

Perancangan Robot Maze Solving Berbasis Arduino

Rahmad Doni¹, Muhammad Reza Fahlevi², Elvin Syahrin³

¹Rekayasa perangkat Lunak, ^{2,3}Rekayasa Sistem Komputer
Universitas Potensi Utama

KL.Yos Sudarso Km. 6,5 No.3A Telp (061) 6640525

¹rahmaddoni113@gmail.com, ²ezafahlevi72@gmail.com, ³elvinsyahrin@gmail.com

Abstract

The maze solving robot is a development of the line follower robot. Where the robot can trace a path that has an intersection. In tracing a maze, apart from finding a way out / finish, it will be more efficient if the robot can find a shorter path through the maze. From the maze samples made, the robot is able to explore a maze and find the shortest path. When the mapping process is carried out, the robot traces 12 path codes, after the path simplification process is carried out the robot is able to shorten the path it takes to 6.

Keywords: *Arduino Uno, Maze solving, line follower, sensor obstacle*

Abstrak

Robot maze solving merupakan pengembangan dari robot line follower. Dimana robot sudah dapat menelusuri jalur yang memiliki persimpangan. Dalam menelusuri sebuah labirin, selain menemukan jalan keluar/finish, akan lebih efisien apabila robot dapat menemukan jalur yang lebih pendek dalam menelusuri labirin. Dari sampel maze yang dibuat, robot mampu menelusuri sebuah labirin dan menemukan jalur terpendeknya. Pada saat proses mapping dilakukan, robot menelusuri 12 kode jalur, setelah proses penyederhanaan jalur dilakukan robot mampu mempersingkat jalur yang di lalui menjadi 6.

Kata kunci: *Arduino Uno, Maze solving, line follower, sensor obstacle*

1. PENDAHULUAN

Teknologi yang terus berkembang sekarang ini telah menciptakan berbagai kemajuan dibidang teknologi, terlebih lagi pada teknologi dibidang robotika. Dimana perkembangan ini telah menciptakan berbagai jenis robot dengan kemampuan yang berbeda-beda. Salah satunya adalah robot maze solving, dimana robot ini akan melewati suatu jalan yang rumit yang disebut maze. Dalam menelusuri sebuah labirin, selain menemukan jalan keluar atau titik finish, robot akan lebih efisien jika dapat menemukan rute terpendek dari labirin tersebut. Hal ini tentunya membutuhkan sistem kendali yang bisa membuat robot mampu melewati maze dengan baik dan dengan tingkat error seminimal mungkin. Teknik pengenalan maze juga sangat dibutuhkan pada sistem kontrol robot. Ini bertujuan agar robot bisa melakukan pemetaan terhadap maze yang akan dilewati tersebut kemudian menemukan jalan keluar untuk rute terpendeknya. Dengan demikian, waktu yang ditempuh untuk mencapai tujuan menjadi lebih efektif.

Labirin atau yang dapat disebut dengan maze merupakan jaringan jalan yang kompleks dan memiliki banyak percabangan. Robot beroda dalam mengatasi jalan rumit dan memiliki banyak bercabangan dapat menggunakan metode simple maze [1]. Terdapat dua jenis maze dalam

bidang robotika yaitu wall maze dan line maze. Wall maze merupakan labirin yang dibangun dari dinding tanpa atap. Sedangkan line maze merupakan labirin yang dibangun dari garis [2].

Robot maze solving merupakan robot mobile dengan aktuator motor serta memiliki proses kerja sama seperti robot Line Follower yaitu bekerja dengan mengikuti garis-garis yang telah disediakan yang menjadikan navigasi untuk robot dalam melakukan manuve [3].

Maze mapping pada umumnya di berbagai sumber menjelaskan dengan istilah path mapping yang konsep dasar dalam pencariannya mengikuti aturan wall follower (pada robot wall follower) atau left/right hand rule (pada robot line tracer) [4].

Maze solving adalah salah satu algoritma yang digunakan pada robot untuk mencari jalur terpendek dari sebuah line maze. Terdapat 2 aturan dalam algoritma ini, yaitu [5]:

a) Left Hand Rule

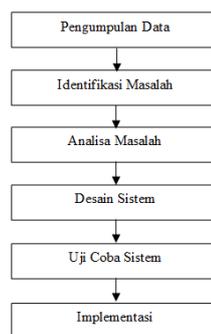
Dalam left hand rule, robot akan lebih memilih untuk belok kiri dari pada lurus atau belok kanan dan jika tidak ada belokan ke kiri akan lebih memilih lurus dari pada belok kanan.

b) Right Hand Rule.

