

Pemodelan Robot Kinematik Manipulator Menggunakan Matlab

Bambang Hari Purwoto, Devita Rindhani Rhokhim, Dwi Indraswari

Program Studi Teknik Elektro

Universitas Muhammadiyah Surakarta

Surakarta, Indonesia

Email: bambang.hari@ums.ac.id

Abstraksi—Manipulator robot memiliki 4 DOF (*Degree of Freedom*), semua sambungan bersifat *revolute* serta dilengkapi dengan pointer di *end-effector*. Proyek ini bertujuan untuk menganalisis kinematika maju dan terbalik dari robot manipulator yang didesain dengan menggunakan *SolidWorks* untuk mendapatkan model 3-D dan menggunakan *Simscape Multibody Link* generasi pertama untuk mengekspor CAD agar mendapatkan format file XML dan STL. Kemudian memersekusi file XML dengan Jendela perintah MATLAB untuk memperoleh diagram blok robot di *Simulink*. Mengembangkan GUI (*Graphical User Interfaces*) menggunakan Matlab untuk mengendalikan lengan dan menentukan kinematika maju dan terbalik. Pertama, membuat skema robot yang akan digambar dengan semua dimensi yang berlabel. Selanjutnya, tetapkan bingkai koordinat ke diagram robot sesuai dengan algoritme DH (*Denavit-Hartenberg*) dan kinematika maju lengkap akan dianalisis pada robot menggunakan persamaan *Homogeneous Transformation Matrix* yang memerlukan notasi. Hasil akhir dari *Homogeneous Transformation Matrix* akan menghasilkan orientasi dan posisi dari *end-effector*. Selain itu, analisis kinematika terbalik juga akan dilakukan pada dua titik yang diasumsikan, misalnya, mengangkat dan menempatkan dalam ruang kerja dengan menggunakan simulasi GUI sehingga akan diperoleh nilai sudut dari setiap *joint*. Dalam penelitian akan dibahas manipulator robotik artikulasi 6 sumbu dengan gerak 6 derajat kebebasan (DOF) yang dirancang dan dibangun dari simulasi. Deskripsi matematis untuk kekuatan dan posisi gabungan disertakan serta penggunaan perangkat lunak untuk mengendalikan kekuatan yang diterapkan pada sambungan dan posisi sambungan.

Kata Kunci—Algoritme DH (*Denavit-Hartenberg*), *homogeneous transformation matrix*, kinematika robot, Matlab GUI, *simscape multibody link*

I. LATAR BELAKANG

Perkembangan ilmu pengetahuan dan teknologi, khususnya di bidang elektronik, telah berkembang pesat. Perkembangan ini sejalan dengan meningkatnya permintaan masyarakat akan barang berkualitas tinggi yang diproduksi oleh industri. Untuk mendukung kualitas tinggi, banyak proses industri beralih dari sistem manual ke sistem otomatis sehingga peran manusia semakin kecil. Salah satu manfaat robot yang dapat menggantikan

peran manusia adalah kerja yang berkaitan dengan kegiatan berulang.

Robotika adalah ilmu teknik khusus yang berkaitan dengan robot dalam hal: desain, pemodelan, pengendalian, dan pemanfaatan. Saat ini robot menemani orang dalam kehidupan sehari-hari dan telah mengambil alih beberapa prosedur sehari-hari mereka hingga yang sangat canggih yang diperlukan untuk eksplorasi ruang angkasa. Beberapa kerja yang umum dijumpai saat ini adalah peran robot untuk proses pengelasan busur, lukisan semprot, perakitan, pemilihan lokasi, pemotongan, penggilingan, pengeboran dan lainnya. Dari kelas peralatan ini, jenis yang semakin populer adalah robot industri. Konfigurasi manipulator yang berbeda tersedia seperti persegi panjang, silinder, bola, *revolute* (putar) dan horizontal bersendi. Salah satu jenis robot yang digunakan dalam dunia industri adalah robot lengan. Dengan semakin pentingnya peran robot lengan di dunia industri, banyak siswa tertarik untuk belajar robot lengan. Sementara itu, membuat robot lengan dalam skala dunia industri tidaklah murah. Untuk alasan ini, kita memerlukan simulator lengan robot yang dapat digunakan untuk menentukan arah dan pergerakan efektor akhir. Untuk menentukan arah dan pergerakan robot lengan maka dalam penelitian mengadakan analisis kinematika maju dan terbalik. Kinematika maju adalah proses menghitung orientasi dan posisi *end-effector* berdasarkan sudut pada sambungan. Sedangkan kinematika terbalik adalah kebalikannya yaitu mengingat posisi *end-effector* dengan menghitung berapa banyak sudut yang harus diubah untuk setiap sambungan untuk dapat mencapai posisi *end-effector*.

