

SIMULASI ROBOT MANIPULATOR 4 DOF SEBAGAI MEDIA PEMBELAJARAN DALAM KASUS ROBOT MENULIS HURUF

*Rendyansyah dan Aditya P. P. Prasetyo

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

*Corresponding author, e-mail: rendyansyah@ilkom.unsri.ac.id

Abstrak— Dalam area industri telah banyak digunakan teknologi robotika untuk melakukan berbagai pekerjaan. Teknologi robotik dimanfaatkan karena memudahkan aktivitas pekerjaan manusia. Salah satu teknologi robotika dalam industri yaitu robot manipulator. Seorang operator perlu mempelajari tentang operasi dan proses dari sistem robot manipulator supaya dapat menggunakan robot tersebut. Penelitian ini menghasilkan simulasi robot manipulator dengan empat derajat kebebasan, contoh kasusnya robot menulis huruf menggunakan metode *cubic trajectory*. Adapun motivasi dari penelitian ini pada dasarnya untuk menggunakan robot manipulator yang ada di Laboratorium Robotika Fakultas Ilmu Komputer UNSRI sebagai bahan ajar matakuliah robotika. Simulasi robot ini dapat memberikan pengetahuan kepada peserta didik untuk memahami perencanaan jalur pada robot manipulator. Hasil simulasi robot berjalan dengan baik dan robot dapat menuliskan pola huruf vokal kapital yang diberikan.

Kata Kunci : Cubic trajectory, Robot manipulator, Simulasi.

Abstract— Industrial area has been widely used robotics technology to perform various work. Robotic technology is used because it allows the activity of human work. One of robotics technology in the industry namely robot manipulator. An operator needs to learn about operations and process of robotic manipulator system to use the robot. This research results a simulated robot manipulator with four degrees of freedom, for example a robot write letters using methods “cubic trajectory”. The motivation of this research is based on the use of robot manipulators in the Robotics Laboratory of the Faculty of Computer Science UNSRI as a teaching materials of robotics introduction course. This robot simulation can provide knowledge for learners to understand the robot manipulator path planning. The result of robot simulation runs well and the robots can write capital vowel patterns which are given.

Keywords : Cubic trajectory, Robot manipulator, Simulation.

Copyright © 2016 JNTE. All rights reserved

1. PENDAHULUAN

Perkembangan teknologi robotika telah dirasakan dalam kehidupan manusia, terutama dalam bidang industri. Teknologi robotika memberi kemudahan dalam aktivitas pekerjaan manusia, salah satu contoh robot industri yang lebih dikenal yaitu robot manipulator. Robot manipulator menjadi salah satu topik dari robotika yang banyak dikaji oleh peneliti atau praktisi, contohnya perencanaan jalur untuk robot.

Robot manipulator merupakan peralatan mekanik yang umumnya dirancang dengan menggunakan aktuator seperti motor listrik, atau pneumatik. Kebanyakan robot manipulator berbentuk lengan yang dirancang khusus sesuai fungsi dan tujuan yang diinginkan [1][2]. Robot

manipulator banyak dipakai dalam industri untuk meringankan pekerjaan yang rumit, contohnya robot pengelasan dan pemotongan bahan logam. Dalam aplikasi robot manipulator dapat dikendalikan baik manual maupun otomatis yang mampu menghasilkan gerakan yang diinginkan.

Gerakan robot manipulator bergantung dari konfigurasi robot. Konfigurasi robot menentukan fungsi dan kegunaan dari robot tersebut [1][2][3]. Salah satu bentuk konfigurasi robot manipulator dalam industri yaitu model robot SCORBOT. Konfigurasi robot SCORBOT dirancang untuk bergerak dalam koordinat (x,y,z) . Dalam aplikasi industri, robot perlu dirancang lintasan pergerakan yang sesuai dengan tugasnya. Perencanaan jalur disebut sebagai metode *trajectory*. Trajectory merupakan suatu lintasan dari gerakan yang dirancang secara spesifik dari

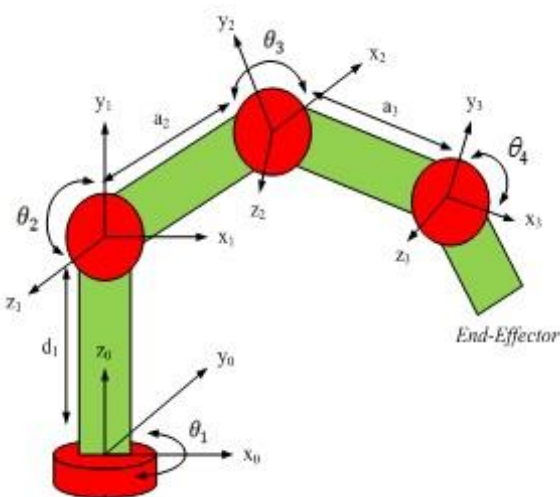
setiap perubahan waktu pada *end-effector* robot [4][5]. Rendyansyah dkk [5] telah menerapkan salah satu *trajectory* robot yaitu *linear trajectory* untuk simulasi robot SCARA. Metode *linier trajectory* telah memberikan hasil yang baik dalam kasus robot menulis huruf, dan gerakan menulisnya berdasarkan titik koordinat (q_1) menuju koordinat (q_2) sehingga huruf dibentuk berdasarkan garis lurus. Adapun metode pengembangan dari *linear trajectory*, yaitu *cubic trajectory planning* [4].

Pada penelitian ini telah dirancang dan disimulasikan pergerakan robot manipulator dengan empat derajat kebebasan menggunakan *cubic trajectory* dalam menuliskan huruf vokal kapital. Penelitian ini untuk memotivasi dalam pembelajaran robotika khususnya penggunaan robot manipulator yang ada di dalam Laboratorium Robotika FASILKOM UNSRI, serta memudahkan pemahaman dalam pembelajaran robot manipulator.

