

Pengembangan *User Interface* Aplikasi Simulasi *Robot Arm Manipulator* Berbasis *Augmented Reality*

Joga Dharma Setiawan^{1,*}, Yahya Fikri Dwiyanto¹, M. Munadi¹
¹ Departemen Teknik Mesin, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH., Tembalang, Semarang 50275
*E-mail: joga.setiawan@gmail.com

Abstract

Augmented reality (AR) is a technology that can make the robot manipulator simulation process more real. AR makes virtual robot objects move to carry out a real-time simulation in the real world. The use of AR in the robot manipulator simulation process is considered to be able to improve the operator's view of visualizing the robot with the surrounding environment. Therefore researchers are interested in creating and designing applications and user interfaces (UI) from AR-based arm manipulator robot simulation applications. This is based on the high rate of accidents in the work environment due to operators who cannot estimate the robot's free distance properly. Researchers developed 2 features that can function to move the joint angle robot manually in the feature labelled as "manual control" and perform trajectory planning robots in the feature labelled as "Matlab control". In the manual control feature, a joint angle control test of the robot is carried out using a slider. In the Matlab control feature, a trajectory planning simulation test based on AR was carried out. The test results on the manual control feature of each slider can function properly in moving the joint angle robot. In the Matlab control feature test results, it was found that the application succeeded in communicating so that it could carry out a simulation following the trajectory planning that had been made. Furthermore, forward and inverse kinematic analysis is performed on the trajectory.

Keywords: *augmented reality; robot manipulator; unity 3d*

Abstrak

*Augmented reality (AR) merupakan teknologi yang dapat membuat proses simulasi robot manipulator menjadi lebih nyata. AR membuat objek virtual robot bergerak menajalankan sebuah simulasi secara realtime di dunia nyata. Penggunaan AR pada proses simulasi robot manipulator dinilai dapat meningkatkan pandangan operator dalam memvisualisasikan robot dengan lingkungan disekitarnya. Oleh sebab itu peneliti tertarik untuk membuat dan merancang aplikasi serta *user interface* (UI) dari aplikasi simulasi *robot arm manipulator* berbasis AR. Hal ini didasari oleh tingginya tingkat kecelakaan dilingkungan kerja akibat operator yang tidak dapat memperkirakan jarak bebas robot dengan baik. Peneliti mengembangkan 2 fitur yang dapat berfungsi untuk menggerakkan *joint angle robot* secara manual yang fiturnya diberi label "pengendalian manual" dan melakukan *trajectory planning robot* yang fiturnya diberi label "pengendalian Matlab". Pada fitur pengendalian manual dilakukan uji kontrol *joint angle robot* menggunakan slider. Pada fitur pengendalian Matlab dilakukan uji simulasi *trajectory planning* berbasis AR. Hasil uji pada fitur pengendalian manual tiap slider dapat berfungsi dengan baik dalam menggerakkan *joint angle robot*. Pada hasil uji fitur pengendalian Matlab didapatkan bahwa aplikasi berhasil melakukan komunikasi, sehingga dapat melakukan simulasi mengikuti *trajectory planning* yang telah dibuat. Selanjutnya dilakukan analisis *forward* dan *inverse kinematic* pada *trajectory*.*

Kata kunci: *augmented reality; robot manipulator; unity 3d*

1. Pendahuluan

Teknologi robotika telah berkembang dengan pesat dan sangat berperan dalam kehidupan manusia saat ini. Teknologi robotika yang dikembangkan saat ini berfungsi untuk menggantikan atau membantu tugas manusia yang berkaitan dengan aktivitas yang membutuhkan konsentrasi serta daya tahan tinggi, terutama dalam melakukan pekerjaan fisik yang berat seperti memindahkan barang, memposisikan benda dan melakukan proses-proses lainnya. Pada dasarnya, robotika dikembangkan untuk mempermudah pekerjaan manusia. Namun pada hakikatnya robot merupakan sebuah mesin yang rumit dan kompleks yang dapat bergerak dengan kecepatan dan arah yang berbeda. Interaksi antara manusia, robot dan lingkungan dapat berkontribusi menciptakan risiko yang membahayakan manusia maupun properti yang terdapat di sekitar lingkungan robot.