Maka dalam penelitian PID akan melakukan proses rancang dan membuat simulator lengan robot 4 DOF (*Degree of Freedom*) yang merupakan representasi dari robot lengan untuk industri yang nantinya dapat digunakan untuk sektor Pendidikan. Simulator lengan robot ini dirancang menggunakan aplikasi desain 3D *SolidWorks* dan diekspor menggunakan generasi pertama *Simscape Multibody Link* untuk mendapatkan format file .XML dan membukanya di *Simulink* Matlab untuk mendapatkan diagram blok dari lengan robot. Simulasi lengan robot berdasarkan GUI (*Graphic User Interface*) Matlab guna mengontrol dan menentukan posisi *end effector* dan sudut setiap sambungan [1].

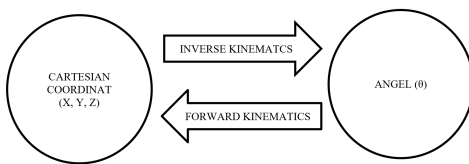
II. TINJAUAN PUSTAKA

Penelitian tentang analisis maju dan inversi kinematika robot lengan telah banyak dipelajari salah satunya adalah

Gouasmi yang berjudul "Pemodelan Kinematika dan Simulasi Robot 2-R Menggunakan SolidWorks dan Verifikasi oleh MATLAB / Simulink," [2]. Penelitian tersebut diperkuat hasilnya oleh peneliti lainnya [3]. Penelitian ini menjelaskan analisis kinematika maju dan mundur dari robot 2 DOF menggunakan *SolidWorks* dan pengerjaan program dengan Simulink Matlab. Proyek dilakukan dengan membandingkan dua gerakan robot dengan sudut lintasan yang sama untuk waktu yang sama dan membuat kode komputasi untuk mendapatkan parameter kinematik yang dinamis seperti yang dilakukan oleh Cheah [4].

Penelitian lain yang terkait dengan analisis kinematika maju dan terbalik dari robot lengan dilakukan oleh Toquica berjudul "Pemodelan Kinematik Manipulator Lengan Robot Menggunakan MATLAB," [5]. Penggunaan Matlab terbukti juga dapat menyelesaikan proses biometrik lewat tapak tangan [6]–[10]. Temuan penelitian ini juga memperkuat hasil luaran dari peneliti sebelumnya [11]. Penelitian ini juga tentang kinematika robot lengan, tetapi proyek yang dikerjakan menggunakan *Robotics Toolbox System* milik MATLAB untuk mengevaluasi dan mensimulasikan kinematika robot. Selanjutnya proyek ini diteruskan dengan membangun prototipe lengan robot dengan menggunakan Arduino sebagai mikrokontroler sebagaimana yang telah dilakukan oleh Kolesnyk [12] dan Goncalves [13].

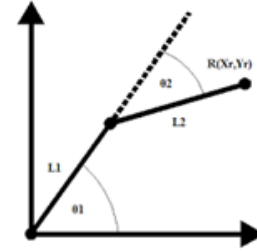
Selain itu, ada juga dari *IEEE International Journal* oleh Zhao yang berjudul "Model dan Simulasi Robot Mitsubishi RV-M1 menggunakan MATLAB" [14]. Temuan penelitian tersebut memperkuat hasil penelitian sebelumnya dari [15] dan [16]. Penelitian tersebut terkait dengan pemodelan kinematika robot, kinematika maju dan mundur dari manipulator robot bernama RV-M1 sebagaimana yang telah dilakukan oleh Dahari [17]. Model kinematik maju didasarkan pada parameter DH, dan mundur dihasilkan dari turunan tertutup. Selanjutnya dalam memahami teori lengan robot, berikut diuraikan beberapa ulasan perihal kinematika lengan robot, dan teori yang mendasari kinetik.



