

Implementasi Estimasi Jarak Relatif Objek Berbasis ROS2 pada Robot Sepak Bola Menggunakan YOLO dan Kamera Omnidirectional 360°

Muhammad Jardin Saputra^{1*}, Agus Khumaidi², Riko Satrya Fajar Jaelani Putra³

^{1,2,3}Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi, Politeknik Perkapalan Negeri Surabaya, Indonesia

*Penulis Korespondensi, e-mail: muhammadjardin@student.ppns.ac.id

Received: 06/04/2026

Revised: 16/05/2026

Accepted: 16/05/2026

ABSTRAK

Penelitian ini bertujuan untuk mengimplementasikan sistem estimasi posisi relatif objek pada robot sepak bola menggunakan kamera omnidirectional 360° berbasis ROS 2 dengan metode deteksi objek YOLOv5s. Sistem dirancang untuk mendeteksi objek berupa bola dan robot secara real-time serta mengestimasi posisi objek dalam bentuk koordinat (X,Y) dalam satuan milimeter berdasarkan titik pusat kamera dan proses konversi koordinat piksel menggunakan faktor kalibrasi. Tahap preprocessing citra dilakukan menggunakan metode fisheye correction untuk mengurangi distorsi akibat lensa omnidirectional. Proses komunikasi data antar-node diimplementasikan menggunakan mekanisme publish-subscribe pada ROS 2 Humble. Dataset pelatihan terdiri dari 5.244 citra objek bola dan robot yang dilatih menggunakan framework PyTorch dengan model YOLOv5s. Hasil pengujian menunjukkan bahwa sistem mampu mengestimasi posisi objek dengan tingkat akurasi yang baik, dengan rata-rata error sebesar 2,83% untuk objek bola dan 8,25% untuk objek robot. Selain itu, peningkatan jarak objek menyebabkan kenaikan error, sedangkan peningkatan intensitas pencahayaan meningkatkan nilai confidence deteksi. Pengujian komunikasi ROS 2 menunjukkan bahwa sistem mampu bekerja secara real-time dengan frame rate sebesar 24–30 frame/s, latency komunikasi sebesar 7–19 ms, serta packet loss pada rentang 0–10%. Dengan demikian, sistem yang dikembangkan mampu mendukung proses navigasi dan persepsi lingkungan robot sepak bola secara efektif dan real-time.

Kata Kunci: Jarak Relatif Objek, Omnidirectional 360°, YOLOv5, ROS 2, Computer Vision

ABSTRACT

This study aims to implement a relative object position estimation system for a soccer robot using a 360° omnidirectional camera based on ROS 2 and the YOLOv5s object detection method. The system is designed to detect objects such as balls and robots in real time and estimate object positions in the form of (X,Y) coordinates in millimeters based on the camera center point and pixel coordinate conversion using a calibration factor. Image preprocessing was performed using the fisheye correction method to reduce distortion caused by the omnidirectional lens. Data communication between nodes was implemented using the publish-subscribe mechanism in ROS 2 Humble. The training dataset consisted of 5,244 images of ball and robot objects trained using the PyTorch framework with the YOLOv5s model. The experimental results show that the system is capable of estimating object positions with good accuracy, achieving an average error of 2.83% for ball objects and 8.25% for robot objects. In addition, increasing object distance caused higher estimation errors, while increasing light intensity improved detection confidence values. ROS 2 communication testing demonstrated that the system was able to operate in real time with a frame rate of 24–30 frame/s, communication latency of 7–19 ms, and packet loss in the range of 0–10%. Therefore, the developed system is capable of effectively supporting navigation and environmental perception processes for soccer robots in real time.

Keywords: Object Relative Distance, Omnidirectional 360°, YOLOv5, ROS 2, Computer Vision

1. PENDAHULUAN

Robot sepak bola membutuhkan kemampuan untuk mendeteksi dan mengetahui posisi objek seperti bola dan robot lawan secara *real-time* guna mendukung pengambilan keputusan. Namun,

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



9 772356 053009

penggunaan kamera konvensional memiliki keterbatasan sudut pandang sehingga tidak mampu mengamati seluruh lingkungan secara optimal. Oleh karena itu, kamera omnidirectional 360° digunakan untuk memberikan cakupan pandangan yang lebih luas dan menyeluruh terhadap lingkungan sekitar robot [1].

Dalam proses deteksi objek, metode berbasis *deep learning* seperti YOLO banyak digunakan karena memiliki kemampuan deteksi yang cepat dan akurat dalam kondisi *real-time* [2]. Informasi hasil deteksi kemudian diproses lebih lanjut untuk memperoleh estimasi posisi relatif objek, seperti jarak yang menjadi dasar dalam sistem navigasi dan pengambilan keputusan [3].

Selain itu, komunikasi data antar sistem pada robot memerlukan arsitektur yang efisien dan terstruktur. Robot Operating System (ROS) merupakan salah satu framework yang banyak digunakan dalam pengembangan robot karena mendukung komunikasi berbasis topik antar komponen sistem. Dengan adanya sistem komunikasi ini, data hasil deteksi dan estimasi posisi dapat didistribusikan secara *real-time* [4]. Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengimplementasikan sistem estimasi posisi relatif objek menggunakan kamera omnidirectional 360° serta mengintegrasikannya dengan komunikasi berbasis ROS 2 pada robot sepak bola.

