

# Research Study of ROS2 Library and Interface Development untuk Robot Lengan Industri Mitsubishi RV2-SDB

Eko Budi Utomo<sup>1</sup>, Mochammad Abdul Aziz<sup>2</sup>

<sup>1,2</sup>Teknik Mekatronika, Departemen Teknik Mekatronika, Politeknik Elektronika Negeri Surabaya, Surabaya, Indonesia

## Article Info

### Article history:

Received January 20, 2025

Revised February 5, 2025

Accepted April 11, 2025

### Keywords:

Robot Lengan  
Otomasi Industri  
ROS2

## ABSTRACT

Kajian penelitian ini bertujuan untuk mengembangkan *ROS2 library* yang menyediakan *interface* bagi robot lengan industri Mitsubishi RV-2SDB. *Library* ini memfasilitasi akses data penting, seperti data *joint*, posisi, *mode jog*, pesan *error*, dan alarm. Dengan memanfaatkan ROS2, *library* ini dirancang untuk mendukung integrasi dan pengembangan robot dalam simulasi proses industri, serta otomatisasi yang lebih luas. Penggunaan ROS2 memberikan modularitas dan skalabilitas, memungkinkan integrasi mudah dengan sistem otomasi yang sudah ada. Kajian penelitian menunjukkan bahwa *library* ini mampu meningkatkan efisiensi dan keandalan operasi robot, mendukung revolusi industri 4.0 dengan menciptakan solusi yang fleksibel dan integratif.

This is an open access article under the [CC BY-SA](#) license.



### Corresponding Author:

Eko Budi Utomo,

Politeknik Elektronika Negeri Surabaya, Jl. Raya ITS, Sukolilo, Surabaya, Indonesia

Email: ekobudi\_u@pens.ac.id

## 1. PENDAHULUAN

Perkembangan teknologi robot pada proses produksi sebuah industri terus dikembangkan. Hal ini menjadi jelas bahwa robot mulai memiliki peran utama di dunia industri. Robot untuk industri dapat didefinisikan sebagai kendali komputer dengan manipulator yang dirancang untuk meniru gerakan manusia, seperti gerakan untuk melaksanakan sejumlah tugas industri yang berbeda tanpa campur tangan manusia. Dengan kemampuan tersebut, robot dipercaya untuk menyelesaikan pekerjaan yang tak mampu dan tak ingin dilakukan oleh manusia, sehingga waktu yang dibutuhkan lebih efisien dan *output* dari proses produksi lebih optimal. Berdasarkan data yang telah dipaparkan oleh lembaga survei *International Federation of Robotics* (IFR) dalam laporan terbaru *World Robotics* yang dirilis Oktober 2022. Hasil survei mengatakan pertumbuhan robot industri pada tahun 2021 ada 517 ribu unit robot produksi baru yang dipasang di pabrik seluruh dunia. Bahkan, selama periode 2011-2021 sudah ada 3.47 juta robot industri yang sudah beroperasi.

Tingkat kebutuhan robot yang ada pada industri yang semakin banyak, mendorong untuk terciptanya sebuah sistem yang dapat memanfaatkan robot untuk membantu proses yang ada di industri. Beberapa tipe robot seperti *mobile robot* dan robot manipulator adalah salah 2 tipe robot yang umum digunakan pada sektor industri. Penerapan *mobile robot* sering diterapkan pada AGV (*Automated Guide Vehicle*) yaitu kendaraan otomatis yang digunakan untuk mengangkut material atau barang dari satu tempat ke tempat lain tanpa perlu pengemudi manusia Sedangkan robot manipulator yang secara fisik dan fungsi menyerupai lengan pada manusia contohnya untuk memindahkan dan mengangkat barang, melakukan perakitan pada suatu komponen *assembly*, dan lain - lain. Salah satu contoh adalah robot lengan yang saya gunakan untuk *project* ini adalah Mitsubishi RV-2SDB, sebuah robot industri *compact* dan ringan yang dirancang untuk aplikasi presisi tinggi dengan kemampuan menangani beban hingga 3 kg. Dengan enam derajat kebebasan, robot ini menawarkan gerakan fleksibel dan cepat, ideal untuk tugas perakitan, pemindahan material ringan, dan pekerjaan presisi di 1 industri elektronik dan otomotif. Dengan dilengkapi sistem kontrol canggih dan perangkat lunak pemrograman, robot lengan Mitsubishi

RV 2SDB dapat dipastikan beroperasi dengan handal dan aman, sehingga proses produksi menjadi lebih efisien.

