

KONTROL POSISI PADA SISTEM PERGERAKAN MOBILE ROBOT RODA MEKANUM MENGGUNAKAN KONTROL PID BERBASIS INVERS KINEMATIC

Arif Rahman Hakim^{*)}, Sumardi, and Munawar Agus Riyadi.

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro,
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia

^{*)}E-mail : arif.rh.eeundip@gmail.com

Abstrak

Robot merupakan alat yang dikembangkan dengan pesat dewasa ini. Salah satu jenis robot itu adalah *mobile robot*. Selama ini kebanyakan teknologi *mobile robot* menggunakan sistem pergerakan *differential drive*. Permasalahan utama dari sistem pergerakan *differential drive* adalah terbatasnya pergerakan robot sehingga kurang efektif dan efisien. Roda mekanum menjawab permasalahan tersebut karena dapat bergerak kesegala arah. Pada penelitian ini, dilakukan perancangan dan pembuatan *mobile robot* roda mekanum yang dapat menuju koordinat tertentu dengan kalkulasi *invers kinematic* dan kontrol PID. Persamaan *invers kinematic* mengubah kecepatan robot dalam V_x , V_y dan ω menjadi kecepatan masing-masing roda yaitu ω_1 , ω_2 , ω_3 , ω_4 dan arah putar roda. Sedangkan kontrol PID berfungsi untuk mempertahankan *setpoint* kecepatan tiap roda. Sensor yang digunakan sebagai *feedback* adalah sensor *encoder hall effect* yang sudah terpasang pada tiap motor DC. Metode yang digunakan untuk *tuning* parameter PID adalah metode *Ziegler Nichols 2*. Pada pengujian motor DC, nilai konstanta hasil *tuning* metode *Ziegler Nichols 2* adalah $K_p=0,325$, $T_i=0$ dan $T_d=0$. Hasil pengujian menunjukkan bahwa *mobile robot* roda mekanum dapat bergerak menuju titik koordinat yang diinginkan dengan metode *driving forward* dan *driving sideways*. Galat rata-rata *heading* dan jarak pada kecepatan 80 rpm adalah 13,75 derajat dan 35 cm, sedangkan pada kecepatan 200 rpm galat rata-ratanya adalah 12,25 derajat dan 38.5 cm.

Kata kunci : *Mobile robot roda mekanum, Invers kinematic, Kontrol PID, Koordinat*

Abstract

Robot is toll which developed very fast now. One of type of robot is mobile robot. So far, most mobile robot technology uses a system of differential movement of the drive. The main problem of the movement of the differential drive system is the limited movement of the robot so that not effective and efficient. Mekanum wheel answer to these problems because it can move in all directions. In this research, design and manufacture of mobile robot that can mekanum wheel towards specific coordinates by inverse kinematic calculation and PID control. Inverse kinematic equations change in the robot speed V_x , V_y and ω be the speed of each wheel is ω_1 , ω_2 , ω_3 , ω_4 and direction swivel wheels. While the PID control function to maintain the setpoint speed of each wheel. The sensor used as a feedback is a hall effect encoder sensor is mounted on each of the DC motor. The method used for PID parameter tuning method is Ziegler Nichols 2 method. In testing the DC motor, a constant value results of Ziegler Nichols tuning method 2 is $K_p = 0.325$, $T_i = 0$ and $T_d = 0$. The test results showed that mekanum wheeled mobile robot can move into the limelight the desired coordinates with driving forward and driving sideways method. Average error heading and distance at a speed of 80 rpm was 13.75 degrees and 35 cm, at a speed of 200 rpm while the average error was 12.25 degrees and 38.5 cm.

Keywords : *Mecanum wheel mobile robot, Inverse kinematic, PID control, Coordinates*

1. Pendahuluan

Pada kontes Robot Abu Indonesia (ABU) peraturan yang digunakan selalu berubah-ubah tiap tahun bergantung pada tuan rumah ABU Robocon. Dengan aturan yang berbeda – beda robot tetap memiliki tugas yang sama.

Robot harus dapat berpindah tempat pada arena perlombaan yang cukup luas untuk memindahkan obyek-obyek pada tempat yang telah ditentukan. Oleh karena itu, dibutuhkan sistem pergerakan yang tepat agar robot dapat menyelesaikan misi dengan cepat [1]. Selain itu peraturan juga menuntut untuk robot bergerak secara otomatis.

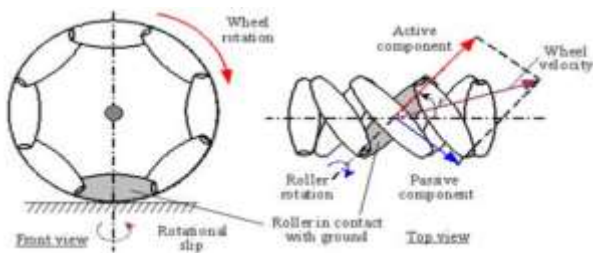
Selama ini kebanyakan teknologi *mobile robot* menggunakan sistem pergerakan *differential drive*. *Differential drive* merupakan salah satu pergerakan robot yang memanfaatkan kecepatan pada roda kiri dan kanannya. Permasalahan utama dari sistem pergerakan *differential drive* adalah terbatasnya pergerakan robot karena hanya bisa bergerak maju, mundur dan belok tetapi robot tidak mampu bergerak ke segala arah atau biasa disebut robot *non-holonomic* [2].

