

Kontrol Posisi Sistem Pergerakan *Mobile Robot* Berbasis Analisa Kinematik

Riemza Zamronnan, Totok Winamo, Eka Mandayatma

Abstrak — *Mobile robot* adalah sebuah robot yang mempunyai ciri khas dengan *actuator* berupa roda untuk menggerakkan seluruh badan robot, sehingga *mobile robot* tersebut mampu berpindah-pindah posisi dari satu titik koordinat ke titik koordinat lainnya. Ada satu permasalahan terkait perancangan robot bergerak pada perlombaan KRAI 2019 (Kontes Robot Aburobocon Indonesia) ini yaitu bagaimana cara mengetahui pergerakan posisi robot pada suatu waktu tertentu yang dinamakan dengan sistem lokalisasi robot dan dapat menghindari rintangan atau tiang, yaitu dengan cara mengontrol posisi sebuah *mobile robot*. Pengujian yang dilakukan pada kontrol posisi *mobile robot* berdasarkan metode navigasi berupa pembacaan sensor *rotary encoder*. Agar *mobile robot* bisa melakukan manuver pada target dengan jarak tertentu yang didapatkan dengan cara percobaan *trial and error*. Setelah melakukan percobaan didapat nilai *error* rata-rata koordinat X sebesar 1,37% dan koordinat Y sebesar 0,76% yang tepat bermanuver ketitik yang dituju.

Kata kunci : Kinematik, Kontrol Posisi, Rotary Encoder

I. PENDAHULUAN

Perkembangan teknologi dalam bidang robot semakin pesat seiring majunya pola pikir manusia yang semakin cerdas dan modern.

Salah satu ilmu pengetahuan dan teknologi yang digunakan untuk mengembangkan ide yaitu dari ilmu elektronika khususnya bidang ilmu robotika

Robot merupakan perangkat atau *hardware* yang bisa melakukan suatu tugas tertentu yang dapat dikontrol secara otomatis (kecerdasan buatan). Salah satu jenis robot itu adalah *mobile robot*. *Mobile robot* adalah sebuah robot yang mempunyai *actuator* yang berciri khas berupa roda untuk menggerakkan seluruh badan robot tersebut, sehingga robot tersebut dapat berpindah-pindah posisi dari titik satu ke titik yang lainnya. Ada satu permasalahan terkait perancangan

robot bergerak pada perlombaan KRAI 2019 (Kontes Robot Aburobocon Indonesia) ini yaitu bagaimana cara mengetahui pergerakan posisi robot pada suatu waktu tertentu yang dinamakan dengan sistem lokalisasi robot dan dapat menghindari rintangan atau tiang.

Pada bidang *mobile robot*, agar robot dapat bernavigasi secara tepat dan cepat dapat menggunakan metode navigasi berupa pembacaan sensor *rotary encoder* pada motor. Metode ini memiliki kelebihan dibandingkan dengan metode navigasi yang lain misalnya metode menggunakan pembacaan garis atau merabah dinding untuk jalur pergerakan robot.

Selain sistem navigasi, analisa gerak yang digunakan salah satunya adalah analisa kinematik. Analisa ini memanfaatkan hasil data dari sensor yang aktif untuk memperkirakan perubahan posisi robot dari waktu ke waktu sepanjang robot bergerak. Agar *mobile robot* bisa melakukan manuver pada target dengan jarak tertentu yang didapatkan dengan cara percobaan *trial and error*.

Berdasarkan permasalahan diatas, supaya robot mampu mencapai titik yang dituju melalui perhitungan pulsa roda untuk memperkirakan perubahan posisi robot, robot dapat menghindari rintangan atau tiang dan data yang didapatkan lebih akurat dalam pembacaan posisi titik koordinat dalam waktu *real-time*.

II. TINJAUAN PUSTAKA

Pada perancangan alat kontrol posisi *Mobile robot* ini menggunakan Mikrokontroler Arduino dan Rasphi sebagai kontroler utama dengan Metode navigasi pembacaan sensor *rotary encoder* dan ada beberapa komponen yang mendukung sistem agar dapat berjalan dengan baik. Komponen pendukung yang dibutuhkan pada alat ini diantaranya adalah sensor *rotary encoder* akan membaca putaran motor yang berfungsi untuk mengetahui berapa jarak tempuh *mobile robot*. Berikut adalah penjelasan dari beberapa komponen yang digunakan dalam perancangan alat kontrol posisi *mobile robot*.

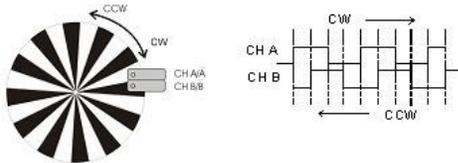
2.1 Sensor Rotary Encoder

Rotary encoder, digunakan untuk mengubah gerakan linear atau putaran menjadi sinyal digital, dimana sensor

Riemza Zamronnan adalah mahasiswa D4 Teknik Elektronika Politeknik Negeri Malang, email : riemzazm@gmail.com
Totok Winamo dan Eka Mndayatma adalah dosen Jurusan Teknik Elektro Politeknik Negeri Malang

putaran memonitor gerakan putar dari suatu alat. Sensor ini biasanya terdiri dari 2 lapis jenis penyandi, yaitu; Pertama, Penyandi rotari tambahan (yang mentransmisikan jumlah tertentu dari pulsa untuk masing-masing putaran) yang akan membangkitkan gelombang kotak pada objek yang diputar. Kedua, Penyandi absolut (yang memperlengkapi kode binary tertentu untuk masing-masing posisi sudut).

