

# Motion Planning Defense Robot Dalam Mengikuti Throw Robot Menggunakan Kontrol Kinematik

Krisnawan Pambudi<sup>1</sup>, Totok Winarno<sup>2</sup>, Supriatna Adhisuwignjo<sup>3</sup>

[Submission: 30-07-2021, Accepted:30-08-2021]

**Abstract**— In the KRAI 2021 race, robots are required to take an arrow at a designated place and throw it in a pot that has been available in accordance with the rules published by the Asia-Pacific Broadcasting Union. There are two robots that each have different functions, *throw robot* that has the task to take arrows that have been provided to be thrown into the pot, then the second robot is *Defense Robot* whose one of the tasks is to give arrows that fail to enter the pot to throw the *Robot* to be thrown back. In this study presented an experiment on robots designed using 3 *omni wheels* controlled using kinematic control, equipped with rotary encoder sensors and compass sensors to follow the movement and direction of other robots in real time. The robot is designed using 3 *wheels omni wheels* and uses kinematic controls capable of moving quickly because the robot is able to move to the desired position by forming a linear trajectory. From test results with a gain or lambda value of 0.8 position errors can be reduced by close to 0% within 8s.

**Intisari**— Rule pada perlombaan KRAI 2021 robot diharuskan untuk mengambil anak panah pada tempat yang telah ditentukan dan melemparkannya ke dalam pot yang telah tersedia. Terdapat dua robot yang masing – masing memiliki fungsi yang berbeda yaitu *Throw Robot* yang memiliki tugas untuk mengambil anak panah yang telah disediakan untuk dilemparkan ke dalam pot, kemudian robot yang kedua adalah *Defense Robot* yang salah satu tugasnya adalah memberikan anak panah yang gagal memasuki pot kepada *Throw Robot* untuk dilemparkan kembali. Dalam penelitian ini dirancang robot dengan menggunakan 3 buah roda *omni wheels* yang dikontrol menggunakan kontrol kinematik, dilengkapi dengan sensor rotary encoder dan sensor kompas untuk mengikuti pergerakan dan arah hadap robot lainnya secara real time. Dari hasil pengujian dengan hasil nilai penguatan atau lambda sebesar 0,8 error posisi dapat berkurang mendekati 0% dalam waktu waktu 8s.

**Kata Kunci**— Kontrol Kinematik, *Mobile Robot Omni Wheel*, Sensor kompas, Sensor rotary encoder.

## I. PENDAHULUAN

Salah satu kontes robot yang bergengsi di Asia-Pasifik adalah *ABU ASIA- PACIFIC ROBOT CONTEST 2021* dengan tema “*Throwing Arrows Into Pots*” dan di Indonesia disebut KRAI 2021 [1]. Robot Indonesia (KRI) adalah ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika. Pada perlombaan KRAI 2021 robot diharuskan untuk mengambil anak panah pada tempat yang telah ditentukan dan

melemparkannya ke dalam pot yang telah tersedia sesuai dengan aturan yang diterbitkan oleh *Asia-Pacific Broadcasting Union*. Terdapat dua robot yang masing – masing memiliki fungsi yang berbeda yaitu *Throw Robot* yang memiliki tugas untuk mengambil anak panah yang telah disediakan untuk dilemparkan ke dalam pot, kemudian robot yang kedua adalah *Defense Robot* yang salah satu tugasnya adalah memberikan anak panah yang gagal memasuki pot kepada *Throw Robot* untuk dilemparkan kembali.

Tantangan yang dihadapi adalah bagaimana robot dapat bernavigasi dan mengetahui dimana posisi robot saat bergerak dengan sudut tertentu untuk mencapai suatu titik target. Robot menggunakan system penggerak differential dengan sensor rotary encoder yang diolah dengan metode odometry sehingga menghasilkan titik koordinat. Robot nantinya bergerak pada posisi awal (0,0) menuju ke titik target (x',y'), untuk menuju ke titik target robot ini di kontrol dengan metode odometry dengan memadukan metode PID, Hasil yang didapat dengan menentukan titik target dengan kecepatan 50 (pwm) menunjukkan bahwa robot dapat mengikuti jalur yang dibuat dengan simpangan terjauh sebesar 8cm pada sumbu x negatif sampai 7cm pada sumbu x positif serta 25cm pada sumbu y negatif dan 16cm pada sumbu y positif [2].