Roda mekanum merupakan teknologi pergerakan *holonomic*. Maka saat ini banyak *mobile robot* yang menggunakan teknologi ini. Namun dibutuhkan pengetahuan mengenai kinematika *mobile robot* yang menggunakan roda mekanum. Kinematika robot sangat penting untuk mendefinisikan arah dan kecepatan robot. Dari arah dan kecepatan masing-masing roda dihasilkan total vektor gaya dengan arah pergerakan robot yang diinginkan tanpa mengubah arah hadap robot maupun sudut kemiringan roda [3]. Dalam tugas akhir ini akan dirancang sebuah *mobile robot* roda mekanum yang dapat bergerak menuju titik koordinat tertentu. Selain itu, digunakan kontrol PID pada tiap motor *mobile robot* untuk mempertahankan *setpoint* kecepatan.

2. Metode

2.1. Roda Mekanum

Roda Mekanum disebut juga roda Ilon atau roda Swedia, diciptakan pada tahun 1973 oleh Bengt Ilon seorang insinyur Swedia [4]. Roda mekanum menggunakan desain roda yang dikelilingi oleh roller dengan membentuk sudut 45°. Gambar 1 menunjukkan bahwa sebuah roda mekanum memiliki 3 DOF (*degree of freedom*). DOF pertama adalah putaran roda, DOF kedua adalah rotasi roller, sedangkan DOF ketiga adalah slip rotasi terhadap sumbu vertikal yang melewati titik kontak dengan permukaan [5].



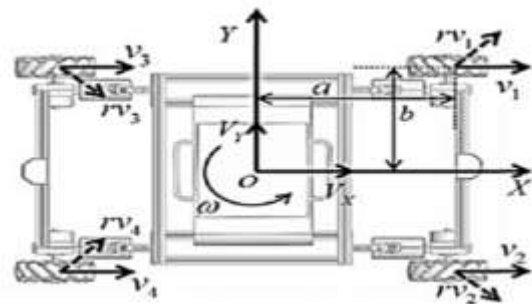
Gambar 1. Derajat kebebasan roda mekanum.

Desain mekanik dari roda mekanum memang lebih sulit bila dibandingkan dengan roda biasa. Namun roda mekanum menawarkan keunggulan dalam kestabilan pada beban tinggi. Biasanya, *mobile robot* yang menggunakan roda mekanum memiliki desain berbentuk persegi atau persegi panjang. Roda yang melekat pada kedua sisi memiliki roller dengan orientasi sudut +45° dan -45° [6].

Arah dan kecepatan masing-masing roda menghasilkan resultan gaya yang menerjemahkan pergerakan kecepatan linier (mobilitas) tanpa mengubah arah hadap robot ataupun perubahan sudut roda [7].

2.2. Kinematika Mobile Robot

Invers kinematic mengacu pada penggunaan persamaan kinematika robot untuk menentukan parameter bersama yang memberikan posisi yang diinginkan pada posisi akhir atau *efektor* [8]. Persamaan kinematik mendefinisikan beberapa bagian dari pergerakan robot, diantaranya kecepatan V_x , V_y dan ω menjadi persamaan kecepatan sudut masing-masing roda.



Gambar 2. Konfigurasi dan vektor pergerakan roda[1].

Konfigurasi parameter dan sistem kecepatan didefinisikan sebagai berikut:

- X, Y adalah sumbu koordinat kartesius;
- V_x, V_y [m/s] adalah kecepatan linier robot;
- v_i adalah kecepatan linier tiap roda terhadap sumbu X ;
- ω [rad/s] adalah kecepatan sudut robot;
- rv_i adalah kecepatan pasif roller pada roda i ;
- a adalah jarak roda depan dengan titik pusat robot;
- b adalah jarak roda samping dengan titik pusat robot.

