

Pemodelan dan Implementasi Gerak Robot Manipulator dengan Menggunakan *Robotics System Toolbox* pada MATLAB

Muhammad Farouk Setiawan¹, Ainun Hasanah²

¹Departemen Teknik Mesin, Universitas Diponegoro, Indonesia

²Department of Urban & Rural Planning, School of Urban Design, Wuhan University, China

mfarouksetiawan@students.undip.ac.id¹, 2021172090001@whu.edu.cn²

Received 19 Juli 2023 | Revised 11 Agustus 2023 | Accepted 14 September 2023

ABSTRAK

Penggunaan robot manipulator dalam dunia industri sudah sangat tidak asing lagi. Namun, apabila melihat dari sudut pandang dunia pendidikan, pembelajaran tentang robot manipulator cenderung mengalami kesulitan yaitu dalam hal biaya pembuatan robot manipulator itu sendiri yang cenderung mahal. Pada penelitian ini bertujuan untuk melakukan pemodelan dan implementasi yang dapat memungkinkan pembelajaran terkait robot manipulator dapat dilakukan dalam dunia pendidikan. Metode yang digunakan dalam penelitian ini adalah dengan menggunakan salah satu *library* pada MATLAB yaitu *Robotics System Toolbox* dan kemudian mengimplementasikannya pada mikrokontroler Arduino Mega 2560 untuk dapat dijalankan pada prototipe robot manipulator. Hasil yang diperoleh bahwa simulasi dan implementasi gerak robot dengan menggunakan *Robotics System Toolbox* dapat terintegrasi secara langsung dan menghasilkan gerakan yang mampu mengikuti input berupa titik koordinat pada bidang x, y, dan z. Blok invers kinematik mampu dengan akurat dalam menghitung secara otomatis besar sudut tiap *joint* robot manipulator yang dapat memposisikan *end effector* robot pada titik koordinat lintasan gerak yang telah ditentukan.

Kata kunci: Robot Manipulator, Invers Kinematik, Robotic System Toolbox, MATLAB

The use of robot manipulators in the industrial world is very familiar. However, when looking from the point of view of education, learning about robot manipulators tends to experience difficulties, namely in terms of the cost of making the robot manipulator itself which tends to be expensive. This study aims to conduct modeling and implementation that can enable learning related to robot manipulators to be carried out in the world of education. The method used in this research is to use one of the libraries in MATLAB, namely the Robotics System Toolbox and then implement it on the Arduino Mega 2560 microcontroller to be run on a prototype robot manipulator. The results obtained are that the simulation and implementation of robot motion using Robotics System Toolbox can be integrated directly and produce movements that are able to follow input in the form of coordinate points in the x, y, and z axes. The inverse kinematics block is able to accurately calculate automatically the angle of each joint of the manipulator robot which can position the robot end effector at the coordinate point of the predetermined motion trajectory.

Keywords: Robot Manipulator, Inverse Kinematics, Robotic System Toolbox, MATLAB

I. PENDAHULUAN

Perkembangan yang sangat pesat dalam ilmu pengetahuan dan teknologi sejalan dengan meningkatnya kebutuhan masyarakat menjadikan banyak proses industri beralih dari sistem manual ke sistem otomatis. Peralihan ini menjadikan setiap proses industri lebih efektif dan efisien. Salah satu teknologi yang diterapkan dalam sistem otomatisasi ini adalah pemanfaatan robot manipulator. Dalam pemanfaatannya, robot manipulator sering digunakan untuk membantu dalam melakukan pekerjaan yang memiliki kecenderungan bahaya yang tinggi, kegiatan *loading* dan *unloading* komponen, proses yang memerlukan kepresisian yang tinggi dan konsistensi, dan lain sebagainya.

Dalam pembuatan robot manipulator di dunia industri tentunya dengan biaya yang tidak murah dikarenakan bahan pembuatan yang harus memiliki daya tahan tinggi terhadap berbagai kondisi serta memiliki gerakan yang fleksibel. Kemudian juga harus memiliki tingkat ketelitian yang tinggi dalam melakukan suatu pekerjaan. Di samping itu dalam dunia pendidikan minat mahasiswa yang tinggi untuk mempelajari terkait robot sehingga perlunya prototipe robot manipulator yang dapat dibuat dengan biaya yang relatif lebih murah, namun tidak menghilangkan konsep dasar dari gerak robot manipulator sesungguhnya.

