Techno.COM, Vol. 24, No. 3, Agustus 2025: 669-684

Evaluasi Kinerja Robot Line Follower Dengan Algoritma LSRB Pada Lintasan Maze Bercabang

Performance Evaluation of Line Follower Robot with LSRB Algorithm on Branched Maze Trajectory

Nina*[1], Wawan Firgiawan*[2], Sulfayanti[3]
[1,2,3] *Teknik Informatika, Universitas Sulawesi Barat*
E-mail : ninaa883867@gmail.com*[1], wawanfirgiawan@unsulbar.ac.id*[2], sulfayanti@unsulbar.ac.id[3]
Received 23 May 2025; Revised 19 June 2025; Accepted 23 June 2025

Abstrak - Navigasi otonom merupakan komponen mendasar dalam pengembangan sistem robot bergerak, terutama saat beroperasi di lingkungan bercabang seperti labirin (*maze*). Pada sistem berbasis mikrokontroler yang memiliki keterbatasan kapasitas pemrosesan dan memori, dibutuhkan algoritma navigasi yang ringan, efisien, serta tidak bergantung pada penyimpanan data jalur. Salah satu solusi yang sesuai dengan kondisi tersebut adalah algoritma *Left–Straight–Right–Back* (LSRB), strategi navigasi berbasis aturan prioritas arah secara stateless. Pendekatan ini memprioritaskan arah kiri terlebih dahulu, diikuti oleh lurus, kanan, dan terakhir mundur saat robot menghadapi percabangan. Penelitian ini bertujuan untuk mengevaluasi efektivitas algoritma LSRB dalam mengarahkan robot *line follower* menuju titik tujuan pada lintasan bercabang dengan struktur yang kompleks. Pengujian dilakukan dengan tiga titik awal berbeda, yang masing-masing mewakili tingkat kompleksitas jalur yang bervariasi. Evaluasi kinerja difokuskan pada dua parameter utama, yaitu waktu tempuh dan kestabilan navigasi robot selama pencarian jalur. Hasil menunjukkan bahwa algoritma LSRB mampu mengarahkan robot secara konsisten hingga mencapai titik akhir yang ditentukan. Namun demikian, waktu tempuh yang dihasilkan bervariasi tergantung pada posisi awal dan struktur jalur. Titik awal A menunjukkan performa terbaik dengan rata-rata 5,29 detik, sementara titik B mencatat waktu tertinggi sebesar 25,51 detik. Temuan ini menunjukkan bahwa LSRB efektif untuk navigasi sederhana dan memiliki potensi untuk ditingkatkan melalui integrasi dengan algoritma pemetaan atau teknik pembelajaran mesin agar lebih adaptif dan optimal.
Kata kunci - Navigasi Otonom, Robot Line Follower, Algoritma LSRB, Labirin, Waktu Tempuh

Abstract - *Autonomous navigation is a fundamental component in the development of mobile robotic systems, particularly when operating in branched environments such as mazes. In microcontroller-based systems with limited processing and memory capacity, it is necessary to implement navigation algorithms that are lightweight, efficient, and independent of path data storage. One suitable solution for such conditions is the Left–Straight–Right–Back (LSRB) algorithm, a stateless navigation strategy based on directional priority rules. This approach prioritizes left turns first, followed by straight, right, and finally backtracking when the robot encounters intersections. This study aims to evaluate the effectiveness of the LSRB algorithm in guiding a line follower robot toward a goal point within a complex branched path. Experiments were conducted using three different starting points, each representing varying levels of path complexity. Performance evaluation focused on two main parameters: travel time and navigation stability during the route-finding process. The results showed that the LSRB algorithm could consistently guide the robot to the designated endpoint. However, travel time varied depending on the initial position and path structure. Starting point A yielded the best performance with an average of 5.29 seconds, while starting point B recorded the longest average time of 25.51 seconds. These findings suggest that LSRB is effective for simple navigation and has potential for improvement through integration with mapping algorithms or machine learning techniques to enhance adaptability and path optimization.*
Keywords - *Autonomous Navigation, Line Follower Robot, LSRB Algorithm, Maze, Travel Time*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

1. PENDAHULUAN

Robot *line follower* atau yang dikenal sebagai robot penjejak garis merupakan suatu sistem robotika yang dirancang untuk dapat bergerak secara otomatis dengan mengikuti pola jalur tertentu yang umumnya di tandai oleh garis berwarna kontras, seperti garis hitam pada permukaan putih atau garis putih pada permukaan hitam [1]. Prinsip kerja robot *line follower* didasarkan pada pemanfaatan sensor yang mampu mendeteksi perbedaan warna atau intensitas cahaya pada permukaan lintasan. Teknologi ini banyak diterapkan dalam bidang pendidikan, penelitian, dan pengembangan sistem otomatisasi, karena mampu memberikan pemahaman dasar mengenai konsep sistem kendali, pemanfaatan sensor, serta perancangan algoritma navigasi [2]. Tidak hanya terbatas pada lintasan lurus atau berkelok sederhana, robot *line follower* juga dapat diaplikasikan untuk menyelesaikan tantangan yang lebih kompleks, seperti *line maze. Line maze* merupakan lintasan berbentuk labirin yang mengharuskan robot untuk mampu menavigasi berbagai percabangan serta menentukan rute tercepat atau paling efisien guna mencapai titik tujuan secara optimal.

Robot *line follower* dilengkapi dengan sensor inframerah (IR) atau sensor LDR (*Light Dependent Resistor*) yang berfungsi membedakan warna garis lintasan dan latar belakangnya. Sensor ini memungkinkan sistem mengenali jalur secara visual berdasarkan perbedaan pantulan cahaya, sehingga robot dapat menyesuaikan arah geraknya sesuai dengan posisi garis yang harus diikuti [3]. Sensor-sensor tersebut terhubung ke mikrokontroler yang bertugas mengolah data dari sensor untuk mengatur gerakan motor penggerak di sisi kiri dan kanan. Mekanisme kendali ini umumnya didukung oleh algoritma logika atau pengendali berbasis PID (*Proportional-Integral-Derivative*), yang bertujuan menjaga presisi gerakan robot di atas jalur. Dalam kondisi lintasan lurus, robot menyeimbangkan kecepatan sedangkan saat menghadapi tikungan atau percabangan, sistem membutuhkan keputusan arah untuk memilih jalur yang tepat menuju tujuan.

