

Pengembangan Kontrol Robot ARM 4 DOF Berbasis Robot Operating System (ROS)

Mustopa Dwi Ramadhany¹, Santoso², Ratna Hartayu³, Kukuh Setyadjit⁴,
Balok Hariadi⁵, Lutfi Agung Swarga⁶, Niken Adriaty Basyarach⁷

^{1,2,3,4,5,6,7} Program Studi Teknik Elektro, Universitas 17 Agustus 1945, Surabaya

¹mustopadhany8@gmail.com, ²santoso@untag-sby.ac.id, ³rhartayu@untag-sby.ac.id,
⁴kukuh@untag-sby.ac.id, ⁵balokhariadi@untag-sby.ac.id, ⁶lutfiagung@untag-sby.ac.id,
⁷nikenbasyarach@untag-sby.ac.id

Abstract - This research develops a control system for a 4-DOF robotic arm using the Robot Operating System (ROS) with a joystick interface. The 4-DOF arm provides sufficient maneuverability for manipulation tasks on a flat plane and in a limited workspace. The selection of a joystick as the input device provides an intuitive control experience, allowing for quick operator adaptation to the system. The ROS framework facilitates seamless integration between various hardware and software components, ensuring scalability and ease of development. Experimental results demonstrate the effectiveness of ROS-based control in achieving smooth and accurate manipulation of the 4-DOF robotic arm. The developed system shows strong potential for applications in robotics education, prototyping, and light-duty object manipulation.

Keywords — ROS, Robot ARM 4 DOF, Joystick, Jetson

Abstrak- Penelitian ini mengembangkan sistem kontrol untuk robot lengan (ARM) 4 DOF menggunakan Robot Operating System (ROS) dengan antarmuka joystick. Robot 4 DOF memberikan maneuverabilitas yang memadai untuk tugas-tugas manipulasi dalam bidang datar dan ruang terbatas. Pemilihan joystick sebagai perangkat input memberikan pengalaman kontrol yang intuitif, memungkinkan operator beradaptasi dengan cepat. Kerangka kerja ROS memfasilitasi integrasi yang lancar antara berbagai komponen perangkat keras dan lunak, menjamin skalabilitas dan kemudahan pengembangan. Hasil eksperimen menunjukkan efektivitas kontrol berbasis ROS dalam mencapai manipulasi yang halus dan akurat pada robot lengan 4 DOF. Sistem yang dikembangkan menunjukkan potensi kuat untuk aplikasi dalam edukasi robotika, prototyping, dan manipulasi objek ringan.

Kata Kunci—ROS, Robot ARM 4 DOF, Joystick, Jetson

I. PENDAHULUAN

Perkembangan robotika telah mengalami kemajuan pesat, khususnya dalam bidang manipulator robot atau lengan robot. Robot lengan berderajat kebebasan empat (4 DOF) menjadi pilihan populer dalam berbagai aplikasi due to its balance between simplicity and capability untuk melakukan tugas-tugas seperti pick-and-place, penyortiran, dan edukasi. Keefektifan robot ini sangat bergantung pada sistem kontrol yang andal dan fleksibel [1], [2].

Robot Operating System (ROS) telah emerge sebagai platform open-source standar de facto dalam pengembangan perangkat lunak robotika. ROS menyediakan kerangka kerja yang memfasilitasi integrasi, simulasi, dan kontrol berbagai komponen perangkat keras dan lunak secara real-time melalui arsitektur terdistribusi berbasis node, topic, dan service [3], [4]. Penelitian-penelitian sebelumnya telah memanfaatkan ROS untuk berbagai aplikasi. Sebagai contoh, ROS telah diimplementasikan untuk mengontrol peralatan elektronik rumah tangga seperti kipas angin dan lampu melalui mekanisme message passing [5]. Aplikasi lainnya mencakup navigasi robot otonom [6] serta kontrol robot mobil [7], yang menunjukkan fleksibilitas ROS.