2. METODE PENELITIAN

2.1 Sistem Deteksi Objek Menggunakan YOLO

Sistem deteksi objek pada penelitian ini menggunakan metode *You Only Look Once* (YOLO), yaitu algoritma berbasis *Convolutional Neural Network* (CNN) yang dirancang untuk melakukan deteksi objek secara *real-time*. YOLO bekerja dengan pendekatan single-stage detection, di mana proses klasifikasi dan lokalisasi objek dilakukan secara bersamaan dalam satu tahap. Metode ini mampu mendeteksi objek pada berbagai posisi dan skala dalam citra dengan kecepatan tinggi, sehingga cocok digunakan pada sistem robot yang membutuhkan respons cepat [8].



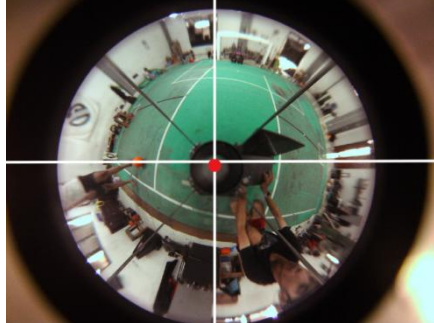
Gambar 1. Flowchart sistem kerja YOLO

Arsitektur YOLOv5 terdiri dari tiga komponen utama, yaitu backbone, neck, dan head. Backbone berfungsi untuk mengekstraksi fitur dari citra masukan, sedangkan neck menggabungkan informasi dari berbagai level fitur untuk meningkatkan kualitas representasi. Selanjutnya, head menghasilkan prediksi berupa bounding box, klasifikasi objek, dan nilai confidence. Dengan arsitektur yang telah dioptimalkan, YOLOv5 mampu memberikan keseimbangan antara kecepatan dan akurasi, sehingga banyak digunakan dalam berbagai aplikasi deteksi objek berbasis visi komputer [9].



2.2 Estimasi Posisi Relatif Objek

Estimasi posisi relatif objek dilakukan untuk mengetahui jarak dan posisi objek terhadap robot berdasarkan hasil deteksi pada citra. Proses ini memanfaatkan koordinat piksel dari objek yang terdeteksi, kemudian dikonversi menjadi informasi posisi relatif terhadap titik pusat kamera [10]. Titik pusat citra digunakan sebagai referensi utama yang merepresentasikan posisi robot. Pada penelitian ini, pusat kamera ditentukan sebagai titik tengah frame, sehingga setiap objek yang terdeteksi akan memiliki selisih koordinat terhadap titik tersebut. Selisih ini digunakan untuk menentukan posisi objek pada bidang citra.



Gambar 2. Titik pusat kamera

Untuk memperoleh posisi relatif objek, dilakukan perhitungan selisih koordinat antara titik objek (x_{obj}, y_{obj}) dan titik pusat citra (x_c, y_c) , yang dinyatakan sebagai :

$$dx = x_{obj} - x_c \quad (1)$$

$$dy = y_c - y_{obj} \quad (2)$$

Nilai dx dan dy menunjukkan posisi objek terhadap pusat kamera dalam satuan piksel. Selanjutnya, nilai tersebut dikonversi ke satuan milimeter (mm) menggunakan faktor kalibrasi k , sehingga diperoleh koordinat relatif objek:

$$X = dx \cdot k \quad (3)$$

$$Y = dy \cdot k \quad (4)$$

Nilai X dan Y merupakan posisi objek terhadap robot pada sumbu horizontal dan vertikal dalam satuan milimeter, yang secara langsung merepresentasikan jarak relatif objek terhadap robot pada bidang dua dimensi.

$$k = \frac{\text{Jarak Aktual (mm)}}{\text{Jarak Piksel (pixel)}} \quad (5)$$

Nilai faktor kalibrasi k diperoleh melalui proses pengukuran langsung antara jarak aktual objek terhadap robot dengan jarak objek pada citra dalam satuan piksel. Pengukuran dilakukan pada beberapa titik referensi untuk memperoleh rasio konversi antara piksel dan milimeter.

Sebagai contoh, objek diletakkan pada jarak 1000 mm dari robot dan menghasilkan selisih koordinat sebesar 120 piksel pada citra. Berdasarkan hasil tersebut diperoleh faktor kalibrasi:

$$k = \frac{1000}{120} = 8.3 \text{ mm/pixel} \quad (6)$$

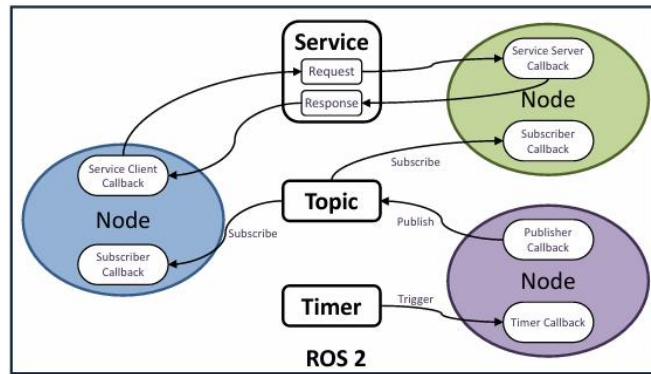
$$X = 120 \times 8.3 = 996 \text{ mm} \quad (7)$$

2.3 Komunikasi Data menggunakan ROS 2

Pada ROS 2, sistem terdiri dari beberapa *node* yang saling berkomunikasi melalui *topic* menggunakan metode *publish-subscribe*. *Node* merupakan unit proses yang menjalankan fungsi tertentu, sedangkan *topic* berfungsi sebagai media pertukaran data antar *node* [11].

