

PERANCANGAN MANIPULATOR ROBOT BEDAH 6-DOF DENGAN REMOTE CENTER OF MOTION MEKANIK BERBASIS INVERSE KINEMATICS

Hanafi Fadzillah^{1*)}, Iwan Setiawan², Sumardi³

¹²³Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro, Semarang, Indonesia

*) Email: Hanafi.fadzil00@gmail.com

Abstrak

Penelitian mengenai robot bedah di dalam dunia medis menjadi salah satu penelitian yang cukup populer karena manfaatnya dalam MIS (*Minimal Invasive Surgery*). Pembedahan ini memungkinkan pasien untuk mengalami sayatan kecil dalam pembedahan sehingga pasien tidak mengalami pendarahan berlebih dan cepat sembuh dari operasi. Oleh karena itu, pada penelitian ini dibahas mengenai perancangan manipulator robot 6 DOF (*Degree of Freedom*) dengan RCM (*Remote Center of Motion*) Mekanik. RCM mekanik memungkinkan rod end effector robot bergerak dengan mempertahankan satu titik statis yang di dalam dunia medis disebut trocar point. Dengan adanya RCM ini, beban sistem kendali robot dapat dikurangi dan titik trocar yang didapatkan tetap. *Differential inverse kinematics* yang memanfaatkan Matriks Jacobian digunakan sebagai sistem kendali robot. Karena metode differential inverse kinematics merupakan metode yang iteratif, algoritma *trajectory planning* dirancang untuk menyatu dengan metode *inverse kinematic*. Posisi *end effector* yang didapatkan melalui metode kendali ini memiliki *error* maksimum rata-rata sebesar 36 mm dengan hasil *trajectory* yang halus menyerupai respon sistem orde 1.

Kata Kunci: *Differential Inverse Kinematics, Remote Center of Motion, Minimal Invasive Surgery, Trajectory Planning*

Abstract

Research around surgical robots in the medical world is popular because of their use in MIS (Minimal Invasive Surgery). This surgery allows the patient to undergo a small incision during surgery so that they do not experience excessive bleeding and recover quickly from hospitalization. Therefore, this research discusses the design of a 6-DOF (Degree of Freedom) robot manipulator with a mechanical RCM (Remote Center of Motion). Mechanical RCM allows the robot's rod end effector to move by maintaining a static point, which in the medical world is called the trocar point. With this RCM, the load on the robot control system can be reduced, and the trocar point obtained is fixed. Differential inverse kinematics that utilizes the Jacobian Matrix is used as a robot control system. Because the differential inverse kinematics nature is an iterative method, the trajectory planning algorithm is designed to integrate with the inverse kinematics method. The end effector position obtained through this control method has an average maximum error of 36 mm, with smooth trajectory results resembling a first-order system response.

Keywords: *Differential Inverse Kinematics, Remote Center of Motion, Minimal Invasive Surgery, Trajectory Planning*

1. Pendahuluan

Perkembangan teknologi robotika telah berkembang pesat sampai ke dunia medis, tepatnya dalam hal operasi dan pembedahan pada manusia. Robot bedah sendiri merupakan robot yang membantu meningkatkan karakteristik indra pembedah dan memungkinkan untuk menjalankan pembedahan yang lebih halus serta prosedur MIS (*minimal invasive surgery*) [1]. Berdasarkan harapan dan peluang tersebut, beberapa perusahaan mulai mengkomersialkan robot untuk pembedahan, salah satu perusahaan tersebut yang terkenal adalah Intuitive Surgical Inc. Kemampuan dan hasil yang ditawarkan dari robot bedah inipun bukanlah omong kosong belaka, hal ini

dibuktikan dengan sudah adanya robot bedah (*surgical robot*) yang sudah beroperasi di rumah sakit - rumah sakit di negara maju seperti Amerika Serikat, Jepang, dan Singapura. Di Jepang teknologi robot bedah ini mulai dikembangkan untuk merealisasikan penggunaan sebenarnya dari telesurgery atau operasi jarak jauh antara rumah sakit utama dan rumah sakit yang lain [2].

Dibalik kelebihan-kelebihan tersebut, penggunaan robot bedah masih memiliki beberapa kelemahan yang harus dihadapi, yakni biaya akuisisi, operasi dan perawatan robot itu sendiri yang masih terlalu mahal dari sudut pandang negara berkembang seperti Indonesia dengan harga berkisar 3568 dollar [3].