Berdasarkan latar belakang untuk memenuhi kebutuhan dalam proses produksi tersebut, diperlukan penelitian lebih lanjut dalam pembuatan *ROS2 library interface* untuk robot lengan Mitsubishi RV 2SDB. Dimana library ini tidak hanya menyediakan akses ke data dasar yang diperlukan untuk mengoperasikan lengan robot, tetapi juga menawarkan kemampuan untuk menangani situasi *error* dan alarm, yang penting untuk memastikan operasi yang aman dan handal. Implementasi mode jog memungkinkan kontrol manual yang lebih presisi dalam lingkungan yang membutuhkan penyesuaian yang halus dan fleksibel. Dalam hal ini, robot lengan Mitsubishi RV-2SDB, penulis menggunakan *controller* tambahan yang digunakan sebagai *controller slave* dari ROS 2, yaitu PLC Siemens S7-1200.

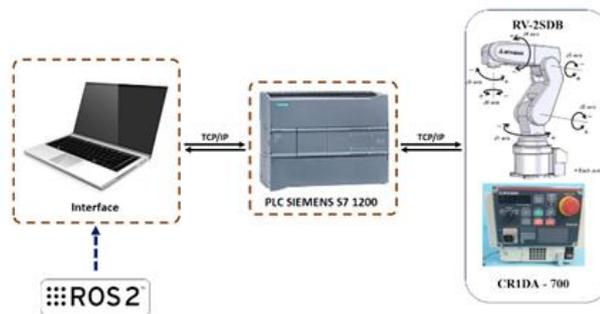
Penggunaan ROS2 sebagai basis pengembangan *library* ini memberikan keuntungan tambahan berupa modularitas dan skalabilitas, yang mendukung adopsi yang lebih luas dalam berbagai aplikasi industri. Selain itu, *ROS2 interface* yang dikembangkan memungkinkan integrasi yang lebih mudah dengan sistem otomasi yang sudah ada, sehingga mempercepat proses implementasi dan mengurangi waktu pengembangan. Diharapkan *library* ini akan menjadi bagian integral dari solusi otomasi di berbagai sektor industri, meningkatkan efisiensi dan efektivitas *factory operations*.

## 2. METODE

Pada BAB ini membahas tentang langkah-langkah proyek akhir yang meliputi perencanaan perangkat keras dan perangkat lunak secara keseluruhan. *Controller Slave* terdiri dari PLC S7-1200 yang terhubung ke PC melalui protokol komunikasi TCP/IP, dan PC ini menjalankan program ROS2 (sebagai *Controller Master*) yang dilengkapi dengan *ROS2 library and interface*. *Controller* ini mengirim perintah dan menerima *feedback* dari robot lengan Mitsubishi RV-2SDB berupa data sensor posisi, serta bisa mengirim dan menerima data dari *ROS2 interface* untuk mengendalikan robot lengan. Robot lengan dapat dijalankan melalui *input* ke CR1DA-700 atau melalui *ROS2 interface*. Dengan *ROS2 interface* mengirim data *mode jog* untuk kontrol posisi. *ROS2 interface* juga menerima data *error*, *joint*, dan posisi dari robot lengan untuk pemantauan, analisis, dan respons cepat terhadap perubahan operasional.

### 2.1. Perancangan Desain Sistem

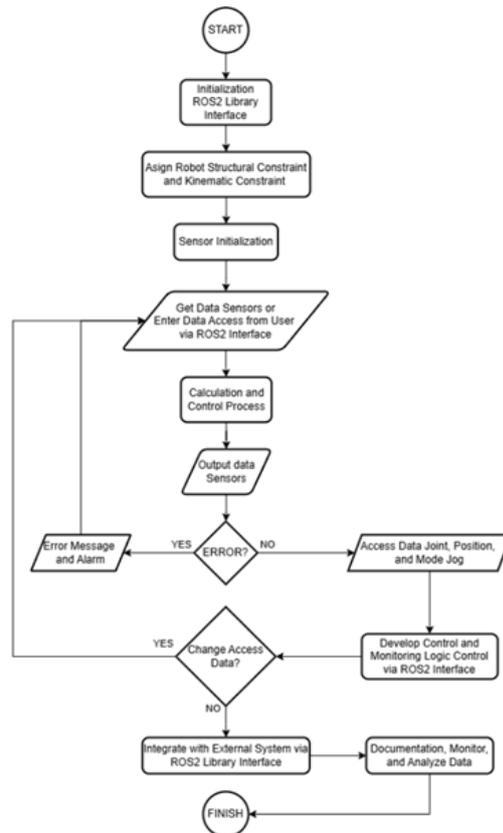
Perancangan desain sistem terdiri dari PLC S7-1200 sebagai *controller slave* yang dikomunikasikan menggunakan protokol TCP/IP ke PC. PC ini digunakan untuk mengolah program ROS2 yang digunakan untuk mengirim perintah dan menerima *feedback* dari robot lengan berupa data sensor posisi. Selain itu *controller* ini juga digunakan untuk mengirim data ke *ROS2 interface* serta untuk menerima dan mengolah *input* dari *ROS2 interface* untuk mengendalikan robot lengan.