Salah satu pendekatan navigasi yang umum digunakan pada percabangan atau tantangan maze solving adalah algoritma LSRB (*Left-Straight-Right-Back*). Algoritma ini bekerja berdasarkan prioritas arah, robot akan mencoba belok ke kiri terlebih dahulu, lalu lurus, kemudian ke kanan, dan terakhir mundur apabila ketiga arah sebelumnya tertutup. Sifat deterministik dari algoritma ini memungkinkan robot melakukan eksplorasi jalur secara sistematis tanpa perlu menyimpan data peta atau memori lingkungan. Oleh karena itu efisiensi komputasinya, LSRB (*Left-Straight-Right-Back*) sangat cocok diterapkan pada sistem berbasis mikrokontroler sederhana seperti Arduino [4]. Namun demikian, penerapan algoritma tanpa sistem penyimpanan seperti LSRB (*Left-Straight-Right-Back*) juga memiliki keterbatasan. Ketiadaan proses perekaman jalur menyebabkan robot mengulangi eksplorasi berdasarkan urutan prioritas yang sama setiap kali menjumpai simpangan. Akibatnya, robot berpotensi menempuh jalur yang lebih panjang atau kurang efisien, terutama pada struktur *maze* yang kompleks. Meskipun demikian, algoritma ini tetap menarik untuk dikaji dalam konteks efisiensi desain sistem dan keterbatasan perangkat keras.

Penelitian mengenai pengembangan robot *line follower* telah dilakukan oleh berbagai peneliti dengan pendekatan dan tujuan yang beragam. Seperti penelitian dari [5] yang mengembangkan robot *line follower* berkaki dengan sistem navigasi berbasis pemrosesan citra. Menggunakan kamera webcam dan platform komputasi Odroid-XU4, sistem ini mendeteksi titik tengah garis sebagai acuan navigasi yang kemudian dikendalikan oleh algoritma PID (*Proportional-Integral-Derivative*) melalui kontroler *OpenCM* 9.04. Parameter PID (*Proportional-Integral-Derivative*) diperoleh secara trial and error (Kp = 5, Kd = 12, Ki = 0,02) dan terbukti mampu menjaga kestabilan pergerakan robot secara efektif.

Sementara itu [6] mengusulkan algoritma perencanaan jalur berbasis *Particle Swarm Optimization* (PSO) yang memungkinkan robot menentukan rute tercepat dan menghindari rintangan secara dinamis. Meskipun mampu meningkatkan akurasi navigasi, pendekatan ini

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

menuntut sumber daya komputasi tinggi karena memerlukan penyimpanan data partikel dan lingkungan.

Pendekatan berbeda ditunjukkan oleh [7] yang menerapkan sistem kendali *fuzzy* berbasis pengolahan citra digital pada robot e-puck dalam simulasi Webots. Hasilnya, robot mampu menyelesaikan lintasan dengan galat rata-rata yang sangat kecil, menandakan akurasi sistem *fuzzy* yang tinggi dalam pengendalian navigasi.

Adapun penelitian dari [8] mengembangkan sistem examination lamp mobile berbasis mikrokontroler Arduino Uno untuk meningkatkan mobilitas dan efisiensi dalam kegiatan pemeriksaan medis. Sistem ini menggunakan sensor inframerah TCRT 5000 sebagai pendeteksi garis pada mekanisme *line follower*, serta sensor proximity E18-D80NK untuk mendeteksi objek di sekitar area kerja. Modul RF 433 MHz digunakan sebagai sistem kendali nirkabel untuk pergerakan maju dan mundur, sehingga tidak memerlukan sambungan kabel secara langsung. Berdasarkan hasil pengujian, sistem mampu menghasilkan intensitas pencahayaan sebesar 8.460 lux pada jarak 100 cm dan 9.310 lux pada jarak 70 cm, sehingga mendukung kebutuhan pencahayaan yang memadai dalam proses pemeriksaan.

Penelitian lain dari [9] merancang robot *line follower* berbasis mikrokontroler ATmega32A sebagai media pembelajaran dasar di lingkungan Universitas Islam Sultan Agung, mengingat keterbatasan akses terhadap teknologi robotika di kawasan timur Indonesia. Proses pengembangan dilakukan melalui tahapan analisis kebutuhan, perancangan mekanik dan elektronik, hingga proses pengujian. Hasil yang diperoleh menunjukkan bahwa robot mampu mengikuti garis hitam pada permukaan lantai putih dan menampilkan informasi melalui layar LCD. Namun, performa robot masih dipengaruhi oleh kecepatan gerak, di mana robot hanya mampu melacak garis secara stabil pada rentang kecepatan 90–150 RPM; di atas kecepatan tersebut, kemampuan pelacakan mengalami penurunan.

Penelitian sebelumnya yang di lakukan oleh [10] merancang sistem kontrol posisi untuk robot *line follower* menggunakan Arduino Mega 2560 sebagai pengendali utama. Sistem ini dilengkapi dengan sensor garis untuk deteksi lintasan hitam, sensor ultrasonik untuk mendeteksi jarak 5 cm dari objek, serta sensor warna untuk mengenali kotak berwarna merah, hijau, dan biru. Robot dirancang untuk mengikuti garis hitam secara presisi dan mengambil kotak berwarna dari tiga ruangan yang berbeda, lalu memindahkannya ke lokasi tujuan. Kontrol PID digunakan untuk menjaga posisi robot tetap berada di tengah lintasan. Hasil tuning parameter PID menunjukkan bahwa nilai Kp = 100 dan Ki = 0,1 memberikan respons kontrol posisi yang paling stabil selama pergerakan menuju target.

Penelitian selanjutnya oleh [11] mengembangkan platform robotik bergerak yang mampu mengikuti target secara real-time menggunakan kombinasi teknologi *Ultra-Wideband* (UWB) dan kamera RGB-D. Sistem ini dirancang tanpa bergantung pada anchor tetap, menjadikan UWB sebagai solusi *real-time localization system* (RTLS) yang fleksibel. Kamera kedalaman digunakan sebagai sensor pelengkap untuk meningkatkan persepsi lingkungan sekaligus mendukung interaksi manusia-robot. Metode kontrol menggunakan gabungan pengendali PD dan regulasi pegas virtual untuk mengikuti gerakan *non-linier* target. Sistem ini ditujukan untuk aplikasi pengangkutan beban, baik di sektor industri, logistik, maupun konstruksi, dengan potensi mengurangi beban kerja manual yang berulang dan berat.

Sementara itu, [12] mengembangkan sistem otomatisasi pemesanan dan pengantaran makanan di restoran menggunakan robot *line follower* berbasis mikrokontroler Arduino. Pemesanan dilakukan melalui LCD 20x4, sementara robot mengikuti garis hitam untuk mengantarkan makanan ke empat meja tujuan. Hasil pengujian menunjukkan bahwa sistem bekerja secara efektif dengan waktu tempuh yang bervariasi sesuai panjang lintasan. Studi ini menunjukkan potensi penerapan robot berbasis jalur dalam meningkatkan efisiensi layanan di sektor restoran.

