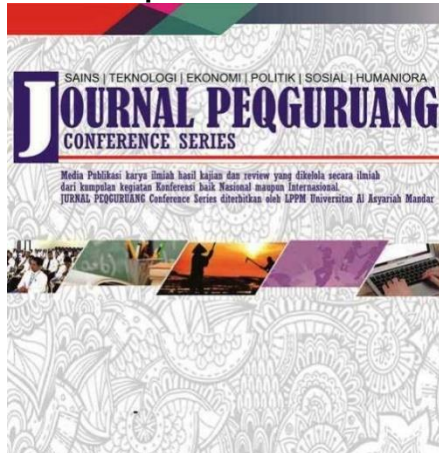


Graphical abstract



PENGEMBANGAN ALGORITMA PENCARI JALAN TERPENDEK PADA ROBOT PEMECAH LABIRIN GARIS MENGGUNAKAN VIRTUAL TURTLE SIMULATION

¹*Lutfi, ²Taufik Muchtar.

^{1,2}Politeknik ATI Makassar.

*Lutfi

lutfi@atim.ac.id

Abstract

This research aims to develop an algorithm for finding the shortest path on a modified robot designed to follow line paths, which is then used to discover the shortest routes within various types of line mazes. The research methodology involves utilizing a virtual turtle simulation using a Zumo robot equipped with six IR reflectance sensors for line tracking, brushed DC gearmotors, piezo sounder, a 3-axis gyroscope, and a compass sensor. The research process consists of two main stages: exploration, where the robot maps the line maze and stores information about it in EEPROM memory. Subsequently, in the optimization phase, the robot selects the shortest path to a predetermined destination point. Test results indicate that the labyrinth path without optimization requires 26 track movements, whereas with optimization, only six track movements are necessary. This leads to an approximately $\pm 77\%$ simplification efficiency.

Keywords: *Maze, Line Follower, Virtual Turtle Simulation.*

Abstrak

Penelitian ini bertujuan mengembangkan suatu algoritma pencarian jalur terpendek pada robot yang telah dimodifikasi untuk mengikuti jalur garis, yang kemudian digunakan untuk menemukan jalur terpendek dalam berbagai jenis labirin garis. Metode penelitian ini memanfaatkan simulasi virtual turtle dengan menggunakan robot Zumo yang telah dilengkapi dengan enam sensor reflektansi IR untuk melacak garis, brushed DC gearmotors, piezo sounder, 3-axis gyroscope, dan sensor kompas. Proses penelitian ini terdiri dari dua tahap utama, yaitu eksplorasi di mana robot akan memetakan labirin garis dan menyimpan informasi mengenai labirin tersebut ke dalam memori EEPROM. Selanjutnya, pada tahap optimasi, robot akan memilih jalur terpendek menuju titik tujuan yang telah ditentukan sebelumnya. Hasil uji coba menunjukkan bahwa jalur labirin tanpa optimalisasi memerlukan track movement sebanyak 26, sedangkan dengan penerapan optimasi, track movement yang dibutuhkan hanya enam. Hal ini menghasilkan efisiensi penyederhanaan sekitar $\pm 77\%$.

Kata kunci: *Labirin, Pengikut Garis, Simulasi Virtual Turtle*

Article history

DOI: [10.35329/jp.v5i2.4879](https://doi.org/10.35329/jp.v5i2.4879)

Received : 17/11/2023 | Received in revised form : 21/11/2023 | Accepted : 24/11/2023

1. PENDAHULUAN

Robot line follower adalah jenis robot yang dirancang untuk mengikuti garis atau lintasan tertentu secara otomatis. Robot ini dilengkapi dengan sensor atau alat pendeteksi yang memungkinkannya untuk mendeteksi perbedaan warna atau kontras antara lintasan yang diikuti dengan latar belakangnya. Dengan menggunakan sensor tersebut, robot dapat mengidentifikasi jalur yang harus diikuti dan mengubah arah atau kecepatannya sesuai dengan informasi yang diterima dari sensor tersebut. Tujuan utama dari robot line follower adalah untuk secara otomatis mengikuti jalur yang telah ditentukan tanpa adanya kendali langsung dari manusia. Robot line follower sering digunakan dalam kompetisi atau aplikasi praktis di mana navigasi otomatis berdasarkan jalur sangat dibutuhkan, seperti dalam industri, edukasi, atau hobi robotika. (Wahyu dkk, 2002), (Pitowarno, 2006), (Budiharto, 2010).

