



CHESSE ROBOT ARM 4 DEGREE OF FREEDOM TRAJECTORY PLANNING WITH INVERSE KINEMATICS METHOD

Ahmad Farih Qolyubie Rosief¹, Indrazno Siradjuddin², Hari Kurnia Safitri³

e-mail: farihqolyubie@gmail.com, indrazno@polinema.ac.id, hari.kurnia@polinema.ac.id

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Malang, Jalan Soekarno Hatta No.9 Malang, Indonesia

Abstract- Robots have become a preferred choice for assisting humans in overcoming issues related to precision, safety, flexibility, and repetitive tasks. Robot arm are among the types of robots that can aid human tasks. This research aims to develop a 4 degree of freedom robot arm based on Arduino, designed to move chess pieces with precision using simple programming language (C++). The study includes the mechanical design and implementation of electronic components to control the robot arm. Additionally, the inverse kinematics algorithm is utilized to calculate the required joint angles to achieve the desired positions and orientations of the robot arm, enabling it to move chess pieces accurately. The 4 degree of freedom robot arm successfully moved chess pieces in five test runs with a 94% success rate, demonstrating a reliable performance in completing assigned tasks. The built 4 degree of freedom robot arm exhibited an error rate of 2,22% along the x-axis, 1,35% along the y-axis, and 0,99% along the z-axis. These results indicate that the developed 4 degree of freedom robot arm can move chess pieces precisely according to the planned movements.

Keywords: Robot arm, Arduino, Inverse Kinematics C++

Abstrak— Robot menjadi pilihan untuk membantu pekerjaan manusia mengatasi masalah kepresisian, keamanan, fleksibilitas dan pekerjaan yang berulang-ulang. Robot arm menjadi salah satu jenis robot yang dapat membantu pekerjaan manusia. Penelitian ini bertujuan untuk mengembangkan robot arm dengan 4 degree of freedom berbasis arduino, yang dirancang untuk memindahkan bidak catur dengan presisi menggunakan bahasa pemrograman sederhana (C++). Penelitian ini mencakup perancangan dan implementasi mekanik serta perangkat elektronik untuk mengendalikan robot arm. Selain itu, algoritma inverse kinematics digunakan untuk menghitung sudut joint yang diperlukan untuk mencapai posisi dan orientasi yang diinginkan dari robot arm agar dapat memindahkan bidak catur dengan tepat. Robot arm 4 degree of freedom berhasil memindahkan bidak catur dalam pengujian sebanyak 5 kali dengan tingkat keberhasilan 94%. Hal ini menunjukkan kinerja yang baik dalam menyelesaikan tugas yang diberikan. Robot arm 4 degree of freedom yang dibangun memiliki tingkat error pada sumbu x sebesar 2,22%, pada sumbu y sebesar 1,35% mm, dan pada sumbu z sebesar 0,99%. Hasil penelitian ini menunjukkan bahwa robot arm 4 degree of freedom yang dikembangkan mampu memindahkan bidak catur secara presisi sesuai dengan pergerakan yang telah direncanakan.

Kata kunci: : Robot Lengan, Arduino Inverse Kinematik C++

1. PENDAHULUAN



Robot merupakan teknologi yang difungsikan untuk membantu pekerjaan manusia tak terkecuali di bidang industri. Robot dapat membantu menyelesaikan pekerjaan manusia dalam berbagai hal, termasuk pada pekerjaan dalam bidang industri. Robot yang dapat kita jumpai pada bidang industri yaitu robot lengan pemindah barang, robot lengan pengecoran logam, robot lengan pengelasan dan robot lengan pengecatan (*painting*). Robot lengan dapat digunakan untuk mengangkat, memindahkan dan memanipulasi benda kerja untuk meringankan kerja manusia. Inovasi perkembangan robot dalam dunia industri yaitu, robot lengan pemindah barang yang dikendalikan melalui *smartphone* android [1]. Robot lengan digunakan dalam bidang industri karena dapat membantu pekerjaan manusia terutama pekerjaan yang membutuhkan ketelitian, pekerjaan yang berulang dan jumlah produksi yang banyak. Penelitian-penelitian terus dilakukan dengan tujuan untuk memperbaiki dan meningkatkan kinerja suatu sistem produksi atau media produksi sebagai dampak positif, ilmu pengetahuan dan teknologi [2].

Salah satu pengembangan Robot *arm* adalah Robot *Arm* yang dapat difungsikan dalam permainan catur atau biasa disebut *Chess Robot arm*. Robot ini dapat menjadi lawan dari pemain catur atau memindahkan bidak catur sesuai dengan perintah yang diberikan. Pada umumnya, proses kendali Robot *arm* dalam permainan catur diprogram dengan bahasa pemrograman yang cukup rumit dan memerlukan mikrokontroler yang mahal, karena Robot *arm* diprogram menggunakan bahasa program *python* yang dikendalikan oleh mikrokontroler *raspberry pi* [3]. Selain itu Robot *arm* juga menggunakan *image processing* dalam membaca posisi bidak catur.

Berdasarkan permasalahan tersebut sangat diperlukan pengembangan Robot *arm* yang dapat memindahkan bidak catur dengan menggunakan bahasa pemrograman yang sederhana yaitu (C++) berbasis arduino. Untuk itu peneliti membuat robot lengan yang dapat dikendalikan dengan mikrokontroler Arduino. Robot lengan ini didesain memiliki 4 DOF (*Degree of Freedom*) dengan seluruh *joint* adalah revolute yang menggunakan motor stepper Nema 23 dan Nema 17. *Degree of Freedom* disingkat DOF yang berarti derajat kebebasan, merupakan penghubung pada lengan robot, yang dapat berputar. DOF digunakan untuk mengetahui cara robot bergerak, jumlah motor lengan robot yang digunakan dan tingkat kerumitan algoritma [1][4]. Kemudian untuk menentukan posisi bidak catur yang ingin diambil oleh *gripper* bisa dicari dengan *input* berupa titik koordinat dari posisi bidak catur, kemudian diolah menggunakan metode *inverse kinematics* yang akan menghasilkan *output* berupa besaran sudut yang dibutuhkan pada masing-masing sendi. Pada penelitian ini, diharapkan robot lengan dapat melakukan pergerakan yang sesuai untuk mengambil maupun meletakkan bidak catur pada posisi target.