Agar *Defense Robot* mampu untuk memberikan anak panah kepada *Throw Robot* dengan tepat maka *Defense Robot* harus mampu mengikuti pergerakan *Throw Robot* dan menyesuaikan arah hadapnya terhadap *Throw Robot*. Pada pergerakan *mobile* dengan metode kontrol differential drive dengan menggunakan empat roda yang diharapkan robot bisa berpindah posisi tanpa merubah arah hadap, dengan memadukan metode PID respon robot menjadi lebih baik dan mendekati setpoint yang di inginkan[3].

Pengembangan desain kontrol kinematik untuk robot beroda sangat dibutuhkan mengingat bahwa robot beroda memiliki banyak parameter yang mampu merubah persamaan kontrol kinematiknya terutama pada kontrol kinematik untuk robot beroda dengan jenis roda omnidirectional, baik dari segi

p-ISSN: 2356-0533; e-ISSN: 2355-9195

<sup>1</sup>Mahasiswa, Program Studi D-IV Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jln. Soekarno Hatta no.9, e-mail:pambudi.krisnawan@gmail.com

<sup>2,3</sup> Dosen, Program Studi D-IV Teknik Elektronika Jurusan Teknik Elektro Politeknik Negeri Malang, Jln. Soekarno Hatta no. 9 Malang, e-mail: totok.winarno@polinema.ac.id, supriatna@polinema.ac.id

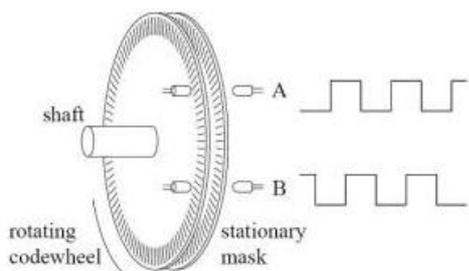


jenis roda yang digunakan hingga jumlah roda penggerak yang digunakan oleh karena itu diperlukan suatu persamaan kinematik yang *general* [4]. Tujuan dari penelitian ini adalah dapat merancang *Motion Planning* dari *Defense Robot* menggunakan kontrol kinematik agar pergerakan *Defense Robot* dapat mengikuti pergerakan *Throw Robot* secara *real time*.

## II. KAJIAN PUSTAKA

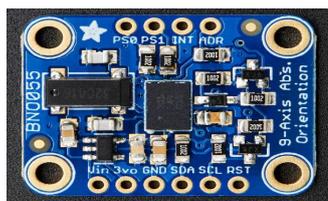
### A. Sensor Rotary Encoder

*Rotary Encoder* terdiri dari dua track atau single track dan dua sensor dinamakan channel A dan channel B. Ketika poros berputar yang deretan pulsa akan muncul di masing masing channel pada frekuensi yang proposional dengan kecepatan putar sedangkan hubungan fasa antara channel A dan B menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur untuk mengetahui arah putaran dengan mengetahui channel mana yang leading terhadap channel satunya karena kedua channel tersebut akan mengalami perbedaan fasa seperempat putaran.



Gambar 1: Kinerja rotary encoder

### B. Sensor Kompas Adafruit BNO055



Gambar 2: Sensor Kompas Adafruit BNO055

Sensor ini terdiri dari 9 DOF diantaranya terdapat akselerometer, giroskop dan magnetometer. Fungsi dari sensor tersebut sangat penting bagi mobile robot dikarenakan sensor ini dapat mendeteksi percepatan dan arah gadap pada mobile robot. Agar pendeteksian percepatan dan arah hadap yang sesuai maka ketiga dari 9 DOF tersebut harus di gunakan dengan perhitungan yang kompleks. Diantara kegunaan dari sensor ini adalah data orientasi absolut 0 – 360° dengan kecepatan gerak orientasi salam satuan quertion

$(x, y, z, \omega)$  dengan satuan rad/s dan percepatan dalam satuan  $m/s^2$

### C. Dasar kontrol kinematik

Berdasarkan pengontrolan kinematik terdapat beberapa permasalahan diantaranya set point yang digunakan adalah posisi sedangkan tidak tahu berapa kecepatan yang digunakan. Maka digunakan rumus *invers kinematik* yang diperoleh dari rumus *global frame* ( ${}^g\dot{x}_R$ ) untuk mencari kecepatan yang digunakan yang mana hasil kecepatan tersebut akan digunakan sebagai rumus *forward kinematik*. Rumus kecepatan *invers kinematik*:

$${}^g\dot{x}_R = J_R \cdot \omega_R \tag{6}$$

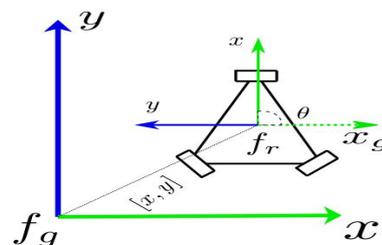
$$J_R^{-1} \cdot {}^g\dot{x}_R = J_R^{-1} \cdot J_R \cdot \omega_R \tag{7}$$

$$\omega_R = J_R^{-1} \cdot {}^g\dot{x}_R \tag{8}$$

Dimana  $J^{-1} \cdot J = I$ ,  $I$  merupakan rumus matrik identitas maka dianggap nilainya 1. Maka persamaan matriksnya menjadi :

$$\underbrace{\begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix}}_{\omega_R} \underbrace{\begin{bmatrix} \cos \theta_1 & \sin \theta_1 & k_1 \\ \cos \theta_2 & \sin \theta_2 & k_2 \\ \vdots & \vdots & \vdots \\ \cos \theta_n & \sin \theta_n & k_n \end{bmatrix}}_{J_R^{-1}} \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 \\ -\sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} {}^g\dot{x}_R \\ {}^g\dot{y}_R \\ {}^g\dot{\theta}_R \end{bmatrix}}_{{}^g\dot{x}_R} \tag{9}$$

### D. Persamaan kinematik robot terhadap referensi frame global



Gambar 3: Referensi robot terhadap lingkungan

Pada kenyataannya mengontrol robot tidak dilakukan dengan menggerakkan robot terhadap dirinya sendiri, tetapi robot dikendalikan berdasarkan letak posisi robot terhadap lingkungannya. Posisi robot (titik tengah robot dinotasikan dengan  $f_r$ ) terhadap *frame global* ( $f_g$ ) dinotasikan dengan  ${}^g\mathbf{x}_r$ . Perlu diketahui bahwa turunannya posisi adalah kecepatan, maka notasi  ${}^g\dot{\mathbf{x}}_r$  dapat diturunkan terhadap waktu.



$$\frac{d {}^g \mathbf{x}_R(t)}{dt} = {}^g \dot{\mathbf{x}}_R(t) \quad (1)$$

Dimana untuk mencari nilai  ${}^g \dot{\mathbf{x}}_R$  dapat dilakukan dengan cara mengkalikan  ${}^R \dot{\mathbf{x}}_R$  dengan matriks rotasi terhadap  $f_g ({}^g \mathbf{Rot}_R)$ , sehingga didapatkan nilai kecepatan perubahan posisi terhadap *frame global* seperti pada gambar 2.4. Sehingga persamaan dasarnya dapat dituliskan :

$${}^g \dot{\mathbf{x}}_R(t) = {}^g \mathbf{Rot}_R(t) \cdot {}^R \dot{\mathbf{x}}_R(t) \quad (2)$$

Setelah disubstitusi maka persamaanya menjadi :

$${}^g \dot{\mathbf{x}}_R(t) = {}^g \mathbf{Rot}_R(t) \cdot {}^R \mathbf{J}_R(t) \cdot \boldsymbol{\omega}_R(t) \quad (3)$$

Maka jika dijabarkan ke persamaan matriks menjadi :

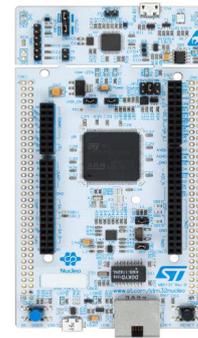
$$\underbrace{\begin{bmatrix} {}^g \dot{x}_R \\ {}^g \dot{y}_R \\ {}^g \dot{\theta}_R \end{bmatrix}}_{{}^g \dot{\mathbf{x}}_R} = r \underbrace{\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{J}_R} \underbrace{\begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & \dots & \cos \alpha_n \\ \sin \alpha_1 & \sin \alpha_2 & \dots & \sin \alpha_n \\ \frac{1}{k} & \frac{1}{k} & \dots & \frac{1}{k} \end{bmatrix}}_{\boldsymbol{\omega}_R} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{bmatrix} \quad (4)$$

Dari persamaan (4) dapat disederhanakan, sehingga akan mendapatkan rumus :

$${}^g \dot{\mathbf{x}}_R(t) = \mathbf{J}_R(t) \cdot \boldsymbol{\omega}_R(t) \quad (5)$$

**E. MIKROKONTROLLER**

Pada penelitian ini mikrokontroller yang digunakan adalah STM32F429ZI yang merupakan rangkaian sirkuit terintegrasi 32-bit yang diproduksi oleh STMicroelectronics. Chip STM32 dikelompokkan ke dalam seri terkait yang berbasis di sekitar inti prosesor ARM 32-bit yang sama, seperti Cortex-M33F, Cortex-M7F, Cortex-M4F, Cortex-M3, Cortex-M0 +, atau Cortex-M0.