Beberapa penelitian sebelumnya yang dilakukan dengan pada robot manipulator di antaranya seperti yang dilakukan Purwoto (2020) yaitu memodelkan gerak robot manipulator 4 derajat kebebasan dengan menggunakan GUI (*Graphic User Interface*) MATLAB yang di dalamnya dimasukkan persamaan kinematika robot. Penelitian yang dilakukan Cahyono et al., (2022) yaitu dengan menggunakan persamaan orde polinomial dalam perencanaan gerak robot yang juga menggunakan metode perhitungan dalam persamaan invers

kinematika. Penelitian yang dilakukan Ramadhan et al., (2021) yaitu merancang pengendalian gerak robot dengan menggunakan metode *fuzzy logic* untuk mengurangi tingkat eror yang terjadi pada penentuan posisi gerak robot dengan memasukkan persamaan *forward* kinematik. Kemudian pada penelitian yang dilakukan Bao et al., (2017) yaitu melakukan simulasi kinematik pada robot manipulator 4 DOF dengan menggunakan kombinasi metode aljabar dan metode geometri untuk persamaan invers kinematik yang kemudian disimulasikan di MATLAB. Berdasarkan pada beberapa penelitian tersebut, dapat dilihat bahwa dalam melakukan gerak robot manipulator masih dengan menggunakan cara manual dimana masih harus dengan memasukkan persamaan kinematika robot dengan perhitungan secara matematis.

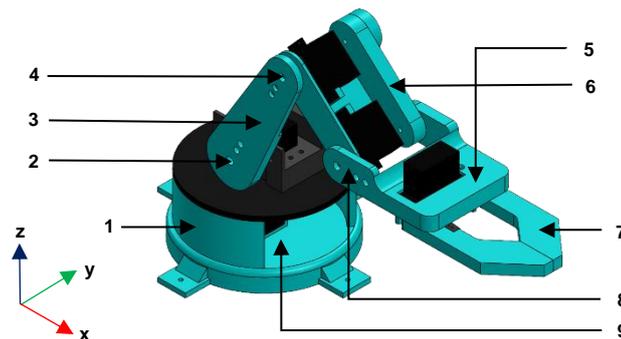
Adapun pada penelitian ini bertujuan untuk memodelkan gerak robot manipulator dengan menggunakan salah satu *library* pada MATLAB yaitu *Robotics System Toolbox* dan kemudian mengimplementasikannya pada mikrokontroler Arduino Mega 2560 untuk dapat dijalankan pada prototipe robot manipulator. Yang berbeda dari penelitian sebelumnya adalah dimana pada penelitian ini tidak akan dilakukan perhitungan secara matematis persamaan kinematika yaitu untuk *forward* kinematik dan invers kinematik robot yang mana merupakan persamaan yang paling penting untuk mendesain gerak robot manipulator, akan tetapi semuanya akan ditangani secara otomatis oleh *library* ini. Adapun dalam penggunaan *library* ini pada robot adalah untuk dapat memudahkan dalam proses simulasi gerak robot dan implementasinya pada mikrokontroler Arduino Mega 2560 yang dapat terintegrasi secara langsung. Prototipe robot manipulator ini memiliki 4 *Degree of Freedom* (DOF) yang dilengkapi dengan 4 buah motor servo sebagai penggerakannya dan *gripper* sebagai *end effector*. Sehingga dalam penelitian pada prototipe robot manipulator ini dapat digunakan untuk memberikan gambaran sistem gerak robot yang dapat disimulasikan dan diimplementasikan sebagai bahan pembelajaran dalam dunia pendidikan.

II. METODE PENELITIAN

A. Desain Robot Manipulator

Robot manipulator adalah seperangkat alat mekanik berupa robot lengan yang dapat diprogram dan didesain agar bisa melakukan suatu tugas fisik dengan spesifik tertentu melalui variabel gerakan yang terprogram dan juga dapat dikontrol atau digerakkan oleh manusia. Pada dasarnya, robot manipulator merupakan sebuah mekanisme yang disusun oleh bagian tubuh yang kaku yang dapat dikatakan sebagai *link*, dan yang menghubungkan antar-*link* dinamakan *joint* (Lynch & Park, 2017). Terdapat dua jenis *joint* robot, yaitu *revolute joint* yang memiliki pergerakan memutar dan *joint* prismatic yang memiliki pergerakan maju atau mundur. Agar robot bergerak dan mencapai titik yang ditentukan pada koordinat kartesius, diperlukan suatu model atau metode yang menghubungkan antara posisi dan orientasi robot pada koordinat kartesius dengan *joint* robot (Whitney et al., 1986). Robot manipulator memiliki 2 sisi yang digunakan dalam melakukan pergerakan tersebut. Satu sisi adalah *base* yang ditanam pada bidang yang diam (statis). Dan sisi yang lain bebas bergerak yang disebut dengan (*end effector*) yang pada umumnya dapat dimuati dengan tool tertentu sesuai dengan tugas robot (Corke, 2017).

Adapun robot manipulator yang digunakan pada penelitian ini adalah robot manipulator 4 DOF yang mana memiliki 4 *joint* yang berputar dengan *gripper* sebagai *end effector*-nya. Adapun desain robot manipulator dapat dilihat pada gambar 1.