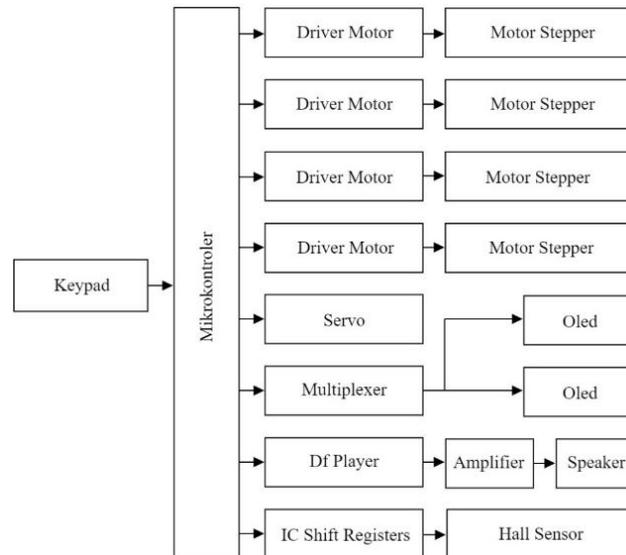
2. METODE PENELITIAN

Metode penelitian yang digunakan yaitu metode experimental. Metode Experimental adalah prosedur penelitian yang dilakukan melalui pengujian untuk mengendalikan posisi lengan robot agar dapat mengangkat dan memindahkan bidak catur.

2.1 Perancangan Sistem

Diagram blok sistem adalah sebuah diagram yang menjelaskan alur setiap komponen yang berperan sebagai *input*, proses, dan juga *output*. Diagram blok sistem yang digunakan pada penelitian ini ditunjukkan pada gambar 1. Diagram blok sistem pada gambar 1 menjelaskan tentang rancangan alur dari *hardware* yang dibuat. Pada Gambar 1, terdapat mikrokontroller yang berfungsi untuk mengontrol pergerakan robot dengan mengirimkan perintah ke driver motor. Kemudian *input push button* yang digunakan untuk memilih sebuah koordinat bidak catur pada layar LCD yang sudah diprogramkan sebelumnya. Dari *input push button* dikirim ke mikrokontroller arduino menggunakan kabel yang dihubungkan pada pin digital arduino yang sudah diintegrasikan. Pada bagian *output* dari mikrokontroler terdapat driver motor yang berfungsi untuk mengubah sinyal PWM menjadi tegangan DC sebagai suplai motor stepper. Selain itu, driver motor juga untuk menggerakkan motor stepper sesuai dengan perintah yang diberikan. Driver motor juga bisa digunakan untuk mengontrol kecepatan dan arah motor. Motor stepper digunakan sebagai penggerak utama dari robot arm dan servo

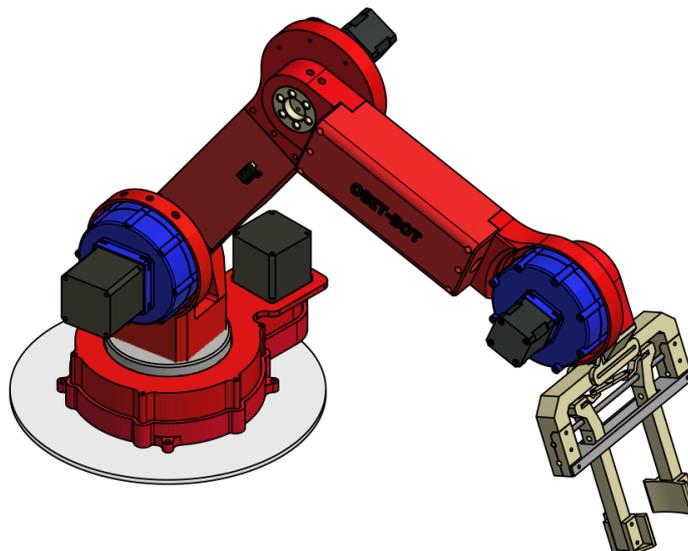
digunakan untuk gripper sebagai pencapit dari bidak catur. Pada semua blok diagram tersebut semua komponen terhubung ke *power supply* sebagai sumber daya.



Gambar 1: Diagram Blok Sistem

2.2 Perancangan Mekanik

Perancangan mekanik meliputi desain dan konstruksi komponen robot arm. Proses perancangan mekanik bertujuan untuk menciptakan sebuah *chess robot arm* dengan 4 Derajat Kebebasan (DOF) berbasis Arduino Mega 2560 yang dapat memindahkan bidak catur.



Gambar 2: Desain Keseluruhan Mekanik *Robot Arm*

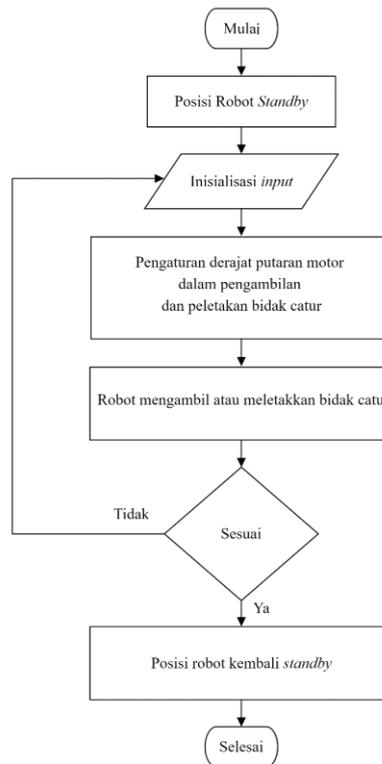
2.3 Perancangan Elektronik

- a. Sumber Tegangan
 - *Power Supply*
 - Voltage* : 12 Volt
 - Current* : 30 Ampere
 - *Step Down*
 - Voltage* : 5 Volt
 - Current* : 3 Ampere
- b. Kontroler : Arduino Mega
- c. Metode : *Invers Kinematics*



- d. Aktuator : Motor Stepper Nema 23, Motor Stepper Nema 17 Motor dan Servo EMAX ES08MA
- e. Sensor : *Hall* Sensor 7411C
- f. Tampilan : Oled
- g. *Input Command* : Keypad