Berbeda dari penelitian sebelumnya, penelitian ini difokuskan pada penerapan algoritma LSRB *(Left-Straight-Right-Back)* yang bersifat stateless, yaitu tidak menggunakan memori

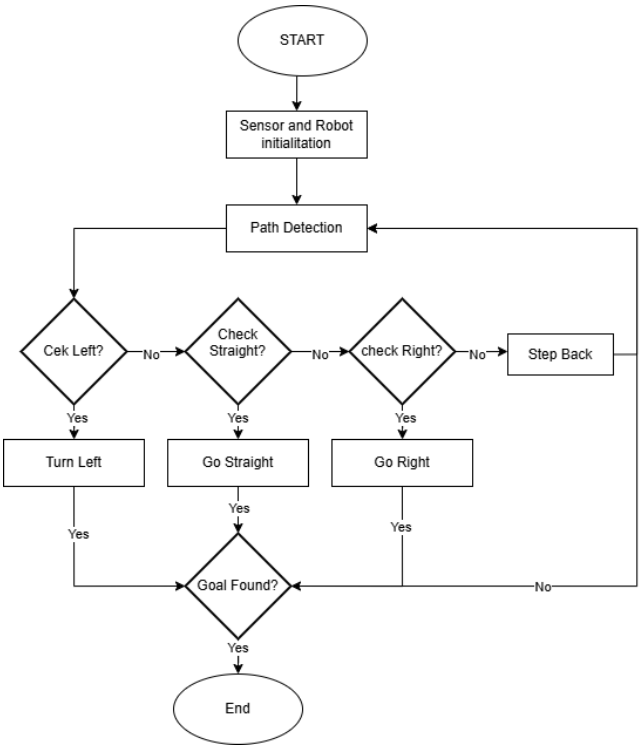
e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

internal untuk menyimpan peta atau posisi yang telah dilalui sebelumnya. Pendekatan ini dirancang untuk mengeksplorasi lintasan bercabang secara eksploratif berdasarkan aturan prioritas arah, dengan tujuan utama untuk mengevaluasi efektivitas algoritma dalam menjelajahi maze tanpa ketergantungan pada penyimpanan data navigasi.

2. METODE PENELITIAN

1.2 Algoritma LSRB (Left Side Rule Based)

LSRB *(Left Side Rule Based)* dalam konteks *line maze solving* adalah metode atau strategi yang digunakan untuk navigasi dan pengambilan keputusan oleh robot dalam menyelesaikan labirin garis [13]. *Line maze solving* sendiri merupakan algoritma yang dirancang untuk memungkinkan robot mencari dan mengikuti jalur dalam suatu labirin garis menuju titik tujuan [14]. Algoritma LSRB mengacu pada empat aturan prioritas arah, yaitu: *Left* (kiri), *Straight* (lurus), *Right* (kanan), dan *Back* (mundur).



Gambar 1. *Flowchart* Algoritma *Line Maze Solving* (LSRB)

Pada gambar 1 algoritma LSRB (*Left–Straight–Right–Back*) memiliki alur kerja yang sistematis dimulai dari tahap *start*, yang dilanjutkan dengan proses inisialisasi sistem robot, termasuk aktivasi sensor. Selanjutnya, robot memasuki tahap *path detection,* yaitu proses mendeteksi keberadaan jalur di sekeliling robot berdasarkan input dari sensor navigasi, seperti sensor inframerah atau kamera. Prinsip dasar dari algoritma ini adalah apabila terdapat kemungkinan untuk berbelok ke kiri, maka robot akan mengambil jalur kiri. Jika tidak memungkinkan, maka robot akan memilih untuk bergerak lurus. Jika tidak tersedia jalur lurus, maka robot akan berbelok ke kanan, dan apabila ketiga pilihan tersebut tidak tersedia, maka robot akan bergerak mundur. Setiap keputusan gerak diambil ketika robot berada pada titik persimpangan, dan proses pemilihan arah selalu mengikuti urutan prioritas yang telah ditentukan oleh aturan LSRB (*Left–Straight–Right–Back)*. Robot akan terus melakukan navigasi berdasarkan urutan prioritas arah sambil mengikuti garis hingga mencapai titik akhir dari labirin *(end of maze*).

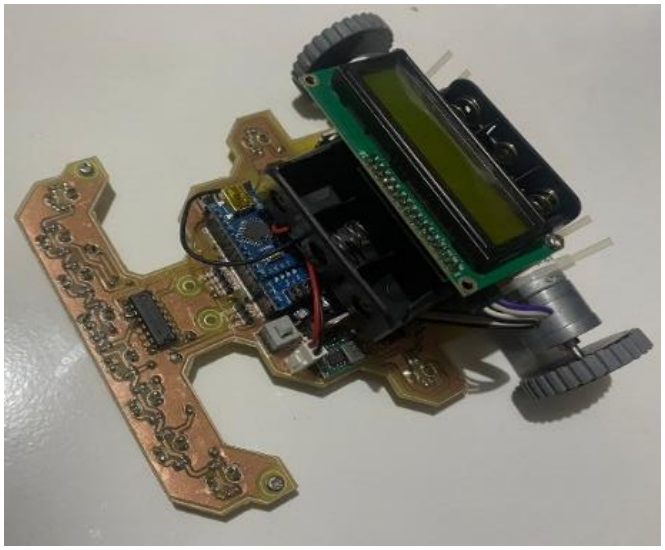
e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

Proses identifikasi jalur selama eksplorasi ini dikenal dengan istilah mapping, meskipun dalam beberapa implementasi, seperti pada algoritma LSRB (*Left–Straight–Right–Back)*, proses ini dapat dilakukan tanpa menyimpan data jalur yang telah dilalui.

Dalam penelitian ini algoritma *Left–Straight–Right–Back* (LSRB) digunakan sebagai inti dari sistem navigasi robot untuk menentukan arah gerak secara *real-time* pada setiap titik persimpangan. Pemilihan algoritma ini didasarkan pada kemampuannya dalam menjalankan navigasi berbasis prioritas arah tanpa memerlukan penyimpanan memori jalur (*stateless*), yang menjadikannya ideal untuk diimplementasikan pada robot dengan keterbatasan perangkat keras seperti mikrokontroler. Dalam konteks penelitian ini, LSRB (*Left–Straight–Right–Back)* dimanfaatkan untuk mengevaluasi sejauh mana robot mampu mencapai titik tujuan dari berbagai titik awal pada lintasan bercabang dengan struktur yang kompleks. Melalui penerapan algoritma ini, dilakukan analisis terhadap efektivitas navigasi, konsistensi pengambilan keputusan arah, serta efisiensi waktu tempuh sebagai indikator utama performa robot dalam menyelesaikan *maze* tanpa bantuan sistem pemetaan eksplisit.

