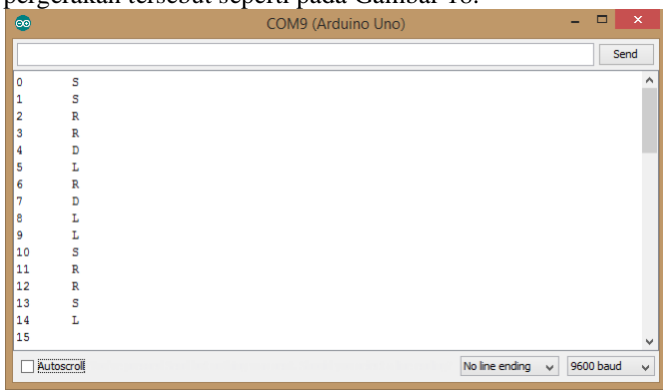
(f)

Gbr 17. Proses Menuju Titik Pusat Dengan Menggunakan Rute Terpendek Pada *Maze* 1

Pada Gambar 17 robot melakukan pergerakan menuju titik pusat dengan menggunakan rute terpendek yang telah dipetakan sebelumnya. Langkah pertama yang dilakukan oleh robot adalah menampilkan labirin flood fill yang telah dipetakan sebelumnya, seperti yang terlihat pada Gambar 17 (a). Setelah itu, robot akan bergerak menuju nilai terkecil dan menandai lokasi yang telah dilaluinya seperti Gambar 17(b). Langkah tersebut dilakukan berulang – ulang sampai robot mencapai titik pusat atau *finish*, seperti yang terlihat pada Gambar 17(c, d, e, f).

d. Hasil Pengujian

Setiap pergerakan robot akan disimpan sebuah data atau tujuan pergerakannya dalam sebuah EEPROM yang berguna untuk melihat jalur yang dibentuk oleh robot tersebut apa sesuai dengan *maze* yang telah dibuat, dimana hasil dari pergerakan tersebut seperti pada Gambar 18.



Gbr 18. Hasil Pembacaan Pergerakan Robot di EEPROM Pada *Maze* 1

Tbl 3. Pengujian Robot Pada *Maze* 1

Per cob aan Ke-	Hasil Pembacaan di EEPROM			Berha sil/ Gagal
	Hasil yang diperol eh	Hasil yang seharus nya	Keterangan	
1	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
2	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
3	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
4	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
5	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
6	Hasil yang diperol eh	Hasil yang seharus nya	Keterangan	Gagal

			Pada pergerakan yang ke- 10 robot salah dalam memetakan labirin, dimana pergerakan yang seharusnya adalah S tetapi robot bergerak menuju R. Sehingga pemetaan yang diperoleh tidak sesuai dengan <i>maze</i> yang sebenarnya.	
7	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
8	Hasil yang diperol eh	Hasil yang seharus nya	Keterangan	Gagal
			Pada pergerakan yang ke- 10 robot salah dalam memetakan labirin, dimana pergerakan yang seharusnya adalah S tetapi robot bergerak menuju L. Sehingga pemetaan yang diperoleh tidak sesuai dengan <i>maze</i> yang sebenarnya.	
9	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil
10	Hasil yang didapatkan sama seperti Gambar 4.7			Berha sil

Berdasarkan tabel 3, dari 10 kali pengujian yang dilakukan pada robot, dimana 8 kali pengujian robot berhasil melakukan pemetaan dan mencapai tujuan dengan menggunakan rute terpendek. Dan 2 kali pengujian mengalami kegagalan karena tidak sampai ke tujuan dan *mapping* yang dilakukan tidak sesuai dengan hasil sebenarnya. Hal ini disebabkan karena terjadinya slip pada roda robot, pengaturan PWM yang tidak sesuai dengan seharusnya, *design* robot yang kurang baik dan daya baterai yang mempengaruhi kecepatan motor. Tingkat keakuratan keberhasilan yang didapatkan sebesar 80%. Dengan menggunakan rumus:

$$akurasi = \frac{X}{N} \times 100\%$$

Keterangan :

X = Jumlah pengujian yang berhasil

N = Jumlah pengujian

$$\begin{aligned}
 akurasi &= \frac{X}{N} \times 100\% \\
 &= \frac{8}{10} \times 100\% \\
 &= 80\%
 \end{aligned}$$

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah dilakukan pengujian dan analisa yang telah dilakukan, maka dapat disimpulkan beberapa hal, yaitu :

1. Telah dibuat *mobile* robot *micromouse* dengan ukuran 12cm x 8cm x 7cm yang mampu diterapkan pada labirin atau *maze* dengan dimensi 20 cm x 20 cm.
2. Algoritma *flood fill* telah berhasil diterapkan untuk mencari rute terpendek dengan keberhasilan sebesar 80%.
3. *Design* robot dapat mempengaruhi jalan robot serta mempengaruhi pergerakan robot seperti posisi sensor dan posisi roda robot.