2.4 Perancangan Kontrol *Robot Arm*



Gambar 3: Flowchart Perancangan Kontrol *Robot Arm*

2.5 Perancangan Kalibrasi Motor Driver DRV8825

Perancangan kalibrasi Motor Driver DRV8825 bertujuan untuk memastikan arus keluaran dari modul motor driver sudah sesuai dengan spesifikasi motor yang dipakai, sehingga motor driver berfungsi dengan akurat dan konsisten sesuai kebutuhan sistem yang dirancang. Langkah-langkah kalibrasi bisa dilakukan dengan cara memutar trimpot pada modul motor driver untuk menyesuaikan arus keluaran terhadap motor yang digunakan. Adapun rumus untuk mengetahui arus keluaran yang sesuai dengan mencari tegangan referensi menggunakan persamaan rumus dibawah ini.

$$V_r = I_m \times 5 \times R_s$$

$$V_r = I_m \times 5 \times 0,1$$

$$V_r = I_m \times 0,5$$

Dimana,

V_r = Tegangan Referensi

I_m = Spesifikasi Arus Motor

R_s = Resistor yang dipakai dimodul driver

Setelah menemukan tegangan referensi yang digunakan, selanjutnya dapat dicoba dengan mengukur modul driver motor menggunakan multimeter pada kutub + di trimpot modul dan kutub – disumber tegangan negatif 12V modul.

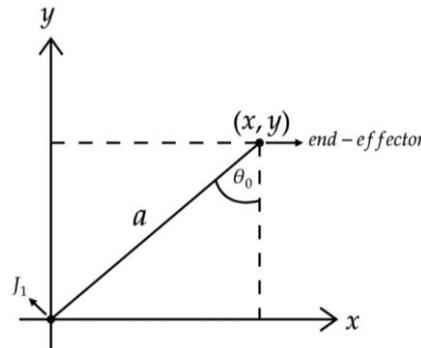
Perancangan *Inverse Kinematics*

Perancangan melibatkan penggunaan metode *inverse kinematics* untuk mengendalikan gerakan *robot arm* 4-DOF. Metode *inverse kinematics* digunakan untuk menghitung setiap *joint robot arm*



berdasarkan posisi dan orientasi ujung *end-effector* yang diinginkan. Metode *inverse kinematics* memungkinkan untuk mengendalikan gerakan *robot arm* dengan memberikan posisi dan orientasi yang diinginkan, kemudian menghitung sudut-sudut yang diperlukan oleh setiap *joint* untuk mencapai tujuan tersebut. Perhitungan yang digunakan pada perancangan *inverse kinematics* seperti berikut:

a. *Robot arm* tampak atas



Gambar 4: Visualisasi 2D Untuk Mencari θ_0

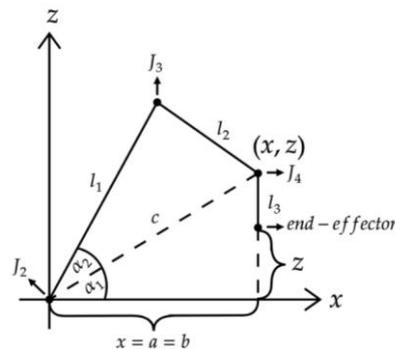
Pada Gambar 4 diketahui nilai $x = 7,5 \text{ cm}$, nilai $y = 17,2 \text{ cm}$ dan nilai a yang belum diketahui, nilai a dapat ditentukan menggunakan rumus *pythagoras* pada persamaan (2) dengan perhitungan seperti berikut:

$$\begin{aligned}
 a &= \sqrt{x^2 + y^2} & (2) \\
 a &= \sqrt{(7,5)^2 + (17,2)^2} \\
 a &= \sqrt{156,25 + 1383,84} \\
 a &= \sqrt{1540,09} \\
 a &= 18,76 \text{ cm}
 \end{aligned}$$

Nilai dari θ_0 dapat ditentukan menggunakan rumus pada persamaan (3).

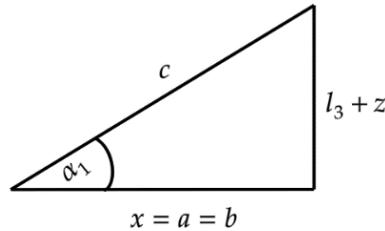
$$\begin{aligned}
 \theta_0 &= \tan^{-1}\left(\frac{x}{y}\right) & (3) \\
 \theta_0 &= \tan^{-1}\left(\frac{7,5}{17,2}\right) \\
 \theta_0 &= \tan^{-1}(0,336) \\
 \theta_0 &= 23,56^\circ
 \end{aligned}$$

b. *Robot arm* tampak samping



Gambar 5: Visualisasi 2D Untuk Mencari θ_1

Pada Gambar 5 terdapat dua bagian visualisasi 2D untuk mencari θ_1 . Sudut pada θ_1 dibagi menjadi dua yaitu α_1 dan α_2 . Dari gambar tersebut diketahui nilai $x = 18,76 \text{ cm}$, $z = 0 \text{ cm}$, $l_1 = 26,5 \text{ cm}$, $l_2 = 26,5 \text{ cm}$, $l_3 = 9,1 \text{ cm}$. Nilai $z = 0 \text{ cm}$ agar ujung gripper sampai pada papan catur agar bisa menjangkau bisak catur. Visualisasi 2D untuk mencari nilai sudut α_1 ditunjukkan pada Gambar 6.



Gambar 6: Visualisasi 2D Untuk Mencari α_1

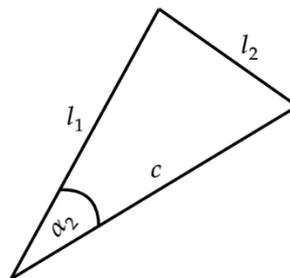
Dari Gambar 6 sudut α_1 dapat ditentukan menggunakan persamaan (4). Nilai sudut α_1 dapat dihitung ketika nilai b sudah diketahui. Nilai b diketahui dari $x = a = b$, jadi nilai $b = 18,76 \text{ cm}$. Sehingga perhitungan sebagai berikut:

$$\begin{aligned} \alpha_1 &= \tan^{-1} \left(\frac{l_3 + z}{b} \right) & (4) \\ \alpha_1 &= \tan^{-1} \left(\frac{9,1 + 0}{18,76} \right) \\ \alpha_1 &= \tan^{-1} \left(\frac{9,1}{18,76} \right) \\ \alpha_1 &= \tan^{-1}(0,23190622) \\ \alpha_1 &= 25,88^\circ \end{aligned}$$

Setelah mengetahui nilai sudut α_1 maka langkah selanjutnya yaitu mencari terlebih dahulu nilai c dengan menggunakan rumus *pythagoras* seperti pada persamaan (5).

$$\begin{aligned} c &= \sqrt{b^2 + (l_3 + z)^2} & (5) \\ c &= \sqrt{(18,76)^2 + (9,1)^2} \\ c &= \sqrt{1539,7776 + 82,81} \\ c &= \sqrt{1622,5876} \\ c &= 20,85 \text{ cm} \end{aligned}$$

Visualisasi 2D untuk menentukan nilai sudut α_2 terdapat pada gambar 7.



