

# Implementasi *Path Planning A\** Untuk Menghindari Robot lawan Menggunakan *Robot Operating System* pada KRSBI-Beroda

Muhammad Ainul Yaqin<sup>1</sup>, Agus Khumaidi<sup>2</sup>, Ryan Yudha Adhitya<sup>3</sup>, Noorman Rinanto<sup>4</sup>, Evi Nafiatus Sholikhah<sup>5</sup>, Riko Satrya Fajar Jaelani Putra<sup>6</sup>

e-mail: [muhammadainul21@student.ppns.ac.id](mailto:muhammadainul21@student.ppns.ac.id)<sup>1</sup>, [aguskhumaidi@ppns.ac.id](mailto:aguskhumaidi@ppns.ac.id)<sup>2</sup>, [ryanyudhaadhitya@ppns.ac.id](mailto:ryanyudhaadhitya@ppns.ac.id)<sup>3</sup>, [noorman.rinanto@ppns.ac.id](mailto:noorman.rinanto@ppns.ac.id)<sup>4</sup>, [evinafiatus@ppns.ac.id](mailto:evinafiatus@ppns.ac.id)<sup>5</sup>, [riko.satriya@ppns.ac.id](mailto:riko.satriya@ppns.ac.id)<sup>6</sup>

<sup>1,2,3,4,6</sup>Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi, Politeknik Perkapalan Negeri Surabaya

<sup>5</sup>Program Studi Teknik Keselamatan dan Kesehatan Kerja, Teknik Permesinan Kapal Kapal, Politeknik Perkapalan Negeri Surabaya

## Informasi Artikel

### Riwayat Artikel

Diterima 16 Mei 2025

Direvisi 30 Mei 2025

Diterbitkan 31 Mei 2025

### Kata kunci:

Robot Sepak Bola  
Algoritma A\*  
Path Planning

### Keywords:

Soccer Robot  
A\* Algorithm  
Path Planning

## ABSTRAK

Robot berperan sebagai perangkat bantu dalam berbagai bidang, termasuk dalam ajang Kontes Robot Sepak Bola Indonesia Beroda (KRSBI-B). Peran robot penyerang sangat krusial karena harus mampu menentukan pergerakan optimal, terutama dalam menghindari lawan yang posisinya dinamis. Permasalahan utama yang dihadapi adalah perubahan posisi bola, robot, dan lawan secara real-time, serta risiko keluar dari batas lapangan. Untuk mengatasi tantangan tersebut, diperlukan sistem navigasi adaptif yang mampu merespons kondisi lapangan secara cepat dan akurat. Penelitian ini menerapkan algoritma A\* sebagai strategi pergerakan robot penyerang dengan pendekatan heuristik berbasis jarak Euclidean. Implementasi dilakukan menggunakan Robot Operating System (ROS) untuk integrasi sistem secara modular. Hasil pengujian menunjukkan bahwa penerapan algoritma A\* mampu meningkatkan tingkat keberhasilan navigasi dalam menghindari lawan dari hanya 10% menjadi 100%. Selain itu, meskipun rata-rata waktu tempuh sedikit meningkat dari 23,01 detik menjadi 27,38 detik, jalur yang ditempuh menjadi lebih adaptif dan aman. Dengan demikian, algoritma A\* terbukti secara efektif meningkatkan efisiensi dan keandalan navigasi robot penyerang dalam lingkungan kompetitif yang dinamis seperti KRSBI-B.

## ABSTRACT

*Robots play a crucial role in various fields, including the Indonesian Wheeled Soccer Robot Contest (KRSBI-B). The striker robot must determine optimal paths while avoiding dynamically positioned opponents. A key challenge is the real-time change in the positions of the ball, robot, and opponents, which increases the risk of collisions or moving out of bounds. This study applies the A\* algorithm with a Euclidean distance heuristic as a path planning strategy for the striker robot. The system is implemented using the Robot Operating System (ROS) for modular integration. Experimental results show that applying A\* significantly improves navigation success, increasing the success rate from 10% to 100%. Although the average travel time increased slightly from 23.01 to 27.38 seconds, the generated paths were more adaptive and safer. These results demonstrate that the A\* algorithm effectively enhances the reliability and efficiency of robot navigation in dynamic and competitive environments like KRSBI-B.*



**Penulis Korespondensi:**

Muhammad Ainul Yaqin

Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi,

Politeknik Perkapalan Negeri Surabaya,

Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia, Kode Pos 60111.

Email: [muhammadainul21@student.ppns.ac.id](mailto:muhammadainul21@student.ppns.ac.id)

Nomer HP/WA aktif: +62 851-5541-0228

## 1. PENDAHULUAN

Robot merupakan perangkat teknologi yang dirancang untuk membantu manusia menyelesaikan tugas-tugas tertentu secara otomatis melalui sistem pemrograman yang tertanam di dalamnya. Penggabungan elemen mekanik, elektronik, dan perangkat lunak memungkinkan robot bekerja secara presisi, efisien, dan aman. Teknologi ini banyak dimanfaatkan dalam berbagai bidang, termasuk kegiatan berisiko tinggi atau pekerjaan berulang yang memerlukan ketelitian tinggi. Di Indonesia, perkembangan teknologi robotik difasilitasi oleh ajang tahunan Kontes Robot Indonesia (KRI), yang menjadi sarana mahasiswa dalam mengembangkan kreativitas dan inovasi. Ajang ini bertujuan untuk meningkatkan penguasaan teknologi dan menyiapkan generasi unggul dalam menghadapi tantangan Revolusi Industri 4.0 [1].

