

Implementasi Komunikasi *Real-Time* RefBox ke Robot Berbasis Node ROS 2 dengan Analisis Latensi

Sufadi Alim^{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: sufadialim@student.ppns.ac.id

Received: 20/04/2026

Revised: 21/05/2026

Accepted: 21/05/2026

ABSTRAK

Penelitian ini mengembangkan sistem komunikasi antara Referee Box (RefBox), Base Station, dan robot sepak bola beroda menggunakan arsitektur modular berbasis node pada ROS 2 untuk meningkatkan modularitas dan skalabilitas dibandingkan pendekatan monolitik. Arsitektur sistem terdiri dari beberapa node utama, yaitu Base Station Node, Robot Node, dan Status Node yang saling berkomunikasi melalui topik ROS 2 dengan protokol *Data Distribution Service* (DDS). Pengujian dilakukan untuk mengevaluasi kesesuaian perintah RefBox, respon sistem, serta performa latensi komunikasi, dengan hasil menunjukkan bahwa seluruh perintah RefBox berhasil diterjemahkan secara akurat menjadi instruksi robot, termasuk penentuan koordinat target sesuai skenario permainan, serta divisualisasikan secara konsisten. Analisis latensi menunjukkan rata-rata waktu komunikasi sebesar 131,6 ms dari RefBox ke Base Station dan 96,6 ms dari Base Station ke robot, dengan total latensi rata-rata 228,2 ms, serta seluruh pengujian menunjukkan komunikasi yang stabil tanpa kehilangan data. Hasil ini membuktikan bahwa implementasi arsitektur modular berbasis node ROS 2 mampu meningkatkan keandalan komunikasi, fleksibilitas pengembangan sistem, dan mendukung operasi robot secara *real-time* dalam sistem multi-node.

Kata Kunci: Base Station, KRSBI-B, Refereebox, ROS 2

ABSTRACT

This study develops a communication system between the Referee Box (RefBox), Base Station, and wheeled soccer robots using a node-based architecture in ROS 2 to improve modularity and scalability compared to conventional monolithic approaches. The system architecture consists of several main nodes, namely the Base Station Node, Robot Node, and Status Node, which communicate through ROS 2 topics using the Data Distribution Service (DDS) protocol. Testing was conducted to evaluate the accuracy of RefBox commands, system responses, and communication latency performance. The results show that all RefBox commands were successfully translated into appropriate robot instructions, including determining target coordinates according to game scenarios, and were consistently visualized. Latency analysis indicates an average communication time of 131.6 ms from RefBox to the Base Station and 96.6 ms from the Base Station to the robot, with a total average latency of 228.2 ms. All tests demonstrated stable communication without data loss. These results confirm that the implementation of a node-based ROS 2 architecture improves communication reliability, system development flexibility, and supports real-time robot operation in a multi-node system.

Keywords: Base Station, KRSBI-B, Refereebox, ROS 2

1. PENDAHULUAN

Perkembangan teknologi robotika di Indonesia telah mengalami kemajuan pesat, salah satunya melalui kompetisi tahunan Kontes Robot Indonesia (KRI). Salah satu divisi yang paling menonjol adalah Kontes Robot Sepak Bola Indonesia Beroda (KRSBI-B), yang menuntut koordinasi tingkat tinggi antara sistem kendali pusat dan unit robot di lapangan [1]. Dalam kompetisi ini, robot harus mampu merespons instruksi dari wasit



secara *real-time* melalui sistem sentral yang dikenal sebagai Referee Box (RefBox). RefBox mengirimkan perintah pertandingan seperti Kick Off, Stop, atau Pick Up ke Base Station yang kemudian diteruskan ke robot [2].

Secara tradisional, komunikasi antara Base Station dan robot dibangun di atas arsitektur monolitik. Sistem tersebut meskipun fungsional, memiliki keterbatasan dalam hal modularitas, skalabilitas, dan integrasi perangkat tambahan, misalnya visualisasi status robot secara *real-time*. Ketika jumlah robot bertambah atau data yang dikirim menjadi lebih kompleks, sistem konvensional menghadapi kesulitan sinkronisasi data dan pemeliharaan kode, yang dapat menghambat pengembangan robot multi-unit [3].

Beberapa penelitian terdahulu telah mencoba mengimplementasikan sistem komunikasi robot berbasis TCP/IP[3], namun pendekatan ini sering kali terbatas pada satu unit robot dan tidak mudah diperluas untuk skala multi-robot. Implementasi berbasis ROS (Robot Operating System) telah menunjukkan keunggulan dalam modularitas dan manajemen node, serta dukungan protokol komunikasi *real-time* berbasis DDS, tetapi sebagian besar masih fokus pada robot simulasi atau skala laboratorium kecil [4].

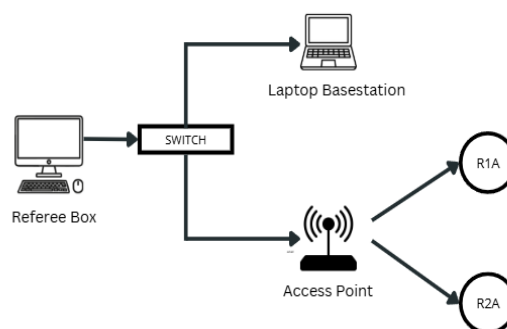
2. METODE PENELITIAN

2.1 Alur Komunikasi Sistem

Pada sistem yang dikembangkan, alur komunikasi dimulai dari perangkat Referee Box (RefBox) yang berfungsi sebagai sumber utama perintah pertandingan. Data perintah dikirimkan melalui jaringan kabel menuju switch sebagai penghubung dalam jaringan lokal (LAN). Penggunaan jaringan lokal dengan protokol komunikasi seperti TCP/IP banyak digunakan dalam sistem terdistribusi karena mampu menjamin pengiriman data yang stabil dan terstruktur antar perangkat [5]. Switch kemudian mendistribusikan data ke perangkat lain dalam jaringan, termasuk laptop base station sebagai pusat pengolahan data.