2.2 Robot Line Follower

Robot *line follower* atau *tracer* robot dalam literatur internasional merupakan perangkat mekanik dan elektronik yang di rancang untuk bergerak secara otomatis mengikuti jalur bentuk garis pada permukaan tertentu [15]. Robot *line tracer* atau *line follower* bekerja dengan memanfaatkan sensor pendeteksi garis, seperti sensor inframerah (*infrared*), yang bertugas untuk membaca keberadaan jalur pada permukaan lintasan [16]. Sensor ini mampu membedakan area garis dengan area latar berdasarkan perbedaan pantulan cahaya atau tingkat kecerahan permukaan. Ketika sensor mendeteksi adanya garis, informasi atau data hasil pembacaan tersebut segera di kirimkan ke sistem kontrol yang kemudian mengatur kinerja robot. Sensor inframerah yang berfungsi untuk membaca kontras antar garis dan permukaan sekitarnya pengolahan sinyal dan sensor ini memungkinkan robot untuk menentukan arah geraknya dengan cepat dan akurat, selain di bidang industri, aplikasi robot *line follower* juga dapat di temukan di bidang logistik, pergudangan, serta di sektor pelayanan umum, seperti rumah sakit untuk pengangkutan obat atau makanan [17]. Dengan perkembangan teknologi, robot ini terus mengalami inovasi, termasuk penggunaan algoritma kecerdasan buatan untuk meningkatkan kecepatan, presisi, dan adaptasi terhadap berbagai kondisi permukaan jalur [18].



Gambar 2. Robot *Line Follower*

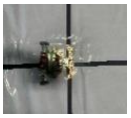
e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

Dalam perancangan robot *line follower* terdapat mikrokontroler ATmega328P berperan sebagai komponen utama yang bertugas mengelola data serta mengendalikan seluruh proses *input* dan *output* pada robot. Sebagai perangka masukan, robot ini dilengkapi dengan sensor *inframerah* sebanyak 8 buah, yang berfungsi untuk mendeteksi garis dengan prinsip kerja *line follower*, sensor ini digunakan mendeteksi garis lintasan berwarna hitam/putih. Tiap sensor terhubung langsung ke jalur PCB dengan sistem pembacaan digital. Sementara itu, perangkat keluaran pada sistem ini meliputi LCD dan Driver motor DC. LCD terhubung ke salah satu output mikrokontroler dan digunakan untuk menampilkan parameter proses yang sedang berlangsung. Sedangkan driver motor DC berfungsi mengontrol pergerakan motor DC kiri dan kanan, yang menjadi penggerak utama robot dalam mengikuti jalur yang telah dipetakan. Hasil perancangan robot *line follower* bisa dilihat pada gambar 4.

Setiap jenis persimpangan pada jalur memiliki pola pembacaan sensor yang berbeda, yang menjadi dasar bagi sistem untuk menentukan arah gerak sesuai dengan urutan prioritas pada algoritma LSRB (*Left–Straight–Right–Back)*. Jalur-jalur tersebut meliputi belokan kiri, belokan kanan, simpang tiga kiri, simpang tiga kanan, perempatan, jalur lurus, jalan buntu, dan titik akhir. Keberagaman bentuk persimpangan ini bertujuan untuk mensimulasikan kondisi nyata dalam struktur labirin, sehingga robot dapat diuji secara menyeluruh dalam berbagai skenario navigasi. Dengan demikian, pengujian ini memungkinkan penilaian yang lebih komprehensif terhadap respon dan konsistensi algoritma LSRB (*Left–Straight–Right–Back).* dalam menghadapi situasi bercabang yang kompleks.

Dalam penelitian ini, robot *line follower* diuji pada beberapa jenis jalur untuk mengevaluasi kemampuan navigasi menggunakan algoritma LSRB (*Left–Straight–Right–Back)*. Setiap jenis jalur mewakili kondisi persimpangan yang berbeda yang umum dijumpai pada lintasan berbasis garis. Adapun klasifikasi jalur yang digunakan dalam pengujian ditampilkan pada Tabel 1.

Tabel 1. Jenis-Jenis Jalur

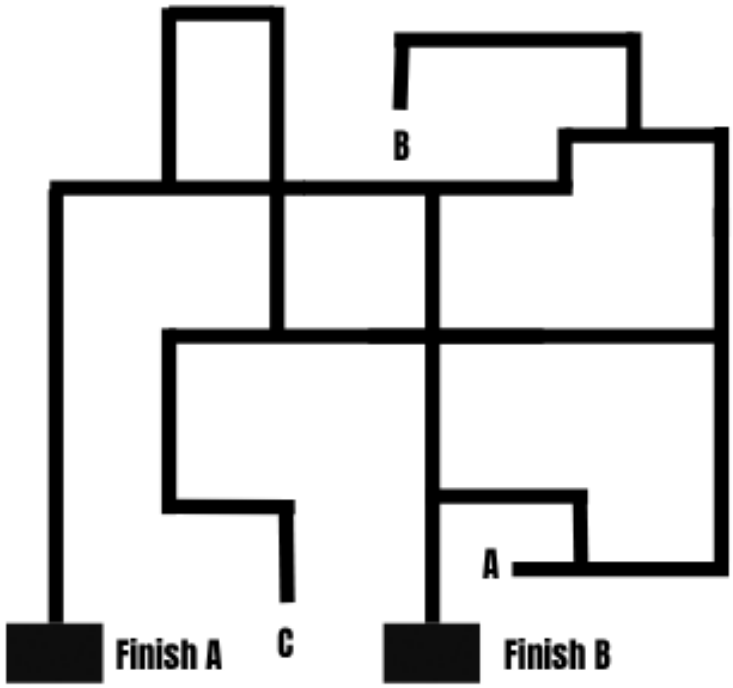
Jalur	Nama Jalur	Kondisi Sensor
	Belok Kiri	
	Belok Kanan	
	Simpang Tiga Kiri	
	Simpang Tiga Kanan	
	Simpang Perempatan	
	Lurus	

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

	Jalan Buntu	
	Akhir/ Finish	

3. HASIL DAN PEMBAHASAN

Pada saat melakukan pengujian pada robot *line follower* berikut jalur yang di gunakan untuk menguji alur jalannya robot *line follower* berikut:



Gambar 3. Rancangan jalur

Pada penelitian ini, jenis *maze* yang digunakan adalah *line maze*, karena sangat cocok untuk pengujian *robot line follower*, di mana robot diarahkan untuk pengujian robot *line follower* saat melintasi berbagai jenis jalur, seperti jalur dengan lebar tetap, lebar yang berubah-ubah, serta jalur yang memiliki percabangan. Selain itu, dilakukan juga perbandingan kinerja dengan robot *line follower* lain yang menggunakan kontrol berbasisi PID (*Proportional-Integral-Derivative*). Komponen utama dalam tahap ini adalah desain lintasan yang digunakan sebagai dasar pengujian performa robot. Terdapat empat tipe lintasan yang digunakan, yaitu lintasan *elips* standar dengan lebar yang sesuai empat titik sensor, lintasan dengan lebar yang bervariasi, lintasan bercabang, serta lintasan lurus yang memiliki perubahan lebar secara bertahap.