Penomoran roda, jarak vertikal, horisontal, arah serta sudut resultan dari roller dan roda didefinisikan sebagai berikut ;

$$\begin{aligned} i &= 1,2,3,4 \\ a_i &= \{a, a, -a, -a\} \\ b_i &= \{b, -b, b, -b\} \\ a_i &= \{1, -1, -1, 1\} \end{aligned}$$

Setiap roda memiliki tambahan komponen kecepatan terhadap sumbu X dan Y akibat kecepatan rotasi ω ;

$$rv_{ix} = \omega \cdot -b \quad (1)$$

$$rv_{iy} = \omega \cdot a \quad (2)$$

Oleh karena itu, total kecepatan linier V_x dan V_y di setiap pusat roda menjadi ;

$$V_{ix} = V_x - \omega b \quad (3)$$

$$v_i + rv_i \cos(ai) = V_x - \omega b \quad (4)$$

$$V_{iy} = Vy + \omega a \quad (5)$$

$$rvi \sin(\alpha i) = Vy + \omega a \quad (6)$$

Untuk mencari kecepatan linier tiap roda maka substitusikan persamaan (4) dan (6) sehingga didapatkan persamaan (8)

$$vi = Vx - \omega b - \frac{Vy + \omega a}{\sin(\alpha i)} \cdot \cos(\alpha) \quad (7)$$

$$vi = Vx - \omega b - \frac{Vy + \omega a}{\tan(\alpha i)} \quad (8)$$

Langkah selanjutnya adalah substitusi konstanta ai , bi , αi kedalam persamaan (8) sehingga di dapatkan persamaan (9), (10), (11), (12) yang mempresentasikan kecepatan linier tiap roda.

$$\begin{aligned} v1 &= Vx - \omega b - [Vy + \omega a] \\ &= Vx - Vy - \omega a - \omega b \\ &= Vx - Vy - \omega(a + b) \end{aligned} \quad (9)$$

$$\begin{aligned} v2 &= Vx + \omega b - [-Vy - \omega a] \\ &= Vx + Vy + \omega a + \omega b \\ &= Vx + Vy + \omega(a + b) \end{aligned} \quad (10)$$

$$\begin{aligned} v3 &= Vx - \omega b - [-Vy + \omega a] \\ &= Vx + Vy - \omega a - \omega b \\ &= Vx + Vy - \omega(a + b) \end{aligned} \quad (11)$$

$$\begin{aligned} v4 &= Vx + \omega b - [Vy - \omega a] \\ &= Vx - Vy + \omega a + \omega b \\ &= Vx - Vy + \omega(a + b) \end{aligned} \quad (12)$$

Langkah terakhir adalah membawa kecepatan linier tiap roda kedalam kecepatan sudut. Karena $v = \omega \cdot r$ dimana r adalah radius roda mekanum, maka bila dibuat persamaan matriksnya menjadi :

$$\begin{bmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \\ \omega 4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & -1 & -(a+b) \\ 1 & 1 & (a+b) \\ 1 & 1 & -(a+b) \\ 1 & -1 & (a+b) \end{bmatrix} \begin{bmatrix} Vx \\ Vy \\ \omega \end{bmatrix} \quad (13)$$

Kemudian balikan persamaan (13) menjadi persamaan kecepatan *mobile robot*, sehingga diperoleh persamaan (14).

$$\begin{bmatrix} Vx \\ Vy \\ \omega \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(a+b)} & \frac{1}{(a+b)} & \frac{1}{(a+b)} & \frac{1}{(a+b)} \end{bmatrix} \begin{bmatrix} \omega 1 \\ \omega 2 \\ \omega 3 \\ \omega 4 \end{bmatrix} \quad (14)$$

Kecepatan longitudinal:

$$Vx(t) = (\omega 1 + \omega 2 + \omega 3 + \omega 4) \cdot \frac{r}{4} \quad (15)$$

Kecepatan transversal:

$$Vy(t) = (-\omega 1 + \omega 2 + \omega 3 - \omega 4) \cdot \frac{r}{4} \quad (16)$$

Kecepatan sudut:

$$\omega(t) = (-\omega 1 + \omega 2 - \omega 3 + \omega 4) \cdot \frac{r}{4(a+b)} \quad (17)$$

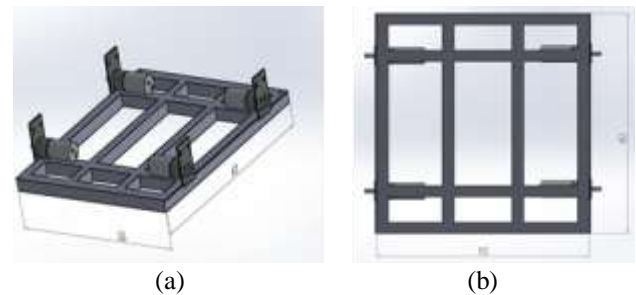
Resultan kecepatan dan arah *mobile robot* terhadap koordinat sumbu (X, Y) dapat dicari dengan persamaan (18) dan (19).

$$\beta = \tan^{-1} \left(\frac{Vy}{Vx} \right) \quad (18)$$

$$Vr = \sqrt{Vx^2 + Vy^2} \quad (19)$$

2.3. Perancangan Desain Mekanik

Perancangan mekanik atau rangka *mobile robot* harus dibuat sepresisi mungkin. Hal ini dapat berpengaruh besar pada sistem pergerakan robot. Desain robot haruslah sesuai dengan CG (*Center of Gravity*) sehingga pergerakan dalam mempertahankan posisi *body* tetap dalam kondisi datar tidak mengalami kesulitan.