Dalam right hand rule, robot akan lebih memilih belok kanan dari pada lurus dan lebih memilih lurus dari pada belok kiri.

2. METODOLOGI PENELITIAN

Kerangka kerja dalam melakukan penelitian dianggap perlu dikarenakan dapat mencapai hasil yang maksimal dalam penelitian seperti yang terdapat pada gambar 1.



Gambar 1. Kerangka Kerja Penelitian

Berdasarkan kerangka kerja pada gambar 1, langkah-langkah kerangka kerja tersebut dapat diuraikan seperti berikut :

a) Pengumpulan Data

Data dikumpulkan dari berbagai sumber. Adapun Metodologi pengumpulan data yang dilakukan dalam penelitian ini adalah studi Literatur. Penelitian yang dilakukan dengan cara mempelajari teori-teori / literatur dan buku-buku ilmiah dan referensi-referensi yang

berhubungan dengan objek dalam penelitian ini. Literatur yang dipelajari disini adalah literatur yang berhubungan dengan Sistem robot maze solving.

b) Identifikasi Masalah

Identifikasi masalah yang diperoleh dari penulisan ini adalah bagaimana cara agar robot dapat menyelesaikan sebuah jalur labirin dengan baik, serta mampu untuk membuat jalur terpendek pada labirin tersebut.

c) Analisis Masalah

Membuat robot dapat menyelesaikan sebuah labirin dengan mengimple mentasikan metode *shortest path*.

d) Desain

Dalam tahap pendesainan merupakan tahap dimana penulis mendesain robot. Dimana penulis harus mengatur tataletak dari setiap komponen agar tertata dengan rapi dan berfungsi dengan baik.

e) Uji Coba Sistem

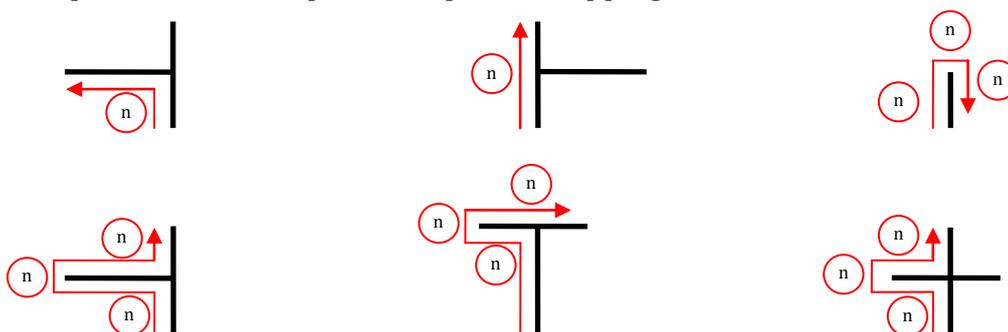
Pada tahap Uji coba, dilakukan percobaan dengan menggunakan beberapa jalur labirin untuk melihat respon robot apakah sudah sesuai atau belum

f) Implementasi

Setelah Melalui proses uji coba, kini robot maze solving sudah siap digunakan untuk menelusuri jalur labirin dengan baik.

2.1. Algoritma Maze Mapping

Algoritma maze mapping memiliki prinsip dasar, dimana ketika robot menjumpai persimpangan maka robot akan mengutamakan untuk belok kiri dibanding lurus atau belok kanan, bila tidak ada pilihan belok kiri maka lurus. Sehingga bila tidak dijumpai pilihan belok kiri maka robot akan mengambil jalan lurus. Pada gambar berikut merupakan contoh bagaimana cara pemberian kode pada saat proses mapping.



Gambar 2. Posisi saat pemberian kode

Pada saat pertama kali dijalankan, robot akan melakukan proses mapping dengan cara memberikan kode pada setiap kali robot menjumpai persimpangan dan jalan yang terputus, kemudian kode-kode yang sudah didapat akan tersimpan dalam memory robot. Kode tersebut akan tersimpan



selama robot belum mencapai posisi target(finish). Adapun kode-kode yang digunakan pada saat mapping adalah:

- a) "L" berarti left. Ini menandakan bahwa robot telah melakukan belok kiri karena melewati persimpangan.
- b) "F" berarti forward atau jalan terus. Ini menandakan kalau robot melakukan jalan terus karena bertemu dengan persimpangan tiga dengan pilihan lurus atau belok kanan.
- c) "O" berarti robot berjumpa dengan jalan yang terputus dan berjalan kembali ke persimpangan yang terakhir.
- d) "B" merupakan kode untuk mempersingkat jalur, dimana kode ini digunakan untuk membuang jalur-jalur yang tidak akan dilalui robot.