Bagian pengujian ini mengcakup evaluasi performa robot *line follower* saat melintasi berbagai jenis jalur, seperti jalur dengan lebar tetap, lebar yang berubah-ubah, serta jalur yang memiliki percabangan. Selain itu, dilakukan juga perbandingn kinerja dengan robot *line follower*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

lain yang menggunakan kontrol berbasis PID (*Proportional-Integral-Derivative*). Komponen utama dalam tahap ini adalah desain lintasan yang digunakan sebagai dasar pengujian performa robot.

Pengujian pada jalur bercabang bertujuan untuk mengevaluasi kemampuan robot *line follower* yang menggunakan algoritma LSRB (*Left–Straight–Right–Back*) dalam menentukan jalur terbaik, khususnya dalam memilih jalur dengan lebar lebih besar di bandingkan jalur sempit. Dalam pengujian ini hanya robot dengan algoritma LSRB (*Left–Straight–Right–Back)* yang digunakan, tanpa pembandingan dari robot dengan kontrol PID (*Proportional-Integral-Derivative*), karena algoritma PID (*Proportional-Integral-Derivative*) tidak dirancang untuk menangani logika pengambilan Keputusan di persimpangan. Uji coba dilakukan sebanyak 10 kali setiap titik awal, dengan lintasan yang mangandung cabang jebakan yakni jalur yang mengarah ke jalan buntu. Robot diharapkan mampu mengidentifikasi dan menghindari jalur jebakan tersebut berdasarkan prinsip pengambilan Keputusan algoritma LSRB (*Left–Straight–Right–Back)*, yang memprioritaskan arah kiri, lurus, dan kanan secara berurutan.

3.1 Pengujian Algoritma LSRB Pada Jalur Robot Line Follower

Dalam pengujian algoritma ini, robot dicoba untuk melakukan tugas yang sebenarnya tujuannya adalah untuk membuktikan algoritma yang di gunakan dapat di terapakan atau tidak pada jalur bercabang berikut merupakan rancangan posisi algoritma LSRB (*Left–Straight–Right–Back)* dapat kita lihat pada gambar 4:

Left **Straight** **Right** **Back**

Gambar 4. Rancangan Posisi Garis Algoritma LSRB

Pada saat robot melakukan proses pencarian jalur dengan cara memberikan kode pada setiap kali robot menjumpai persimpangan jalan buntu, kode yang sudah di terapkan pada robot tersebut akan trus berjalan setiap kali robot menemui persimpangan dan jalan buntu, sampai robot mencapai target. Adapun kode-kode yang digunakan pada saat pencarian jalur yaitu:

a. L berarti *left*. Ini menandakan bahwa robot telah melakukan belok kiri karena melewati persimpangan.
b. berarti *straight* atau jalan terus. Ini menandakan kalau robot melakukan jalan persimpangan tiga dengan pilihan lurus atau belok kanan.
c. R berarti *right*. Ini menandakan bahwa robot telah melakukan belok kanan karena melewati persimpangan.
d. B berarti *back*. Yang di mana ini menandakan bahwa robot bertemu dengan jalan terputus atau jalan buntu makan akan berjalan kembali mencari jalur yang lain.

Dalam algoritma LSRB (*Left-Straight-Right-Back*), pengambilan keputusan arah oleh robot *line follower* saat berada di persimpangan dilakukan berdasarkan urutan prioritas arah, yaitu belok kiri, lurus, belok kanan, dan terakhir mundur. Dengan demikian, implementasi jalur kiri pada algoritma LSRB (*Left–Straight–Right–Back*) memudahkan robot untuk secara otomatis dan sistematis menjelajahi *maze* pada saat penelusuran jalan pada *maze*.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

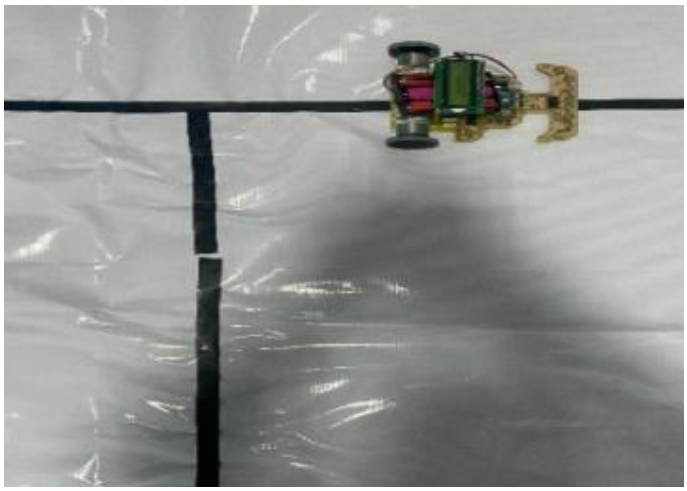
Gambar 5. Robot *Line Follower* Saat Belok Kiri

Ketika robot berada di persimpangan, langkah pertama adalah memeriksa apakah terdapat jalur di sebelah kiri. Jika tersedia, maka robot langsung memilih berbelok ke kiri tanpa mempertimbangkan arah lain, bisa dilihat pada gambar 5.

Gambar 6. Robot *Line Follower* Saat Lurus

Pada Gambar 6 bahwa jika jalur kiri tidak maka robot akan memeriksa jalur bergerak lurus ke depan. Pada robot *line follower* sensor yang di gunakan untuk mendeteksi arah ini adalah dua sensor tengah di bagian depan robot, ketika kedua sensor tersebut mendeteksi ini, robot akan mempertahankan kecepatan secara seimbang untuk terus bergerak maju sesuai dengan jalur yang sudah di tetapkan pada sebuah *maze* dan tetap mengikuti aturan prioritas yang ada pada algoritma LSRB (*Left–Straight–Right–Back)*, dan akan selalu berjalan secara *real time* karena robot tersebut tidak memiliki memori untuk menyimpan hasil penelusuran yang terjadi pada saat melaukan pencarian jalur titik akhir.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849



Gambar 7. Robot *Line Follower* Saat Belok Kanan

Kemudian pada gambar 7, jika jalur ke kiri dan lurus tidak tersedia, maka langkah berikutnya adalah memeriksa sisi kanan. Jalur kanan dipilih hanya jika kedua arah sebelumnya tidak dapat dilalui. Garis yang berada di sisi kanan robot akan mengirim sinyal ketika mendeteksi garis hitam. Jika aktif, maka robot akan melakukan *manuver* berbelok ke kanan, biasanya dengan memperlambat motor sebelah kiri dan mempercepat atau mempertahankan motor sebelah kanan. Keputusan untuk berbelok kanan menandakan bahwa jalur tersebut menjadi satu-satunya alternatif logis yang memungkinkan robot melanjutkan eksplorasi.