Dari permasalahan tersebut munculah penelitian – penelitian baru mengenai perkembangan teknologi robot

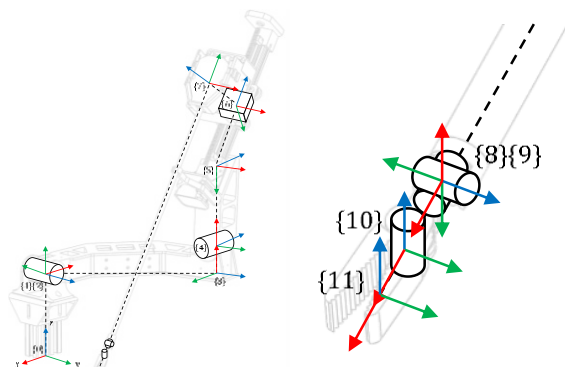
bedah medis, seperti robot bedah Raven II yang dikembangkan oleh University of California. Robot ini memiliki konfigurasi 7 DOF (6 DOF + *grasp*) dan bekerja dengan memanfaatkan sistem *teleoperation*. Sistem *teleoperation* terdiri dari dua komponen utama, yaitu komponen yang bertindak sebagai *master device* yang dikendalikan oleh operator, dan komponen yang bertindak sebagai *slave device* yang melakukan aksi sesuai perintah yang diberikan oleh *master device* [4]. Pada *slave device* robot Raven II sendiri memiliki *constraint* mekanik yang memungkinkan robot untuk menggerakkan *end effector* dengan *remote center of motion* [5]. *Remote center of motion* ini memungkinkan *end effector* untuk masuk ke dalam tubuh pasien melalui satu titik yang di sebut *trocar point*. *Trocar point* inilah yang menjadi kunci utama dalam pelaksanaan prosedur MIS. MIS adalah teknik pembedahaan berdasarkan akses ke rongga dalam melalui beberapa sayatan kecil di tubuh pasien atau yang disebut *trocar point*. Adanya batasan ini menyebabkan pergerakan *end effector* terbatas menjadi 4 *Degree of Freedom* (DOF) [6].

Berdasarkan uraian tersebut, makalah ini membahas mengenai perancangan manipulator robot bedah 6-DOF dengan *remote center of motion* mekanik berbasis *inverse kinematics* untuk menguji pergerakan trajectory beserta tingkat presisi dan akurasi posisi *end effector*.

2. Metode

2.1. Perancangan Perangkat Keras

Manipulator yang didesain pada makalah ini menyerupai desain robot lengan Raven II, dimana robot ini memiliki 6 DOF (derajat kebebasan) dengan satu *grasper* (pencengkram) seperti pada Gambar 1 [4]. Desainnya yang unik ini dimaksudkan untuk mendapatkan pergerakan pada *rod end effector* dengan *remote center of motion*. *Remote center of motion* ini memungkinkan *rod* pada *end effector* bergerak dengan tetap mempertahankan satu titik yang disebut *trocar point*. *Modified DH parameter* robot dapat dilihat pada Tabel 1.

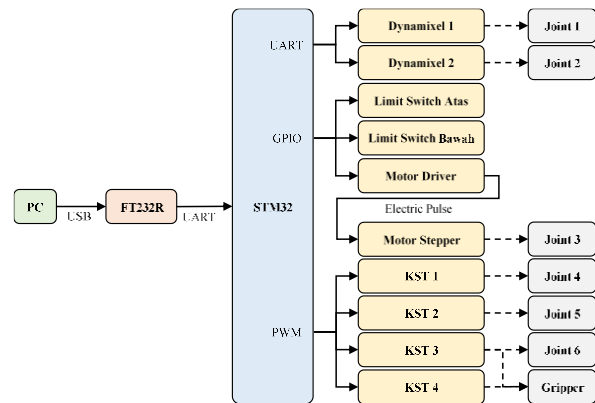


Gambar 1. Diagram kinematik manipulator

Tabel 1. Modified DH parameter manipulator

j	α_{j-1} (°)	a_{j-1} (mm)	d_j (mm)	θ_j (°)	Type	Min	Max
1	0	0	148	180	Fixed	-	-
2	90	0	0	0	Revolute	0°	90°
3	0	240,21	179,33	90	Fixed	-	-
4	105	38,5	0	0	Revolute	-55°	55°
5	0	150,32	0	90	Fixed	-	-
6	52	0	90	0	Prismatic	39,37	188,37
7	90	0	125	0	Fixed	-	-
8	90	0	473,5	90	Revolute	-45°	225°
9	90	0	0	90	Revolute	0°	180°
10	90	9,5	0	0	Revolute	-90°	90°
11	0	7,5	0	0	Fixed	-	-

Berdasarkan Tabel 1 dapat dilihat bahwa robot didesain dengan 11 *joint* dimana 5 *joint* bertipe *fixed*, 5 *joint* bertipe *revolute* dan 1 *joint* bertipe *prismatic*. *Joint* 2, 4 dan 6 berfungsi untuk mengatur posisi *end effector* sedangkan *joint* 8, 9, 10 berfungsi untuk mengatur orientasi *end effector*. Konfigurasi link ini memungkinkan pergerakan robot mengikuti *constraint remote center of motion*. Gambar 2 menunjukkan konfigurasi system elektrik manipulator



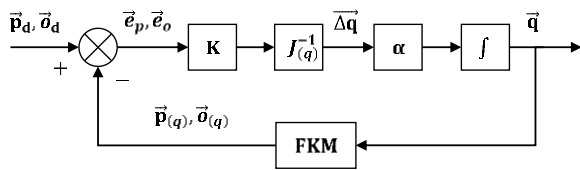
Gambar 2. Konfigurasi sistem elektrik manipulator

Sistem elektrik manipulator berpusat pada PC dengan program bahasa python dalam mengkomputasi nilai *joint* robot seperti yang ditunjukkan oleh Gambar 2. Nilai *joint* yang didapatkan ditransmisikan menuju STM32 melalui FT232R untuk mengaktuiasi aktuator. Selain itu, terdapat pula *Limit Switch* untuk menjaga pergerakan *joint* 3 agar tidak melebihi nilai yang seharusnya.