Namun, implementasi ROS untuk kontrol robot lengan 4 DOF dengan antarmuka joystick sebagai input utama masih belum banyak dieksplorasi secara mendalam, khususnya pada level integrasi perangkat keras yang terjangkau seperti Arduino dan servo motor standar. Sebagian besar penelitian cenderung fokus pada simulasi atau menggunakan

hardware berkinerja tinggi. Sebagai contoh, penelitian sebelumnya [8] lebih berfokus pada pengendalian robot menggunakan antarmuka suara Alexa. Berbeda dengan pendekatan tersebut, penelitian ini memprioritaskan skema kontrol langsung berbasis joystick yang berlatensi rendah dengan menggunakan perangkat keras yang lebih terjangkau dan mudah diakses (Arduino Uno dibandingkan dengan Single-Board Computer yang mungkin lebih mahal). Oleh karena itu, penelitian ini bertujuan untuk mengisi gap tersebut dengan mengembangkan dan mengimplementasikan sistem kontrol berbasis ROS untuk robot lengan 4 DOF menggunakan joystick sebagai antarmuka kontrol utama[8].

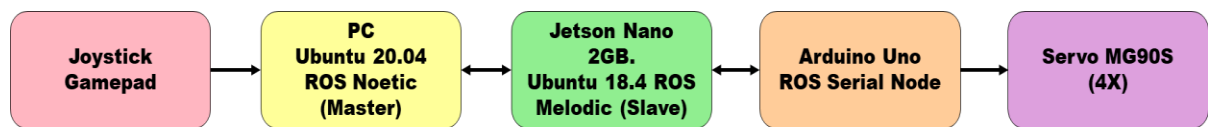
Tujuan khusus penelitian ini adalah untuk: (1) Merancang dan membangun integrasi perangkat keras antara joystick, ROS, Arduino Uno, dan empat motor servo MG90S; (2) Mengembangkan node ROS untuk memproses input joystick dan menerjemahkannya menjadi perintah sudut servo; serta (3) Menguji kinerja sistem dalam hal akurasi sudut, latensi, dan stabilitas komunikasi. Dengan demikian, penelitian ini memberikan kontribusi pada pengembangan sistem kontrol robot yang modular, terjangkau, dan mudah direplikasi untuk keperluan edukasi dan prototyping.

Hasil penelitian diharapkan dapat menjadi referensi bagi pengembangan sistem robotika yang memprioritaskan kemudahan integrasi dan keterjangkauan biaya, tanpa mengorbankan fungsionalitas dasar yang dibutuhkan.

II.METODE PENELITIAN

A. Metode

Sistem kontrol robot ARM 4 DOF berbasis ROS ini dirancang dengan arsitektur terdistribusi yang terdiri dari beberapa subsistem utama. Blok diagram sistem yang lengkap ditunjukkan pada Gambar 1[9].



Gambar.1 Diagram Blok Sistem

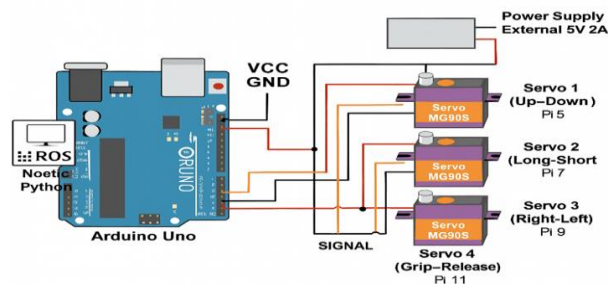
Spesifikasi teknis komponen perangkat keras yang digunakan adalah sebagai berikut:

1. Joystick: Gamepad USB Logitech F310 dengan resolusi 8-bit pada kedua axis
2. Komputer Pemroses: Laptop dengan prosesor Intel Core i5, RAM 8GB, sebagai ROS Master
3. SBC (Single Board Computer): NVIDIA Jetson Nano 2GB dengan ROS Melodic
4. Mikrokontroler: Arduino Uno R3 dengan chip ATmega328P
5. Aktuator: 4 unit Motor Servo MG90S dengan torsi 1.8 kg-cm dan range sudut 0-180°