Node yang mengirim data disebut sebagai *publisher*, sedangkan *node* yang menerima data disebut sebagai *subscriber*. *Publisher* mengirimkan data ke sebuah *topic*, kemudian data tersebut dapat diakses oleh satu atau lebih *subscriber* tanpa memerlukan koneksi langsung antar *node* [12].





Gambar 3. Komponen Komunikasi

Dalam mekanisme *publish-subscribe*, data yang dihasilkan oleh publisher dikirimkan ke sebuah topic sebagai media perantara. Selanjutnya, *subscriber* yang terhubung pada topic tersebut menerima data tanpa memerlukan koneksi langsung dengan publisher. Dengan demikian, komunikasi dalam ROS 2 berlangsung secara tidak langsung melalui topic, sehingga memungkinkan pertukaran data antar node secara fleksibel dan terdistribusi [13].

Struktur komunikasi tersebut ditunjukkan pada Gambar 3, yang menggambarkan hubungan antara node, topic, serta mekanisme *publish-subscribe* dalam ROS 2. Pada penelitian ini, mekanisme *publish-subscribe* diimplementasikan untuk mengirimkan data posisi objek dalam bentuk koordinat (X,Y) antar node dalam sistem robot yang dikembangkan

3. HASIL DAN PEMBAHASAN

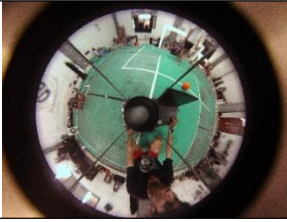
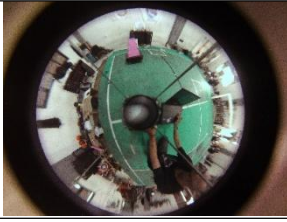
3.1 Dataset dan Pra-Pemrosesan Data Model YOLO

Dataset yang digunakan dalam penelitian ini berupa kumpulan citra objek bola yang disusun berdasarkan karakteristik visual utama, yaitu warna dominan (oranye) dan bentuk geometris (bulat). Pengambilan dataset dilakukan pada beberapa kondisi pencahayaan dan posisi objek untuk meningkatkan kemampuan generalisasi model. Seluruh citra dikategorikan ke dalam satu kelas deteksi bola untuk mendukung proses pelatihan model pada sistem visi komputer robot sepak bola. Sebagaimana ditunjukkan pada Tabel 1, pengelompokan ini bertujuan untuk memudahkan model dalam membedakan objek target dari latar belakang maupun objek lain yang terdapat di lingkungan permainan. Citra diperoleh dari berbagai sumber, kemudian melalui tahap seleksi dan praproses untuk menjaga kualitas serta konsistensi data. Setiap gambar diberi anotasi sesuai kelasnya guna mendukung proses pelatihan model YOLO dalam mengenali objek bola secara spesifik. Dengan penyusunan dataset yang relevan dan terstruktur sesuai konteks robot sepak bola, diharapkan sistem deteksi objek dapat bekerja secara optimal, baik dari segi akurasi maupun kecepatan. Hal ini menjadi faktor penting dalam mendukung respons robot terhadap pergerakan bola secara real-time di lapangan.

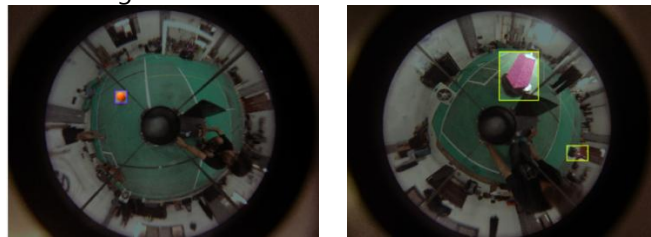
Kemudian pada tahap pra-pemrosesan data, dilakukan analisis terhadap hasil deteksi objek menggunakan model YOLO yang dilatih dengan sekitar 5.244 citra yang memuat objek bola dan robot. Setiap citra dianotasi menggunakan *bounding box* dan label kelas untuk membedakan objek bola dan robot. Proses anotasi dilakukan secara manual menggunakan perangkat lunak Roboflow agar kualitas pelabelan tetap konsisten dan akurat.



TABEL I: Dataset Bola dan Robot

Class	Bola	Robot
Dataset		
Jumlah	2051	3193
JUMLAH DATASET		5244

Hasil pelabelan ini digunakan sebagai dasar pembelajaran model dalam mengenali posisi serta karakteristik masing-masing objek. Sistem menunjukkan performa deteksi yang baik selama objek tidak mengalami tumpang tindih secara signifikan.



Gambar 4. Anotasi Objek Bola dan Robot

Sebelum proses pelatihan, dataset dibagi menjadi tiga bagian, yaitu data pelatihan (70%), validasi (20%), dan pengujian (10%). Pembagian ini bertujuan untuk memastikan model dapat belajar secara optimal, menghindari overfitting, serta mampu melakukan generalisasi terhadap data baru yang belum pernah dilihat sebelumnya.

TABEL II: Pembagian Training Data

Class	Train (70 %)	Valid (20 %)	Test (10 %)
Bola	1.436	410	205
Robot	2.235	639	319

3.2 Pelatihan Model YOLO

Proses pelatihan model dilakukan menggunakan platform Google Colab Pro. Model YOLOv5 yang digunakan pada penelitian ini adalah YOLOv5s dengan framework PyTorch. Proses pelatihan dilakukan menggunakan GPU NVIDIA Tesla T4 pada platform Google Colab Pro dengan sistem operasi berbasis Linux. Parameter yang digunakan selama proses pelatihan ditunjukkan pada Tabel III. Dengan parameter pelatihan yang digunakan pada Tabel III, model menunjukkan performa yang optimal ketika jumlah epoch mencapai lebih dari 250. Hal tersebut ditunjukkan melalui grafik train/cls_loss dan val/cls_loss yang mengalami penurunan dan cenderung stabil selama proses pelatihan berlangsung.