Salah satu divisi dalam KRI adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI-B), yang mengadopsi format kompetisi dari *RoboCup Middle Size League* (MSL). Dalam KRSBI-B, dua tim robot bertanding untuk mencetak gol sebanyak mungkin, meniru permainan sepak bola sungguhan [1]. Pada tahun 2024, kompetisi ini dibagi menjadi dua tahap: tingkat wilayah yang dilakukan secara daring, dan tingkat nasional yang dilakukan secara luring. Pada tingkat wilayah, pertandingan dilakukan dengan dua robot penyerang per tim yang harus bekerja sama selama tiga menit untuk mencetak gol. Sedangkan pada tingkat nasional, pertandingan dilakukan dengan formasi dua robot penyerang dan satu kiper, berhadapan dengan tim dari perguruan tinggi lain dalam format permainan sepak bola penuh [1]. Peran robot penyerang dalam pertandingan langsung sangat vital karena berhadapan langsung dengan robot lawan. Robot penyerang dituntut memiliki kemampuan manuver yang adaptif dan cerdas, termasuk kemampuan untuk menghindari lawan dan memilih posisi terbaik untuk menjalankan strategi tim [2]. Dalam praktiknya, strategi pergerakan robot di lapangan tidak mudah diprediksi karena tergantung pada banyak faktor, seperti posisi robot, posisi lawan, dan arah gerakan bola. Salah langkah dapat menyebabkan robot keluar lapangan atau berhadapan langsung dengan lawan secara tidak menguntungkan [3]. Oleh karena itu, sistem navigasi robot penyerang harus mampu menyesuaikan diri secara real time dengan kondisi di lapangan .

Untuk mengatasi tantangan tersebut, penelitian ini mengusulkan penggunaan sistem berbasis algoritma *path planning A\** yang dirancang untuk menentukan jalur optimal dalam kondisi dinamis. Sistem ini dikombinasikan dengan metode deteksi objek *You Only Look Once* (YOLO) versi 5, yang berfungsi untuk mendeteksi posisi objek seperti bola, lawan, dan kawan di lapangan. Komunikasi antar komponen dalam sistem ini dilakukan melalui *Robot Operating System* (ROS) yang memanfaatkan struktur node terpusat. Dengan pendekatan ini, diharapkan robot penyerang dapat mengambil keputusan pergerakan yang lebih efisien, khususnya dalam menghindari lawan dan mempertahankan posisi strategis selama pertandingan berlangsung.

## 2. METODE PENELITIAN

### 2.1 *Path Planning A\**

Algoritma *A\** termasuk salah satu metode dalam perencanaan jalur yang dikembangkan dari algoritma *Dijkstra* dan dianggap sebagai versi lanjutan dari algoritma *Breadth-First Search* (BFS) [4]. Algoritma ini dikenal memiliki efisiensi yang lebih tinggi dalam menemukan jalur optimal dibandingkan pendahulunya. Untuk meningkatkan performanya, *A\** menerapkan pendekatan *heuristik*, yaitu metode pencarian cerdas yang menggabungkan perhitungan biaya aktual dan estimasi biaya menuju tujuan [5]. Pendekatan ini melibatkan fungsi

p-ISSN: 2356-0533; e-ISSN: 2355-9195



evaluasi yang menghitung total biaya dari simpul awal ke simpul saat ini serta estimasi biaya dari simpul tersebut ke simpul tujuan. Selama proses pencarian, algoritma A\* akan mengevaluasi semua simpul di sekitar simpul aktif dan memilih jalur dengan nilai minimum dari hasil perhitungan tersebut [6]. Dengan strategi ini, A\* mampu menghasilkan rute terpendek yang efisien dan dapat dilalui dengan aman.

Fungsi evaluasi utama yang digunakan dalam algoritma ini dirumuskan sebagai:

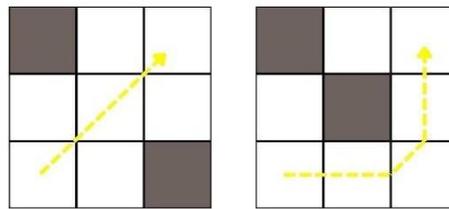
$$f(n) = g(n) + h(n) \tag{1}$$

di mana  $f(n)$  merupakan total nilai evaluasi pada simpul  $n$ ,  $g(n)$  adalah biaya aktual dari titik awal ke simpul  $n$ , dan  $h(n)$  adalah estimasi biaya dari simpul  $n$  ke tujuan [4].

Salah satu metode umum untuk menghitung nilai heuristik  $h(n)$  adalah dengan menggunakan *Euclidean Distance*, karena cocok untuk menghitung estimasi jarak dalam arah diagonal. Rumusnya dituliskan sebagai berikut:

$$h(n) = \sqrt{(x_n - x_t)^2 + (y_n - y_t)^2} \tag{2}$$

Rumus ini memperhitungkan jarak lurus antara dua titik koordinat dalam bidang dua dimensi, yaitu antara posisi saat ini  $(x_n, y_n)$  dan posisi tujuan  $(x_t, y_t)$  [7].



Gambar 1. Simulasi path planning A\*

## 2.2 Robot Operating System (ROS)

*Robot Operating System* (ROS) merupakan sebuah platform perangkat lunak berbasis sistem operasi Linux yang secara khusus dikembangkan untuk mendukung pengembangan robot bergerak (*mobile robot*). ROS menyediakan lingkungan kerja yang fleksibel bagi para pengembang dalam membangun, mengelola, dan mengintegrasikan berbagai sistem robotik yang kompleks [8]. Struktur ROS terdiri dari unit-unit kecil yang disebut *node*, dimana masing-masing *node* bertanggung jawab menjalankan fungsi tertentu secara terpisah. Interaksi dan pertukaran data antar-*node* dilakukan melalui perantara yang disebut *ROS Master*, yang berfungsi sebagai pengatur lalu lintas komunikasi dan penghubung antar *node*. Dengan arsitektur ini, ROS memungkinkan integrasi sistem robotik yang modular dan *scalable*, serta mendukung kolaborasi antar komponen robot yang saling berinteraksi [9].



