

Trajectory Planning pada Case packer Delta Robot dengan Interpolasi Polinomial Kubik

Dwiky Fajri Syahbana^{1*}, Bramasta Triananda Putra¹, dan Imam Arifin²

¹Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

²Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

*Email: dwiky.syahbana@gmail.com

Received on 30-04-2025 | Revised on 02-05-2025 | Accepted on 25-06-2026

Abstrak—Mekanisme *pick and place* dinamis untuk pemindahan objek umumnya menggunakan robot paralel, seperti Delta robot 3-axis, yang membutuhkan koordinasi sistem yang teratur dan bekerja dengan cepat. Objek dijalankan melalui konveyor *infeed* secara kontinyu sehingga robot perlu bergerak dengan cepat dan tepat agar tidak ada produk yang terlewati. Dengan *trajectory planning* robot, resiko produk tidak terambil dapat diminimalkan. Dalam suatu kasus di sebuah industri, selama 10 menit hanya memiliki *pick rate* 82%, masih di bawah target standar, 90%. Dalam artikel ini dijelaskan implementasi metode interpolasi polinomial kubik dan *inverse kinematic* untuk membuat *trajectory planning*. Pengembangan metode baru dengan modifikasi gerakan robot lebih efisien dibandingkan dengan metode sebelumnya. Rangkaian gerakan robot awal dipangkas dari 5 menjadi 4 gerakan, menghasilkan lintasan yang lebih pendek dan mengurangi waktu bergerak yang tidak diperlukan. Dengan penggunaan metode tersebut didapatkan hasil simulasi *pick rate* produk sebesar 91,7% dan sudah melampaui target yang ditentukan.

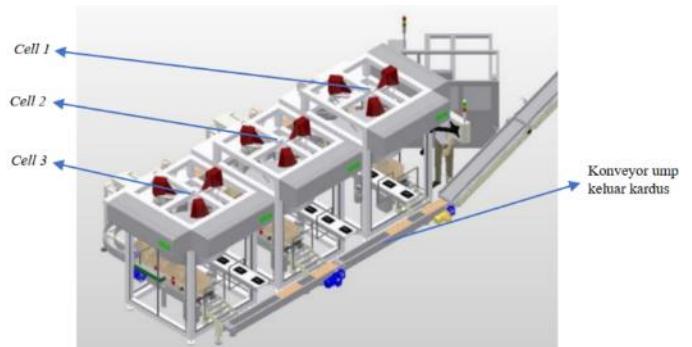
Keywords: Inverse Kinematics, Pick-and-Place, Polinomial Kubik, Trajectory Planning

Abstract—Dynamic pick and place mechanisms for object transfer generally use parallel robots, such as the 3-axis Delta robot, which requires regular system coordination and fast work. Objects are run through the infeed conveyor continuously so that the robot needs to move quickly and precisely so that no products are missed. With robot trajectory planning, the risk of unpicked products can be reduced to a minimum. A case in an industry shows that for 10 minutes it only had a pick rate of 82%, still below the standard, 90%. This article explains the application of the polynomial cubic interpolation method and inverse kinematics to create trajectory planning. The development of a new method with a modification of the robot's movement is more efficient than the previous method. The initial series of robot movements was cut from 5 to 4 movements, resulting in a shorter path and reducing unnecessary moving time. By using this method, the simulation results of the product pick rate were 91.7% and had exceeded the specified target.

Keywords: Inverse Kinematics, Pick-and-Place, Polynomial Cubic, Trajectory Planning

I. INTRODUCTION

PROSES pengemasan yang diterapkan oleh PT. X pada beberapa klien menggunakan mesin *case packer* robot Delta 3-axis. Proses ini terbagi menjadi beberapa segmen, yaitu transportasi produk dan karton, pemindahan produk ke dalam kardus (*pick and place*), dan penyegelan kardus. Pada sebuah kasus, *pick and place* adalah proses penting yang saat ini dilakukan dengan sistem konveyor *stop and go*, di mana konveyor berhenti saat produk diambil oleh robot. Sistem ini sangat tergantung pada robot dan orientasi robot tersebut, sehingga produk harus mengikuti koordinat tetap. Namun, sistem *pick and place* statis ini kurang efisien dari sisi jumlah produk yang diproses setiap waktunya. Kondisi saat ini, robot masih bergerak dengan menuju posisi *standby* sebelum mengambil produk, mengakibatkan adanya resiko produk terlewati atau tidak terambil. Hasil pengamatan dengan *sampling* menunjukkan 51 produk tidak terambil dari 284 produk dalam 10 menit, dengan *product pick rate* sebesar 82%.



Gambar 1. Desain mesin case packer keseluruhan

Sistem *pick and place* dinamis dapat diterapkan untuk meningkatkan efisiensi tersebut, dengan konveyor yang bergerak terus tanpa berhenti, dan robot bisa mengambil produk dengan cepat dari konveyor yang bergerak [1]. Namun mekanisme ini memiliki resiko bahwa robot mungkin tidak sempat mengambil produk yang berjalan di konveyor. Kegagalan ini bisa disebabkan oleh berbagai faktor, seperti kecepatan konveyor yang terlalu tinggi, kinerja robot yang lambat, prioritas pengambilan produk yang kurang tepat, dan

prediksi koordinat pengambilan yang tidak akurat. Hal tersebut dapat diatasi dengan penerapan *trajectory planning* [1], [2], salah satunya dengan interpolasi polinomial kubik untuk meningkatkan *pick rate* [3], [4]. Penggunaan algoritma polinomial ini telah dilakukan pada kasus robot *picker* untuk pemetikan buah [5] yang menghasilkan *trajectory* yang lancar. Selain itu, dalam penerapan di kasus lain, seperti *trajectory planning* pada manipulator yang memerlukan presisi tinggi [6] dan *mobile robot* [7], metode ini juga telah terbukti dapat menghasilkan gerakan robot yang mengikuti *trajectory* tanpa gerakan patah-patah.