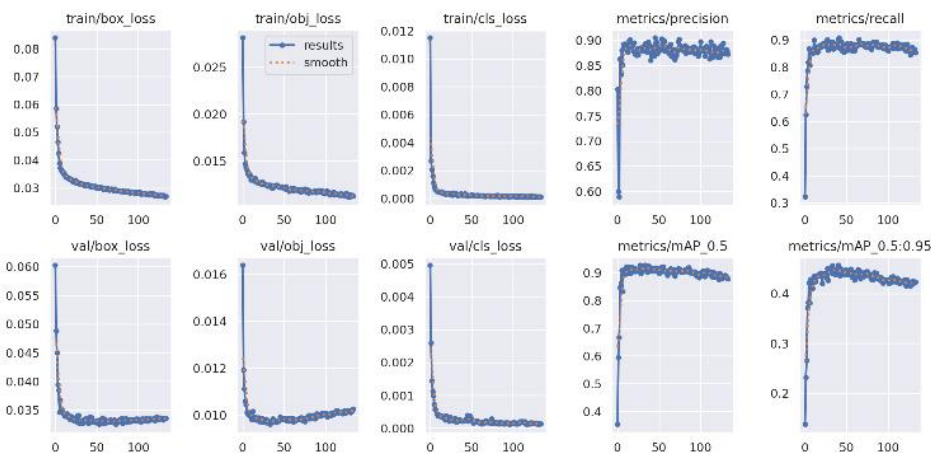
TABEL III: Parameter Pelatihan Model YOLO

Parameter	Value
Image Size	416 x 416
Batch Size	16
Epoch	250

Berikut merupakan hasil performa keseluruhan model dari proses pelatihan yang telah dilakukan pada penelitian ini.



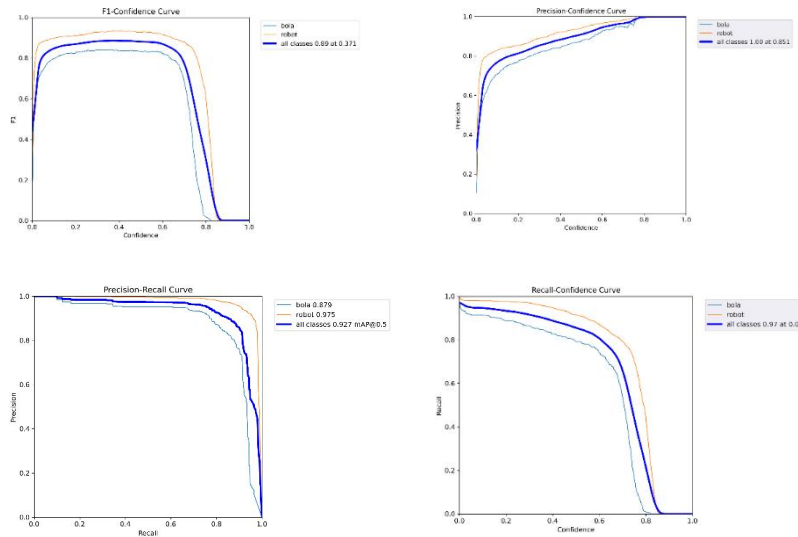
- Precision: Digunakan untuk mengukur tingkat ketepatan model dalam melakukan prediksi positif, yaitu seberapa banyak prediksi positif yang sesuai dengan kondisi sebenarnya.
- Recall: Digunakan untuk mengetahui kemampuan model dalam mendeteksi seluruh data positif yang tersedia. Nilai ini menunjukkan tingkat keberhasilan model dalam mengenali objek target.
- F1 Score: Merupakan nilai rata-rata harmonik antara precision dan recall. Metode ini digunakan untuk mengevaluasi keseimbangan performa model, terutama pada kondisi jumlah data antar kelas tidak seimbang.
- Accuracy: Menunjukkan tingkat ketepatan model secara keseluruhan berdasarkan jumlah prediksi yang benar dibandingkan total seluruh prediksi yang dilakukan.
- Confusion Matrix: Digunakan untuk menampilkan jumlah prediksi benar dan salah pada masing-masing kelas, seperti objek bola dan robot. Metode ini membantu dalam menganalisis jenis kesalahan yang dilakukan model, baik false positive maupun false negative.
- Average Precision (AP): Digunakan untuk mengevaluasi performa model berdasarkan rata-rata nilai precision pada berbagai threshold deteksi. Nilai AP diperoleh dari Precision-Recall Curve dan umum digunakan sebagai parameter evaluasi pada sistem Computer Vision.



Gambar 5. Hasil training dengan parameter yang telah ditentukan

Grafik pada Gambar 6 menunjukkan hasil evaluasi performa model YOLOv5 menggunakan beberapa parameter evaluasi, yaitu Precision-Recall Curve, F1-Confidence Curve, Precision-Confidence Curve, dan Recall-Confidence Curve untuk kelas bola dan robot. Pengujian model dilakukan menggunakan 250 epoch dengan ukuran citra 416×416 piksel. Pada grafik Precision-Recall Curve diperoleh nilai mAP@0.5 sebesar 0.927 untuk seluruh kelas, dengan nilai average precision sebesar 0.879 untuk kelas bola dan 0.975 untuk kelas robot. Hasil tersebut menunjukkan bahwa model mampu mendeteksi objek dengan tingkat presisi dan recall yang cukup tinggi. Grafik F1-Confidence Curve menunjukkan nilai F1-score maksimum sebesar 0.89 pada confidence threshold 0.371 untuk seluruh kelas. Hal ini menunjukkan bahwa model memiliki keseimbangan yang baik antara precision dan recall pada threshold tersebut.