**Gambar 1.** Diagram Desain Sistem

Bagian pertama yaitu sistem yang mengirim perintah ke robot lengan. Robot lengan bisa dijalankan dengan memberikan input ke CRIDA - 700 (controller khusus pada Robotic Arm RV-2SDB) atau memberikan input melalui ROS2 interface yang dikomunikasikan menggunakan protokol TCP/IP ke controller. ROS2 interface bisa mengirimkan data mode jog untuk mengontrol posisi robot lengan. Sedangkan bagian yang kedua, ROS2 interface bisa menerima data error, joint, dan posisi dari robot lengan untuk pemantauan, analisis, dan respons yang cepat terhadap perubahan dalam lingkungan operasional atau keadaan robot.

### 2.2. Perancangan Sistem



**Gambar 2.** Diagram Alir Sistem

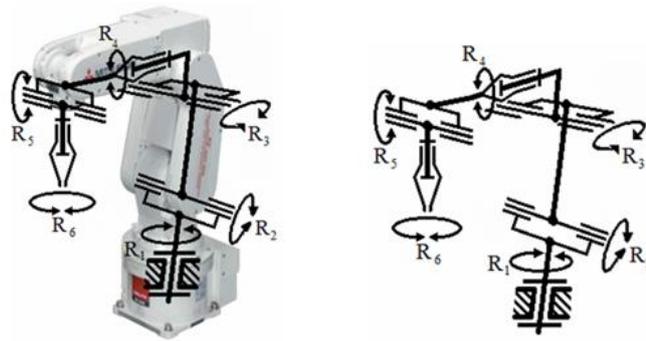
Proses kalkulasi dan kontrol dilakukan berdasarkan data yang diperoleh, dan hasilnya dihasilkan sebagai output dari sensor. Sistem kemudian memeriksa apakah ada kesalahan dalam proses tersebut. Jika ditemukan kesalahan, pesan kesalahan dan alarm akan ditampilkan. Jika tidak, sistem akan mengakses data terkait joint, posisi, dan mode jog dari robot. Pada titik ini, jika ada perubahan dalam akses data, sistem kembali ke langkah pengambilan data. Jika tidak ada perubahan, logika kontrol dan pemantauan dikembangkan melalui *ROS2 interface*. Integrasi sistem dengan sistem eksternal dilakukan melalui antarmuka *ROS2 library*. Akhirnya, sistem melakukan dokumentasi, pemantauan, dan analisis data untuk evaluasi dan peningkatan lebih lanjut. Langkah terakhir menandakan bahwa seluruh rangkaian proses telah selesai.

### 2.3. Perancangan Sistem Komunikasi

Sistem komunikasi dirancang menggunakan protokol TCP/IP untuk mengirim dan menerima data antar perangkat keras, dengan PLC Siemens S7-1200 sebagai perantara antara PC (yang menjalankan ROS2 Library and Interface) dan robot lengan Mitsubishi RV-2SDB melalui modul CR1DA-700. Protokol ini dipilih karena fleksibilitas, kompatibilitas dengan standar industri, dan kemampuannya mendukung komunikasi real-time. Dengan memanfaatkan paket ROS Industrial bernama *ros\_industrial\_siemens*, komunikasi antara PLC dan ROS2 dilakukan melalui protokol PROFINET atau TCP/IP. Modul CR1DA-700 dikonfigurasi di TIA Portal untuk mengontrol input dan output sesuai kebutuhan, sehingga integrasi antara ROS2, PLC, dan robot lengan RV-2SDB memungkinkan pengendalian aksi robot secara efisien dan real-time berdasarkan data sensor yang dikirim oleh PLC.

### 2.4. Kinematics Robot Lengan Industri Mitsubishi RV2SDB

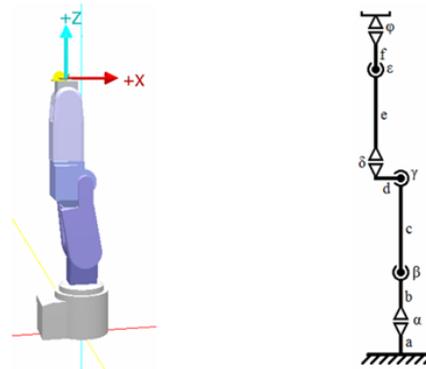
Struktur kinematika robot lengan ini digambar berdasarkan model robot industri, kemudian model tersebut dihilangkan, sehingga hanya menyisakan struktur kinematika 3D dari lengan robot.



**Gambar 3.** Struktur Kinematik dan 3D Robot Lengan Mitsubishi RV-2SDB

Sumber : Papcun Peter, Jadlovský Ján, (2015), “Mathematical Model of Robot Melfa RV-2SDB”, Technical University of Kosice, pp. 3.

Struktur kinematika 3D digambar ulang menjadi struktur kinematika klasik untuk menyederhanakannya. Pada struktur kinematika klasik ini, lengan-lengan diberi penanda huruf dari a hingga f, sedangkan sambungan-sambungan R1 hingga R6 ditandai dengan huruf Yunani ( $\alpha, \beta, \gamma, \delta, \epsilon, \text{ dan } \varphi$ ).