Setelah kode-kode tersusun, dilakukan penyederhanaan jalur untuk menemukan jalan keluar dari maze tersebut. Sehingga akan diperoleh susunan kode-kode yang baru. Kunci dari penyederhanaan jalur ini adalah bila tidak ada lagi kode "O" yang menandakan bahwa robot sudah tidak bertemu dengan jalan terputus lagi. Proses penyederhanaan ini dapat dilihat pada tabel 1 berikut.

Tabel 1. Konversi Jalur Tahap 1.

Kode 1	Kode 2	Kode 3	Hasil	Gambar	Gambar Hasil penyederhanaan
L	O	L	F		
F	O	L	R		
L	O	F	R		
F	O	F	B		
R	O	L	B		
L	O	R	B		

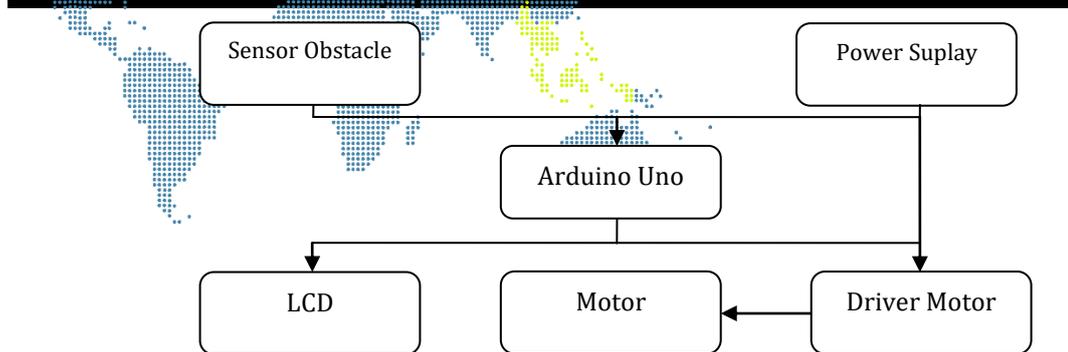
Proses penyederhanaan jalur dapat dilakukan dengan satu kali konversi. Tetapi jika pada konversi pertama ditemukan kode "B" maka harus disederhanakan lagi dengan aturan konversi 2. Proses penyederhanaan tersebut dapat dilihat pada tabel 2.

Tabel 2. Konversi Jalur Tahap 2.

Kode 1	Kode 2	Kode 3	Hasil	gambar	Gambar Hasil penyederhanaan
F	B	-	O		
F	B	F	O		
F	B	L	R		
R	B	L	O		
L	B	L	F		
L	B	R	B		

2.2. Kerangka Rancangan

Kerangka rancangan yang dibuat berupa rancangan dalam bentuk diagram. Diagram tersebut merupakan rancangan yang dapat dimengerti dengan lebih mudah, kerangka tersebut dapat dilihat pada gambar 3



Gambar 3. Diagram Blok.

Pada gambar blok diagram diatas, sensor obstacle difungsikan untuk membaca garis atau jalur yang akan dilalui robot. Dalam pengontrolan motor DC, menggunakan Driver motor yang daya nya langsung di ambil dari power suplay agar tidak membebani daya pada arduino.

