JTERA (Jurnal Teknologi Rekayasa), Vol. 6, No. 2, Desember 2021, Hal. 273-284
Terakreditasi "**Peringkat 3**" oleh Kemenristek/BRIN, Nomor SK: 85/M/KPT/2020
DOI: 10.31544/jtera.v6.i2.2021.273-284
p-ISSN 2548-737X
e-ISSN 2548-8678

Kendali Kecepatan Robot Beroda Omni dengan Kemampuan Menuju Posisi dan Orientasi yang Diinginkan Berbasis Pengendali PID

Faisal Wahab[#], Clement Zaskie, Bagus Made Arthaya
Universitas Katolik Parahyangan
Jl. Ciumbuleuit No. 94 Bandung 40141, Jawa Barat, Indonesia
[#]faisal.wahab@unpar.ac.id

Abstrak

Sebuah robot yang menggunakan penggerak roda omni memiliki kemampuan dapat bergerak ke segala arah tanpa harus mengubah orientasi dari robot itu sendiri. Kemampuan tersebut dapat dibuktikan jika kecepatan setiap motor dapat terkendali dan diterapkan sebuah algoritma supaya robot beroda omni dapat bergerak menuju posisi dan orientasi yang diinginkan. Pada makalah ini akan disajikan kendali kecepatan robot beroda omni dengan kemampuan menuju posisi dan orientasi yang diinginkan menggunakan pengendali Proporsional Integral Derivatif (PID). Tahapan penelitian dimulai dari penentuan kinematika dari robot yang digunakan, perancangan sistem elektrik dan mekanik, perancangan kendali motor DC, dan penentuan diagram blok sistem. Parameter PID yang digunakan untuk kendali motor adalah Kp = 0,48, Ki = 11,16, dan Kd = 0 yang diperoleh dari hasil PID *tuner* dan untuk pergerakkan robot menuju posisi dan orientasi yang diinginkan menggunakan kendali *Model Predictive Control* (MPC). Hasil pengujian secara simulasi menunjukkan bahwa robot beroda omni dapat bergerak sesuai dengan posisi dan orientasi yang berjumlah lima buah *set-point* berdasarkan kinematika yang sudah dirancang. Hasil pengujian secara eksperimental berdasarkan data pembacaan *encoder* menunjukkan bahwa robot beroda omni telah berhasil menuju posisi dan orientasi yang diinginkan. Namun, secara visual terdapat *error* sebesar 9,6% dari keseluruhan *set-point* yang disebabkan oleh faktor teknis lapangan.

Kata kunci: robot beroda omni, kinematika, simulasi, PID, algoritma

Abstract

A robot that uses omni-wheel drive has the ability to move in all directions without having to change the orientation of the robot itself. This ability can be proven if the speed of each motor can be controlled and an algorithm is applied so that the omni-wheeled robot can move to the desired position and orientation. In this paper, an omni wheeled robot speed control will be presented with the ability to go to the desired position and orientation using a Proportional Integral Derivative (PID) controller. The research stage starts from determining the kinematics of the robot used, designing electrical and mechanical systems, designing DC motor control, and determining the system block diagram. The PID parameters used for motor control are Kp = 0.48, Ki = 11.16, and Kd = 0 which are obtained from the results of the PID tuner and to move the robot to the desired position and orientation using Model Predictive Control (MPC). The results of the simulation test show that the omni-wheeled robot can move according to the position and orientation of five set-points based on the kinematics that has been designed. Experimental test results based on encoder reading data show that the omni-wheeled robot has succeeded in getting to the desired position and orientation. However, visually there is an error of 9.6% of the total set-point caused by field technical factors.

Keywords: *omnidirectional wheel robot, kinematics, simulation, PID, algorithm*

I. PENDAHULUAN

Perkembangan robot saat ini berjalan sangat pesat dan diprediksi berlanjut di masa yang akan datang. Perkembangan tidak hanya dari desain dan kegunaan, namun juga berbagai bagian atau komponen dalam robot. Seperti penggunaan algoritma yang digunakan, jenis mikrokontroler, bahkan juga dari akuator yang digunakan. Salah satu jenis roda yang populer dan banyak digunakan

Diterima: 28 Oktober 2021; Direvisi: 2 Desember 2021; Disetujui: 13 Desember 2021
JTERA, Vol. 6, No. 2, Desember 2021

© 2021 JTERA (Jurnal Teknologi Rekayasa), Politeknik Sukabumi

saat ini adalah roda omni yang merupakan salah satu jenis roda yang memiliki inti roda tengah dan juga roda pada sekeliling roda inti. Berbeda dengan jenis roda biasa atau roda diferensial yang memiliki keterbatasan saat bermanuver yaitu harus merubah orientasi saat berputar atau berbelok, robot dengan tipe roda omni memiliki kelebihan dapat bergerak ke segala arah tanpa mengubah orientasi robot, sehingga robot beroda omni dikategorikan jenis robot *holonomic*. Penerapan roda omni pada dunia robot telah banyak digunakan, diantaranya dapat melakukan tugas-tugas berbahaya [1], robot beroda omni dengan pencarian sumber gas [2], *path planning* [3], perlombaan robot [4], dan digunakan *mapping* sebuah ruangan [5]. Pada robot beroda omni, salah satu hal penting saat bermanuver adalah pengendalian kecepatan setiap motor. Pengendalian kecepatan ini sangat penting, karena akan menentukan pergerakan dari robot itu sendiri. Jika kecepatan setiap motor tidak dapat dikendalikan, maka kemungkinan robot roda omni tidak dapat berfungsi atau bermanuver secara optimal.