Selanjutnya, laptop base station menerima data dari RefBox untuk diproses lebih lanjut, seperti parsing dan konversi format data agar sesuai dengan kebutuhan sistem robot. Pada sistem robot modern, proses ini sangat penting untuk memastikan data dapat digunakan secara efisien dalam komunikasi antar perangkat maupun pengambilan keputusan sistem [6]. Selain itu, base station juga berperan dalam mengatur distribusi data ke robot melalui jaringan komunikasi, sehingga mendukung sistem multi-perangkat yang terintegrasi.

Setelah proses pengolahan, data dikirimkan ke *access point* untuk didistribusikan secara nirkabel ke robot. Penggunaan komunikasi *wireless* pada sistem robot memungkinkan fleksibilitas tinggi serta mendukung mobilitas robot dalam lingkungan kerja. Setiap robot kemudian menerima data perintah secara *real-time* dan mengeksekusi instruksi sesuai dengan perintah yang diberikan. Sistem komunikasi seperti ini banyak digunakan dalam sistem multi-robot karena mampu meningkatkan efisiensi koordinasi serta kecepatan respon antar robot [7].

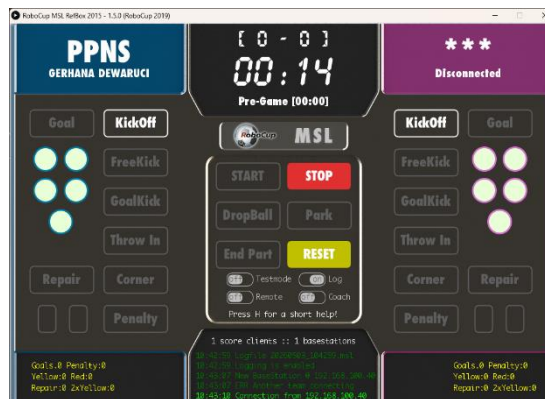


Gambar 1: Alur Komunikasi Data Referee Box

2.2 RefereeBox



Referee Box (RefBox) merupakan sistem kontrol utama yang digunakan dalam kompetisi robot sepak bola beroda untuk mengirimkan instruksi pertandingan ke robot secara *real-time*. Dalam literatur, RefBox dikembangkan sebagai interface antara wasit manusia dan robot, yang mengirimkan perintah seperti Kick Off, Stop, Free Kick, atau Pick Up melalui protokol komunikasi jaringan seperti TCP/IP atau UDP ke Base Station tim robot. Sistem ini memungkinkan robot untuk menerima dan mengeksekusi perintah secara otomatis selama pertandingan berlangsung. Penelitian menunjukkan bahwa komunikasi antara RefBox, base station, dan robot dilakukan secara *real-time* melalui jaringan kabel dan wireless sehingga robot dapat merespons perintah secara langsung sesuai kondisi permainan [8]. Oleh karena itu, penelitian ini memanfaatkan konsep RefBox sebagai titik awal pengiriman data yang akan diintegrasikan dengan arsitektur modular berbasis node ROS 2 untuk distribusi perintah secara *real-time*.



Gambar 2: RefereeBox KRSBI-B

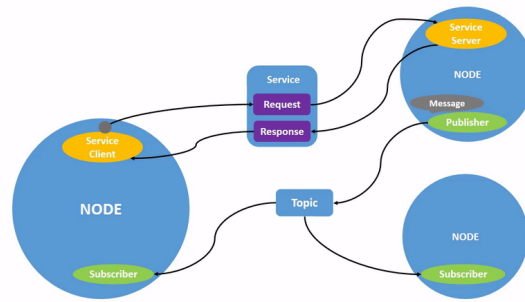
2.3 Node Based ROS 2 Humble

ROS 2 Humble merupakan framework robotika berbasis *middleware* yang menggunakan arsitektur modular berbasis node untuk mendukung pengembangan sistem robot terdistribusi. Pada ROS 2, setiap node memiliki fungsi spesifik, seperti penerimaan data, pengolahan sensor, pengiriman perintah aktuator, dan visualisasi status sistem, yang saling berkomunikasi melalui mekanisme *publisher* dan *subscriber* pada topik tertentu. Pendekatan ini memungkinkan proses pengembangan sistem menjadi lebih modular karena setiap node dapat dijalankan, diuji, dan dikembangkan secara independen tanpa memengaruhi node lainnya.

Keunggulan utama ROS 2 dibandingkan ROS generasi sebelumnya terletak pada penggunaan protokol *Data Distribution Service* (DDS) sebagai lapisan komunikasi utama [9]. DDS memungkinkan pertukaran data secara *real-time* dengan reliabilitas tinggi, latensi rendah, serta mendukung komunikasi antar perangkat dalam sistem robot multi-*node* [10]. Selain itu, ROS 2 juga menyediakan fitur *Quality of Service* (QoS) yang dapat disesuaikan berdasarkan kebutuhan sistem, seperti reliabilitas pengiriman data, durasi penyimpanan data, dan prioritas komunikasi.

Penelitian terdahulu menunjukkan bahwa implementasi ROS 2 pada sistem robotika mampu meningkatkan fleksibilitas pengembangan perangkat lunak serta efisiensi komunikasi data antar node [8]. Oleh karena itu, pada penelitian ini ROS 2 Humble digunakan untuk membangun sistem komunikasi berbasis node yang terdiri atas Base Station Node, Robot Node, dan Status Node yang saling terhubung melalui topik ROS, sehingga perintah dari RefBox dapat diteruskan ke robot dengan mekanisme komunikasi *real-time* serta monitoring status sistem secara terintegrasi.