Gambar 1. Konsep kinematika maju dan mundur

A. Kinematika Lengan Robot

Robot kinematika adalah studi tentang analisis gerakan robot lengan terhadap sistem kerangka koordinat referensi yang diam terlepas dari gaya yang menyebabkan gerakan. Seperti penggunaan kinetik robot dengan model *fuzzy* [18] maka model kinematika mewakili hubungan *end-effector* di ruang tiga dimensi dengan sambungan variabel di ruang kirim. Persamaan kinematika maju menggambarkan posisi dan orientasi efektor akhir yang dinyatakan dalam posisi sambungan. Sedangkan persamaan kinematika mundur menggambarkan konfigurasi posisi sambungan untuk menghasilkan posisi dan orientasi tertentu dari efektor akhir. Ada dua konsep kinematika yaitu kinematika maju dan kinematika mundur/terbalik [19] seperti yang terlihat



Gambar 2. Vektor proyeksi Robot DOF

dalam Gambar 1. **Kinematika Maju.** *Forward Kinematics* adalah metode untuk menentukan orientasi dan posisi *end-effector* dan biasanya selalu sudut sambungan dan panjang lengan robot. Maju persamaan kinematika diperoleh berdasarkan jumlah DOF dan jenis manipulator robot rantai kinematika. **Kinematika Terbalik/mundur.** *Reverse Kinematics* adalah merupakan Kinematika terbalik yang lebih umum digunakan dalam lengan robot karena penggunaan robot yang sebenarnya karena pengaturan sambungan-sendi tidak lagi diutamakan. Fokus utama adalah bagaimana efektor akhir mencapai posisi objek dengan baik berdasarkan penempatan referensi koordinat yang telah ditentukan.

Kinematika lengan robot 4 DOF menggunakan persamaan trigonometri seperti penelitian yang telah dikerjakan oleh Mohammed [20] dan Emmanouil [21]. Jika ada manipulator dengan dua lengan dan terhubung dengan sendi putar seperti pada Gambar 2 berikut. Persamaan vektor dari Gambar 1 dapat dituliskan dalam Persamaan 1.

$$\begin{aligned} X &= L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2) \\ Y &= L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2) \\ Z &= \sin \theta_3 \times r \end{aligned} \quad (1)$$

Sehingga posisi *end-effector* adalah posisi X , Y , dan Z dapat dihitung dengan mengetahui nilai $\theta_1, \theta_2, \theta_3$ serta panjang tautan L_1 dan panjang L_2 dalam Persamaan 1. Nilai r dapat diperoleh melalui Persamaan 2 berikut.

$$r = \sqrt{k_1^2 + k_2^2} \quad (2)$$

Korelasi Persamaan 1 dengan 2 dapat dikaitkan dalam bentuk Persamaan 3 berikut.

$$\begin{aligned} k_1 &= L_1 + L_2 \cos (\theta_2) \\ k_2 &= L_2 \sin (\theta_2) \end{aligned} \quad (3)$$

Selanjutnya untuk menghitung nilai kinetik mundur dapat diperoleh dari temuan nilai semua θ yang dapat dijabarkan dalam Persamaan 4.

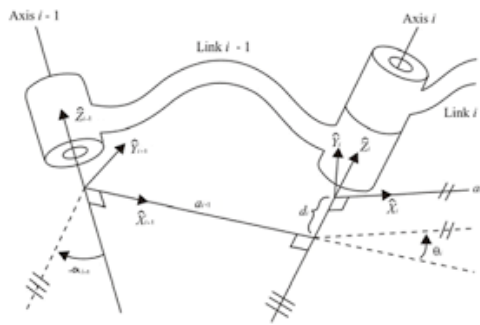
$$\begin{aligned} \theta_1 &= \arctan_2 \left(\frac{Y}{r}, \frac{X}{r} \right) - \arctan_2 (k_2, k_1) \\ \theta_2 &= \arctan_2 [\sin (\theta_2), \cos (\theta_2)] \\ \sin \theta_2 &= \sqrt{1 - \cos (\theta_2^2)} \\ \cos \theta_2 &= \frac{X^2 + Y^2 - L_1^2 - L_2^2}{2L_1L_2}, \end{aligned} \quad (4)$$

nilai X dan θ_3 diperoleh dari bentuk Persamaan 5 berikut.

$$\begin{aligned} X &= r \times \cos \theta_3 \\ \sin \theta_3 &= \frac{Z}{r} \\ \cos \theta_3 &= \frac{X}{r} \end{aligned} \quad (5)$$