Penelitian berjudul *Robot Accident Investigation* menjelaskan bahwa sejak tahun 1984 telah terjadi 43 kecelakaan terkait robot, dan 61 kematian akibat kecelakaan terkait robot [1]. Sebagian besar kecelakaan di lingkungan kerja disebabkan oleh kelalaian operator, salah satu faktor penyebab yaitu operator yang tidak dapat memperkirakan jarak bebas

dari robot dengan baik, hal tersebut mengakibatkan terjadinya tabrakan antara robot dengan manusia ataupun properti di sekitar lingkungan robot.

Sebuah penelitian berjudul *Augmented Reality for Robotics* dijelaskan bahwa penggunaan teknologi AR dapat menggabungkan objek virtual ke dalam dunia nyata secara *real-time* dinilai dapat meningkatkan persepsi dari operator untuk memvisualisasikan robot dengan lingkungan disekitarnya [2]. Penggunaan AR pada bidang robotika khususnya dalam proses simulasi robot dinilai dapat meningkatkan kepercayaan diri operator, dan meningkatkan keamanan serta keselamatan.

Oleh karena itu, pada penelitian ini, peneliti melakukan sebuah perancangan aplikasi simulasi *robot arm manipulator* berbasis AR. Perancangan aplikasi ini meliputi pengembangan *user interface* (UI), dan pembuatan fitur yang dapat menjalankan sebuah simulasi *robot arm manipulator* berbasis AR yang terintegrasi dengan *software* MATLAB. Sehingga aplikasi dapat menjalankan simulasi berdasarkan lintasan yang telah dibuat pada *software* MATLAB. Selanjutnya, dilakukan analisis pada lintasan *robot arm manipulator* berdasarkan persamaan kinematik.

2. Material dan Metode Penelitian

2.1 Augmented reality (AR)

AR merupakan sebuah teknologi yang dapat menggabungkan objek *virtual* dua atau tiga dimensi ke dalam dunia nyata, kemudian memproyeksikan objek *virtual* tersebut kedalam dunia nyata secara *real-time* [3]. Teknologi AR pertama kali ditemukan sekitar tahun 1957 sampai 1962 oleh Morton Heilig [4]. Dalam menjalankan sistem AR diperlukan peralatan seperti perangkat *device*, kamera, dan dalam suatu kasus tertentu memerlukan perangkat khusus [5]. Hingga saat ini AR sudah banyak dimanfaatkan dalam berbagai kepentingan, salah satu pemanfaatan AR yaitu pada bidang robotika. Pada bidang robotika penggunaan AR dapat diaplikasikan pada hal seperti pembelajaran dan pengenalan fitur pada robot, pelatihan pengoperasian robot, serta simulasi. Dunia simulasi dan *virtual* ini dapat digunakan untuk memperoleh pandangan perspektif yang berbeda. Secara umum teknologi AR dapat memberikan visualisasi yang lebih baik saat melakukan simulasi dibandingkan dengan metode simulasi biasa.

2.2 TCP/IP

TCP/IP atau *Transmission Control Protocol/Internet Protocol* merupakan protokol untuk menghubungkan komunikasi suatu komputer dengan komputer lainnya melalui internet. Protokol TCP/IP merupakan sekumpulan protokol komunikasi yang saat ini digunakan secara luas dalam komunikasi global jaringan *computer* [6]. Dalam penelitian ini TCP/IP digunakan sebagai protokol komunikasi. Melalui lintasan jalur robot yang dibuat, komunikasi dihasilkan melalui protokol TCP/IP yang memungkinkan untuk mengirim atau menerima data secara *realtime*. Dalam prosesnya, variabel tiap sendi robot dikirim dari pemodelan simulasi menuju aplikasi secara *realtime*. Sehingga karakter robot pada aplikasi dapat bergerak mengikuti lintasan yang telah ditentukan.