**Gambar 4.** Skema Kinematik Robot Lengan Mitsubishi RV-2SDB Saat Joint Robot 0 Derajat

Sumber : Papcun Peter, Jadlovský Ján, (2015), “Mathematical Model of Robot Melfa RV-2SDB”, Technical University of Kosice, pp. 4.

Pada **Gambar 4.** di atas adalah kondisi robot lengan Mitsubishi RV-2SDB dalam kondisi *default* di saat masing – masing *joint* bernilai  $0^\circ$ . Posisi *default* ini menjadi titik awal dimulainya *forward kinematics* tiap *joint*  $0^\circ$  dan *inverse kinematics* X, Y, Z, Roll, Pitch, Yaw bernilai  $0, 0, 0, 0^\circ, 0^\circ, 0^\circ$ .

**Tabel 1.** Informasi Penting Robot Lengan Mitsubishi RV-2SDB

Dimensi Masing-Masing Lengan	
$a = 0.0$ meter	$d = 0.05$ meter
$b = 0.295$ meter	$e = 0.27$ meter
$c = 0.23$ meter	$f = 0.7$ meter
Batasan Masing-Masing Joint	
$\alpha = -240 \text{ deg} ; 240 \text{ deg}$	$\delta = -200 \text{ deg} ; 200 \text{ deg}$
$\beta = -120 \text{ deg} ; 120 \text{ deg}$	$\epsilon = -120 \text{ deg} ; 120 \text{ deg}$
$\gamma = 0 \text{ deg} ; 160 \text{ deg}$	$\varphi = -360 \text{ deg} ; 360 \text{ deg}$

**2.4.1. Forward Kinematics**

$$A = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 1)}$$

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 2)}$$

$$C = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 3)}$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 4)}$$

$$E = \begin{bmatrix} \cos \gamma & 0 & \sin \gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \gamma & 0 & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 5)}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & -d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 6)}$$

$$G = \begin{bmatrix} \cos \delta & -\sin \delta & 0 & 0 \\ \sin \delta & \cos \delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 7)}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & e \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 8)}$$

$$I = \begin{bmatrix} \cos \varepsilon & 0 & \sin \varepsilon & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \varepsilon & 0 & \cos \varepsilon & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 9)}$$

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 10)}$$

$$K = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 & 0 \\ \sin \varphi & \cos \varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 11)}$$

*Composite homogeneous transformation matrix* adalah hasil dari perhitungan transformasi matriks di atas.

$$T = A.B.C.D.E.F.G.H.I.J.K \quad \text{(Persamaan 12)}$$

Selanjutnya didapatkan hasil dari *resultant composite homogeneous transformation matrix*, sebagai berikut :

$$T = \begin{bmatrix} n_x & o_x & a_x & x \\ n_y & o_y & a_y & y \\ n_z & o_z & a_z & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Persamaan 13)}$$

Dimana :

- Elemen  $n_x, n_y, n_z$  : Komponen orientasi *end-effector* dalam arah sumbu X.
- Elemen  $o_x, o_y, o_z$  : Komponen orientasi *end-effector* dalam arah sumbu Y.
- Elemen  $a_x, a_y, a_z$  : Komponen orientasi *end-effector* dalam arah sumbu Z.
- Elemen  $x, y, z$  : Posisi (koordinat) *end-effector* di ruang global.

#### 2.4.2. Inverse Kinematics

*Inverse kinematic* pada robot lengan adalah proses menentukan sudut setiap *joint* agar *end-effector* mencapai posisi dan orientasi tertentu di ruang 3D. Dengan metode *Jacobian matrix*, perhitungan ini menghubungkan perubahan kecil pada posisi *end-effector* dengan perubahan sudut *joint*, di mana *Jacobian matrix* ( $J(\theta)$ ) adalah hubungan linear antara kecepatan *joint-space* ( $\dot{\theta}$ ) dan kecepatan *Cartesian-space* ( $\dot{X}$ ).

Konsep *Jacobian matrix* dinyatakan sebagai :

$$\dot{X} = J(\theta) \cdot \dot{\theta} \quad \text{(Persamaan 14)}$$

Dimana :

- $\dot{X}$  : Kecepatan *end-effector* (translasi dan rotasi).
- $J(\theta)$  : *Jacobian matrix*, bergantung pada posisi sendi ( $\theta$ ).
- $\dot{\theta}$  : Kecepatan sudut pada tiap *joint*.