Oleh karena itu, diperlukan sebuah kendali untuk dapat mengatur kecepatan setiap motor sesuai yang diinginkan. Kendali kecepatan motor pada robot roda omni dapat dirancang menggunakan metode Proporsional Integral Derivatif (PID) [6], [7]. Kendali ini cukup efektif untuk mengatur kecepatan setiap motor. Untuk menguji penerapan kendali kecepatan setiap roda omni pada robot, diperlukan sebuah algoritma untuk mengatur pergerakkan robot supaya menuju posisi dan oritentasi yang diinginkan. Terdapat beberapa algoritma yang dapat digunakan, seperti *Model Predictive Control* (MPC) [8], *Bezier Curve Tracing Technique* [9], logika *fuzzy* [10], dan *Adaptive Neuro Fuzzy Inference System* (ANFIS) [11]. Algoritma MPC menjadi pilihan dikarenakan mampu beradaptasi dengan berbagai perubahan parameter sistem yang dapat digunakan untuk menguji pergerakan robot roda omni [12].

Makalah ini menguraikan rancang bangun kendali kecepatan robot beroda omni dengan kemampuan dapat menuju ke posisi dan orientasi yang diinginkan yang diuji secara simulasi dan eksperimen menggunakan pengendali PID. Pengujian pada makalah ini, berfokus pada pengujian pengaturan kecepatan motor roda omni menggunakan kendali PID, sedangkan penggunaan MPC sebatas untuk kebutuhan manuver robot beroda omni untuk menuju posisi dan orientasi yang diinginkan secara simulasi berdasarkan kinematika yang sudah dirancang. Simulasi dilakukan supaya pergerakan robot dapat diprediksi dengan baik juga mengurangi kerusakan pada robot diakibatkan metode *trial* dan *error*.

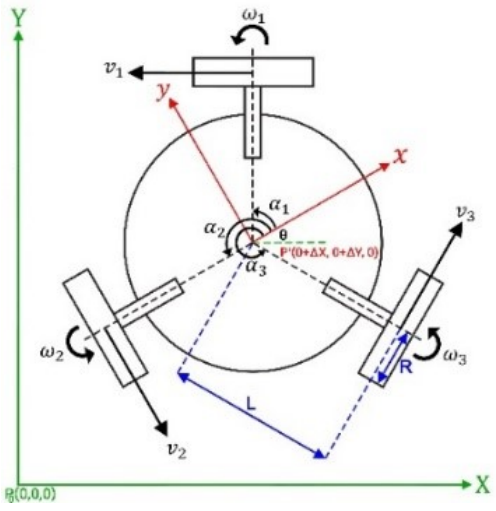
II. METODE PENELITIAN

Terdapat beberapa tahapan dalam rancang kendali kecepatan robot beroda omni dengan kemampuan menuju posisi dan orientasi yang diinginkan. Tahapan tersebut dijelaskan pada sub bagian A sampai dengan sub bagian C.

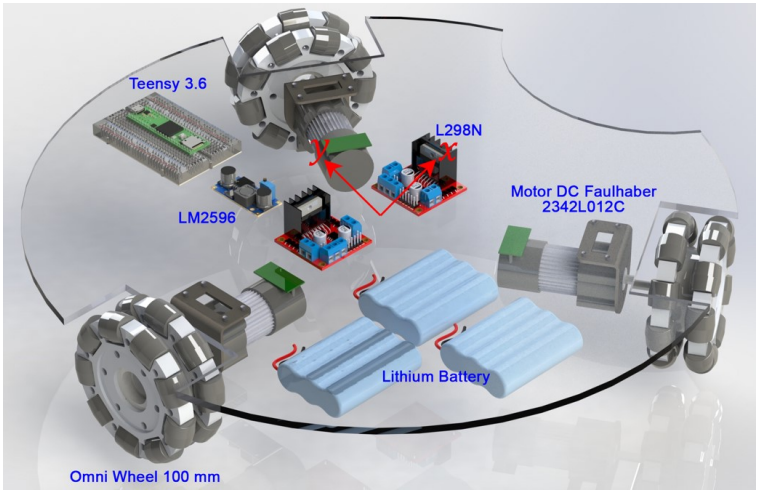
A. Kinematika Robot Beroda Omni

Pada penelitian ini robot dirancang dengan menggunakan kombinasi tiga roda omni. Dalam penerapan kombinasi tersebut menggunakan dua model kinematika, yang pertama adalah kinematika maju (*forward kinematics*) dan yang kedua adalah kinematika mundur (*inverse kinematics*). Kinematika maju digunakan untuk mencari kecepatan robot terhadap sumbu x, y dan rotasi terhadap sumbu z (x, y, dan θ) dari kecepatan masing-masing putaran roda (ω_1, ω_2, ω_3) yang diketahui. Sementara kinematika mundur digunakan untuk mencari kecepatan masing-masing roda (ω_1, ω_2, ω_3) dari kecepatan robot terhadap sumbu x, y, dan rotasi terhadap sumbu z ($\dot{x}$, $\dot{y}$, dan $\dot{\theta}$) yang diketahui. Kinematika robot beroda omni diilustrasikan oleh Gambar 1.