Secara umum prinsip kerja *rotary encoder* dapat diilustrasikan seperti dalam gambar berikut ini:



Gambar 1 Prinsip Kerja Rotary Encoder

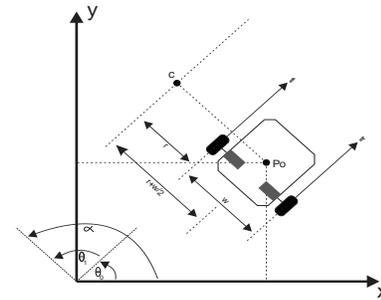
Dua buah sensor optis (chanel A/A dan chanel B/B) pendeteksi “hitam dan putih” digunakan digunakan sebagai acuan untuk menentukan arah gerakan. Searah jarum jam (clock-wise. CW atau berlawanan arah jarum jam (counter clock-wise. CCW). Sedangkan jumlah pulsa (baik A atau B) dapat dihitung (menggunakan prinsip counter) sebagai banyak langkah yang ditempuh. Dengan demikian arah gerakan dan posisi dapat dideteksi dengan baik oleh *rotary encoder*. Biasanya encoder dipasang segaris dengan poros (shaft) motor, gearbox, sendi atau bagian berputar lainnya. Beberapa tipe encoder memiliki poros berlubang (*hollow shaft encoder*) yang didesain untuk system sambungan langsung ke poros obyek dideteksi.

2.2 Mikrokontroler

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara tertentu. Disini digunakan mikrokontroler dengan tujuan agar dapat diprosesnya data input baik itu dari komponen ataupun dari hasil pembacaan sensor dan diolah serta dapat ditentukannya arah, tujuam, laju dari alat yang dikontrol.

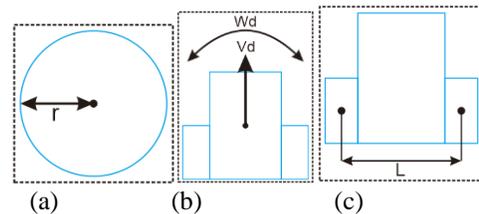
2.3 Metode Kinematik

Robot mobil tipe penggerak *differensial* memiliki 2 buah roda penggerak yang terpisah (kanan dan kiri). Kedua roda ini digerakkan oleh motor DC yang ditempatkan pada satu sumbu secara terpisah. Sehingga kedua roda ini berfungsi sebagai penggerak sekaligus sebagai kemudi robot mobil. Sebagai penyeimbang umumnya robot mobil ini dilengkapi juga dengan satu atau dua buah roda castor yang ditempatkan dibagian depan robot mobil tersebut.



Gambar 2 Skema Putar

Teknik klasik bagi robot beroda dalam menghitung posisinya adalah dengan menjejak lokasi robot tersebut melalui sebuah pengukuran secara kontinyu dari rotasi roda robot, seperti pada gambar 3



Gambar 3 (a) Jari-Jari Roda, (b) Kecepatan Linier dan Anguler Robot, (c) Jarak Titik Tengah Roda Kanan Sampai Kiri Robot

- 1) Keterangan :
 - a) r_r, r_l = jari-jari roda
 - b) L = jarak titik tengah roda kanan ke titik tengah roda kiri
 - c) J = Matrix Jacobian
 - d) V_d = kecepatan linier (lurus)
 - e) W_d = kecepatan anguler kanan / kiri (berbelok)
 - f) θ = theta (sudut)
 - g) $\Delta tick$ = jumlah lubang yang dibaca encoder
 - h) N = Maksimal jumlah lubang yang dibaca encoder dalam 1 putaran (360°)

Posisi diketahui mempunyai 3 titik yaitu x, y dan θ , sebagai berikut:

$$\text{Posisi} = [x \ y \ \theta] \rightarrow X \text{ atau shi}$$

Setelah posisi diketahui, turunan 1 dari posisi yaitu kecepatan, sebagai berikut:

$$\text{Kecepatan} = [\dot{x} \ \dot{y} \ \dot{\theta}] \rightarrow \dot{X} \text{ atau shu}$$

Mencari *Wheel rotation* kanan dan kiri (ω_r, ω_l), sebagai berikut:

$$\omega_r = 2\pi * r_r * \Delta tick_r / N \dots(1)$$

$$\omega_l = 2\pi * r_l * \Delta tick_l / N \dots(2)$$

2) Forward Kinematik

Pada forward kinematic ini mencari Wheel rotation yang akan digunakan untuk update posisi, sebagai berikut:

$$\omega = J^{-1} * \dot{Shu}$$

penjabaran dari rumus diatas :

$$\begin{bmatrix} \omega r \\ \dot{\omega} l \end{bmatrix} = \begin{bmatrix} \frac{r \cdot \cos \theta}{2} & \frac{r \cdot \cos \theta}{2}^{-1} \\ \frac{r \cdot \sin \theta}{2} & \frac{r \cdot \sin \theta}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} * \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \dots(3)$$