Gambar 8. Robot *Line Follower* Saat jalur Buntu

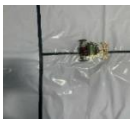
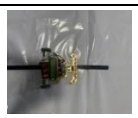
Kemudian pada gambar 8, jika semua jalur (kiri, lurus, kanan) tertutup atau jalan buntu, maka robot tidak memiliki pilihan lain selain melakukan putar balik (*U-turn*). Pada kondisi ini, robot dianggap berada di ujung jalur (*Dead End*). Maka, tindakan yang dilakukan adalah memutar arah 180^0 untuk kembali ke *node* (titik persimpangan) sebelumnya. Sensor garis akan mendeteksi kondisi ini jika semua sensor depan membaca warna terang (putih), dan tidak ada sensor samping yang mendeteksi garis hitam. Mikrokontroler kemudian memerintahkan motor kanan dan kiri untuk berputar kearah berlawanan dalam waktu singkat, hinga robot berbalik arah. Dari jalan buntu sebagai jalur tidak efektif, dan robot akan memutar balik untuk melakukan penelusuran lagi pada jalur yang lain.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

3.2 Pengujian Persimpangan

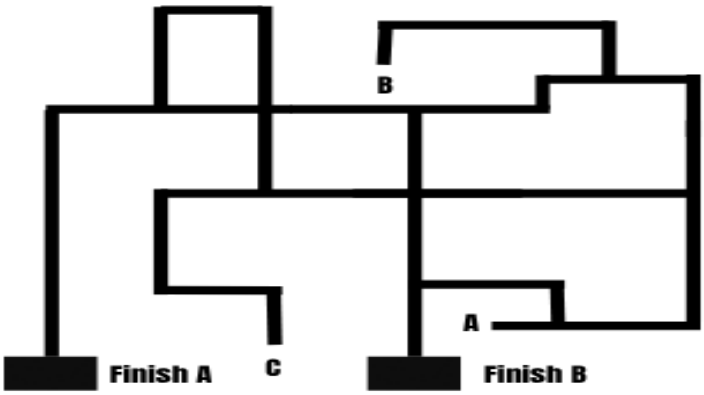
pada pengujian beberapa persimpangan, robot akan melakukan pergerakan terhadap persimpangan berikut dapat di lihat pada tabel di bawah ini:

Tabel 2. pengujian jalur

Bentuk Jalur	Nama	Aksi Robot	Keterangan
	*Lefht*	Belok kiri	Berhasil
	*Stright*	Jalan Lurus	Berhasil
	*Right*	Belok Kanan	Berhasil
	*Back*	Putar Balik	Berhasil

3.3 Hasil Pencarian Jalur

Pengujian ini dirancang dengan tujuan utama untuk mengevaluasi secara menyeluruh kinerja algoritma LSRB (*Left–Straight–Right–Back*) yang diterapkan pada robot *line follower* dalam menjalankan tugas navigasi pada lintasan *maze* berbasis garis, tanpa melibatkan penggunaan sistem penyimpanan jalur atau memori internal untuk merekam lintasan yang telah dilalui. Dalam pengujian ini, robot diarahkan untuk memulai perjalanan dari tiga titik awal yang berbeda, yaitu titik A, titik B, dan titik C, dengan sasaran akhir berupa area berwarna hitam yang menandakan *zona* tujuan atau *goal*. Untuk memastikan kestabilan sistem dan menghindari variabel luar yang dapat memengaruhi hasil pengujian, kondisi operasional robot diatur secara optimal, antara lain dengan memastikan bahwa tegangan baterai berada dalam keadaan penuh serta kecepatan motor disesuaikan pada nilai tetap sebesar 140 PWM. Pengendalian kecepatan dan arah dilakukan menggunakan sistem kendali PID (*Proportional-Integral-Derivative*), yang dikenal efektif dalam menjaga kestabilan dan presisi gerakan robot selama mengikuti garis lintasan.



Gambar 9. Struktur lintasan *maze* dengan tiga titik awal (A, B, dan C) dan dua titik finish yang digunakan dalam pengujian algoritma LSRB pada robot *line follower*

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

Struktur lintasan yang digunakan dalam pengujian ini divisualisasikan melalui Gambar 9, yang menampilkan jalur dengan tingkat kompleksitas, ditandai dengan keberadaan banyak simpangan serta dua titik akhir. Metode navigasi yang diterapkan dalam pengujian ini tidak melibatkan elemen penyimpanan memori, artinya robot tidak merekam atau mengingat rute yang telah dilaluinya pada percobaan sebelumnya. Setiap kali robot memulai dari titik awal, ia harus melakukan eksplorasi jalur secara independen tanpa bantuan informasi historis, dan harus membuat keputusan arah secara *real-time* berdasarkan urutan prioritas keputusan yang ditentukan oleh algoritma LSRB (*Left–Straight–Right–Back)*, yakni mengambil arah belok kiri (*Left*) terlebih dahulu jika tersedia, kemudian lurus (*Straight*), diikuti oleh kanan (*Right*), dan terakhir, mundur (*Back*) jika semua pilihan sebelumnya tidak memungkinkan.

Strategi navigasi yang berbasis pada urutan prioritas ini menunjukkan kinerja yang cukup andal dan konsisten, bahkan ketika dihadapkan pada kondisi variatif seperti perbedaan titik awal maupun konfigurasi simpangan jalur yang kompleks. Hasil dari pengujian ini menunjukkan bahwa meskipun terdapat perbedaan waktu tempuh antar titik awal yang disebabkan oleh panjang jalur dan jumlah simpangan yang harus dilalui algoritma LSRB (*Left–Straight–Right–Back)* tetap mampu mengarahkan robot menuju titik tujuan secara efektif. Dengan kata lain, pendekatan ini terbukti dapat memberikan solusi praktis untuk navigasi *maze* berbasis garis tanpa memerlukan sistem yang kompleks seperti peta digital atau pemrograman penyimpanan jalur, walaupun demikian, pendekatan ini memiliki keterbatasan dalam hal adaptabilitas terhadap perubahan jalur yang dinamis atau lingkungan yang tidak terstruktur.