Gambar 2. Robot Operating System (ROS)

*Robot Operating System* (ROS) juga dilengkapi dengan fitur visualisasi simulasi yang memungkinkan pengguna untuk memantau dan memvalidasi perilaku sistem robotik secara virtual sebelum diterapkan langsung pada perangkat fisik. Fitur ini memungkinkan pengguna untuk memvisualisasikan lingkungan, posisi robot, dan elemen-elemen lain yang terlibat dalam sistem [8]. Dalam penelitian ini, digunakan versi ROS "Noetic", yaitu salah satu distribusi ROS yang memiliki status *Long-Term Support* (LTS) [10]. Versi ini dipilih karena menyediakan dukungan jangka panjang dan telah dilengkapi dengan berbagai pustaka (*library*) serta alat bantu pengembangan

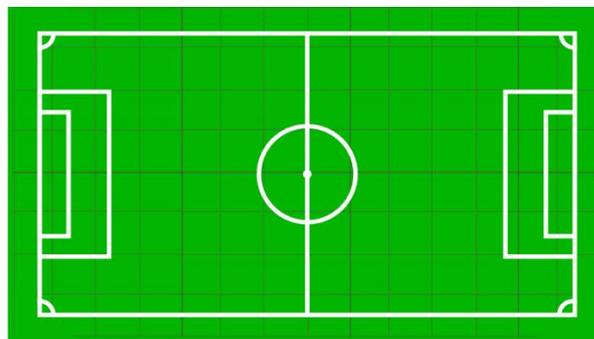


yang dibangun oleh komunitas ROS secara luas. Keberadaan dukungan komunitas dan ketersediaan perangkat lunak yang melimpah menjadikan ROS Noetic sebagai pilihan tepat untuk mempermudah dan mempercepat proses pengembangan sistem robotik secara efisien [9].

### 2.3 Pembagian *Grid* Lapangan

Dalam penerapan algoritma *path planning*  $A^*$ , representasi lingkungan dalam bentuk *grid* diperlukan untuk mempermudah perhitungan jalur yang optimal [11]. Pada penelitian ini, lapangan robot memiliki ukuran 8 meter pada sumbu X dan 12 meter pada sumbu Y, sehingga dibagi ke dalam *grid* berukuran 1 meter  $\times$  1 meter. Dengan pembagian ini, diperoleh total 8 kolom (sumbu X) dan 12 baris (sumbu Y), menghasilkan 96 sel *grid* secara keseluruhan. Setiap *grid* merepresentasikan satu *node* dalam pemetaan algoritma  $A^*$ , yang dapat berupa titik yang bisa dilalui (bebas) atau tidak bisa dilalui (terhalang) [12].

Setiap *node* pada *grid* akan dievaluasi menggunakan fungsi *heuristic* untuk mencari jalur dari titik awal menuju titik tujuan. Fungsi evaluasi pada algoritma  $A^*$  menggunakan rumus persamaan 1 adalah biaya aktual dari titik awal ke *node*  $n$ , dan  $h(n)$  adalah estimasi biaya ke titik tujuan, yang dihitung menggunakan metode *Euclidean Distance*. Koordinat lapangan dimodelkan dalam sistem kartesian dua dimensi dengan titik asal (0,0) berada di pojok kiri bawah, menjadikan arah pergerakan robot lebih terstruktur dan dapat diatur baik secara ortogonal maupun diagonal [13].



Gambar 3. Pembagian *grid* lapangan

Pembagian *grid* ini memudahkan integrasi data dengan sistem deteksi objek seperti YOLOv5, di mana posisi objek (seperti robot lawan) dapat dipetakan ke dalam *node* tertentu dan dijadikan sebagai *obstacle* pada proses pencarian jalur. Implementasi *grid* juga digunakan sebagai masukan dalam *node* ROS agar pergerakan robot penyerang dapat dihitung secara dinamis dalam lingkungan simulasi yang mencerminkan kondisi lapangan pertandingan secara realistis.

### 2.4 Desain Robot

Pada tahap perancangan mekanik, acuan yang digunakan adalah peraturan Kontes Robot Sepak Bola Indonesia Beroda tahun 2024 yang diterbitkan oleh Puspresnas. Berdasarkan ketentuan tersebut, dimensi maksimum robot adalah sebagai berikut:

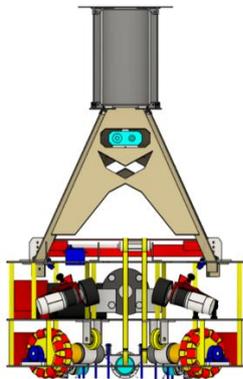
- Panjang: 520 mm
- Lebar: 520 mm
- Tinggi: 800 mm

Dengan mengacu pada spesifikasi tersebut, perancangan sistem mekanik dilakukan menggunakan perangkat lunak Fusion 360, yang digunakan untuk merancang seluruh struktur mekanik, mulai dari kerangka utama, sistem penggerak, pengiring bola, penendang, hingga struktur pendukung lainnya. Tahapan ini menjadi langkah

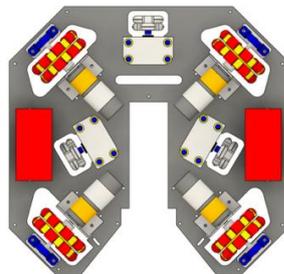
p-ISSN: 2356-0533; e-ISSN: 2355-9195



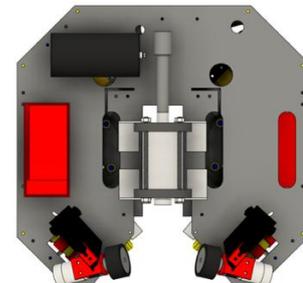
awal yang sangat penting dalam proses pembuatan robot, karena desain mekanik yang baik akan menentukan kemudahan perakitan, kekuatan struktur, serta ketahanan terhadap benturan, terutama saat terjadi kontak langsung dengan robot lawan di lapangan [14]. Desain disusun secara bertingkat, yang masing-masing divisualisasikan pada gambar sebagai berikut:



Gambar 4. Tampak depan



Gambar 5. Tampak lantai 1

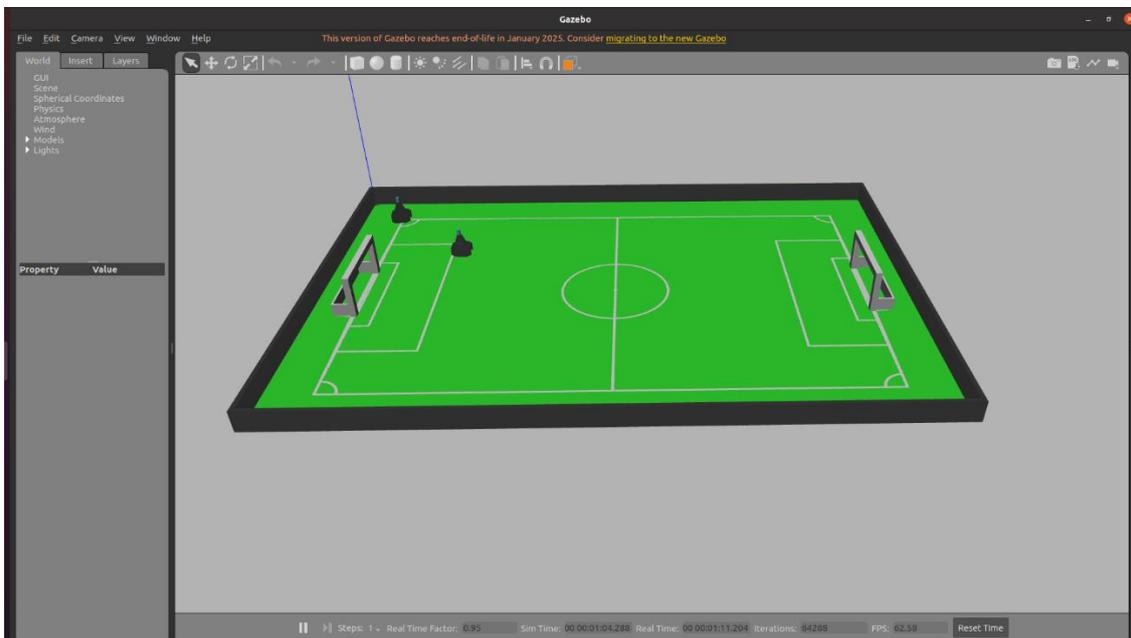


Gambar 6. Tampak lantai 2

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Simulasi gazebo

Untuk menguji hasil perencanaan jalur menggunakan algoritma A\*, dilakukan simulasi pada lingkungan *virtual* menggunakan Gazebo. Lingkungan ini dirancang menyerupai kondisi lapangan sebenarnya pada KRSBI-B, dengan model robot dan gawang yang telah disesuaikan skala dan ukurannya. Tujuan dari simulasi ini adalah untuk mengamati perilaku robot saat mengikuti jalur yang telah direncanakan serta memastikan sistem dapat bekerja secara optimal sebelum diterapkan secara fisik.



Gambar 7. Visualisasi lapangan berukuran 8m x 12m.

Pada Gambar 7 ditampilkan simulasi lapangan sepak bola beroda dengan ukuran 8 meter pada sumbu X dan 12 meter pada sumbu Y yang diimpor ke Gazebo sebagai file model URDF/SDF [15]. Lapangan memiliki dua

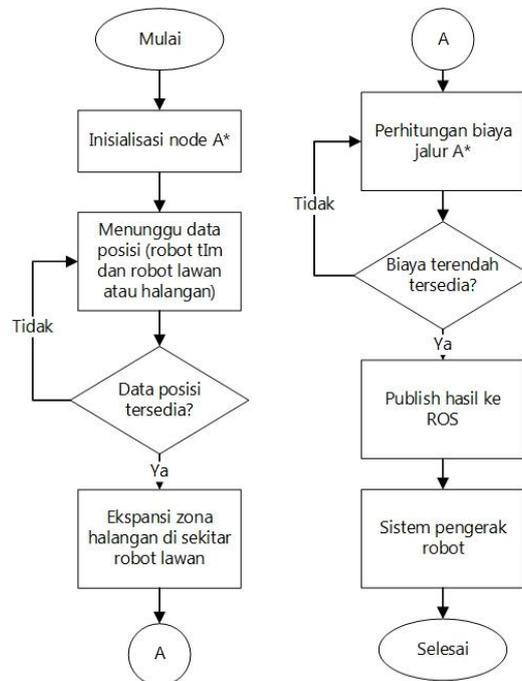


buah gawang yang diletakkan pada sisi kiri dan kanan sebagai area target permainan. Garis-garis putih menunjukkan batas dan zona strategis permainan, termasuk lingkaran tengah dan kotak penalti.

Dua buah robot ditampilkan berwarna hitam dengan penanda biru dan magenta di bagian atasnya, berfungsi sebagai indikator arah dan identitas. Posisi awal robot diatur pada titik tertentu untuk menguji jalur dari posisi tersebut menuju target berdasarkan hasil perhitungan algoritma A\*. Setiap pergerakan robot dikendalikan melalui *node* pada *Robot Operating System* (ROS) yang memanggil *service /gazebo/set\_model\_state* untuk mengubah posisi dan orientasi secara dinamis. Simulasi ini tidak hanya berfungsi sebagai validasi algoritma, tetapi juga untuk mengevaluasi apakah lintasan yang dihasilkan benar-benar dapat diikuti oleh robot dalam kondisi virtual dengan batasan lingkungan seperti dinding lapangan dan area yang tidak dapat dilalui. Dengan pendekatan ini, jalur optimal dan aman dapat diuji secara visual dan kuantitatif.