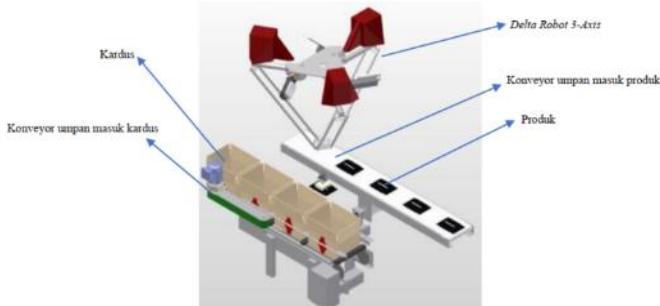
Dalam artikel ini, simulasi *trajectory planning* dilakukan menggunakan interpolasi polinomial kubik dengan *inverse kinematics* untuk meningkatkan *pick rate* produk yang ditargetkan lebih dari 90%. Simulasi dilakukan pada MATLAB untuk menguji rancangan *trajectory planning* sebelum diterapkan pada robot di lapangan. Simulasi tersebut akan menyesuaikan parameter robot dan lingkungan dengan mesin *case packer* yang digunakan.

II. METODE

A. Mesin Case-Packer dan Robot Delta 3-Axis

1) Sistem Case-Packer

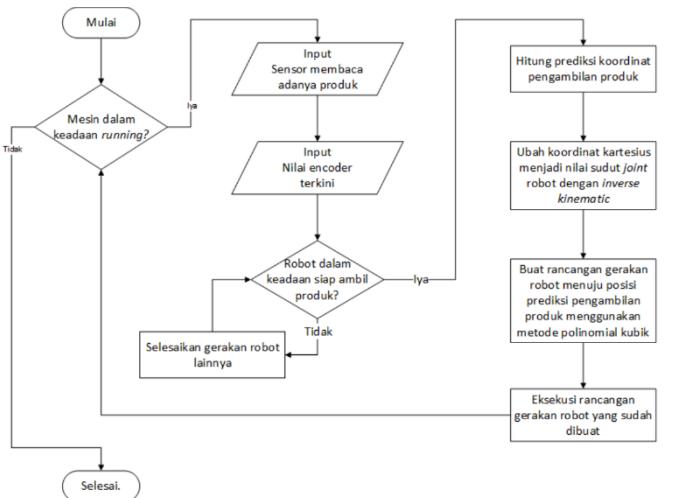
Mesin *Case packer* sebagai objek penelitian ini sudah didesain oleh PT. X seperti yang ditunjukkan pada Gambar 1. Mesin ini memiliki sejumlah 3 *cell* yang terpisah dan 1 jalur yang sama untuk keluaran kardus. Setiap *cell* (Gambar 2) memiliki robot Delta 3-axis, PLC, konveyor *infeed* produk, dan konveyor *infeed* kardus. Masing-masing *cell* bekerja secara otomatis dan paralel terhadap satu sama lain. Dalam pengembangan *trajectory planning* ini, hanya difokuskan satu *cell* dari mesin *case packer* tersebut. Konveyor *infeed* produk dan kardus tersebut digerakkan oleh motor induksi secara kontinyu dan konstan dengan kecepatan yang telah disesuaikan menjadi 420 mm/s sesuai kebutuhan klien. Delta robot yang digunakan memiliki *end effector* berupa *vacuum gripper* untuk mengambil produk sehingga produk dengan kemasan plastik akan dapat diambil dengan mudah.



Gambar 2. Desain salah satu cell pada mesin case-packer

Gambar 3 menjelaskan bahwa kinerja sistem pick and place dinamis yang dirancang ini dimulai dari pembacaan produk oleh sensor dan PLC. Selanjutnya, PLC akan menunggu robot menyelesaikan gerakan lainnya hingga memasuki *state* siap untuk mengambil produk. Gerakan lain ini meliputi gerakan robot menuju posisi *standby*, menuju posisi penempatan produk, dan menempatkan produk. Ketika robot sudah

memasuki *state* siap untuk mengambil produk, robot akan melakukan perhitungan prediksi pengambilan produk dalam koordinat kartesius. Untuk menghitung koordinat prediksi pengambilan produk ini, robot akan memproses waktu yang dibutuhkan untuk bergerak dari posisi awal menuju atas konveyor dan waktu tersebut selanjutnya akan digunakan dalam perhitungan dengan kecepatan konveyor untuk menghitung jarak perubahan koordinat produk. Titik koordinat prediksi ini kemudian diproses lebih lanjut dengan *inverse kinematic* sehingga didapatkan nilai sudut *joint* dari robot tersebut. PLC selanjutnya akan membuat rancangan *trajectory* dari nilai sudut *joint state* awal robot menuju nilai sudut *joint* yang ditentukan menggunakan interpolasi polinomial kubik. Metode ini digunakan agar gerakan dari robot tidak kaku dan tidak menghasilkan getaran. Setelah robot mencapai pada koordinat, selanjutnya robot akan mengambil produk dan melanjutkan proses *pick and place*.

