Kendali Gerakan Robot Openmanipulator-X Menggunakan MoveIt RViz dan GUI Control Berbasis Robot Operating System 2

Charles Harjito[1], Romy Budhi Widodo[2]
Human-Machine Interaction Research Center-Teknik Informatika, Universitas Ma Chung
e-mail: 312310011@student.machung.ac.id[1], romy.budhi@machung.ac.id[2]

Kata Kunci:

OpenManipulator-X
ROS 2
MoveIt RViz
GUI Control
Gazebo

ABSTRAK

Perkembangan teknologi otomasi pada era Industri 4.0 menggunakan robotika sebagai penggerak utama pada lingkungan manufaktur cerdas. Penelitian ini bertujuan membangun simulasi interaktif dan membandingkan efektivitas serta karakteristik dua metode kendali, yaitu Movelt RViz dan GUI Control, dalam menggerakkan robot OpenManipulator-X berbasis Robot Operating System 2 (ROS 2) Jazzy Jalisco. Metode yang digunakan bersifat eksperimental kuantitatif, meliputi persiapan lingkungan kerja pada sistem operasi Ubuntu 24.04 LTS, pengujian gerak *single-point* dan multi-point pada simulator Gazebo, serta pengujian pada robot fisik secara real-time dengan tiga tingkat kecepatan pada Movelt RViz. Hasil penelitian menunjukkan seluruh pengujian gerak pada simulator maupun robot fisik berhasil dijalankan tanpa kegagalan. Pada kecepatan default, GUI Control memperoleh waktu eksekusi rata-rata lebih cepat dibandingkan Movelt RViz, namun Movelt RViz pada kecepatan tinggi mampu menurunkan waktu eksekusi hingga 71% pada gerak *single-point* dan 69% pada gerak multi-point. GUI Control juga terbukti menjadi satu-satunya metode yang mampu melakukan ekspor data posisi joint ke berkas CSV. Kedua metode terbukti saling melengkapi sesuai kebutuhan pengguna.

Keyword:

OpenManipulator-X
ROS 2
MoveIt RViz
GUI Control
Gazebo

ABSTRACT

The development of automation technology in the Industry 4.0 era has utilized robotics as a key driver in smart manufacturing environments. This study aims to build an interactive simulation and to compare the effectiveness and characteristics of two control methods, Movelt RViz and GUI Control, in the OpenManipulator-X robot arm based on Robot Operating System 2 (ROS 2) Jazzy Jalisco. A quantitative experimental method was used, covering environment setup on Ubuntu 24.04 LTS, single-point and multi-point motion testing on the Gazebo simulator, and real-time testing on the physical robot at three speed levels on Movelt RViz. The results show that all motion tests on both the simulator and the physical robot were completed successfully without failure. At default speed, GUI Control achieved a faster average execution time than Movelt RViz, yet Movelt RViz at high speed reduced execution time by up to 71% for single-point motion and 69% for multi-point motion. GUI Control was also found to be the only method capable of exporting joint position data to a CSV file. Both methods proved complementary depending on user needs.

Copyright © SENASIF 2026
ISSN: 2598-0076

I. PENDAHULUAN

Perkembangan teknologi otomasi di era Industri 4.0 membuat konsep pabrik cerdas sebagai hal utama di sektor manufaktur, yang menggabungkan teknologi otomasi dengan sistem fisik-siber untuk membuat rangkaian produksi yang lebih efisien, konsisten, dan meminimalisir risiko *human error* [1]. Sebagai penggerak utama pada lingkungan manufaktur, robotika memegang peranan yang tidak dapat dilepas dari proses konstruksi, desain, hingga perakitan sistem produksi [2]. Robot didefinisikan sebagai seperangkat alat mekanik yang dapat menjalankan tugas fisik, baik melalui pengawasan manusia maupun melalui program kecerdasan buatan yang sudah ditentukan sebelumnya [3]. Perkembangan ini mendorong perluasan bagi fungsi robot ke berbagai bidang di luar industri, seperti pelayanan, edukasi, kesehatan, dan hiburan, hal ini sejalan dengan tren penggantian pekerjaan manusia untuk tugas yang bersifat rutin dan mekanis [4].