Gambar 6. Grafik kurva pelatihan model

Pada grafik Recall-Confidence Curve diperoleh nilai recall maksimum sebesar 0.97 pada confidence threshold 0.000 untuk seluruh kelas. Nilai ini menunjukkan bahwa model mampu mendeteksi hampir seluruh objek positif pada threshold rendah, meskipun kemungkinan false positive masih dapat terjadi. Sementara itu, grafik Precision-Confidence Curve menunjukkan nilai precision maksimum sebesar 1.00 pada confidence threshold 0.851 untuk seluruh kelas. Hal ini menunjukkan bahwa model memiliki tingkat ketepatan deteksi yang sangat baik pada threshold tersebut. Secara keseluruhan, hasil evaluasi menunjukkan bahwa model YOLOv5 yang digunakan memiliki performa yang baik dan stabil dalam mendeteksi objek bola dan robot pada berbagai kondisi pengujian.

3.3 Hasil Estimasi Jarak Relatif Objek

Koordinat hasil deteksi kemudian digunakan untuk menghitung posisi relatif objek terhadap robot dalam bentuk koordinat (X, Y) dalam satuan milimeter. Titik pusat citra digunakan sebagai referensi utama, kemudian selisih koordinat dikonversi menggunakan faktor kalibrasi. Hasil pengujian menunjukkan bahwa sistem mampu merepresentasikan posisi objek secara dua dimensi dengan cukup baik.

TABEL IV: Pengujian deteksi bola

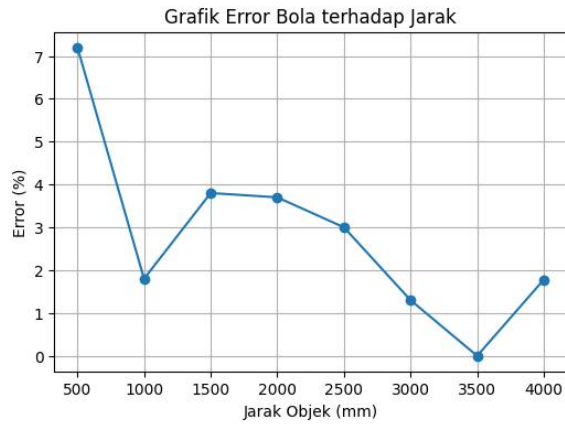
No	Gambar	Jarak Nyata (mm)	X Bola (mm)	Y Bola (mm)	Error %	Nilai Confidence	Lux
1		500	0	536	7.20%	0.75	82
2		1000	0	1018	1.80%	0.72	109



3



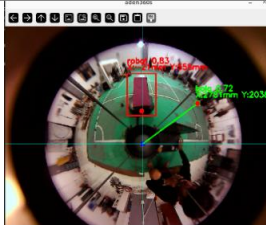
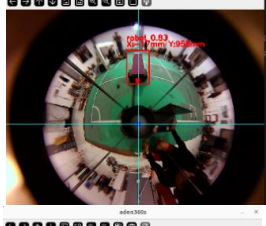

1500 0 1557 3.80% 0.81 182



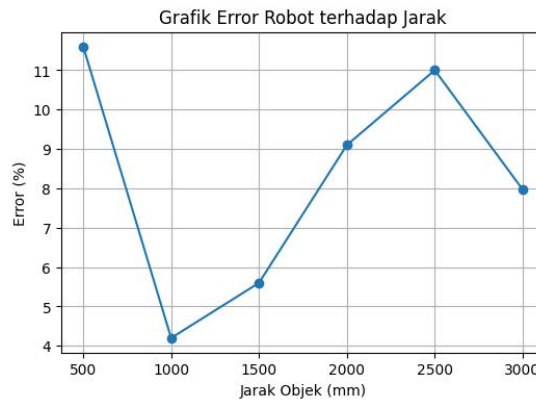
Gambar 7. Grafik error deteksi bola terhadap jarak objek

Titik pusat citra digunakan sebagai referensi utama, kemudian selisih koordinat dikonversi menggunakan faktor kalibrasi. Hasil pengujian menunjukkan bahwa sistem mampu merepresentasikan posisi objek secara dua dimensi dengan cukup baik.

TABEL V: Pengujian deteksi robot

No	Gambar	Jarak Nyata (mm)	X Robot (mm)	Y Robot (mm)	Error %	Nilai Confidence	Lux
1		500	-2	558	11.60%	0.83	98
2		1000	-1	958	4.20%	0.83	102
3		1500	-1	1416	5.60%	0.81	119





Gambar 8. Grafik error deteksi robot terhadap jarak objek

Grafik hubungan error deteksi robot terhadap jarak objek menunjukkan bahwa nilai error pada objek robot cenderung lebih tinggi dibandingkan objek bola. Hal ini disebabkan bentuk robot yang tidak simetris sehingga centroid bounding box lebih sulit ditentukan secara stabil.