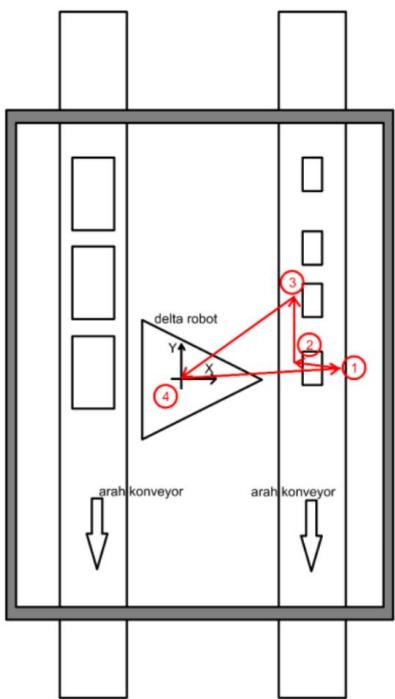


Gambar 3. Flowchart kerja mesin case-packer

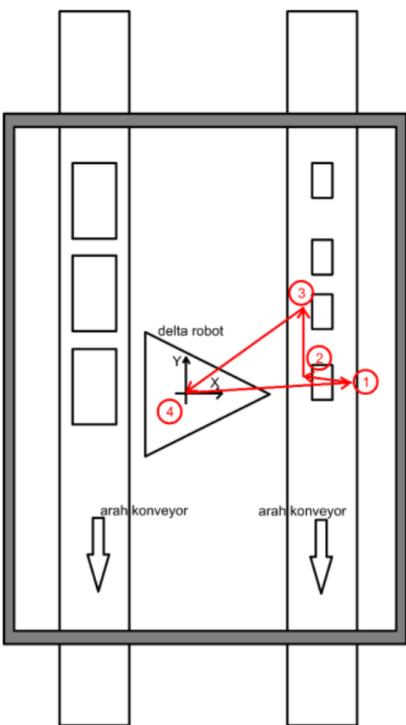
Seperti yang terlihat pada Gambar 4, terdapat 5 gerakan robot pada rangkaian gerakan robot referensi/ tersedia di lapangan. Rangkaian gerakan tersebut terdiri dari *standby 1*, *picking product 1*, *standby 2*, *picking product 2*, *standby pick*. Gerakan tersebut akan dijalankan secara urut dan berulang hingga mesin dalam *state* selain *running*.

Pada gerakan ke-1, *standby picking product 1*, robot akan bergerak memposisikan gripper kiri ke atas konveyor umpan masuk produk dan ke posisi paling depan dari arah masuknya konveyor tersebut. Langkah ini memastikan gripper kiri berada dalam posisi optimal untuk melakukan pengambilan produk pertama dengan hanya bergerak ke belakang mengikuti arah konveyor tanpa memperhatikan untuk bergerak ke kiri atau ke kanan lagi. Setelah itu, pada gerakan ke-3, *standby picking product 2*, robot akan kembali bergerak ke atas konveyor umpan masuk produk dan ke posisi paling depan dari arah masuknya konveyor tersebut. Namun pada fase ini robot akan memposisikan gripper bagian kanannya untuk persiapan pengambilan produk kedua. Sama seperti sebelumnya, langkah ini juga digunakan untuk memastikan gripper kanan berada dalam posisi optimal untuk pengambilan produk berikutnya.

dengan hanya bergerak ke belakang mengikuti arah konveyor.



Gambar 4. Ilustrasi gerakan robot referensi (awal)



Gambar 5. Ilustrasi gerakan robot modifikasi

Seperti yang terlihat pada gambar 4, terdapat 5 gerakan robot pada rangkaian gerakan robot referensi. Rangkaian gerakan tersebut terdiri dari standby picking product 1, picking product 1, standby picking product 2, picking product 2, standby pick. Gerakan tersebut akan dijalankan secara urut dan berulang

hingga mesin dalam state selain running. Rangkaian gerakan ini dibuat berdasarkan pengamatan pada robot di industri.

Pada gerakan ke-1, standby picking product 1, robot akan bergerak memposisikan gripper kiri ke atas konveyor umpan masuk produk dan ke posisi paling depan dari arah masuknya konveyor tersebut. Langkah ini memastikan gripper kiri berada dalam posisi optimal untuk melakukan pengambilan produk pertama dengan hanya bergerak ke belakang mengikuti arah konveyor tanpa memperhatikan untuk bergerak ke kiri atau ke kanan lagi.

Selanjutnya, pada gerakan ke-2, picking product 1, robot yang sudah di posisi standby picking product 1 sebelumnya akan langsung bergerak ke belakang dan ke bawah menuju produk. Pengambilan produk pertama ini menggunakan gripper kiri. Penggunaan gripper kiri untuk pengambilan produk pertama ini bertujuan agar selanjutnya robot tidak bergerak ke kanan lagi.

Setelah itu, pada gerakan ke-3, standby picking product 2, robot akan kembali bergerak ke atas konveyor umpan masuk produk dan ke posisi paling depan dari arah masuknya konveyor tersebut. Namun pada fase ini robot akan memposisikan gripper bagian kanannya untuk persiapan pengambilan produk kedua. Sama seperti sebelumnya, langkah ini juga digunakan untuk memastikan gripper kanan berada dalam posisi optimal untuk pengambilan produk berikutnya dengan hanya bergerak ke belakang mengikuti arah konveyor.

Pada gerakan ke-4, picking product 2, robot yang sudah berada di posisi siap mengambil produk dengan gripper kanannya akan bergerak ke belakang dan ke bawah menuju produk. Pengambilan produk kedua ini menggunakan gripper kanan sehingga pada saat ini, kedua gripper sudah dalam kondisi membawa produk.