Untuk *inverse kinematics*, *Jacobian* digunakan untuk menghitung perubahan kecil pada sudut ( $\Delta\theta$ ) untuk mencapai posisi target ( $X_d$ ) :

$$\Delta\theta = J^{-1}(\theta) \cdot \Delta X \quad \text{(Persamaan 15)}$$

*Jacobian* terdiri dari dua bagian :

a. *Jacobian* Translasi ( $J_v$ ):

- Menghubungkan kecepatan translasi *end-effector* dengan kecepatan *joint*.
- Elemen  $J_v[:, i]$  dihitung sebagai:

$$J_v[:, i] = z_i \times (p_e - p_i) \quad \text{(Persamaan 16)}$$

b. *Jacobian* Rotasi ( $J_\omega$ ):

- Menghubungkan kecepatan rotasi *end-effector* dengan kecepatan *joint*.

$$J_\omega[:, i] = z_i \quad \text{(Persamaan 17)}$$

Dimana :

- $z_i$  : Sumbu rotasi *joint* ( $i$ ) dalam ruang global.
- $p_e$  : Posisi *end-effector*.
- $p_i$  : Posisi *joint* ( $i$ ).
- $[:, i]$  : Seluruh baris dalam kolom ( $i$ ).

Gabungan  $J_v$  dan  $J_\omega$  :

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \quad \text{(Persamaan 18)}$$

Iterasi *inverse kinematics* hingga  $\Delta X$  mendekati 0 :

a. *Error* posisi dan orientasi :

- Target posisi :

$$X_d = [x_d, y_d, z_d, \phi_d, \theta_d, \psi_d] \quad \text{(Persamaan 19)}$$

- Posisi *actual* :

$$X_a = [x, y, z, \phi, \theta, \psi] \quad \text{(Persamaan 20)}$$

- *Error* :

$$\Delta X = X_d - X_a \quad \text{(Persamaan 21)}$$

b. Perubahan sudut *joint* ( $\Delta\theta$ ) :

- Jika *Jacobian* terbalik  $J^{-1}$ , dihitung secara langsung :

$$\Delta\theta = J^{-1}(\theta) \cdot \Delta X \quad \text{(Persamaan 22)}$$

- Jika  $J$  tidak terbalik ( $J$  tidak persegi), gunakan *pseudo-inverse* :

$$\Delta\theta = J^+(\theta) \cdot \Delta X \quad \text{(Persamaan 23)}$$

$$J^+ = J^T (J J^T)^{-1} \quad \text{(Persamaan 24)}$$

c. Update posisi *joint* :

$$\theta_{new} = \theta_{old} + \Delta\theta \quad \text{(Persamaan 25)}$$

Persamaan *Jacobian* untuk robot lengan Mitsubishi RV-2SDB :

$$J = \begin{bmatrix} z_1 \times (p_e - p_1) & \cdots & z_6 \times (p_e - p_6) \\ z_1 & \cdots & z_6 \end{bmatrix} \quad \text{(Persamaan 26)}$$

## 2.5. Perencanaan Pengujian Sistem

Perencanaan sistem dimulai dengan menentukan indikator kegagalan, dampaknya, dan langkah pencegahan untuk mengurangi risiko error. Sistem dibagi menjadi beberapa sub-sistem untuk mempermudah analisis dan penanganan masalah, meliputi pengujian simulasi ROS2 (termasuk forward dan inverse kinematics), komunikasi robot melalui RS232 atau TCP/IP, fungsi ROS2 Library, integrasi sistem dengan sensor, monitoring real-time melalui ROS2 Interface, serta pengujian performa real-time. Selain itu, dilakukan pengujian antarmuka ROS2 dengan uji beban dan stabilitas komunikasi antara ROS2 Interface dan perangkat keras untuk memastikan sistem berfungsi optimal.

## 2.6. Metode Pengambilan Data

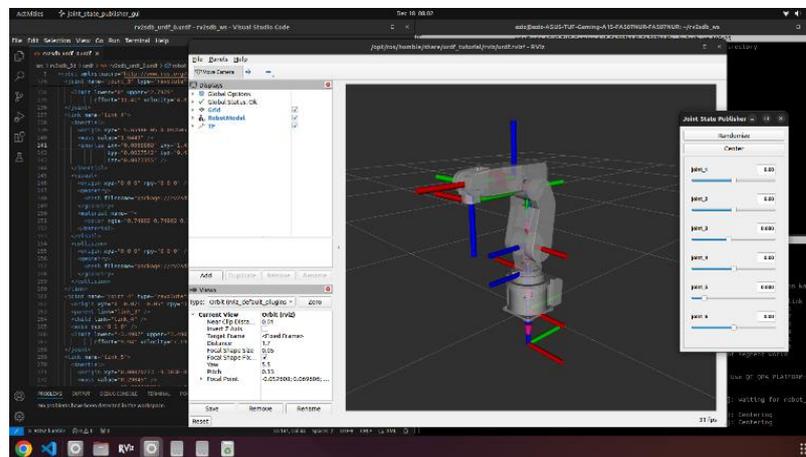
Metode pengambilan data melibatkan pengumpulan informasi selama pengujian dan analisis kinerja sistem, mencakup data kontrol posisi dan kecepatan robot dengan tambahan beban tertentu, data error handling untuk mendeteksi jenis error dan alarm melalui ROS2 Interface, serta analisis performa real-time dengan mengukur waktu pengiriman perintah, respons, dan penyelesaian tugas menggunakan timestamp dan logger pada ROS2.