Gambar 7: Visualisasi 2D Untuk Mencari α_2

Untuk menghitung sudut α_2 dapat menggunakan rumus cos segitiga tidak beraturan seperti pada persamaan (6).

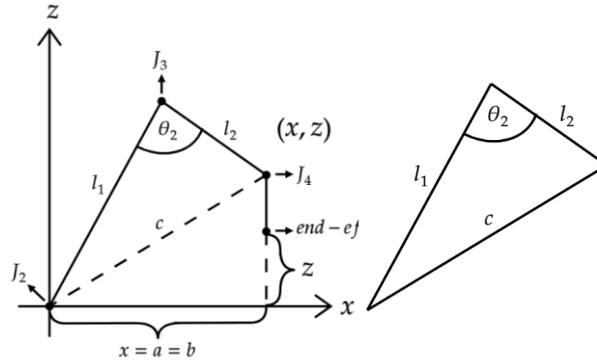
$$\begin{aligned} \alpha_2 &= \cos^{-1} \left(\frac{l_1^2 + c^2 - l_2^2}{2 \cdot l_1 \cdot c} \right) & (6) \\ \alpha_2 &= \cos^{-1} \left(\frac{(26,5)^2 + (20,85)^2 - (26,5)^2}{2 \cdot 26,5 \cdot 40,28} \right) \\ \alpha_2 &= \cos^{-1} \left(\frac{702,25 + 1622,4784 - 702,25}{2134,84} \right) \\ \alpha_2 &= \cos^{-1} \left(\frac{1622,4784}{2134,84} \right) \\ \alpha_2 &= \cos^{-1}(0,76) \\ \alpha_2 &= 66,83^\circ \end{aligned}$$

Setelah mengetahui nilai α_2 maka α_1 dan α_2 dijumlahkan agar mendapatkan sudut θ_1 seperti pada persamaan (7).

$$\begin{aligned} \theta_1 &= \alpha_1 + \alpha_2 & (7) \\ \theta_1 &= 25,88 + 66,83 \\ \theta_1 &= 92,71^\circ \end{aligned}$$



Selanjutnya visualisasi gambar 2D untuk mencari sudut θ_2 seperti pada Gambar 8.



Gambar 8: Visualisasi 2D Untuk Mencari θ_2

Sudut θ_2 dapat ditentukan dengan menggunakan rumus cos segitiga tidak beraturan seperti pada persamaan (8).

$$\theta_2 = \cos^{-1} \left(\frac{l_2^2 + l_1^2 - c^2}{2 \cdot l_2 \cdot l_1} \right) \tag{8}$$

$$\theta_2 = \cos^{-1} \left(\frac{(26,5)^2 + (26,5)^2 - (20,85)^2}{2 \cdot 26,5 \cdot 26,5} \right)$$

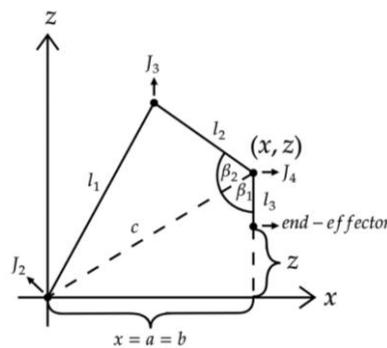
$$\theta_2 = \cos^{-1} \left(\frac{702,25 + 702,25 - 1622,4784}{1404,5} \right)$$

$$\theta_2 = \cos^{-1} \left(\frac{-217,9784}{1404,5} \right)$$

$$\theta_2 = \cos^{-1}(-0,1552)$$

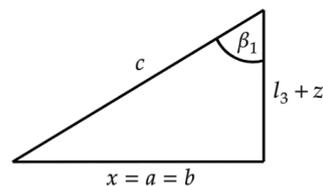
$$\theta_2 = 46,33^\circ$$

Setelah mengetahui sudut θ_2 maka selanjutnya dapat mencari sudut θ_3 .



Gambar 9: Visualisasi 2D Untuk Mencari θ_3

Dari Gambar 9 agar lebih mudah dalam mencari θ_3 maka dapat membagi sudut θ_3 menjadi 2 bagian yaitu sudut β_1 dan β_2 .



Gambar 10: Visualisasi 2D Untuk Mencari β_1

Dari Gambar 10 sudut β_1 dapat dihitung menggunakan persamaan (9).

$$\beta_1 = \tan^{-1} \left(\frac{b}{l_3 + z} \right) \tag{9}$$

$$\beta_1 = \tan^{-1} \left(\frac{18,76}{9,1 + 0} \right)$$

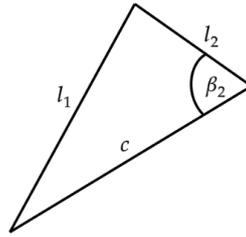
$$\beta_1 = \tan^{-1} \left(\frac{18,76}{9,1} \right)$$

$$\beta_1 = \tan^{-1}(4,31208)$$

$$\beta_1 = 64,12^\circ$$



Setelah mengetahui β_1 maka selanjutnya dapat mencari sudut β_2 .