2.3 Robot arm manipulator

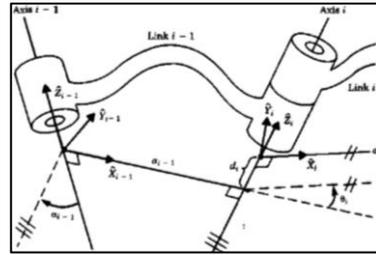
Robot arm manipulator merupakan alat yang sangat membantu proses bisnis sehingga banyak digunakan di oleh Industri. Sebuah manipulator terdiri dari satu atau lebih *joints* dan satu atau lebih *links* [7]. Berdasarkan sebuah buku berjudul *Introduction to Robotics Mechanics and Control* dijelaskan bahwa terdapat beberapa jenis konfigurasi mekanik yang digunakan pada manipulator yaitu *cartesian*, *articulated*, SCARA, silinder, dan polar [8]. Dalam penelitian ini manipulator digunakan sebagai objek dari aplikasi, manipulator akan melakukan sebuah simulasi mengikuti *trajectory* yang telah dibuat. *Robot arm manipulator* yang digunakan dalam penelitian ini adalah Kuka LBR iiwa 7.

2.4 Pemodelan kinematika

Kinematika adalah cabang dari mekanika yang membahas gerakan benda tanpa memperhitungkan penyebab gerakan dari gaya tersebut [9]. Kinematika memiliki dua topik mendasar. Pertama *forward kinematics* yaitu suatu fungsi untuk menghitung orientasi serta posisi dari *end-effector* berdasarkan parameter *joint angle* robot [10]. Selanjutnya *inverse kinematics* yaitu suatu fungsi untuk menghitung besaran sudut *joint* berdasarkan posisi *end-effector*. *Inverse kinematic* sering digunakan dalam melakukan perencanaan lintasan *robot arm manipulator* [11]. Dalam penelitian ini pemodelan kinematik digunakan untuk analisis *kinematic* pada *trajectory* simulasi.

2.4.1 Forward kinematics

Analisis *forward kinematics* menggunakan metode Denavit-Hartenberg (DH) yang menyediakan pendekatan secara standar. Denavit-Hartenberg merupakan salah satu langkah dalam memodelkan manipulator secara kinematika yaitu penentuan nilai parameter serta variabel yang terdapat pada setiap *link* [12]. Penjelasan lebih lanjut mengenai parameter dan variabel pada *arm* robot dapat dilihat pada Gambar 1.



Gambar 1. Notasi Denavit-Hartenberg [13].

Setiap notasi memiliki ketetapan diantaranya yaitu i adalah link ke i , $i - 1$ adalah link ke $i - 1$, α_i adalah sudut putaran antara sumbu z_i dan z_{i+1} terhadap sumbu x_i , a_i adalah jarak dari sumbu z_i dan z_{i+1} terhadap sumbu x_i , θ_i adalah sudut putaran antara sumbu x_{i+1} dan x_i terhadap sumbu z_i , d_i adalah jarak dari sumbu x_{i-1} ke x_i terhadap sumbu z_i . Tabel 1 menunjukkan DH-Parameter dari robot arm manipulator Kuka LBR Iiwa 7 dengan tambahan end-effector berupa gripper.