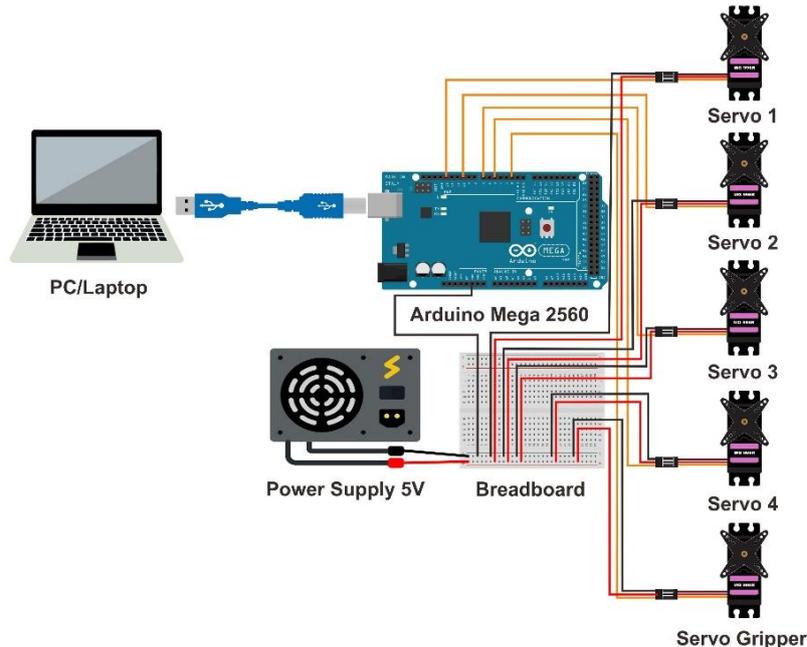
Gambar 1. Desain Robot Manipulator 4 DOF

Adapun penjelasan keterangan pada gambar 1 adalah sebagai berikut.

- | | |
|-------------|------------------|
| 1 : Base | 6 : Link 3 |
| 2 : Joint 2 | 7 : End Effector |
| 3 : Link 1 | 8 : Joint 4 |
| 4 : Joint 3 | 9 : Joint 1 |
| 5 : Link 4 | |

B. Rangkaian Skematik Perangkat Keras

Dalam gerak robot manipulator digunakan mikrokontroler Arduino Mega 2560 yang menggerakkan 5 buah motor servo MG996R, dimana 4 buah pada aktuator joint robot dan 1 buah pada aktuator *gripper* melalui program input yang dimasukkan dari PC/laptop. Adapun sebagai sumber catu daya yaitu berasal dari *power supply* yang telah diatur pada tegangan output sebesar 5V. Kemudian perangkat keras tambahan yaitu *breadboard* yang digunakan sebagai perantara penyaluran sumber tenaga listrik dari *power supply* agar tidak langsung masuk ke mikrokontroler dan motor servo, dan kabel USB to type-B sebagai penghubung antara mikrokontroler dan PC/laptop. Untuk lebih mudah dalam memahaminya dapat dilihat pada gambar 2.



Gambar 2. Rangkaian Skematik Perangkat Keras

Program input yang dimasukkan ke dalam Arduino Mega 2560 diteruskan ke motor servo dari pin digital PWM melalui kabel sinyal (warna oranye). Kemudian sumber catu daya memberikan tenaga listrik + dan - ke *breadboard* yang kemudian juga untuk tiap motor servo kabel + (warna merah) dan - (warna hitam) dihubungkan pada jalur yang sama dengan + dan - power supply untuk mendapatkan arus langsung dari *power supply*. Untuk dapat terhubung antara PC/Laptop dan motor servo, ground (-) dari Arduino Mega 2560 juga harus dihubungkan pada jalur yang sama dengan ground (-) *power supply* dan motor servo.

C. Desain Gerak Robot Manipulator

Dalam gerak robot manipulator diperlukan konsep kinematika. Kinematika adalah ilmu yang mempelajari terkait gerak yang dalam implementasinya tanpa memperhatikan pengaruh gaya yang menyebabkan pergerakan tersebut. Dalam kinematika berkaitan erat dengan posisi dan kecepatan gerak robot yang mempertimbangkan terhadap bentuk geometri robot dan sifat geraknya terhadap waktu (Craig, 1989). Pada jenis robot manipulator, umumnya adalah berupa *open kinematic chain* dimana satu sisi robot adalah statis dan sisi lainnya dinamis. Kemudian setiap *joint* biasanya didefinisikan dalam satu variabel yaitu variabel sudut (θ), sehingga jumlah dari *joint* sama dengan jumlah DOF robot tersebut.

Sisi robot yang statis seringkali dikatakan sebagai *base*, sedangkan sisi lainnya yang dinamis adalah *end effector*. Secara umum posisi dari robot didefinisikan dengan memberikan deskripsi *end effector* yang relatif terhadap *base*. Sehingga untuk memperoleh definisi tersebut adalah dengan memberikan sekumpulan sudut *joint*. Untuk menentukan hubungan posisi dan sudut *joint* ini dikenal metode *forward* kinematik dan *invers* kinematik.