Setelah gripper kiri dan kanan sudah membawa produk, kemudian terakhir, pada gerakan ke-5, standby pick, robot akan bergerak ke posisi awal seperti semula. Gerakan ini dirancang sebagai asumsi untuk gerakan placing produk, dikarenakan kurangnya parameter yang dapat digunakan untuk merancang gerakan tersebut, seperti kecepatan konveyor umpan masuk kardus, posisi penempatan kardus yang kosong, dan dimensi dari kardus tersebut.

Setelah robot kembali ke posisi semula, robot akan mengulangi gerakannya kembali mulai dari gerakan ke-1 hingga ke-5. Pengulangan ini dilakukan terus-menerus hingga mesin tidak dalam keadaan running. Ketika tidak ada produk yang berjalan pada konveyor umpan masuk produk, robot akan standby pada posisi gerakan ke-2 ataupun ke-4, tergantung dengan jumlah produk yang sudah dibawa oleh gripper.

Pada rangkaian gerakan robot modifikasi, peneliti menghilangkan gerakan standby picking product 1, sehingga hanya terdapat 4 gerakan robot. Selain itu, pada gerakan standby picking product 2, robot tidak akan bergerak menuju ke posisi paling depan dari arah masuknya konveyor. Rangkaian gerakan robot tersebut dirancang untuk meningkatkan efisiensi dari rangkaian gerakan robot sebelumnya atau referensi.

Seperti yang terlihat pada Gambar 5, rangkaian gerakan robot modifikasi memiliki lintasan yang relatif lebih pendek dibandingkan dengan rangkaian gerakan robot referensi. Hal ini

dapat meningkatkan efisiensi dari gerakan robot tersebut. Dengan menggunakan lintasan yang lebih pendek, total cycle time robot juga akan berkurang. Pengurangan waktu ini disebabkan terpangkasnya waktu kosong pada robot, sehingga waktu yang dibutuhkan robot dari gerakan ke-1 untuk bisa kembali ke gerakan ke-1 lagi menjadi lebih sedikit.

Pada gerakan ke-1, picking product 1, robot yang berada di posisi awal akan langsung bergerak menuju ke produk pertama. Sama seperti rangkaian gerakan robot sebelumnya, pengambilan produk pertama ini menggunakan gripper kiri. Penggunaan gripper kiri ini bertujuan agar selanjutnya robot tidak bergerak ke kanan lagi.

Setelah itu, pada gerakan ke-2, standby picking product 2, robot akan bergerak ke atas menghindari produk yang ada di konveyor dan ke kiri agar gripper kanan tepat berada di lintasan produk. Berbeda dengan rangkaian gerakan robot sebelumnya, lintasan gerakan ini relatif lebih pendek. Robot nantinya akan siap bergerak gerak maju atau mundur memposisikan gripper kanan mengikuti produk kedua yang akan diambil.

Selanjutnya pada gerakan ke-3, picking product 2, robot yang sudah berada di posisi siap mengambil produk dengan gripper kanannya akan bergerak ke belakang atau ke depan dan ke bawah menuju produk. Pengambilan produk kedua ini menggunakan gripper kanan sehingga pada saat ini, kedua gripper sudah dalam kondisi membawa produk.

Setelah gripper kiri dan kanan sudah membawa produk, kemudian terakhir, pada gerakan ke-4, standby pick, sama seperti rangkaian gerakan robot sebelumnya, robot akan bergerak ke posisi awal seperti semula. Gerakan ini dirancang sebagai asumsi untuk gerakan placing produk, dikarenakan kurangnya parameter yang dapat digunakan untuk merancang gerakan tersebut, seperti kecepatan konveyor umpan masuk kardus, posisi penempatan kardus yang kosong, dan dimensi dari kardus tersebut.

B. Inverse Kinematics & Trajectory Planning

I) Inverse Kinematics

Pada simulasi ini parameter robot dibuat berdasarkan parameter aktual dari robot Schneider Delta Robot P4 VRKP4S0FNC00000. Robot ini memiliki 3 lengan utama yang terhubung secara paralel pada plat end-effector. Pada masing-masing lengan utama memiliki joint sejumlah 3. Pemberian parameter pada robot ini dapat dilihat dari Gambar 5.

Pada perhitungan *inverse kinematic*, diasumsikan bahwa posisi dan orientasi end effector dari robot sudah ditentukan dan tujuannya adalah untuk mencari besarnya variabel sudut joint pada robot [8], [9], [10], [11]. Dari penentuan parameter kinematik robot aktual sebelumnya dapat dibuat struktur delta robot yang mirip dan disederhanakan seperti pada Gambar 6, sehingga mempermudah perhitungan inverse kinematic [12], [13]. Seperti yang terlihat pada Gambar 7, lengan utama kedua dan ketiga merupakan lengan utama paralel dari lengan utama pertama yang dirotasi 120° dan 240° berlawanan dengan arah jarum jam.

Untuk mencari nilai dari θ_1 , θ_2 dan θ_3 , serta menentukan koordinat pada masing-masing titik tersebut maka perlu ditentukan terlebih dahulu koordinat titik A₂, A₃, C₂, dan C₃.