Tabel 1. DH-Parameter Kuka LBR iwa 7 [14]

Link	DH Parameters			
	θ_i (°)	α_i (°)	a_i (mm)	d_i (mm)
1	θ_1	-90°	0	340
2	θ_2	90°	0	0
3	θ_3	90°	0	400
4	θ_4	-90°	0	0
5	θ_5	-90°	0	400
6	θ_6	90°	0	0
7	θ_7	0°	0	320

Setelah parameter D-H didapatkan, selanjutnya yaitu melakukan transformasi matrik. Dengan transformasi matriks antara link, maka akan diperoleh hasil akhir berupa posisi i -link bergerak relatif terhadap link yang direferensikan. Dalam perhitungan matrik transformasi, parameter yang digunakan masih berupa simbol. Hal tersebut bertujuan agar memudahkan dalam memasukkan hasil perhitungan ke dalam program lintasan robot.

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \cdot \sin\theta_i & \sin\alpha_i \cdot \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cdot \cos\theta_i & -\sin\alpha_i \cdot \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Untuk mendapatkan transformasi matrik link 7 terhadap link 0 yang akan menentukan posisi serta orientasi end-effector terhadap base, maka dapat mengalikan matrik transformasi antar link. Hasil dari transformasi link 7 terhadap link 0 ditunjukkan pada persamaan 2, dimana (p_x, p_y, p_z) menunjukkan posisi dan $(\{n_x, n_y, n_z\}, \{o_x, o_y, o_z\}, \{a_x, a_y, a_z\})$ menunjukkan orientasi pada end-effector.

$${}^0T_7 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

2.4.2 Inverse kinematics

Inverse kinematics yang dilakukan pada penelitian kali ini bertujuan untuk mendapatkan nilai sudut joint yang diperlukan dalam mencapai posisi yang diinginkan. Dalam menentukan sudut joint yang diperlukan (inverse kinematics) harus menyelesaikan persamaan nonlinier. Untuk menghitung sudut θ_1 dan θ_2 diperlukan posisi siku (X,Y,Z) yang berkaitan dengan base seperti yang ditunjukkan pada persamaan 3.

$$P_{elbow}^B = A_3^B P_{elbow}^3 = A_1^B A_2^1 A_3^2 P_{elbow}^3 \tag{3}$$

Dengan $P_{elbow}^3 = [0, 0, 0, 1]^T$ (asal dari sistem koordinat 3). Ini menghasilkan sistem persamaan nonlinier yang ditunjukkan pada persamaan 4.

$$\begin{bmatrix} P_{elbow}^B \cdot x \\ P_{elbow}^B \cdot y \\ P_{elbow}^B \cdot z \\ 1 \end{bmatrix} = \begin{bmatrix} d_3 \cos \theta_1 \cdot \sin \theta_2 \\ d_3 \sin \theta_1 \cdot \sin \theta_2 \\ d_1 + d_3 \cdot \cos \theta_2 \\ 1 \end{bmatrix} \quad (4)$$

Maka dapat dihasilkan θ_1 yang ditunjukkan pada persamaan 5.

$$\theta_1 = \text{atan2}(P_{elbow}^B \cdot y, P_{elbow}^B \cdot x) \quad (5)$$

Dan θ_2 dapat dihasilkan dengan persamaan pada persamaan 6.

$$\theta_2 = \text{acos}\left(\frac{P_{elbow}^B \cdot z - d_1}{d_3}\right) \quad (6)$$

Selanjutnya θ_3 dan θ_4 dapat dihitung dengan persamaan 7.

$$P_{wrist}^B = A_5^B P_{wrist}^5 = A_1^B A_2^1 A_3^2 A_4^3 A_5^4 P_{wrist}^5 \quad (7)$$

Dengan $P_{wrist}^5 = [0,0,0,1]^T$ (asal dari sistem koordinat 5).

$$(A_2^1)^{-1} (A_1^B)^{-1} P_{wrist}^B = A_3^2 A_4^3 A_5^4 P_{wrist}^5 \quad (8)$$

Dari persamaan 8, maka dapat dilanjutkan dengan persamaan 9.