Berdasarkan hasil pengujian pada Tabel 3 dan Tabel 4, error dihitung dengan rumus sebagai berikut,

$$Error(\%) = \frac{|Jarak\ Aktual - Jarak\ Estimasi|}{Jarak\ Aktual} \times 100\% \quad (8)$$

Sebagai contoh, pada pengujian objek bola dengan jarak aktual 500 mm dan hasil estimasi sebesar 536 mm, maka nilai error dihitung sebagai:

$$Error = \frac{|500 - 536|}{500} \times 100\% = 7.2\% \quad (9)$$

Sistem mampu mengestimasi posisi objek dalam bentuk koordinat (X, Y) dengan tingkat akurasi yang baik. Pada objek bola, rata-rata error yang diperoleh sebesar 2,83%, sedangkan pada objek robot sebesar 8,25%. Hal ini menunjukkan bahwa sistem memiliki performa yang cukup akurat dalam merepresentasikan posisi objek terhadap robot.

Perbedaan tingkat error antara objek bola dan robot dipengaruhi oleh karakteristik bentuk objek. Bola memiliki bentuk geometris yang simetris sehingga menghasilkan titik pusat (centroid) yang lebih stabil dan mudah ditentukan. Sebaliknya, robot memiliki bentuk yang lebih kompleks dan tidak simetris, sehingga proses penentuan centroid menjadi kurang konsisten dan berkontribusi terhadap peningkatan error.

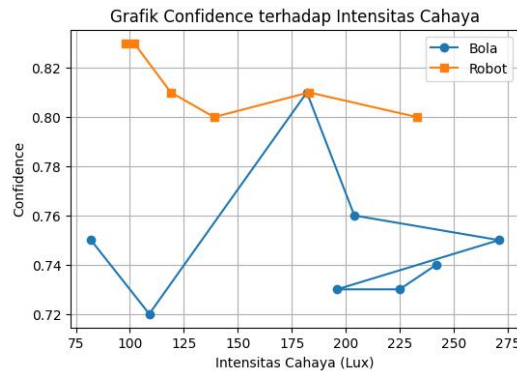
Selain itu, hasil pengujian menunjukkan bahwa nilai error cenderung meningkat seiring dengan bertambahnya jarak objek dari kamera. Fenomena ini disebabkan oleh efek distorsi pada kamera omnidirectional serta keterbatasan resolusi citra, di mana perubahan kecil pada koordinat piksel dapat menghasilkan perbedaan jarak yang lebih signifikan dalam satuan milimeter setelah proses konversi. Peningkatan error pada jarak yang lebih jauh dipengaruhi oleh penurunan resolusi objek pada citra. Semakin jauh objek dari kamera, ukuran bounding box menjadi lebih kecil sehingga perubahan kecil pada koordinat piksel menghasilkan perubahan estimasi jarak yang lebih besar setelah proses konversi ke satuan milimeter.

Pada objek robot, nilai error yang lebih besar dibandingkan objek bola dipengaruhi oleh bentuk robot yang tidak simetris sehingga titik centroid bounding box lebih sulit ditentukan secara konsisten.

Berdasarkan data pengujian, intensitas pencahayaan (lux) berada pada rentang 82 hingga 271 lux untuk objek bola dan 98 hingga 233 lux untuk objek robot. Secara umum, sistem tetap mampu mendeteksi objek dengan baik pada berbagai kondisi pencahayaan tersebut. Namun demikian, terdapat kecenderungan bahwa peningkatan nilai lux berbanding lurus dengan peningkatan nilai confidence dari model YOLO.

Pada kondisi pencahayaan rendah (sekitar 82–109 lux), nilai confidence berada pada kisaran 0,72–0,75, sedangkan pada kondisi pencahayaan yang lebih tinggi (di atas 180 lux), nilai confidence meningkat hingga mencapai sekitar 0,81. Hal ini menunjukkan bahwa kualitas pencahayaan berpengaruh terhadap kemampuan model dalam mengenali objek secara lebih akurat.





Gambar 9. Grafik confidence deteksi terhadap intensitas cahaya

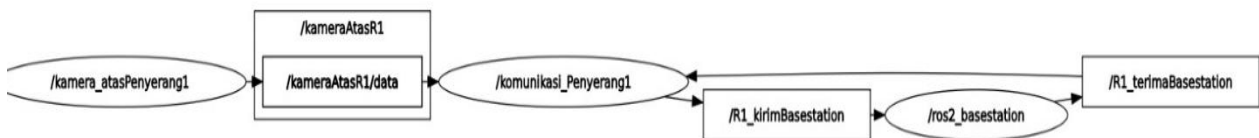
Berdasarkan grafik hubungan confidence deteksi terhadap intensitas cahaya, terlihat bahwa peningkatan nilai lux cenderung meningkatkan nilai confidence model YOLO. Hal ini menunjukkan bahwa kualitas pencahayaan berpengaruh terhadap kemampuan sistem dalam mengenali objek secara lebih akurat.

Meskipun nilai error objek robot sebesar 8,25% lebih tinggi dibandingkan objek bola, sistem masih dapat digunakan untuk kebutuhan navigasi robot sepak bola karena mampu memberikan estimasi arah dan posisi objek secara real-time. Akurasi sistem masih dapat ditingkatkan melalui penambahan dataset pelatihan, augmentasi citra, optimasi kalibrasi kamera, serta penggunaan model deteksi yang lebih ringan dan stabil terhadap objek kompleks.

Dibandingkan dengan metode deteksi konvensional berbasis warna atau thresholding, penggunaan YOLOv5 pada penelitian ini memberikan keunggulan dalam kemampuan mendeteksi objek pada berbagai kondisi pencahayaan dan posisi objek secara lebih stabil. Selain itu, integrasi dengan ROS 2 memungkinkan distribusi data posisi objek secara modular dan real-time.