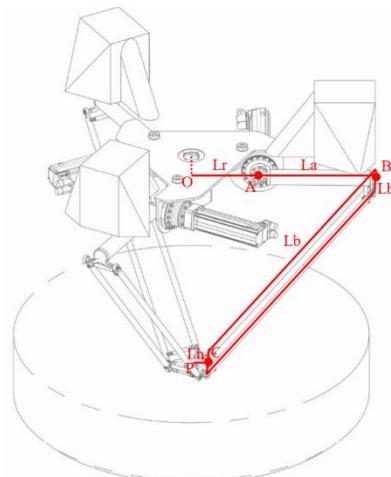
dan matrix rotasi M_n .

$$M_n = \begin{bmatrix} \cos(l_n) & -\sin(l_n) & 0 \\ \sin(l_n) & \cos(l_n) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Dengan, $n = 1, 2, 3$ dan $[l_1, l_2, l_3] = [0, 120, 240]$

$$\vec{A}_n = \vec{A}_n M_n \quad (2)$$

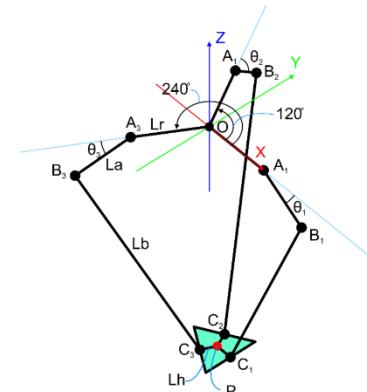
$$\vec{C}_n = \vec{P} + \vec{C}_n M_n \quad (3)$$



Gambar 6. Schematics drawing delta robot 3-axis

Dengan asumsi titik P diketahui, A₁ dan C₁ dapat ditentukan dengan matrix koordinat:

$$\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \vec{A}_1 = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} L_r \\ 0 \\ a_z \end{bmatrix}, \vec{C}_1 = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \begin{bmatrix} x + L_h \\ c_y \\ c_z \end{bmatrix} \quad (4)$$



Gambar 7. Struktur parameter kinematika delta robot 3-axis

Setelah A₂, A₃, C₂, dan C₃ dapat ditentukan, selanjutnya yaitu merotaskannya kembali ke sumbu x. Hal ini dilakukan agar perhitungan B₂ dan B₃ lebih mudah untuk dikerjakan. Untuk merotasikan kembali ke sumbu, diperlukan matrix yang invers dari M_n atau dapat disimbolkan IM_n.

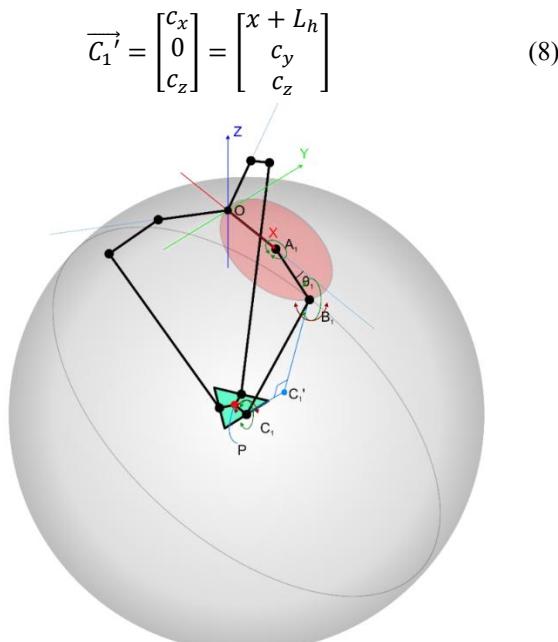
$$IM_n = \begin{bmatrix} \cos(l_n) & \sin(l_n) & 0 \\ -\sin(l_n) & \cos(l_n) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Dengan, $n = 1, 2, 3$ dan $[l_1, l_2, l_3] = [0, 120, 240]$

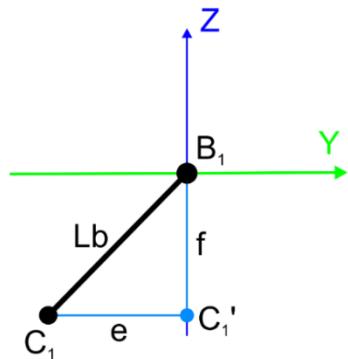
$$\overrightarrow{A_n''} = \overrightarrow{A_n} IM_n \quad (6)$$

$$\overrightarrow{C_n''} = \vec{P} + \overrightarrow{C_n} IM_n \quad (7)$$

Setelah mendapatkan koordinat A_n dan C_n , perhitungan selanjutnya dapat difokuskan menjadi seperti perhitungan pada lengan utama pertama seperti pada Gambar 8. Terdapat titik baru yaitu C_1' yang merupakan proyeksi titik C_1 pada sumbu x.



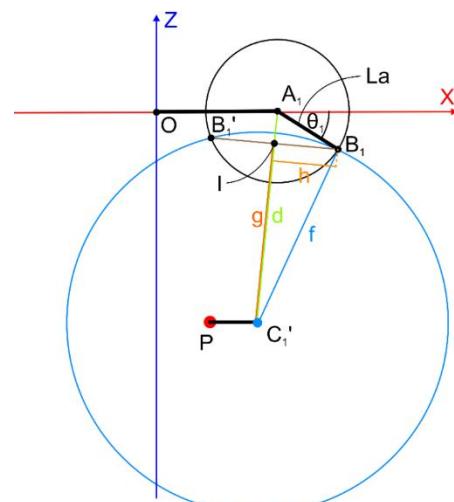
Gambar 8. Model geometri keseluruhan delta robot 3-axis