2. TINJAUAN PUSTAKA

2.1. Robot Manipulator

Robot manipulator umumnya berbentuk lengan yang dapat digunakan dalam berbagai aplikasi dibidang industri dan medis. Robot manipulator memiliki sendi yang dapat bergerak menggunakan aktuator. Dalam perancangan robot manipulator yang sederhana umumnya paling banyak digunakan motor servo karena mudah dalam penggunaan dan mengontrol gerakannya [6].



Gambar 1. Bentuk robot manipulator 4 DOF.

Pada bagian *end-effector* robot manipulator memiliki peran aktif dalam melakukan aksi yang diinginkan, misalnya terdapat sensor pendeteksi untuk *men-sensing* suatu zat atau gripper untuk mengambil objek, dan lain-lain [1][3][7]. Adapun bentuk robot manipulator dengan 4 derajat kebebasan (DOF) seperti yang ditunjukkan pada Gambar 1.

Berdasarkan pada Gambar 1, didapatkan hasil perumusan *forward* dan *invers kinematic*. Persamaan *forward kinematic* didapatkan dengan menggunakan metode Denavit-Hartenberg disingkat DH [8], seperti pada (1). Adapun parameter DH dari robot manipulator 4 DOF ditunjukkan pada Tabel 1.

Tabel 1. Parameter DH untuk robot manipulator 4 DOF.

Joint i	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	d_1	θ_1
2	0	a_2	0	θ_2
3	0	a_3	0	θ_3
4	0	a_4	0	θ_4

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

Persamaan (1) diintegrasikan dengan *joint-joint* pada Tabel 1 sehingga diperoleh empat matrik A seperti pada (2), (3), (4) dan (5).

$${}^0A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^1A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^2A_3 = \begin{bmatrix} C\theta_3 & -S\theta_3 & 0 & a_3 C\theta_{23} \\ S\theta_3 & C\theta_3 & 0 & a_3 S\theta_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^3A_4 = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & a_4C\theta_{234} \\ S\theta_4 & C\theta_4 & 0 & a_4S\theta_{234} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Dengan mengalikan empat matrik A seperti pada ${}^0\hat{Q} = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4\hat{Q}$, dimana matrik ${}^4\hat{Q} = [0 \ 0 \ 0 \ 1]^T$ untuk mendapatkan solusi kinematiknya yaitu pada kolom keempat dari hasil perkalian empat matrik A tersebut (${}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$) sehingga menjadi matrik ${}^0\hat{Q}$ seperti pada (6).

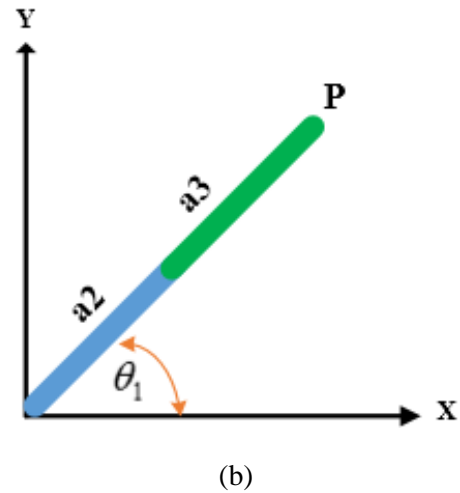
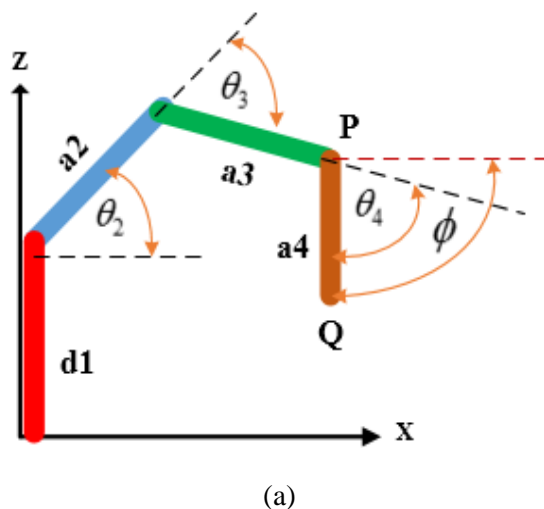
$${}^0\hat{Q} = \begin{bmatrix} C\theta_1(a_2C\theta_2 + a_3C\theta_{23} + a_4C\theta_{234}) \\ S\theta_1(a_2C\theta_2 + a_3C\theta_{23} + a_4C\theta_{234}) \\ d_1 + a_2S\theta_2 + a_3S\theta_{23} + a_4S\theta_{234} \\ 1 \end{bmatrix} \quad (6)$$

$$q_x = C\theta_1(a_2C\theta_2 + a_3C\theta_{23} + a_4C\theta_{234}) \quad (7)$$

$$q_y = S\theta_1(a_2C\theta_2 + a_3C\theta_{23} + a_4C\theta_{234}) \quad (8)$$

$$q_z = d_1 + a_2S\theta_2 + a_3S\theta_{23} + a_4S\theta_{234} \quad (9)$$

Oleh karena itu hasil persamaan *forward kinematic* untuk sumbu x, y dan z seperti pada (7), (8) dan (9). Ini berarti bahwa nilai x, y dan z diperoleh apabila diberi masukkan sudut $\theta_1, \theta_2, \theta_3$, dan θ_4 .



Gambar 2. Skematik robot manipulator 4 DOF, (a) tampak samping, (b) tampak atas.

Selanjutnya untuk mendapatkan persamaan *invers kinematic* menggunakan konsep trigonometri [5][8]. Pada *invers kinematic* yang menjadi masukan yaitu nilai koordinat x, y dan z sehingga diperoleh nilai keluaran masing-masing sudut. Persamaan *invers kinematic* pada robot menggunakan bantuan ilustrasi pada Gambar 1 yang juga diuraikan dalam bentuk skematik seperti pada Gambar 2.