Gambar 4: STM32F429ZI

**F. Odroid XU-4**

Odroid adalah sebuah single board computer atau biasa disebut mini PC yang dibuat oleh perusahaan asal korea selatan yaitu Hardkernel Co., Ltd. Meskipun bernama Odroid yang berarti Open Android, tetapi perangkat keras ini tidak sepenuhnya bersifat open source karena berkaitan dengan rahasia perusahaan. Odroid tidak hanya mampu menjalankan android tetapi mencakup linux juga. Bahkan performa odroid lebih tinggi jika dibandingkan dengan raspberry pi. Perbedaan odroid dengan raspberry pi adalah raspberry pi tergolong mini PC low end, sedangkan odroid adalah mini PC high end, pada robot ini penulis menggunakan Odroid-XU4 dimana ordroid ini mengusung solusi HMP (*Heterogeneous Multi-Processing*).



Gambar 5: Odroid XU-4

**G. Driver Motor H-Bridge IBT\_2 (BTS7960A)**

Kontrol Motor DC *High Current* pada driver motor DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan sumber DC yang dapat diberikan antara 5.5V – 27V DC, sedangkan tegangan input level antara 3.3V – 5Vdc, driver motor ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi arus yang Panas atau berlebihan.





Gambar 6: Modul Driver BTS7970A

H. Motor DC PG-45 500RPM

Motor dc PG-45 adalah motor dc geared yaitu motor dc dengan tambahan gearbox. Fungsi dari tambahan gearbox ini adalah untuk menambah besar torsi pada motor dc tersebut. Motor DC PG-45 ini memiliki torsi kurang lebih 30kgcm dengan kecepatan maksimum sebesar 500RPM.

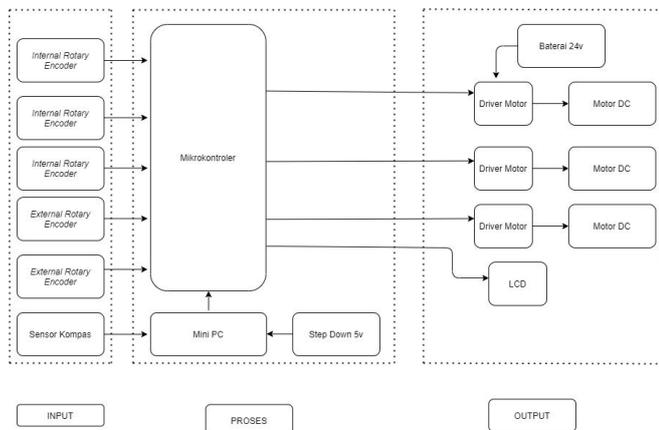


Gambar 7: Motor DC PG45

III. METODE PENELITIAN

I. DIAGRAM BLOK SISTEM

Gambar 3 merupakan diagram blok sistem dari penelitian ini. Diagram blok sistem ini mempunyai 3 blok, yaitu input, proses dan output.

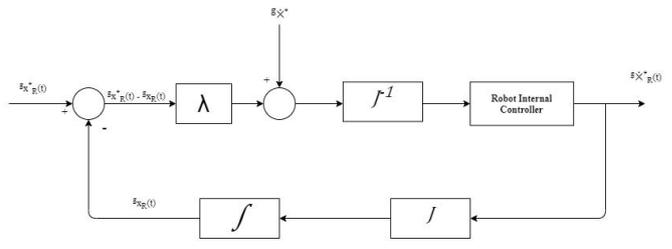


Gambar 8 : Blok diagram sistem.

Dari blok diagram tersebut dapat kita amati bahwa kita menggunakan input dari sensor rotary encoder dan sensor kompas. Sensor rotary encoder digunakan untuk mendeteksi pergerakan motor sedangkan sensor kompas digunakan untuk mengetahui arah hadap dari robot itu sendiri

Kemudian data arah hadap robot tersebut diolah dengan algoritma kinematik oleh mini PC Odroid dan mikrokontroler STM32. Kemudian setelah diproses melalui mini pc dan mikrokontroler output dari system ini adalah PWM yang digunakan untuk menggerakkan aktuator berupa motor.