Untuk meningkatkan efisiensi dan performa robot, penulis telah melakukan berbagai penelitian, baik pada robot bergerak seperti robot pengikut garis maupun pada robot stasioner seperti lengan robot. Pada robot pengikut garis, pada tahun 2018, penulis berhasil mengembangkan Sistem Kendali PID pada Robot Pemantau Jalur dengan Sensor Auto Kalibrasi. Penelitian tersebut melibatkan pengujian pengendali P, PI, PD, dan PID saat regenerasi PWM, representasi sudut, dan efek swing. Sementara pada lengan robot, pada tahun 2019 dan 2020, penulis berhasil mengembangkan formula untuk menentukan Alamat EEPROM pada sistem navigasi yang merekam beberapa pola gerakan pada Lengan Robot Manipulator 3 DOF dengan gripper. Penelitian ini menguji sistem pada dua pola gerakan dengan enam posisi titik, mulai dari pengambilan objek hingga posisi kembali ke posisi siap. (Lutfi, 2018), (Lutfi, 2019), (Lutfi, 2020).

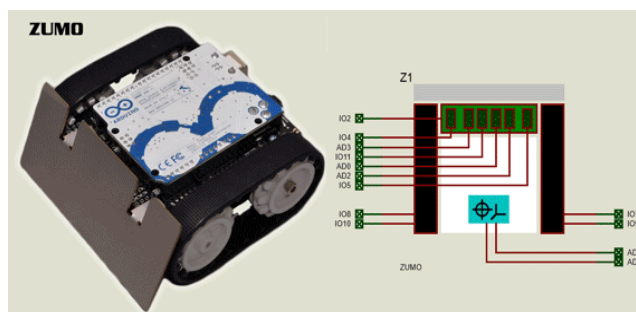
Tabel 1. Relevansi Literatur terhadap penelitian terkini

Tahun	Judul	Relevansi
2018	Rancang Bangun Sistem Kendali PID pada Robot Line Tracker dengan Sensor Auto Kalibrasi.	Robot line follower menggunakan sensor garis.
2019	Pengembangan Sistem Perekaman Gerakan Otomatis pada Lengan Robot Manipulator.	Perekaman data gerakan menggunakan fitur EEPROM
2020	Sistem Navigasi Perekaman Dua Pola Gerakan pada Lengan Robot Manipulator. Makassar.	

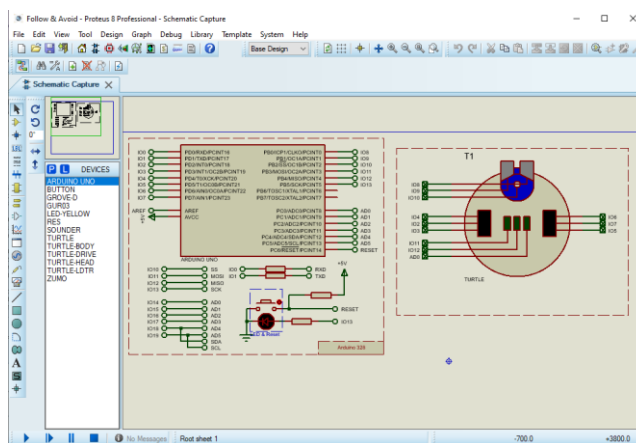
Dalam mata kuliah robotika industri di Politeknik ATI Makassar, mahasiswa mempelajari studi kasus robot pengikut garis dan lengan robot manipulator. Pada semester ini, dosen praktikum elektronika memilih robot pengikut garis sebagai materi pengajaran. Meskipun robot tersebut menggunakan pengendali utama berupa IC

OpAmp, prinsip kerjanya tetap sama dengan robot pengikut garis yang diajarkan dalam mata kuliah robotika industri. Hal ini mendorong penulis sebagai dosen pengampu mata kuliah robotika industri untuk meningkatkan prinsip kerja robot, agar tidak hanya mampu mengikuti garis, tetapi juga mampu mencari jalur terpendek pada labirin garis.