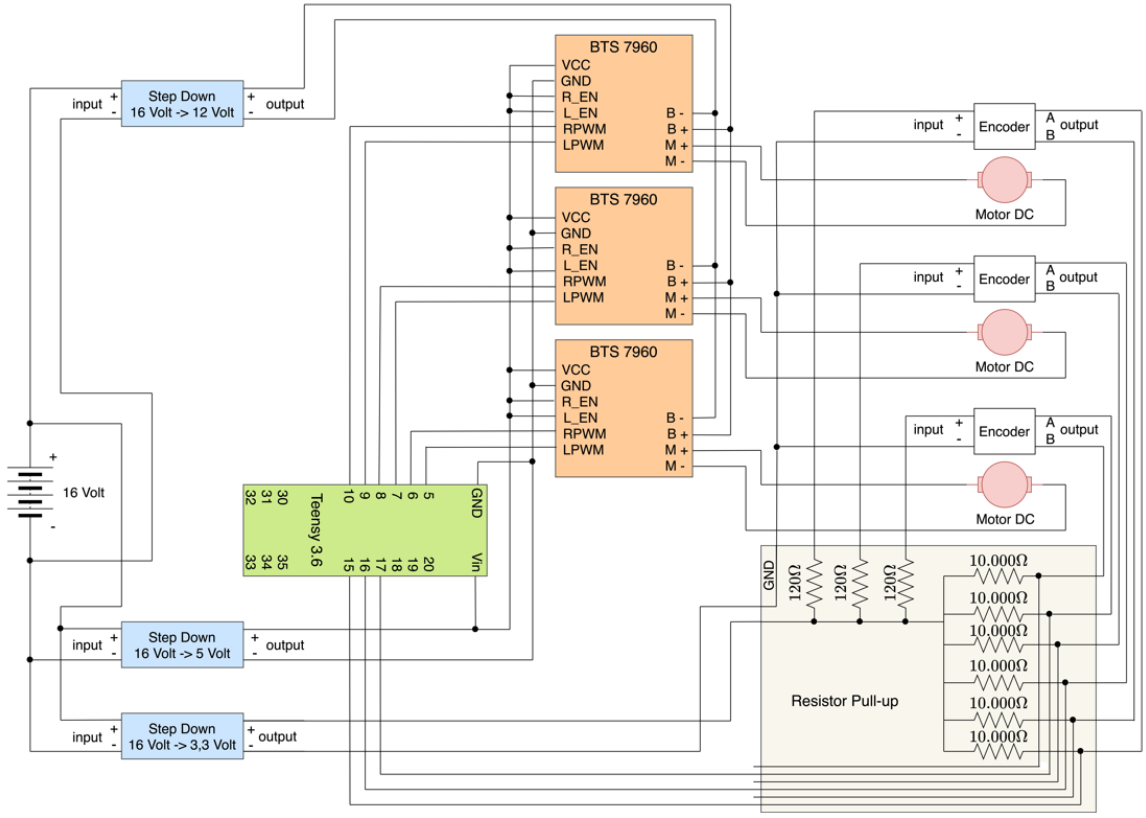
Pada Gambar 1 dimana ukuran panjang dari pusat robot ke pusat roda ditandai dengan (L), panjang jari-jari dari roda (R) dan setiap roda dipasang pada sasis dengan sudut antar roda adalah 120 derajat. Muka dari robot (bagian depan) sejajar dengan sumbu x lokal. Besar sudut antara roda pertama dengan sumbu x lokal ditandai dengan $\alpha_i$dan memiliki besar 60 derajat. Kecepatan putar roda dinotasikan dengan ω_i dan kecepatan tangensial dari roda diberi simbol v_i (i = 1, 2, 3). Kecepatan robot terhadap sumbu x lokal dilambangkan dengan $\dot{x}$ dan kecepatan robot terhadap sumbu y lokal ditandai dengan $\dot{y}$.



Gambar 1. Kinematika robot beroda omni



Gambar 2. Rancangan robot beroda omni



Gambar 3. Skematik rangkaian robot beroda omni

Garis sumbu berwarna merah (x, y) menandakan Kerangka Koordinat Lokal (KKL) yang dapat berputar mengikuti orientasi robot dan terdapat garis sumbu berwarna hijau (X,Y), yang menandakan Kerangka Koordinat Global (KKG) yang tidak berubah-ubah dan dianggap sebagai acuan posisi robot dari pergerakan. Pada mulanya, robot dapat diketahui posisi dan orientasinya berdasarkan KKG. Pada saat robot berjalan maka perlu sebuah matriks rotasi untuk mengubah dari KKG menjadi KKL. Berikut (1) matriks rotasi untuk menentukan posisi dan orientasi robot secara KKL.

$$\begin{bmatrix} x_R \\ y_R \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_G \\ y_G \\ \theta \end{bmatrix} \quad (1)$$

Setelah berada pada KKL, selanjutnya adalah menentukan kinematika maju. Berdasarkan kinematika maju dari robot beroda omni dengan kombinasi tiga buah roda adalah sebagai berikut.

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\theta} \end{bmatrix} = R \begin{bmatrix} \cos\left(\alpha_1 + \frac{\pi}{2}\right) & \cos\left(\alpha_2 + \frac{\pi}{2}\right) & \cos\left(\alpha_3 + \frac{\pi}{2}\right) \\ \sin\left(\alpha_1 + \frac{\pi}{2}\right) & \sin\left(\alpha_2 + \frac{\pi}{2}\right) & \sin\left(\alpha_3 + \frac{\pi}{2}\right) \\ \frac{1}{L} & \frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (2)$$

Dengan dimasukan parameter yang digunakan R = 0,1 meter dan L = 0,25 meter pada (2), selanjutnya dilakukan *inverse* dari (2) dan didapatkan persamaan kinematika mundur (3).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} -3.33 & 5.773 & 0.83 \\ -3.33 & -5.773 & 0.83 \\ 6.67 & 0 & 0.83 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \dot{\theta} \end{bmatrix} \quad (3)$$