B. Saran

Pada penelitian ini masih terdapat kekurangan, maka diberikan beberapa saran agar penelitian ini dapat dikembangkan lebih lanjut, yaitu:

1. Penggunaan *webcam* untuk navigasi jalur.
2. Menggunakan mini PC lain yang memiliki RAM yang besar dari 512 Mb, dan prosesor yang cepat dari 700 MHz, agar bisa mendeteksi warna barang lebih cepat.

REFERENSI

- [1] Rusmini. 2011. [Pencarian Rute Terpendek Untuk Robot Micromouse Dengan Menggunakan Algoritma Backtracking](http://repo.eepis-its.edu/1459/1/[A%2DD206%2D9]_pp.41%2D48_Pencarian_Rute_Terpendek_Untuk_Robot_Micromouse_Dengan.pdf). URL : [http://repo.eepis-its.edu/1459/1/\[A%2DD206%2D9\]_pp.41%2D48_Pencarian Rute Terpendek Untuk Robot Micromouse Dengan.pdf](http://repo.eepis-its.edu/1459/1/[A%2DD206%2D9]_pp.41%2D48_Pencarian_Rute_Terpendek_Untuk_Robot_Micromouse_Dengan.pdf). Diakses pada tanggal 20 Februari 2015, pukul 15.30 WIB.
- [2] Anggoro, Beni. 2013. *Desain Pemodelan Kinematik dan Dinamik Humanoid Robot*. URL: http://eprints.undip.ac.id/41644/3/BAB_2.pdf. Diakses pada tanggal 10 Maret 2015, pukul 15.30 WIB.
- [3] Rahmawan, A. 2013. *Optimasi Gripper Dua Lengan dengan Menggunakan Metode Genetic Algorithm pada Simulator Arm Robot 5 DOF (Degree of Freedom)*. URL: http://eprints.undip.ac.id/41662/8/BAB_2.pdf. Diakses pada tanggal 13 Maret 2015, pukul 10.56 WIB.
- [4] Firdaus, Ahmad Riyad. 2005. *MicroMouse ELIND'BOT I*. URL: <http://www.jurnal.lipi.go.id/publikasi.cgi?tampilpublikasi2&1074192018&Indonesia&vol=24&1083846229>. Diakses pada tanggal 13 Maret 2015, pukul 17.13 WIB.
- [5] Rahmat, Ajang. 2014. *8 Komponen Utama Dalam Robot*. URL: <http://www.kelasrobot.com/2014/11/8-komponen-utama-dalam-robot.html>. Diakses pada tanggal 13 Maret 2015, pukul 11.48 WIB.
- [6] Pitowarno, E, 2006, *Robotika, Desain, Kontrol, dan Kecerdasan Buatan*, Penerbit Andi: Yogyakarta.
- [7] Fajar, Andi. 2014. *Sensor Ultrasonic*. URL: <http://kuliah.andifajar.com/sensor-ultrasonic/>. Diakses pada tanggal 22 Maret 2015, pukul 12.54 WIB.
- [8] Arifianto, B. 2009. *Modul Training Microcontroller For Beginer*. URL: <http://www.max-tron.com>. Diakses pada tanggal 22 Maret 2015, pukul 22.52 WIB.
- [9] Hari Sasongko, Bagus. 2012. *Pemrograman dengan Mikrokontroler AVR ATMEGA8535 dengan Bahasa C*. Andi: Yogyakarta.
- [10] Atmel. 2009. *ATmega48PA/88PA/168PA/328P Datasheet Summary*. URL: www.atmel.com/Images/8161s.pdf. Diakses pada tanggal 29 Maret 2015, pukul 17.00 WIB.
- [11] Arduino. *Arduino Uno*. Tanpa Tahun. URL: <http://arduino.cc/en/Main/ArduinoBoardUno>. Diakses pada tanggal 29 Maret 2015, pukul 20.00 WIB.
- [12] Djuandi, Feri. 2011. *Pengenalan Arduino*. URL: <http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>. Diakses pada tanggal 29 Maret 2015, pukul 20.57 WIB.
- [13] Rahman, Abdullah. M. N. 2010. *Penerapan Algoritma Flood Fill untuk Menyelesaikan Maze pada Line Follower Robot*. URL: repo.pens.ac.id/369/1/1115.pdf. Diakses pada tanggal 29 Maret 2015, pukul 21.10 WIB.
- [14] "Arduino Shield Manual". URL: <http://www.dfrobot.com/image/data/Common/Arduino%20Shield%20Manual.pdf> Diakses pada 30 Maret 2015, pukul 11.00 WIB.
- [15] Budiharto, W, 2006, *Belajar Sendiri Membuat Robot Cerdas*. PT Elex Media Komputindo: Jakarta.