Gambar 11: Visualisasi 2D Untuk Mencari β_2

Pada Gambar 11 untuk menghitung sudut β_2 dapat menggunakan rumus cos segitiga tidak beraturan seperti pada persamaan (10).

$$\beta_2 = \cos^{-1} \frac{l_2^2 + c^2 - l_1^2}{2 \cdot l_2 \cdot c} \quad (10)$$

$$\beta_2 = \left(\frac{(26,5)^2 + (20,85)^2 - (26,5)^2}{2 \cdot 26,5 \cdot 40,28} \right)$$

$$\beta_2 = \cos^{-1} \left(\frac{702,25 + 1622,4784 - 702,25}{2134,84} \right)$$

$$\beta_2 = \cos^{-1} \left(\frac{1622,4784}{2134,84} \right)$$

$$\beta_2 = \cos^{-1}(0,76)$$

$$\beta_2 = 66,83^\circ$$

Setelah mengetahui nilai β_2 maka dapat menjumlahkan β_1 dan β_2 agar mendapatkan nilai θ_3 seperti pada persamaan (11).

$$\theta_3 = \beta_1 + \beta_2 \quad (11)$$

$$\theta_3 = 64,12 + 66,83$$

$$\theta_3 = 130,95^\circ$$

Setelah nilai $\theta_0, \theta_1, \theta_2$, dan θ_3 diketahui, langkah selanjutnya yaitu menentukan jumlah *step* yang dibutuhkan *planatery*. Pada spesifikasi, motor stepper membutuhkan 200 *step* untuk menghasilkan 360° , dikarenakan peneliti menggunakan *planatery* dengan perbandingan 1:6 untuk *planatery* lengan dan 1:3,975 untuk *planatery base*, maka dapat dibandingkan terlebih dahulu *step* motor dengan *step planatery* seperti pada persamaan (12). *Step* yang dibutuhkan *planatery* lengan dapat ditentukan seperti pada perhitungan berikut:

$$\left(\frac{1}{6} \right) = \left(\frac{Sm}{Sp} \right) \quad (12)$$

$$\left(\frac{1}{6} \right) = \left(\frac{200}{Sp} \right)$$

$$Sp = 200 \times 6$$

$$Sp = 1200 \text{ step}$$

Keterangan:

Sm = *Step Motor*

Sp = *Step Planatery*

Untuk mengetahui *step* yang dibutuhkan *planatery base* dapat ditentukan seperti pada perhitungan berikut:

$$\left(\frac{1}{3,975} \right) = \left(\frac{Sm}{Sp} \right) \quad (13)$$

$$\left(\frac{1}{6} \right) = \left(\frac{200}{Sp} \right)$$

$$Sp = 200 \times 3,975$$

$$Sp = 795 \text{ step}$$

Keterangan:

Sm = *Step Motor*

Sp = *Step Planatery*



Kemudian, dapat mencari nilai *step* yang diperlukan untuk menghasilkan sudut θ_0 dengan melakukan perbandingan seperti rumus persamaan (14). *Step* yang diperlukan untuk mencapai sudut pada θ_0 dapat ditentukan dengan perhitungan berikut:

$$Sd = \left(\frac{\theta_0}{360^\circ} \right) \times 795 \quad (14)$$

$$Sd = \left(\frac{23,56}{360} \right) \times 795$$

$$Sd = 52 \text{ Step}$$

Keterangan:

Sd = *Step* yang diperlukan

Untuk mencari *step* yang dibutuhkan oleh *planatery* lengan tidak dapat mencari *step* yang dibutuhkan dengan cara seperti mencari sudut θ_0 atau sudut *planatery base* dikarenakan sudut awal *planatery* lengan robot yang tidak diketahui sudutnya, maka harus menentukan posisi setiap *joint* robot pada posisi awal menjadi 90° untuk semua *joint* lengan dengan cara manual agar mempermudah perhitungan dalam mencari *step* yang dibutuhkan.

Setelah berhasil membuat *planatery* lengan menjadi 90° maka dapat menghitung sudut yang dibutuhkan pada setiap *jointnya*. Dikarenakan posisi awal robot adalah 90° maka harus mengurangi posisi awal robot dengan sudut yang dibutuhkan agar diketahui selisih dari sudut yang dihasilkan. *Step* untuk mencapai sudut pada θ_1 dapat ditentukan dengan perhitungan seperti di bawah ini.

$$Ss = Sa - \theta_1 \quad (15)$$

$$Ss = 90 - 92,71$$

$$Ss = 2,71^\circ$$

Keterangan:

Ss = Sudut selisih

Sa = Sudut Posisi Awal

Setelah itu mencari *step* yang dibutuhkan oleh sudut yang dihasilkan oleh perhitungan diatas:

$$Sd = \left(\frac{Ss}{360^\circ} \right) \times 1200 \quad (16)$$

$$Sd = \left(\frac{2,71}{360} \right) \times 1200$$

$$Sd = 9 \text{ Step}$$

Keterangan:

Sd = *Step* yang diperlukan

Selanjutnya dapat menghitung *step* yang dibutuhkan dengan cara menambahkan atau mengurangi *step* yang dibutuhkan oleh sudut awal 90° dengan *step* yang diperlukan, sudut awal 90 derajat *joint* 2 memerlukan 247 *step*. Dikarenakan *joint* 2 sudut yang dihasilkan ketika bertambah *step* yaitu sudut akan bertambah kecil dan sudut yang dibutuhkan lebih besar dari 90° maka kita akan mengurangi *step* sudut awal 90° dengan *step* selisih sudut yang telah dihitung agar dapat menghasilkan sudut $92,71$ derajat.

$$247 - 9 = 238 \text{ Step}$$

Untuk mencapai sudut θ_2 dapat ditentukan berapa *step* yang diperlukan seperti pada perhitungan berikut:

$$Sa - \theta_2 = Ss \quad (17)$$

$$90 - 46,33 = 43,67$$

Keterangan:

Ss = Sudut selisih

Sa = Sudut Posisi Awal

Setelah itu mencari *step* yang dibutuhkan oleh sudut yang dihasilkan oleh perhitungan diatas:

$$Sd = \left(\frac{Ss}{360^\circ} \right) \times 1200 \quad (18)$$

$$Sd = \left(\frac{43,67}{360} \right) \times 1200$$

$$Sd = 146 \text{ Step}$$

Keterangan:



Sd = *Step* yang diperlukan
 Ss = Sudut selisih

Setelah itu dapat menghitung *step* yang dibutuhkan dengan cara menambahkan atau mengurangi *step* yang dibutuhkan oleh sudut awal 90° dengan *step* yang diperlukan, sudut awal 90° *joint* 3 memerlukan 213 *step*. Dikarenakan *joint* 3 sudut yang dihasilkan ketika bertambah *step* yaitu sudut akan bertambah besar dan sudut yang dibutuhkan lebih kecil dari 90° maka *step* sudut awal 90° dikurangi dengan *step* selisih sudut yang telah dihitung agar dapat menghasilkan sudut 46,33 derajat.

$$213 - 146 = 67 \text{ Step}$$

Untuk mencapai sudut θ_3 dapat ditentukan berapa *step* yang diperlukan seperti pada perhitungan berikut:

$$Sa - \theta_3 = Ss$$

$$90 - 130,95 = 40,95$$

Keterangan:

Ss = Sudut selisih

Sa = Sudut Posisi Awal

Setelah itu mencari *step* yang dibutuhkan oleh sudut yang dihasilkan oleh perhitungan diatas:

$$Sd = \left(\frac{Ss}{360^\circ} \right) \times 1200$$

$$Sd = \left(\frac{40,95}{360} \right) \times 1200$$

$$Sd = 136 \text{ Step}$$

Keterangan:

Sd = *Step* yang diperlukan

Ss = Sudut selisih

Kemudian dapat menghitung *step* yang dibutuhkan dengan cara menambahkan atau mengurangi *step* yang dibutuhkan oleh sudut awal 90° dengan *step* yang diperlukan, sudut awal 90° *joint* 4 memerlukan 745 *step*. Dikarenakan *joint* 4 sudut yang dihasilkan ketika bertambah *step* yaitu sudut akan bertambah kecil dan sudut yang dibutuhkan lebih besar dari 90° maka kita akan mengurangi *step* sudut awal 90° dengan *step* selisih sudut yang telah dihitung agar dapat menghasilkan sudut 130,95 derajat.

$$745 - 136 = 610 \text{ Step}$$

3. HASIL DAN PEMBAHASAN

3.1 Pengujian Kalibrasi Motor Driver DRV8825

Pengujian kalibrasi motor driver dilakukan menggunakan persamaan (21).

$$V_{ref} = I_m \times 5 \times R_{sense}$$

$$V_{ref} = I_m \times 0.5$$

$$V_{ref} = 1,8 \times 0.5$$

$$V_{ref} = 0,9$$

Hasil dari perhitungan tegangan referensi untuk menyesuaikan arus yang dihasilkan oleh driver motor DRV8825 sama dengan spesifikasi dari motor stepper yang dipakai, setelah perhitungan tegangan referensi diketahui peneliti dapat memutar trimpot pada driver motor dan mengukur menggunakan *voltmeter* dengan kutub positif di kaki trimpot dan kutub negatif pada *input* motor driver. Hasil dari kalibrasi driver motor DRV8825 ditunjukkan pada tabel I.

Tabel I: Hasil Kalibrasi Motor Driver DRV8825

No	V_{ref} (volt)	Output (volt)		Hasil	
		<i>voltmeter</i>	Selisih	% error	
1	0,9	0,9	0	0 %	
Rata-rata error				0 %	

Pada tabel I dapat disimpulkan bahwa kalibrasi motor driver DRV8825 telah menghasilkan *output* yang sama dengan perhitungan tegangan referensi yang telah dihitung sebelumnya. Nilai rata-rata

(19)

(20)

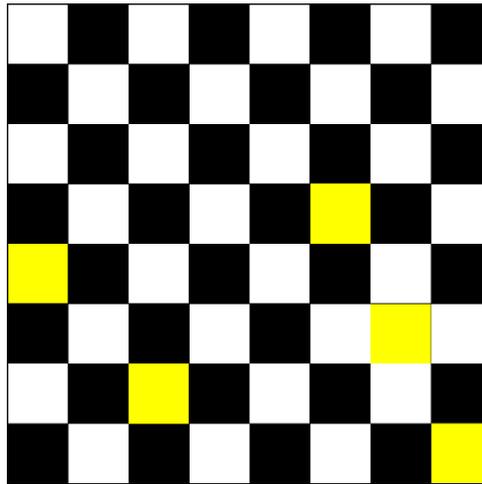
(21)



error yang dihasilkan sebesar 0 %. Nilai rata-rata *error* didapatkan dari penjumlahan % *error* yang dibagi dengan banyaknya data yang diambil.

3.2 Pengujian *Inverse Kinematics*

Pengujian *inverse kinematics* dilakukan dengan melakukan serangkaian percobaan dimana posisi tujuan atau titik akhir dari robot *arm* ditentukan. Dengan menggunakan metode *inverse kinematics*, dihitung sudut-sudut yang diperlukan oleh setiap *joint* robot *arm* untuk mencapai posisi tujuan tersebut. Data hasil perhitungan *inverse kinematics* dibandingkan dengan posisi tujuan dan *step* motor yang dibutuhkan yang ditentukan sebagai acuan. Dengan membandingkan kedua data ini, dapat dievaluasi sejauh mana metode *inverse kinematics* mampu menghasilkan sudut-sudut yang akurat untuk mencapai posisi tujuan yang diinginkan. Pengujian dilakukan dengan mencoba memposisikan *end-effector* ke beberapa titik koordinat seperti pada Gambar 12.



Gambar 12: Posisi Koordinat Data Yang Diambil

Tabel II: Titik Koordinat Masing-Masing Kotak Yang Diuji

Nomor Papan Catur	Sumbu <i>x</i>	Sumbu <i>y</i>	Sumbu <i>z</i>
H1	17,5	12,2	0
C2	7,5	17,2	0
G3	12,5	22,2	0
A4	17,5	27,2	0
F5	7,5	32,2	0

Tabel II memberikan informasi mengenai titik koordinat mana saja yang akan diuji dalam pengujian *inverse kinematics*.

Tabel III: Sudut Yang Dibutuhkan Masing – Masing *Joint*

Nomor Papan Catur	<i>Joint</i> 1	<i>Joint</i> 2	<i>Joint</i> 3	<i>Joint</i> 4
H1	55,12°	87,15°	51,89°	130,95°
C2	23,56°	92,71°	46,33°	130,95°
G3	29,38°	78,95°	61,40°	129,65°
A4	32,76°	66,56°	78,41°	125,04°
F5	13,11°	65,08°	80,63°	124,3°



Tabel III menunjukkan sudut yang dibutuhkan pada masing-masing *joint* dan pada Tabel IV menunjukkan *step* yang dibutuhkan untuk mencapai posisi akhir *end-effector* yang diinginkan.