Pada gambar 2.a, dapat dicari sudut θ_2, θ_3 dan θ_4 dengan menerapkan konsep aturan kosinus [4]. Pada Gambar 2.a terlihat adanya sudut ϕ yaitu sudut orientasi atau penjumlahan nilai sudut θ_2, θ_3 dan θ_4 . Pada penelitian ini ditentukan sudut ϕ sebesar -90° terhadap sumbu-x supaya *end-effector* (titik Q) selalu menghadap ke arah bawah (sumbu-x), sehingga dapat digunakan dalam proses menulis. Oleh karena itu dalam mencari sudut θ_2, θ_3 dan θ_4 maka perlu dicari nilai P(x,y) terlebih dahulu seperti pada (10),

$$\begin{aligned} Q_z &= \sqrt{(Q_x)^2 + (Q_y)^2} \\ P_x &= Q_z - (a_4 \cdot c\phi) \\ P_y &= Q_z - d_1 - (a_4 \cdot s\phi) \end{aligned} \quad (10)$$

sehingga diperoleh sudut θ_3, θ_2 dan θ_4 seperti pada (11), (12) dan (13).

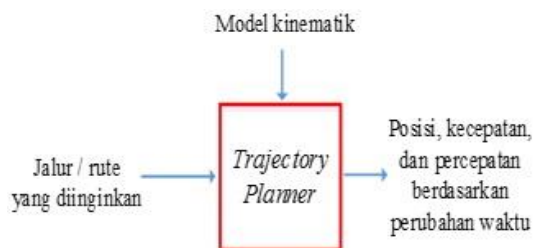
$$\theta_3 = \cos^{-1} \left(\frac{(P_x)^2 + (P_y)^2 - (a_2)^2 - (a_3)^2}{2a_2a_3} \right) \quad (11)$$

$$\theta_2 = \tan^{-1} \left(\frac{P_y}{P_x} \right) - \tan^{-1} \left(\frac{a_3 \cdot s \theta_3}{a_2 + a_3 \cdot c \theta_3} \right) \quad (12)$$

$$\theta_4 = \phi - \theta_2 - \theta_3 \quad (13)$$

Selanjutnya nilai $\cos(\theta_1)$ diperoleh dengan menganalisis skematik robot tampak atas (Gambar 2.b), dimana nilai sumbu- x dibagi panjang lengan (P) sehingga didapatkan sudut θ_1 seperti pada (14).

$$\theta_1 = \cos^{-1} \left(\frac{Q_x}{a_2 \cdot c \theta_2 + a_3 \cdot c \theta_{23} + a_4 \cdot c \theta_{234}} \right) \quad (14)$$



Gambar 3. Diagram blok *trajectory planning*.

2.2. Cubic trajectory planning.

Trajectory merupakan suatu pendekatan dari perencanaan jalur yang diinginkan dengan menggunakan fungsi polynomial, dengan *step by step* gerakannya berdasarkan perubahan waktu [9]. Adapun diagram blok *trajectory planning* seperti ditunjukkan pada Gambar 3. *Trajectory* ini umumnya dikombinasikan dengan perumusan *invers kinematic* dari lengan robot, dalam hal ini dapat disebut sebagai pendekatan berbasis *workspace* [5][10]. *Trajectory* mempunyai beberapa metode yaitu *linier*, *parabolic*, *cubic trajectory* dan lain-lain. Dalam penelitian ini diimplementasikan metode *cubic trajectory*, yaitu metode pengembangan dari *linier* dan *parabolic trajectory*.

Metode *cubic trajectory* pada dasarnya merupakan persamaan polynomial orde tiga seperti yang ditunjukkan pada (15). Polynomial orde tiga akan menghasilkan tiga respon yaitu *position*, *velocity* dan *acceleration*.

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (15)$$

Persamaan (15) ini digunakan untuk menentukan persamaan dari titik 1 ke titik 2 secara terhubung. Misalkan koordinat titik 1 (x, y, z) dengan $t = t_0$ detik menuju koordinat titik 2 (x, y, z) di $t = t_f$ detik, dan solusinya terlebih dahulu dimulai pada (16).

$$\text{Posisi: } q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (16)$$

$$\text{kecepatan: } \dot{q}(t) = a_1 + 2a_2t + 3a_3t^2$$

Substitusikan waktu awal dan akhir pada (16), sehingga menjadi (17),

$$t = 0 \rightarrow q(0) = a_0, \dot{q}(0) = a_1$$

$$t = t_f \rightarrow q(t_f) = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 \quad (17)$$

$$\dot{q}(t_f) = a_1 + 2a_2t_f + 3a_3t_f^2$$

asumsikan posisi awal dan akhir ditunjukkan dalam (18).

$$t = 0 \rightarrow q(0) = q_s, \dot{q}(0) = v_s \quad (18)$$

$$t = t_f \rightarrow q(t_f) = q_f, \dot{q}(t_f) = v_f$$

Inisialisasikan bahwa $v_s = 0$ dan $v_f = 0$, maka (17) dan (18) dapat disubstitusikan dan eliminasi sehingga menghasilkan (19),

$$\begin{cases} a_0 = q_s \\ a_1 = v_s = 0 \\ a_2 = 3 \left(\frac{q_f - q_s}{t_f^2} \right) \\ a_3 = -2 \left(\frac{q_f - q_s}{t_f^3} \right) \end{cases} \quad (19)$$

sehingga didapatkan penyelesaian dari metode *cubic trajectory planning* seperti pada (20).

$$q(t) = q_s + 3 \left(\frac{q_f - q_s}{t_f^2} \right) t^2 - 2 \left(\frac{q_f - q_s}{t_f^3} \right) t^3 \quad (20)$$

Persamaan (20) merupakan persamaan *cubic trajectory* yang diimplementasikan ke dalam simulasi robot manipulator 4 DOF dalam menuliskan pola huruf. Pergerakan robot manipulator tersebut berdasarkan perubahan koordinat $Q_1(x, y, z)$ menuju $Q_2(x, y, z)$ dari setiap perubahan waktunya, dan rumusnya seperti pada (21), (22) dan (23).