J. Diagram Blok Kontrol

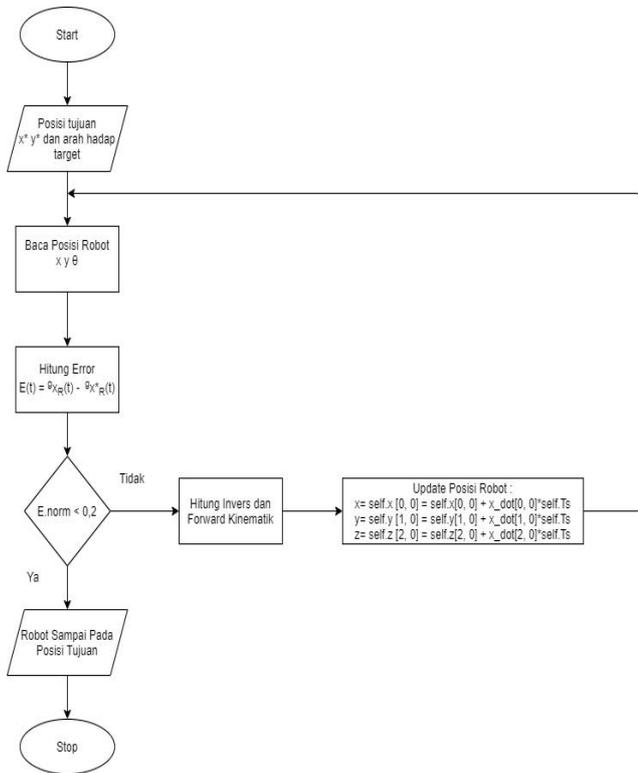


Gambar 9: Blok diagram kontrol.

Pada blok diagram kontrol di atas pembacaan nilai eror posisi didapatkan dari data odometry yang dibandingkan dengan set point. Kemudian terdapat tambahan set point ketika robot masih berjalan. Tambahan set point tersebut adalah data posisi tujuan robot yang berpindah posisi dari posisi awalnya.

K. Algoritma Robot

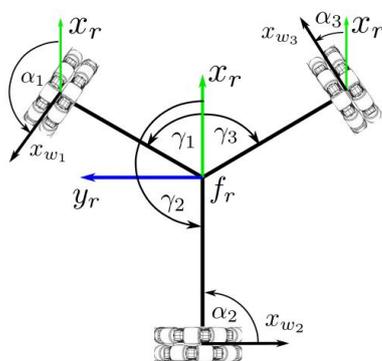




Gambar 10: Diagram alir robot.

Pada saat robot dinyalakan robot akan mendeteksi posisi dan arah hadap target yang hendak dituju. Setelah itu robot akan mendeteksi posisi robot itu sendiri dan arah hadapnya. Kemudian robot akan menghitung nilai eror yang didapat dari posisi robot sekarang dengan posisi yang akan dituju. Data – data tersebut digunakan untuk melakukan perhitungan kinematic dimana hasil perhitungan kinematic adalah data kecepatan motor yang digunakan robot untuk menuju posisi tujuan. Setelah itu robot akan melakukan perhitungan normalisasi eror apabila nilai normalisasi eror kurang dari 0,2 maka robot dianggap telah mencapai posisi tujuan dan robot tersebut akan berhenti.

L. Pemodelan Kinematik 3 roda Omniwheel



Gambar 11: Pemodelan Kinematik 3 roda Omniwheels.

Kinematik dari roda yang menggunakan 3 buah roda omni wheels yang digunakan sama seperti teori pada bagian 2.3 dimana nilai variabel dimasukkan ke persamaan yang disesuaikan dengan spesifikasi robot. didapatkan beberapa nilai parameter yang digunakan yaitu :

- $\gamma_1 = 60^\circ$  (sudut antara titik tengah roda 1 dengan arah depan robot)
- $\gamma_2 = -60^\circ$  (sudut antara titik tengah roda 2 dengan arah depan robot)
- $\gamma_3 = 180^\circ$  (sudut antara titik tengah roda 3 dengan arah depan robot)
- $\alpha_1 = 150^\circ$  (sudut antara titik tengah roda 1 dengan frame arah depan robot )
- $\alpha_2 = -90^\circ$  (sudut antara titik tengah roda 2 dengan arah depan robot )
- $\alpha_3 = 30^\circ$  (sudut antara titik tengah roda 3 dengan arah depan robot )
- $l = 0.3 \text{ m}$  (jarak titik tengah robot dengan titik tengah roda)
- $r = 0.075 \text{ m}$  (jari jari roda pada robot)