Gambar 3: Node ROS 2 Humble

2.4 Base Station

Base Station pada KRSBI-B berperan sebagai penghubung utama antara RefBox dan robot, yang bertanggung jawab dalam menerima, memproses, dan meneruskan perintah ke masing-masing robot secara *real-time*. Sistem ini memungkinkan robot untuk bergerak secara otomatis berdasarkan instruksi pertandingan yang dikirimkan oleh RefBox melalui jaringan komunikasi tertentu. Selain sebagai penerima data, Base Station juga berfungsi sebagai pengolah informasi yang memastikan setiap robot menerima perintah secara tepat sesuai kondisi permainan. Komunikasi antara Base Station dan robot umumnya memanfaatkan jaringan *wireless* atau protokol TCP/IP untuk meningkatkan kecepatan dan keandalan pengiriman data [11].

Namun, studi literatur menunjukkan bahwa Base Station konvensional masih menggunakan arsitektur terpusat (monolitik), di mana proses penerimaan data, pemrosesan, dan distribusi perintah dilakukan dalam satu sistem, sehingga kurang fleksibel dalam pengembangan sistem yang lebih kompleks seperti multi-robot [12]. Pendekatan ini menyulitkan integrasi fitur tambahan seperti monitoring dan visualisasi serta kurang optimal dalam hal skalabilitas. Oleh karena itu, pada penelitian ini dikembangkan Base Station berbasis ROS 2 Humble dengan pendekatan berbasis *node* yang memisahkan fungsi sistem ke dalam beberapa *node* independen [13], sehingga meningkatkan modularitas, skalabilitas, dan kemampuan sistem dalam menangani komunikasi data secara *real-time*.

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan pada penelitian ini menyajikan implementasi sistem komunikasi berbasis ROS 2 serta performa komunikasi antara RefBox, Base Station, dan robot. Data yang diperoleh ditampilkan dalam bentuk narasi, gambar, dan tabel untuk menunjukkan kinerja sistem secara terukur. Setiap hasil yang disajikan berfokus pada pengujian komunikasi dan respon sistem tanpa memberikan kesimpulan akhir, melainkan sebagai dasar analisis terhadap performa sistem yang dikembangkan.

3.1 Arsitektur Komunikasi Berbasis Node ROS 2

Sistem komunikasi pada penelitian ini menggunakan arsitektur modular berbasis *node* ROS 2 untuk memisahkan fungsi penerimaan data, pemrosesan perintah, kontrol robot, dan monitoring sistem secara independen. Arsitektur komunikasi yang digunakan ditunjukkan pada Gambar 4, yang terdiri atas tiga bagian utama yaitu Sistem RefereeBox, Base Station, dan Sistem Robot yang saling terhubung melalui protokol TCP/IP dan komunikasi *publisher-subscriber* ROS 2.

Pada lapisan Sistem RefereeBox, perangkat RefBox berfungsi sebagai sumber utama perintah pertandingan. RefBox mengirimkan data pertandingan melalui komunikasi TCP/IP menuju Base Station. Perintah yang dikirim meliputi instruksi pertandingan seperti START, STOP, KICKOFF, PENALTY, dan CORNER yang digunakan sebagai acuan pergerakan robot selama pertandingan berlangsung.

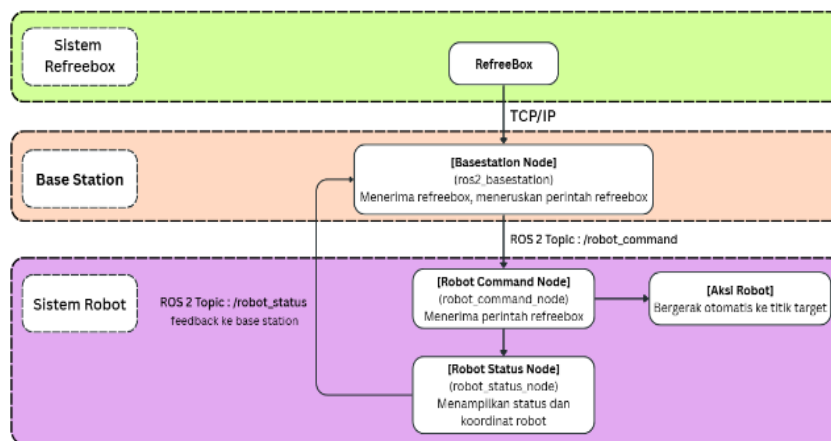
Pada sisi Base Station, data dari RefBox diterima oleh Basestation Node yang berfungsi sebagai interface komunikasi antara sistem RefBox dan ROS 2. Node ini bertugas membaca, melakukan parsing, dan



meneruskan data perintah pertandingan ke sistem ROS 2 menggunakan mekanisme *publisher-subscriber*. Komunikasi antar node pada sistem ini memanfaatkan topik ROS 2 `/robot_command` dan `/robot_status` sehingga distribusi data dapat dilakukan secara *real-time*.

Pada lapisan Sistem Robot, Robot Command Node menerima data perintah dari Base Station melalui topik ROS 2 kemudian memproses instruksi tersebut menjadi target pergerakan robot. Selanjutnya, hasil pemrosesan diteruskan ke aktuator robot sehingga robot dapat bergerak menuju koordinat target sesuai kondisi pertandingan. Selain itu, Robot Status Node berfungsi untuk mempublikasikan status robot secara *real-time* berupa posisi robot, status aktuator, dan kondisi sistem melalui topik `/robot_status` untuk kebutuhan monitoring dan evaluasi komunikasi sistem.