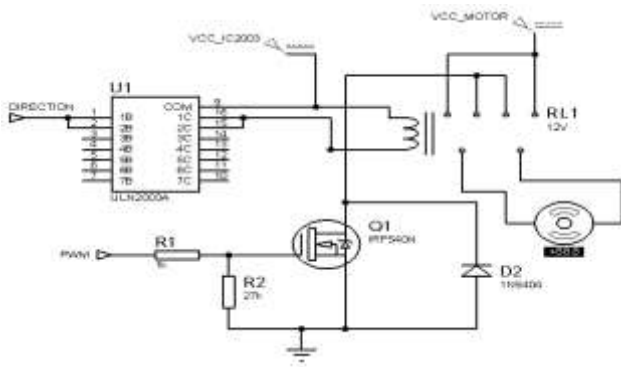


Gambar 3. Perancangan desain mekanik *mobile robot*, (a) Tampak samping, (b) Tampak bawah.

Dalam perancangan dan pembuatan, rangka *mobile robot* berbentuk persegi panjang dengan ukuran luar 600mm x 550mm. Sedangkan roda mekanum dihubungkan langsung dengan motor DC tanpa menggunakan *timing belt* karena antara roda mekanum dan motor DC terdapat *gear box*.

2.4. Perancangan Driver Motor

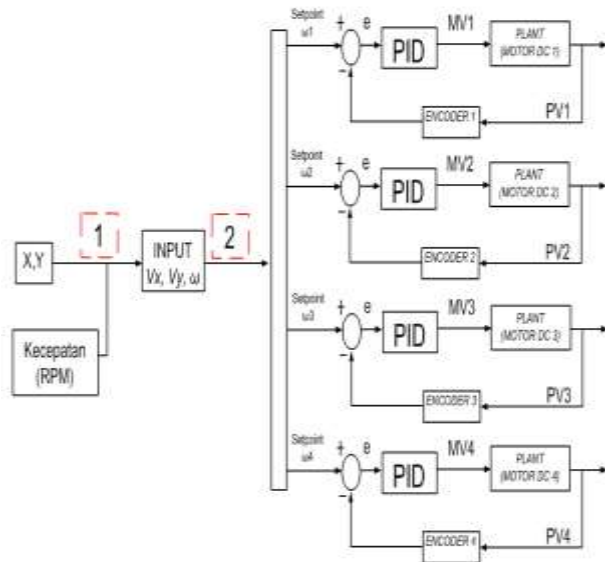
Rangkaian *driver* motor DC berfungsi sebagai antarmuka antara mikrokontroler dengan rangkaian daya dan pembalik arah putaran motor DC. Rangkaian *driver* motor pada *mobile robot* ini menggunakan relay DPDT sebagai gerbang tegangan *h-bridge* dan IC 2003 sebagai *driver* relay. Tipe MOSFET yang digunakan pada rangkaian *driver* motor ini adalah IRF 540. Sinyal PWM (*pulse width modulator*) dari mikrokontroler terhubung dengan kaki *gate* MOSFET untuk mengatur besar kecilnya tegangan *supply* yang diberikan ke motor DC. Supply motor adalah aki kering dengan tegangan 12V. Rangkaian *driver* motor DC dapat dilihat pada Gambar 4.



Gambar 4. Rangkaian driver motor.

2.5. Perancangan Kendali PID

Sebuah robot yang cerdas tentunya memiliki kondisi untuk mempertahankan *setpoint*. Dalam perancangan *mobile robot* ini kendali yang digunakan adalah kontrol PID.



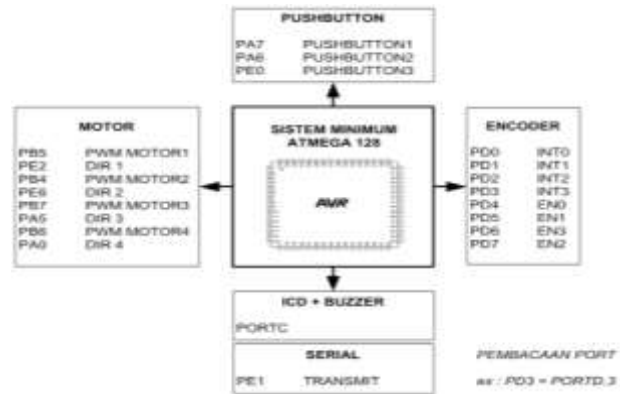
Gambar 5. Diagram blok sistem kontrol posisi

Dari gambar 5 dapat dilihat bahwa Sinyal umpan balik untuk kontrol PID berasal dari pembacaan sensor *encoder*. Selisih antara nilai *setpoint* dengan pembacaan *encoder* akan dijadikan masukan kendali PID. Keluaran dari kontrol PID ini akan digunakan untuk mengatur kecepatan dan arah putar penggerak berupa motor DC. Dalam perancangan *mobile robot*, kendali PID untuk masing-masing motor terpisah namun nilai parameter kontrol *proporsional* (K_p), *integral* (T_i), dan *derivatif* (T_d) sama karena pertimbangan karakteristik dan spesifikasi motor yang hampir sama.