$$\begin{bmatrix} m \\ n \\ p \\ 1 \end{bmatrix} = \begin{bmatrix} d_5 \cos \theta_3 \cdot \sin \theta_4 \\ d_5 \sin \theta_3 \cdot \sin \theta_4 \\ d_3 + d_5 \cdot \cos \theta_4 \\ 1 \end{bmatrix} \quad (9)$$

Maka demikian didapatkan θ_3 dan θ_4 yang ditunjukkan pada persamaan 10 dan 11.

$$\theta_3 = \text{atan2}(n, m) \quad (10)$$

$$\theta_4 = \text{acos}\left(\frac{p - d_3}{d_5}\right) \quad (11)$$

Selanjutnya sudut θ_5 dan θ_6 dapat dihitung dengan persamaan 12.

$$P_{target}^B = A_7^B P_{target}^7 = A_1^B A_2^1 A_3^2 A_4^3 A_5^4 A_6^5 A_7^6 P_{target}^7 \quad (12)$$

Dengan $P_{target}^7 = [0,0,0,1]^T$ (asal dari sistem koordinat 7).

$$(A_4^3)^{-1} (A_3^2)^{-1} (A_2^1)^{-1} (A_1^B)^{-1} P_{target}^B = A_5^4 A_6^5 A_7^6 P_{target}^7 \quad (13)$$

$$\begin{bmatrix} m \\ n \\ p \\ 1 \end{bmatrix} = \begin{bmatrix} d_7 \cos \theta_5 \cdot \sin \theta_6 \\ d_7 \sin \theta_5 \cdot \sin \theta_6 \\ d_5 + d_7 \cdot \cos \theta_6 \\ 1 \end{bmatrix} \quad (14)$$

Maka demikian didapatkan θ_5 dan θ_6 yang ditunjukkan pada persamaan 15 dan 16.

$$\theta_5 = \text{atan2}(n, m) \quad (15)$$

$$\theta_6 = \text{acos}\left(\frac{p - d_5}{d_7}\right) \quad (16)$$

Tabel 2. Koordinat tiap pos pada trajectory planning robot.

Pos	Posisi (cm)	
1	X	-54.9
	Y	0
	Z	30.35
2	X	-55
	Y	0
	Z	5
3	X	45
	Y	-40
	Z	20
4	X	0
	Y	0
	Z	146

3.2 Hasil akhir Aplikasi

Hasil dari penelitian ini berupa aplikasi simulasi *robot arm manipulator* yang berbasis AR. Terdapat dua fitur pada aplikasi ini yaitu fitur pengendalian manual yang berfungsi untuk melakukan kontrol *joint angle* secara manual menggunakan *slider*, kemudian fitur pengendalian Matlab yang berfungsi untuk melakukan kontrol *trajectory planning* menggunakan *inverse kinematics* yang terintegrasi dengan MATLAB. Gambar 3 menunjukkan setiap menu yang terdapat pada aplikasi simulasi *robot arm manipulator* berbasis AR.



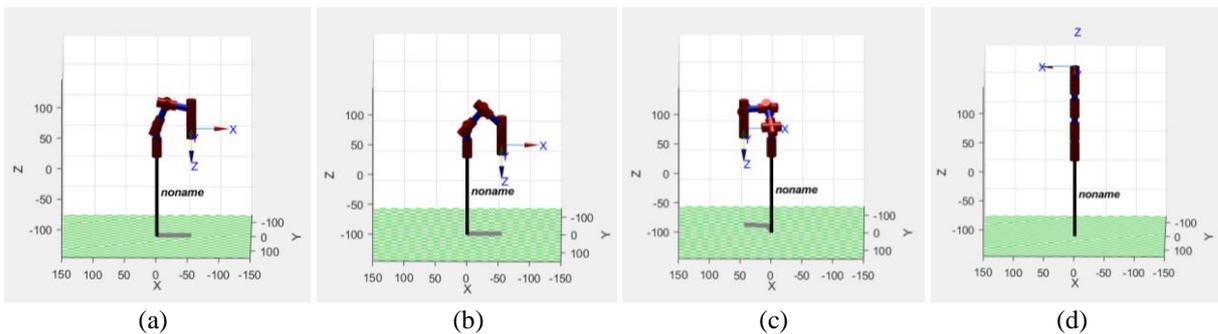
Gambar 3. Aplikasi AR (a) Menu awal, (b) Fitur pengendalian manual, (c) Fitur Pengendalian MATLAB.