Penggunaan robot *arm* untuk memindahkan objek secara langsung tanpa adanya pengujian awal berisiko sangat tinggi dan dapat menimbulkan kesalahan lintasan maupun kegagalan sistem perangkat lunak, yang dapat menyebabkan tabrakan antar-komponen dan kerugian biaya perbaikan yang besar [5]. Oleh karena itu, simulasi dan pengujian bertahap sebelum menggunakan robot fisik adalah langkah krusial untuk meminimalkan risiko tersebut. Penelitian terdahulu terkait kendali gerak robot OpenManipulator-X telah dilakukan menggunakan platform MATLAB [2], [5], yang efektif untuk mempermudah perhitungan kinematika, akan tetapi memiliki keterbatasan pada aspek keamanan distribusi data dan skalabilitas aplikasi.

Pengembangan lanjutan mengarah pada penggunaan ekosistem *Robot Operating System* 2 (ROS 2), yang menggantikan arsitektur master node tunggal pada ROS 1 dengan protokol *Data Distribution Service* (DDS) untuk lapisan *middleware* komunikasi berbasis pola *publish-subscribe* [6], yang mendukung konfigurasi *Quality of Service* (QoS) serta optimasi performa komunikasi secara *real-time* [7], sehingga menawarkan komunikasi yang lebih andal, keamanan yang lebih baik, dan arsitektur yang lebih modular [8], [9]. Ekosistem ROS 2 ini kemudian digabungkan dengan simulator Gazebo, perangkat perencanaan gerak Movelt RViz, serta antarmuka grafis GUI Control untuk pengoperasian robot OpenManipulator-X, baik dalam bentuk simulasi maupun pada unit fisik secara *real-time*.

Berdasarkan hal tersebut, penelitian ini bertujuan untuk membangun simulasi interaktif dan representatif bagi robot OpenManipulator-X menggunakan integrasi ROS 2, Gazebo, dan Movelt RViz, serta membandingkan efektivitas dan karakteristik dua metode kendali, yaitu Movelt RViz dan GUI Control, termasuk pengaruh variasi tingkat kecepatan pada Movelt RViz. Penelitian ini juga bertujuan membuat prosedur pengujian yang aman pada simulator sebelum program akan dijalankan pada unit robot fisik, sehingga proses pengoperasian bisa dilakukan dengan lebih aman dan minim risiko kerusakan perangkat keras.

Copyright © SENASIF 2026
ISSN: 2598-0076

II. METODE

Metode yang digunakan pada penelitian ini bersifat eksperimental kuantitatif, dilakukan melalui empat tahapan utama, yaitu persiapan lingkungan kerja perangkat lunak, pengujian pada simulator, pengujian pada robot fisik, serta analisis perbandingan hasil.

Tahap awal adalah persiapan *environment* kerja, dimulai dengan instalasi sistem operasi Ubuntu 24.04 LTS (*Noble Numbat*) [10] di atas mesin virtual Oracle VirtualBox [11], karena sistem operasi berbasis Linux dinilai lebih sesuai untuk kebutuhan pengembangan sistem robotika dibandingkan sistem operasi lain [12]. Selanjutnya dilakukan instalasi ROS 2 *Jazzy Jalisco* [13] serta paket pendukung lainnya, yaitu open_manipulator_x_bringup, Gazebo *Harmonic*, Movelt 2 dengan paket open_manipulator_x_moveit_config, serta GUI Control (open_manipulator_x_gui), yang seluruhnya dibangun menggunakan *colcon build* pada *workspace* ROS 2.

Robot yang digunakan dalam penelitian ini adalah OpenManipulator-X, sebuah robot manipulator yang bersifat open-source dengan memiliki 4 derajat kebebasan (4 DOF) yang dilengkapi *gripper*, robot ini dibuat untuk mendukung riset dan pengembangan di bidang robotika industri [14], sebagaimana ditunjukkan pada Gambar 1. Robot ini digerakkan oleh motor servo Dynamixel XM430-W350-T [15] yang terhubung ke komputer melalui modul U2D2, sebagai jembatan komunikasi antara unit komputasi dan aktuator [16], [17].