2.2. Perancangan Metode Kontrol

Metode *inverse kinematics* yang digunakan pada makalah ini adalah metode *differential kinematics*, dimana metode ini akan memberikan suatu Persamaan yang menghubungkan antara kecepatan dari *joint* (derivatif dari nilai *joint*) dan kecepatan linear dan sudut dari *end effector* (derivatif dari *pose end effector*) [7]. Pemetaan ini nantinya akan direpresentasikan dengan matriks jacobian [8].

Gambar 3 menunjukkan diagram blok metode differential inverse kinematics

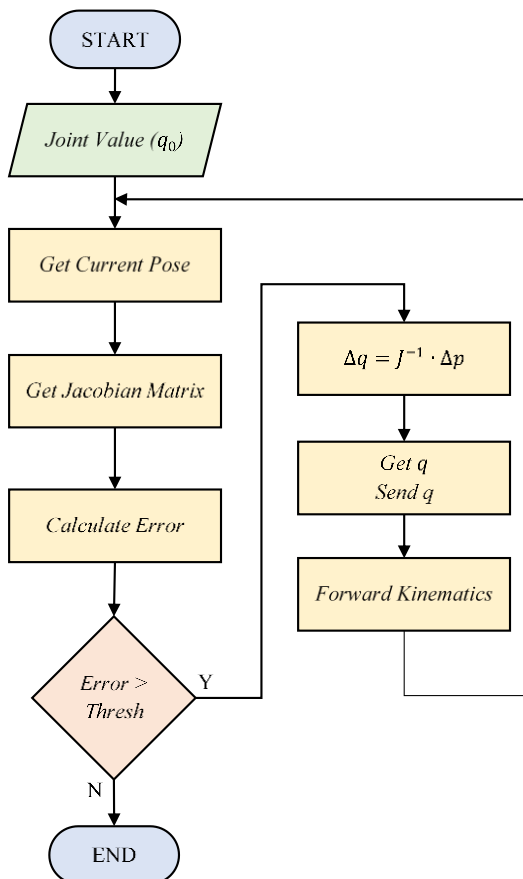


Gambar 3. Diagram blok metode differential inverse kinematics

Secara mudahnya, metode differential inverse kinematic dapat dijelaskan melalui diagram blok pada Gambar 3 berikut. Selain itu, Metode ini juga sekaligus menjadi algoritma trajectory planning dari pergerakan end effector robot. Karena sifat dari metode ini adalah metode yang iteratif maka sistem akan merubah nilai joint sedikit demi sedikit yang bergantung dari error dan parameter update rate sistem hingga pose akhir yang diharapkan tercapai.

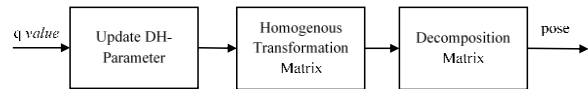
2.3. Perancangan Perangkat Lunak

Pembangunan program inverse kinematics berdasarkan dari diagram blok pada Gambar 3 sedangkan Gambar 4 merupakan diagram alir untuk perancangan perangkat lunak.



Gambar 4. Diagram alir metode differential inverse kinematics

Metode differential inverse kinematics merupakan sistem iteratif sehingga metode ini akan terus berulang hingga pose end effector mendekati set point yang diharapkan dengan error dibawah ambang batas yang ditentukan. Masukan Joint Value hanya dilakukan sekali yaitu, disaat fase inialisasi pada robot sehingga pada iterasi-iterasi berikutnya, proses Get Current Pose mendapatkan masukan joint baru dari proses forward kinematics. Gambar 5. Diagram blok forward kinematics



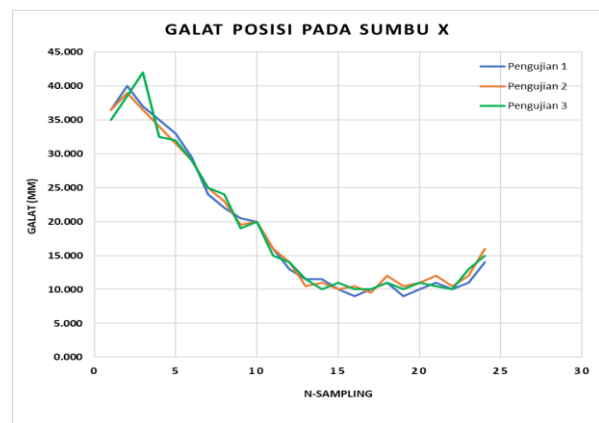
Gambar 5. Diagram blok forward kinematics

Forward Kinematics dari Manipulator dapat dijelaskan melalui Diagram Blok pada Gambar 5 diatas. Masukan nilai joint (q) digunakan untuk memperbarui parameter DH sehingga didapatkan postur robot baru. Parameter DH yang telah diperbarui sebelumnya digunakan untuk memperbarui matriks transformasi homogen. Dengan mendekomposisi Matrik transformasi inilah dapat diketahui pose robot di setiap frame-nya, termasuk juga pose pada frame terakhir yakni pose end effector.