B. Kinematika Maju

Terdapat beragam algoritme kinematika maju [22]. Salah satunya adalah metode dari parameter Denavit-Hartenberg [15]. Denavit-Hartenberg adalah langkah dan algoritme dalam memodelkan kinematika lengan robot dalam menentukan nilai parameter dan variabel yang ditemukan di setiap tautan. Penjelasan tentang parameter dan variabel pada lengan robot dapat dilihat pada Gambar 3 berikut. Keterangan dari Gambar



Gambar 3. Notasi parameter Denavit-Hartenberg

3 adalah sebagai berikut:

- i adalah tautan ke i
- $i - 1$ adalah tautan ke $i - 1$
- α_1 adalah sudut antara Z_1 dan Z_{i-1} pada sumbu X_i
- a_i adalah jarak antara X_{i-1} dan X_i pada sumbu Z_{i-1} yang diukur sepanjang sumbu X_i
- θ_i adalah sudut antara X_{i-1} dan X_i pada sumbu Z_i
- d_i adalah jarak antara X_{i-1} menuju X_i yang diukur sepanjang sumbu Z_i

C. Kinematika Mundur

Kinematika mundur atau terbalik memiliki fokus utama untuk mencapai posisi objek akhir yang didasarkan atas referensi tertentu [23]. Secara umum, masalah dalam kinematika terbalik dapat diselesaikan dengan beberapa teknik. Metode yang paling umum digunakan adalah matriks aljabar, pendekatan numerik, dan persamaan trigonometri [24]. Kinematika berkenaan dengan robotika erat berkaitan terhadap matriks rotasi dan matriks transformasi. Matriks rotasi adalah matriks yang memetakan vektor atau posisi pada sistem koordinat ke sistem koordinat lain dalam gerakan rotasi. Pada penelitian ini metode yang digunakan adalah pendekatan aljabar untuk solusi lengan Robot 4 DOF kinematika mundur [25]. Hal ini disebabkan oleh penurunan sifat matriks transformasi yang didasarkan atas efisien waktu dibandingkan dengan penggunaan pendekatan lainnya.

Pendekatan aljabar adalah dengan mencari solusi turunan persamaan kinematika dan matriks transformasi secara homo-

gen sehingga sudut-sudut sambungan dapat dipetakan dalam Persamaan 6 berikut.

$${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3, \quad (6)$$

dengan ${}^0T_1 = {}^0A_1$, ${}^1T_2 = {}^1A_2$, ${}^2T_3 = {}^2A_3$. Untuk menemukan besarnya persamaan sudut pada setiap sambungan, dapat dilakukan dengan memindahkan elemen dari bagian kanan ke bagian kiri dari Persamaan 6 sehingga menjadi bentuk Persamaan 7 berikut.

$$\begin{aligned} ({}^0T_1)^{-1} {}^0T_3 &= {}^1T_2 {}^2T_3 \\ ({}^0T_1)^{-1} {}^0T_3 &= {}^1T_3. \end{aligned} \quad (7)$$

Dengan menggunakan Persamaan 7, dapat dicari besarnya sambungan yang dibutuhkan untuk mencapai suatu posisi. Jika masih ada sudut sambungan yang belum diperoleh, maka gerakan elemen lagi di kanan ke kiri, sehingga menjadi hubungan Persamaan 8 berikut.

$$\begin{aligned} ({}^0T_1)^{-1} \cdot ({}^1T_2)^{-1} \cdot {}^0T_3 &= {}^2T_3 \\ ({}^0T_2)^{-1} \cdot {}^0T_3 &= {}^2T_3. \end{aligned} \quad (8)$$

Dengan bantuan Persamaan 8, solusi untuk kinematika terbalik dapat diperoleh.

D. Matriks Transformasi

Untuk menentukan posisi dan orientasi dapat menggunakan perhitungan transformasi. Matriks transformasi adalah merupakan matriks yang memetakan vektor atau posisi pada sistem koordinat tunggal ke sistem koordinat lain yang berkaitan dengan hubungan rotasi, translasi, penskalaan, dan perspektif dari titik sudut pandang tertentu. Perhitungan matriks transformasi dapat dilakukan setelah parameter dan variabel pada robot *link-arm* telah ditentukan. Sehingga dapat menentukan posisi T (*end-effector*) ke arah standar (0T_n) dengan bentuk korelasi $T = {}^1T_n = {}^2T_n = \dots = {}^{n-1}T_n$. Untuk menemukan sistem koordinat yang berdekatan ${}^{i-1}A_i$ dalam contoh sistem koordinat i dan sistem koordinat $i - 1$ adalah dalam Persamaan 9 sebagai berikut:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Hubungan antara kinetik maju dan mundur melibatkan perubahan sudut gerakan dan perubahan posisi dan ujung dari efektor.