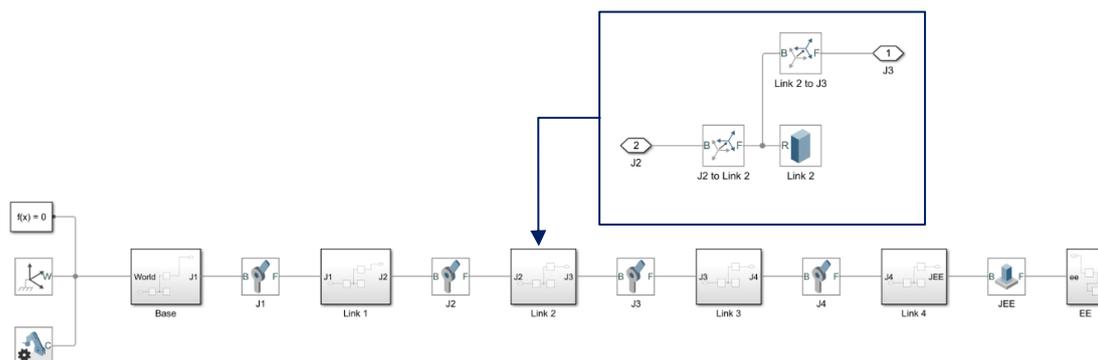
Forward kinematik adalah metode untuk menentukan orientasi dan posisi *end effector* dari besarnya sudut sendi dan panjang *link* pada robot manipulator. Pada penentuan *forward* kinematik digunakan konsep transformasi homogen dimana berupa penggabungan matriks transformasi R (rotasi), P (posisi atau gerak translasi) dan faktor skala dalam satu matriks. Sehingga konsep ini dapat memberikan informasi terkait dengan orientasi dan posisi *end effector* robot manipulator yang digunakan untuk menggunakan metode *forward* kinematik. Sedangkan *invers* kinematik adalah metode yang digunakan untuk menentukan besar sudut pada *joint* robot manipulator dengan diketahui posisi dan orientasi *end effector* (Frisyras et al., 2018). Adapun dalam

perhitungan dengan metode invers kinematik dapat dilakukan dengan menggunakan pendekatan secara aljabar dan pendekatan geometris. Pendekatan secara aljabar adalah berkaitan dengan menghitung invers kinematik dari hubungan parameter-parameter pada notasi *Denavit-Hartenberg*. Sedangkan pendekatan geometris adalah berkaitan dengan menghitung invers kinematik pada hubungannya pada bidang kartesius. Akan tetapi tidak seperti forward kinematik, invers kinematik cenderung lebih sulit dalam perhitungannya yang disebabkan karena adanya singularitas dan nonlinear pada persamaannya (Kucuk & Bingul, 2006). Oleh karena itu, hanya untuk robot manipulator dengan jumlah DOF yang tidak banyak dapat ditemukan solusi secara analitis untuk metode invers kinematik (Küçük & Bingül, 2004).

Untuk menggerakkan robot manipulator pada penelitian ini digunakan Simulink MATLAB. Simulink disini akan menjadi *workspace* untuk melakukan simulasi dan implementasi gerak pada robot manipulator. Dalam gerak robot manipulator ini nantinya akan diberikan tugas pada robot manipulator untuk melakukan gerak untuk mencapai posisi yang telah ditentukan. Dimana nantinya akan diberikan input berupa titik koordinat pada sumbu x, y, dan z yang mengacu pada posisi *end effector*. Berdasarkan hal ini akan dilihat apakah gerak pada hasil simulasi akan sama dengan yang terjadi secara *real time*.

Konsep yang diberikan untuk gerak disini adalah dengan menggunakan metode invers kinematik. Namun disini tidak akan dilakukan perhitungan matematis secara manual bagaimana untuk mendapatkan nilai sudut dari setiap *joint* agar dapat mencapai titik koordinat yang telah ditentukan. Akan tetapi semua perhitungan tersebut akan dilakukan oleh *Robotic System Toolbox* pada *library* MATLAB yang di dalamnya sudah memberikan fungsi invers kinematik. Jadi nantinya hanya diperlukan titik koordinat posisi sebagai input yang akan dimasukkan pada diagram blok tersebut beserta dengan informasi terkait dimensi robot dan posisi robot.

Dalam hal ini informasi terkait dengan robot harus didefinisikan terlebih dahulu yaitu dengan menggunakan *Simscape Multibody* yang merupakan salah satu *library* yang juga ada di MATLAB. Adapun terkait dengan dimensi robot yang didefinisikan disini adalah berkaitan dengan posisi *joint*, panjang *link* dan juga posisi *frame* pada sumbu x, y, dan z. Untuk diagram blok yang mendefinisikan dimensi robot pada Simulink dapat dilihat pada gambar 3.