Pada tabel 3 merupakan tabel hasil percobaan robot saat bergerak dari titk awal A menuju area kotak hitam.

Tabel 3. Percobaan Robot Dari Titik A

Percobaan ke-	Waktu Penelusuran (s)	Karakter Pemetaan
1	4,08	START A – L – L – L – FINISH B
2	6,4	START A – S – L – L – L – S – FINISH B
3	5,27	START A – L – L – L – FINISH B
4	5,42	START A – L – L – L – FINISH B
5	6,04	START A – S – L – L – L – S – FINISH B
6	5,9	START A – L – L – L – FINISH B
7	4,69	START A – L – L – L – FINISH B
8	4,05	START A – L – L – L – FINISH B
9	5,23	START A – L – L – L – FINISH B
10	5,83	START A – L – L – L – FINISH B
Rata-rata	**5,29**	

Berdasarkan tabel 2 diperoleh bahwa dari titik A, robot menunjukkan performa navigasi tercepat dengan waktu rata-rata 5,29 detik. Karakter pemetaan sebagian besar memperlihatkan pola arah L – L – L, yang mengindikasikan bahwa jalur menuju *finish* B bersifat langsung dan efisien. Variasi kecil terjadi pada percobaan ke-2 dan ke-5, di mana robot mengambil arah lurus sebelum kembali ke pola utama, yang menyebabkan waktu penelusuran sedikit lebih lama dibanding percobaan lainnya. Meskipun demikian, robot tetap berhasil mencapai tujuan secara konsisten, menunjukkan stabilitas algoritma LSRB (*Left–Straight–Right–Back)* pada struktur jalur yang sederhana dan minim percabangan.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

Tabel 4. Percobaan Robot Dari Titik B

Percobaan ke-	Waktu Penelusuran (s)	Karakter Pemetaan
1	27,11	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
2	23,26	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
3	26,82	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
4	23,04	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
5	25,68	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
6	27,05	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
7	24,7	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
8	27,22	START B – R – R – L – R – S – R – S – B – S – L – L – L – S – FINISH B
9	26,65	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
10	23,61	START B – R – R – L – R – S – R – S – B – L – L – L - FINISH B
Rata-rata	**25,51**	

Tabel 4 menunjukkan bahwa navigasi dari titik B menghasilkan waktu tempuh rata-rata tertinggi, yaitu 25,51 detik, yang mencerminkan kompleksitas jalur yang dilalui. Karakter pemetaan menunjukkan bahwa robot melewati banyak percabangan dengan dominasi belokan kanan (R), serta beberapa perubahan arah sebelum akhirnya mencapai finish B. Hal ini menunjukkan bahwa struktur jalur dari titik B tidak memberikan banyak pilihan arah kiri, sehingga algoritma LSRB secara bertahap mengarahkan robot ke jalur yang lebih panjang.

Menariknya, pada percobaan ke-8, terjadi penyimpangan jalur yang ditandai dengan tambahan belokan lurus (S) sebelum kembali ke lintasan utama. Meskipun demikian, robot tetap berhasil mencapai tujuan akhir. Hal ini menguatkan bahwa meskipun tidak menggunakan sistem memori, algoritma LSRB (*Left–Straight–Right–Back)* masih memungkinkan robot untuk menyelesaikan navigasi dengan adaptasi berbasis urutan prioritas arah, walaupun tidak selalu dalam waktu optimal.

Tabel 5. Percobaan Robot Dari Titik C

Percobaan ke-	Waktu Penelusuran (s)	Karakter Pemetaan
1	17,84	START C – L – R – R – L – L – S – L – FINISH A
2	14,96	START C – L – R – R – L – L – S – L – FINISH A
3	15,85	START C – L – R – R – L – L – S – L – FINISH A
4	18,03	START C – L – R – R – L – L – S – L – FINISH A
5	20,04	START C – L – R – R – L – L – S – L – FINISH A

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

6	14,64	START C – L – R – R – L – L – S – L – FINISH A
7	16,06	START C – L – R – R – L – L – S – L – FINISH A
8	16,47	START C – L – R – R – L – L – S – L – FINISH A
9	18,8	START C – L – R – R – L – L – S – L – FINISH A
10	16,5	START C – L – R – R – L – L – S – L – FINISH A
Rata-rata	**16,92**	

Dari titik C, robot mencatat waktu rata-rata 16,92 detik, lebih cepat dibandingkan titik B namun masih lebih lambat dari titik A. Jalur yang dilalui memiliki tingkat kompleksitas menengah, ditandai dengan kombinasi beberapa belokan kanan dan kiri sebelum mencapai finish A. Karakter pemetaan menunjukkan adanya perubahan arah yang cukup beragam, sehingga memberikan tantangan tersendiri bagi algoritma LSRB (*Left–Straight–Right–Back)*. Hal ini mencerminkan bahwa lintasan dari titik C memiliki lebih banyak percabangan dibanding titik A, namun masih dapat diselesaikan secara konsisten dengan waktu yang *relatif* stabil.

Tabel 6. Rekap Pengujian Pencarian Jalur

Titik Awal	Rata-rata Waktu Penelusuran (s)	Tujuan
A	5.29	Finish B
B	25.51	Finish B
C	16.92	Finish A

Hasil pengujian menunjukkan bahwa algoritma LSRB (*Left–Straight–Right–Back)* mampu mengarahkan robot mencapai titik tujuan dari ketiga titik awal yang berbeda. Titik A menghasilkan waktu tercepat karena struktur jalurnya lebih langsung dan konsisten. Titik B memiliki jalur paling kompleks dan panjang, menyebabkan waktu tempuh tertinggi. Titik C berada di posisi tengah dalam hal efisiensi. Meskipun tidak menggunakan memori atau peta, algoritma LSRB (*Left–Straight–Right–Back)* tetap memberikan performa yang dapat diandalkan dalam skenario *maze* sederhana hingga menengah.