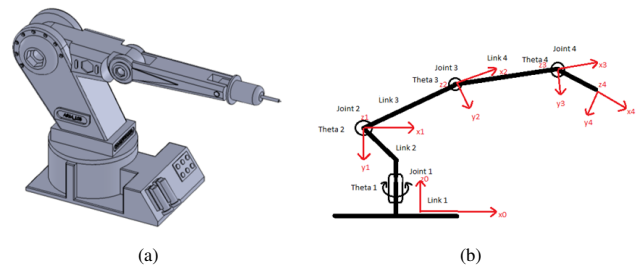
III. METODE PENELITIAN

Perangkat utama yang digunakan dalam penelitian adalah SolidWorks versi 2016 serta Matlab versi 2016. Adapun diagram sistem kerja penelitian diperlihatkan dalam Gambar 4.

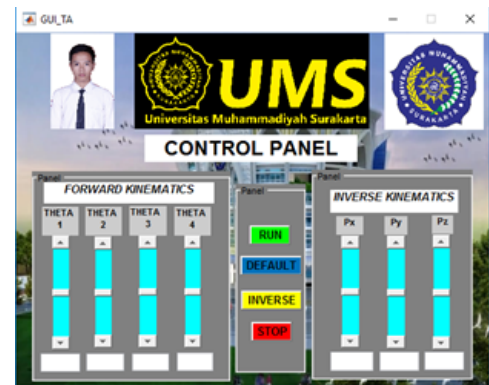
- 1) Studi literatur.
- 2) Desain model 3D dari lengan robot dengan SolidWorks.
- 3) Pembuatan program menggunakan GUI MATLAB.
- 4) Menguji simulator lengan robot.
- 5) Analisis dan persiapan laporan.

Diagram alir dari sistem simulasi robot lengan dijelaskan sebagai berikut: Langkah pertama adalah merancang model

3D robot menggunakan SolidWorks. Kemudian ekspor model ke simulink / MATLAB menggunakan tautan multibody simscape generasi pertama dalam bentuk file format XML dan STL. Di jendela perintah MATLAB jalankan file untuk mendapatkan diagram blok di simulink. Ubah diagram blok dengan menambahkan beberapa sensor, aktuator, slider gain dan konektor untuk mengendalikan lengan robot. Evaluasi robot kinematika dan verifikasi menggunakan kotak alat robotik untuk MATLAB untuk memastikan bahwa program kinematika benar. Bangun GUI (Antarmuka Pengguna Grafis) menggunakan GUIDE MATLAB Environment untuk mempermudah pemantauan dan pengendalian lengan robot.



Gambar 5. Pemodelan dan parameter (a) Robot hasil dari SolidWorks (b) parameter yang dikenakan dalam robot



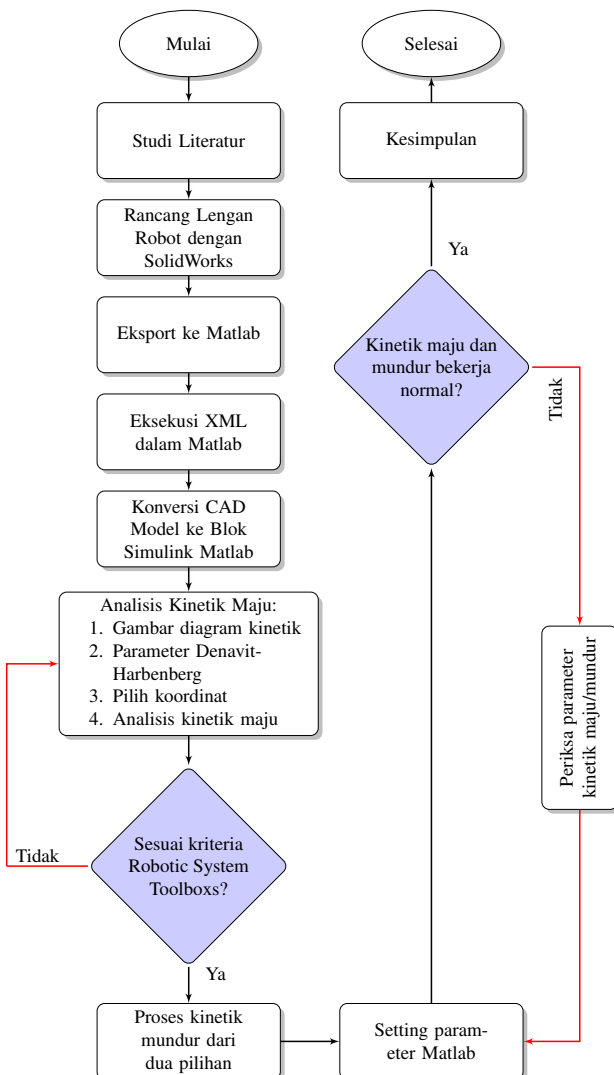
Gambar 6. GUI Matlab untuk kinetik maju dan mundur

