

## Implementasi Algoritme *Mapping* Dengan *Backtracking* Pada *Mobile Robot* Dalam *Maze*

Fikri Fauzan<sup>1</sup>, Dahnia Syauqy<sup>2</sup>, Rizal Maulana<sup>3</sup>

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>fikrifauzan17@gmail.com, <sup>2</sup>dahnial87@ub.ac.id, <sup>3</sup>rizal\_lana@ub.ac.id

### Abstrak

*Autonomous Mobile Robot* marak dikembangkan dalam banyak penelitian terutama robot *Maze Wall Follower* yang dapat mencari jalan keluar dari sebuah labirin yang sangat rumit. Pada penelitian ini *Robot Maze Wall Follower* menggunakan Algoritme *Backtracking* yaitu menelusuri seluruh kemungkinan jalur yang ada pada *maze* dan mengenali setiap node yang dilewati agar tidak mengulangi jalur yang sudah dilewati. *Prototype* ini dibangun menggunakan beberapa modul yaitu Arduino Nano sebagai pemrosesannya, sensor Ultrasonik, sensor Photodiode dan L298N Driver Motor. Hasil pengujian dari pengukuran jarak dinding menggunakan sensor Ultrasonik HC SR04 yang dilakukan pengujian sebanyak 4 kali dengan jarak yang berbeda-beda yang didapatkan semakin jauh jarak sensor ke dinding semakin kecil juga tingkat keakuratan sensor, pembacaan garis dilakukan dengan 3 warna berbeda dan setiap warna dilakukan 3 kali pengujian hasilnya hanya warna hitam yang tidak dapat memantulkan cahaya yang dapat dibaca oleh sensor photodiode. Pengujian sistem kendali robot, pada belokan kiri keberhasilan mencapai 50% pada belokan kanan mencapai 70% dan ketika putar balik hanya mencapai 40%. Setelah melakukan pemetaan *maze* menggunakan Algoritme *Backtracking* semua jalur yang telah dilewati robot akan ditandai, setelah semua jalur itu sudah tertanda robot maka robot akan berhenti bergerak.

**Kata kunci:** *Mobile Robot, Mapping, Backtracking, Maze*

### Abstract

*Autonomous Mobile Robot* is rife in many researches especially the *Maze Wall Follower* robot that can find its way out of a very complicated labyrinth. In this study the *Robot Maze Wall Follower* using *Backtracking* algorithm is to trace all possible paths in the maze and recognize each node that is passed so as not to repeat the path that has been passed. This prototype is built using several modules of Arduino Nano as its processing, Ultrasonic sensor, Photodiode sensor and L298N Driver Motor. Test results from measurement of wall distance using HC SR04 Ultrasonic sensor which is tested as much as 4 times with different distance obtained by the distance of sensor distance to wall the smaller also the level of accuracy of sensor, line reading done with 3 different color and every color is done 3 times the test result is just a black color that can't reflect light that can be read by the photodiode sensor. Testing the robot control system, the left turn of success reaches 50% at the right turn reaches 70% and when the turnaround only reaches 40%. After mapping the maze using the *Backtracking* algorithm all paths that have been passed by the robot will be marked, after all the lines have been robot so the robot will stop moving.

**Keywords:** *Mobile Robot, Mapping, Backtracking, Maze*

## 1. PENDAHULUAN

*Mobile robot* menjadi salah satu yang belakangan ini menarik minat para peneliti untuk dikembangkan. Suatu robot yang dapat bergerak sendiri dari suatu tempat lain disebut *mobile robot* (Hartanto, 2005). Fungsi *mobile robot*

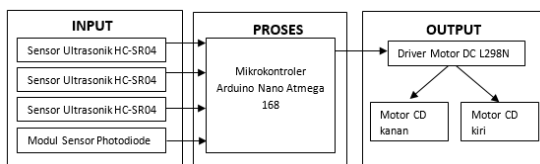
dapat meniru dari fungsi makhluk hidup yang dapat berpindah dari satu tempat ke tempat lain, robot ini dirancang untuk berbagai keperluan seperti transportasi, mengangkut barang, alat untuk hiburan dan lain-lain.