Salah satu kemampuan visual designer dalam aplikasi Proteus 8 adalah simulasi virtual turtle, yang merupakan simulasi robot populer seperti Funduino atau Robot Zumo yang dapat disimulasikan dalam skenario Line Following, Obstacle Avoidance, maupun Escape Maze. Dalam simulasi virtual turtle ini, memungkinkan untuk membangun lingkungan virtual untuk robot, berupa jalur garis yang akan diikuti, menghindari rintangan, atau menelusuri teka-teki labirin. Robot yang digunakan adalah robot Zumo yang dijalankan dalam simulasi virtual turtle dengan fitur-fitur seperti enam sensor reflektansi IR untuk mengikuti garis, brushed DC gearmotors, piezo sounder, 3-axis gyroscope, dan sensor kompas. (Proteus, 2023), (Arduino, 2023).



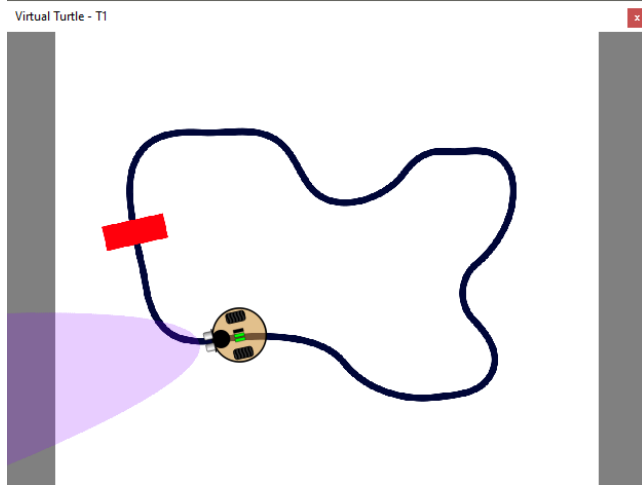
Gambar 1. Bentuk fisik dan simbol robot zumo



Gambar 2. Tampilan Virtual Turtle pada Proteus

Dengan adanya simulasi virtual turtle di proteus maka mahasiswa dalam proses pembelajaran robotika dapat lebih fokus pada algoritma dan logika proses dari robot itu sendiri. Tingkat kesulitan pada robot line follower dapat diatur melalui jalur lintasannya, apakah lintasannya cukup sederhana seperti pada gambar tiga,

atau bisa ditambahkan sebuah obstacle atau halangan. Tingkat kesulitan robot juga dapat berupa lintasan yang dibuat seperti labirin sehingga membutuhkan algoritma penyederhanaan. (Ma'arif dkk, 2019).



Gambar 3. Tampilan Line Follower sederhana

Berdasarkan penjelasan di atas, penelitian ini bertujuan mengembangkan algoritma pencarian jalur terpendek pada robot pengikuti jalur garis menggunakan simulasi virtual turtle. Adapun bentuk jalur line maze pada lapangan robot telah dibuat sendiri menggunakan aplikasi microsoft paint kemudian diekspor ke aplikasi simulasi virtual turtle pada proteus.