3. Hasil dan Pembahasan

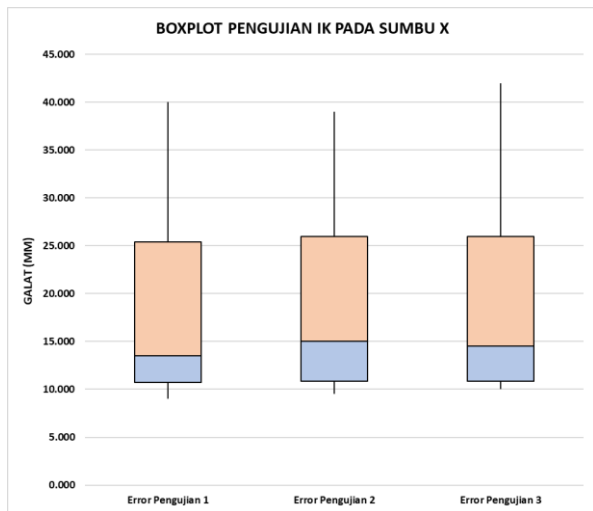
3.1. Pengujian Kendali Posisi dengan Inverse Kinematic

Hasil pengujian kendali posisi dengan inverse kinematics pada sumbu X dapat dilihat pada Gambar 6 dan Gambar 7. Gambar 6 menunjukkan data error posisi aktual end effector terhadap set point. Dari Gambar 6 terlihat bahwa error yang dihasilkan masih cukup tinggi dengan nilai maksimal 43 mm dan nilai minimal sebesar 9 mm. Gambar 7 menunjukkan boxplot dari pengujian inverse kinematics pada sumbu X. Boxplot membantu dalam melihat persebaran data pada error yang terjadi mulai dari nilai minimum, quartil 1 (Q1), quartil 2 (Q2) atau median, quartil 3 (Q3), dan nilai maksimum. Gambar 6 menunjukkan Error posisi pada sumbu X



Gambar 6. Error posisi pada sumbu X

Gambar 7 menunjukkan *Boxplot error* posisi pada sumbu X



Gambar 7. *Boxplot error* posisi pada sumbu X

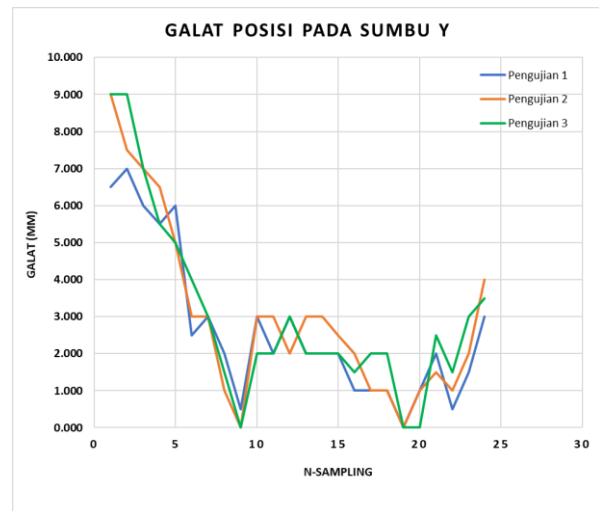
Berdasarkan Gambar 6 dan Gambar 7, dapat ditarik kesimpulan bahwa pengendalian *end effector* pada sumbu X memiliki tingkat akurasi yang kurang baik namun tingkat presisi yang tinggi. Hal ini dikarenakan, pada setiap data sampling posisi *end effector* tidak pernah mencapai nilai *set point* yang diharapkan yang ditunjukkan dari *error* posisi. Namun, di setiap pengujiannya posisi *end effector* selalu mirip antara pengujian satu dengan pengujian yang lain.

Dari pengujian pertama, kedua dan ketiga didapatkan nilai MAE dan RMSE sebagai berikut,

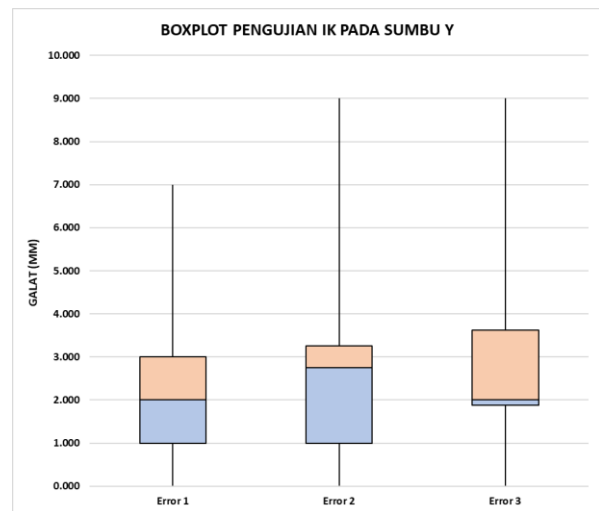
$$\begin{aligned} MAE_1 &= 18.938 & RMSE_1 &= 21.580 \\ MAE_2 &= 19.146 & RMSE_2 &= 21.527 \\ MAE_3 &= 19.125 & RMSE_3 &= 21.675 \end{aligned}$$