tbody Link dalam bentuk file XML dan STL melalui Simscape Multibody Link. File XML selanjutnya diproses dalam lingkup Matlab. Model CAD dari robot selanjutnya dikonversi menjadi diagram blok di Simulink. Selanjutnya dengan menambahkan beberapa bagian seperti joint actuator dan slider maka akan dapat mengendalikan robot.

IV. HASIL PENELITIAN

Dengan mengembangkan sistem menggunakan simulasi MATLAB, kinematika maju dan kinematika terbalik lengan robot dapat dievaluasi. GUI Matlab dikembangkan dengan menggunakan Guide Environment. Antarmuka kinematika maju dan terbalik ditunjukkan pada Gambar 6.

Tabel I menunjukkan ringkasan kinematika robot. Terdapat empat kondisi posisi dan orientasi robot. Posisi default ditunjukkan pada Gambar 7 (a), di mana robot memiliki posisi awal. Kinematika maju pada Gambar 7 (b) dan kinematika terbalik pada Gambar 7 (c) dan Gambar 7 (d) dengan mengasumsikan dua nilai sebagai nilai Pick dan Place. Posisi dan orientasi (P_x, P_y, P_z) dari robot dapat di edit. Posisi bahu lengan robot adalah pada nol dan end-effector robot dalam ruang tiga dimensi di nilai (313, 0, 145) yang ditunjukkan pada Gambar 7 (a). Untuk memulai simulasi, user cukup menjalankan program GUI MATLAB. Klik tombol run pada GUI maka secara otomatis program akan berjalan dan membuka diagram blok di Simulink Matlab. Setelah kompilasi selesai, klik tombol default dan robot akan berada di posisi nol. User dapat mengubah nilai sudut ($\theta_1, \theta_2, \theta_3, \theta_4$) untuk mengevaluasi kinematika maju lengan robot dengan mengubah slider, maka posisi end-effector (P_x, P_y, P_z) akan diperoleh dan posisi robot akan berubah juga. Untuk menghitung kinematika terbalik lengan robot ($\theta_1, \theta_2, \theta_3, \theta_4$), ubah nilai (P_x, P_y, P_z) dengan



Gambar 4. Diagram proses kerja penelitian kinetik lengan robot

SolidWorks merupakan software desain pemodelan yang dinamakan 'parametrik' untuk model 3D. Parametrik berarti bahwa dimensi memiliki hubungan satu sama lain dan dapat berubah saat dalam proses desain serta secara otomatis mengubah bagian padat serta nilai korelasi yang berkaitan. Gambar 5 (a) berikut merupakan pemodelan yang dibuat dengan SolidWorks dengan nilai parameter seperti terlihat dalam Gambar 5 (b).

Model 3D robot selanjutnya diekspor dari lingkungan SolidWorks ke lingkungan Matlab ataupun semisal Simscape Mul-

Tabel I. Posisi dan orientasi robot dari perubahan kinetik (κ) maju dan mundur

Kondisi	θ_1^0	θ_2^0	θ_3^0	θ_4^0	P_x	P_y	P_z
Default	0	0	0	0	313	0	145
(κ) Maju	-72	14.4	-72	100.8	122.19	-32.06	197.66
(κ) mundur: Pick	34.01	-31.2	101.31	-70.08	200	135	145
(κ) mundur: Place	-29.05	-57.49	39.17	18.31	180	-100	308



Gambar 7. Posisi antarmuka lengan robot (a) default (b) kinetik maju (c) kinetik terbalik titik A (d) kinetik terbalik titik B

menggerakkan bilah geser lalu klik tombol kebalikan. Tombol *stop* akan menghentikan program dan menutup antarmuka GUI.