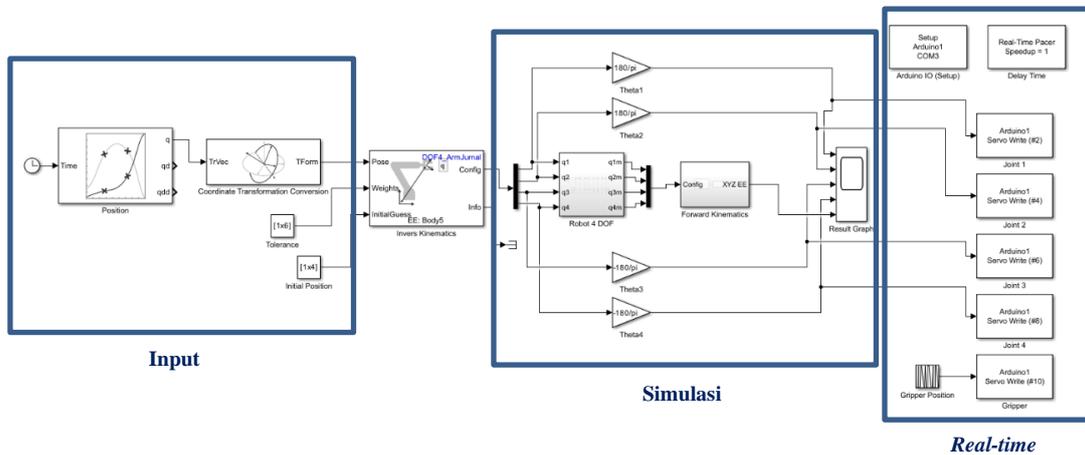
Gambar 3. Desain Robot Simulasi

Berdasarkan pada gambar 3, setiap *link* dan *joint* dari robot didefinisikan satu per satu dengan mendefinisikan hubungan dari setiap *link* dan *joint* terhadap yang lainnya. Pada *Simscape Multibody*, definisi hubungan ini juga memberikan informasi awal posisi robot (kondisi normalisasi) yaitu dimana pada posisi tersebut semua *joint* akan bernilai 0. Dalam hal ini, umumnya kondisi normalisasi yang diberikan adalah dengan memposisikan robot dengan posisi lurus secara horizontal. Namun untuk penentuan kondisi normalisasi seperti ini tidak baku, semuanya nanti akan bergantung dari keadaan robot terhadap lingkungan di sekitarnya dan juga untuk memudahkan dalam menghitung besar sudut setiap *joint*, serta mempertimbangkannya dengan posisi peletakkan motor servo.

Untuk dapat menjalankan robot, input yang dimasukkan sebagai posisi yang harus dilewati oleh robot. Posisi yang diberikan yaitu terdiri dari titik awal, titik lintasan pada detik ke 3, 5 dan 7, dan titik akhir. Titik awal yang diberikan adalah titik dimana posisi robot dalam keadaan *standby*, titik lintasan merupakan titik-titik yang harus dilewati oleh robot, dan titik akhir adalah titik dimana robot kembali ke posisi *standby*. Adapun untuk lebih jelasnya terkait dengan posisi titik yang diberikan beserta pada waktu posisi tersebut berjalan dapat dilihat pada tabel 1. Kemudian juga dapat dilihat pada gambar 4 untuk diagram blok sistem gerak robot manipulator secara lengkap.

Tabel 1. Titik Posisi Gerak Robot dan Waktu

Posisi Benda (m)	x	y	z
Titik Awal (t = 0 detik)	0.012	0	0.446
Titik (t = 3 detik)	0.18	0.18	0.15
Titik (t = 5 detik)	0.22	0.11	0.17
Titik (t = 7 detik)	0.25	0	0.15
Titik Akhir (t = 11 detik)	0.012	0	0.446