Robot *micromouse* mempunyai 2 tipe yaitu dalam *solving maze* dan tipe *mapping*. Tipe

*solving* robot akan melalui semua jalur yang ada pada *maze*, sedangkan tipe *mapping* robot akan bergerak sesuai dengan data yang tersimpan pada memori pada *microcontroller* yang di dapatkan dari tipe *explorasi*. Bahasa pemrograman yang digunakan adalah bahasa C dan *microcontroller* yang biasanya digunakan adalah sebuah modul *microcontroller* 168 yang biasa disebut *Arduino* serta Algoritme yang dipakai adalah *backtracking*. Algoritme *Backtraking* adalah Algoritme yang berdasar pada Algoritme DFS (*Deep Field Search*) untuk menelusuri semua solusi permasalahan lebih mangkus dari Algoritme lain. *Backtracking* adalah Algoritme yang lebih kompleks dari Algoritme *Brute-Force*, kedua algoritma tersebut bekerja secara sistematis dalam mencari solusi dari masalah yang memiliki solusi yang kompleks (D.H. Lehmer, 1950). Prinsip dasarnya adalah mencoba semua kemungkinan yang ada untuk mencari solusi.

Pada penelitian yang akan dilakukan, projek ini menggunakan perangkat *Arduino Nano Atmega 168*. *Arduino* merupakan mikrokontroler yang dirancang untuk mengembangkan objek, mengambil dan memasukan atau input pada sensor dan dapat mengontrol berbagai jenis aktuator ataupun rangkaian output lainnya. *Robot ini* dapat menggunakan Algoritme *mapping* dengan *backtracking* yang diterapkan pada *explorasi maze*. Pada sistem yang ada pada *mobile robot*, peneliti menggunakan sensor ultrasonik yang berfungsi sebagai pendeteksi halangan tembok yang terdapat pada *maze*. Kemudian *Mobile robot* menentukan belokan setiap persimpangan dan arah jalur yang dilalui *Mobile robot* dan mengolahnya dengan metode *backtrcking* untuk *explorasi map* yang ada pada *maze*.

**2. PERANCANGAN DAN IMPLEMENTASI SISTEM**



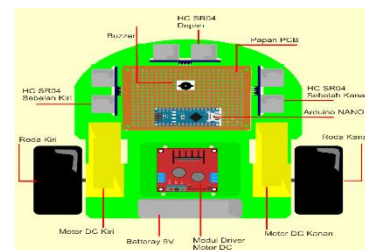
Gambar 1 Diagram Blok Sistem

Gambaran yang ditunjukkan oleh Gambar 1 tersebut adalah gambaran yang berkaitan dengan perancangan perangkat dari sistem yang akan dibuat. Pada bagian input terdapat 3 sensor Ultrasonik yang menghadap kanan kiri dan depan yang fungsinya untuk mendeteksi objek di depan kanan dan kiri robot. Sesnor photodiode

digunakan untuk mendeteksi garis hitam pada labirin. Pada bagian proses terdapat Mikrokontroler *Arduino Nano* untuk memproses semua data yang masuk pada sensor. Pada proses output terdapat 2 motor dc yang berfungsi sebagai aktuator atau penggerak dari robot. Pada 2 motor tersebut dikendalikan oleh driver motor dc L298N yang dapat mengendalikan arah putar dan kecepatan motor.

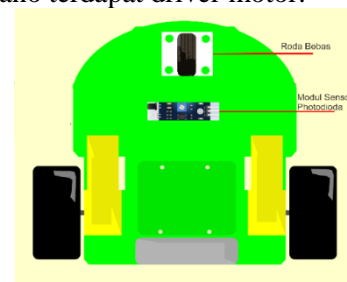
**2.1 Perancangan Perangkat Keras**

Dalam perancangan perangkat keras harus diperhatikan letak dari setiap komponen yang digunakan supaya pada tahap implementasi atau pada perakitan alat tidak menimbulkan masalah dan robot berfungsi sebagaimana fungsinya.