$$q_x(t) = q_{xs} + 3\left(\frac{q_{xf} - q_{xs}}{t_{yf}^2}\right)t^2 - 2\left(\frac{q_{xf} - q_{xs}}{t_{yf}^3}\right)t^3 \quad (21)$$

$$q_y(t) = q_{ys} + 3\left(\frac{q_{yf} - q_{ys}}{t_{yf}^2}\right)t^2 - 2\left(\frac{q_{yf} - q_{ys}}{t_{yf}^3}\right)t^3 \quad (22)$$

$$q_z(t) = q_{zs} + 3\left(\frac{q_{zf} - q_{zs}}{t_{zf}^2}\right)t^2 - 2\left(\frac{q_{zf} - q_{zs}}{t_{zf}^3}\right)t^3 \quad (23)$$

3. METODOLOGI

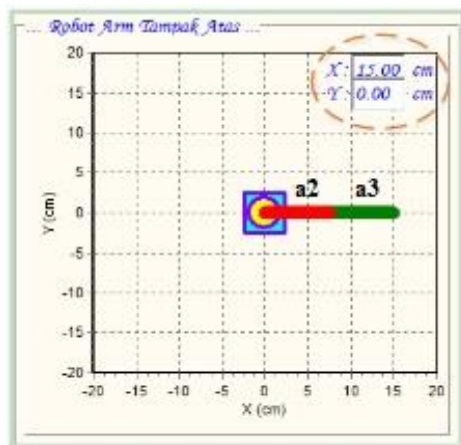
Robot manipulator 4 DOF dalam penelitian ini menggunakan simulasi dalam program visual. Program simulasi dibagi menjadi dua bagian yaitu simulasi robot manipulator untuk kondisi *forward* dan juga kondisi *invers*. Persamaan

invers kinematic pada robot manipulator dikombinasikan dengan rumus *cubic trajectory* sehingga dapat digunakan untuk aplikasi tertentu, dalam hal ini aplikasi robot menulis huruf kapital.

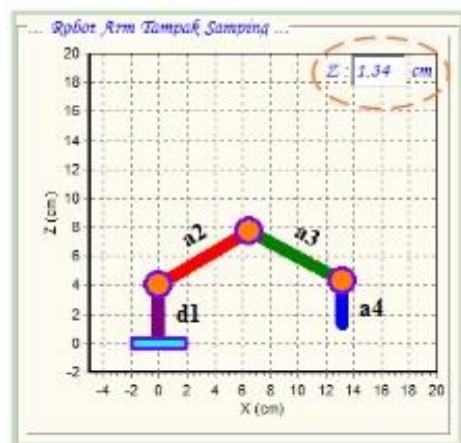
3.1. Perancangan Simulasi Robot Manipulator untuk Kondisi Forward Kinematic.

Persamaan (7), (8) dan (9) merupakan posisi *end-effector* jika diberikan masukkan sudut $\theta_1, \theta_2, \theta_3$ dan θ_4 . Pada perancangan simulasi perlu diketahui ukuran masing-masing link lengan robot. Dalam penelitian ini ditentukan link (d_1) 4 cm, link a_2 dan a_3 masing-masing 7.5 cm dan link (a_4) 3 cm. ukuran link-link tersebut menentukan seberapa jauh jangkauan lengan robot tersebut. Gambar 4 menunjukkan robot manipulator tampak atas dan tampak samping. Berdasarkan Gambar 4(a) bahwa jangkauan robot sejauh 15 cm karena link a_2 ditambah dengan link a_3 .

Pada Gambar 4(b) menunjukkan seberapa tinggi saat robot bergerak terhadap koordinat sumbu-z. Perubahan nilai koordinat x, y , dan z dipengaruhi oleh semua sudut robot. Sudut θ_1 memiliki rentang sudut 0° sampai 360° , sudut θ_2 yaitu 0° sampai 90° , sudut θ_3 -180° sampai 0° , dan sudut θ_4 -90° sampai 0° .

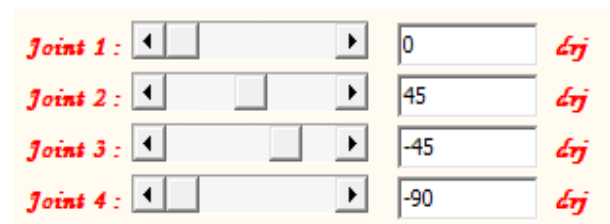


(a)



(b)

Gambar 4. Simulasi robot manipulator (a) tampak atas (sumbu x dan y), dan (b) tampak samping (sumbu x dan z)



Gambar 5. Tampilan input sudut *joint* robot manipulator.