Dibandingkan pendekatan monolitik konvensional, arsitektur berbasis node pada ROS 2 memberikan fleksibilitas yang lebih baik karena setiap fungsi sistem dipisahkan ke dalam node independen. Pendekatan ini mempermudah proses *debugging*, pengembangan fitur, dan integrasi sistem multi-*node* tanpa memengaruhi keseluruhan sistem. Selain itu, penggunaan *middleware Data Distribution Service (DDS)* pada ROS 2 memungkinkan komunikasi data berlangsung dengan latensi rendah dan reliabilitas tinggi sehingga sesuai untuk mendukung kebutuhan komunikasi *real-time* pada robot sepak bola beroda [9].



Gambar 4: Arsitektur Komunikasi Berbasis-Node ROS 2

3.2 Pengujian Perintah RefBox terhadap Sistem

Pengujian pada tahap ini dilakukan untuk mengetahui kesesuaian antara perintah yang dikirim oleh RefBox dengan respon yang dihasilkan oleh Base Station. Hasil pengujian ditampilkan pada Tabel I yang menunjukkan hubungan antara jenis perintah RefBox, karakter input yang diterima sistem, keluaran perintah pada Base Station, serta koordinat target yang dihasilkan.

Berdasarkan Tabel I, setiap perintah RefBox berhasil diterjemahkan oleh Base Station ke dalam bentuk perintah yang sesuai. Perintah `COMM_KICKOFF` dengan input karakter `K/k` menghasilkan perintah "START KICKOFF" dengan koordinat target `(0.0, 0.0)`. Perintah `COMM_PENALTY` dan `COMM_CORNER` masing-masing menghasilkan koordinat `(4.0, 0.0)` dan `(6.0, 4.0)`, yang menunjukkan bahwa sistem mampu mengarahkan robot ke posisi tertentu sesuai skenario permainan. Sementara itu, perintah `COMM_READY` menghasilkan kondisi Stop tanpa perubahan posisi (current position), yang menunjukkan bahwa sistem dapat mempertahankan posisi robot.

TABEL I. PENGUJIAN PERINTAH REFBOX TERHADAP SISTEM ROBOT



Perintah RefBox	Karakter	Perintah Base Station	Koordinat Target (x, y)
COMM_KICKOFF	K / k	REFEREE: START KICKOFF!	(0.0, 0.0)
COMM_READY	Stop	REFEREE: STOP!	Current Pos
COMM_PENALTY	P / p	REFEREE: START PENALTY!	(4.0, 0.0)
COMM_CORNER	C / c	REFEREE: START CORNER!	(6.0, 4.0)
COMM_GOALKICK	G / g	REFEREE: START GOALKICK!	(-5.0, 1.0)

Selain itu, tampilan output pada Base Station menunjukkan bahwa setiap perintah berhasil diproses dan divisualisasikan dengan benar dalam bentuk pesan teks. Hal ini menandakan bahwa proses parsing data dari RefBox ke Base Station berjalan sesuai dengan rancangan sistem. Secara keseluruhan, hasil pengujian ini menunjukkan bahwa sistem mampu menginterpretasikan perintah RefBox secara konsisten dan mengonversinya menjadi instruksi yang dapat digunakan oleh robot.

3.3 Analisis Latensi Komunikasi Sistem

Pengujian latensi dilakukan untuk mengukur waktu yang dibutuhkan dalam proses komunikasi dari RefBox ke Base Station serta dari Base Station ke Robot Node. Hasil pengujian ditampilkan pada Tabel II yang menunjukkan waktu *delay* pada setiap tahapan komunikasi.

TABEL II . PENGUJIAN LATENSI PADA KOMUNIKASI SISTEM

Percobaan Ke-	Perintah RefBox	Latensi RefBox ke BS (ms)	Latensi BS ke Robot Node (ms)	Total Latensi (ms)	Status
1	s (START)	120	85	205	Aman
2	S (STOP)	145	110	255	Aman
3	K (KICKOFF)	130	95	225	Aman
4	N (DROPBALL)	125	92	217	Aman
5	P (PENALTY)	138	102	240	Aman
6	C (CORNER)	142	108	250	Aman
7	F (FREEKICK)	118	80	198	Aman
8	G (GOALKICK)	122	88	210	Aman
9	L (PARK)	140	115	255	Aman
10	T (THROWIN)	136	91	227	Aman
Rata-rata	-	131.6	96.6	228.2	STABIL



Berdasarkan data pada Tabel II, waktu rata-rata komunikasi dari RefBox ke Base Station sebesar 131.6 ms, sedangkan dari Base Station ke Robot Node sebesar 96.6 ms. Total latensi sistem berada pada rata-rata 228.2 ms. Variasi nilai latensi pada setiap percobaan menunjukkan adanya perbedaan waktu transmisi yang dipengaruhi oleh kondisi jaringan dan proses pemrosesan data. Namun, secara umum seluruh percobaan menunjukkan status komunikasi yang stabil tanpa adanya kegagalan pengiriman data.

Hasil ini menunjukkan bahwa sistem komunikasi yang dibangun mampu menangani pengiriman data secara *real-time* dengan waktu respon yang relatif konsisten pada setiap percobaan. Data latensi ini menjadi indikator performa sistem dalam mendukung komunikasi antara RefBox, Base Station, dan robot dalam satu jaringan terintegrasi.