Tabel IV: *Step* Yang Dibutuhkan Masing – Masing *Joint*

Nomor Papan Catur	<i>Joint</i> 1	<i>Joint</i> 2	<i>Joint</i> 3	<i>Joint</i> 4
H1	122 <i>Step</i>	249 <i>Step</i>	85 <i>Step</i>	610 <i>Step</i>
C2	52 <i>Step</i>	230 <i>Step</i>	66 <i>Step</i>	612 <i>Step</i>
G3	65 <i>Step</i>	276 <i>Step</i>	117 <i>Step</i>	615 <i>Step</i>
A4	72 <i>Step</i>	317 <i>Step</i>	173 <i>Step</i>	630 <i>Step</i>
F5	29 <i>Step</i>	322 <i>Step</i>	181 <i>Step</i>	633 <i>Step</i>

Setelah pengujian *inverse kinematics* dilakukan, sudut yang dihasilkan masing-masing *joint* terdapat pada Tabel V.

Tabel V: Sudut Yang Dihasilkan Masing – Masing *Joint*

Nomor Papan Catur	<i>Joint</i> 1	<i>Joint</i> 2	<i>Joint</i> 3	<i>Joint</i> 4
H1	56,02°	88,05°	52,79°	132,75°
C2	22,56°	94,51°	45,43°	131,85°
G3	29,38°	78,05°	62,3°	128,75°
A4	28,48°	67,46°	80,21°	124,24°
F5	14,01°	64,18°	78,83°	126,10°

Sudut yang dihasilkan pada Tabel V dibandingkan dengan sudut awal yang dibutuhkan. Dari hasil perbandingan tersebut, dapat diketahui selisih sudut yang dihasilkan sehingga nilai *error* dari hasil pengujian dapat diketahui. Hasil perbandingan sudut pada *joint* 1 terdapat pada Tabel VI.

Tabel VI: Perbandingan Sudut *Joint* 1 Pengujian *Inverse Kinematics*

Nomor Papan Catur	Sudut Yang Dibutuhkan	Sudut Yang Dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	55,12°	56,02°	0,9	1,63%
C2	23,56°	25,36°	1,8	7,64%
G3	29,38°	30,28°	0,9	3,06%
A4	32,76°	33,66°	0,9	2,74%
F5	13,11°	14,01°	0,9	6,86%
Rata-rata <i>error</i>				4,39%

Dari hasil pengujian pada *joint* 1, rata-rata *error* yang dihasilkan sebesar 4,39%. Rata-rata *error* pada Tabel VI diperoleh dari hasil penjumlahan % *error* kemudian dibagi dengan jumlah data.



Tabel VII: Perbandingan Sudut *Joint 2* Pengujian *Inverse Kinematics*

Nomor Papan Catur	Sudut Yang Dibutuhkan	Sudut Yang Dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	87,15°	88,05°	0,9	1,03%
C2	92,71°	94,51°	1,8	1,94%
G3	78,95°	78,05°	0,9	1,13%
A4	66,56°	67,46°	0,9	1,35%
F5	65,08°	64,18°	0,9	1,38%
Rata-rata <i>error</i>				1,36%

Dari hasil pengujian pada *joint 2*, rata-rata *error* yang dihasilkan dari pengujian pada *joint 2* sebesar 1,36%. Perbandingan sudut yang dibutuhkan dengan sudut yang dihasilkan pada *joint 3* ditunjukkan pada Tabel VIII.

Tabel VIII: Perbandingan Sudut *Joint 3* Pengujian *Inverse Kinematics*

Nomor Papan Catur	Sudut Yang Dibutuhkan	Sudut Yang Dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	51,89°	52,79°	0,9	1,73%
C2	46,33°	45,43°	0,9	1,94%
G3	61,40°	62,3°	0,9	1,46%
A4	78,41°	80,21°	1,8	2,29%
F5	80,63°	78,83°	1,8	2,23%
Rata-rata <i>error</i>				1,93%

Dari hasil pengujian, rata-rata *error* yang dihasilkan pada *joint 3* sebesar 1,93%. Perbandingan sudut yang dibutuhkan dengan sudut yang dihasilkan pada *joint 4* ditunjukkan pada Tabel IX.

Tabel IX: Perbandingan Sudut *Joint 4* Pengujian *Inverse Kinematics*

Nomor Papan Catur	Sudut Yang Dibutuhkan	Sudut Yang Dihasilkan	Hasil	
			Selisih	% <i>error</i>
H1	130,95°	132,75°	1,8	1,37%
C2	130,95°	131,85°	0,9	0,68%
G3	129,65°	128,75°	0,9	0,69%
A4	125,04°	124,24°	0,8	0,63%
F5	124,3°	126,10°	1,8	1,44%
Rata-rata <i>error</i>				0,96%

Dari hasil pengujian, rata-rata *error* yang dihasilkan pada *joint 4* sebesar 0,96%. Hasil pengujian pada *joint 1, 2, 3* dan *4* menunjukkan bahwa robot *arm* dapat dikatakan berhasil dengan rata-rata *error* yang masih dalam batas toleransi. *Error* yang terjadi disebabkan karena beban yang diterima oleh *planetary* terlalu berat yang mengakibatkan terjadinya *missing gear*.



Posisi koordinat akhir dari *end-effector* dapat dilihat pada Tabel X dan selanjutnya dapat diketahui *error* koordinat yang seharusnya dengan koordinat yang sudah dihasilkan.

Tabel X: Posisi Koordinat Akhir Dari *End- Effector*

Nomor Papan Catur	Sumbu x	Sumbu y	Sumbu z
H1	17,1	12,0	0,2
C2	7,6	17,1	0
G3	12,2	21,9	0,1
A4	17,7	27,5	0,1
F5	7,2	32,3	0,3

Hasil pengujian memberikan informasi penting tentang akurasi dan keandalan metode *inverse kinematics* yang digunakan dalam perancangan robot *arm*. Pengujian *inverse kinematics* dilakukan pada masing masing sumbu, yaitu: sumbu x , sumbu y , dan sumbu z . Pengujian *inverse kinematics* pada masing-masing sumbu dilakukan pada lima titik koordinat bidak catur.