Gambar 2 Desain Robot Tampak Atas

Gambar 2 diatas robot menunjukkan rancangan letak setiap komponen yang dipasang pada bagian atas robot. Pada bagian depan menunjukkan 3 sensor ultrasonik yang masing-masing menghadap kanan kiri dan depan yang digunakan untuk menentukan jarak objek. Mikrokontroler *Arduino Nano* terletak pada bagian tengah robot. Pada bagian belakang *Arduino Nano* terdapat driver motor.



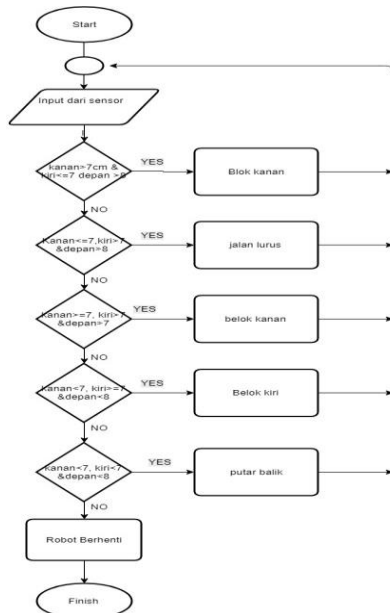
Gambar 3 Desain Robot Tampak Atas

Gambar 3 menunjukkan robot bagian bawah. Pada bagian depan menunjukkan roda 360 derajat yang dapat berputar ke segala arah. Pada bagian tengah terdapat sensor Photodiode yang mendeteksi garis hitam pada labirin. Pada bagian samping kanan dan kiri terdapat aktuator robot yang berupa *gearbox* motor dc.

**2.2 Perancangan Perangkat Lunak**

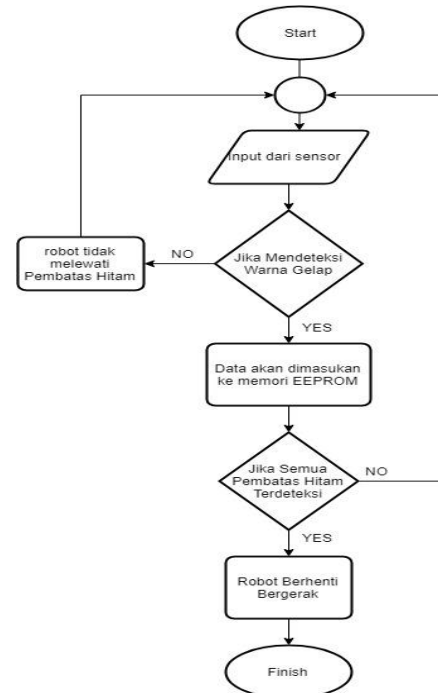
Pada Gambar 4 menunjukkan *flowchart*

perancangan perangkat lunak robot. Pada gerakan dasar robot dalam menelusuri labirin. Robot akan memprioritaskan gerakan belok kanan. Jika ada objek di depan maka robot akan belok kanan. Jika ada objek di kiri dan kanan maka robot akan jalan lurus. Jika ada objek di kanan dan depan maka robot akan belok kiri. Jika ada objek di ketiga sisi yaru depan kiri dan kanan maka robot akan putar balik.



Gambar 4 Follow Right

Gambar 5 menjelaskan algoritma dari sistem tersebut. Sensor yang digunakan untuk Algoritme ini adalah sensor Photodiode. Sensor Photodiode mendeteksi garis hitam yan terletak pada setiap batas node. Pada algoritma *Backtracking* robot akan menelusuri semua grid yang ada pada semua labirin. Labirin yang digunakan 4x4 dimana setiap robot melewati garis hitam maka data dari robot apakah belok kanan, jalan lurus, belok kiri atau putar balik. Data terssebut akan disimpan ke memori EEPROM sampai robot dapat melewati semua node. Setiap alamatnya berisi 1 data node. Dan jika robot tidak mendeteksi garis hitam maka robot hanya akan memaca alamat yang sebelumnya ditulis dan menampilkannya ke serial monitor.