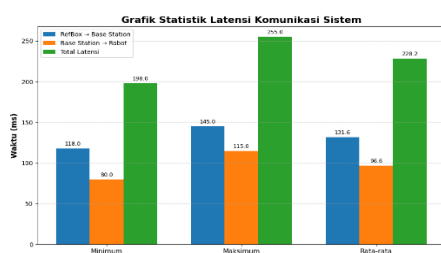
3.4 Analisis Stabilitas dan Performa Komunikasi Sistem

Pengujian dilakukan menggunakan 4 *node* ROS 2 aktif, yaitu *refbox_node*, *basestation_node*, *robot_node*, dan *status_node*. Komunikasi antara RefBox dan Base Station menggunakan jaringan LAN Ethernet untuk menjaga kestabilan transmisi data, sedangkan komunikasi antara Base Station dan Robot Node menggunakan jaringan *wireless* (WiFi). Pengujian dilakukan pada kondisi sistem aktif dengan pengiriman pesan ROS 2 bertipe string secara periodik menggunakan mekanisme *publisher-subscriber*. Seluruh pengujian dilakukan pada jaringan lokal tanpa trafik tambahan untuk meminimalkan interferensi komunikasi selama proses pengukuran latensi.

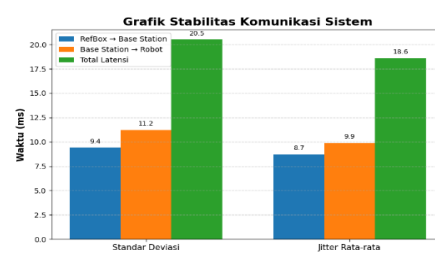
Untuk mengevaluasi performa komunikasi sistem, dilakukan analisis terhadap parameter latensi, *jitter* komunikasi, dan *packet delivery rate* (PDR). Pengujian dilakukan sebanyak 10 kali menggunakan berbagai perintah pertandingan dari RefBox seperti START, STOP, KICKOFF, PENALTY, CORNER, FREEKICK, dan THROWIN. Parameter yang diukur meliputi waktu komunikasi dari RefBox ke Base Station, waktu komunikasi dari Base Station ke Robot Node, serta total *delay* keseluruhan sistem komunikasi.

TABEL III . ANALISIS STATISTIK LATENSI KOMUNIKASI SISTEM

Parameter	RefBox ke BS (ms)	BS ke Robot (ms)	Total Latensi (ms)
Minimum	118	80	198
Maksimum	145	115	255
Rata-rata	131.6	96.6	228.2
Standar Deviasi	9.4	11.2	20.5
Jitter Rata-rata	8.7	9.9	18.6



Gambar 5: Grafik Statistik Latensi Komunikasi Sistem



Gambar 6: Grafik Stabilitas Komunikasi Sistem

Berdasarkan Tabel III dan Gambar 5, diperoleh rata-rata latensi komunikasi sebesar 131,6 ms dari RefBox ke Base Station dan 96,6 ms dari Base Station ke Robot Node, dengan total rata-rata latensi sistem sebesar 228,2 ms. Nilai minimum total latensi tercatat sebesar 198 ms, sedangkan nilai maksimum mencapai 255 ms. Hasil tersebut menunjukkan bahwa komunikasi antar *node* mampu berjalan secara konsisten tanpa terjadi lonjakan *delay* yang signifikan selama pengujian.



Variasi latensi dipengaruhi oleh kondisi jaringan, proses parsing data, serta komunikasi antar *node* ROS 2. Namun demikian, penggunaan *middleware* DDS pada ROS 2 mampu menjaga kestabilan distribusi data melalui mekanisme *publisher-subscriber* secara *real-time* [14].

Analisis stabilitas komunikasi pada Gambar 6 menunjukkan bahwa total standar deviasi latensi sebesar ±20,5 ms dengan nilai *jitter* rata-rata sebesar 18,6 ms. Nilai tersebut menunjukkan bahwa perubahan *delay* antar percobaan masih relatif kecil sehingga komunikasi data dapat berlangsung secara stabil dan sinkron.

Nilai standar deviasi dan *jitter* yang rendah menunjukkan bahwa sistem memiliki performa komunikasi yang cukup stabil untuk mendukung kebutuhan komunikasi *real-time* pada robot sepak bola beroda [14]. Secara keseluruhan, implementasi arsitektur modular berbasis *node* ROS 2 dengan *middleware* DDS mampu memberikan komunikasi yang responsif, stabil, dan reliabel pada sistem multi-*node* [9].

TABEL IV . HASIL PENGUJIAN PACKET DELIVERY RATE (PDR)

Parameter	Nilai
Jumlah Paket Dikirim	100
Jumlah Paket Diterima	100
Paket Hilang	0
Packet Delivery Rate	100%

```

python3 robot_node.py ID426
[INFO] [177987389.728096348] [refbox_node]: RefBox Send: START
[INFO] [177987391.728731368] [refbox_node]: RefBox Send: STOP
[INFO] [177987393.728733697] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987395.728736138] [refbox_node]: RefBox Send: PENALTY
[INFO] [177987397.729012834] [refbox_node]: RefBox Send: START
[INFO] [177987399.727707680] [refbox_node]: RefBox Send: STOP
[INFO] [177987401.728079133] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987403.728547746] [refbox_node]: RefBox Send: PENALTY
[INFO] [177987405.728094118] [refbox_node]: RefBox Send: START
[INFO] [177987407.728091892] [refbox_node]: RefBox Send: STOP
[INFO] [177987409.728744674] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987411.729146468] [refbox_node]: RefBox Send: START
[INFO] [177987413.728876667] [refbox_node]: RefBox Send: STOP
[INFO] [177987415.728857778] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987417.728981678] [refbox_node]: RefBox Send: PENALTY
[INFO] [177987419.728978793] [refbox_node]: RefBox Send: START
[INFO] [177987421.729119851] [refbox_node]: RefBox Send: STOP
[INFO] [177987423.729126736] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987425.728686863] [refbox_node]: RefBox Send: PENALTY
[INFO] [177987427.728685523] [refbox_node]: RefBox Send: START
[INFO] [177987429.728892319] [refbox_node]: RefBox Send: STOP
[INFO] [177987431.728802756] [refbox_node]: RefBox Send: KICKOFF
[INFO] [177987433.728724460] [refbox_node]: RefBox Send: PENALTY
[INFO] [177987435.728817980] [refbox_node]: RefBox Send: START
[INFO] [177987437.728947684] [refbox_node]: RefBox Send: STOP
    
```