3) Invers Kinematik

Pada Invers Kinematik ini mencari pendekatan nilai error, dimana E = nilai error (selisih) dari titik yang dituju dikurangi titik sekarang. Berikut penjabaran dari rumus diatas bagian *Shu*:

a) Pendekatan Error 1 :

$$E = shi^* - shi$$

Dijabarkan menjadi, sebagai berikut:

$$E = \begin{bmatrix} x^*_{t-1} \\ y^*_{t-1} \\ \theta^*_{t-1} \end{bmatrix} - \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Turunan diatas menjadi,, sebagai berikut:

$$\dot{E} = -shi$$

b) Pendekatan error 2 :

$$E = e^{-\lambda t}$$

Turunan diatas menjadi, sebagai berikut:

$$\dot{E} = -\lambda * e^{-\lambda t}$$

$$\dot{E} = -\lambda * E$$

Rumus diatas dikembalikan kerumus awal menjadi, sebagai berikut:

$$\omega = J^{-1} * \dot{E}$$

$$\omega = \lambda * J^{-1} * (shi - shi) \dots(4)$$

4) Mencari Sudut Theta (θ)

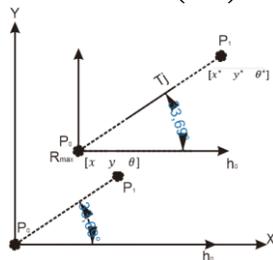
$$Tj = R_{max} * (P_1 - P_0)$$

penjabaran dari rumus diatas :

$$Tj = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}^{-1} * \begin{bmatrix} x_1 - x_0 \\ y_1 - y_0 \end{bmatrix} \dots(5)$$

Rumus diatas akan menemukan x dan y, hasil dari rumus dibawah dimasukan ke θ^* dirumus pendekatan Error 1:

$$\theta^* = \theta + \arctan(y/x) \dots(6)$$



Gambar 4 Skema Sudut

5) Update Posisi

Untuk robot yang dikendalikan secara differential, lokasi robot selalu diperbaharui secara tetap menggunakan rumus di bawah ini.

Update posisi disini posisi sebelumnya + posisi yang baru, sebagai berikut:

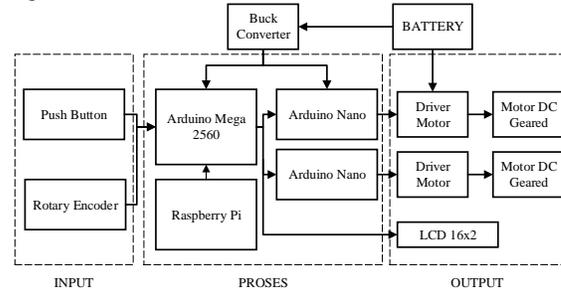
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x^*_{t-1} \\ y^*_{t-1} \\ \theta^*_{t-1} \end{bmatrix} + \begin{bmatrix} Vd * \cos \theta \\ Vd * \sin \theta \\ Wd \end{bmatrix} * dt \dots(7)$$

Dengan membangkitkan fungsi-fungsi di atas secara kontinyu maka akan didapatkan nilai X, Y dan θ setiap saat.

III. METODE PENELITIAN

3.1 Diagram Blok Sistem

Diagram blok sistem Analisa kinematik *mobile robot* dalam gambar berikut:



Gambar 6 Diagram Blok Sistem

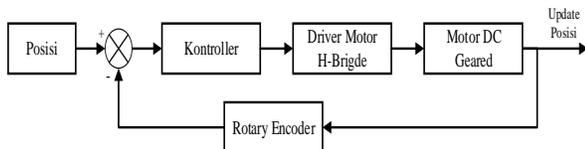
Prinsip Kerja

Robot dibuat untuk menentukan posisi berapa jarak antara robot dengan tiang yang telah ditentukan track atau lintasannya. Robot akan dapat mengetahui posisi dan bermanuver sesuai plan yang telah dipersiapkan sehingga robot dapat sampai pada target sesuai misi yang ada pada kontes robot Aburobocon 2019. Prinsip kerja mobile robot ini yaitu tahap awal menentukan jarak antara titik tengah robot. Kemudian menentukan jarak dari titik posisi tiang terhadap titik yang akan dituju. Titik yang pertama dituju berada pada titik (0,0). robot nantinya harus bergerak menuju titik yang dituju, yang dimasukkan diprogram invers kinematic yang berada di Mini Komputer (Rasberry pi).