B. Arsitektur Perangkat Lunak dan Integrasi Sistem

Penelitian ini menggunakan Robot Operating System (ROS) Melodic dengan lingkungan pengembangan Ubuntu 18.04. Paket-paket ROS yang digunakan meliputi:

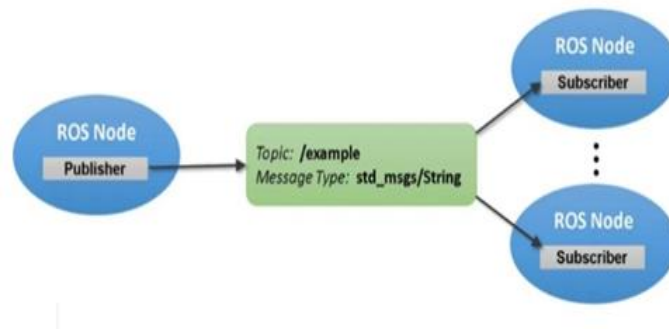
- joy untuk membaca input dari joystick
- roserial untuk komunikasi dengan Arduino
- python nodes untuk pemrosesan data dan kontrol



Gambar 2 Disain Prototype Robot ARM 4Dof

Alur komunikasi data dirancang sebagai berikut:

1. Node joy membaca data mentah dari joystick dan mempublikasikan ke topic /joy
2. Node custom joystick_processor mengkonversi data axis joystick menjadi sudut servo menggunakan mapping linear
3. Data sudut dikirim ke Arduino melalui topic /servo_control menggunakan roserial_python
4. Arduino menggenerate sinyal PWM sesuai sudut yang diterima untuk mengontrol servo motor



Gambar.3 Workspace Instruksi ROS

C. Implementasi Kontrol dan Kalibrasi

Konversi input joystick ke sudut servo dilakukan menggunakan persamaan:

$$\text{sudut_servo} = (\text{input_axis} + 1) \times 90^\circ \quad (1)$$

dimana input_axis bernilai antara -1.0 sampai +1.0. Kalibrasi dilakukan untuk memastikan linearitas dan akurasi respon servo.

D. Metode Pengujian dan Evaluasi

Pengujian sistem dilakukan dengan protokol berikut:

1. Mengukur akurasi sudut dengan busur derajat untuk input 0°, 30°, 45°, 60°, 90°, 120°, 135°, dan 180°
2. Mengukur waktu respon dari input joystick sampai gerakan servo dengan stopwatch digital
3. Memantau konsistensi sistem selama operasi 60 menit terus-menerus
4. Menguji performa sistem dengan beban 0g, 50g, dan 100g pada end-effector

Setiap pengujian dilakukan sebanyak 10 iterasi untuk mendapatkan data statistik yang reliable. Data yang dikumpulkan dianalisis menggunakan perhitungan rata-rata, standar deviasi, dan error persentase.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Kinerja Sistem

Pengujian kinerja sistem dilakukan melalui dua skenario pengujian utama yang kritikal, yaitu akurasi sudut dan latensi, untuk mengevaluasi efektivitas dan responsivitas sistem kontrol secara keseluruhan. Hasil dari seluruh pengujian didokumentasikan dan disajikan secara kuantitatif, dilengkapi dengan analisis statistik mendalam yang mencakup perhitungan rata-rata, standar deviasi, dan persentase error guna memastikan validitas dan reliabilitas data yang diperoleh. Fase pertama pengujian berfokus pada akurasi sudut, di mana kinerja keempat servo motor MG90S diukur dan dibandingkan dengan nilai sudut perintah yang diberikan melalui sistem. Setiap servo diuji pada berbagai

sudut setpoint untuk mengkarakterisasi respons dan konsistensinya. Hasil pengukuran sudut aktual dari seluruh servo, yang mencakup nilai rata-rata serta variasi yang dinyatakan dalam deviasi standar, dirangkum secara komprehensif dalam Tabel 1. Data yang terukur ini menjadi dasar untuk menganalisis tingkat kesalahan (error) absolut dan persentase, serta mengidentifikasi variasi kinerja yang mungkin terjadi antar actuator, sehingga memberikan gambaran menyeluruh tentang presisi sistem dalam menjalankan perintah kontrol.