Gambar 9. Model geometri delta robot 3-axis pada bidang Y-Z

Selanjutnya jari-jari bidang bola C_1' dapat ditentukan berdasarkan Gambar 9, sehingga dapat dituliskan pada persamaan (9).

$$f = \sqrt{Lb^2 - |e|^2} \quad (9)$$



Gambar 10. Model geometri delta robot 3-axis pada bidang X-Z

Setelah jari-jari bidang bola C_1' diketahui, langkah selanjutnya yaitu mencari titik perpotongan antara lingkaran berpusat A_1 dengan lingkaran berpusat C_1' . Seperti yang terlihat pada Gambar 10, model geometri Delta Robot dibuat pada bidang X dan Z sehingga,

$$d = \sqrt{(a_x - c_x)^2 + (a_z - c_z)^2} \quad (10)$$

$$g = \frac{(f^2 - La^2 + d^2)}{2d} \quad (11)$$

$$\vec{i} = \begin{bmatrix} i_x \\ i_z \end{bmatrix} \quad (12)$$

$$i_x = a_x + g \frac{(c_x - a_x)}{d} \quad (13)$$

$$i_z = a_z + g \frac{(c_z - a_z)}{d} \quad (14)$$

$$h = \sqrt{f^2 - g^2} \quad (15)$$

$$\overrightarrow{B_1} = \begin{bmatrix} b_x \\ 0 \\ b_z \end{bmatrix}, \overrightarrow{B_1'} = \begin{bmatrix} b_x' \\ 0 \\ b_z' \end{bmatrix} \quad (16)$$

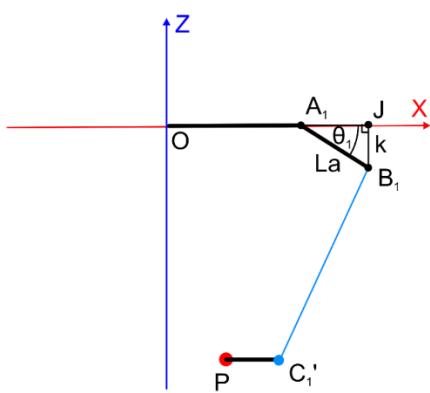
$$b_x = i_x + h \frac{(c_x - a_x)}{d} \quad (17)$$

$$b_z = i_z - h \frac{(c_z - a_z)}{d} \quad (18)$$

$$b_x' = i_x - h \frac{(c_x - a_x)}{d} \quad (19)$$

$$b_z' = i_z + h \frac{(c_z - a_z)}{d} \quad (20)$$

Setelah didapatkan koordinat dari titik B_1 dan B_1' pada persamaan (16), akan dipilih satu titik perpotongan antara B_1 dan B_1' yang memiliki nilai paling besar pada sumbu x. Satu titik perpotongan yang terpilih akan menjadi titik B_1 dan dapat dilanjutkan dengan perhitungan θ_1 seperti yang terlihat pada Gambar 11, sehingga θ_1 dapat dihitung dengan trigonometri seperti persamaan (21).



Gambar 11. Model Geometri Delta Robot 3-Axis pada bidang X-Z

$$\begin{aligned}\theta_1 &= \sin^{-1}\left(\frac{k}{La}\right) \\ \theta_1 &= \sin^{-1}\left(\frac{b_z}{La}\right)\end{aligned}\quad (21)$$

2) Trajectory Planing

Pergerakan garis lurus pada robot adalah pergerakan paling umum yang digunakan di industri. Namun, pergerakan garis lurus ini sering kali didapatkan dari menentukan perpindahan sudut joint dengan waktu diskrit pada laju waktu konstan. Kecepatan dan percepatan dari setiap perpindahan sudut joint dapat dihitung menggunakan turunan waktu dari persamaan perpindahan sudut joint.

Pada penelitian ini, persamaan perpindahan sudut *joint* menggunakan metode polinomial kubik. Metode polinomial kubik digunakan untuk membuat persamaan dari sudut joint awal hingga ke sudut joint akhir dengan waktu tertentu[14], [15]. Metode ini merupakan polinomial tingkat tiga yang dapat dituliskan pada persamaan (22) dan diturunkan menjadi persamaan kecepatan (23) dan percepatan (24).

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (22)$$

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (23)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \quad (24)$$

Untuk pergerakan yang halus, polinomial kubik memiliki 4 batasan. Persamaan batasan-batasan tersebut dapat dituliskan seperti pada persamaan (25) s.d. (28).

$$\theta(t_0) = \theta_0 \quad (25)$$

$$\theta(t_f) = \theta_f \quad (26)$$

$$\dot{\theta}(t_0) = \dot{\theta}_0 = 0 \quad (27)$$

$$\dot{\theta}(t_f) = \dot{\theta}_f = 0 \quad (28)$$

Dengan memasukkan parameter persamaan (25), (26), (27), dan (28) pada persamaan (22), (23), dan (24) didapatkan nilai konstanta *a* menjadi persamaan (29) s.d. (32).

$$a_0 = \theta_0 \quad (29)$$

$$a_1 = \dot{\theta}_0 = 0 \quad (30)$$

$$a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) \quad (31)$$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) \quad (32)$$

Sehingga didapatkan persamaan polinomial kubik dari perpindahan sudut *joint*, persamaan kecepatan, dan persamaan percepatan sudut *joint* seperti persamaan (33) s.d. (35).

$$\theta(t) = \theta_0 + \frac{3}{t_f^2}(\theta_f - \theta_0)t^2 - \frac{2}{t_f^3}(\theta_f - \theta_0)t^3 \quad (33)$$