3.4 Flow Komunikasi Data Menggunakan ROS 2

Alur komunikasi sistem berbasis ROS 2 ditunjukkan pada Gambar 12 melalui visualisasi *RQT Graph*. Berdasarkan gambar tersebut, sistem terdiri dari beberapa *node* yang saling terhubung melalui *topic* menggunakan mekanisme *publish-subscribe*.



Gambar 10. Flow Komunikasi Data Kamera menggunakan ROS 2

Node `/kamera_atasPenyerang1` berperan sebagai sumber data citra yang kemudian diteruskan ke node `/kameraAtasR1/data`. Node ini berfungsi untuk memproses data kamera dan menghasilkan informasi yang akan digunakan dalam sistem. Selanjutnya, data tersebut dikirim ke node `/komunikasi_Penyerang1` yang bertindak sebagai penghubung komunikasi antar modul. Node ini meneruskan data ke node `/R1_kirimBasestation`, yang berfungsi sebagai *publisher* untuk mengirimkan data ke sistem eksternal melalui topic `/ros2_basestation`. Data yang telah dikirim kemudian diterima oleh node `/R1_terimaBasestation` sebagai *subscriber*. Node ini berfungsi untuk menerima data hasil komunikasi dan menggunakannya dalam proses lanjutan pada sistem robot. Struktur komunikasi ini menunjukkan bahwa pertukaran data antar node tidak dilakukan secara langsung, melainkan melalui *topic* sebagai perantara.

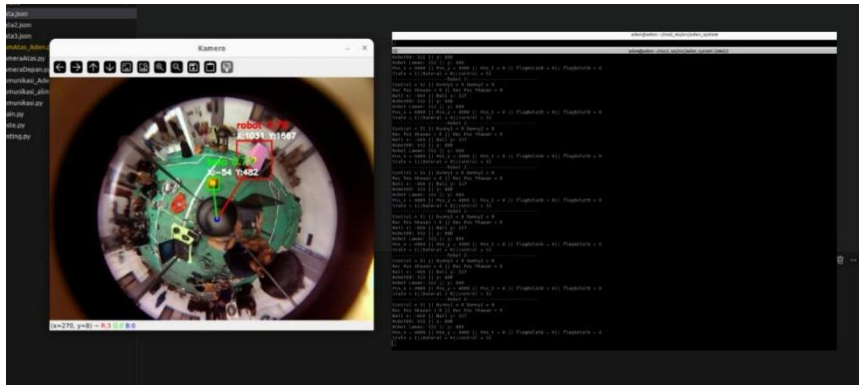
3.5 Hasil Komunikasi Data YOLO pada ROS 2

Hasil pengujian komunikasi data pada sistem menunjukkan bahwa data hasil deteksi objek dari model YOLO dapat dikirimkan melalui ROS 2 dalam bentuk koordinat posisi objek secara real-time.



Berdasarkan Gambar 11, data yang dihasilkan oleh sistem berupa informasi posisi objek dalam bentuk koordinat ($X \cdot Y$) dalam satuan milimeter.

Proses pengiriman data berlangsung secara kontinu tanpa adanya kehilangan data (*data loss*) selama pengujian. Selain itu, tidak ditemukan keterlambatan (*delay*) yang signifikan dalam proses komunikasi, sehingga data dapat diterima secara langsung oleh sistem penerima. Hasil ini menunjukkan bahwa integrasi antara sistem deteksi objek menggunakan YOLO dan komunikasi berbasis ROS 2 berjalan dengan baik. Sistem mampu mendistribusikan informasi posisi objek secara cepat dan stabil, sehingga dapat mendukung kebutuhan navigasi robot dalam kondisi real-time.



Gambar 13. Data objek dari kamera

Untuk mengevaluasi performa komunikasi real-time pada ROS 2, dilakukan pengujian terhadap parameter latency, jitter, dan packet loss selama proses pengiriman data koordinat objek antar-node berlangsung.

TABEL VI: Pengujian Komunikasi node ROS 2

Parameter	Node Pengiriman	Node Penerimaan
Frame/s	24-30 frame/s	24-30 frame/s
Latency rata-rata	18-19 ms	7-10 ms
Jitter	0-4,2 ms	0,2-20,7 ms
Packet Loss	0-10%	0-10%
Status Komunikasi	Stabil	Stabil

Berdasarkan hasil pengujian komunikasi ROS 2 pada node pengiriman dan penerimaan, diperoleh latency rata-rata sebesar 18-19 ms pada node pengiriman dan 7-10 ms pada node penerimaan. Nilai jitter berada pada rentang 0-4,2 ms untuk node pengiriman dan 0,2-20,7 ms pada node penerimaan. Selain itu, packet loss selama pengujian berada pada rentang 0-10%, di mana sebagian besar proses pengiriman data berlangsung tanpa kehilangan paket. Sistem juga mampu berjalan pada frame rate sebesar 24-30 FPS menggunakan GPU NVIDIA RTX 3050 sehingga proses deteksi objek dan komunikasi data dapat berlangsung secara real-time. Hasil tersebut menunjukkan bahwa komunikasi antar-node menggunakan ROS 2 mampu berjalan secara cepat dan stabil untuk mendukung kebutuhan navigasi robot sepak bola.