3. HASIL DAN PEMBAHASAN

3.1. Pengujian Gerak Robot

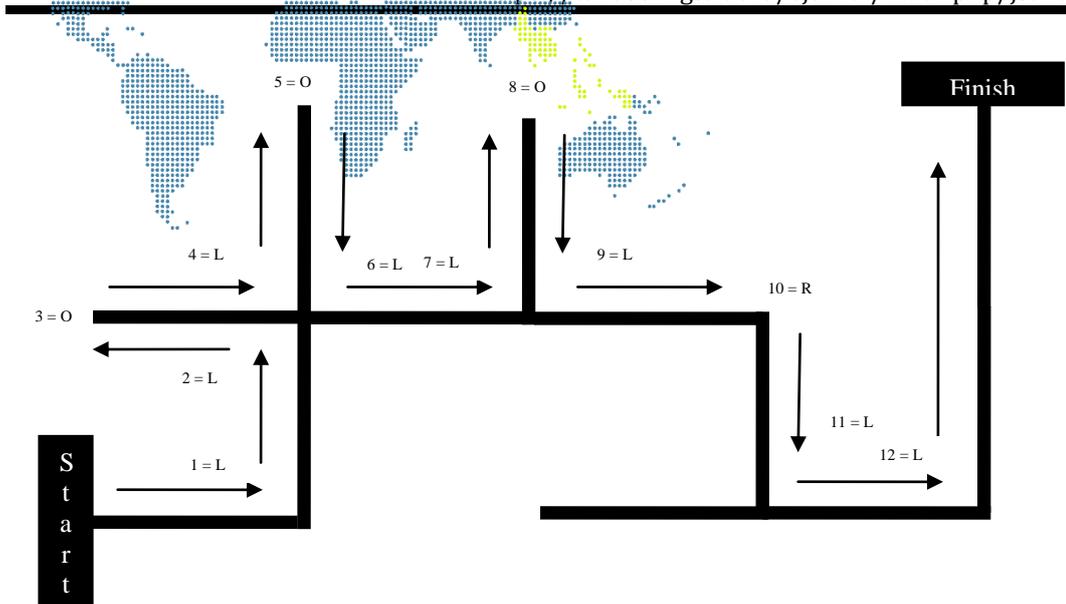
Pada tahap pengujian gerak robot, dilakukan penyesuaian arah putaran roda agar robot dapat bergerak sesuai dengan yang jalur yang ada. Perubahan arah roda di atur melalui Driver motor dengan inputan PWM dari arduino. Hasil pengujian dapat di lihat pada tabel 3.

Tabel 3. Pengujian Arah Gerak Robot.

No	Kondisi	Putaran Motor DC		Arah Putaran Motor DC	
		Kanan	Kiri	Kanan	Kiri
1	Maju	Berpuputar	Berputar	Ke depan	Ke depan
2	Belok Kanan	Berhenti	Berputar	Diam	Kedepan
3	Belok Kiri	Berputar	Berhenti	Kedepan	Diam
4	Putar Balik	Berputar	Berputar	Kebelakang	Kedepan
5	Berhenti	Berhenti	Berhenti	Diam	Diam

3.2. Pengujian Maze Solving

Tahap pertama dalam pengujian ini adalah dengan melakukan proses Mapping, apabila melewati persimpangan dan jalan buntu, maka data array akan disimpan didalam memory sesuai dengan kode yang diberikan. pada percobaan ini menunjukkan adanya penyimpanan data array ketika robot melewati pertigaan, perempatan, jalan buntu, dan finish.



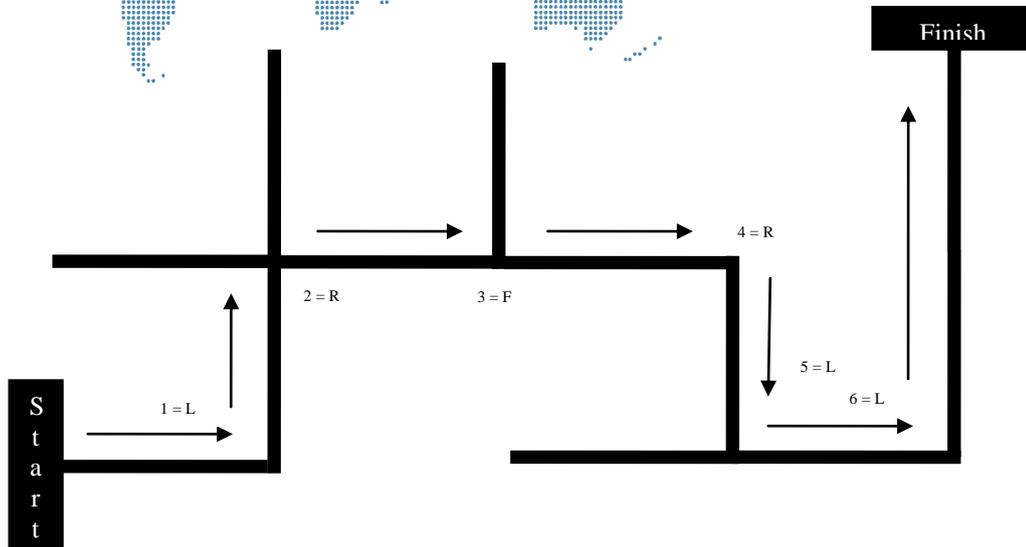
Gambar 4. Pengujian Proses Mapping.