## 3. HASIL DAN PEMBAHASAN

Pada BAB ini dijelaskan tentang hasil pengujian dari sistem yang dibuat beserta dengan analisa pengujian sistem tersebut.

### 3.1. Tampilan Robot Lengan Mitsubishi RV-2SDB di Rviz2

RViz2 (*Robot Visualization Version 2*) adalah salah satu *tools* dari ROS2 (*Robot Operating System 2*) yang digunakan untuk visualisasi desain 3D di komputer. Selain visualisasi 3D, RViz2 bisa digunakan untuk memuat informasi secara *real-time* posisi robot, kecepatan, data *joint*, *trajectory*, dan lainnya.



Gambar 4. Tampilan Robot Lengan Mitsubishi RV-2SDB di Rviz2

Gambar menunjukkan simulasi robot lengan RV-2SDB di RViz2, alat visualisasi ROS2, dengan sumbu koordinat X (merah), Y (hijau), dan Z (biru) pada setiap joint untuk memahami orientasi dan konfigurasi robot. Panel *Joint State Publisher* memungkinkan pengaturan manual posisi *joint* melalui *slider*, membantu memverifikasi kinematika, mengevaluasi orientasi *end-effector*, dan memastikan gerakan sesuai sebelum implementasi fisik.

### 3.2. Pengujian Posisi pada Robot Lengan Mitsubishi RV-2SDB

Pengujian posisi ini dilakukan dengan robot lengan Mitsubishi RV-2SDB langsung secara fisik dengan dibantu *software CIROS Programming* sebagai *tools* tambahan.



**Gambar 5.** Robot Lengan Mitsubishi RV-2SDB

Pada **Tabel 4.1** di bawah adalah posisi referensi dari *end-effector*, meliputi posisi  $X, Y, Z$  dan orientasi *Roll, Pitch, Yaw*. Berikutnya, adalah data perbandingan posisi referensi dan posisi actual pada robot lengan Mitsubishi RV-2SDB. Dari data ini, penulis dapat mengetahui seberapa besar *error* yang dihasilkan dari gerak sederhana robot lengan Mitsubishi RV-2SDB.

**Tabel 2.** Posisi Robot Lengan Mitsubishi RV-2SDB di CIROS Programming

No	Position ( $x, y, z$ )	Orientation ( $\phi, \theta, \psi$ )	Comment
P1	270.0, 0.0, 505.0	180, 0, 180, R, A, N	Home Position
P2	417.7, 152.0, 462.7	-180, 0, -180, R, A, N	Posisi di atas persiapan
P3	417.6, 152.0, 266.6	-180, 0, -180, R, A, N	Posisi mengambil
P4	2.4, 89.4, 417.6	-178, 20, -89, R, A, N	Persiapan_1
P5	0.0, 151.6, 476.9	0, 90, 90, R, A, N	Persiapan_2
P6	0.0, 491.9, 555.1	180, 86, -90, R, A, N	Menjulang
P7	35.6, 445.1, 531.9	-26, 49, -15, R, A, N	Menjulang serong ke atas
P8	-161.5, -51.7, 579.6	77, 22, -61, L, A, N	Menjulang_1
P9	-161.5, -51.7, 579.6	77, 22, -4, L, A, N	Menjulang_2

Dari **Tabel 2.** dijadikan sebagai posisi referensi untuk kontrol posisi robot lengan Mitsubishi RV-2SDB. Lalu didapatkan posisi actual dan deviasi pada **Tabel 3.** berikut :

**Tabel 3.** Pengujian Posisi pada Robot Lengan Mitsubishi RV-2SDB Dibantu dengan CIROS Programming

Ke-1	Ke-2	Posisi Referensi [ $x, y, z ; \phi, \theta, \psi$ ]	Posisi Aktual [ $x, y, z ; \phi, \theta, \psi$ ]	Deviasi
P1	P2	417.7, 152.0, 462.7 ; 180, 0, 180	417.66, 152.01, 462.68 ; 180, 0, 180	0.046 mm, 0°
P2	P3	417.7, 152.0, 266.6 ; 180, 0, 180	417.65, 152.02, 266.57 ; 180, 0, 180	0.061 mm, 0°
P3	P2	417.66, 152.01, 462.68 ; 180, 0, 180	417.66, 152.01, 462.68 ; 180, 0, 180	0 mm, 0°
P2	P1	270.0, 0.0, 505.0 ; 180, 0, 180	270, 0, 505 ; 180, 0, 180	0 mm, 0°
P1	P4	2.4, 89.4, 417.6 ; -178, 20, -89	2.360, 89.370, 417.60 ; -177.88, 19.99, -89.45	0.05 mm, 0.47°
P4	P5	0.0, 151.6, 476.9 ; 0, 90, 90	0, 151.6, 476.92 ; 116.57, 90, -153.43	0.02 mm, 202.03°
P5	P6	0.0, 491.9, 555.1 ; 180, 86, -90	0.01, 491.94, 555.1 ; 179.990, 86.06, -90.01	0.05 mm, 0.07°
P6	P7	35.6, 445.1, 531.9 ; -26, 49, -15	35.590, 445.140, 531.930 ; -25.500, 49.390, -15.080	0.06 mm, 0.48°