Gambar 1. Robot OpenManipulator-X

Tahap kedua adalah pengujian gerak pada simulator Gazebo, yang dilakukan dalam dua skenario, yaitu gerak *single-point* (satu titik tujuan) dan gerak *multi-point* (beberapa titik tujuan secara berurutan). Setiap pengujian diawali dari posisi init pose (seluruh *joint* bernilai 0 radian) menuju titik koordinat tujuan (Home, B1, B2, B3, dan B4), yang dijalankan menggunakan dua metode kendali kontrol, yaitu Movelt RViz dengan fitur *motion planning* pada *joint space* [8], dan GUI Control yang mengirimkan nilai koordinat setiap joint secara langsung menuju *trajectory* tanpa melalui proses *motion planning* maupun *collision checking*.

Tahap selanjutnya adalah pengujian pada robot fisik OpenManipulator-X secara real-time hal ini sama seperti pada simulator (*single-point* dan *multi*-point), ditambah pengujian variasi tiga

Copyright © SENASIF 2026

ISSN: 2598-0076

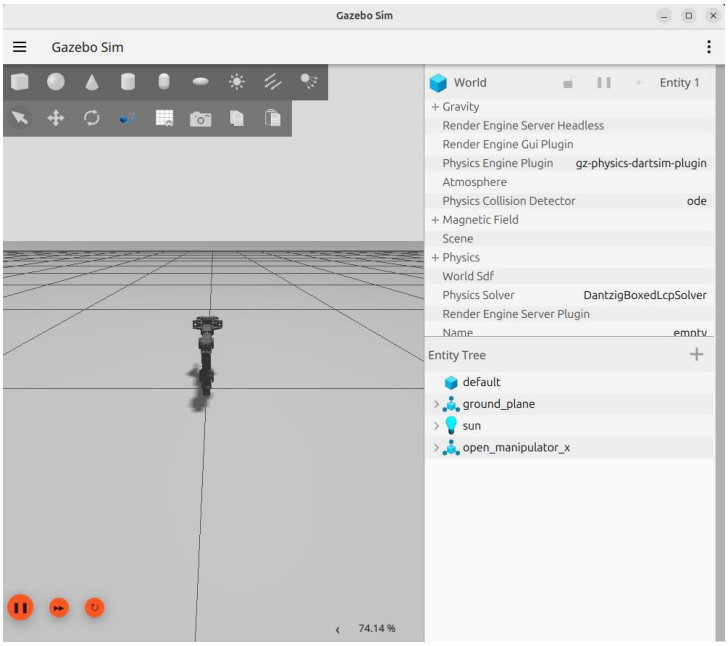
tingkat kecepatan (rendah, sedang, dan tinggi) pada Movelt RViz. Waktu eksekusi setiap gerakan dicatat menggunakan *stopwatch*, dimulai saat perintah eksekusi (*execute*) dikirimkan sampai robot berhenti pada posisi tujuan yang telah ditentukan.

Tahap terakhir adalah analisis data, yang meliputi perbandingan kuantitatif waktu eksekusi antara Movelt RViz dan GUI Control, perbandingan kualitatif karakteristik kedua metode (kemudahan penggunaan, presisi input, keamanan lintasan, dan visualisasi), serta pengujian kemampuan ekspor data posisi *joint* menggunakan perintah ros2 bag record, ros2 bag play, dan ros2 topic echo pada Gazebo, fitur *Stored* States pada Movelt RViz, dan fitur *Task Constructor* pada GUI Control.

III. HASIL DAN PEMBAHASAN

Pengujian Gerak pada Simulator Gazebo

Pengujian awal dilakukan pada simulator Gazebo untuk memvalidasi rencana lintasan sebelum diterapkan pada robot fisik. Pengujian gerak *single-point* dilakukan dengan memberi perinah robot bergerak dari posisi *init pose* menuju lima titik tujuan (Home, B1, B2, B3, dan B4) menggunakan metode MoveIt RViz dan GUI Control. Pengujian gerak *multi-point* dilakukan dengan memberi perintah robot bergerak menuju beberapa titik tujuan secara berurutan tanpa kembali ke posisi awal. Gambar 2 menunjukkan tampilan simulator Gazebo saat robot OpenManipulator-X berada pada posisi *init pose*.



Gambar 2. Tampilan Simulator Gazebo Pada Posisi Init Pose

Hasil pengujian menunjukkan seluruh hasil skenario gerak, baik *single-point* maupun multi-point, telah berhasil dijalankan tanpa mengalami kegagalan rencana gerak (*motion planning failure*), busa diluhat pada Tabel 1.