### 3.2 Flowchart Sistem

Perencanaan jalur (*path planning*) merupakan salah satu aspek penting dalam sistem navigasi robot agar dapat bergerak secara efisien dan menghindari halangan di lingkungannya. Pada penelitian ini, metode algoritma A\* digunakan untuk merancang jalur pergerakan robot dari titik awal menuju titik tujuan dengan mempertimbangkan keberadaan rintangan di lapangan berupa posisi robot lawan. Untuk memberikan pemahaman yang lebih jelas terhadap alur logika sistem, berikut disajikan diagram alir (*flowchart*) dari proses perhitungan jalur algoritma A\* yang telah diimplementasikan pada sistem.



Gambar 8. Flowchart path planning A\*

*Flowchart* pada Gambar 8 menggambarkan tahapan-tahapan utama dalam sistem perencanaan jalur robot. Proses dimulai dari inisialisasi *node* algoritma A\* yang dijalankan dalam ROS. Setelah *node* aktif, sistem menunggu masukan berupa tiga parameter penting, yaitu posisi robot sendiri (tim magenta), posisi robot lawan (sebagai *obstacle*), dan titik tujuan (goal). Ketiga parameter ini diterima secara asinkron melalui topik-topik ROS yang telah disubscribe.



Ketika ketiga parameter posisi telah tersedia, sistem melanjutkan ke tahap ekspansi zona larangan. Zona larangan ini dibentuk di sekitar posisi robot lawan menggunakan pendekatan lingkaran dengan radius tertentu (misalnya 100 cm). Seluruh titik yang berada dalam zona ini dianggap sebagai area yang tidak dapat dilalui oleh robot saat merencanakan jalur. Langkah selanjutnya adalah menjalankan algoritma A\* untuk mencari jalur terbaik. Proses pencarian ini dilakukan dengan menghitung nilai evaluasi dari setiap *node* menggunakan persamaan 1, di mana  $g(n)$  merupakan biaya dari titik awal ke *node* tersebut, dan  $h(n)$  adalah estimasi biaya ke tujuan berdasarkan *heuristik Manhattan Distance*. Algoritma secara iteratif memilih *node* dengan nilai  $f(n)$  paling kecil hingga mencapai titik tujuan. Apabila jalur berhasil ditemukan, hasilnya dikemas ke dalam pesan bertipe *nav\_msgs/Path* dan dipublikasikan ke dalam topik */astar\_result\_path*. Jalur ini selanjutnya digunakan oleh sistem penggerak robot untuk mengikuti lintasan yang telah direncanakan secara dinamis di dalam lingkungan simulasi.

### 3.3 Perhitungan path planning A\*

Perencanaan jalur (*path planning*) merupakan salah satu komponen penting dalam sistem navigasi robot bergerak. Pada penelitian ini, algoritma A\* digunakan untuk menentukan jalur optimal dari posisi awal robot menuju titik tujuan, dengan mempertimbangkan keberadaan robot lawan sebagai rintangan dinamis. Jalur yang dihasilkan harus tidak hanya optimal secara jarak, tetapi juga aman, dalam arti mampu menghindari daerah yang telah ditetapkan sebagai area terlarang (*obstacle area*).

Dalam skenario yang diuji, posisi awal robot penyerang (robot magenta) berada pada koordinat (200, 200) dan titik tujuan berada pada (400, 400). Sementara itu, robot lawan (robot cyan) ditempatkan di titik (300, 300). Titik ini diasumsikan sebagai *obstacle* utama yang harus dihindari. Untuk memperluas area larangan di sekitar robot lawan, digunakan fungsi yang menghasilkan zona hindaran berbentuk lingkaran dengan radius 100 cm. Zona ini mencakup semua titik  $(x, y)$  yang memenuhi:

$$(x - 300)^2 + (y - 300)^2 \leq 100^2$$

Sehingga, semua titik dalam radius tersebut akan dimasukkan ke dalam daftar *node* yang tidak dapat dilalui oleh algoritma A\*. Hal ini dilakukan agar robot tidak hanya menghindari titik pusat lawan, tetapi juga memberi jarak aman untuk menghindari tabrakan langsung.

Algoritma A\* yang digunakan mengandalkan fungsi evaluasi:

$$f(n) = g(n) + h(n)$$

dengan bobot heuristik pada penelitian ini diatur sebagai berikut:

- Bobot  $g(n) = 1$  untuk gerakan vertikal/horizontal, dan  $\sqrt{2} \approx 1.41$  untuk gerakan diagonal.
- $h(n)$  dihitung menggunakan *Manhattan Distance*:

$$h(n) = |x_n - x_g| + |y_n - y_g| \quad (3)$$

- Tidak ada penyesuaian bobot khusus antara  $g$  dan  $h$  ( $\omega = 1$ ), sehingga penilaian jalur murni berbasis pada kombinasi jarak aktual dan estimasi.

Proses pencarian jalur dilakukan dengan mengevaluasi tetangga dari setiap *node* yang sedang aktif (*current node*), menggunakan delapan arah pergerakan (atas, bawah, kiri, kanan, dan diagonal). Jika *node* tetangga termasuk dalam *zona obstacle*, maka *node* tersebut akan diabaikan dalam proses pencarian. Dengan adanya *zona* larangan di sekitar titik (300,300) jalur lurus diagonal dari titik awal ke tujuan menjadi tidak valid karena harus melewati area



yang dilarang. Sebagai akibatnya, algoritma A\* secara otomatis akan memilih jalur alternatif yang menghindari *zona* tersebut. Contoh urutan jalur hasil perhitungan menunjukkan bahwa robot awalnya mengikuti garis diagonal, tetapi kemudian mulai menyimpang untuk menghindari *obstacle*. Berikut adalah contoh beberapa perhitungan *path planning* A\*:

- *Node* 1: Start Node (200, 200)

$$g(200,200) = 0 \text{ (biaya dari start ke start = 0)}$$

$$h(200,200) = |400 - 200| + |400 - 200| = 200 + 200 = 400$$

$$f(200,200) = 0 + 400 = 400$$

- *Node* 2: (201, 201)

$$g(201,201) = 0 + 1.41 = 1.41$$

$$h(201,201) = |400 - 201| + |400 - 201| = 199 + 199 = 398$$

$$f(201,201) = 1.41 + 398 = 399.41$$

- *Node* 3: (202, 202)