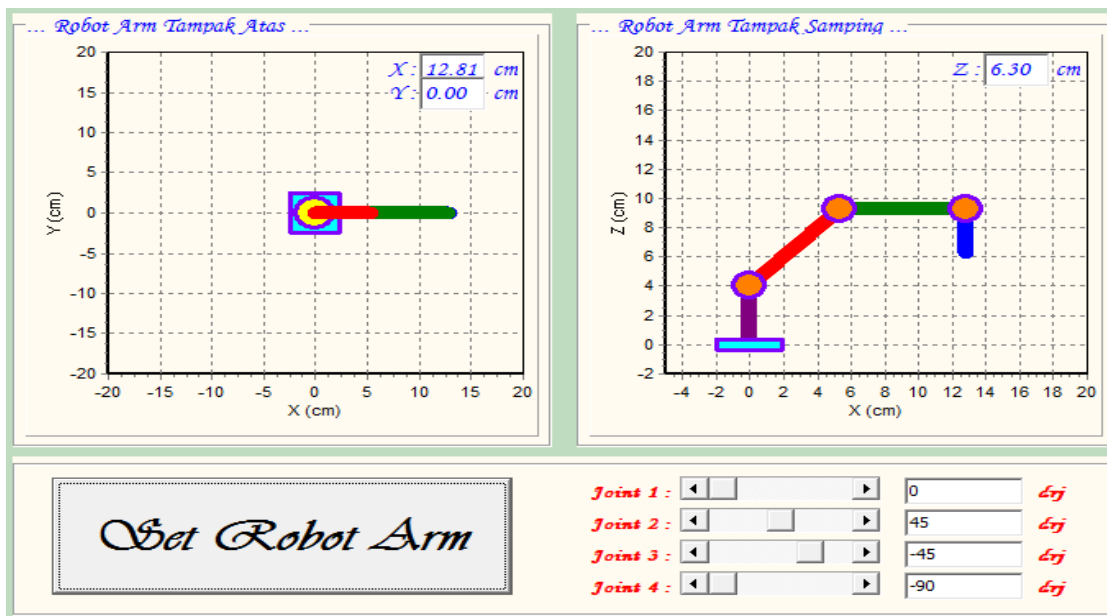
Adapun tampilan masukan sudut *joint* robot manipulator seperti ditunjukkan pada Gambar 5, dan tampilan simulasi robot manipulator untuk perumusan *forward kinematic* ditunjukkan pada Gambar 6.

3.2. Perancangan Simulasi Robot Manipulator untuk Kondisi Invers Kinematic.

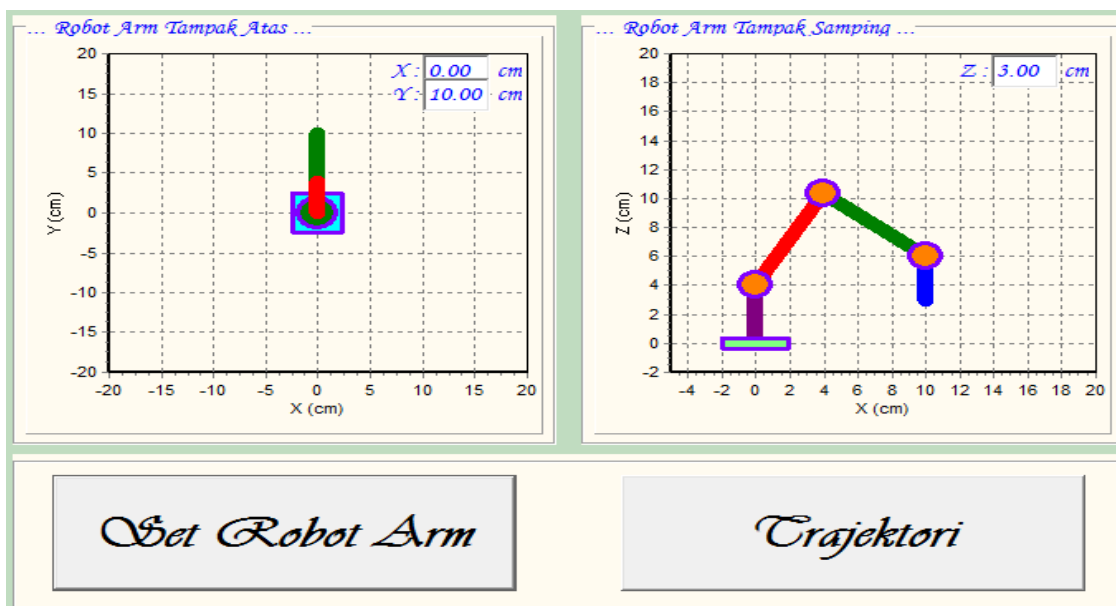
Pada tahap ini sama halnya seperti pada perancangan tahap sebelumnya, namun yang

membedakannya yaitu masukkan. Masukkan berupa koordinat x , y dan z untuk mendapatkan nilai sudut $joint$. Dalam koordinat tersebut dapat dibentuk sebuah pola tertentu, dalam penelitian ini dibuatlah pola yang membentuk huruf kapital. Adapun tampilan simulasi robot manipulator untuk kondisi perumusan *invers kinematic* ditunjukkan pada Gambar 7. Pada Gambar 7, posisi origin robot manipulator pada $Q(0,10,3)$ cm. Robot manipulator tersebut bergerak sesuai koordinat yang ditentukan. Pergerakan dari titik

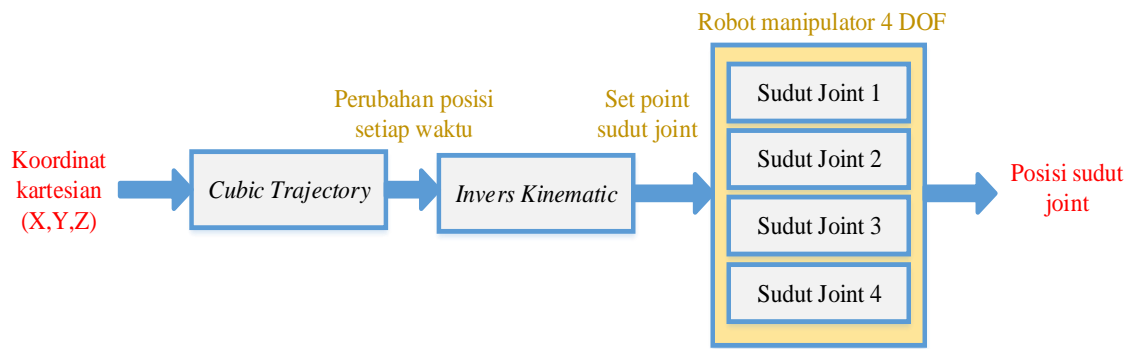
koordinat satu ke titik lainnya yang membentuk sebuah jalur. Di dalam penelitian ini pergerakan robot menggunakan *cubic trajectory* seperti pada (21) sampai (23). Untuk lebih jelasnya ditunjukkan kombinasi *cubic trajectory* dengan *invers kinematic* pada Gambar 8. Adapun sudut θ_1 sampai θ_4 diperoleh dari analisa *invers kinematic* pada perolehan rumus dalam (11) sampai (14), dalam hal ini guna menyesuaikan pola jalur dalam koordinat (x,y,z) .