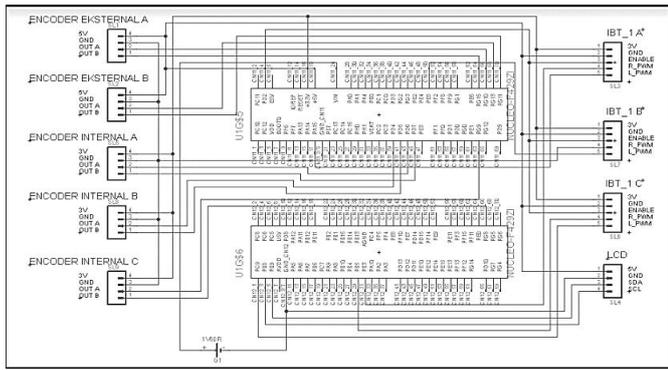
Setelah disubstitusikan maka persamaan matriksnya menjadi :

$$\begin{bmatrix} {}^R\dot{x}_R(t) \\ {}^R\dot{y}_R(t) \\ {}^R\dot{\theta}_R(t) \end{bmatrix} = 0.075 \begin{bmatrix} \cos 5/6 \pi & \cos -1/2 \pi & \cos 1/6 \pi \\ \sin 5/6 \pi & \sin -1/2 \pi & \sin 1/6 \pi \\ 1/0.3 & 1/0.3 & 1/0.3 \end{bmatrix} \begin{bmatrix} \omega_1(t) \\ \omega_2(t) \\ \omega_3(t) \end{bmatrix} \quad (21)$$

M. Perancangan Elektronik

Gambar 12 adalah rangkaian elektronik dari system robot. Rangkaian tersebut terdiri dari STM32, I2C untuk LCD, sensor rotary encoder, sensor kompas, dan driver motor.



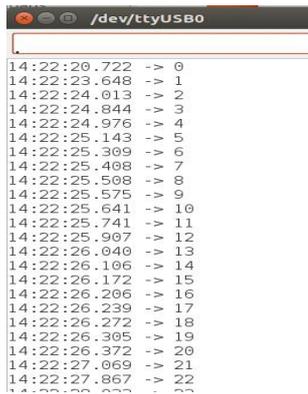


Gambar 12: Rangkaian Elektrik Secara Keseluruhan

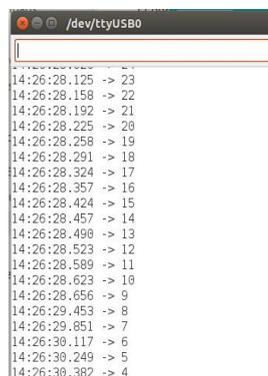
IV. HASIL DAN PEMBAHASAN

N. Pengujian dan Analisa Sensor Rotary Encoder

Pengujian pada *sensor rotary encoder* dilakukan untuk mengetahui data yang didapatkan oleh *sensor rotary encoder* pada saat terjadi perputaran pada motor DC. Pada percobaan *rotary encoder* menggunakan STM32 ini *sensor rotary encoder* harus mampu mendeteksi arah putar dari motor DC pada robot, jika motor searah jarum jam, maka *rotary encoder* nilainya akan bertambah, dan jika motor berputar berlawanan jarum jam, maka data *sensor rotary encoder* akan berkurang.



Gambar 13: Pengujian Penambahan Data Ecoder



Gambar 14: Pengujian Pengurangan Data Ecoder

O. Pengujian dan Analisa Sensor Kompas

Pada percobaan nilai data sensor kompas yang semula berada pada range 0 - 360 diubah menjadi 180 hingga -180 . Sehingga dengan begitu tidak ada masalah bila lapangan menghadap kemanapun. Orientasi robot akan tetap 0 ketika searah dengan sumbu x lapangan. Dengan ini hasil pengujian sensor BNO055 mampu memberikan data yang akurat dan perancangan data tersebut mampu mengubah arah data kompas yang akan digunakan oleh robot terhadap lapangan. Perubahan range tersebut juga untuk memudahkan penentuan set point arah hadap robot. Dari percobaan sensor kompas di atas dapat kita amati bahwa sensor kompas cukup akurat dengan nilai presentase eror yang kecil. Berikut adalah data hasil pengujian sensor kompas :

TABEL 1  
 DATA PENGUJIAN ORIENTASI KOMPAS DENGAN THRESHOLD YANG BERBEDA BEDA

No.	Arah hadap robot	Data dari sensor kompas	Error	Persentase Error
1.	0	0,21°	0,21	1,21%
2.	20	20,25°	0,25	1,25%
3.	40	40,10°	0,1	0,25%
4.	60	60,27°	0,27	0,45%
5.	80°	80,15°	0,15	0,18%
6.	100	100,17°	0,17	0,17%
7.	120	120,03°	0,03	0,02%
8.	140	140,2°	0,2	0,14%
9.	160	160,26°	0,26	0,16%
10.	180	179,97°	0,03	0,01%