2.6. Perancangan Mikrokontroler

Rangkaian sistem minimum mikrokontroler ATmega128 berfungsi untuk memproses data keseluruhan dari sistem

yang telah dibuat. Alokasi *port* pada sistem minimum mikrokontroler ATmega128 dapat dilihat pada Gambar 6.



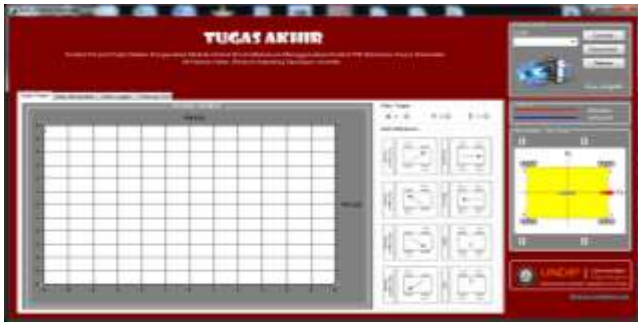
Gambar 6. Alokasi port pada mikrokontroler ATmega128.

2.7. Perancangan Perangkat-Lunak

Perancangan perangkat-lunak merupakan perancangan algoritma program untuk merealisasikan sistem kendali *mobile robot*. Pada perancangan ini meliputi dua hal yaitu perancangan perangkat-lunak pada mikrokontroler dan perancangan perangkat lunak pada komputer.

Perancangan perangkat lunak pada mikrokontroler merupakan perancangan yang dilakukan agar mikrokontroler dapat mengambil, mengolah, dan mengirim data *setpoint* koordinat dan kecepatan yang kemudian di konversi dalam gerakan robot yang berupa keluaran PWM motor DC. Perancangan perangkat lunak pada mikrokontroler dilakukan dengan bahasa C dan menggunakan *Integrated Development Environment (IDE) CodeVisionAVR (CVAVR)*. Perancangan perangkat lunak pada mikrokontroler meliputi perancangan pewaktuan program, perancangan program *invers kinematic*, perancangan pembacaan sensor, perancangan pengendali PID dan perancangan algoritma kontrol posisi di dalam *compiler CVAVR*.

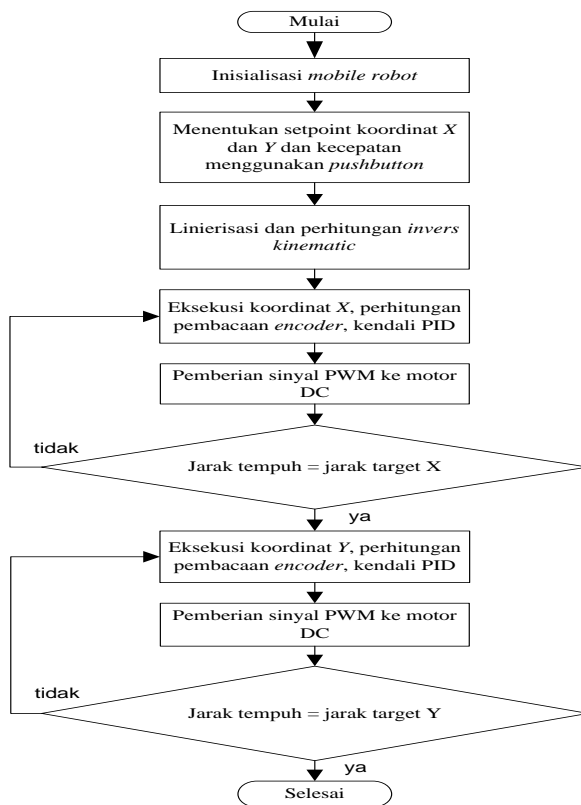
Perancangan perangkat lunak pada komputer dilakukan agar data dari mikrokontroler dapat diterima, diolah dan ditampilkan di komputer. Perancangan perangkat lunak pada komputer dibuat menggunakan *IDE Microsoft Visual Studio 2010* dengan bahasa C#. Perancangan perangkat lunak pada komputer terdiri dari perancangan penerimaan data serial, dan penampilan data.



Gambar 7. Tampilan GUI *mobile robot*

2.8. Perancangan Sistem Keseluruhan

Secara umum, diagram alir algoritma kontrol posisi pada sistem *mobile robot* dapat dilihat pada Gambar 8.



Gambar 8. Diagram alir *mobile robot*

Pada gambar 7 diketahui bahwa algoritma kontrol posisi dimulai dengan menentukan *setpoint* koordinat dan kecepatan menggunakan inputan *pushbutton*. Selanjutnya adalah proses linierisasi koordinat dan kecepatan menjadi kecepatan *mobile robot* dalam V_x dan V_y . Proses ini dapat dilihat pada gambar 5 dan ditandai dengan kotak bernomor 1. Kemudian dilanjutkan pada proses di kotak bernomor 2 yaitu perhitungan *invers kinematic* yang menghasilkan *setpoint* kecepatan pada masing-masing

roda. Setelah proses tersebut maka *mobile robot* akan bergerak sesuai dengan arah gerak tiap roda. Pergerakan *mobile robot* diawali dengan mengeksekusi sumbu X . Setelah jarak tempuh *mobile robot* melebihi *setpoint* sumbu X maka *mobile robot* mengeksekusi sumbu Y . Bila jarak tempuh *mobile robot* melebihi *setpoint* sumbu Y maka robot berhenti.