Gambar 6. Tampilan simulasi robot manipulator untuk kondisi perumusan *forward kinematic*.



Gambar 7. Tampilan simulasi robot manipulator untuk kondisi perumusan *invers kinematic*.



Gambar 8. Kombinasi *cubic trajectory* dengan *invers kinematic*.

4. HASIL DAN PEMBAHASAN

Penelitian ini terdapat dua kondisi robot manipulator yaitu *forward* dan *invers* sehingga terdapat dua pengujian. Pengujian pertama untuk kondisi *forward kinematic* robot manipulator, dan kedua untuk kondisi *invers kinematic* yang dalam hal ini langsung pergerakan robot yang membentuk pola huruf.

4.1. Pengujian robot manipulator untuk kondisi *forward kinematic*.

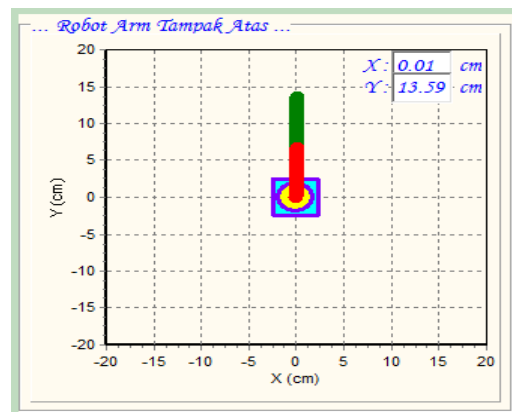
Pada pengujian ini robot manipulator diberi masukkan sudut θ_1 sampai θ_4 sehingga robot bergerak berdasarkan besarnya nilai sudut tersebut. Pada Gambar 6 memperlihatkan posisi origin di sudut θ_1 sampai θ_4 masing 0° , 45° , -45° dan -90° , yang membentuk koordinat (12.81, 0, 6.30) cm. Gambar 9 menunjukkan simulasi pengujian robot manipulator untuk kondisi *forward kinematic*.

Tabel 2. Hasil pengujian simulasi robot manipulator untuk kondisi *forward kinematic*.

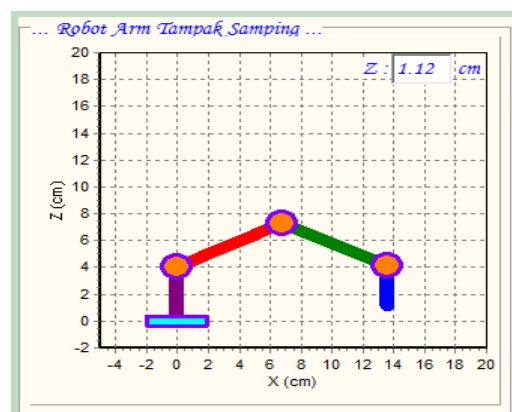
Input sudut joint				Koordinat (cm)		
1	2	3	4	X	Y	Z
90°	26°	-51°	-64°	0	13.59	1.12
51°	62°	-123°	-30°	4.48	5.53	1.06
154°	23°	-57°	-58°	-11.70	5.72	-0.26
0°	0°	0°	-90°	15	0	1
45°	45°	-45°	-45°	10.56	10.55	7.18

Pada Gambar 9 dapat dilihat simulasi robot manipulator bisa bergerak apabila diberi salah satu masukkan sudut θ_1 sampai θ_4 . Secara acak

diberikan masukkan sudut θ_1 sampai θ_4 masing-masing di-set 90° , 26° , -51° dan -64° , sehingga *end-effector* robot berada pada koordinat $x = 0$ cm, $y = 13.59$ cm dan $z = 1.12$ cm. Percobaan ini dilakukan sebanyak 5 kali dengan masukkan sudut *joint* yang berbeda-beda sehingga keluaran koordinatnya juga berbeda. Hasil pengujian simulasi robot manipulator untuk kondisi *forward kinematic* seperti ditunjukkan pada Tabel 2.



(a)



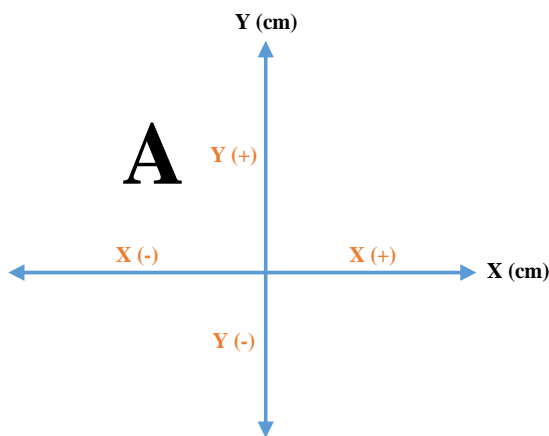
(b)



(c)

Gambar 9. Pengujian robot manipulator untuk kondisi *forward kinematic*.

- (a) pergerakan robot tampak atas,
- (b) pergerakan robot tampak samping, dan
- (c) input sudut $joint \theta_1$ sampai θ_4 dalam derajat.



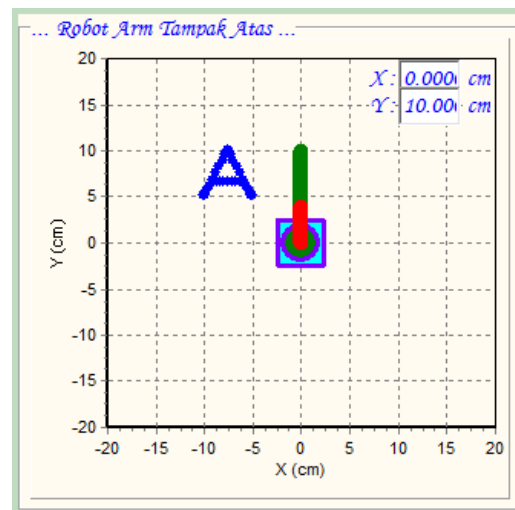
Gambar 10. Contoh koordinat yang membentuk pola huruf.