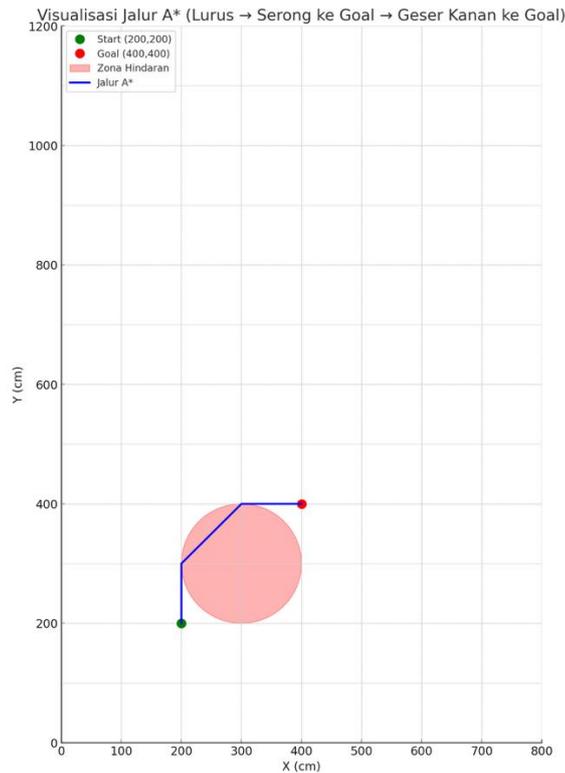
$$g(202,202) = 1.41 + 1.41 = 2.82$$

$$h(202,202) = |400 - 202| + |400 - 202| = 198 + 198 = 396$$

$$f(202,202) = 2.82 + 396 = 398.82$$

Dari perhitungan setiap node akan dipilih node dengan biaya termurah untuk menuju titik tujuan tapi tetap dengan mempertimbangkan area halangan, visualisasi hasil perhitungan sebagai berikut:





Gambar 8. Visualisasi hasil perhitungan biaya path planning A\*

Visualisasi hasil perhitungan algoritma A\* dalam skenario pergerakan robot dari titik awal (200, 200) menuju titik tujuan (400, 400) menunjukkan bahwa robot awalnya bergerak lurus ke depan, kemudian membelok serong ke kanan untuk menghindari zona larangan berbentuk lingkaran dengan radius 100 cm yang mengelilingi posisi robot lawan di titik (300, 300). Setelah keluar dari zona tersebut, robot melanjutkan pergerakan mengikuti garis horizontal menuju titik tujuan. Jalur yang terbentuk mencerminkan bagaimana algoritma A\* dapat merencanakan pergerakan yang tidak hanya efisien secara jarak, tetapi juga aman dari risiko tabrakan dalam lingkungan dua dimensi yang mengandung rintangan dinamis.

Selain itu, perhitungan algoritma A\* menggunakan evaluasi fungsi biaya total  $f(n) = g(n) + h(n)$ , dengan  $g(n)$  sebagai biaya aktual dari titik awal dan  $h(n)$  sebagai estimasi ke tujuan berdasarkan Manhattan Distance. Nilai heuristik tidak diberi bobot tambahan dalam penelitian ini (bobot standar  $\omega = 1$ ), namun dapat disesuaikan untuk kebutuhan lain. Dari pengamatan pada 10 kali pengujian, rata-rata waktu perjalanan fisik robot dengan A\* adalah 27,38 detik, sedangkan estimasi waktu komputasi perencanaan jalur berada dalam kisaran 15–45 milidetik tergantung kompleksitas lingkungan dan jumlah node evaluasi. Hal ini menunjukkan bahwa A\* cukup efisien untuk digunakan dalam sistem real-time berbasis ROS, dengan keunggulan utama dalam menghasilkan lintasan aman dan adaptif terhadap kondisi lapangan yang berubah-ubah.

### 3.3 Percobaan path planning A\*

Pengujian dilakukan untuk membandingkan efektivitas perencanaan jalur robot menggunakan metode *path planning* A\* dengan metode tanpa A\*. Tujuan dari pengujian ini adalah untuk menentukan sejauh mana algoritma A\* dapat meningkatkan akurasi pergerakan robot dalam mencapai tujuan sambil menghindari lawan di lapangan.

Beberapa parameter yang diuji dalam pengujian ini meliputi waktu tempuh, jarak tempuh, dan keberhasilan menghindari lawan. Waktu tempuh dihitung sejak robot mulai bergerak dari titik awal hingga mencapai titik tujuan. Jarak tempuh dihitung berdasarkan total lintasan yang dilalui oleh robot, baik dalam kondisi menggunakan A\*



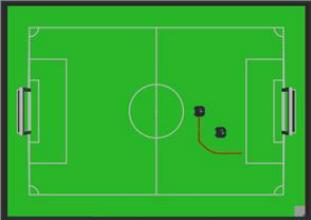
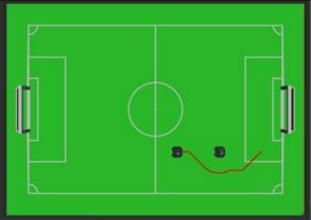
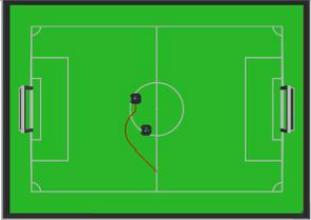
maupun tanpa A\*. Sementara itu, keberhasilan menghindari lawan dievaluasi berdasarkan kemampuan robot mencapai tujuan tanpa menabrak robot lawan yang ditempatkan sebagai obstacle.