Pada saat robot distart Raspi akan mengirim data hasil perhitungan kinematik ke Arduino Mega untuk dibaca dan dikelola, setelah Arduino Mega(Master) membaca lalu mengirim data ke Arduino Nano(slave) Arduino nano mengirim nilai PWM ke Driver motor. Saat motor berputar,Arduino Nano membaca data Sensor Rotary Encoder beberapa tick, lalu Arduino Nano mengirim ke Arduino Mega sehingga data dapat dikelola dan dikirim kembali keRaspi, dan diraspi akan dimonitor berapa X,Y dan Z yang dibutuhkan untuk mencapai ketitik yang dituju, Perhitungan dan pembacaan dari sensor rotary encoder akan diakumulasi. Hasil dari akumulasi tersebut dijadikan nilai PWM untuk mengatur pergerakan motor.[7] Akan diketahui berapa besar sudut yang dibutuhkan robot untuk berotasi tegak lurus terhadap titik yang dituju setelah mengetahui sudut yang dituju, maka mobile robot akan berotasi berapa besar sudut yang dibutuhkan dari hasil perhitungan kinematik. Ketika robot sudah berotasi, robot akan bergerak maju kedepan dengan kecepatan tertentu semakin besar nilai Error maka robot akan bergerak semakin cepat, apabila robot mendekati nilai error, robot akan bergerak perlahan

Proses perhitungan jarak dan sudut ini sampai pada titik tujuan terakhir dengan menggunakan analisa yang sama. yang dikendalikan oleh driver H-bridge.

3.2 Diagram Blok Kontrol



Gambar 7 Diagram Blok Kontrol

Penjelasan Diagram blok kontrol

Dari gambar diatas dapat diuraikan sebagai berikut:

1. Mikrokontroler digunakan untuk memproses perhitungan nilai yang sudah terbaca dan mengolah data sensor, kemudian akan dikonversikan menjadi sinyal PWM.
2. Driver Motor H-Bridge IBT-2 digunakan untuk mengatur kecepatan motor DC dengan cara mengubah sinyal PWM yang dihasilkan oleh mikrokontroler.
3. Motor DC digunakan untuk menggerakkan roda.
4. Sensor Rotary Encoder digunakan untuk menghitung putaran motor dengan tujuan mengetahui posisi robot, kemudian akan dikirim dan diproses oleh mikrokontroler.
5. Kecepatan motor (rpm) merupakan hasil akhir yang dikendalikan.
6. Output dari sistem adalah posisi *mobile robot* yang diproses melalui pembacaan sensor rotary encoder dan disesuaikan dengan nilai set point yang ditentukan oleh titik target.

3.3 Perhitungan Kinematik

Diket : $r = 0.06$ //jari-jari roda
 $L = 0.2$ //jarak tengah roda kanan dan kiri
 $\lambda = 10$ //lamda
 $N = 135$ //maxtick $P_0 = [0.2, 1.3, 5]$
 $dt = 0,1$ //waktu $P_1 = [2, 2, 90]$

❖ $T_j = R_{max}^{-1} * (P_1 - P_0)$

$$T_j = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} * \begin{bmatrix} x_1 & - & x_0 \\ y_1 & - & y_0 \end{bmatrix}$$

$$T_j = \begin{bmatrix} \cos 5 & -\sin 5 \\ \sin 5 & \cos 5 \end{bmatrix}^{-1} * \begin{bmatrix} 2 & - & 0.2 \\ 2 & - & 1.3 \end{bmatrix}$$

$$T_j = \begin{bmatrix} 0.996 & -0.087 \\ 0.086 & 0.996 \end{bmatrix}^{-1} * \begin{bmatrix} 1.8 \\ 0.7 \end{bmatrix}$$

$$T_j = \begin{bmatrix} 0.996 & 0.087 \\ -0.086 & 0.996 \end{bmatrix} * \begin{bmatrix} 1.8 \\ 0.7 \end{bmatrix}$$

$$T_j = \begin{bmatrix} 1.854 \\ 0.540 \end{bmatrix}$$

$$\theta^* = \arctan \left[\frac{T_j y}{T_j x} \right] + (5 * \pi / 180)$$

$$\theta^* = 0.37 \text{ rad}$$

❖ $w = J * E$

$$J = \begin{bmatrix} \cos(5) & 0 \\ \sin(5) & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0.06/2 & 0.06/2 \\ 0.06/0.2 & -0.06/0.2 \end{bmatrix}$$

$$J = \begin{bmatrix} 0.0298 & 0.0298 \\ 0.002 & 0.002 \\ 0.3 & 0.3 \end{bmatrix}^{-1}$$

$$J = \begin{bmatrix} 16.6 & 1.45 & 1.666 \end{bmatrix}$$

$er = x_{des} - x_{pos}$

$$er = \begin{bmatrix} 2 \\ 2 \\ 0.087 \end{bmatrix} * \begin{bmatrix} 1.2 \\ 0.3 \\ 0.37 \end{bmatrix} = \begin{bmatrix} 1.8 \\ 0.7 \\ 0.283 \end{bmatrix}$$