4.2. Pengujian Robot Manipulator untuk Kondisi Invers Kinematic.

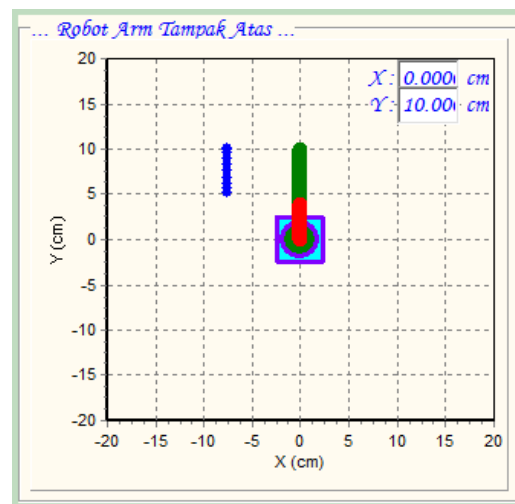
Pada tahap pengujian ini diberikan masukkan koordinat yang membentuk pola huruf seperti yang ditunjukkan pada Gambar 10. Pada arah sumbu-z merupakan gerakan end-effector saat menulis atau tidak menulis. Pola huruf yang diujikan yaitu huruf vokal “A”, “I”, “U”, “E” dan “O” seperti ditunjukkan pada Tabel 3. Pada Tabel 3 dapat dilihat bahwa pola huruf “I” memiliki x, y dan z paling sedikit, sedangkan pola huruf “O” paling banyak. Perbedaan nilai x, y dan z pada setiap pola karena koordinat x, y dan z ditentukan dengan mengilustrasikan arah gerak end-effector robot saat menuliskan pola huruf vokal tersebut. Oleh karena itu trajectory pola huruf “A” terbentuk dari enam langkah gerakan, pola “I” tiga langkah, pola “U” tujuh

langkah, pola “E” dua belas langkah dan pola “O” empat belas langkah. Adapun nilai-nilai x, y dan z tersebut dipilih berdasarkan ketentuan masukkan dari user yang membentuk sebuah trajectory pola huruf, yang mana data tersebut disimpan di dalam notepad. Dengan kata lain trajectory pola huruf bisa juga di titik koordinat lain asalkan masih bisa dijangkau oleh robot.

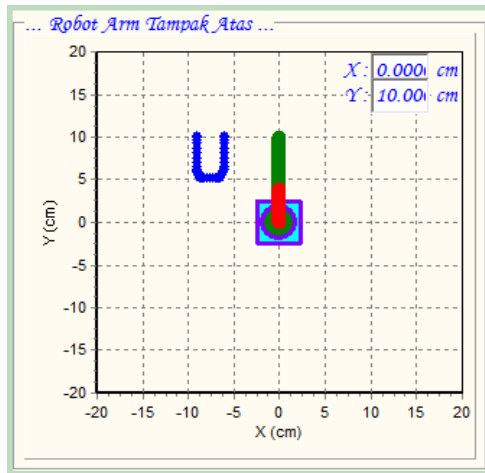
End-effector robot bergerak sesuai koordinat $Q1(x,y,z)$ menuju $Q2(x,y,z)$ dan seterusnya sampai selesai. Adapun hasil simulasi robot manipulator dalam menuliskan huruf “A”, “I”, “U”, “E” dan “O” masing-masing seperti ditunjukkan pada Gambar 11, Gambar 12, Gambar 13, Gambar 14 dan Gambar 15.



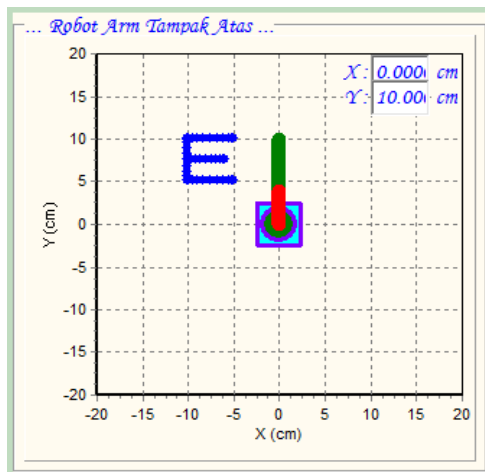
Gambar 11. Hasil simulasi robot manipulator dalam menulis huruf “A”.



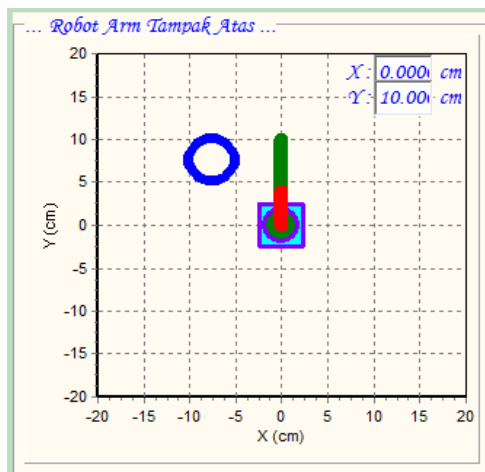
Gambar 12. Hasil simulasi robot manipulator dalam menulis huruf “I”.



Gambar 13. Hasil simulasi robot manipulator dalam menulis huruf “U”.



Gambar 14. Hasil simulasi robot manipulator dalam menulis huruf “E”.



Gambar 15. Hasil simulasi robot manipulator dalam menulis huruf “O”.