3. Hasil dan Analisis

3.1. Pengujian Pembacaan Sensor *Encoder Hall Effect*

Proses perhitungan pulsa dilakukan secara kontinu oleh mikrokontroler. Pengujian kecepatan dilakukan dengan membandingkan data pembacaan mikrokontroler dengan pembacaan tachometer. Data hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Perbandingan antara pembacaan kecepatan roda mikrokontroler dan tachometer.

PWM (8bit)	Pembacaan Mikrokontroler (rpm)	Pembacaan Tachometer (rpm)	Galat
10	0	0	0
30	75.21	76.5	1.29
50	128.63	131.1	2.47
70	164.5	167.2	2.7
90	187.94	190.2	2.26
110	202.74	208.3	5.56
130	213.64	216.8	3.16
150	220.1	224.8	4.7
170	225.63	230.4	4.77
190	229.9	234.1	4.2
210	234.35	238.8	4.45
230	236.35	241.2	4.85
250	238.71	244.5	5.79
Rata-rata galat (rpm)			3.55

Dari tabel 1 dapat dilihat bahwa pembacaan mikrokontroler dan tachometer tidak jauh berbeda hanya terdapat galat kecepatan dengan rata-rata 3.55 rpm. Hal ini menunjukkan bahwa pembacaan sensor *encoder* dan diproses oleh mikrokontroler sudah benar.

3.2. Pengujian Pengendali PID

Kendali PID dibutuhkan untuk memperbaiki respon sistem. Untuk mendapatkan respon sistem yang baik maka diperlukan *tuning* terhadap nilai parameter PID yaitu K_p , T_i dan T_d . Pada penelitian kali ini, metode *tuning Ziegler-Nichols 2* dipilih dalam menentukan parameter PID karena kemudahan dalam percobaan. Pengujian dilakukan dengan salah satu motor dengan *setpoint* 80 rpm.

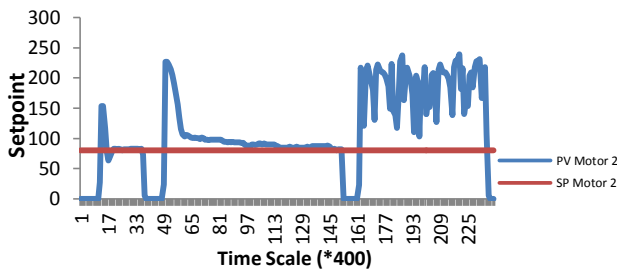
Langkah pertama dalam melakukan metode *tuning Ziegler-Nichols 2* adalah menambahkan nilai K_p sampai sistem berosilasi kesinambungan, dan tetapkan nilai tersebut sebagai nilai K_u . Langkah selanjutnya adalah mencari nilai P_u . Nilai P_u adalah Periode osilasi pada saat *gain* maksimum K_u dan didapatkan melalui pengamatan

respon sistem. Setelah mendapatkan nilai K_u dan P_u maka kalikan dengan konstanta metode *tuning Ziegler Nichols 2*.

Tabel 2. Hasil pengali dengan konstanta gain metode *tuning Ziegler Nichols 2*

Tipe Kontrol	K_p	T_i	T_d
P	0.325	0	0
PI	0.29	1.33	0
PID	0.39	0.8	0.2

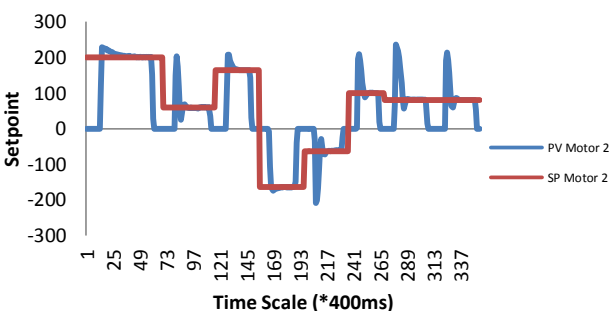
Setelah mendapatkan parameter PID langkah selanjutnya adalah pemilihan tipe kontrol dengan pengamatan grafik respon sistem. Respon sistem yang paling baik akan digunakan sebagai parameter yang digunakan dalam sistem kontrol. Gambar 9 adalah grafik keluaran respon sistem.