Saat pengujian, robot diletakkan pada jalur Start. Kemudian robot melewati tikungan kiri pertama maka robot akan menyimpan data array 1 = L. lalu robot menjumpai persimpangan pertama, dikarenakan robot memiliki acuan left line following, maka robot akan berbelok ke kiri, maka data array yang tersimpan adalah 2 = L. robot berjalan lurus dan menemukan jalan buntu yang mengharuskan robot harus berbalik arah, maka robot akan menyimpan data array 3 = O. selanjutnya robot menjumpai persimpangan kembali dan akan memilih berbelok ke kiri, maka data array 4 = L. selanjutnya robot kembali menjumpai jalan buntu, maka data array 5 = O. setelah kembali, lagi – lagi robot menjumpai persimpangan dan mengambil jalur ke kiri maka data array 6 = L. Selanjutnya robot menjumpai persimpangan kembali dan berbelok ke kiri, maka data array 7 = L. kemudian robot menjumpai jalan buntu kembali, maka data array 8 = O. Robot kembali dan menemukan persimpangan dan berbelok ke kiri, maka data array 9 = L. selanjutnya robot mengikuti jalur berbelok ke kanan, data array 10 = R. kemudian robot menjumpai persimpangan dan berbelok ke kiri dan menyimpan nilai array 11 = L. kemudian robot mengikuti jalur berbelok ke kiri dan sampai ke finish dan menyimpan data array 12 = L. Dari proses mapping yang telah dilakukan, maka data array yang tersimpan pada robot adalah “LLOLOLLOLRLL”. Setelah proses mapping selesai, kemudian dilakukan penyederhanaan jalur sesuai dengan aturan konversi jalur 1 dan 2. Tabel penyederhanaan jalur percobaan ini dapat dilihat pada tabel 4.

Tabel 4. Penyederhanaan Jalur.

No	Jalur Awal	Kode 1	Kode 2	Kode 3	Hasil	Setelah disederhanakan
1	LLOLOLLOLRLL	L	O	L	F	LFOLLLOLRLL
2	LFOLLLOLRLL	F	O	L	R	LRLLOLRLL

No	Jalur Awal	Kode 1	Kode 2	Kode 3	Hasil	Setelah disederhanakan
3	LRLO RLL	L	O	L	F	LRFRL



Gambar 5. Hasil Penyederhanaan Jalur.

Gambar 5 menampilkan rute terpendek yang dilewati oleh robot menggunakan array yang sudah disederhanakan, yaitu “LRFRL”. Jalur yang dilewati robot tidak lagi memperhatikan acuan Left Following dan tidak melewati jalan buntu lagi. Pada saat melewati jalur terpendek, maze yang dilalui oleh robot harus sama dan titik start dan finish harus sama dan tetap seperti pada saat proses mapping sebelumnya.

4. SIMPULAN

Berdasarkan hasil dari tahapan perancangan dan implementasi sistem yang dirancang. Dapat disimpulkan bahwa:

- Robot yang dirancang mampu menelusuri maze dengan memperhatikan acuan Left Following dan telah mampu mencapai finish.
- Robot yang dirancang mampu membuat jalur terpendek berdasarkan aturan – aturan yang telah dibuat hal tersebut dibuktikan dengan jalur yang dilalui setelah proses mapping menjadi lebih sedikit, yakni dari jalur awal yang harus melewati 12 node/array diperkecil menjadi 6 node/array

DAFTAR PUSTAKA

- [1] Fajar, Y. H., Syauqy, D., & Maulana, R. (2019). Implementasi Maze Mapping pada Robot Line Follower untuk menentukan Shortest Path. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* e-ISSN, 2548, 964X.



- [2] Maarif, A. (2014). Perancangan Line Maze Solving Robot Dengan Algoritma Short Path Finder.
- [3] Prayudha, J., Saripurna, D., & Nugroho, N. B. (2017). Implementasi Backpropagation Untuk Pengenalan Warna Garis Lintasan Robot Maze Solving Berbasis Arduino. *Jurnal SAINTIKOM*, 16(2).
- [4] Nurmalasari, M., Triyanto, D., & Brianorman, Y. (2015). Implementasi Algoritma Maze Solving Pada Robot Line Follower. *Coding Jurnal Komputer dan Aplikasi*, 3(2).
- [5] Anshori, Y., Dodu, A. E., & Kurniawan, F. (2019). Perancangan Robot Penelusur Menggunakan Algoritma Dijkstra dan Metode Maze Solver. *Techno. Com*, 18(2), 166-177.