Copyright © SENASIF 2026
ISSN: 2598-0076

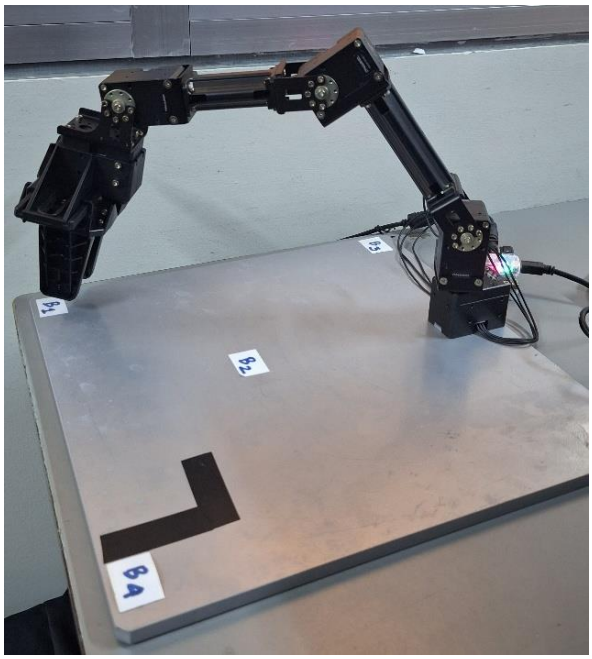
Tabel 1. Ringkasan hasil pengujian gerak pada simulator Gazebo

Skenario Pengujian	Jumlah Pengujian	Berhasil	Tingkat Keberhasilan
Single-point (Home, B1-B4)	5	5	100%
Multi-point (kombinasi titik)	6	6	100%

Keberhasilan pengujian pada simulator ini menandakan bahwa *controller* virtual yang ada pada MoveIt RViz maupun GUI Control dapat menggerakkan *end-effector* menuju titik koordinat yang sudah ditentukan, hal ini dapat dilanjutkan ke tahap pengujian robot fisik.

Pengujian Gerak pada Robot Fisik Secara Real-Time

Pengujian selanjutnya dilakukan pada robot fisik OpenManipulator-X yang terhubung melalui modul U2D2 ke motor Dynamixel XM430-W350-T, terlihat pada Gambar 3. Ilustrasi pengujian sama dengan pengujian pada simulator, yaitu gerak *single-point* dan multi- point, menggunakan metode MoveIt RViz dan GUI Control pada kecepatan default (normal).



Gambar 3. Pengujian Gerak Pada Robot Fisik OpenManipulator-X

Seluruh gerakan dari posisi *init pose* menuju titik Home, B1, B2, B3, dan B4 berhasil dijalankan tanpa kegagalan pada kedua metode kendali. Tabel 2 menunjukkan rata-rata waktu hasil eksekusi pengujian pada robot fisik untuk kecepatan default (normal).

Tabel 2. Rata-rata waktu eksekusi robot fisik pada kecepatan default

Metode Kontrol	Rata-rata Single-point	Rata-rata Multi-point
MoveIt RViz	4.18 detik	8.44 detik
GUI Control	4.00 detik	8.11 detik

Hasil pada Tabel 2 menunjukkan bahwa pada kecepatan default, GUI Control mendapatkan rata-rata waktu eksekusi yang sedikit lebih cepat dibandingkan dengan MoveIt RViz, baik pada gerak

Copyright © SENASIF 2026
ISSN: 2598-0076

single-point maupun *multi-point*. Hal ini disebabkan Movelt RViz melalui tahap *motion planning* dan *collision checking* sebelum eksekusi, sedangkan GUI Control mengirimkan perintah koordinat secara langsung tanpa melalui tahap tersebut.

Pengaruh Variasi Kecepatan pada MoveIt RViz

Pengujian tambahan dilakukan pada Movelt RViz dengan tiga tingkat kecepatan (rendah, sedang, dan tinggi) dan menggunakan fitur motion planning, kemudian dibandingkan dengan waktu tercepat GUI Control pada kecepatan normal. Hasil pengujian ditampilkan pada Tabel 3.