Selanjutnya untuk menentukan posisi dan orientasi robot secara global perlu digunakan matriks rotasi. *Invers* matriks rotasi ditampilkan pada (4).

$$\begin{bmatrix} x_G \\ y_G \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ \theta \end{bmatrix} \quad (4)$$

B. Desain Sistem Mekanik dan Elektrik

Gambar 2 menunjukkan desain dalam 3 dimensi dari robot beroda omni yang dirancang. Sasis menggunakan material akrilik berwarna bening. Motor DC diletakkan pada bagian bawah sasis untuk menambah jarak dari sasis ke permukaan lantai dan komponen elektrik disimpan pada bagian atas sasis. Sistem elektrik dari robot beroda omni terdiri dari sumber tegangan, regulator tegangan, mikrokontroler, *motor driver*, motor DC, dan *encoder* (*built-in* pada motor). Skematik rangkaian elektrik ditampilkan dalam Gambar 3. Adapun penjelasan lebih detil setiap komponen yang digunakan diberikan dalam Tabel 1.

Tabel 1. Komponen yang digunakan

Komponen	Fungsi
Teensy 3.6	Pengendali robot beroda omni
Motor driver BTS7960	Mengontrol motor *gearbox* yang berjenis Faulhaber 2342L012CR
DC *step down* XL4016	Menurunkan tegangan dari baterai 16 V menjadi 12 V
XL4015	Menurunkan tegangan baterai 16 V menjadi 5 V
LM2596	*Step down* DC *to* DC untuk rangkaian *resistor pull-up* pada *encoder* dan juga mengolah data yang keluar dan masuk ke Teensy harus bertegangan 3,3 V

C. Perancangan Kendali Motor DC

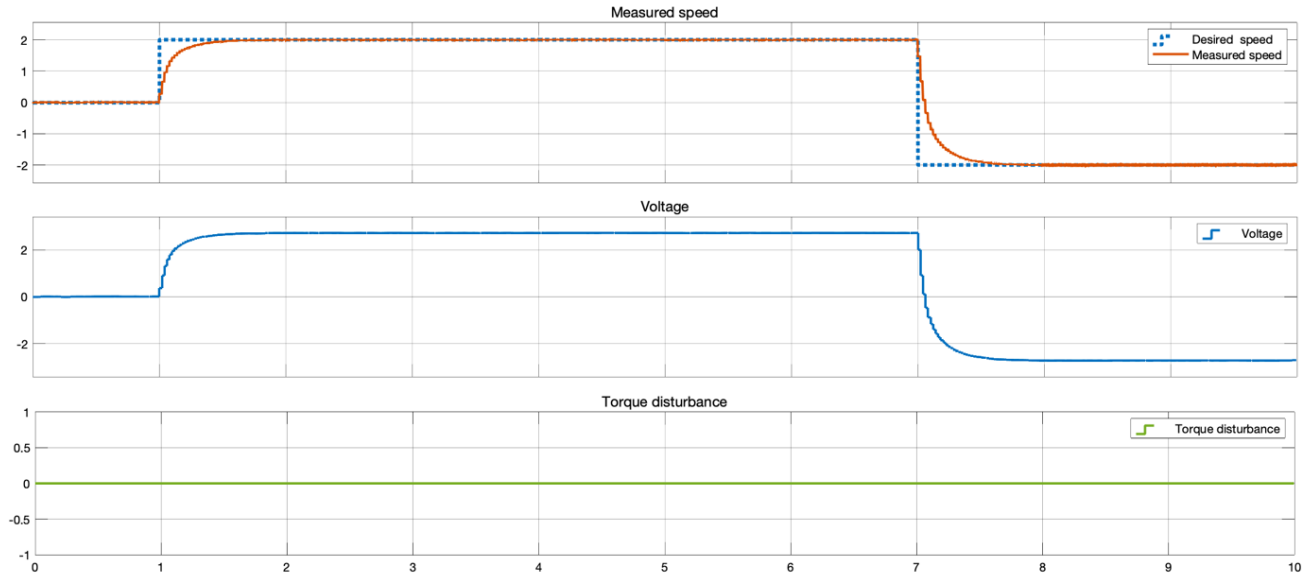
Sebelum merancang kendali untuk pengaturan kecepatan motor DC, akan dilakukan pemodelan sistem terlebih dahulu dari Motor DC Faulhaber 2342L012CR yang digunakan. Parameter motor DC menggunakan dari [13] yang akan diterapkan pada (5).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{Kt}{J}\left(\frac{r_2}{r_1}\right) \\ -\frac{k_b}{L}\left(\frac{r_2}{r_1}\right) & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u \quad (5)$$
$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

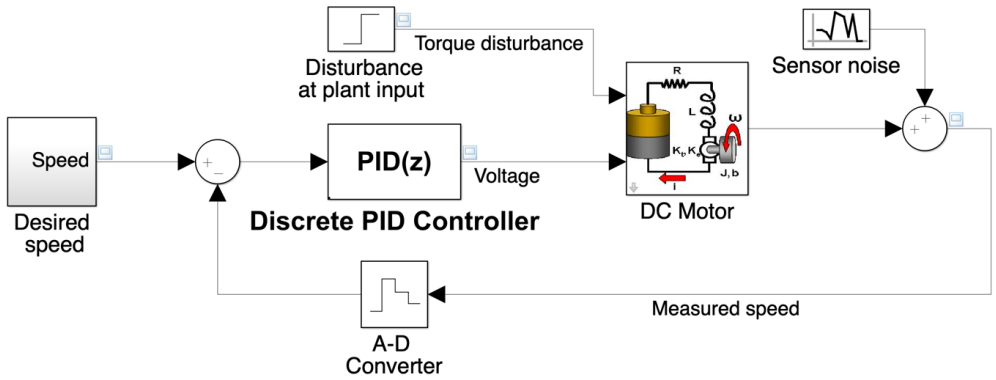
dimana $J = (J_1(r_2/r_1)^2 + J_2)$ adalah inersia rotor, $b = (b_1(r_2/r_1)^2 + b_2)$ merupakan gaya gesek motor, r_1 dan r_2 adalah rasio gir, k_b dan k_t adalah konstanta *back emf* dan konstanta torsi, begitu juga R dan L merupakan tahanan dan induktansi motor.