Antarmuka kinematika maju seperti yang ditunjukkan Gambar 7 (b) memiliki nilai input sudut $(-72, 14, 4, -72, 100, 8)$ derajat. Melalui kinematika maju posisi *end-effector* adalah $(-122.1962, -32.0664, 197.6632)$ derajat.

Antarmuka kinematika terbalik ditunjukkan pada Gambar 7 (c) dan 7 (d). Dengan metode kalibrasi terbalik, maka juga dapat digunakan untuk mendapatkan nilai posisi titik yaitu dari A ke B. Posisi *end-effector* pada titik A adalah $(200, 135, 145)$ dan titik B $(180, -100, 308)$. Keempat sudut bersama pada titik A adalah $(34.0193, -31.221, 101.3105, -70.0894)^0$ dan titik B $(-29.0546, 57.4943, 39.1755, 18.3188)^0$ dengan menggunakan persamaan kinematika terbalik. Kadang posisi *end-effector* tidak dapat mencapai sehingga robot tidak bergerak.

V. KESIMPULAN DAN REKOMENDASI

Mengevaluasi kinematika terbalik lebih kompleks daripada kinematika maju ketika tidak ada solusi analitis tunggal. Ada beberapa metode untuk mengevaluasi kinematika terbalik salah satunya pada penelitian ini dengan menggunakan metode pendekatan aljabar lebih efisien. Dengan memasukkan nilai input sudut $(-72, 14.4, -72, 100.8)^0$ maka didapat posisi *end-effector* kinematika maju adalah $(-122.1962, -32.0664, 197.6632)$. Dengan menggunakan persamaan kinematika terbalik maka

posisi *end-effector* pada titik A dengan nilai $(200, 135, 145)$ menuju titik B $(180, -100, 308)$ akan di dapat empat sudut bersama yaitu A $= (34.0193, -31.221, 101.3105, -70.0894)^0$ dan B $= (-29.0546, 57.4943, 39.1755, 18.3188)^0$. Robot dapat bergerak sesuai dengan nilai input yang dimasukkan x, y dan z . Dari hasil penelitian menunjukkan platform dapat melakukan gerakan omnidirectional secara penuh. Lengan robot dapat mencapai arah antara 0° hingga 360° tanpa mengubah orientasinya. Adapun rekomendasi dari penelitian ini adalah bahwa luaran dari penelitian ini dapat digunakan sebagai alat simulasi studi bidang robotika. Siswa akan lebih mudah untuk menganalisis model kinematika robot lengan dengan menggunakan simulator. Selanjutnya pengembangan dapat dilakukan dengan mengganti algoritme yang digunakan serta menambahkan gerakan maju mundur robot di samping gerakan lengan robot yang telah dibahas.

DAFTAR PUSTAKA

- [1] M. Tamre, R. Hudjakov, D. Shvarts, A. Polder, M. Hiiemaa, and M. Juurma, "Implementation of integrated wireless network and matlab system to control autonomous mobile robot," *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 1, no. 1, pp. 18–25, Oct. 2019. [Online]. Available: <http://www.ijitis.org/index.php/ijitis/article/view/10>
- [2] M. Gouasmi, M. Ouali, B. Fernini, and M. Meghatria, "Kinematic modelling and simulation of a 2-r robot using solidworks and verification by matlab/simulink," *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 245, 2012. [Online]. Available: <https://doi.org/10.5772/50203>