3.3 Hasil Analisis Trajectory

Trajectory planning yang dibuat berdasarkan *inverse kinematics* disimulasikan serta dihitung menggunakan *Robotics Toolbox* Matlab. Tabel 3 menunjukkan hasil dari perhitungan *inverse kinematics* pada tiap pos lintasan. Pada pos 1 robot bergerak ke posisi yang telah ditentukan, kemudian pada pos 2 *end-effector* robot menuju posisi bola untuk mengambil bola. Pada pos 3 *end-effector* robot menuju posisi keranjang untuk menempatkan bola ke keranjang, lalu pada pos 4 robot kembali ke posisi semula. Plot posisi dari *trajectory planning* yang telah dibuat dapat dilihat pada gambar 4.

Tabel 3. Hasil perhitungan *inverse kinematics* pada tiap pos.

Pos	Posisi (cm)		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
1	X	-54.9	0°	-23°	0°	79°	0°	-77.5°	0°
	Y	0							
	Z	30.35							
2	X	-55	0°	-40.4°	0°	93°	0°	-46.6°	0°
	Y	0							
	Z	5							
3	X	45	-79.6°	55.8°	72.4°	-76.5°	-52.5°	83.6°	-5.4°
	Y	-40							
	Z	20							
4	X	0	0°	0°	0°	0°	0°	0°	0°
	Y	0							
	Z	146							

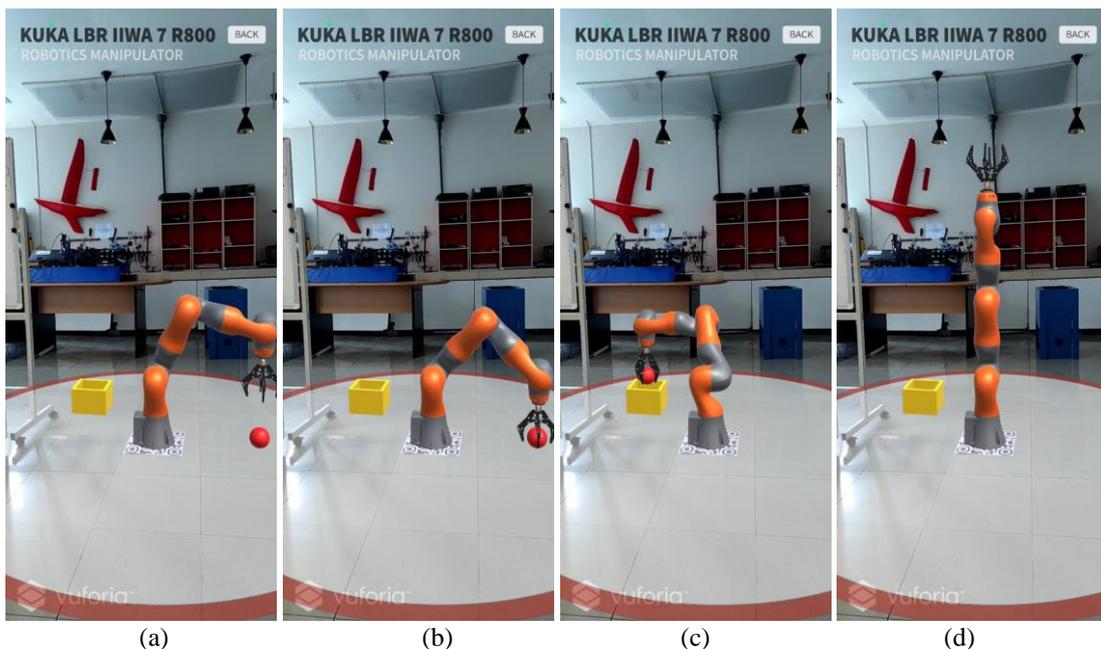


Gambar 4. Plot posisi robot menggunakan Robotics Toolbox (a) Pos 1, (b) Pos 2, (c) Pos 3, (d) Pos 4.