Gambar 4 menunjukkan respon motor DC saat diberikan sebuah masukan secara simulasi. Pengujian model matematika ini dimulai dengan pemberian *input* berupa *unit step* atau masukan target kecepatan yang ingin dicapai. Kecepatan motor yang terukur harus diubah dari *analog* menjadi *digital* karena set kecepatan yang diinginkan menggunakan *unit step*. *Error* akan muncul saat terjadi perbedaan antara *unit step* dan kecepatan motor yang terukur.

Untuk memberikan respon yang baik, maka diperlukan sebuah pengaturan supaya antara masukan dan keluaran tidak terdapat perbedaan. Pada penelitian ini, kendali yang digunakan adalah menggunakan PID. Pada beberapa penelitian, PID cukup efektif untuk mengendalikan kecepatan motor DC. Pada penggunaan kendali PID, terdapat nilai konstanta yang perlu ditentukan. Penentuan parameter PID pada penelitian ini, menggunakan *tuning* PID yang terdapat pada Simulink dengan memanfaatkan PID *tuner*. Respon dari blok dapat disesuaikan dengan respon yang sudah di-*tune*. *Response time* dapat dipilih lambat atau cepat, dan kondisi *transient behavior* dapat dipilih antara *aggressive* atau *robust.* Rancangan yang dibuat ditampilkan pada Gambar 5.



Gambar 4. Simulasi respon motor sebelum diberikan kendali



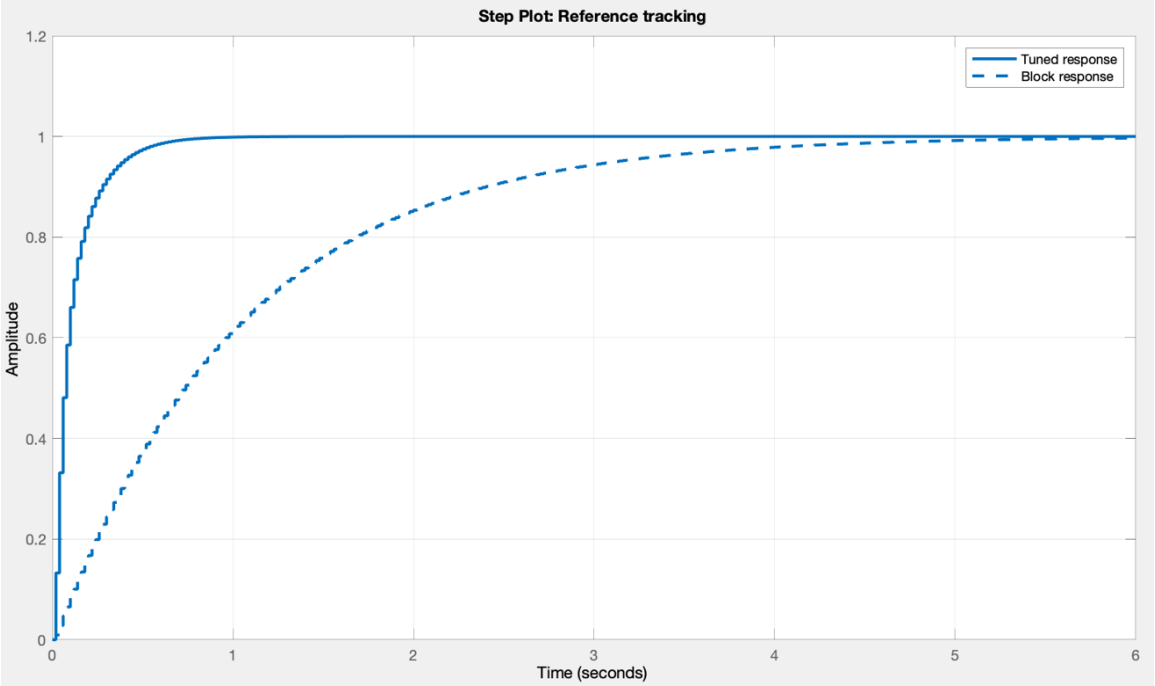
Gambar 5. Rancangan kendali motor menggunakan Simulink

Pemilihan PID *tuning* berdasarkan paling cepat dan *robust*. Hasil *tuner* otomatis ini menghasilkan nilai Kp sebesar 0,48, nilai Ki sebesar 11,16 dan nilai Kd sebesar 0. Pada Gambar 6 menunjukkan hasil respon Motor DC setelah diberikan kendali PID yang telah dimasukan nilai parameternya.