$w = \lambda * J * er$

$$\omega = 10 * \begin{bmatrix} 16.6 & 1.45 & 1.666 \end{bmatrix} * \begin{bmatrix} 1.8 \\ 0.7 \\ 0.283 \end{bmatrix}$$

$$\omega = \begin{bmatrix} 4.727 \\ -4.727 \end{bmatrix}$$

❖ Mencari Vd dan Wd

$$Vd = \frac{4.727 * 0.06 + (-4.727 * 0.06)}{2} = 2.382 e^{-18}$$

$$Wd = \frac{4.727 * 0.06 - (-4.727 * 0.06)}{0.2} = 2.836 e^{-18}$$

$$x_{dot}[0,0] = vd * \cos(5) * dt = 1.382 e^{-16}$$

$$x_{dot}[1,0] = vd * \sin(5) * dt = 1.209 e^{-16}$$

$$x_{dot}[2,0] = wd * dt = 0.2836$$

❖ $x_{pos} = x_{pos} + x_{dot}$

$$x_{pos} = \begin{bmatrix} 0.2 \\ 1.3 \\ 0.087 \end{bmatrix} + \begin{bmatrix} 1.382 \\ 1.209 \\ 0.283 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 1.3 \\ 0.37 \end{bmatrix}$$

Jadi, robot harus berputar sebesar 0.37 rad atau 21.25°

3.4 Perancangan Mekanik

Pada perancangan mekanik ini akan dibahas bentuk fisik mobile robot. Pada mobile robot otomatis ini memerlukan motor PG-45 yang akan digunakan untuk membantu pergerakan mobile robot tersebut dari 1 titik ke titik yang lain.

Spesifikasi Alat

Dimensi

- Panjang : 40 cm
- Lebar : 40 cm
- Tinggi : 10 cm

Jenis Driver : Modul Driver motor IBT-2

Jenis Motor : Motor DC Gearbox

Jenis Processor : Rasberry PI 3b, Arduino Mega dan Arduino nano

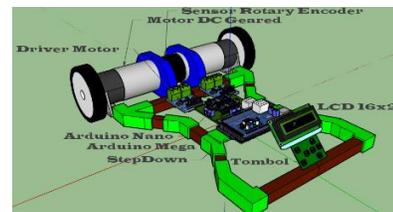
Jenis LCD : 16x2

Jenis Sensor : Sensor rotary encoder

Bahan casing/base : PLA, Alumunium

Kapasitas beban : ±6 kg

Sumber Daya : Baterai Lipo 3S 6200mAH



Gambar 5 Mekanik Keseluruhan

IV. HASIL DAN ANALISA

4.1 Pengujian LCD.

Pengujian pada *display* LCD 16x2 dilakukan untuk memastikan apakah lcd dapat menampilkan karakter sesuai dengan program yang didownloadkan pada arduino dengan bahasa C.[9]



Gambar 8 Pengujian LCD

Berdasarkan gambar 8 LCD dapat menampilkan karakter yang dibuat pada program yaitu “BISMILLAH” pada baris pertama dan “0xff” pada baris kedua sehingga LCD layak digunakan pada robot.

4.2 Pengujian Sensor Rotary Encoder

Pengujian pada sensor *rotary encoder* dilakukan untuk mengetahui data yang didapatkan oleh *rotary encoder* ketika terjadi perputaran pada motor DC sesuai dengan program yang telah didownloadkan pada kontroler Arduino Nano. Sensor *rotary encoder* harus bisa mendeteksi arah putaran dari motor DC, jika motor berputar searah jarum jam, maka data *rotary encoder* akan bertambah, jika motor berputar berlawanan dengan arah jarum jam, maka data *rotary encoder* akan berkurang. Berikut ini program untuk pembacaan *rotary encoder* dan hasil percobaan.



Gambar 9 Data Pengujian Motor Berputar Searah Jarum Jam

Gambar 9 merupakan data pembacaan *rotary encoder*, ketika motor DC diputar searah dengan arah jarum jam. Data menunjukkan ketika motor diputar searah jarum jam, data pembacaan sensor akan bertambah. Ketika motor diputar sebanyak 1 kali putaran penuh, data maksimal yang didapatkan dari satu kali putaran motor adalah 60 pulsa HIGH.



Gambar 10 Data Pengujian Motor Berputar Berlawanan Arah Jarum Jam

Gambar 10 merupakan data pembacaan *rotary encoder*, ketika motor DC diputar berlawanan arah dengan arah jarum

jam. Data menunjukkan ketika motor diputar berlawanan arah jarum jam, data pembacaan sensor akan berkurang. hal ini menunjukkan bahwa pembacaan *rotary encoder* tetaplah stabil untuk 1 kali putaran baik searah jarum jam maupun berlawanan dengan arah jarum jam.