Gambar 4. Diagram Blok Gerak Robot Manipulator

Pada bagian input yang dimasukkan sebagai posisi dari benda dalam satuan meter (m). Setelah itu sebelum masuk ke dalam blok invers kinematik perlu dikonversikan menjadi bentuk transformasi homogen terlebih dahulu dengan blok *Coordinate Transformation Conversion* agar dapat diproses. Setelah input posisi benda, toleransi dimasukkan ke input *weights* dan posisi awal robot dimasukkan ke input *initial guess*. Toleransi yang dimasukkan disini adalah berupa matriks dengan ukuran 1x6 dimana tiga angka bagian pertama akan mendefinisikan toleransi terhadap orientasi, dan tiga angka berikutnya mendefinisikan toleransi terhadap posisi *end effector* robot. Sedangkan masukan pada posisi awal robot adalah didefinisikan dalam matriks dengan ukuran 1x(jumlah joint yang bergerak dalam robot) dengan definisi setiap angka adalah posisi sudut tiap *joint* dalam satuan rad. Namun, hal ini tentunya akan berkaitan dengan bagaimana kondisi normalisasi robot sebelumnya. Pada penelitian ini menggunakan kondisi normalisasi pada posisi lurus horizontal. Akan tetapi, pada kondisi awal robot didefinisikan pada posisi robot lurus vertikal. Hal ini bertujuan agar selain lebih mudah untuk menghitung besar sudut tiap *joint*, juga mengurangi beban yang diberikan pada motor servo saat kondisi *standby*. Kemudian input akan diproses dan diteruskan ke bagian simulasi. Sebagai catatan pada blok invers kinematik Solver Paramater yang digunakan adalah *Levenberg-Marquardt* dan *Simulate Using* yang digunakan adalah *Code Generation*. Dalam hal ini output dari blok invers kinematik adalah posisi sudut setiap joint robot (dalam satuan: rad). Untuk melakukan simulasi, output ini akan digunakan kembali dengan memasukkan ke dalam desain robot simulasi sebelumnya yang kemudian akan dikonversikan kembali ke posisi x, y, dan z dengan menggunakan blok *forward* kinematik. Sehingga nantinya dapat diketahui juga tingkat keakuratan dari hasil simulasi penggunaan blok invers kinematik. Untuk pembahasan lebih detail mengenai konsep simulasi robot dengan metode ini dapat dilihat pada video tutorial (Abbas, M. R, 2022). Sedangkan untuk mengimplementasikan secara *real-time*, digunakan Arduino IO Package yang dihubungkan dengan output yang dihasilkan dari blok invers kinematik sebelumnya. Namun perlu dikonversikan output tersebut ke dalam satuan derajat agar menyesuaikan dengan nilai input yang dapat diterima pada blok Arduino IO. Pada bagian blok Arduino IO, setiap bagian yang ingin digerakkan seperti *joint* dan *gripper* akan diberikan masing-masing satu blok yang mendefinisikan pin Arduino beserta *sample time* dalam pembacaan programnya. Kemudian juga ada blok tersendiri yang diberikan untuk mendefinisikan *port* Arduino dan juga *delay time* pada pembacaan program. Sebagai catatan output hasil perhitungan pada blok invers kinematik sebelum masuk ke blok Arduino IO harus dilakukan penyesuaian dengan kondisi normalisasi robot agar tidak terjadi ketidaksesuaian gerak robot antara simulasi dan *real-time*.

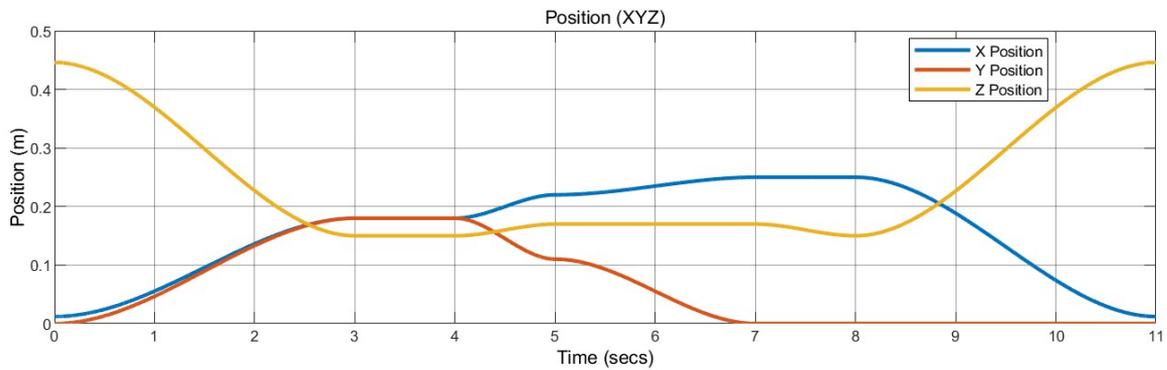
III. HASIL DAN PEMBAHASAN

Dengan melakukan proses *running* pada Simulink MATLAB pada rentang waktu t = 0 detik sampai t = 11 detik diperoleh hasil simulasi dan implementasi yang dijalankan secara bersama-sama berdasarkan pada

diagram blok *Simulink* gerak robot pada gambar 4. Adapun hasil pengujian gerak robot manipulator adalah dapat diuraikan sebagai berikut.

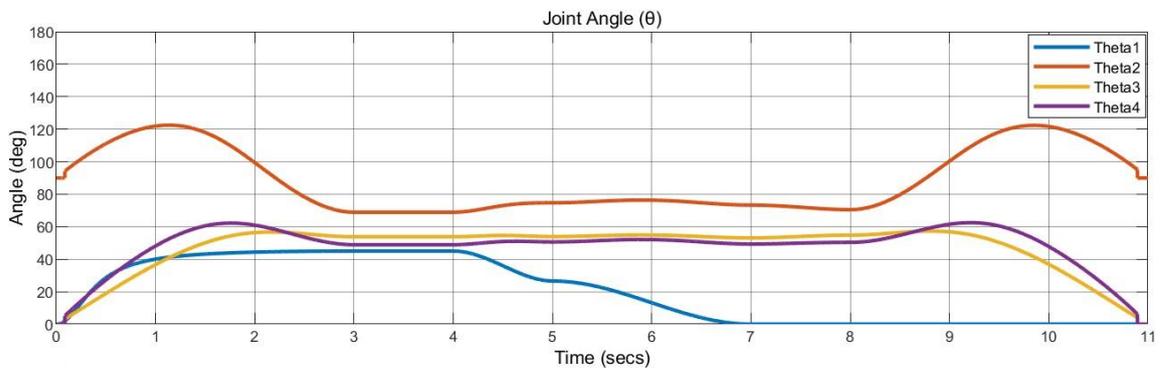
A. Simulasi Gerak Robot pada Simulink MATLAB