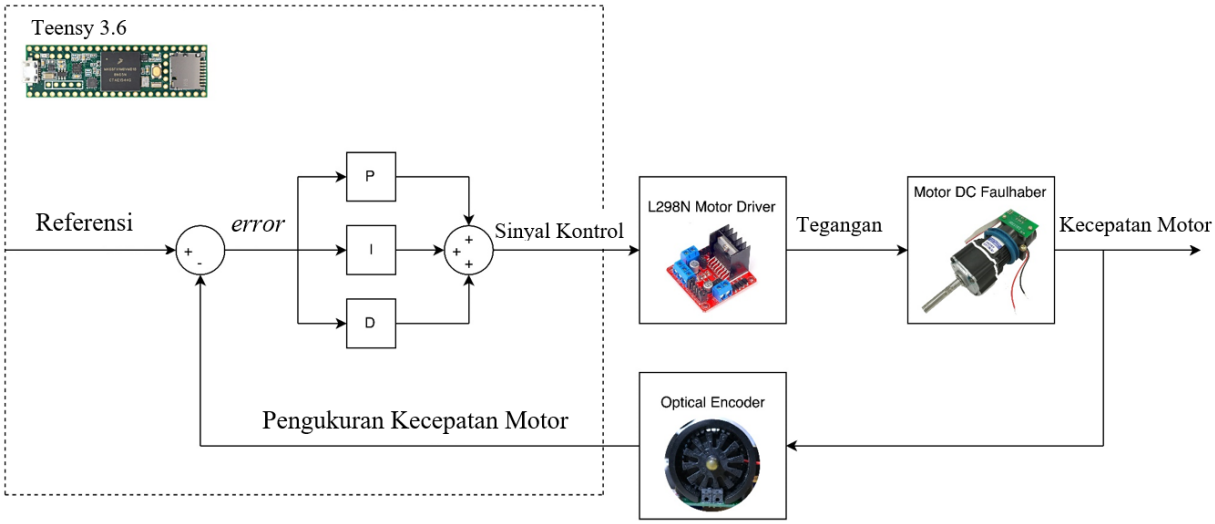
Setelah dilakukan secara simulasi, selanjutnya pengujian kendali motor DC secara nyata. Gambar 7 menunjukkan diagram dari pengujian kendali PID pada motor DC secara nyata.

Mikrokontroler yang digunakan menggunakan Teensy 3.6 karena memiliki spesifikasi *clock internal* yang cukup tinggi dibandingkan dengan mikrokontroler tipe lainnya sehingga respon yang dihasilkan dapat lebih akurat. Masukan yang diberikan berupa kecepatan motor sebesar 15 rad/s. dari mikrokontroler Teensy 3.6 akan dikonversi menjadi nilai *pulse width modulation* (PWM) dan selanjutnya *driver* motor akan mengubah nilai PWM tersebut menjadi tegangan. Keluar kecepatan dari DC motor selanjutnya dibaca oleh *encoder* yang terpasang *built-in* di dalam motor tersebut. Dari *encoder* akan dibaca kembali oleh Teensy 3.6 sebagai pulsa-pulsa dan nantinya akan dikonversi kembali menjadi satuan kecepatan. Sensor *encoder* bertugas untuk mengirimkan sinyal kepada mikrokontroler sehingga dapat dihitung jumlah pulsa yang diterjemahkan menjadi data kecepatan putaran roda dengan membagi jumlah hitungan dengan nilai *Count per Revolution* (CPR) dari *encoder*. Kemudian akan didapat data jumlah putaran roda persatuan waktu (jumlah putaran roda per-100 ms), yang dapat diubah menjadi kecepatan putar roda (rad/s).

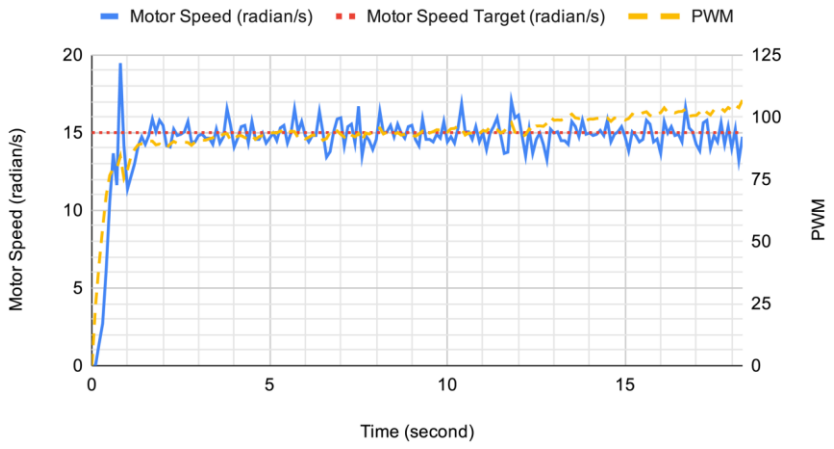
Gambar 8 menunjukkan hasil respon DC motor menggunakan konstanta Kp = 0,48, Ki = 11, 16, dan Kd = 0. Didapatkan hasil respon yang baik, walaupun terjadi *overshoot*, tapi dengan cepat mencapai kondisi *steady state* kurang dari 2 detik, dan dengan sedikit osilasi.