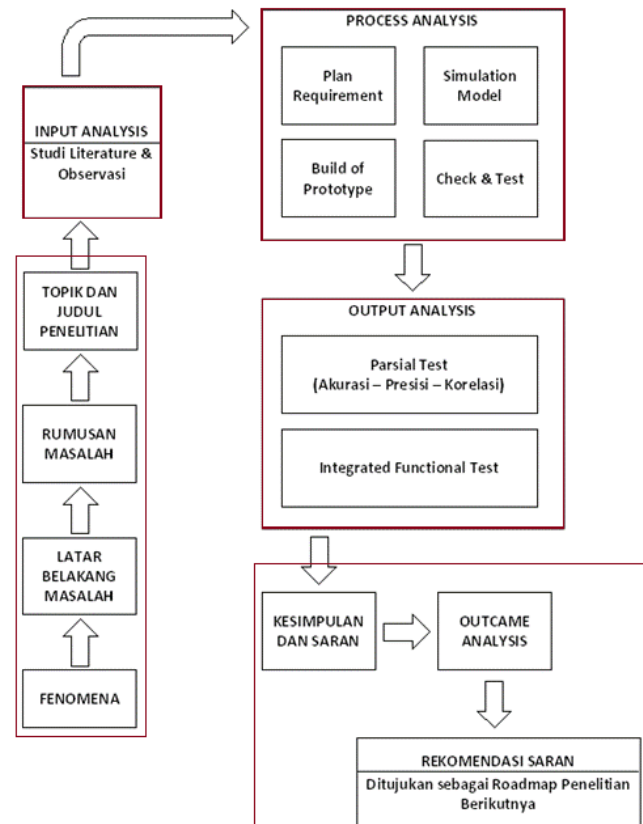
2. METODE PENELITIAN

Penelitian ini merupakan penelitian kuantitatif deduktif yang bersifat eksperimen melalui dua tahap, yaitu tahap eksplorasi dan tahap optimasi. Pada tahap eksplorasi, robot akan memetakan labirin garis dan menyimpan informasi labirin ke dalam memori EEPROM. Selanjutnya, pada tahap optimasi, robot akan memilih jalan terpendek menuju titik tujuan.

Data penelitian dapat dibagi menjadi data primer berupa data kalibrasi sensor garis, data logika motor penggerak, data informasi hasil eksplorasi labirin dan data hasil optimasi pencarian jalan terpendek. Selain itu juga terdapat data sekunder berupa kajian pustaka pendukung untuk simulasi terhadap parameter data sensor dan nilai optimasi pencarian jalan terpendek.

Penelitian ini melibatkan lima langkah yang terdiri dari *Define, Explore, Design, Test, dan Evaluate*. Tahap pertama dimulai dengan pengamatan terhadap suatu fenomena yang dapat dijadikan landasan masalah, yakni tentang kebutuhan materi kuliah robot yang dapat meningkatkan motivasi mahasiswa dalam mengikuti kompetisi robot. Setelah itu, dirumuskan masalah sesuai dengan tujuan penelitian serta ditetapkan judulnya. Pada langkah kedua, dilakukan eksplorasi melalui studi literatur dari penelitian terkait sebelumnya (Road Map Penelitian). Kemudian, pada tahap ketiga, dilakukan

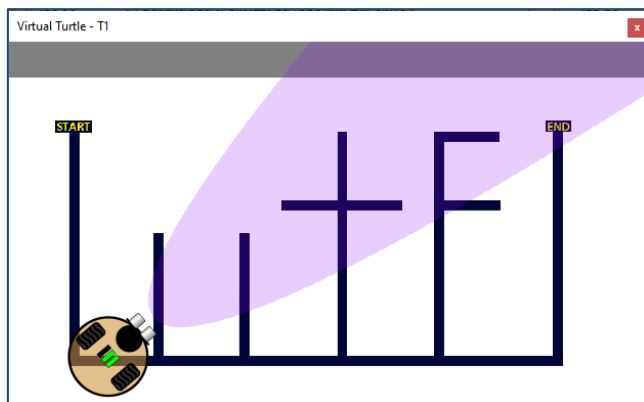
analisis proses lebih lanjut untuk menghasilkan model desain sistem yang sesuai. Model sistem tersebut diuji melalui simulasi menggunakan aplikasi virtual plant. Beberapa pengujian parsial seperti sensor garis dan stabilitas gerakan robot dalam mengikuti garis telah dilakukan pada penelitian sebelumnya. Fokus penelitian ini adalah kemampuan robot dalam menyederhanakan lintasan jalur hingga yang terpendek. Hasil dari pengujian simulasi akan dianalisis untuk menentukan tingkat efisiensi penyederhanaan lintasan jalur dengan membandingkan jumlah track movement sebelum dan setelah penyederhanaan.