P. Pengujian dan Analisa Odometry

Pengujian ini dilakukan dengan cara yang pertama yaitu menentukan titik acuan koordinatnya yaitu (0.0), titik koordinat ini digunakan sebagai acuan statis atau tidak boleh berubah ubah dan saat sudah dikalibrasi maka arah depan robot menjadi sumbu x pada lapangannya. Kemudian mendorong robot secara manual ke sumbu x dan y pada lapangan dengan syarat robot tidak boleh terangkat agar sensor rotary encoder terbaca, setelah itu jarak yang telah



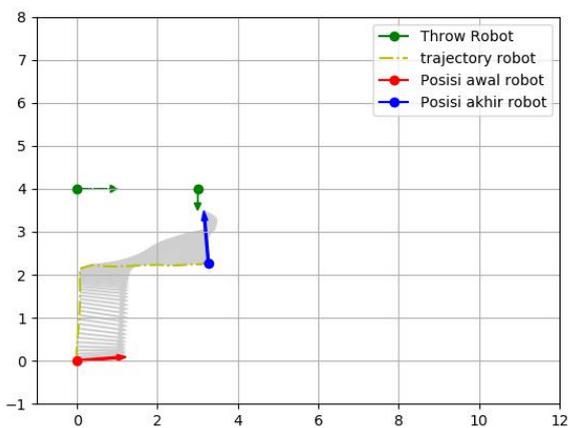
ditempuh oleh robot tersebut diukur dengan meteran. Berikut adalah hasil pengujian *odometry* :

TABEL 2  
 PENGUJIAN DATA *REAL* DAN DATA ODOMETRY

Odometry $x(m)$	Odometry $y(m)$	Odometry $\theta(^{\circ})$	$x(m)$ real.	$y(m)$ real.	$\theta(^{\circ})$ real.
2	0,2	45	2,02	0,1	67,07
2	0.5	45	2,075	0,401	79,45
1	1	90	0,91	1,243	103,87
2	0	90	2,03	0,02	99,28
3	0.5	90	3,05	0,67	67,64
4	0	0	4,46	0,08	12,03
1	3	45	1,17	2,18	47,63
2	2	-45	2,04	1,95	-38,40
3	0	-90	3,05	-0,26	-53,31

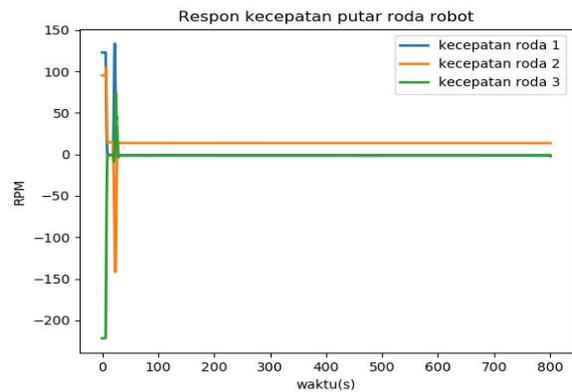
*Q. Pengujian Real Time Motion Planning yang telah dirancang*

Hasil yang dijadikan pembahasan berupa grafik respon system yang didapatkan dari pengujian robot secara *real time*. Berikut adalah hasil dari pengujian robot dalam mengikuti pergerakan robot lain



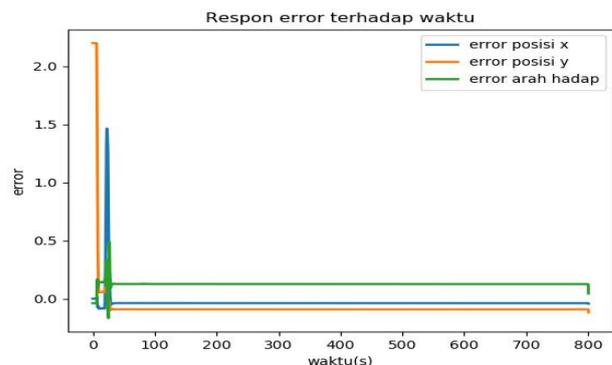
Gambar 15: Trajectory Robot secara *real time*.