Gambar 6. Respon kendali motor setelah diberikan kendali



Gambar 7. Diagram blok kendali motor



Gambar 8. Respon kendali motor eksperimental

III. HASIL DAN PEMBAHASAN

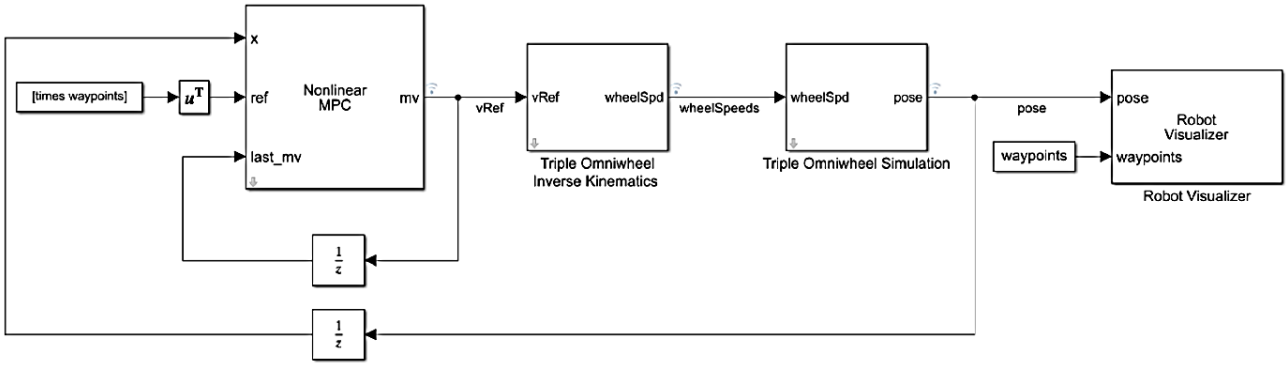
A. Simulasi Pergerakan Robot

Setelah diketahui persamaan *forward kinematics* dan *inverse kinematics* yang digunakan, selanjutnya adalah dilakukan pengujian secara simulasi. Pengujian menggunakan perangkat lunak MATLAB dengan Simulik.

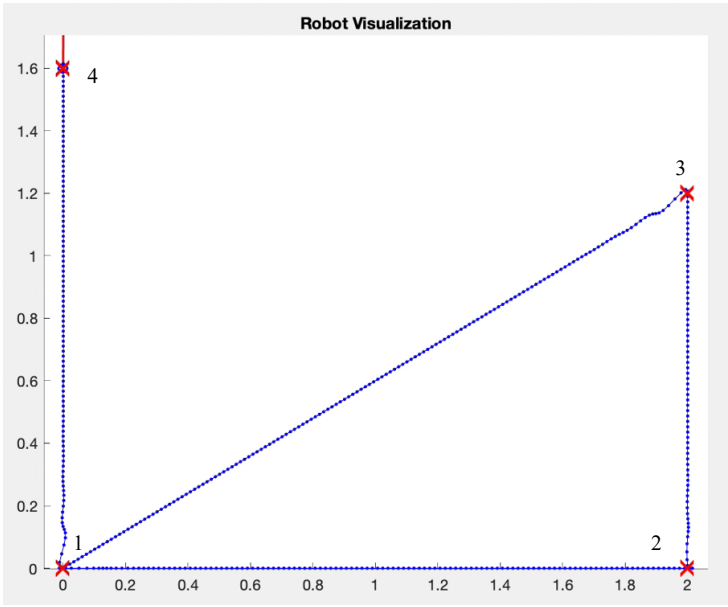
Gambar 9 menunjukkan diagram blok sistem pergerakan robot menggunakan Simulink. Terdapat blok yang digunakan, diantaranya *Nonlinear MPC* yang digunakan untuk menentukan posisi dan orientasi yang diinginkan, *Triple Omniwheel Inverse Kinematics* berfungsi untuk mengubah dari kecepatan robot menjadi kecepatan roda, *Triple Omniwheel Simulation* berfungsi untuk mengubah kecepatan dari roda menjadi posisi robot dalam dua dimensi, sedangkan *Robot Visualizer* digunakan untuk menampilkan posisi robot secara visual.

Pada pengujian secara simulasi, terdapat lima titik atau *set-point* pada KKG yang ditampilkan Gambar 10. Posisi awal robot berada pada titik X = 0, Y = 0, dan $\theta = 0$ dan posisi akhir robot berada pada X = 0, Y = 1.6, dan $\theta = 90^{\circ}$ dengan lintasan 1-2-3-1-4.

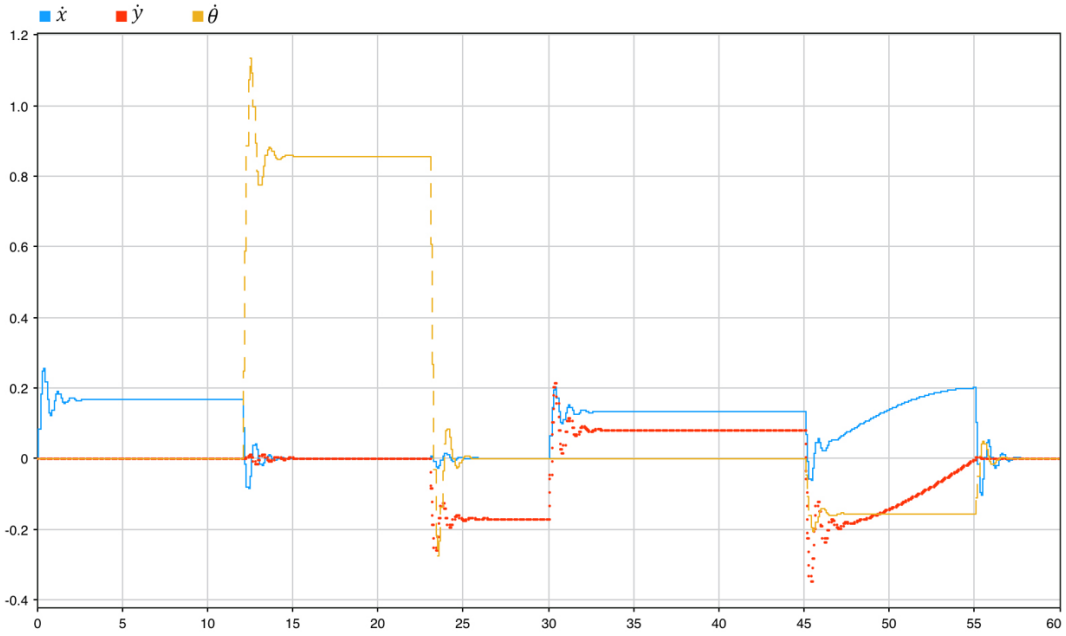
Hasil dari simulasi menggunakan berupa data kecepatan robot terhadap sumbu x dan y lokal ($\dot{x}$,$\dot{y}$), serta kecepatan putar robot robot terhadap poros sumbu z lokal ($\dot{\theta}$), ditampilkan oleh grafik dalam Gambar 11. Gambar 11 menampilkan data kecepatan robot terhadap sumbu x, y, dan θ. Garis tidak putus berwarna biru atau disebut $\dot{x}$ menunjukkan kecepatan robot terhadap sumbu x lokal. Garis titik-titik disebut $\dot{y}$ menunjukkan kecepatan robot terhadap sumbu y lokal. Garis putus-putus disebut $\dot{\theta}$ menunjukkan kecepatan rotasi robot terhadap sumbu z lokal. Grafik sumbu Y menunjukkan kecepatan dalam satuan rad/s dan grafik sumbu X menunjukkan waktu dalam satuan detik.