4.3 Pengujian Driver Motor.

Berikut adalah tabel hasil pengujian rangkaian driver motor IBT-2:

Tabel 1 Pengujian Kinerja Driver IBT-2

INPUT1	INPUT2	PWM	Output
HIGH	LOW	HIGH	Maju
LOW	HIGH	HIGH	Mundur
LOW	LOW	HIGH	Berhenti dengan Pengereman Mundur
HIGH	HIGH	LOW/HIGH	Berhenti dengan Pengereman Maju
LOW	LOW	LOW	Berhenti tanpa ada pengereman
HIGH	LOW	LOW	Berhenti tanpa ada pengereman
LOW	HIGH	LOW	Berhenti tanpa ada pengereman

Setelah melakukan pengujian pada *driver* motor, hasil yang didapatkan dari pengujian tersebut *driver* bekerja sesuai dengan keinginan. Untuk pengontrol *driver* diberikan logika sesuai dengan data tabel 4.5. Ketika logika dari arduino diberikan secara bergantian, yaitu ketika input 1 diberi HIGH, dan input 2 diberi LOW, atau sebaliknya, motor akan berputar ketika nilai PWM adalah HIGH juga. Jika PWM diberi LOW, motor akan berhenti tanpa melakukan pengereman. Ketika input diberikan logika LOW semua, dengan kondisi PWM adalah HIGH, maka motor yang dikendalikan akan berhenti dengan melakukan pengereman ketika motor diputar mundur. Dan ketika input diberi HIGH semua, dan tidak diperhatikan nilai PWM yang diberikan, maka motor akan berhenti dengan melakukan pengereman ketika motor diputar maju. [10]

Tabel 2 Pengujian Kecepatan Motor

No.	PWM		RPM	
	Kanan	Kiri	Kanan	Kiri
1	100	100	50	53
2	200	200	101	102
3	300	300	149	151
4	400	400	200	202
5	500	500	250	252
6	600	600	300	302
7	700	700	352	352
8	800	800	404	405

Dari pengujian kecepatan motor menggunakan PWM 10bit, maksimal nilai PWM yaitu 1023, setelah melakukan pengujian, diketahui motor kanan lebih pelan dari motor kiri dan dapat diketahui bahwa nilai rpm terukur tachometer berbeda setiap motor meskipun input pwm sama. Hal ini dapat dipengaruhi oleh beberapa faktor seperti, kualitas setiap komponen yang ada di driver motor berbeda walaupun sama

jenisnya. Faktor kondisi motor yang berbeda, baik usia ataupun kerusakan yang terjadi di dalamnya

Dari pengujian tersebut, membuktikan bahwa driver H-Bridge IBT-2 mampu dioperasikan secara *fullbridge*, karena motor yang disambungkan pada driver tersebut, bisa dioperasikan tidak hanya untuk maju dan mundur saja, melainkan bisa dioperasikan dalam hal pengereman arah putar motor.

4.4 Pengujian Sistem

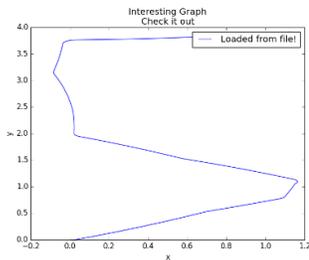
Dalam pengujian ini, robot dijalankan pada simulasi track yang ada dan mengetest akan sistem keseluruhan dari robot.

Pada pengujian pertama dilakukan 2 kali percobaan dengan titik awal koordinat (0,0) kemudian menentukan titik koordinat selanjutnya, sebagai berikut:

- 1) Titik koordinat (1.2, 0.8)
- 2) Titik koordinat (1.2, 1.2)
- 3) Titik koordinat (0.0, 2.0)
- 4) Titik koordinat (0.0, 3.8)
- 5) Titik koordinat (1.2, 3.8)

Dari titik tujuan diatas *mobile robot* akan bergerak dengan kecepatan tertentu dan membaca rotary untuk mengetahui nilai posisi *mobile robot*, dari data yang dibaca oleh rotary encoder, akan dikirim kelaptop untuk memonitoring data tersebut, setelah selesai sampai ketitik yang terakhir, maka terbentuklah grafik yang menghasilkan sebuah data sebagai berikut:

1. Pengujian Pertama Data ke-1



Gambar 11 Pengujian Pertama Data ke-1

Dari data gambar 11 percobaan pertama robot melalui titik koordinat sebagai berikut:

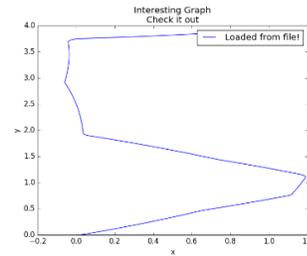
- 1) Titik koordinat (1.16, 0.85)
- 2) Titik koordinat (1.17, 1.2)
- 3) Titik koordinat (0.0, 1.98)
- 4) Titik koordinat (0.04, 3.8)
- 5) Titik koordinat (1.18, 3.81)

Tabel 3 Hasil Pengujian Pertama Data ke-1

Percobaan	Ideal		Real		Error (%)	
	X	Y	X	Y	X	Y
1	1,2	0,8	1,16	0,85	3,45	5,88
2	1,2	1,2	1,17	1,2	2,56	0
3	0	2	0	1,98	0	1
4	0	3,8	0,04	3,8	0	0

5	1,2	3,8	1,18	3,85	1,69	1,3
---	-----	-----	------	------	------	-----

2. Pengujian Pertama Data ke-2



Gambar 12 Pengujian Pertama Data ke-2

Dari data gambar 12 percobaan kedua robot melalui titik koordinat sebagai berikut:

- 1) Titik koordinat (1.17, 0.82)
- 2) Titik koordinat (1.19, 1.2)
- 3) Titik koordinat (0.0, 1.98)
- 4) Titik koordinat (0.0, 3.8)
- 5) Titik koordinat (1.18, 3.79)

Tabel 4 Hasil Pengujian Pertama Data ke-2

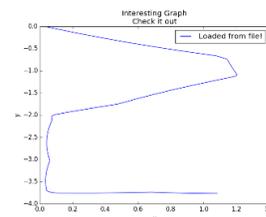
Percobaan	Ideal		Real		Error (%)	
	X	Y	X	Y	X	Y
1	1,2	0,8	1,17	0,82	2,56	2,44
2	1,2	1,2	1,19	1,2	0,84	0
3	0	2	0	1,98	0	1
4	0	3,8	0,04	3,8	0	0
5	1,2	3,8	1,18	3,79	1,69	0,3

Dari pengujian pertama percobaan pertama dan kedua menggunakan titik koordinat tujuan yang sama, akan tetapi dari pengujian diatas data kedua lebih baik dari data pertama yang dikarenakan data kedua lebih mendekati titik koordinat yang dituju daripada data pertama.

Pada pengujian kedua dilakukan 2 kali percobaan dengan titik awal koordinat (0,0) kemudian menentukan titik koordinat selanjutnya, sebagai berikut:

- 1) Titik koordinat (1.2, -0.8)
- 2) Titik koordinat (1.2, -1.2)
- 3) Titik koordinat (0.0, -2.0)
- 4) Titik koordinat (0.0, -3.8)
- 5) Titik koordinat (1.2, -3.8)

3. Pengujian kedua Data ke-1



Gambar 13 Pengujian Kedua Data ke-1

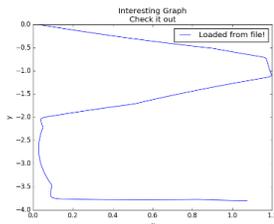
Dari data gambar 13 percobaan pertama robot melalui titik koordinat sebagai berikut:

- 1) Titik koordinat (1.12, -0.86)
- 2) Titik koordinat (1.19, -1.2)
- 3) Titik koordinat (0.14, -1.96)
- 4) Titik koordinat (0.08, -3.79)
- 5) Titik koordinat (1.14, -3.8)

Tabel 5 Hasil Pengujian Kedua Data ke-1

Percobaan	Ideal		Real		Error(%)	
	X	Y	X	Y	X	Y
1	1,2	-0,8	1,18	-0,9	1,69	6,98
2	1,2	-1,2	1,19	-1,2	0,84	0
3	0	-2	0,14	-2	0	2
4	0	-3,8	0,08	-3,8	0	0,3
5	1,2	-3,8	1,14	-3,8	5,26	0,26

4. Pengujian kedua Data ke-2



Gambar 14 Pengujian kedua data ke-2

Dari data gambar 14 percobaan kedua robot melalui titik koordinat sebagai berikut:

- 1) Titik koordinat (1.17, -0.81)
- 2) Titik koordinat (1.19, -1.2)
- 3) Titik koordinat (0.04, -2.03)
- 4) Titik koordinat (0.08, -3.81)
- 5) Titik koordinat (1.16, -3.8)

Tabel 6 Hasil Pengujian Kedua Data ke-2

Percobaan	Ideal		Real		Error(%)	
	X	Y	X	Y	X	Y
1	1,2	-0,8	1,17	-0,8	2,56	1,23
2	1,2	-1,2	1,19	-1,2	0,84	0
3	0	-2	0,04	-2	0	1,48
4	0	-3,8	0,08	-3,8	0	0,26
5	1,2	-3,8	1,16	-3,8	3,45	0

Dari pengujian percobaan pertama dan kedua menggunakan titik koordinat tujuan yang sama, akan tetapi dari pengujian diatas data kedua lebih baik dari data pertama yang dikarenakan data kedua lebih mendekati titik koordinat yang dituju daripada data pertama.

Sistem respon pembacaan sensor rotary encoder, serta sistem pengolahan data yang diberikan memperoleh hasil data :

- a. Saat robot dijalankan pada min pwm 200 pada motor kanan dan min pwm 200 pada motor kiri maju, maka robot akan maju dan bergerak lambat.
- b. Saat robot dijalankan pada max pwm 700 pada motor kanan dan max pwm 700 pada motor kiri maju, maka robot akan maju dan bergerak cepat, dan terkadang menimbulkan slip pada roda.
- c. Disaat robot maju dan membaca sensor rotary encoder maka robot akan secara melakukan perhitungan yaitu dengan odometer yang bertujuan menghitung nilai posisi yang dituju dengan posisi yang sekarang.
- d. Pada saat robot sudah menemukan titik yang dituju robot akan berhenti akan tetapi jika robot belum menemukan titik yang dituju robot akan terus bergerak hingga data x dan y sesuai titik yang diinginkan.
- e. Didapatkan nilai – nilai lamda, min PWM dan max PWM yang tepat untuk melakukan manufer pada target dengan jarak tertentu yang sudah diatur ketitik yang dituju dengan respon waktu rata-rata menuju 1 titik target yaitu 6 detik (dapat dilihat digambar 11, 12, 13, dan 14)
- f. Ketika lamda diberi nilai kurang robot menjadi kurang peka terhadap pembacaan nilai error sehingga robot berada jauh ketitik koordinat yang dituju, akan tetapi ketika lamda diberi nilai yang tepat , robot menjadi lebih peka terhadap pembacaan nilai error sehingga robot berada tidak terlalu jauh ketitik koordinat yang dituju.