Gambar 15 adalah trajectory *Defense Robot* terhadap *Throw Robot* secara *real time*. Pada pengujian secara *real time* ini dapat kita amati bahwa pergerakan trajectory dari *defense robot* tidak sebaik yang ada pada simulasi terutama pada saat robot berada pada posisi koordinat  $x = 3, y = 2$  hal itu disebabkan karena pada saat *Throw robot* berpindah posisinya dari koordinat  $x = 0, y = 4$  proses perpindahannya tidak benar – benar lurus sehingga menyebabkan data posisi yang dikirimkan kepada *Defense robot* menjadi tidak akurat dan mempengaruhi pergerakan maupun arah hadap *Defense robot* itu sendiri. Selanjutnya adalah data dari respon kecepatan motor yang dihasilkan pada saat pengujian secara *real time*.



Gambar 16: Respon kecepatan secara *real time*.

Dari Gambar 16 dapat kita amati bahwa motor bergerak dengan RPM tinggi saat robot berada di start awal menuju posisi tujuan dan perubahan kecepatan motor relatif terjadi dengan waktu yang cukup singkat tidak seperti yang ditunjukkan pada saat simulasi hal ini terjadi karena pada saat percobaan secara *real time* perpindahan *throw robot* juga berlangsung cukup cepat tidak seperti pada saat simulasi dimana posisi *throw robot* hanya diplot seperti biasa tanpa memperhitungkan masalah waktu.



Gambar 17: Error Posisi secara *real time*.



Gambar 17 adalah grafik hasil eror posisi yang dihasilkan dari pergerakan robot menuju posisi tujuannya. Dapat kita amati bahwa eror posisi x,y nya sangat kecil sekali sehingga bias dikatakan posisi *defense robot* telah sesuai dengan tujuan yang diinginkan. Namun terdapat nilai eror yang cukup besar untuk arah hadap dari *defense robot* itu sendiri hal ini dapat disebabkan oleh pada saat pengujian sensor kompas dengan posisi yang sudah pasti ternyata masih terdapat eror arah hadap yang cukup besar sehingga berpengaruh pada saat pengujian *motion planning* secara real time.

#### V. PENUTUP

Dari penelitian yang telah dilakukan, didapatkan kesimpulan sebagai berikut,

1. Hasil pengujian secara *real time* menunjukkan *Defense Robot* mampu mengikuti pergerakan *Throw Robot* termasuk dalam mengikuti arah hadapnya
2. Dari hasil pengujian secara *real time* posisi dan arah hadap akhir dari *Defense Robot* tidak benar – benar simetris atau linier yang disebabkan dari data posisi dan arah hadap yang dikirimkan oleh *Throw Robot* tidak presisi.

3. *Defense Robot* mampu mempertahankan posisi akhirnya yaitu koordinat y dari *Throw Robot* dikurangi sebesar dua meter dan apabila *Defense Robot* digerakkan secara paksa maka ia akan kembali ke posisi sebelumnya yaitu koordinat y dari *Throw Robot* dikurangi sebesar dua meter.

#### REFERENSI

- [1] Siradjuddin, Indrazno, Gilang Al Azhar, Anggit Murdani, & Mukhamad Luqman Muttaqin Faizin. 2020. Desain dan pemodelan kontrol kinematik pergerakan robot beroda dengan menggunakan 6 roda omni-wheels. *JURNAL ELTEK* Vol. 18 No.1, April 2020
- [2] Putra, Riza Lukmana, Hari Kurnia Safitri, dan Indrazno Siradjuddin. 2015. Pemodelan dan Analisis Sistem Kontrol Kinematik Omni Directional pada Robot. *Jurnal Elektronika Otomasi Industri*, Vol. 2 No. 1, Oktober 2015
- [3] Asrofi, Anan, Achmad Komarudin, dan Agus Pracoyo. 2014. Navigasi Robot Mobil 3wd Omni wheeled dengan Metode Odometri. *Jurnal Elektronika Otomasi Industri*, Vol. 1 No. 1, Mei 2014
- [4] P. Coelho and U. Nunes, "Path-following control of mobile robots in presence of uncertainties," in *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 252-261, April 2005, doi: 10.1109/TRO.2004.837240.
- [5] Syam, Rafiuddin, Jumaddil Hair. 2016. Desain Kerjasama Mobile Manipulator Robot. *Jurnal Otomasi Kontrol dan Instrumentasi*, Vol. 8 No. 2, 2016
- [6] ABU Asia Pasific Robot Contest 2021. *Throwing Arrows into Pots* (<http://www.robcon2021.com/>) Diakses pada 17 November 2020.