- [3] S. M. Mariappan and A. Veerabathiran, "Modelling and simulation of multi spindle drilling redundant scara robot using solidworks and matlab/simmechanics," *Revista Facultad De Ingenieria-universidad De Antioquia*, pp. 63–72, 2016.
- [4] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate jacobian control for robots with uncertain kinematics and dynamics," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.
- [5] A. Toquica, L. Ortiz Martinez, R. Rodriguez, A. Chavarro Chavarro, and T. Cardozo, "Kinematic modelling of a robotic arm manipulator using matlab," *Journal of Engineering and Applied Sciences*, vol. 12, 04 2017.
- [6] M. Kusban, A. Susanto, and O. Wahyunggoro, "Feature extraction for palmpoint recognition using kernel-pca with modification in gabor parameters," in *2016 1st International Conference on Biomedical Engineering (IBIOMED)*, 2016, pp. 1–6.
- [7] —, "Combination a skeleton filter and reduction dimension of kernel pca based on palmpoint recognition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, pp. 3255–3261, 12 2016.
- [8] —, "Excellent performance of palmpoint recognition by using wavelet filter," *ICIC Express Letters*, vol. 11, pp. 1315–1321, 08 2017.
- [9] M. Kusban, A. Budiman, and B. P., "An excellent system in palmpoint recognition," *IOP Conference Series: Materials Science and Engineering*, vol. 403, p. 012037, 10 2018.
- [10] M. Kusban, B. P., and A. Budiman, "Palmpoint recognition using the cosine method," *IOP Conference Series: Materials Science and Engineering*, vol. 674, p. 012041, 11 2019.
- [11] V. Fedak, F. Durovsky, and R. Uveges, "Analysis of robotic system motion in simmechanics and matlab gui environment," in *MATLAB*, K. Bennett, Ed. Rijeka: IntechOpen, 2014, ch. 20. [Online]. Available: <https://doi.org/10.5772/58371>
- [12] K. Kolesnyk, R. Panchak, V. Pylypenko, I. Abliazizov, O. Fedosyeyev, and R. Ferens, "Managing robot kinematics based on arduino controllers using a unity system," in *2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, 2017, pp. 44–46.
- [13] J. Gonçalves and P. Costa, "Differential mobile robot controller study: A low cost experiment based on a small arduino based prototype," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, 2017, pp. 945–950.
- [14] H. Zhao, Z. Lu, C. Liu, and H. Wang, "Model and simulation of the mitsubishi rv-m1 robot using matlab," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, 2016, pp. 582–585.
- [15] P. I. Corke, "A simple and systematic approach to assigning denavit hartenberg parameters," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 590–594, 2007.
- [16] X. Xiao, Y. Li, and H. Tang, "Kinematics and interactive simulation system modeling for robot manipulators," in *2013 IEEE International Conference on Information and Automation (ICIA)*, 2013, pp. 1177–1182.
- [17] M. Dahari and J. Tan, "Forward and inverse kinematics model for robotic welding process using kr-16ks kuka robot," in *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, 2011, pp. 1–6.
- [18] P.-J. Lee and W.-J. Wang, "Robotic fish kinetics design based on a fuzzy control," in *Latest Advances in Robot Kinematics*, J. Lenarcic and M. Husty, Eds. Dordrecht: Springer Netherlands, 2012, pp. 67–74.
- [19] J. M. I. Zannatha and R. C. Limon, "Forward and inverse kinematics for a small-sized humanoid robot," in *2009 International Conference on Electrical, Communications, and Computers*, 2009, pp. 111–118.
- [20] A. A. Mohammed and M. Sunar, "Kinematics modeling of a 4-dof robotic arm," in *2015 International Conference on Control, Automation and Robotics*, 2015, pp. 87–91.
- [21] E. Emmanouil, G. Wei, and J. S. Dai, "Spherical trigonometry constrained kinematics for a dexterous robotic hand with an articulated palm," *Robotica*, vol. 34, no. 12, p. 2788–2805, 2016.
- [22] P. Jamwal, S. Xie, Y. Tsoi, and K. Aw, "Forward kinematics modelling of a parallel ankle rehabilitation robot using modified fuzzy inference," *Mechanism and Machine Theory*, vol. 45, no. 11, pp. 1537 – 1554, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0094114X10001199>
- [23] M. A. Ali, H. Andy Park, and C. S. G. Lee, "Closed-form inverse kinematic joint solution for humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 704–709.
- [24] S. Kucuck and Z. Bingul, "The inverse kinematics solutions of fundamental robot manipulators with offset wrist," in *IEEE International Conference on Mechatronics, 2005. ICM '05.*, 2005, pp. 197–202.
- [25] Z. Fu, W. Yang, and Z. Yang, "Solution of Inverse Kinematics for 6R Robot Manipulators With Offset Wrist Based on Geometric Algebra," *Journal of Mechanisms and Robotics*, vol. 5, no. 3, 06 2013, 031010. [Online]. Available: <https://doi.org/10.1115/1.4024239>