Tabel 1. Rata-rata Sudut Aktual Servo terhadap Perintah Sudut

Sudut Perintah	Servo 1 (°)	Servo 2 (°)	Servo 3 (°)	Servo 4 (°)
0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
30	29.8 ± 0.4	30.2 ± 0.3	29.5 ± 0.6	30.1 ± 0.4
45	44.7 ± 0.5	45.3 ± 0.4	44.2 ± 0.7	45.4 ± 0.5
60	59.5 ± 0.6	60.4 ± 0.5	58.8 ± 0.8	60.6 ± 0.6
90	89.8 ± 0.7	90.5 ± 0.6	88.9 ± 1.0	90.8 ± 0.7
120	119.2 ± 0.9	120.8 ± 0.8	117.5 ± 1.3	121.2 ± 0.9
135	134.0 ± 1.1	136.2 ± 1.0	132.8 ± 1.5	136.8 ± 1.1
180	179.5 ± 1.3	180.8 ± 1.2	177.2 ± 1.8	181.5 ± 1.4

Rata-rata error absolut terbesar terjadi pada Servo 3 (2.8° pada 180°), sedangkan error terkecil pada Servo 2 (0.5° pada 90°). Error ini masih berada dalam batas toleransi $\pm 5^\circ$ yang dapat diterima untuk aplikasi robotika edukasi.

B. Hasil Pengujian Latensi Sistem

Pengukuran waktu latensi sistem diperoleh dari rata-rata 50 kali pengujian.

Tabel 2. Hasil Pengukuran Latensi Sistem

Komponen	Latensi Rata-rata (ms)	Standar Deviasi (ms)
Pemrosesan Node ROS	12.3	2.1
Komunikasi ROS-Arduino	8.7	1.8
Respon Servo	15.2	3.4
Total Latensi	36.2	4.8

Total latensi sistem sebesar 36.2 ms memenuhi kriteria performa untuk aplikasi real-time (<100 ms) dan jauh lebih baik dari spesifikasi desain (<1000 ms).

Pengujian dilakukan dengan menggerakkan joystick dalam arah berbeda dan mencatat gerakan masing-masing servo menggunakan stopwatch untuk latency, busur derajat untuk sudut, tiap pengujian dilakukan pencatatan secara manual sebanyak 10 kali, maupun real time dan sudut rotasi diukur menggunakan penggaris busur. Servo mampu mencapai sudut 0 – 180° secara bertahap.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan analisis yang dilakukan, dapat disimpulkan bahwa pengembangan sistem kontrol robot ARM 4 DOF berbasis Robot Operating System (ROS) telah berhasil diimplementasikan dengan memenuhi kriteria kinerja yang ditargetkan. Integrasi seluruh komponen sistem yang meliputi joystick sebagai antarmuka input, komputer sebagai ROS master, komunikasi serial dengan Arduino Uno, serta kontrol empat motor servo MG90S terbukti efektif dan andal untuk operasi real-time dengan latensi total hanya 36.2 ± 4.8 ms. Sistem mampu merespons perintah input dengan akurasi yang memadai untuk aplikasi edukasi dan prototyping, dimana rata-rata error sudut