Dari ketiga nilai MAE dan RMSE tersebut didapatkan MAE rata-rata sebesar 19.09 dan RMSE sebesar 21.594. Dari kedua nilai tersebut dapat ditarik kesimpulan bahwa *error* rata-rata pada sumbu X adalah 19.06 mm. Namun dengan adanya selisih antara nilai MAE rata-rata dan RMSE rata-rata sekitar 2.5 menunjukkan adanya deviasi nilai antar satu sampling data dengan sampling data yang lain.

Hasil pengujian kendali posisi dengan *inverse kinematics* pada sumbu Y dapat dilihat pada Gambar 8 dan Gambar 9. Gambar 8 menunjukkan data *error* posisi aktual *end effector* terhadap *set point*. Dari Gambar 8 terlihat bahwa *error* yang dihasilkan cukup rendah dengan nilai maksimal 9 mm dan nilai minimal sebesar 0 mm. Gambar 9 menunjukkan boxplot dari pengujian *inverse kinematics* pada sumbu Y. Boxplot membantu dalam melihat persebaran data pada *error* yang terjadi mulai dari nilai minimum, kuartil 1 (Q1), kuartil 2 (Q2) atau median, kuartil 3 (Q3), dan nilai maksimum.



Gambar 8. *Error* posisi pada sumbu Y



Gambar 9. *Boxplot error* posisi pada sumbu Y

Berdasarkan Gambar 8 dan 9, dapat ditarik kesimpulan bahwa pengendalian *end effector* pada sumbu Y memiliki tingkat akurasi yang lebih baik dan tingkat presisi yang cukup tinggi jika dibandingkan dengan pengujian sebelumnya. Hal ini dikarenakan, pada setiap data sampling posisi *end effector* sudah mendekati nilai *set point* yang diharapkan. Selain itu, di setiap pengujiannya posisi *end effector* selalu mirip antara pengujian satu dengan pengujian yang lain.

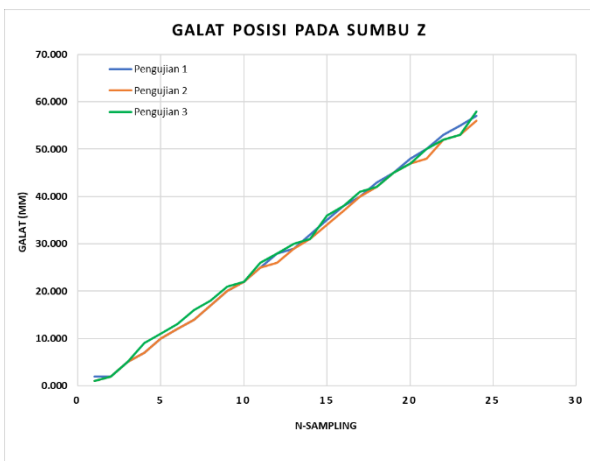
Dari pengujian pertama, kedua dan ketiga didapatkan nilai MAE dan RMSE sebagai berikut,

$$\begin{aligned} MAE_1 &= 2.667 & RMSE_1 &= 3.332 \\ MAE_2 &= 3.000 & RMSE_2 &= 3.813 \\ MAE_3 &= 3.042 & RMSE_3 &= 3.886 \end{aligned}$$

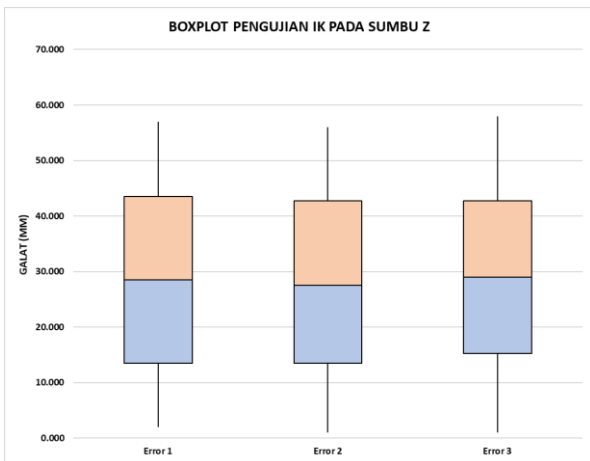
Dari ketiga nilai MAE dan RMSE tersebut didapatkan MAE rata-rata sebesar 2.903 dan RMSE sebesar 3.677. Dari kedua nilai tersebut dapat ditarik kesimpulan bahwa *error* rata-rata pada sumbu Y adalah 2.903 mm. Selain itu, dengan adanya selisih antara nilai MAE rata-rata dan

RMSE rata-rata sekitar 0.7 menunjukkan sedikit adanya deviasi nilai antar satu sampling data dengan sampling data yang lain.