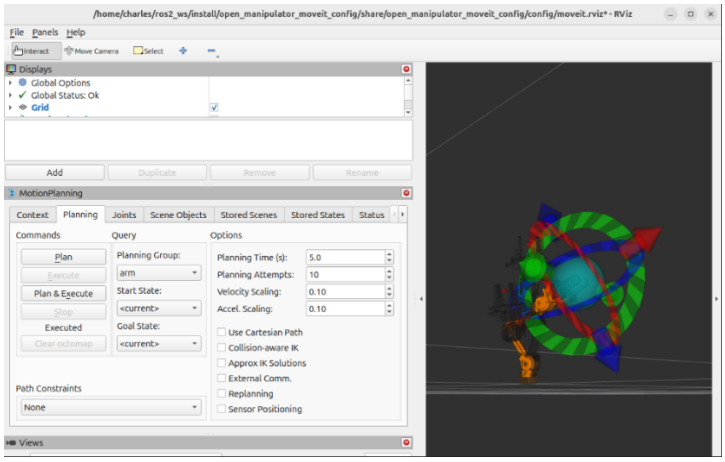
Tabel 3. Perbandingan waktu tercepat pada variasi kecepatan

Skenario	Metode / Kecepatan	Waktu
Single-point	MoveIt RViz – Rendah	3.88 detik
Single-point	MoveIt RViz – Sedang	1.81 detik
Single-point	MoveIt RViz – Tinggi	1.13 detik
Single-point	GUI Control – Normal	3.69 detik
Multi-point	MoveIt RViz – Rendah	10.27 detik
Multi-point	MoveIt RViz – Sedang	4.04 detik
Multi-point	MoveIt RViz – Tinggi	3.17 detik
Multi-point	GUI Control – Normal	6.80 detik

Hasil pada Tabel 3 menunjukkan bahwa peningkatan parameter kecepatan pada Movelt RViz secara signifikan menurunkan waktu eksekusi, yaitu sekitar 71% pada gerak *single-point* dan 69% pada gerak *multi-point* ketika kecepatan diatur ke tingkat tinggi, dibandingkan dengan kecepatan rendah. Pada kecepatan tinggi, Movelt RViz lebih cepat dibandingkan GUI Control pada kecepatan normal. Hal ini membuktikan bahwa selisih waktu eksekusi pada kecepatan *default* sebelumnya lebih dipengaruhi oleh parameter kecepatan yang digunakan, bukan dipengaruhi oleh perbedaan mekanisme kendali.

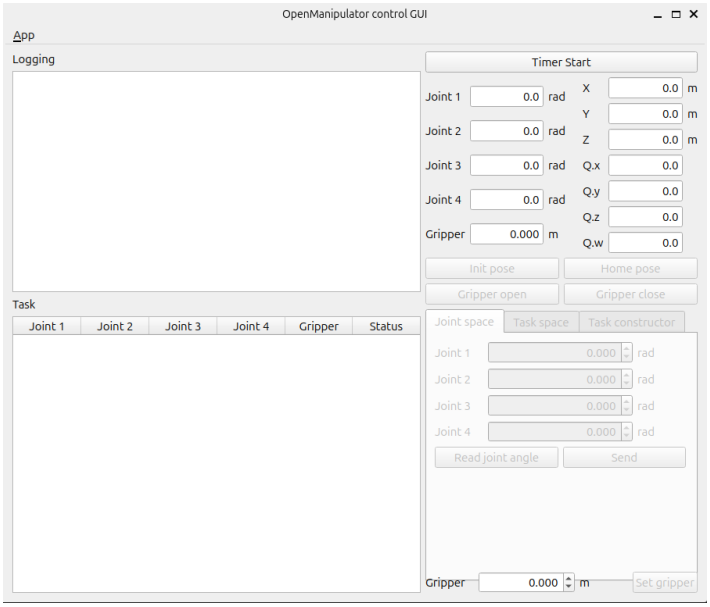
Perbandingan Karakteristik Metode Kontrol

Selain aspek kuantitatif berupa waktu eksekusi, kedua metode kendali juga memiliki karakteristik kualitatif yang berbeda. Gambar 4 menunjukkan antarmuka MoveIt RViz beserta panel Motion Planning dan visualisasi 3D interaktifnya, kemudian Gambar 5 menunjukkan antarmuka GUI Control pada tab *Joint Space*, dan Tabel 4 merangkum perbandingan karakteristik kedua metode.