Gambar 4. Tahapan Penelitian Robotika Industri

3. HASIL DAN PEMBAHASAN

Algoritma pencari jalan terpendek pada robot sumo digunakan untuk menemukan jalan terpendek pada labirin garis yang telah dibuat sesuai pada gambar 5 di bawah.



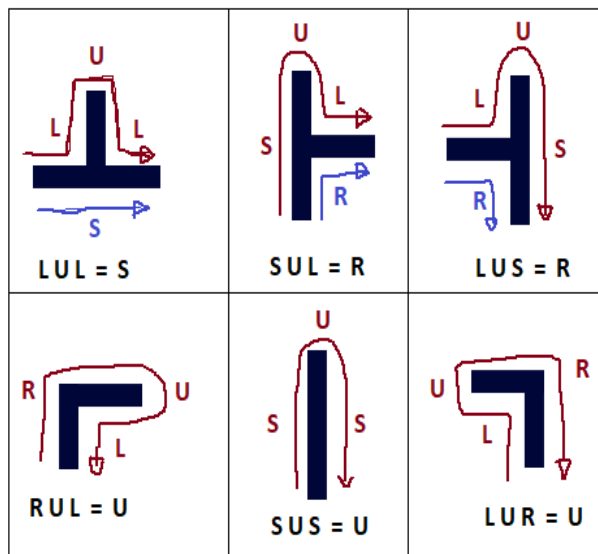
Gambar 5. Tampilan Labirin garis (Line Maze)

Algoritma ini terbagi menjadi dua mode, yaitu mode pencarian jalur terpendek dan mode optimasi. Pada mode pencarian, robot akan bergerak dari posisi START ke posisi END dengan mengikuti jalur garis pada labirin. Prioritasnya adalah melakukan belokan ke kiri di setiap persimpangan, kemudian melanjutkan perjalanan lurus jika tidak ada persimpangan, diikuti dengan belokan ke kanan, dan terakhir melakukan putaran balik. Rincian lebih lanjut dapat dilihat pada tabel di bawah ini.

Tabel 2. Inisialisasi pergerakan robot pengikut garis

Movement	Initialization
Belok Kiri / Turn Left	L
Jalan lurus / Move Straight	S
Belok Kanan / Turn Right	R
Putar Balik / U-turn	U

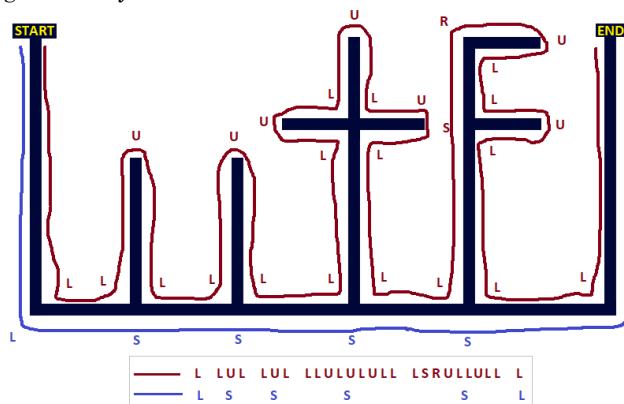
Mode berikutnya menyederhanakan jalur lintasan guna mencapai jalur terpendek dalam labirin. Pada fase ini, robot akan memanfaatkan fitur EEPROM untuk menyimpan inisialisasi setiap gerakan robot dan menyederhanakan jalur hingga tidak ada pergerakan balik (U-turn). Misalnya robot bergerak melalui jalur dengan tiga track movement, belok ke kiri, lalu putar balik, kemudian belok ke kiri. Maka penyederhanaannya cukup berjalan lurus saja. Untuk lebih jelasnya, dapat diamati kemungkinan-kemungkinan penyederhanaan jalur robot pada gambar enam.