Hasil pengujian kendali posisi dengan *inverse kinematics* pada sumbu Z dapat dilihat pada Gambar 10 dan Gambar 11. Gambar 10 menunjukkan data *error* posisi aktual *end effector* terhadap *set point*. Dari Gambar 10 terlihat bahwa *error* yang dihasilkan cukup tinggi dengan nilai maksimal 58 mm dan nilai minimal sebesar 10 mm. Gambar 11 menunjukkan boxplot dari pengujian *inverse kinematics* pada sumbu Z. Boxplot membantu dalam melihat persebaran data pada *error* yang terjadi mulai dari nilai minimum, kuartil 1 (Q1), kuartil 2 (Q2) atau median, kuartil 3 (Q3), dan nilai maksimum.



Gambar 10. *Error* posisi pada sumbu Y



Gambar 11. *Boxplot error* posisi pada sumbu Z

Berdasarkan Gambar 10 dan 11, dapat ditarik kesimpulan bahwa pengendalian *end effector* pada sumbu Z memiliki tingkat akurasi yang kurang baik namun tingkat presisi yang cukup tinggi mirip seperti yang terjadi pada sumbu X.

Dari pengujian pertama, kedua dan ketiga didapatkan nilai MAE dan RMSE sebagai berikut,

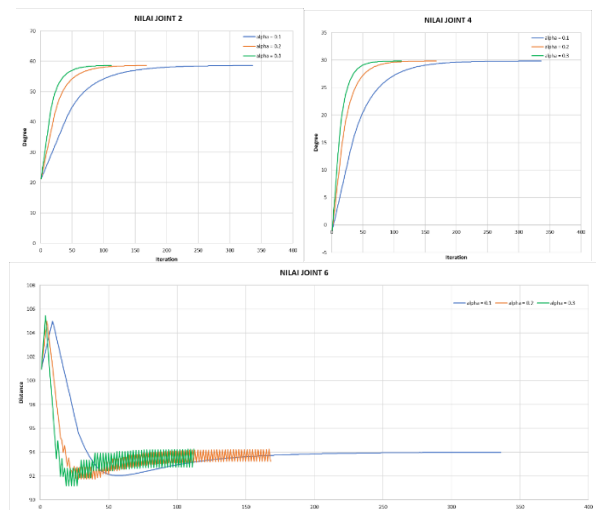
$$\begin{aligned} MAE_1 &= 28.708 & RMSE_1 &= 33.540 \\ MAE_2 &= 28.125 & RMSE_2 &= 32.832 \end{aligned}$$

$$MAE_3 = 28.958 \quad RMSE_3 = 33.555$$

Dari ketiga nilai MAE dan RMSE tersebut didapatkan MAE rata-rata sebesar 28.597 dan RMSE sebesar 33.309. Dari kedua nilai tersebut dapat ditarik kesimpulan bahwa *error* rata-rata pada sumbu Z adalah 28.597 mm. Selain itu, dengan adanya selisih antara nilai MAE rata-rata dan RMSE rata-rata sekitar 4.8 menunjukkan adanya deviasi nilai antar satu sampling data dengan sampling data yang lain.

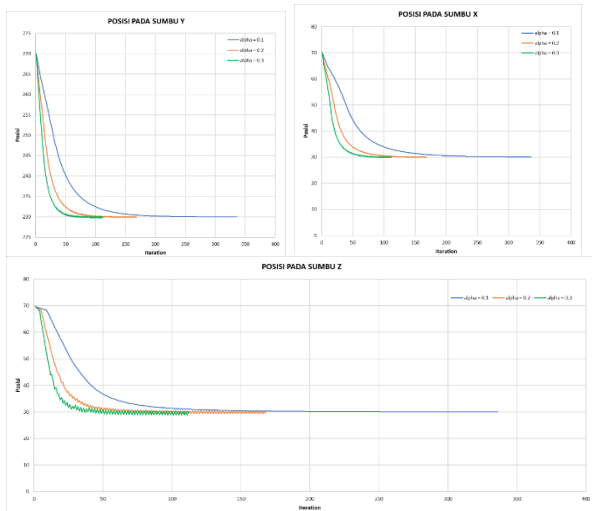
3.2. Pengujian *Trajectory Planning*

Gambar 12 menunjukkan perbandingan nilai *joint* (q) dengan nilai α (*update rate*) yang berbeda. Dari gambar tersebut terlihat bahwa setiap *joint* mencapai pada nilai yang sama namun dengan jumlah iterasi yang berbeda. Hal ini disebabkan oleh nilai α . Pada saat memperbarui nilai *joint*, perbedaan nilai *joint* (kecepatan *joint*) dikalikan dengan nilai α sebelum dijumlahkan dengan nilai *joint* sehingga nilai *joint* selalu proporsional dengan kecepatan *joint*. Jika nilai α sebesar 0.3 maka perubahan nilai *joint* adalah 0.3 dari kecepatan *joint*. Dari sinilah alasan mengapa semakin kecil nilai α semakin kecil osilasi yang terjadi pada nilai *joint*. Nilai α berfungsi untuk mengatur perubahan nilai *joint* di setiap iterasi. Akibatnya, diperlukan iterasi yang lebih banyak untuk mencapai nilai *joint* yang diharapkan.