Tabel XI: Perbandingan Koordinat Sumbu x Pengujian *Inverse Kinematics*

Nomor Papan Catur	Posisi koordinat yang dibutuhkan	Posisi koordinat yang dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	17,5	17,1	0,4	2,28%
C2	7,5	7,6	0,1	1,3%
G3	12,5	12,2	0,3	2,4%
A4	17,5	17,7	0,2	1,14%
F5	7,5	7,2	0,3	4%
Rata-rata <i>error</i>			2,22%	

Hasil pengujian *inverse kinematics* pada sumbu x terlihat pada Tabel XII memiliki rata-rata *error* pada lima titik koordinat bidak catur sebesar 2,22%.

Tabel XII: Perbandingan Koordinat Sumbu y Pengujian *Inverse Kinematics*

Nomor Papan Catur	Posisi koordinat yang dibutuhkan	Posisi koordinat yang dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	12,2	12,0	0,2	1,63%
C2	17,2	17,1	0,1	0,58%
G3	22,2	21,9	0,3	1,35%
A4	27,2	27,5	0,3	1,10%
F5	32,2	32,3	0,1	0,31%
Rata-rata <i>error</i>			0,99%	

Hasil pengujian *inverse kinematics* pada sumbu y sebagaimana terlihat pada Tabel XII memiliki *error* yang lebih kecil dari pengujian pada sumbu x . Pengujian *inverse kinematics* pada sumbu y



didapatkan *error* maksimal pada titik koordinat H1 dengan *error* sebesar 1,63%. Pada pengujian *inverse kinematics* sumbu *y* didapatkan *error* rata-rata sebesar 0,99%.

Tabel XIII: Perbandingan Koordinat Sumbu *z* Pengujian *Inverse Kinematics*

Nomor Papan Catur	Posisi koordinat yang dibutuhkan	Posisi koordinat yang dihasilkan	Hasil	
			Selisih	<i>error</i>
H1	0	0,2	0,2	0%
C2	0	0	0	0%
G3	0	0,1	0,1	0%
A4	0	0,1	0,1	0%
F5	0	0,3	0,3	0%
Rata-rata <i>error</i>				0%

Hasil pengujian *inverse kinematics* pada sumbu *z* sebagaimana terlihat pada Tabel XIII memiliki *error* yang lebih kecil dari pengujian pada sumbu *y*. Pengujian *inverse kinematics* pada sumbu *z* didapatkan nilai *error* rata-rata sebesar 0%.

Dari pengujian *invers kinematics* dengan robot *arm* 4 DOF yang dilakukan sebanyak lima titik koordinat percobaan dihasilkan *error* rata-rata pada sumbu *x* sebesar 2,22%, pada sumbu *y* sebesar 0,99%, pada sumbu *z* sebesar 0%. *Error* pada pengujian *invers kinematics* terjadi karena *planatery* yang kurang presisi dan juga diakibatkan oleh banyak faktor.

4. KESIMPULAN

1. Dengan menggunakan metode *invers kinematics*, robot *arm* berhasil mencapai tujuan koordinat yang telah ditentukan. Robot *arm* 4 degree of freedom berhasil memindahkan bidak catur dalam pengujian sebanyak 1 siklus pengambilan dan peletakan bidak catur.
2. Penggunaan metode *inverse kinematics* pada robot *arm* menghasilkan sudut yang presisi tetapi tetap berhubungan dengan hasil dari sudut yang dihasilkan oleh *planatery* itu sendiri. Pendekatan ini membantu meningkatkan kualitas pergerakan robot secara keseluruhan.
3. Penggunaan metode *trajectory planning* menghasilkan pergerakan motor *stepper* berjalan dengan mulus dan tidak melewati batas posisi koordinat yang telah ditentukan.
4. Robot *arm* 4 degree of freedom yang dibangun memiliki tingkat *error* yang kecil pada sumbu *x* sebesar 2,22%, pada sumbu *y* sebesar 1,35%, dan pada sumbu *z* sebesar 0,99%. Ini menunjukkan bahwa robot memiliki presisi yang tinggi dalam pergerakan. *Error* ini disebabkan oleh ketidakpresisian *planatery* pada saat robot *arm* menjalankan tugas.

5. UCAPAN TERIMA KASIH

Terimakasih kepada kedua orang tua yang telah mendukung saya dan dosen pembimbing yang telah memberikan ilmu kepada saya.



DAFTAR PUSTAKA

- [1] Salmon, S., & Harpad, B. (2020). Prototype Lengan Robot Berbasis Arduino Dengan Menggunakan Algoritma Kinematics. *Jurnal Ilmiah Matrik*, 22(2), 208-214.
- [2] Didi, M. (2016). Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno. *Jurnal Teknik Elektro Universitas Tanjungpura*, 1(1). [3] R. Srivatsan, S. Badrinath, dan G. L. Sutha, "Autonomous chess-playing robotic arm using Raspberry PI," dalam *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, IEEE, 2020, hlm. 1–6.
- [3] Sinaga, M. A. H. K., & Pamungkas, D. S. (2020, November). INVERS KINEMATIK ROBOT ARM 4 DOF MENGGUNAKAN SENSOR LEAP MOTION. In *Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV)* (Vol. 6, No. 1, pp. 363-371).
- [4] Rahman, A. Z., Jauhari, K., Sumantri, D., Hendro, T., Nugraha, I. D., & Amrullah, S. (2019). Inverse Kinematics dan Pengukuran Akurasi Pergerakan pada Model Robot Manipulator Lengan. *J. Tek. MESIN*, 3(2), 53-58.
- [5] Sinaga, M. A. H. K., & Pamungkas, D. S. (2020, November). INVERS KINEMATIK ROBOT ARM 4 DOF MENGGUNAKAN SENSOR LEAP MOTION. In *Prosiding Seminar Nasional Terapan Riset Inovatif (SENTRINOV)* (Vol. 6, No. 1, pp. 363-371).
- [6] Utomo, B., Setyaningsih, N. Y. D., & Iqbal, M. (2020). Kendali Robot Lengan 4 DoF Berbasis Arduino Uno dan Sensor MPU-6050. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 11(1), 89-96. [4]