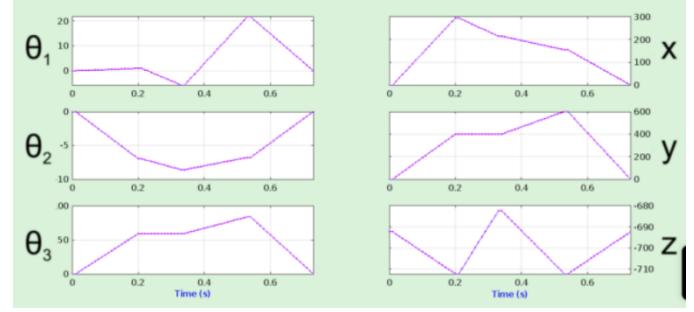
$$\dot{\theta}(t) = \frac{6}{t_f^2}(\theta_f - \theta_0)t - \frac{6}{t_f^3}(\theta_f - \theta_0)t^2 \quad (34)$$

$$\ddot{\theta}(t) = \frac{6}{t_f^2}(\theta_f - \theta_0) - \frac{12}{t_f^3}(\theta_f - \theta_0)t \quad (35)$$

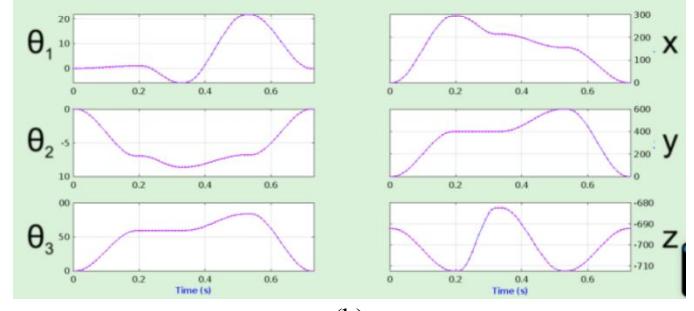
III. HASIL DAN PEMBAHASAN

A. Pengujian 1 Cycle

Pada percobaan simulasi ini akan dianalisa penerapan dari trajectory planning yang sudah diimplementasikan pada simulasi. Selain pengujian trajectory planning yang sudah dirancang, pada pengujian ini juga akan menguji trajectory planning tanpa menggunakan polinomial kubik. Pengujian pertama ini akan dilakukan sebanyak 5 kali iterasi untuk mengetahui konsistensi dari masing-masing trajectory planning. Data yang diperoleh juga akan digunakan untuk mengevaluasi apakah metode polinomial kubik memberikan keuntungan tambahan dalam hal kecepatan maupun presisi gerakan.



(a)



(b)

Gambar 12. Trajectory hasil pengujian (a) Tanpa polinomial kubik (b) Menggunakan polinomial kubik

Pada pengujian 1 cycle time dengan membandingkan antara trajectory yang memiliki kecepatan konstan dan trajectory polinomial kubik yang memiliki kecepatan berubah, didapatkan hasil berupa gerakan robot yang lebih halus. Gerakan yang lebih halus ini dibuktikan dengan Gambar 12 yang menjelaskan tentang hentakan sudut pada joint θ_1 . Grafik menunjukkan bahwa pada trajectory tanpa polinomial kubik, terdapat fluktuasi yang signifikan pada hentakan, dengan nilai hentakan yang lebih tinggi dan lebih tajam. Sedangkan, trajectory dengan polinomial kubik, fluktuasi pada hentakan lebih rendah dan lebih halus dibandingkan dengan trajectory tanpa polinomial kubik.

TABEL I
HASIL PENGUJIAN PRODUCT PICK RATE

Gerakan Robot	Pengujian ke-					
	1	2	3	4	5	6
Picking product 1 (ms)	300	200	200	200	200	200
Standby picking 2(ms)	200	200	150	150	150	130
Picking product 2 (ms)	300	300	300	200	200	200
Standby picking (ms)	300	300	300	300	200	200
Cycle time (ms)	1100	1000	950	850	750	730
Average product pick rate (%)	77,63	79,4	81,92	85,26	89,39	91,7

B. Pengujian Parameter Cycle Time Optimal

Selanjutnya yaitu uji pengaruh perubahan parameter *cycle time* dari setiap gerakan robot pada *product pick rate* untuk 284 produk. Pada percobaan simulasi ini akan dianalisa pengaruh perubahan parameter *cycle time* dari setiap gerakan robot pada *product pick rate* robot. Parameter ini akan disesuaikan ulang hingga dapat mencapai target *product pick rate* yaitu lebih dari 90%. Setiap perubahan parameter juga akan dilakukan pengujian ulang sebanyak 5 kali untuk mengetahui kestabilan *product pick rate*. Data dari setiap perubahan parameter akan dianalisis untuk mengevaluasi apakah perubahan *cycle time* berdampak positif terhadap peningkatan *product pick rate*.

Seperti yang terlihat pada tabel 1, dari perubahan parameter pada pengujian pertama hingga terakhir, perubahan paling signifikan dapat dilihat pada pengujian ke 5. Pengujian ke 5 mengubah parameter *time* dari gerakan standby picking sebanyak 100 ms. Dengan perubahan ini peningkatan presentase *product pick rate* didapatkan hingga sebesar 4,13%. Perubahan yang besar ini disebabkan karena gerakan ini menjadi salah satu gerakan dengan trajectory terpanjang. Dengan mengurangi *cycle time* pada gerakan dengan trajectory terjauh ini dapat mempercepat proses untuk menuju ke gerakan berikutnya, sehingga robot memiliki waktu lebih untuk mengambil produk selanjutnya.