Pada hasil simulasi gerak robot ditunjukkan gerakan robot yang mengikuti posisi yang sudah ditentukan sebelumnya. Hal ini ditunjukkan dari hasil grafik posisi yang merupakan output dari blok *forward* kinematik sebelumnya yang dapat dilihat pada gambar 5.



Gambar 5. Grafik Hasil Posisi dari Simulasi Gerak Robot

Pada grafik dapat dilihat bahwa pada posisi awal atau *standby* ($t = 0$ detik), posisi x berada pada sekitar 0.012, posisi y berada pada 0, dan z berada pada sekitar 0.446. Kemudian pada $t = 6$ detik dapat dilihat posisi x berada pada sekitar 0.24, posisi y berada pada 0.055, dan posisi z berada pada 0.17. Sedangkan pada posisi akhir ($t = 11$ detik), posisi robot kembali lagi ke posisi *standby* yaitu pada posisi x berada pada sekitar 0.012, posisi y berada pada 0, dan z berada pada sekitar 0.446. Hal ini menunjukkan bahwa blok invers kinematik dari *Robotics System Toolbox* dapat menghasilkan gerak robot yang dapat secara otomatis menghitung besar setiap *joint* robot untuk dapat mencapai titik yang telah ditentukan. Adapun besar sudut yang dihasilkan dari blok invers kinematik tersebut adalah dapat dilihat pada gambar 6.



Gambar 6. Grafik Hasil Besar Sudut *Joint* dari Simulasi Gerak Robot

Berdasarkan pada hasil sudut yang diberikan oleh blok invers kinematik, dapat dilihat bahwa setiap *joint* bergerak dari posisi awal robot yang telah ditentukan pada titik awal ($t = 0$ detik) dan *initial guess* sebelumnya, kemudian bergerak menuju posisi titik-titik yang telah ditentukan. Sebagai catatan bahwa antara penentuan titik awal pada $t = 0$ detik pada sumbu x, y, dan z harus sesuai dengan pada *initial guess* yang diberikan. Dalam penelitian ini, posisi awal robot didefinisikan pada posisi robot dengan semua *joint* lurus vertikal yaitu pada posisi $x = 0.012$, $y = 0$, dan $z = 0.446$, serta pada posisi sudut awal yaitu $\theta_1 = 0$, $\theta_2 = 1.57$, $\theta_3 = 0$, dan $\theta_4 = 0$ (dalam satuan : rad). Sehingga dapat dilihat bahwa gerak yang dihasilkan setiap *joint* berubah secara bertahap untuk dapat mencapai posisi yang telah ditentukan sebelumnya. Dapat dilihat bahwa pada sudut θ_1 , θ_3 , dan θ_4 , pada kondisi awal berada pada sudut 0 derajat. Sedangkan pada sudut θ_2 , pada kondisi awal berada pada sudut 90 derajat. Hal ini menunjukkan bahwa hasil simulasi memberikan informasi sudut yang sesuai dengan input posisi awal yang sudah dimasukkan sebelumnya. Namun dapat dilihat pada grafik, terdapat nilai sudut yang bernilai negatif. Hal ini dikarenakan pada Simulink MATLAB mendefinisikan posisi sumbu

putar yaitu berada pada sumbu z dan gerakan joint apabila gerakannya mengarah berlawanan arah jarum jam maka akan bernilai positif. Sedangkan apabila gerakannya mengarah searah jarum jam maka akan bernilai negatif. Berdasarkan pada definisi ini nantinya perlu dipertimbangkan hubungannya dengan bagaimana posisi motor servo pada implementasi robot.

B. Implementasi Gerak Robot pada Robot Manipulator

Berdasarkan hasil simulasi dapat dilihat bahwa robot manipulator dapat bergerak untuk dapat mencapai posisi yang diinginkan. Pada implementasi juga demikian ditunjukkan bahwa robot mampu untuk bergerak seperti halnya yang dilakukan dalam simulasi yaitu robot menerima input dari mikrokontroler yang berasal dari hasil perhitungan blok invers kinematik yang kemudian diteruskan ke motor servo untuk dapat menjalankan robot tersebut. Adapun implementasi gerak robot dapat dilihat pada gambar 7.