3.4 Hasil Pengujian Aplikasi

3.4.1 Pengujian Fitur Pengendalian Matlab

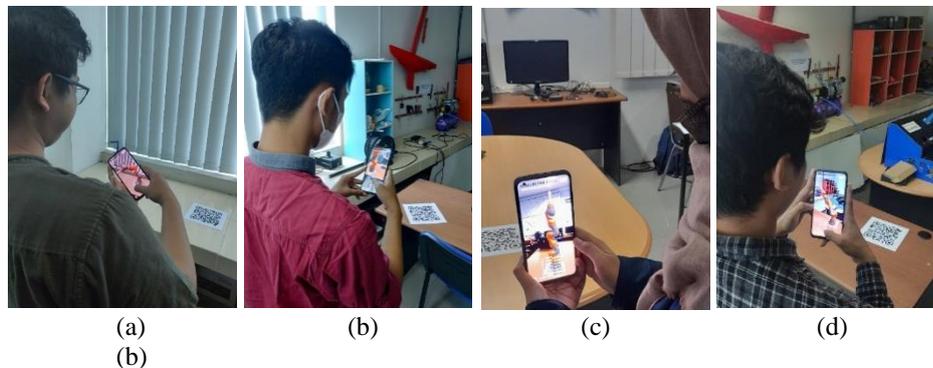
Pengujian simulasi *trajectory planning* berbasis AR pada fitur pengendalian Matlab dapat dijalankan ketika merunning *trajectory planning* pada MATLAB. Dapat dilihat pada gambar 5 bahwa aplikasi dapat menjalankan simulasi dengan baik. Hasil pengujian menunjukkan bahwa robot berhasil menjalankan simulasi sesuai dengan hasil perhitungan *inverse kinematics* yang telah dilakukan. Hal tersebut menunjukkan bahwa fitur pengendalian Matlab dapat berkomunikasi dengan baik dengan Matlab menggunakan protokol TCP/IP.



Gambar 5. Pengujian simulasi *trajectory planning* pada fitur pengendalian Matlab (a) Pos 1, (b) Pos 2, (c) Pos 3, (d) Pos 4.

3.4.2 Pengujian Fitur Pengendalian Manual

Pegujian untuk fitur pengendalian manual telah dilakukan oleh empat responden mahasiswa Universitas Dipoengoro. Terdapat beberapa parameter yang akan dinilai dalam pengujian ini, diantaranya yaitu apakah fitur dapat memunculkan visual robot virtual dengan baik. Kemudian apakah setiap slider dapat menggerakkan masing-masing *joint* robot sesuai dengan perintah slider. Dari hasil pengujian yang dilakukan oleh empat responden ini seperti ditunjukkan dalam gambar 6, dapat disimpulkan bahwa fitur pengendalian *joint angle* robot secara manual dapat berfungsi dengan baik serta dapat digunakan dengan mudah.



Gambar 6. Pengujian pengendalian *joint angle* robot menggunakan *slider* pada fitur pengendalian manual
(a) Responden 1, (b) Responden 2, (c) Responden 3, (d) Responden 4.