Gambar 9. Perbandingan hasil tipe kontrol pada metode *tuning Ziegler-Nichols 2*

Dari gambar dapat dilihat bahwa tipe kontrol P sendiri memiliki respon yang lebih baik bila tidak dikombinasikan dengan I dan ID. Maka dari itu, nilai parameter tipe kontrol *proportional* akan menjadi nilai parameter yang digunakan *mobile robot*.

Selanjutnya adalah pengujian pada *setpoint* kecepatan yang berbeda-beda untuk melihat tipe kontrol yang sudah dipilih selalu stabil atau tidak. Gambar Berikut adalah hasil respon sistem dengan nilai $K_p=0.325$, $T_i=0$, $T_d=0$ dan *setpoint* kecepatan yang berbeda. *Setpoint* 1 hingga 8 berturut-turut adalah 200, 60, 164, -164, -64, 100, 80 dan 80 rpm.



Gambar 10. Percobaan grafik respon di *setpoint* kecepatan berbeda

Dari Gambar 10 dapat disimpulkan bahwa nilai parameter yang dicari menggunakan metode *tuning Ziegler-Nichols 2* berhasil karena tidak ada grafik respon yang tidak stabil hanya saja di tiap *setpoint* kecepatan yang berbeda memiliki *rise time*, *overshoot* dan *time delay* yang berbeda pula.

3.3. Pengujian Perhitungan Kinematik

Pengujian kinematika robot dilakukan dengan memberikan nilai kepada variabel kecepatan V_x , V_y dan ω . Kemudian nilai tersebut dimasukan dalam persamaan kinematika robot untuk menghasilkan variabel baru berupa kecepatan sudut tiap roda ω_1 , ω_2 , ω_3 , ω_4 .

Tabel 3. Hasil pengujian perhitungan *invers kinematic robot*

Sudut	Kecepatan Robot			ω_1	ω_2	ω_3	ω_4
	V_x	V_y	ω				
0°	1.2	0	0	15.7	15.7	15.7	15.7
45°	0.5	0.5	0	0	13.1	13.1	0
90°	0	0.7	0	-9.2	9.21	9.2	-9.2
135°	-0.3	0.3	0	-7.8	0	0	-7.8
180°	-0.7	0	0	-9.2	-9.2	-9.2	-9.2
225°	-0.4	-0.4	0	0	-10.2	-10.2	0
270°	0	-0.9	0	11.8	-11.8	-11.8	11.8
315°	0.8	-0.8	0	21.0	0	0	21.0

Dari tabel 3 dan gambar 12 dapat dilihat bahwa arah pergerakan tiap roda untuk membentuk sudut hadap *mobile robot* sudah benar.

Pengujian kedua adalah membandingkan kecepatan roda robot dengan perhitungan hasil *invers kinematic*. Pengujian dilakukan kepada *mobile robot* dalam kondisi roda mekanum tidak menyentuh lantai.

Tabel 4. Perbandingan kecepatan roda robot

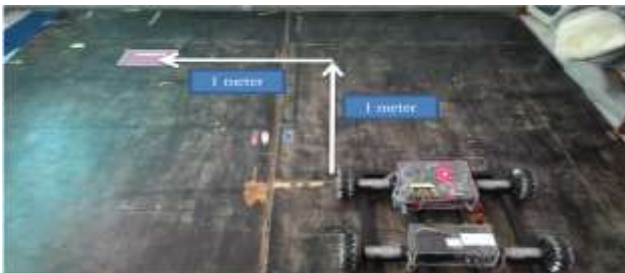
sudut	Teori / Uji	Kecepatan Roda (rpm)			
		R1	R2	R3	R4
0°	Teori	150.85	150.85	150.85	150.85
	Uji	149.33	149.33	149.33	149.33
45°	Teori	0	125.71	125.71	0
	Uji	0	125.35	125.35	0
90°	Teori	-88	88	88	-88
	Uji	-87.02	86.2	88.2	-88.2
135°	Teori	-75.42	0	0	-75.42
	Uji	-75.21	0	0	-75.21
180°	Teori	-88	-88	-88	-88
	Uji	-87.2	-88.2	-89.3	-86.11
225°	Teori	0	-100.5	-100.5	0
	Uji	0	-99.1	-100.2	0
270°	Teori	-113.1	113.1	113.1	-113.1
	Uji	-112.2	112.18	112.2	-113.3
315°	Teori	201.13	0	0	201.13
	Uji	198.38	0	0	200.56
Galat (rpm)		0.89	0.79	0.57	0.58

Dari tabel 4 dapat diketahui bahwa perbandingan antara kecepatan motor dalam perhitungan dengan yang sebenarnya adalah kesalahan motor satu sebesar 0,89 rpm, motor dua 0,79 rpm, motor tiga 0,57 rpm dan motor empat 0,58 rpm. Penelitian yang dilakukan oleh Veri[1]

menghasilkan galat kecepatan lebih besar bila dibandingkan dengan galat yang dihasilkan pada tabel 4. Hal ini karena pengendalian yang dilakukan pada penelitian ini menggunakan kontrol *closeloop*.