Gambar 7. Hasil Implementasi Gerak Robot Manipulator

Pada gambar 7, ditunjukkan bahwa antara gerak simulasi dan gerak secara *real-time* dapat terintegrasi secara langsung. Dimana saat simulasi dijalankan, robot manipulator juga akan bergerak secara bersama-sama dengan simulasi. Namun, dalam implementasi ini perlu diperhatikan bahwa pada saat kondisi *running* baru saja dijalankan, robot akan melakukan kalibrasi terhadap sudut 0 derajat pada setiap *joint*-nya terlebih dahulu. Setelah itu, robot akan memposisikan pada posisi sudut awal (*intial guess*) dan melanjutkannya untuk bergerak mengikuti posisi yang telah ditentukan.

IV. KESIMPULAN DAN SARAN

Adapun kesimpulan dari penelitian ini adalah diperoleh bahwa hasil simulasi dan implementasi gerak robot dengan menggunakan *Robotics System Toolbox* dapat terintegrasi secara langsung dan menghasilkan gerakan yang mampu mengikuti input berupa titik koordinat pada bidang x, y, dan z. Blok invers kinematik mampu dengan akurat dalam menghitung secara otomatis besar sudut tiap *joint* robot manipulator yang dapat memposisikan *end effector* robot pada titik koordinat lintasan gerak yang telah ditentukan yaitu sebagai contoh besar sudut pada kondisi awal menunjukkan bahwa untuk besar sudut θ_1 , θ_3 , dan θ_4 adalah 0 derajat. Sedangkan besar sudut θ_2 adalah 90 derajat. Hal ini menunjukkan bahwa hasil simulasi dan implementasi memberikan informasi sudut yang sesuai dengan input posisi awal yang sudah dimasukkan sebelumnya. Sehingga hal ini tentunya dapat mempermudah dalam proses perhitungan invers kinematik yang pada umumnya memiliki tingkat kesulitan yang tinggi apalagi dalam pengaplikasian robot dengan jumlah DOF yang banyak. Untuk penelitian selanjutnya, dapat ditambahkan input berupa sensor ataupun penggunaan kamera sebagai pengganti input posisi yang dimasukkan pada robot. Sehingga robot dapat bergerak menyesuaikan pada kondisi lingkungan di sekitarnya.

DAFTAR PUSTAKA

- Abbas, M. R. (2022). *Robotics: A Practical Approach Based on Modern Tools*. Retrieved from Learning Orbis website: https://www.youtube.com/playlist?list=PLWF9TXck7O_ymYWT8Q33omPb5K-A5v4Ae.
- Bao, G., Liu, S., & Zhao, H. (2017). *Kinematics Simulation of 4-DOF Manipulator*. 2nd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE 2017), Vol. 123, 400-408.

- Cahyono, G. R., Nurmahaludin, Setiawan, M. F., Rizal, Y., & Riadi, J. (2022). *Comparison of 4 DOF Arm Robot for Trajectory Planning with 3rd and 5th Polynomial Orders*. Proceedings - 11th Electrical Power, Electronics, Communications, Control, and Informatics Seminar, EECCIS 2022, 281–286. <https://doi.org/10.1109/EECCIS54468.2022.9902924>
- Corke, P. (2017). *Robotics, Vision and Control* (Vol. 118). Springer International Publishing. <https://doi.org/10.1007/978-3-319-54413-7>
- Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control (Second Edition)*. USA : Addison-Wesley.
- Frisyras, E. K., Moulianitis, V. C., & Aspragathos, N. A. (2018). *ANNs to approximate all the inverse kinematic solutions of non-cuspidal manipulators*. IFAC-PapersOnLine, 51(22), 418–423. <https://doi.org/10.1016/j.ifacol.2018.11.586>
- Kücüük, S., & Bingül, Z. (2004). *The Inverse Kinematics Solutions of Industrial Robot Manipulators*. Proceedings of the IEEE International Conference on Mechatronics 2004, ICM'04, 274–279. <https://doi.org/10.1109/icmech.2004.1364451>
- Kucuk, S., & Bingul, Z. (2006). *Robot Kinematics: Forward and Inverse Kinematics, In Industrial Robotics: Theory, Modelling and Control*. Pro Literatur Verlag, Germany / ARS, Austria. <https://doi.org/10.5772/5015>
- Lynch, K. M., & Park, F. C. (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge : Cambridge University Press.
- Purwoto, B. H. (2020). *Pemodelan Robot Kinematik Manipulator Menggunakan Matlab*. Emitor : Jurnal Teknik Elektro, 20(2), 141-146.
- Ramadhan, D. R., Al Tahtawi, A. R., & Wijayanto, K. (2021). *Kendali Posisi Robot Lengan pada Misi Pick and Place dengan Metode Fuzzy Logic Control*. Prosiding The 12th Industrial Research Workshop and National Seminar Bandung, 96-102.
- Whitney, D. E., Lozinski, C. A., & Rourke, J. M. (1986). *Industrial Robot Forward Calibration Method and Results*. Journal of Dynamic Systems, Measurement, and Control, 108(1), 1–8. <https://doi.org/10.1115/1.3143737>