4. Kesimpulan

Pada penelitian ini telah dibuat aplikasi simulasi *robot arm manipulator* berbasis AR yang memiliki 2 fitur yaitu fitur pengendalian manual yang berfungsi untuk melakukan kontrol *joint angle* secara manual menggunakan *slider*. Selanjutnya fitur pengendalian Matlab yang berfungsi untuk melakukan kontrol *trajectory planning* menggunakan *inverse kinematic* yang terintegrasi dengan Matlab. Kemudian telah dibuat *trajectory planning* menggunakan Robotics Toolbox Matlab yang dikembangkan oleh Peter Corke berdasarkan perhitungan *inverse kinematics*. Pengujian fitur aplikasi telah dilakukan meliputi pengujian fitur pengendalian Matlab yaitu pengujian simulasi *trajectory planning* berbasis AR, dan pengujian fitur pengendalian manual yaitu pengujian kontrol *joint angle* robot menggunakan *slider*. Pada pengujian fitur pengendalian Matlab didapatkan bahwa aplikasi dapat berkomunikasi dengan *software* Matlab menggunakan protokol TCP/IP, dan melakukan simulasi mengikuti *trajectory planning* yang telah dibuat. Pada pengujian fitur pengendalian manual didapatkan bahwa slider dapat menggerakkan *joint angle* robot dengan baik.

Daftar Pustaka

- [1] Winfield, A. F. T., Winkle, K., Webb, H., Lyngs, U., & Jirotko, M. (2021). Robot Accident Investigation: a case study in Responsible Robotics. *Software Engineering for Robotics*, August 2021.
- [2] Makhataeva, Z., & Varol, H. A. (2020). Augmented reality for robotics: A review. *Robotics*, 9(2).
- [3] de Souza Cardoso, L. F., Mariano, F. C. M. Q., & Zorzal, E. R. (2020). A survey of industrial augmented reality. *Computers and Industrial Engineering*, 139(October 2019).
- [4] Setiawan, A. (2020). Aplikasi Media Promosi Penjualan Rumah Berbasis Augmented Reality. *INTERNAL (Information System Journal)*, 3(1), 1–14.
- [5] Borman, R. I. (2017). Implementasi Augmented Reality pada Aplikasi Android Pegenalan Gedung Pemerintahan Kota Bandar Lampung. *Jurnal Teknoinfo*, 11(1), 1.
- [6] Prasetya, F. X. H., & Triantoni, D. (2020). *Komunikasi Suara Menggunakan Protocol TCP/IP*. 1–6.
- [7] Ekrem, Ö., & Aksoy, B. (2023). Trajectory planning for a 6-axis robotic arm with particle swarm optimization algorithm. *Engineering Applications of Artificial Intelligence*, 122, 106099.
- [8] Craigh, J. (2005). Book Review: Introduction to Robotics: Mechanics and Control. *The International Journal of Electrical Engineering & Education*, 41(4), 388–388.
- [9] Syam, R. (2015). *Kinematika dan Dinamika Robot Lengan*. 76.
- [10] Gonçalves, F., Ribeiro, T., Ribeiro, A. F., Lopes, G., & Flores, P. (2022). A recursive algorithm for the forward kinematic analysis of robotic systems using euler angles. *Robotics*, 11(1), 15.
- [11] Zhao, G., Jiang, D., Liu, X., Tong, X., Sun, Y., Tao, B., ... & Fang, Z. (2022). A tandem robotic arm inverse kinematic solution based on an improved particle swarm algorithm. *Frontiers in Bioengineering and Biotechnology*, 10, 832829.
- [12] Zhong, F., Liu, G., Lu, Z., Han, Y., Liu, F., & Ye, T. (2023). Inverse Kinematics Analysis of Humanoid Robot Arm by Fusing Denavit–Hartenberg and Screw Theory to Imitate Human Motion With Kinect. *IEEE Access*.
- [13] Siciliano, B., & Khatib, O. (2018). *Springer Handbook of Robotics-Springer*.
- [14] Corke, P. (1996). A Robotics Toolbox. *Robotics & Automation Magazine, IEEE*, Vol.3, No.1, March, pp.24–32.