P1	P8	-161.5, -51.7, 579.6 ; 77, 22, -61	-161.53, -51.67, 579.57 ; 76.740, 21.730, -60.920	0.05 mm, 0.34°
P8	P9	-161.5, -51.7, 579.6 ; 77, 22, -4	-161.53, -51.67, 579.57 ; 76.74, 21.73, -3.81	0.05 mm, 0.37°

Persamaan yang digunakan untuk menghitung deviasi antara posisi referensi dan posisi *actual* pada **Tabel 3.** adalah sebagai berikut :

➤ Deviasi Translasi =

$$\sqrt{(x_{target} - x_{aktual})^2 + (y_{target} - y_{aktual})^2 + (z_{target} - z_{aktual})^2} \quad \text{(Persamaan 27)}$$

➤ Deviasi Orientasi =

$$\sqrt{(\phi_{target} - \phi_{aktual})^2 + (\theta_{target} - \theta_{aktual})^2 + (\psi_{target} - \psi_{aktual})^2} \quad \text{(Persamaan 28)}$$

### 3.3. Hasil Pengujian

Berdasarkan hasil pengujian, akurasi gerakan robot dievaluasi menggunakan metrik deviasi translasi (dalam milimeter) dan deviasi orientasi (dalam derajat). Berikut adalah hasil analisis:

1. Deviasi Translasi :
  - a. Sebagian besar titik memiliki deviasi translasi yang kecil (<0.1 mm), menunjukkan bahwa robot berhasil mencapai target posisi dengan presisi tinggi.
  - b. Pada titik-titik tertentu, seperti [nomor titik dengan deviasi terbesar], ditemukan deviasi translasi lebih besar (misalnya, >0.5 mm). Hal ini kemungkinan disebabkan oleh:
    - Kesalahan dalam parameter model kinematika.
    - *Noise* pada sensor simulasi atau komunikasi perangkat lunak.
2. Deviasi Orientasi :
  - a. Hampir semua pengujian menunjukkan deviasi orientasi yang sangat kecil atau bahkan nol (0°), yang mengindikasikan bahwa kontrol orientasi end-effector robot berjalan dengan baik.
  - b. Pada [nomor titik dengan deviasi orientasi terbesar], terdapat deviasi rotasi hingga 202.03° (jika kasus tersebut ada). Hal ini disebabkan oleh perubahan drastis dalam orientasi yang mungkin salah terinterpretasi dalam simulasi atau model.

## 4. KESIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa:

1. Sistem pengendalian robot Mitsubishi RV-2SDB yang diimplementasikan melalui bantuan CIROS *Programming* berhasil memindahkan berpindah dari posisi referensi ke posisi target dengan tingkat presisi yang tinggi.
  - Rata-rata deviasi translasi pada pengujian kecil (<0.1 mm) menunjukkan keakuratan sistem dalam mencapai target posisi.
  - Rata-rata deviasi orientasi juga sangat kecil (0° pada sebagian besar kasus), yang mengindikasikan efektivitas kontrol orientasi *end-effector* robot.
2. Penggunaan simulasi dibantu dengan CIROS *Programming* memungkinkan pengujian dan evaluasi *forward kinematics* dan *inverse kinematics* secara efisien, meskipun terdapat keterbatasan dalam merepresentasikan *noise* fisik yang mungkin terjadi di perangkat keras secara *real-time*.
3. Sistem menunjukkan stabilitas baik selama pengujian, tanpa adanya kegagalan besar atau perilaku tak terduga, meskipun terdapat deviasi yang sedikit lebih besar pada beberapa posisi tertentu, kemungkinan disebabkan oleh:
  - *Noise* pada model simulasi.
  - Kesalahan kecil pada parameter model kinematika.