Gambar 7: Terminal RefBox Node

```

python3 basestation_node.py ID426
[INFO] [177987413.730446350] [basestation_node]: Publish Robot Command: Execute START
[INFO] [177987415.729580809] [basestation_node]: Received RefBox: STOP
[INFO] [177987417.730420412] [basestation_node]: Publish Robot Command: Execute STOP
[INFO] [177987419.730477266] [basestation_node]: Publish Robot Command: Execute KICKOFF
[INFO] [177987421.729328532] [basestation_node]: Publish Robot Command: Execute PENALTY
[INFO] [177987423.729365595] [basestation_node]: Received RefBox: START
[INFO] [177987425.729483751] [basestation_node]: Publish Robot Command: Execute START
[INFO] [177987427.729165703] [basestation_node]: Received RefBox: STOP
[INFO] [177987429.72881652] [basestation_node]: Publish Robot Command: Execute STOP
[INFO] [177987431.729633172] [basestation_node]: Publish Robot Command: Execute KICKOFF
[INFO] [177987433.729339664] [basestation_node]: Received RefBox: PENALTY
[INFO] [177987435.729793237] [basestation_node]: Publish Robot Command: Execute PENALTY
[INFO] [177987437.729574335] [basestation_node]: Received RefBox: START
[INFO] [177987439.728889224] [basestation_node]: Publish Robot Command: Execute START
[INFO] [177987441.729412687] [basestation_node]: Received RefBox: STOP
[INFO] [177987443.730563393] [basestation_node]: Publish Robot Command: Execute STOP
[INFO] [177987445.729433968] [basestation_node]: Received RefBox: KICKOFF
[INFO] [177987447.731317555] [basestation_node]: Publish Robot Command: Execute KICKOFF
[INFO] [177987449.729576255] [basestation_node]: Received RefBox: PENALTY
[INFO] [177987451.729678980] [basestation_node]: Publish Robot Command: Execute PENALTY
[INFO] [177987453.729552548] [basestation_node]: Received RefBox: START
[INFO] [177987455.730747462] [basestation_node]: Publish Robot Command: Execute START
    
```

Gambar 8: Terminal Base Station Node

```

python3 robot_node.py ID426
[INFO] [177987389.730711583] [robot_node]: Robot Execute: Execute START
[INFO] [177987391.730996666] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987393.731093358] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987395.731682781] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987397.731240969] [robot_node]: Robot Execute: Execute START
[INFO] [177987399.731478746] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987401.730899582] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987403.730477239] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987405.731458454] [robot_node]: Robot Execute: Execute START
[INFO] [177987407.731291585] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987409.731362187] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987411.731291848] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987413.731496037] [robot_node]: Robot Execute: Execute START
[INFO] [177987415.731466256] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987417.731423781] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987419.731318959] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987421.731122379] [robot_node]: Robot Execute: Execute START
[INFO] [177987423.731339888] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987425.731225981] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987427.731403529] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987429.731281189] [robot_node]: Robot Execute: Execute START
[INFO] [177987431.731466499] [robot_node]: Robot Execute: Execute STOP
[INFO] [177987433.731686209] [robot_node]: Robot Execute: Execute KICKOFF
[INFO] [177987435.731385138] [robot_node]: Robot Execute: Execute PENALTY
[INFO] [177987437.731678748] [robot_node]: Robot Execute: Execute START
    
```

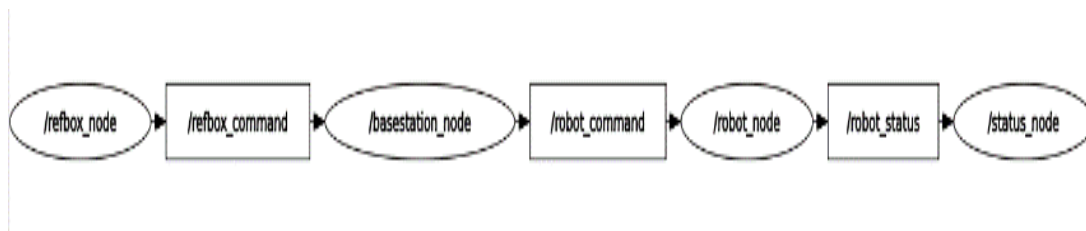
Gambar 9: Terminal Robot Node

```

python3 status_node.py ID426
[INFO] [177987389.732672813] [status_node]: System Monitor: Movement Success
[INFO] [177987391.731942828] [status_node]: System Monitor: Movement Success
[INFO] [177987393.732624678] [status_node]: System Monitor: Movement Success
[INFO] [177987395.733597899] [status_node]: System Monitor: Movement Success
[INFO] [177987397.734111717] [status_node]: System Monitor: Movement Success
[INFO] [177987399.733353379] [status_node]: System Monitor: Movement Success
[INFO] [177987401.732703497] [status_node]: System Monitor: Movement Success
[INFO] [177987403.732363087] [status_node]: System Monitor: Movement Success
[INFO] [177987405.733556446] [status_node]: System Monitor: Movement Success
[INFO] [177987407.733193945] [status_node]: System Monitor: Movement Success
[INFO] [177987409.733227273] [status_node]: System Monitor: Movement Success
[INFO] [177987411.732657992] [status_node]: System Monitor: Movement Success
[INFO] [177987413.733396643] [status_node]: System Monitor: Movement Success
[INFO] [177987415.733482331] [status_node]: System Monitor: Movement Success
[INFO] [177987417.732979783] [status_node]: System Monitor: Movement Success
[INFO] [177987419.733236888] [status_node]: System Monitor: Movement Success
[INFO] [177987421.732989597] [status_node]: System Monitor: Movement Success
[INFO] [177987423.731617212] [status_node]: System Monitor: Movement Success
[INFO] [177987425.733018266] [status_node]: System Monitor: Movement Success
[INFO] [177987427.733181893] [status_node]: System Monitor: Movement Success
[INFO] [177987429.733184919] [status_node]: System Monitor: Movement Success
[INFO] [177987431.733455522] [status_node]: System Monitor: Movement Success
[INFO] [177987433.733538118] [status_node]: System Monitor: Movement Success
[INFO] [177987435.732967862] [status_node]: System Monitor: Movement Success
[INFO] [177987437.733767133] [status_node]: System Monitor: Movement Success
    
```