3.6 Keterbatasan Penelitian

Penelitian ini masih memiliki keterbatasan karena pengujian dilakukan pada kondisi lingkungan yang terkontrol dan belum sepenuhnya merepresentasikan kondisi pertandingan robot sepak bola yang dinamis dengan banyak objek bergerak secara simultan. Selain itu, proses estimasi posisi masih dipengaruhi oleh distorsi kamera omnidirectional dan perubahan pencahayaan lingkungan. Penggunaan metode estimasi berbasis koordinat piksel juga masih memiliki keterbatasan pada objek dengan bentuk tidak simetris seperti robot. Pengembangan selanjutnya dapat dilakukan melalui pengujian pada kondisi pertandingan nyata, implementasi multi-object tracking, serta integrasi dengan sistem navigasi autonomous untuk meningkatkan kemampuan generalisasi sistem.



4. KESIMPULAN

Sistem estimasi posisi relatif objek menggunakan kamera omnidirectional 360° dan YOLOv5 berhasil diterapkan pada robot sepak bola. Sistem mampu mendeteksi bola dan robot secara real-time serta menghasilkan koordinat posisi objek dalam satuan milimeter. Hasil pengujian menunjukkan rata-rata error sebesar 2,83% untuk bola dan 8,25% untuk robot, di mana error robot lebih besar karena bentuknya lebih kompleks dibandingkan bola yang simetris. Performa sistem dipengaruhi oleh jarak objek dan pencahayaan; jarak yang semakin jauh meningkatkan error, sedangkan pencahayaan yang baik meningkatkan confidence dan kestabilan deteksi. Komunikasi data berbasis ROS 2 juga berjalan stabil secara real-time tanpa kehilangan data atau keterlambatan signifikan. Secara keseluruhan, sistem ini layak digunakan untuk mendukung navigasi dan pengambilan keputusan pada robot sepak bola.

5. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik Perkapalan Negeri Surabaya (PPNS), khususnya Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi, atas dukungan fasilitas dan lingkungan akademik yang telah diberikan selama pelaksanaan penelitian ini. Penulis juga menyampaikan apresiasi kepada dosen pembimbing atas bimbingan, arahan, dan masukan yang sangat berharga dalam penyusunan penelitian ini. Ucapan terima kasih juga diberikan kepada tim **Gerhana Dewaruci** KRSBI-B PPNS yang telah memberikan dukungan, kerja sama, serta kontribusi dalam proses pengembangan, pengujian, dan implementasi sistem robot pada penelitian ini. Selain itu, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu sehingga penelitian ini dapat diselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] S. S. Hidayatullah and E. Saputra, "Perancangan Omni Vision Camera sebagai Sistem Pendeteksi Objek untuk Robot Sepak Bola," vol. 4, pp. 229–235, 2022.
- [2] A. R. Ramadhan, A. Khumaidi, M. K. Mayangsari, M. Syai, and I. Sutrisno, "Penerapan Harris Corner Detection dan YOLOv5 pada Kamera Stereo Vision untuk Estimasi Jarak Robot Sepak Bola Beroda KRSBI-B," vol. 12, 2025.
- [3] A. A. Fikri, S. Robot, O. Avoidance, O. Scenario, and I. Learning, "Journal of Intelligent System and Imitation Learning Based Obstacle Avoidance for MSL Soccer Robot in Offensive Scenario," vol. 1, pp. 84–95, 2024.
- [4] A. A. Wahyudi, Agus Khumaidi, Mohammad Basuki Rahmat, Dimas Pristovani Riananda, Mat Syai'in, and Joko Endrasmono, "Implementasi Robot Operating System (ROS) Untuk Meningkatkan Akurasi Deteksi Bola Menggunakan YOLO V5 Pada KRSBI-Beroda," *J. Elektron. dan Otomasi Ind.*, vol. 11, no. 2, pp. 648–661, 2024, doi: 10.33795/elkolind.v11i2.5234.
- [5] A. Irwansyah, A. Z. Putra, and R. Akbar, "Omnidirectional Camera untuk Positioning Robot Soccer dengan Metode 2-Fixed-Point," *Techné J. Ilm. Elektrotek.*, vol. 22, no. 2, pp. 273–284, 2023, doi: 10.31358/techné.v22i2.374.
- [6] A. Bonci, F. Gaudeni, M. C. Giannini, and S. Longhi, *Robot Operating System 2 (ROS2)-Based Frameworks for Increasing Robot Autonomy: A Survey*, vol. 13, no. 23, 2023. doi: 10.3390/app132312796.
- [7] E. Endre, M. Dahl, K. Bengtsson, A. Hanna, and P. Falkman, "ScienceDirect A ROS2 based communication architecture for control in collaborative and intelligent automation systems," vol. 00, pp. 1–9, 2019.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection".
- [9] R. Wulanningrum and A. Sanjaya, "Implementasi YOLO Dalam Deteksi Jumlah Kendaraan," vol. 8, pp. 1274–1281, 2024.
- [10] R. Dikairono, M. A. Ismail, H. Dendy, and K. Arief, "Artificial Neural Network Untuk Pengukuran Posisi Bola Menggunakan Kamera Omnidireksional Pada Robot Sepak Bola Beroda," pp. 1–5.
- [11] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design , Architecture , and Uses In The Wild", doi: 10.1126/scirobotics.abm6074.
- [12] T. Ishikawa-aso and S. Kato, "ROS 2 Agnocast : Supporting Unsized Message Types for True Zero-Copy Publish / Subscribe IPC," 2025.
- [13] A. S. Al-Batati, A. Koubaa, M. Abdelkader, K. Gabr, and H. Aloqaily, "ROS 2 in a Nutshell: A Survey," vol. V3, no. October, 2025, doi: 10.20944/preprints202410.1204.v2.