Gambar 12. Nilai *joint* variasi α

Gambar 13 menunjukkan perbandingan posisi komputasi atau simulasi dengan nilai α ($update\ rate$) yang berbeda.



Gambar 13. Posisi komputasi variasi alpha

Dari gambar tersebut terlihat bahwa posisi pada sumbu X, Y, dan Z mencapai titik yang diharapkan yaitu (30, 230, 30) namun dengan jumlah iterasi yang berbeda. Seperti halnya pada bagian sebelumnya, perbedaan jumlah iterasi dipengaruhi oleh nilai α . Lalu, posisi sumbu Z yang sedikit beresilasi diakibatkan oleh perubahan nilai $joint\ 6$ yang beresilasi karena besar nilai $joint\ 6$ sangat mempengaruhi posisi pada sumbu Z. Posisi juga berakhir dengan nilai yang sama meski dengan jumlah iterasi yang berbeda dimana nilai α yang kecil akan memiliki jumlah iterasi terbanyak. Gambar 12 dan 13 sama-sama membentuk fungsi respon sistem orde satu orde yang menyebabkan pergerakan perubahan $end\ effector$ menjadi halus.

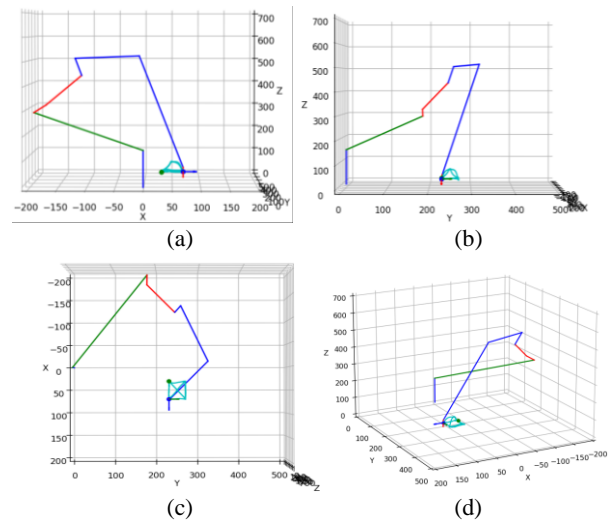
Hal yang sama juga terjadi dengan variasi jarak namun dengan nilai $joint$ dan posisi akhir yang berbeda di setiap variasi. Iterasi di setiap variasi pengujian juga mengalami perbedaan dimana semakin jauh jarak tempuh $end\ effector$ semakin besar iterasi yang diperlukan.

Dari pengujian *trajectory planning* didapatkan nilai $joint$ yang mirip antara nilai yang dikirimkan dari hasil simulasi dan nilai aktual yang diaktuator oleh aktuatur. Oleh karena itu, adanya $error$ posisi yang terjadi pada pengujian posisi dihipotesis disebabkan oleh $error$ mekanis dari robot. Hal ini karena posisi komputasi yang dicapai sudah mendekati posisi $set\ point$ yang diberikan berbeda dengan posisi aktual yang didapatkan. Posisi tersebut didapatkan melalui komputasi *forward kinematics* yang dibangun dalam bahasa python dan library numpy yang notabene merupakan bahasa dan library yang digunakan untuk komputasi ilmiah, maka dapat ditarik hipotesis bahwa permasalahan berawal dari $error$ mekanik yang terjadi akibat perbedaan kerangka robot aktual dibandingkan kerangka robot desain sehingga menyebabkan kesalahan dalam pemodelan robot dalam komputasi *inverse*

kinematics. Penyebab dari $error$ mekanik ini ada beberapa hal, diantaranya adalah $error$ yang terjadi dalam 3D Printing karena sebagian besar kerangka robot memiliki material filament PLA+, perubahan bentuk kerangka karena harus menopang beban robot, dan juga ketidakpresisian dalam pemotongan dan pembentukan kerangka. Meski demikian, tidak dapat dipungkiri bahwa penyebab $error$ posisi juga disebabkan oleh pembangunan algoritma *inverse kinematic*.

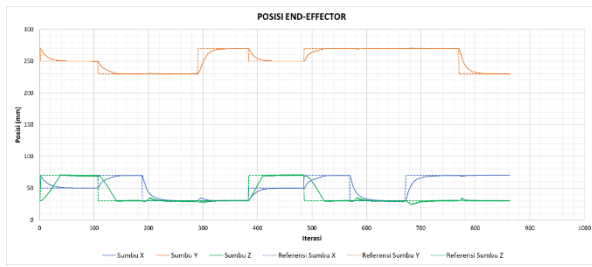
3.3. Pengujian Penggambaran Pola Piramida