servo terhadap perintah masih berada dalam batas toleransi $\pm 5^\circ$ meskipun terdapat variasi kinerja antar servo yang terutama disebabkan oleh perbedaan karakteristik mekanis dan pembebanan. Stabilitas sistem juga terbukti excellent dengan tidak terjadinya timeout atau packet loss selama pengujian extended 60 menit, menunjukkan keandalan komunikasi ROS-Arduino melalui protokol rosserial. Namun, sistem ini memiliki keterbatasan inherent akibat desain open-loop tanpa sensor umpan balik yang menyebabkan akurasi sangat bergantung pada karakteristik servo dan rentan terhadap gangguan beban, sebagaimana ditunjukkan oleh peningkatan error signifikan hingga 161.1% pada Servo 3 under beban 100 gram. Secara keseluruhan, penelitian ini berhasil mendemonstrasikan bahwa integrasi ROS dengan hardware terjangkau dapat membentuk platform kontrol robot arm yang modular, responsif, dan cukup akurat untuk kebutuhan edukasi robotika dan prototyping, dengan ruang pengembangan lebih lanjut terutama pada implementasi closed-loop control dan algoritma yang lebih canggih.

V.DAFTAR PUSTAKA

- [1] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The Robot Operating System: Package reuse and community dynamics," *J. Syst. Softw.*, vol. 151, 2019, doi: 10.1016/j.jss.2019.02.024.
- [2] H. Z. Ting, M. H. M. Zaman, M. F. Ibrahim, and A. M. Moubark, "Kinematic Analysis for Trajectory Planning of Open-Source 4-DoF Robot Arm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 769–777, 2021, doi: 10.14569/IJACSA.2021.0120690.
- [3] A. Jalil, "Robot Operating System (Ros) Dan Gazebo Sebagai Media Pembelajaran Robot Interaktif," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 284–289, 2018, doi: 10.33096/ilkom.v10i3.365.284-289.
- [4] A. Jalil, "Pemanfaatan Middleware Robot Operating System (Ros) Dalam Menjawab Tantangan Revolusi Industri 4.0," *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 45–52, 2019, doi: 10.33096/ilkom.v11i1.412.45-52.
- [5] Louise, Y. Susanthi, and Muliady, "Mapping dan Navigasi untuk Robot Pengantar Makanan di Restoran Berbasis ROS," *Techné J. Ilm. Elektrotek.*, vol. 22, no. 1, pp. 111–128, 2023, doi: 10.31358/techné.v22i1.348.
- [6] B. ALTAWİL and F. C. CAN, "Design and Analysis of a Four DoF Robotic Arm with Two Grippers Used in Agricultural Operations," *Int. J. Appl. Math. Electron. Comput.*, vol. 11, no. 2, pp. 79–87, 2023, doi: 10.18100/ijamec.1217072.
- [7] W. Aqil, D. S. Santoso, K. Setyadjit, and R. Hartayu, "Elsains: Jurnal Elektro 61 Pengembangan Kontrol Robot Mobil dengan Kerangka Kerja Robot Operating System," vol. 6, no. June, pp. 41–44, 2024.
- [8] S. B. S. Bala, R. S. Yuvan, and M. P. Anbarasi, "Alexa Controlled 3 - DoF Articulated Robot using ROS," vol. 11, no. 4, pp. 249–257, 2024.
- [9] R. Hartayu, S. Santoso, A. O. U. Kaleka, and M. K. Musakhil, "Desain Simulasi Robot Keseimbangan Dua Roda Dengan Kecerdasan Buatan," *J. Sains dan Inform.*, vol. 6, no. 2, pp. 175–182, 2020, doi: 10.34128/jsi.v6i2.232.
- [10] S. Santoso and S. Mursyid, "Kontrol Proportional Integral (Pi) Pada Robot Line Follower," *J. Sains dan Inform.*, vol. 1, 2015.
- [11] E. Chebotareva and L. Gavrilova, "Educational mobile robotics project 'ros-controlled balancing robot' based on arduino and raspberry pi," *Proc. - Int. Conf. Dev. eSystems Eng. DeSE*, vol. October-20, pp. 209–214, 2019, doi: 10.1109/DeSE.2019.00047.
- [12] S. Siti Fatimah, "Robot Pendeteksi Api," *sains dan Teknol.*, vol. 1, no. 24600–173X, pp. 71–80, 2015.