IV. KESIMPULAN

Pengembangan *trajectory planning* robot Delta 3-axis dengan rangkaian gerakan robot modifikasi menghasilkan *trajectory* yang lebih efisien dibandingkan *rancangan trajectory planning* rangkaian gerakan robot referensi. Rangkaian gerakan robot referensi dipangkas dari 5 menjadi 4 gerakan pada rangkaian gerakan robot modifikasi. Pada penelitian ini disimulasikan menggunakan 4 gerakan robot dari rangkaian gerakan robot modifikasi yaitu *picking product 1* dengan *cycle time* 200 ms, *standby 2* dengan *cycle time* 130 ms, *picking product 2* dengan *cycle time* 200 ms, *standby picking 3* dengan *cycle time* 200 ms, dan didapatkan hasil *product pick rate* sebesar 91,7%. Dengan *product pick rate* seperti ini, dapat dikatakan pengembangan dengan simulasi ini sudah memenuhi target yaitu lebih dari 90%. Hasil simulasi ini perlu diimplementasikan secara langsung pada robot di mesin *case-packer* secara nyata.

REFERENCES

- [1] Q. Zu, Q. Liu, and J. Wu, "Dynamic Pick and Place Trajectory of Delta Parallel Manipulator," in *Human Centered Computing*, Q. Zu, Y. Tang, and V. Mladenovic, Eds., 2021, pp. 1–11. doi: 10.1007/978-3-030-70626-5_1.
- [2] D. Zhu, Y. He, X. Yu, and F. Li, "Trajectory Smoothing Planning of Delta Parallel Robot Combining Cartesian and Joint Space," *Mathematics*, vol. 11, no. 21, Nov. 2023, doi: 10.3390/math11214509.
- [3] T. Li, Y. Zhang, and J. Zhou, "Trajectory planning of high precision collaborative robots," *CMES - Computer Modeling in Engineering and Sciences*, vol. 118, no. 3, pp. 583–598, 2019, doi: 10.31614/cmes.2018.04891.
- [4] Y. Li, J. Wang, and Y. Ji, "Function Analysis of Industrial Robot under Cubic Polynomial Interpolation in Animation Simulation Environment," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 4, p. 105, 2020, doi: 10.9781/ijimai.2020.11.012.
- [5] H. Wang, Q. Zhao, H. Li, and R. Zhao, "Polynomial-based smooth trajectory planning for fruit-picking robot manipulator," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 112–122, Mar. 2022, doi: 10.1016/j.inpa.2021.08.001.
- [6] N. Minh Tuan, T.-T. Vuong, T.-P. Nguyen, and M.-T. Nguyen, "Smooth and Time Optimization Trajectory Planning For Robots Using Polynomial Interpolation", [Online]. Available: <https://www.researchgate.net/publication/384839338>
- [7] T. Prima Satya, J. Hendry, A. Putra Yudhananta, and Z. F. Bagus Meliawan, "Cheap and Fast Implementation of Linear and Cubic Piecewise Interpolation for Robot Path Smoothing on Arduino Uno Board," *JuLIET*, vol. 3, no. 2, 2022.
- [8] S. Kang and W. Chou, "Kinematic Analysis, Simulation and Manipulating of a 5-DOF Robotic Manipulator for Service Robot," in *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE, Aug. 2019, pp. 643–649. doi: 10.1109/ICMA.2019.8816413.
- [9] R. Ghaedrahmati and C. Gosselin, "Kinematic Analysis of a New 3-DOF Parallel Wrist-Gripper Assembly with a Large Singularity-Free Workspace," *Actuators*, vol. 12, no. 11, Nov. 2023, doi: 10.3390/act12110421.
- [10] A. H. Dastjerdi, M. M. Sheikhi, and M. T. Masouleh, "A complete analytical solution for the dimensional synthesis of 3-DOF delta parallel robot for a prescribed workspace," *Mech Mach Theory*, vol. 153, Nov. 2020, doi: 10.1016/j.mechmachtheory.2020.103991.
- [11] S. Pakzad, S. Akhbari, and M. Mahboubkhah, "Kinematic and dynamic analyses of a novel 4-DOF parallel mechanism," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 41, no. 12, Dec. 2019, doi: 10.1007/s40430-019-2058-3.
- [12] G. Li, J. Song, W. Zhang, and Q. Wu, "Trajectory Planning of 6-DOF Manipulator Based on Joint Space," in *2024 7th International Conference on Advanced Algorithms and Control Engineering, ICAACE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 1369–1372. doi: 10.1109/ICAACE61206.2024.10548112.
- [13] A. Gholami, T. Homayouni, R. Ehsani, and J. Q. Sun, "Inverse kinematic control of a delta robot using neural networks in real-time," *Robotics*, vol. 10, no. 4, Dec. 2021, doi: 10.3390/robotics10040115.
- [14] H. Afrisal, A. L. Hakim, M. J. Shiddiq, and I. Setiawan, "Position Control and Trajectory Planning of 3-DOF Arm Manipulator for Test Tube Handling," in *2019 International Biomedical Instrumentation and Technology Conference (IBITEC)*, IEEE, Oct. 2019, pp. 52–57. doi: 10.1109/IBITEC46597.2019.9091714.
- [15] L. Shijie, "Research Progress on Trajectory Planning of Industrial Robots," *Current Journal of Applied Science and Technology*, vol. 42, no. 2, pp. 25–36, Feb. 2023, doi: 10.9734/cjast/2023/v42i24052.