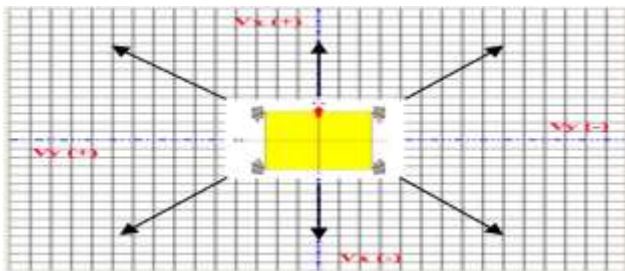
3.4. Pengujian Keseluruhan Sistem

Pengujian keseluruhan dikatakan berhasil bila arah *mobile robot* menuju titik target. Pengujian dilakukan di lantai berbahan kayu dengan ukuran 3x3 meter, sehingga pengujian hanya dapat dilakukan dengan target 1 meter. Pengujian dilakukan pada kecepatan 80 rpm dan 200 rpm.



Gambar 11. Mobile robot saat start pada pengujian keseluruhan

Setelah melakukan seluruh pengujian arah koordinat maka data disajikan dalam bentuk tabel. Tabel 5 dan 6 menunjukkan hasil dari pengujian yang telah dilakukan di lapangan uji berbahan kayu. Sedangkan gambar 12 adalah gambar untuk memudahkan pemahaman mengenai arah koordinat.



Gambar 12. Koordinat kartesius dan arah hadap mobile robot

Tabel 5. Pengujian keseluruhan pada kecepatan 80 rpm

X (m)	Y (m)	Posisi Akhir (cm)	Galat heading (°)	Galat jarak (cm)
1	1	173	19	32
-1	1	168	10	27
-1	-1	186	11	45
1	-1	177	15	36
Rata-rata			13.75	35

Tabel 6. Pengujian keseluruhan pada kecepatan 200 rpm

X (m)	Y (m)	Posisi Akhir (cm)	Galat heading (°)	Galat jarak (cm)
1	1	185	10	44
-1	1	177	15	36
-1	-1	183	13	42
1	-1	173	11	32
Rata-rata			12.25	38.5

Berdasarkan tabel dapat diperoleh hasil rata-rata galat sudut *heading* robot dengan kecepatan 80 rpm sebesar 13,75° dengan galat jarak sejauh 35 cm, sedangkan kecepatan 200 rpm galat *heading* robot sebesar 12,25° dengan galat jarak 38,5 cm. Meskipun terdapat galat *heading* dan jarak namun arah hadap *mobile robot* dalam mencapai semua titik target sudah benar. Dapat disimpulkan bahwa pembuatan *mobile robot* untuk mencapai titik koordinat tertentu dengan satu langkah berhasil dilakukan.

4. Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Telah berhasil dirancang *mobile robot* roda mekanum yang dapat bergerak otomatis menuju koordinat tertentu menggunakan kalkulasi *invers kinematic* dan kendali PID.
2. Tuning parameter kontrol PID pada *mobile robot* menggunakan metode *tuning Ziegler Nichols 2* menghasilkan respon yang selalu stabil dengan rentang nilai *setpoint* 50-235.
3. Pada pengujian keseluruhan yang telah dicobakan dengan kecepatan 80 dan 200 rpm. *Mobile robot* mempunyai rata-rata galat *heading* sebesar 13,75 derajat dan 12,25 derajat, sedangkan galat jaraknya sebesar 35 cm dan 38,5 cm.

Referensi

- [1]. Hendrayawan, V., Implementasi Invers Kinematics pada Sistem Pergerakan Mobile Robot Roda Mekanum, Tugas Akhir S-1, Universitas Brawijaya, Malang, 2014.
- [2]. Boreinstein, Everett. Navigation Mobile Robot. Wellesley: Masschuttes. 1996.
- [3]. Tlale, Nkgatho. 2008. Kinematics and Dinamics Modelling of a Mecanum Wheeled Mobile Platform. IEEE Mechatronics and Machine Vision in Practice, Vol.15, pp.657- 662.
- [4]. Taheri, Hamid., Bing Qiao, and Nurallah Ghaeminezhad, Kinematic Model of a Four Mecanum Wheeled Mobile Robot", International Journal of Computer Applications (0975 – 8887), Volume 113 – No. 3, March 2015.
- [5]. P. Muir, C. Neuman "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot", Proceedings, IEEE International Conference in Robotics and Automation, Vol 4. pp. 1772-1778, 1987.
- [6]. T. Braunl, "Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems", First Edition. Springer-Verlag, Berlin, 2003, pp. 113-121.

- [7]. West,M, Asada,H.1997.Design of Ball Wheel Mechanisms for Omnidirectional Vehicle with Full Mobility and Invariant Kinematics. Journal of Mechanical Design, Vol.119, pp.153-161.
- [8]. Kusuma, W.J., Penerapan Invers Kinematik Terhadap Pergerakan Kaki pada Robot Hexapod.Tugas Akhir S-1, STMIK GI MDP, Palembang, 2013.