Gambar 4. Antarmuka MoveIt RViz

Copyright © SENASIF 2026
ISSN: 2598-0076



Gambar 5. Antarmuka GUI Control

Tabel 4. Perbandingan karakteristik metode kontrol

Aspek	MoveIt RViz	GUI Control
Kemudahan pemakaian	Sedang	Mudah
Presisi input	Tinggi (slider)	Sedang (input)
Perencanaan lintasan	Ya	Tidak
Visualisasi 3D	Ya	Tidak
Cocok untuk pemula	Tidak	Ya
Cocok untuk presisi tinggi	Ya	Ya
Cocok untuk robot fisik	Ya	Ya
Input satuan	Derajat	Radian

Berdasarkan Tabel 4, Movelt RViz unggul dalam aspek perencanaan lintasan (*motion planning*) dan visualisasi 3D interaktif yang memberikan keamanan gerak lebih tinggi terhadap potensi tabrakan, sehingga lebih sesuai digunakan pada skenario yang membutuhkan presisi dan keamanan tinggi. Sebaliknya, GUI Control lebih unggul dalam kemudahan penggunaan bagi pengguna pemula serta kecepatan eksekusi pada konfigurasi *default*, sehingga lebih sesuai untuk operasi rutin yang membutuhkan kepraktisan.

Analisis Kemampuan Ekspor Data

Pengujian kemampuan ekspor data posisi *joint* dilakukan pada ketiga antarmuka. Pada Gazebo, penggunaan perintah ros2 bag record, ros2 bag play, dan ros2 topic echo menghasilkan pesan "*a message was lost*" hal ini menandakan berkas CSV yang dihasilkan tidak lengkap. Pada Movelt RViz, fitur *Stored States* pada panel *Motion Planning* tidak dapat menyimpan data karena bergantung pada basis data eksternal yang tidak dijalankan pada konfigurasi ROS 2 yang digunakan. Sebaliknya, fitur *Task Constructor* pada GUI Control berhasil menyimpan data koordinat setiap joint ke dalam berkas CSV secara lengkap dan dapat dibaca kembali. Hasil ini menunjukkan bahwa GUI Control merupakan satu-satunya metode dalam penelitian ini yang memiliki mekanisme pencatatan data posisi *joint* secara otomatis dan andal.

Copyright © SENASIF 2026
ISSN: 2598-0076

IV. SIMPULAN

Implementasi kendali gerak menggunakan robot OpenManipulator-X berbasis ROS 2 *Jazzy Jalisco* pada sistem operasi Ubuntu 24.04 LTS berhasil dilakukan menggunakan integrasi Gazebo, Movelt RViz, dan GUI Control, baik pada simulator maupun robot fisik secara *real-time*. Seluruh pengujian gerak *single-point* dan *multi-point* menuju titik koordinat Home, B1, B2, B3, dan B4 berhasil dijalankan tanpa kegagalan pada kedua metode kendali. Pada konfigurasi kecepatan *default*, GUI Control memperoleh rata-rata waktu eksekusi yang lebih cepat dibandingkan Movelt RViz, namun peningkatan parameter kecepatan pada Movelt RViz mampu menurunkan waktu eksekusi hingga 71% pada gerak *single-point* dan 69% pada gerak *multi-point*, sehingga lebih cepat dibandingkan GUI *Control* pada kecepatan tinggi. Berdasarkan karakteristik kualitatif, Movelt RViz lebih unggul dalam aspek perencanaan lintasan dan visualisasi 3D interaktif yang aman terhadap potensi tabrakan, sedangkan GUI Control lebih unggul pada kemudahan penggunaan dan kemampuan ekspor data posisi *joint* secara otomatis ke berkas CSV. Kedua metode kendali terbukti saling melengkapi dan dapat dipilih sesuai kebutuhan spesifik pengguna dalam mengoperasikan robot OpenManipulator-X.

Penelitian selanjutnya disarankan untuk menerapkan skenario pengujian pick-and-place menggunakan kedua metode kendali, serta menambahkan sistem machine vision untuk mendeteksi objek secara langsung sehingga robot dapat merespons posisi objek secara dinamis pada lingkungan real-time.

DAFTAR RUJUKAN

[1] N. A. P. Pangestu, J. Parung, and E. Wibisono, "Penerapan Teknologi Industri 4.0 Dalam Lean Warehousing: Literature Review," *Jurnal Ilmiah Teknik Industri*, vol. 12, no. 2, pp. 79–90, Oct. 2024, doi: 10.24912/jitiuntar.v12i2.31009.