Gambar 5 Flowchart Algoritme Mapping dengan Input Sensor Photodiode

1	2	3	4
10	3	6	5
9	8	7	6
10	9	8	9

Gambar 6 Variabel pada Setiap Array

Gambar 6 meunjukkan rute robot dalam memetakan atau *mapping* pada labirin 4x4. Pada rute yang ditempuh robot setiap node pada proses *mapping* terdapat 26 step atau 26 node yang ditempuh diantaranya robot melakukan *backtrack* ke node yang telah di lewati agar robot dapat mengenali seluruh node yang ada pada labirin. Pada proses *mapping* robot dapat mengenali setiap node dengan setiap robot melewati garis hitam robot membaca sensor ultrasonik apakah node tersebut terdapat persimpangan atau tidak. Dengan mengenali setiap persimpangan yang ada pada setiap node maka robot dapat mengenali setiap node yang ada pada labirin.

**2.3 Implementasi Perangkat Keras**

Pada Gambar 7 di bawah ini ukuran dari robot adalah 14x 12 cm. Untuk bahan penunjangnya agar semua komponen melekat adalah acrylic yang menjadi landasan dari semua komponen. Pada sisi tengah robot terdapat pcb

tepat driver motor melekat dan untuk tempat baterainya terletak pada bagian bawah robot. Untuk karton yang digunakan adalah karton yang tebal dan dapat menampung dengan kuat semua komponen yang melekat padanya.



Gambar 7 Robot Bagian Atas

Yang melekat pada papan *acrylic* sebelah depan adalah 3 sensor ultrasonik yang ditempatkan di depan kanan dan kiri agar dapat mengetahui objek dinding disekitarnya. Arduino menempel pada bagian belakang *acrylic* dan disebelahnya terdapat roda yang terlihat dari atas robot. Untuk driver motor nya tidak langsung melekat pada *acrylic* akan tetapi driver motornya yang berada pada tengah robot.



Gambar 8 Robot Bagian Bawah

Pada bagian bawah terdapat dua motor dc kit yang terpasang pada bagian belakang karton. Untuk sensor photodiodanya terletak pada sebelah roda 360 derajat.

**2.4 Implementasi Perangkat Lunak**

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Gambar 9 Variabel pada Setiap Array

Pada setiap array dinamai setiap variabel yang digunakan contoh  $maze[0][0] = a$ ,  $maze[0][1] = b$  dan seterusnya. Pemberian nama pada setiap array akan memudahkan untuk mengakses array tersebut pada saat disimpan

pada EEPROM. Nilai dari sensor photodiode yang di dapat dari melewati *Grid* yang berwarna gelap akan terisi pada variabel yang telah tersimpan pada EEPROM. Variabel sudah terisi array tersebut akan dinyatakan sudah dilewati oleh robot. Jika robot melewati variabel atau *Grid* yang sudah tersisi maka nilai dari sensor Photodiode akan diabaikan dan robot akan terus sampai menemukan *Grid* yang belum terisi. Jika variabel di dalam array tersebut sudah terisi semua maka robot akan otomatis berhenti.