Gambar 6. Penyederhanaan Jalur Robot (Optimasi)

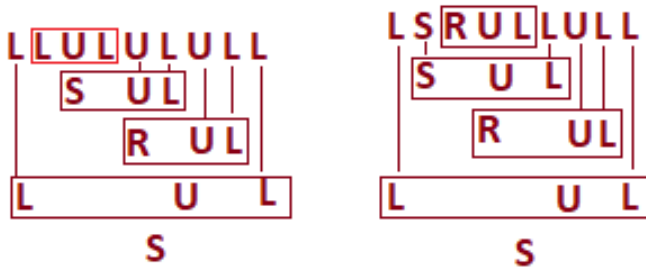
Pada gambar enam di atas dapat dilihat bahwa garis merah merupakan jalur normal (track movement) pada robot. Sedangkan garis biru merupakan hasil optimasi jalur terpendek (Short path). Terdapat enam kombinasi penyederhanaan yang mungkin terjadi yakni L-U-L = S, S-U-L = R, L-U-S = R, R-U-L = U, S-U-S = U dan L-U-R = U.

Misalkan labirin garis yang ada pada gambar lima akan disederhanakan maka akan terbentuk jalur seperti gambar tujuh berikut :



Gambar 7. Hasil Pengujian Simulasi Optimasi Robot

Garis berwarna merah merupakan jejak yang telah dilalui oleh robot dari awal hingga akhir. Garis berwarna biru merupakan hasil optimasi jalur terpendek. Untuk jalur L-U-L akan disederhanakan menjadi S. Sedangkan untuk optimasi jalur LLULULULL dan LSRULLULL menjadi S, dapat dilihat proses penyederhanaannya pada gambar berikut :



Gambar 8. Proses penyederhanaan jalur labirin

Pada ilustrasi proses penyederhanaan jalur labirin garis di gambar delapan, terlihat bahwa langkah optimal dilakukan untuk menghapus pergerakan putaran balik (U-turn) dengan merujuk pada gambar enam. Pada mode pencarian, robot menyimpan data-data track movement pada jalur lintasan menggunakan fitur EEPROM. Selanjutnya saat mode optimasi digunakan maka seluruh data track tersebut akan disimpan ke sebuah array satu dimensi untuk diolah menggunakan algoritma pencari jalan terpendek. Dengan bantuan perintah pengulangan (Looping), setiap track ke i akan diperiksa apakah terdapat track putar balik (U-turn) atau tidak. Jika terdapat track putar balik maka track ke $i-1$, track ke i dan track ke $i+1$ akan dianalisa sesuai data penyederhanaan jalur robot (sesuai gambar enam). Selanjutnya data array akan diupdate kembali sesuai dengan perubahan hasil penyederhanaan lintasan.

Pada lintasan jalur untuk huruf L, tidak terdapat pergerakan putaran balik (U-Turn) sehingga tidak terjadi penyederhanaan. Robot akan bergerak dengan satu track movement yakni belok ke kiri atau left (L). Persentasi efisiensi sebesar 0%.

Pada lintasan jalur untuk huruf U, terdapat dua buah pergerakan putar balik (U-Turn) yakni L-U-L dan L-U-L sehingga terjadi penyederhanaan sebanyak dua kali. Robot dua kali bergerak lurus atau straight (S). Penyederhanaan jalur dari enam track movement menjadi dua track movement memiliki persentasi efisiensi sebesar 66,67%.

Pada lintasan jalur untuk huruf T, terdapat tiga buah pergerakan putar balik (U-Turn) yakni L-L-U-L-U-L-U-L-L sehingga terjadi empat kali penyederhanaan yang pertama L-L-U-L menjadi L-S, kemudian yang kedua L-S-U-L menjadi L-R, selanjutnya yang ketiga, L-R-U-L menjadi L-U dan terakhir yang keempat, L-U-L menjadi S. Pada saat robot menlintasi huruf T maka robot akan bergerak lurus atau straight (S). Penyederhanaan jalur robot dari sembilan track movement menjadi dua track movement memiliki persentasi efisiensi sebesar 88,89%.