## REFERENCES

- Ahdiat, A. "Penggunaan Robot Industri Meningkatkan, Capai Rekor pada 2021", MANUFACTURER, 01 November 2022.
- Adi Ahdiat, "Penggunaan Robot Industri Meningkatkan, Capai Rekor pada 2021,"<https://databoks.katadata.co.id/datapublish/2022/11/01/penggunaan-robot-industri-meningkat-capai-rekor-pada-2021>.
- Bonci, A., Gaudeni, F., Giannini, M.C., & Longhi, S. (2023). *Robot Operating System 2 (ROS2) - Based Frameworks for Increasing Robot Autonomy*. Department of Information Engineering (DII), Polytechnic University of Marche, Italy.
- C. Han, H. Ma, W. Zuo, S. Chen and X. Zhang, "A General 6 DOF Industrial Robot Arm Control System Based on Linux and FPGA," 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 2018, pp. 1220-1225, doi: 10.1109/CCDC.2018.8407315.
- "CIROS Studio", <https://www.verosimsolutions.com/en/industry/ciros-studio/>
- "CIROS ® VR Virtual Reality Dimulation for Immersive Automation Training.", F. Didactic, Tersedia pada: [www.festodidactic.de](http://www.festodidactic.de).
- Endrika Decky Marselino, "Modul Praktikum Otomasi Pick And Place Robot Sebagai Pembelajaran Industrial Motion, Vision Sensor, Dan Web Server Gui", 2021, Teknik Mekatronika. Politeknik Elektronika Negeri Surabaya.
- Fabrice Boisset, "The History of Industrial Automation in Manufacturing,". <https://www.automate.org/editorials/the-history-of-industrial-automation-in-manufacturing>.
- Faudin, A. "Apa itu protokol komunikasi RS485?", Nyebar Ilmu, 16 November 2019. Dapat diakses <https://www.nyebarilmu.com> (diakses 15 Juli 2024).
- Info Solutech. "PENGERTIAN, FUNGSI, DAN CARA KERJANYA." Info Solutech, 1 November 2023. Dapat diakses dari <https://solutech.id/> (diakses 16 Juli 2024).
- Mitsubishi Electric Corporation. (n.d.). *RV-2SD/2SDB Instruction Manual: Robot Arm Setup & Maintenance*. BFP-A8791-D. Mitsubishi Electric Corporation.
- "MELFA RV-2SDB / RV-2SQB Series". <https://us.mitsubishielectric.com/fa/en/support/technicalsupport/knowledge-base/getdocument/?docid=3E26SJWH3ZZR-38-1447>.
- Papcun Peter, Jadlovský Ján, (2015), "Mathematical Model of Robot Melfa RV-2SDB", Technical University of Kosice, pp. 1-10.
- P. Bora and V. Nandi, "Low Cost Shadow Function Based Articulated Robotic Arm" 2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE), Shillong, India, 2015, pp. 1-4, doi: 10.1109/EPETSG.2015.7510079.
- Rasyidin, A. N. (2024). *Implementasi Gerak Robot Lengan Mitsubishi RV-2SDB (Palletizer) Menggunakan Metode Hierarchical Finite State Machine untuk Sistem Sortir Berdasarkan Ukuran Box*. Proyek Akhir, Program Studi Sarjana Terapan Teknik Mekatronika, Politeknik Elektronika Negeri Surabaya.
- Raza, N. "ABB robot arm (IRB 1200) with Siemens (S7 1200) PLC - can communicate via Ethernet/IP Anybus Adapter?", Siemens Support, 17 Juli 2019. Dapat diakses dari <https://support.industry.siemens.com/> (diakses 15 Juli 2024).
- Raza, N. "Controlling ABB robot arm (irb 1200) with Siemens PLC", Robot Forum, 19 Juli 2019. Dapat diakses dari <https://www.robot-forum.com/> (diakses 15 Juli 2024).
- Seung, J.R., and Kosuke, I. "Using Cost Based FMEA to Enhance Reliability and Serviceability", Journal of Advanced Engineering Informatics, Vol.17, pp. 179-188, 2003.
- "Siemens S7-1200 PLC CPU – 14 (Digital Input, 2 switch as Analogue Input) Input, 10 (Digital Output, Transistor Output)", <https://www.rs-online.id/p/simatic-s7-1200-cpu-1214c-dc-dc-dc/>
- Softscients. "Pengertian dan Penggunaan Modul I2C: Memahami Komunikasi Serial Antar-Perangkat", Softscients, 29 November 2023. Dapat diakses <https://softscients.com/> (diakses 15 Juli 2024).
- "Spesifikasi Arm Robot Mitsubishi MELFA RV-2SDB", Mitsubishi Electric, 2010, <https://us.mitsubishielectric.com/fa/en/support/technicalsupport/knowledge-base/getdocument/?docid=3E26SJWH3ZZR-38-1447>.
- Teloken, F. "ESP32 I2C Communication: Set Pins, Multiple Bus Interfaces and Peripherals (Arduino IDE)", Random Nerd Tutorials, 2 Oktober 2019. Dapat diakses <https://randomnerdtutorials.com/> (diakses 15 Juli 2024).
- Y. Zhang, X. Li, H. Zhu and Ning Tan, "Joint-Angle-Drift Remedy Of Three-Link Planar Robot Arm Performing Different Types Of End-Effector Trajectories" 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, 2009, pp. 581-585, doi: 10.1109/ICICISYS.2009.5358320.