**3. PEGUJIAN DAN ANALISIS**

**3.1 Pengujian Sensor Ultrasonik HC SR04**

Tabel 1 Pengujian Sensor Ultrasonik Kiri

Sensor kiri		
Jarak yang di ukur(cm)	Jarak yang di ukur(cm)	Akurasi (%)
2	2	100
4	4	100
6	6	100
8	8	100
10	10	100
12	12	100
20	20	100
40	40	100
50	53	99.2
Rata-Rata Persentasi		99
Akurasi sensor kiri		

Tabel 2 Pengujian Sensor Ultrasonik Depan

Sensor depan		
Jarak yang di ukur(cm)	Jarak yang di ukur(cm)	Akurasi (%)
2	2	100
4	4	100
6	6	100
8	8	100
10	10	100
12	12	100
20	20	100
40	40	100
50	51	99.3
Rata-Rata Persentasi		99
Akurasi sensor tengah		

Tabel 3 Pengujian Sensor Ultrasonik Kanan

Sensor Kanan		
Jarak yang di ukur(cm)	Jarak yang di ukur(cm)	Akurasi (%)

2	2	100
4	4	100
6	6	100
8	8	100
10	10	100
12	12	100
20	20	100
40	40	100
50	55	98
Rata-Rata Persentasi Akurasi sensor kanan		99

Sensor Ultrasonik bekerja berdasarkan perbandingan dari berapa waktu yang ditangkap setelah gelombang dipancarkan. Semakin jauh benda maka semakin jauh benda maka waktu pantulan akan semakin lama sedangkan jika semakin dekat benda maka pantulan akan semakin cepat. Jadi semakin jauh pantulan gelombang maka akurasi dari sensor pun akan berkurang. Hasil dapat dilihat pada Tabel 1, Tabel 2 dan Tabel 3 pada jarak 50 cm akurasi sensor mulai berkurang.

**3.2 Pengujian Pembacaan Sensor Photodiode**

Hasil dari pengujian tingkat akurasi keberhasilan dari sensor Photodiode yang digunakan dalam sistem ditampilkan pada Tabel 2.

Tabel 4 Hasil Pengujian Sensor Photodiode

Warna Objek	Pengujian Ke-	Pembacaan Sensor Photodiode
Merah	1	Warna Terang
	2	Warna Terang
	3	Warna Terang
Putih	1	Warna Terang
	2	Warna Terang
	3	Warna Terang
Hitam	1	Warna Gelap
	2	Warna Gelap
	3	Warna Gelap

Berdasarkan analisis Tabel 4 Modul sensor Photodiode ada beberapa warna yang di uji yaitu merah, putih, hitam. Pada warna- warna tersebut hampir semua dapat memantulkan cahaya yang dapat ditangkap kembali oleh sensor Photodiode kecuali warna hitam. Karena warna hitam tidak dapat memantulkan cahaya yang dipancarkan LED.

**3.3 Pengujian Sistem Kendali Robot**

Hasil dari pengujian tingkat akurasi

keberhasilan dari L298N Driver Motor yang digunakan dalam sistem ditampilkan pada Tabel 5.

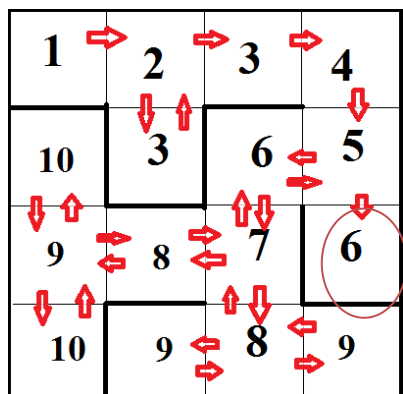
Tabel 5 Hasil Pengujian Pergerakan Robot.

Pergerakan	Percobaan	Keterangan	% Keberhasilan
Belok Kanan	1	Berhasil	70%
	2	Tidak Berhasil	
	3	Tidak Berhasil	
	4	Berhasil	
	5	Berhasil	
Belok kiri	1	Berhasil	50%
	2	Berhasil	
	3	Tidak Berhasil	
	4	Tidak Berhasil	
	5	Berhasil	
Putar Balik	1	Tidak Berhasil	40%
	2	Tidak Berhasil	
	3	Tidak Berhasil	
	4	Berhasil	
	5	Berhasil	