Penyebab terjadinya error pada pengujian diatas :

- a. Terjadinya slip pada roda yang menyebabkan meleceng dari titik koordinat yang dituju.
- b. Kecepatan motor kanan dan kiri berbeda yang menyebabkan *mobile robot* berbelok ketika berjalan lurus.
- c. Faktor kondisi motor yang berbeda, baik usia ataupun kerusakan yang terjadi di dalamnya.
- d. Faktor kondisi driver motor yang berbeda V(tegangan) output pada driver motor 1 dan 2.

V. PENUTUP

5.1 Kesimpulan

Dari hasil Pengujian, Pengamatan dan Analisa dapat diperoleh beberapa kesimpulan. Dengan tujuan utama penganalisa *mobile robot* dalam hal mengontrol posisi menggunakan analisa kinematik.

- a. Dari data pengujian sensor *rotary encoder* Data menunjukkan ketika motor diputar searah jarum jam, data pembacaan sensor akan bertambah dan data menunjukkan ketika motor diputar berlawanan arah jarum jam, data pembacaan sensor akan berkurang. hal ini menunjukkan bahwa pembacaan *rotary encoder* tetaplh stabil untuk 1 kali putaran baik searah jarum jam maupun berlawanan dengan arah jarum jam.
- b. Dari data percobaan 1 diperoleh nilai error dengan rata-rata koordinat X 1,54% dan koordinat Y 1,2%.
- c. Dari data percobaan 2 diperoleh nilai error dengan rata-rata koordinat X 1,02% dan koordinat Y 0,2%.
- d. Dari data percobaan 3 diperoleh nilai error dengan rata-rata koordinat X 1,56% dan koordinat Y 1%.

- e. Dari data percobaan 4 diperoleh nilai error dengan rata-rata koordinat X 1,37% dan koordinat Y 0,6%.

Dalam percobaan 1, percobaan 2, percobaan 3, dan percobaan 4, diperoleh tingkat error posisi *mobile robot* rata-rata dari koordinat X sebesar 1,37%, dan koordinat Y sebesar 0,76%, nilai error dengan rata – rata tersebut maka *mobile robot* dapat dikatakan akurat atau tepat ketitik yang dituju.

5.2 Saran

- Selain dengan mengatur lamda untuk meminimalisir error pada gerakan robot, bisa dilakukan dengan cara lain yaitu dengan cara mengatur kecepatan putaran.
- Selain menggunakan sensor rotary encoder untuk mengetahui titik koordinat robot, dapat ditambahkan sebuah sensor kompas.
- Agar pembacaan lebih *real time* dengan Raspberry, dapat digantikan dengan Mini PC.

DAFTAR PUSAKA

- [1] Pitowarno, 2006:68 – *Robotika : Disain, Kontrol, dan kecerdasan Buatan*
- [2] Lukman, Riza Putra. 2015. *Pemodelan dan Analisa Kontrol Kinematik Omnidirectional pada Robot*. Malang : Politeknik Negeri Malang.
- [3] Putra, Braheimy R. 2013. *Implementasi metode navigasi Dead Reckoning pada robot mobile pencari target dan penghindar halangan dengan menggunakan sensor kompas*. Jember : Universitas Jember
- [4] Kurniawan, R. Indra L. 2015. *Pengendalian Robot Mobil Pencari Target Dengan Kemampuan Menghindari Rintangan*. Semarang : Universitas Ponorogo.
- [5] Datasheet Raspberry Pi 3b, <http://raspberrypi.org>, diakses pada tanggal 30 April 2019.
- [6] Datasheet *Arduino Genuine Mega 2560*, <http://arduino.cc/en/Product/Datasheet-ArduinoMega2560.doc>, diakses pada tanggal 20 April 2019.
- [7] Datasheet *Motor PG-45*, <http://www.brontoseno.com/produk/pg45m775-geared-motor-series/>, diakses pada tanggal 26 April 2019.
- [8] Datasheet *IBT-2 full-H-Bridge Driver*, <http://labelelektronika.com/2016/02/high-current-motor-driver-ibt-2-arduino.html?m=1>, diakses pada tanggal 25 April 2019.
- [9] *Elektronika Dasar. TEORI LIQUID CRYSTAL DISPLAY*. (http://elektronika-dasar.web.id/wp-content/Teori_Liquid_Crystal_Display), diakses pada tanggal 18 April 2019.
- [10] PPPPTK BOE, teori kaidah tangan kanan, www.vedcmalang.com, diakses pada 21 April 2019