[2] D. K. Winarta, R. B. Widodo, and M. Subianto, "Rancang Bangun Pengontrol Gerakan Robot OpenManipulator dengan Matlab," *Sainsbertek Jurnal Ilmiah Sains & Teknologi* , vol. 5, pp. 83–91, 2024, doi: 10.33479/sb.v5i1.343.

[3] A. I. I. Muslim, "Definisi Robot," 2022, Accessed: May 06, 2026. [Online]. Available: https://www.researchgate.net/publication/364315540_DEFINISI_ROBOT

[4] World Economic Forum, "The Future of Jobs Report 2023." Accessed: May 06, 2026. [Online]. Available: https://www.weforum.org/publications/the-future-of-jobs-report-2023/

[5] M. A. I. Al-Ikhsan, "Pengendalian Gerakan Robot Openmanipulator Untuk Operasi Pemindahan Barang Berbasis Matlab," 2025.

[6] G. Deng, G. Xu, Y. Zhou, T. Zhang, and Y. Liu, "On the (In)Security of Secure ROS2," *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 739–753, 2022, doi: doi.org/10.1145/3548606.3560681.

[7] Y. Ye, Z. Nie, X. Liu, F. Xie, Z. Li, and P. Li, "ROS2 Real-time Performance Optimization and Evaluation," *Chinese Journal of Mechanical Engineering*, vol. 36, no. 1, p. 144, Dec. 2023, doi: 10.1186/s10033-023-00976-5.

[8] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, May 2022, doi: 10.1126/scirobotics.abm6074.

Copyright © SENASIF 2026
ISSN: 2598-0076

[9] A. Bonci, F. Brunella, M. Colletta, A. Di Biase, A. F. Dragoni, and A. Libofsha, “ROS 2-Based Architecture for Autonomous Driving Systems: Design and Implementation,” *Sensors*, vol. 26, no. 2, p. 463, Jan. 2026, doi: 10.3390/s26020463.
[10] Canonical Ltd., “Ubuntu 24.04 LTS (Noble Numbat) Release Notes.” Accessed: Jun. 12, 2026. [Online]. Available: https://discourse.ubuntu.com/t/noble-numbat-release-notes/
[11] Oracle Corporation, *Oracle VM VirtualBox User Manual for Release 7.0*. 2024. Accessed: Jun. 12, 2026. [Online]. Available: https://www.virtualbox.org/manual/UserManual.html
[12] R. Al Fajar, Zidan, and A. Lestari, “Analisis Perbandingan Sistem Operasi Windows 11 dan Linux Ubuntu Menggunakan Metode Studi Literatur (Studi Kasus: Kinerja Sistem, Keamanan dan Biaya),” *Jurnal Bitwise*, vol. 1, no. 2, pp. 74–82, 2025, Accessed: Jun. 12, 2026. [Online]. Available: https://jurnal-bitwise.org/index.php/bitwise/article/view/20/14
[13] Open Robotics, “ROS 2 Jazzy Jalisco Released.” Accessed: Jul. 04, 2026. [Online]. Available: https://www.openrobotics.org/blog/2024/5/ros-jazzy-jalisco-released
[14] M. A. M. Adzeman, M. H. M. Zaman, M. F. Nasir, M. F. Ibrahim, and S. M. Mustaza, “Kinematic Modeling of A Low Cost 4 DOF Robot Arm System,” *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 10, pp. 6828–6834, Oct. 2020, doi: 10.30534/ijeter/2020/328102020.
[15] ROBOTIS e-Manual, “Dynamixel.” Accessed: Jul. 09, 2026. [Online]. Available: https://emanual.robotis.com/docs/en/dxl/
[16] J. Kim, D. C. Mathur, K. Shin, and S. Taylor, “PAPRAS: Plug-And-Play Robotic Arm System,” 2023, doi: https://doi.org/10.48550/arXiv.2302.09655.
[17] ROBOTIS e-Manual, “U2D2 Power Hub.” Accessed: Jul. 09, 2026. [Online]. Available: https://emanual.robotis.com/docs/en/parts/interface/u2d2_power_hub/

Copyright © SENASIF 2026

ISSN: 2598-0076