Pengujian penggambaran pola piramida dilakukan didalam *working space* yang mampu dicapai oleh $end\ effector$ robot. Posisi $end\ effector$ akan digerakkan menuju 10 titik $set\ point$ untuk membentuk pola piramida. Pada pengujian ini akan dianalisis tingkat keakuratan dan kepresisian $end\ effector$ dalam mencapai titik $set\ point$, sekaligus kemampuan algoritma *trajectory planning* dalam menggambar garis. Gambar 14 menunjukkan komputasi Penggambaran Pola Piramida Oleh Manipulator (a) Tampak Depan (b) Tampak Samping (c) Tampak Atas (d) Tampak 3 Dimensi



Gambar 14. Komputasi Penggambaran Pola Piramida Oleh Manipulator (a) Tampak Depan (b) Tampak Samping (c) Tampak Atas (d) Tampak 3 Dimensi

Hasilnya $end\ effector$ robot berhasil menggambar piramida namun dengan hasil gambar yang belum sempurna seperti yang ditunjukkan oleh Gambar 14. Setiap garis yang menghubungkan titik sudut piramida tidak lurus dan sedikit berbelok meski akhirnya berakhir dititik yang diharapkan. Hal ini disebabkan karena perpindahan satu titik ke titik lainnya pada sumbu kartesian tidak linear namun membentuk layaknya fungsi sistem orde satu seperti yang ditunjukkan pada Gambar 15. Perbelokan garis ini semakin terlihat ketika perpindahan titik berpindah tidak hanya dalam satu sumbu, namun 2 atau 3 sekaligus seperti yang terjadi pada sisi tegak piramida.



Gambar 15. Posisi komputasi penggambaran pola piramida oleh manipulator

Seperti halnya apa yang terjadi pada pengujian sebelumnya, posisi *end effector* pada Gambar 15 merupakan posisi *end effector* hasil komputasi *forward kinematics*. posisi aktual yang dihasilkan memiliki tingkat akurasi yang cukup rendah dengan error maksimum yang terjadi pada pengujian ini sebesar 19.5 mm untuk sumbu X, 10 mm untuk sumbu Y, dan 37 mm untuk sumbu Z. Meski demikian, tingkat presisi dari robot ini teruji kembali dengan nilai yang tinggi karena setiap kali *end effector* kembali ke titik *set point* yang sama, *end effector* aktual berada di posisi yang mendekati posisi sebelumnya.

4. Kesimpulan

Berdasarkan pembahasan diatas, dapat disimpulkan sistem manipulator robot bedah 6 DOF dengan ROC mekanik untuk engendalian *end-effector* berhasil dirancang dengan baik. Tingkat akurasi pengendalian posisi *end effector* robot cukup rendah dimana nilai MAE dan RMSE rata-rata pada sumbu X, Y, dan Z berturut-turut adalah 19.09 dan 21.594, 2.903 dan 3.677, 28.597 dan 33.309. Meski demikian, dari tiga kali perulangan pengujian didapatkan nilai MAE dan RMSE yang sama sehingga pengendalian posisi *end effector* robot memiliki tingkat akurasi yang rendah, namun dengan tingkat presisi yang tinggi. Selain itu, Algoritma *trajectory planning* menyatu dengan algoritma *inverse kinematics* dan dipengaruhi oleh α (*update rate*) dan juga jarak tempuh *end effector* yang menghasilkan pergerakan perubahan posisi yang halus. Algoritma tersebut menyebabkan pergerakan perpindahan posisi *end effector* di setiap sumbunya yang mirip seperti respon sistem orde 1.

Referensi

- [1] D. Oleynikov, "Robotic Surgery," *Surg. Clin. North Am.*, vol. 88, no. 5, pp. 1121–1130, 2008, doi: 10.1016/j.suc.2008.05.012.
- [2] K. Kikuchi, K. Suda, S. Shibasaki, T. Tanaka, and I. Uyama, "Challenges in improving the minimal invasiveness of the surgical treatment for gastric cancer using robotic technology," *Ann. Gastroenterol. Surg.*, vol. 5, no. 5, pp. 604–613, 2021, doi: 10.1002/ags3.12463.
- [3] C. P. Childers, "Estimation of the Acquisition and

Operating Costs for Robotic Surgery," *J. Am. Med. Assoc.*, vol. 320, no. 8, pp. 835–836, 2018, doi: 10.1111/ans.13856.

- [4] B. Hannaford *et al.*, "Raven-II: An open platform for surgical robotics research," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 954–959, 2013, doi: 10.1109/TBME.2012.2228858.
- [5] H. King, S. N. Kosari, and B. Hannaford, "Kinematic Analysis of the Raven-II TM Research Surgical Robot Platform UWEE Technical Report Number UWEETR-2012-0006 Kinematic Analysis of the Raven-II TM Research Surgical Robot Platform," no. 206, 2012.
- [6] M. M. Marinho, M. C. Bernardes, and A. P. L. Bo, "Using General-Purpose Serial-Link Manipulators for Laparoscopic Surgery with Moving Remote Center of Motion," *J. Med. Robot. Res.*, vol. 1, no. 4, 2016, doi: 10.1142/S2424905X16500070.
- [7] D. Nenchev, *Humanoid Robotics: A Reference*. 2018.
- [8] P. Corke, *Robotics, Vision and Control*, vol. 118. Cham: Springer International Publishing, 2017.