Penelitian ini menunjukkan bahwa algoritma LSRB (*Left-Straight-Right-Back*) mampu mengarahkan robot *line follower* secara konsisten menuju titik akhir pada lintasan *maze* bercabang tanpa menggunakan penyimpanan jalur atau peta, namun efisiensi waktu tempuh bervariasi tergantung kompleksitas rute dengan titik A paling efisien (5,29 detik), titik C menengah (16,92 detik), dan titik B paling lambat (25,51 detik) sehingga meskipun cocok untuk sistem mikrokontroler sederhana, pengembangan lebih lanjut seperti integrasi pemetaan atau pembelajaran mesin diperlukan untuk meningkatkan efisiensi dan adaptabilitas terhadap struktur jalur yang rumit. Secara umum, algoritma LSRB (*Left–Straight–Right–Back)* mampu bekerja dengan baik tanpa memori, namun memiliki keterbatasan dalam efisiensi waktu jika diterapkan pada struktur jalur yang kompleks. Diskusi juga menyimpulkan bahwa aturan prioritas arah (kiri, lurus, kanan, mundur) memungkinkan penelusuran jalur secara eksploratif, tetapi tidak menjamin pilihan jalur tercepat. Oleh karena itu, meskipun cocok untuk sistem robot berbasis mikrokontroler sederhana, pengembangan lebih lanjut disarankan, seperti integrasi dengan

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

metode pemetaan atau algoritma pembelajaran mesin untuk meningkatkan efisiensi dan adaptabilitas terhadap struktur lintasan yang lebih kompleks.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa algoritma LSRB (*Left–Straight–Right–Back)* mampu diterapkan secara efektif pada robot *line follower* untuk menavigasi lintasan *maze* bercabang tanpa sistem penyimpanan jalur. Dari tiga titik awal yang diuji, titik A memberikan hasil waktu tempuh tercepat karena jalur yang lebih langsung, sedangkan titik B memiliki waktu tempuh terlama akibat jalur yang lebih kompleks. Meskipun tidak menggunakan memori, algoritma LSRB (*Left–Straight–Right–Back*) tetap mampu mengarahkan robot ke titik tujuan secara konsisten. Hal ini membuktikan bahwa pendekatan berbasis aturan prioritas arah seperti LSRB (*Left–Straight–Right–Back)* layak digunakan untuk skenario navigasi dengan kompleksitas rendah hingga menengah. Namun, keterbatasannya dalam efisiensi waktu membuka peluang pengembangan lebih lanjut melalui integrasi dengan metode pemetaan atau algoritma berbasis kecerdasan buatan untuk meningkatkan performa dan adaptasi terhadap lingkungan yang lebih kompleks.

REFERENSI

[1] A. S. Priambodo, A. Nasuha, and O. A. Dhewa, "Integrated Implementation of Computer vision and PID Control for Dynamic Speed Control of Line follower Robot," *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 12, no. 1, pp. 81–93, Jul. 2024, doi: 10.34010/telekontran.v12i1.13323.

[2] F. Yanto and I. Welly, "Analisa dan Perbaikan Algoritma Line Maze Solving Untuk Jalur Loop, Lancip, dan Lengkung pada Robot Line Follower (LFR)," *Jurnal CoreIT*, vol. Vol.1, No.2, 2020.

[3] I. Pramana and A. D. Futra, "Implementasi Algoritma Q Learning Pada," *JOURNAL OF APPLIED ELECTRICAL ENGINEERING*, Dec. 2021.

[4] R. Pranata, R. Tri Wahyuni, P. Studi Teknik Elektronika, and P. Caltex Riau Riau, "Robot Line Maze Pencari Jalur Tercepat," 2018.

[5] Y. H. Fajar, D. Syauqy, and R. Maulana, "Implementasi Maze Mapping pada Robot Line Follower untuk menentukan Shortest Path," 2019. [Online]. Available: http://j-ptiik.ub.ac.id

[6] R. Mardiyanto, A. Suhartono, and R. F. Siregar, "Development of path planning of line follower robot with obstacles avoidance based on particle swarm optimization," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/732/1/012098.

[7] F. M. T. Huda, F. S. Rahmat, D. C. W. Nugroho, M. P. Nurlita, Z. Nugraha, and A. S. Priambodo, "Implementasi Fuzzy Logic Controller untuk Kendali Robot Line Follower berbasis Computer Vision," *Aviation Electronics, Information Technology, Telecommunications, Electricals, and Controls (AVITEC)*, vol. 6, no. 2, p. 147, Aug. 2024, doi: 10.28989/avitec.v6i2.2304.

[8] A. Haris Kuspranoto and M. Fa'iz Alfatih, "Rancang bangun robot line follower pada examination lamp berbasis mikrokontroler arduino design and development of line follower robot using arduino microcontroller based examination lamp," 2023.

[9] A. Latif, H. A. Widodo, R. Rahim, and K. Kunal, "Implementation of line follower robot based microcontroller atmega32a," *Journal of Robotics and Control (JRC)*, vol. 1, no. 3, pp. 70–74, May 2020, doi: 10.18196/jrc.1316.

[10] I. Riyanto, L. Margatama, R. Rizkia, and E. Marantika, "Robot Forklift Line Follower dengan Kendali PID dan Sensor Warna," 2021. [Online]. Available: http://journal.univpancasila.ac.id/index.php/joule/

[11] P. Janousek, Z. Slanina, and W. Walendziuk, "Target-following Robotic Platform Based on UWB Localization and Depth Camera," in *IFAC-PapersOnLine*, Elsevier B.V., Jun. 2024, pp. 247–252. doi: 10.1016/j.ifacol.2024.07.404.

[12] S. Anam, M. Iqbal, and A. Rozaq, "Prototipe robot pengantar pesanan otomatis berbasis arduino," 2022.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849

[13] S. Sakib, A. Chowdhury, S. T. Ahamed, and S. I. Hasan, "Maze solving algorithm for line following robot and derivation of linear path distance from nonlinear path," in *2013 16th International Conference on Computer and Information Technology, ICCIT 2013*, Institute of Electrical and Electronics Engineers Inc., Dec. 2014, pp. 478–483. doi: 10.1109/ICCITechn.2014.6997314.

[14] T. Phutane, "A Stateflow based Approach for Simulation of Line Following Maze Solver Robot," *Int J Res Appl Sci Eng Technol*, vol. 9, no. 4, pp. 1167–1180, Apr. 2021, doi: 10.22214/ijraset.2021.33883.

[15] Sudimanto and Kevin, "Perancangan robot pemindah barang berbasis line follower," 2020.

[16] M. Arya Nugraha, D. Syauqy, R. Regasari, and M. Putri, "Perancangan dan Implementasi Robot Line Follower Menggunakan Avoid Obstacle dengan Metode Wall Following," 2017. [Online]. Available: http://j-ptiik.ub.ac.id

[17] A. Adella, M. Kamal, and A. Finawan, "RANCANG BANGUN ROBOT MOBILE LINE FOLLOWER PEMINDAH MINUMAN KALENG BERBASIS ARDUINO," *JURNAL TEKTRO*, vol. 2, no. 2, 2018.

[18] D. A. N. Janis, D. Pang, and J. O. Wuwung, "Rancang Bangun Robot Pengantar Makanan Line follower," 2014.

e-ISSN : 2356-2579 | p-ISSN : 1412-2693, DOI :10.62411/tc.v24i3.12849