Gambar 10: Terminal Status Node





Gambar 11: Visualisasi Komunikasi Node Menggunakan rqt_graph ROS 2

Selain pengujian latensi, evaluasi reliabilitas komunikasi juga dilakukan menggunakan parameter *Packet Delivery Rate* (PDR). Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan pengiriman data antar *node* ROS 2 selama sistem beroperasi secara *real-time*.

Berdasarkan hasil pengujian pada Tabel IV, seluruh paket data yang dikirim dari RefBox menuju Base Station dan diteruskan ke Robot Node berhasil diterima dengan baik tanpa kehilangan paket data. Dari total 100 paket yang dikirimkan, seluruh paket berhasil diterima sehingga diperoleh nilai *Packet Delivery Rate* sebesar 100%. Hasil tersebut menunjukkan bahwa sistem komunikasi memiliki reliabilitas yang baik dalam proses distribusi data antar node.

Visualisasi terminal komunikasi pada Gambar 7 hingga Gambar 10 menunjukkan bahwa setiap node ROS 2 berhasil menjalankan fungsi komunikasi sesuai rancangan sistem. Pada Gambar 7, *refbox_node* melakukan publish perintah pertandingan seperti START, STOP, KICKOFF, dan PENALTY secara berkala melalui topik */refbox_command*. Selanjutnya, Gambar 8 menunjukkan bahwa *basestation_node* berhasil menerima data dari RefBox dan meneruskannya ke Robot Node melalui topik */robot_command*.

Pada sisi robot, Gambar 9 menunjukkan bahwa *robot_node* berhasil menerima dan mengeksekusi perintah yang dikirim oleh Base Station. Setelah proses eksekusi selesai, robot mengirimkan status keberhasilan pergerakan melalui topik */robot_status*. Informasi tersebut kemudian diterima oleh *status_node* sebagaimana ditunjukkan pada Gambar 10 untuk kebutuhan monitoring sistem secara *real-time*.

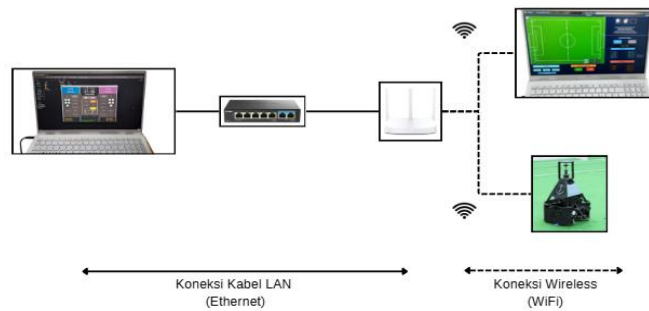
Selain itu, hubungan komunikasi antar node divisualisasikan menggunakan *rqt_graph* pada Gambar 11. Visualisasi tersebut menunjukkan bahwa komunikasi *publisher-subscriber* ROS 2 berjalan sesuai dengan arsitektur sistem yang dirancang, di mana *refbox_node* terhubung dengan *basestation_node* melalui topik */refbox_command*, kemudian diteruskan menuju *robot_node* melalui topik */robot_command*, dan akhirnya dipublikasikan kembali ke *status_node* melalui topik */robot_status*.

Hasil pengujian ini menunjukkan bahwa mekanisme komunikasi *publisher-subscriber* pada ROS 2 dengan middleware DDS mampu menjaga konsistensi pengiriman data secara stabil pada sistem multi-*node*. Tidak ditemukannya *packet loss* selama pengujian menunjukkan bahwa sistem komunikasi yang dikembangkan memiliki reliabilitas yang baik untuk mendukung kebutuhan komunikasi *real-time* pada robot sepak bola beroda.

TABEL V . EVALUASI PERFORMA REAL-TIME SISTEM

Parameter	Nilai
Rata-rata Total Delay	228.2 ms
Kategori Sistem	Responsif untuk komunikasi <i>real-time</i>
Status Sistem	Memenuhi kebutuhan <i>real-time</i>





Gambar 12: Topologi Pengujian

Berdasarkan Tabel V, rata-rata total delay sistem sebesar 228,2 ms menunjukkan bahwa komunikasi antar *node* masih mampu berlangsung secara responsif untuk mendukung kebutuhan komunikasi *real-time* pada robot sepak bola beroda. Nilai *delay* tersebut menunjukkan bahwa sistem masih dapat merespons perintah pertandingan secara cepat dan stabil selama proses pengujian berlangsung. Hasil tersebut menunjukkan bahwa sistem mampu memberikan respon komunikasi yang cukup cepat untuk mendukung koordinasi robot selama pertandingan.