Pada pengujian ini, dilakukan dua skenario:

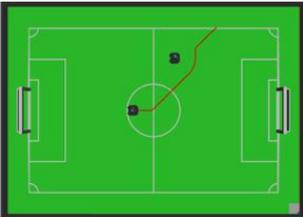
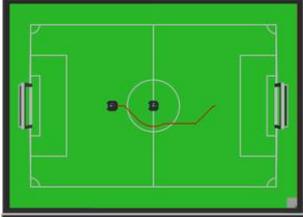
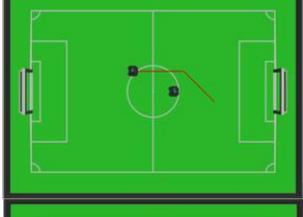
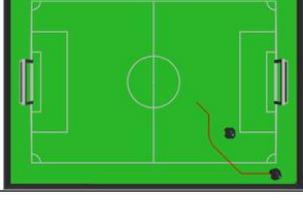
1. Metode A\*: Robot bergerak menggunakan path planning A\* yang secara aktif menghindari zona larangan di sekitar robot lawan.
2. Tanpa Metode A\*: Robot bergerak langsung menuju titik tujuan tanpa memperhitungkan adanya rintangan, sehingga ada kemungkinan bertabrakan dengan robot lawan.

Untuk setiap skenario, dilakukan beberapa kali uji coba dengan variasi posisi robot lawan pada lapangan. Hal ini bertujuan untuk mengetahui konsistensi metode dalam kondisi yang berbeda. Parameter utama yang dicatat adalah waktu tempuh dalam detik, jarak tempuh dalam meter, dan persentase keberhasilan menghindari lawan.

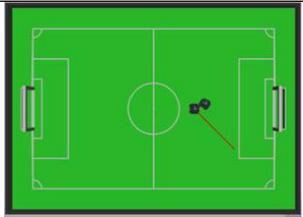
TABEL I : HASIL PENGUJIAN DENGAN MENGGUNAKAN METODE PATH PLANNING A\*

No	Gambar	Titik Awal (X,Y)	Titik Tujuan (X,Y)	Titik halangan (X,Y)	Waktu (s)	Jarak (cm)	Keberhasilan Menghindar
1		300,300	500,00	400,400	20.59	342	Berhasil
2		300,200	300,600	300,400	26.6	400	Berhasil
3		300,400	700,400	500,400	26.03	400	Berhasil
4		200,700	550,800	400,750	24.11	350	Berhasil
5		600,550	400,900	500,700	27.77	442	Berhasil

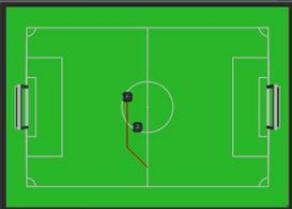
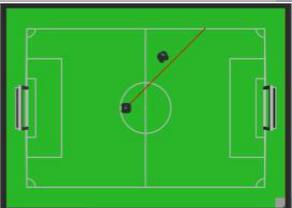
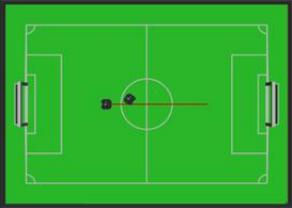


6		900,400	500,800	750,600	31.35	492	Berhasil
7		500,400	500,900	500,700	34.28	500	Berhasil
8		300,700	700,700	480,700	26.91	400	Berhasil
9		450,400	600,800	500,600	25.26	400	Berhasil
10		400,490	50,110	250,330	30.88	512	Berhasil

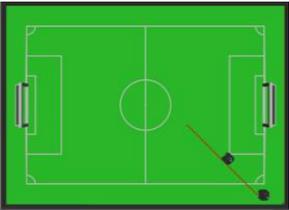
TABEL II : HASIL PENGUJIAN TANPA MENGGUNAKAN METODE PATH PLANNING A\*

No	Gambar	Titik Awal (X,Y)	Titik Tujuan (X,Y)	Titik halangan (X,Y)	Waktu (s)	Jarak (cm)	Keberhasilan Menghindar
1		300,300	500,00	400,400	13.09	200	Tidak



2		300,200	300,600	300,400	22.77	400	Tidak
3		300,400	700,400	500,400	24.49	400	Tidak
4		200,700	550,800	400,750	22.25	350	Tidak
5		600,550	400,900	500,700	22.26	350	Tidak
6		900,400	500,800	750,600	24.11	400	Tidak
7		500,400	500,900	500,700	30.18	500	Tidak
8		300,700	700,700	480,700	23.71	400	Tidak
9		450,400	600,800	500,600	24.88	400	Berhasil



10		400,490	50,110	250,330	24.33	380	Tidak
----	---	---------	--------	---------	-------	-----	-------

Analisis terhadap hasil percobaan navigasi robot menunjukkan bahwa algoritma A\* mampu memberikan performa yang jauh lebih baik dibandingkan metode tanpa perencanaan jalur. Dari sepuluh percobaan yang dilakukan tanpa algoritma A\*, hanya satu percobaan yang berhasil menghindari rintangan, menghasilkan tingkat keberhasilan sebesar 10%. Sebaliknya, ketika menggunakan algoritma A\*, seluruh sepuluh percobaan berhasil, menunjukkan tingkat keberhasilan 100%. Hal ini menandakan bahwa A\* sangat efektif dalam mengarahkan robot untuk menghindari rintangan *dinamis* di lapangan seperti posisi lawan. Meskipun waktu tempuh rata-rata mengalami sedikit peningkatan, dari 23,01 detik (tanpa A\*) menjadi 27,38 detik (dengan A\*), peningkatan ini merupakan kompromi yang wajar demi memperoleh jalur yang aman dan efisien. Selain itu, jarak tempuh sedikit meningkat dari rata-rata 363,0 cm menjadi 413,8 cm, namun hal ini terjadi karena jalur harus menghindari zona larangan berbentuk lingkaran dengan radius 100 cm di sekitar lawan. Visualisasi hasil perhitungan menunjukkan bahwa robot awalnya bergerak lurus dari titik awal ke arah tujuan, namun mulai menyimpang secara cerdas untuk menghindari zona larangan, sebelum akhirnya kembali mengikuti arah menuju tujuan. Dengan demikian, A\* memberikan keseimbangan optimal antara efisiensi navigasi dan keamanan, menjadikannya sangat layak diterapkan dalam sistem robot penyerang yang beroperasi di lingkungan dinamis seperti pada kompetisi KRSBI-B.