Pada lintasan jalur untuk huruf F, terdapat dua buah pergerakan putar balik (U-Turn) yakni L-S-R-U-L-L-U-L-L sehingga terjadi empat kali penyederhanaan yang pertama L-S-R-U-L menjadi L-S-U, kemudian yang kedua L-S-U-L menjadi L-R, selanjutnya yang ketiga, L-R-U-L menjadi L-U dan terakhir yang keempat, L-U-L menjadi S. Pada saat robot menlintasi huruf F maka robot akan bergerak lurus atau straight (S). Penyederhanaan jalur

robot dari sembilan track movement menjadi dua track movement memiliki persentasi efisiensi sebesar 88,89%.

Pada lintasan jalur untuk huruf I, tidak terdapat pergerakan putaran balik (U-Turn) sehingga tidak terjadi penyederhanaan. Robot akan bergerak dengan satu track movement yakni belok ke kiri atau left (L). Persentasi efisiensi sebesar 0%

Hasil pengujian menunjukkan bahwa jalur tanpa optimasi akan membutuhkan track movement sebanyak 26 buah sedangkan hasil optimasi hanya membutuhkan track movement sebanyak enam buah. Dengan demikian hasil optimasi yang didapatkan untuk jalur labirin garis yang telah dibuat pada gambar lima, memiliki nilai persentasi tingkat efisiensi penyederhanaan sebesar $\pm 77\%$.

4. SIMPULAN

Algoritma pencari jalan terpendek pada robot pemecah labirin garis menggunakan virtual turtle simulation telah berhasil dikembangkan. Hasil uji coba menunjukkan bahwa jalur labirin tanpa optimalisasi memerlukan track movement sebanyak 26, sedangkan dengan penerapan optimasi, track movement yang dibutuhkan hanya enam. Hal ini menghasilkan efisiensi penyederhanaan sekitar $\pm 77\%$.

Algoritma ini optimal untuk diimplementasikan pada lintasan bertipe lurus dengan tingkat keberhasilan 100%. Namun, perlu studi dan penelitian lebih lanjut untuk menerapkannya pada lintasan bertipe melingkar. Sensor garis menjadi komponen krusial pada robot pemecah labirin garis. Komponen ini sangat penting dalam desain robot. Sensor ini perlu dikonfigurasi dan diatur dengan benar untuk mendukung pengendali kecepatan motor dan algoritma Pencari Jalur Pendek dalam menyelesaikan labirin garis.

DAFTAR PUSTAKA

- Arduino. (2023, 3 10). Retrieved from <https://docs.arduino.cc/>
- Budiharto, W. (2010). Robotika - Teori dan Implementasinya. Yogyakarta: CV. Andi Offset.
- Lutfi. (2018). Rancang Bangun Sistem Kendali PID pada Robot Line Tracker dengan Sensor Auto Kalibrasi. Makassar: Politeknik ATI Makassar.
- Lutfi. (2019). Pengembangan Sistem Perekaman Gerakan Otomatis pada Lengan Robot Manipulator. Buku Prosiding Seminar Nasional Sains, Teknologi, dan Sosial Humaniora (pp. 1-7). Makassar: Universitas Indonesia Timur.
- Lutfi. (2020). Sistem Navigasi Perekaman Dua Pola Gerakan pada Lengan Robot Manipulator. Makassar: Politeknik ATI Makassar.

- Ma'arif, A., Iskandar, S., & Iswanto. (2019). New Design of Line Maze Solving Robot With Speed Controller and Short Path Finder Algorithm. *International Review of Automatic Control (I.R.E.A.CO.)*, 154-162.
- Pitowarno, E. (2006). *Robotika : desain, kontrol, dan kecerdasan buatan / Endro Pitowarno*. Yogyakarta: Andi.
- Proteus. (2023, 3 10). Retrieved from <https://www.labcenter.com/turtles/>
- Wahyu, T., Hartanto, D., & Prasetyo, Y. (2002). *Analisis dan Desain Sistem Kontrol dengan Matlab*. Yogyakarta: Andi.