Topologi pengujian pada Gambar 12 menunjukkan bahwa sistem terdiri atas RefBox, switch, *access point*, Base Station, dan Robot Node yang terhubung dalam satu jaringan lokal. Komunikasi antara RefBox dan Base Station menggunakan koneksi kabel LAN (Ethernet), sedangkan komunikasi dari Base Station menuju Robot Node menggunakan jaringan wireless (WiFi).

Hasil pengujian menunjukkan bahwa mekanisme *publisher-subscriber* ROS 2 dengan middleware DDS mampu menjaga komunikasi data secara stabil dan sinkron pada sistem multi-*node*. Selain itu, tidak ditemukannya *packet loss* selama pengujian menunjukkan bahwa sistem memiliki reliabilitas komunikasi yang baik untuk mendukung kebutuhan komunikasi *real-time* pada robot sepak bola beroda. Secara keseluruhan, implementasi arsitektur modular berbasis node ROS 2 berhasil memberikan performa komunikasi yang responsif, stabil, dan reliabel pada sistem robot sepak bola multi-*node*.

4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis sistem komunikasi antara RefBox, Base Station, dan robot menggunakan arsitektur modular berbasis node ROS 2, dapat disimpulkan bahwa sistem yang dikembangkan mampu bekerja secara *real-time*, modular, dan stabil. Sistem berhasil menginterpretasikan seluruh perintah dari RefBox dengan akurat dan mengonversinya menjadi instruksi yang sesuai untuk robot, termasuk penentuan koordinat target. Selain itu, komunikasi antar node berjalan dengan baik melalui mekanisme *publisher-subscriber* ROS 2, sehingga mendukung pemisahan fungsi sistem secara independen. Dari hasil pengujian latensi, diperoleh rata-rata total *delay* sebesar 228,2 ms, yang menunjukkan bahwa sistem masih berada dalam kategori respons cepat dan layak digunakan untuk aplikasi robot sepak bola yang membutuhkan respon *real-time*. Seluruh percobaan menunjukkan status komunikasi stabil tanpa kehilangan data. Dengan demikian, arsitektur modular berbasis node ROS 2 yang diterapkan terbukti mampu meningkatkan modularitas, skalabilitas, serta keandalan komunikasi sistem, sekaligus mendukung visualisasi data secara konsisten.

5. UCAPAN TERIMAKASIH

Penulis menyampaikan terima kasih kepada Politeknik Perkapalan Negeri Surabaya (PPNS), khususnya Jurusan Teknik Kelistrikan Kapal, Program Studi Teknik Otomasi, atas penyediaan fasilitas serta lingkungan akademik yang mendukung selama proses penelitian berlangsung. Penulis juga mengucapkan terima kasih kepada dosen pembimbing atas bimbingan, arahan, serta saran yang konstruktif sehingga penelitian ini dapat tersusun dengan baik. Apresiasi turut diberikan kepada tim Gerhana Dewaruci KRSBI-B PPNS atas dukungan, kerja sama, dan kontribusi yang diberikan dalam proses pengembangan, pengujian, hingga implementasi



sistem robot pada penelitian ini. Selain itu, penulis mengucapkan terima kasih kepada seluruh pihak yang telah berperan serta dalam membantu penyelesaian penelitian ini.

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] F. Tjoanapessy *et al.*, "Aplikasi Base Station Untuk Robot Sepak Bola Beroda," vol. 14, no. 3, pp. 285–290, 2019.
- [3] H. Yoga and N. Anwar, "Sistem Komunikasi Otomatis Robot Sepak Bola Beroda dengan Komunikasi Real-Time melalui Infrastruktur Jaringan Base Station," vol. 12, no. 1, pp. 1–8, 2024.
- [4] A. Jalil, "ROBOT OPERATING SYSTEM (ROS) DAN GAZEBO SEBAGAI MEDIA PEMBELAJARAN ROBOT INTERAKTIF," vol. 10, pp. 284–289, 2018.
- [5] J. Triyono, E. Abram, A. Sitepu, and K. I. Dhamayanti, "PERANCANGAN KOMUNIKASI DATA ANTAR SISTEM INFORMASI MENGGUNAKAN TRANSMISI KONTROL PROTOKOL (TCP) PADA IOT SMART FARMING," vol. 14, no. 1.
- [6] S. S. Alam, R. Hidayat, and M. Abdi, "Otomatisasi Penyemprotan Polyester Menggunakan Kawasaki Cobot dengan Human Machine Interface (HMI) Berbasis Web," vol. 16, no. 2, pp. 119–129, 2024.
- [7] M. I. Yamin, S. Kuswadi, and S. Sukaridhoto, "Real Performance Evaluation On MQTT and COAP Protocol in Ubiquitous Network Robot Platform (UNR-PF) for Disaster Multi-robot Communication," vol. 6, no. 2, pp. 369–385, 2018.
- [8] E. A. Habibie, M. Muskhir, J. Prof, H. Air, T. Padang, and S. Barat, "Perancangan Monitoring Robot Kiper Melalui Acces Point Sebagai Media Kendali Robot," vol. 2, no. 1, pp. 113–118, 2021.
- [9] 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.
- [10] O. Robotics, "ROS 2 Documentation."
- [11] T. Ardhiansyah, I. Syarifuddin, M. R. Naufal, Y. Pramono, O. T. Hartatik, and I. Mustaqim, "Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station," pp. 82–86, 2017.
- [12] K. Edy *et al.*, "ROBOT SEPAK BOLA BERODA DENGAN," vol. 9, no. 4, pp. 17–23, 2022.
- [13] S. Lee, H. Park, J. Chae, and K. Park, "Probabilistic Latency Analysis of the Data Distribution Service in ROS 2," pp. 1–12.
- [14] A. Vehicle, "Performance Evaluation of ROS2-DDS middleware implementations facilitating Cooperative Driving in," vol. 9183, 2024.