#### 4. KESIMPULAN

P Penelitian ini berhasil membuktikan bahwa penerapan algoritma path planning A\* pada robot penyerang KRSBI-Beroda secara signifikan meningkatkan efektivitas sistem navigasi. Dari hasil pengujian sepuluh skenario, metode A\* menunjukkan tingkat keberhasilan 100% dalam menghindari rintangan (robot lawan), dibandingkan hanya 10% keberhasilan pada pendekatan tanpa perencanaan jalur. Selain itu, meskipun jalur yang ditempuh oleh robot dengan A\* sedikit lebih panjang, namun secara rata-rata waktu tempuh hanya meningkat sekitar 4 detik, yang merupakan kompromi wajar demi keamanan dan keberhasilan navigasi.

Penggunaan A\* memungkinkan robot secara adaptif menghindari zona larangan yang diatur dalam bentuk lingkaran di sekitar posisi lawan, menghasilkan jalur yang efisien dan aman. Simulasi pada platform ROS dan Gazebo memperkuat bukti bahwa jalur yang dihitung dapat diikuti dengan baik oleh robot dalam lingkungan dinamis. Implementasi A\* tidak hanya memastikan robot mencapai tujuan dengan aman, tetapi juga meningkatkan responsivitas dan keandalan sistem. Dengan hasil tersebut, integrasi algoritma A\* sangat direkomendasikan sebagai bagian dari strategi navigasi robot penyerang untuk meningkatkan performa dalam kompetisi robot sepak bola beroda seperti KRSBI-B.

#### DAFTAR PUSTAKA

- [1] B. Kusumoputro *et al.*, "Pedoman Kontes Robot Indonesia (Kri) Pendidikan Tinggi Tahun 2024," *Kementeri. Pendidikan, Kebudayaan, Riset, dan Teknol.*, pp. 1–164, 2024.
- [2] W. Darmawan, M. Basuki Rahmat, A. Khumaidi, R. Yudha Adhitya, and D. Pristovani Riananda, "Perancangan Strategi Keputusan Robot Sepak Bola Beroda menggunakan Metode Decision Tree," *J. Elektron. dan Otomasi Ind.*, vol. 10, no. 2, pp. 175–182, 2023, doi: 10.33795/elkolind.v10i2.3020.
- [3] Y. Octavian, H. A. Widodo, and A. Khumaidi, "Optimasi Deteksi Bola Pada Robot Sepak Bola," 2021.
- [4] D. W. Widodo, S. Julian, and I. S. Luluk, "[2] View of Implementasi Algoritma Path Planning A\_ Pada Base Station Robot Sepak Bola Beroda.pdf," 2023.
- [5] A. MUHAMMAD, "A GENERALIZED LASER SIMULATOR ALGORITHM FOR OPTIMAL PATH PLANNING IN CONSTRAINTS ENVIRONMENT



AISHA MUHAMMAD DOCTOR OF PHILOSOPHY”.

- [6] L. I. Safitri, S. Julian, and W. W. Danang, “Implementasi Algoritma Path Planning A\* Pada Base Station Robot Sepak Bola Beroda,” *Gener. J.*, vol. 7, no. 3, pp. 56–63, 2023, doi: 10.29407/gj.v7i3.20545.
- [7] E. Mirs, “Implementation of A\*(Star) Algorithm in Robots Object Movement,” -, vol. XIII, no. 3, pp. 254–260, 2010.
- [8] A. SUCIPTO, R. S. DEWANTO, and D. PRAMADIHANTO, “Gerak Robot Berkaki Dua menggunakan ROS dan RViz sebagai Visualisasi Interaktif,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 1, p. 43, 2021, doi: 10.26760/elkomika.v9i1.43.
- [9] A. Jalil, “PANDUAN LENGKAP ROBOT OPERATING SYSTEM (ROS),” 2023.
- [10] H. Hendri, L. Hoki, V. Agusman, and D. Aryanto, “Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga,” *J. TIMES*, vol. 10, no. 1, pp. 1–5, 2021, doi: 10.51351/jtm.10.1.2021645.
- [11] Y. Damayanti and Y. Akbar, “View of Implementasi Algoritma A\_ (A-Star) untuk Mencari Rute Terpendek dari Kelurahan Cibubur ke Perpustakaan Nasional Republik Indonesia.pdf.”
- [12] S. Rohmadina, “OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO MENGGUNAKAN ALGORITMA A-STAR UNTUK OPTIMASI RUTE WISATA KABUPATEN MOJOKERTO MENGGUNAKAN ALGORITMA A-STAR UNTUK,” 2024.
- [13] D. F. Adryady, A. Prasetyo, H. P. Shatyaziamawan, and A. S. Priambodo, “Implementasi Algoritma Path Planning dan Mapping Arena pada Mobile Robot,” *J. Tek. Elektro dan Komput. TRIAC*, vol. 9, no. 2, pp. 41–45, 2022, [Online]. Available: <https://journal.trunojoyo.ac.id/triac/article/view/13215>
- [14] N. Hasyim, A. Khumaidi, R. Indarti, A. Z. Arfianto, R. Y. Aditya, and H. Agus, “Pengendalian Kecepatan Tendangan Robot Sepak Bola Beroda Menggunakan Metode Decision Tree,” vol. 11, 2024.
- [15] D. C. Ananda, “Simulator Robot Sepak Bola Beroda Berbasis Robot Operating System,” *Nucl. Phys.*, vol. 13, no. 1, pp. 104–116, 2023.