Tabel 3. Koordinat pola huruf sebagai *trajectory* pada robot manipulator.

Pola	Koordinat (cm)		
	X	Y	Z
A	0	10	3
	-10	5	1
	-7.5	10	1
	-5	5	1
	-9	6.5	3
	-9	6.5	1
	-6	6.5	1
I	0	10	3
	-7.5	10	1
	-7.5	5	1
	0	10	3
U	0	10	3
	-9	10	1
	-9	6	1
	-8.5	5	1
	-6.5	5	1
	-6	6	1
	-6	10	1
	0	10	3
E	0	10	3
	-10	10	1
	-10	5	1
	-10	10	3
	-10	10	1
	-5	10	1
	-10	7.5	3
	-10	7.5	1
	-6	7.5	1
	-10	5	3
	-10	5	1
	-5	5	1
	0	10	3
O	0	10	3
	-8	5	1
	-9	5.5	1
	-10	7	1
	-10	8	1
	-9	9.5	1
	-8	10	1
	-7	10	1
	-6	9.5	1
	-5	8	1
	-5	7	1
	-6	5.5	1
	-7	5	1
	-8	5	1
0	10	3	

Berdasarkan percobaan simulasi robot tersebut memperlihatkan bahwa robot manipulator mampu menuliskan pola huruf vokal yang telah ditentukan. Pola huruf ini merupakan *trajectory* berdasarkan titik-titik koordinatnya sehingga perubahan sudut θ_1 sampai θ_4 selalu menyesuaikan jalur *end-effector* robot manipulator. Gerakan *end-effector* robot dalam menuliskan huruf vokal ini dengan menggunakan metode *cubic trajectory*. *Cubic trajectory* yang diimplementasikan ke dalam robot manipulator mampu menuliskan huruf dengan baik, ini dapat dilihat dari huruf yang terplot di dalam simulasi pada Gambar11 sampai Gambar15. Setelah robot menuliskan huruf maka robot kembali pada posisi aslinya.

5. KESIMPULAN

Robot manipulator sama halnya dengan lengan robot. Pada penelitian ini telah dirancang simulasi robot manipulator dengan empat derajat kebebasan. Simulasi ini sebagai media ajar matakuliah robotika di Jurusan Sistem Komputer UNSRI untuk membuktikan kebenaran dari teori yang dipelajari. Pengujian robot telah dilakukan, baik untuk kondisi *forward* maupun *invers*, dan hasil simulasinya baik. Pada percobaan *invers kinematic* langsung diaplikasikan untuk menulis huruf dengan mengimplementasikan *cubic trajectory* ke dalam simulasi. Hasilnya menunjukkan bahwa robot manipulator dapat menuliskan pola huruf vokal yang diberikan. Adapun pengembangan penelitian selanjutnya yaitu simulasi robot dalam menuliskan kata.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Jurusan Sistem Komputer dan Fakultas Ilmu Komputer Universitas Sriwijaya.

DAFTAR PUSTAKA

- [1] Gustica, A.N., Rivai, M., dan Tasripan, *Implementasi Sensor Gas pada Kontrol Lengan Robot Untuk Mencari Sumber Gas*. Jurnal Teknik POMITS, vol. 3, (2014).
- [2] Darwison, Rusydi, M.I., dan Bentar. *Kontrol Posisi Robot Manipulator Planar Tiga Derajat Kebebasan Berbasis Visual*.

- Jurnal Nasional Teknik Elektro. Vol. 2, no. 1 (2013).
- [3] Ramya, V., Palaniappan, B., dan Akilan, T. *Embedded System for Robotic Arm Movement Control Using Web Server and Zigbee Communication*. International Conference on Research Trends in Computer Technologies, ICRTCT, pp. 30-34, (2013).
- [4] Biagiotti, L., dan Melchiorri, C. *Trajectory Planning for Automatic machines and Robots*, Springer, Germany, (2008).
- [5] Rendyansyah, Passarella, R., dan Exaudi, K. *Aplikasi Linier Trajectory Planning Pada Simulasi Pergerakan Robot SCARA 3 DOF Dalam Menulis Huruf*. Proceeding Annual Research Seminar, vol. 1, no. 1, (2015).
- [6] Li, S., Weng, C., Chen, Y., Lo, C., Yang, M., Lin, Y., Hsieh, M., and Wong, C. *Servo Motor Controller Design for Robotic Manipulator*. IEEE International Symposium on Intelligent Signal Processing and Communication System (ISPACS), pp. 254-257, (2012).
- [7] Rivai, M., Rendyansyah dan Purwanto, D. *Implementation of Fuzzy Logic Control in Robot Arm for Searching Location of Gas Leak*. International Seminar on Intelligent System and Its Applications, (2015).
- [8] Khare, R., Sapra, A., Gupta, I., Kumar, V., Khare, A., Wig, T., dan Chawla, M., *Robotics*, Infinity Science, New Delhi, (2007).
- [9] Biagiotti, L., dan Melchiorri, C. *Trajectory Planning for Automatic machines and Robots*. Springer, Germany, (2008).
- [10] Faris, M. F., Triwiyatno, A., dan Setiawan, I. *Perancangan Arm Manipulator 4 DOF Dengan Menggunakan Pengendalian Cartesian Space-Trajectory Planning*. Transient, vol 1. No. 4, hal 151-158, (2012).

Biodata Penulis

Rendyansyah, pernah bekerja sebagai tenaga laboratorium kendali dan robotika pada Fakultas Ilmu Komputer UNSRI tahun 2011 sampai 2013. Penulis sekarang sebagai tenaga pengajar S1 Sistem Komputer Universitas Sriwijaya.

Aditya P.P Prasetyo, pernah bekerja sebagai tenaga laboratorium kendali dan robotika pada Fakultas Ilmu Komputer UNSRI tahun 2011 sampai 2013. Penulis sekarang sebagai tenaga pengajar S1 Sistem Komputer Universitas Sriwijaya.