Untuk mengetahui tingkat keberhasilan dan kehandalan sistem kendali robot yang digunakan, maka dilakukan pengujian terhadap pergerakan robot. Gerakan-gerakan yang sering dilakukan oleh robot seperti maju lurus, putar kanan, putar kiri, putar kiri. Sama halnya ketika robot melakukan pergerakan berputar penggunaan motor driver berperan sangat penting agar ketepatan berputar selalu terpenuhi. Untuk membuktikan keberhasilan pergerakan robot, maka dilakukan pengujian sebanyak 5 kali. Pada saat Putar balik persentase paling kecil dikarenakan robot sering sekali mmenabrak dinding ynag ukurannya kecil untuk melakukan putar balik dengan mulus

**3.4 Pengujian Algoritme Mapping dengan Backtracking**

Hasil dari pengujian Algoritme Mapping dapat dilihat dari Gambar 9. yang menggambarkan arah jalan robot dan penomoran setiap jalan yang dilewati agar robot dapat menandai setiap jalan yang telah dilewati.



Gambar 4 Arah Jalannya Robot dan Hasil Mapping

Dari gambar di atas dapat dilihat start terdapat pada nomor 1 dan robot berhenti pada nomor 6. Setelah melewati salah satu *Grid maze* robot memberi tanda berupa nomor yang tertera pada gambar yang artinya robot telah melewati *Grid* tersebut. Jika robot melewati *Grid* yang sama untuk mencari *Grid* yang kosong *Grid* yang pernah dilewati tidak dapat di beri tanda lagi agar tidak terjadi perulangan.

**4. KESIMPULAN**

Dari hasil pengujian sensor ultrasonik sebanyak 7 kali dengan jarak yang berbeda-beda hasilnya semakin jauh keberadaan sensor dengan objek maka semakin tidak akurat juga hasilnya. Sedangkan untuk mendeteksi garis pada setiap batas node pada dasar labirin menggunakan modul sensor photodiode yang pengujiannya menggunakan 3 warna berbeda yaitu merah, putih dan hitam. Setiap warna dilakukan pengujian sebanyak 3 kali. Hasilnya hanya warna hitam yang tidak dapat memantulkan cahaya yang dapat diterima oleh sensor photodiode.

Setiap node pada labirin ditandai dengan garis hitam yang membatasi antara satu node dengan yang lainnya. Dari garis hitam tersebut robot dapat mengenali setiap node dengan bantuan sensor photodiode. Setiap sensor membaca garis hitam maka nilai dari sensor akan tersimpan di dalam Array yang terdapat pada memori EEPROM yang nilainya akan terus tersimpan meskipun robot mati nilai pada memori EEPROM akan terus ada. Dari nilai yang dimasukkan ke memori EEPROM robot dapat mengenali setiap node yang telah dilewati robot. Keakuratan dari pengujian sensor mencapai 100% pada setiap warna yang diuji.

Dengan metode dasar *follow right* pada robot maka robot akan meng eksplor maze dengan memprioritaskan belok kanan. Robot akan terus bergerak jika ada array yang masih kosong. Agar robot tidak dapat melewati jalur yang sama maka nilai dari sensor photodiode akan masuk ke array dan jika array itu sudah penuh maka robot otomatis akan berhenti bergerak. Ketepatan dalam sensor mendeteksi garis hitam dan datanya langsung tersimpan dalam memori EEPROM mencapai 100%

**5. DAFTAR PUSTAKA**

Ahyar, S. (2011). Pelacak Jarak Untuk Navigasi Gerak Robot.

Dong W, K., Ty A, L., & Steven A, V. (2013). Autonomus Multi-mobile Robot System: Simulation and Implementation Using Fuzzy Logic. 1.

Hendriawan, A., & Akbar, R. (2017). Penyelesaian Jalur Terpendek dengan menggunakan Algoritme Flood Fill pada Line Maze . 1.

Tirian, & Gelu, O. (2015). *Maze Solving Mobile Robot*.