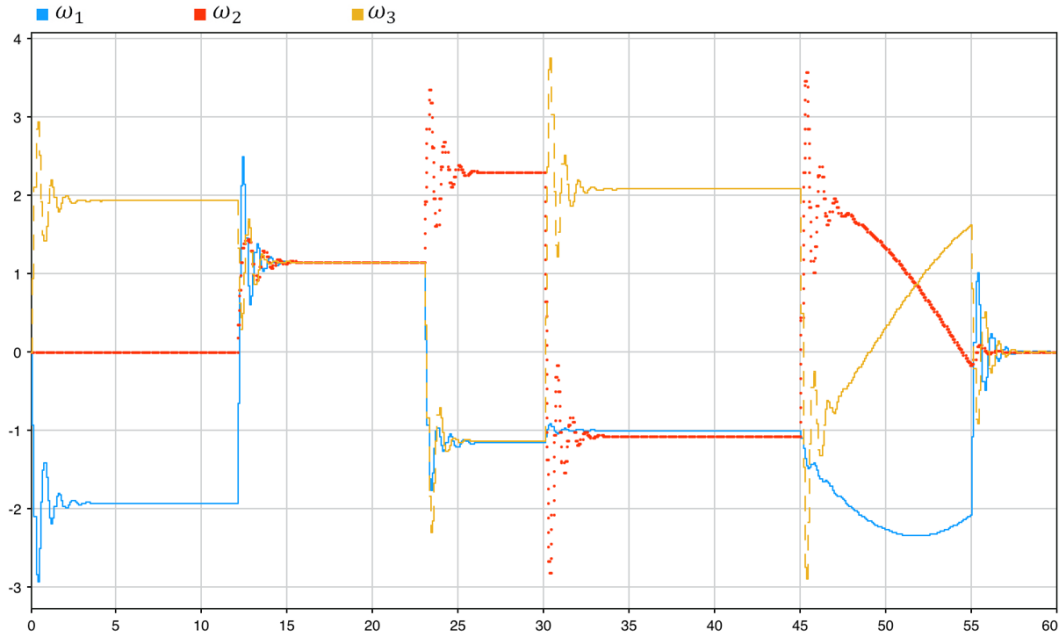
Gambar 9. Diagram blok simulasi pada MATLAB/Simulink



Gambar 10. Simulasi pergerakan robot



Gambar 11. Kecepatan robot terhadap sumbu lokal



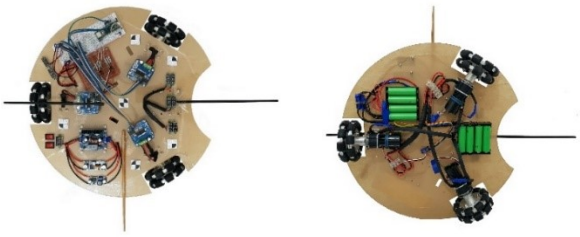
Gambar 12. Data kecepatan putar setiap roda

Grafik pada Gambar 11 merupakan hasil dari simulasi pergerakan robot menuju 5 titik/kasus. Kasus pertama diselesaikan dalam waktu 12 detik (0-12), kasus kedua diselesaikan dalam waktu 11 detik (12-23), kasus ketiga diselesaikan dalam waktu 7 detik (23- 30), kasus keempat diselesaikan dalam waktu 15 detik (30-45) dan kasus kelima diselesaikan dalam waktu 12 detik (45-57). Ketika data kecepatan robot terhadap sumbu x, y, dan rotasi z lokal $(\dot{x}, \dot{y}, \dot{\theta})$ sudah diperoleh kecepatan putar dari masing-masing roda $(\omega_1, \omega_2, \omega_3)$ bisa dicari menggunakan persamaan *inverse kinematics*. Kecepatan dari masing-masing roda omni dari hasil simulasi dapat dilihat pada Gambar 12.

Gambar 12 menampilkan data kecepatan dari masing-masing roda (ω_i). Garis tidak putus (ω_1) adalah roda omni ke 1, garis titik-titik disebut (ω_2) adalah roda omni ke 2, garis putus-putus (ω_3) adalah roda omni ke 3.

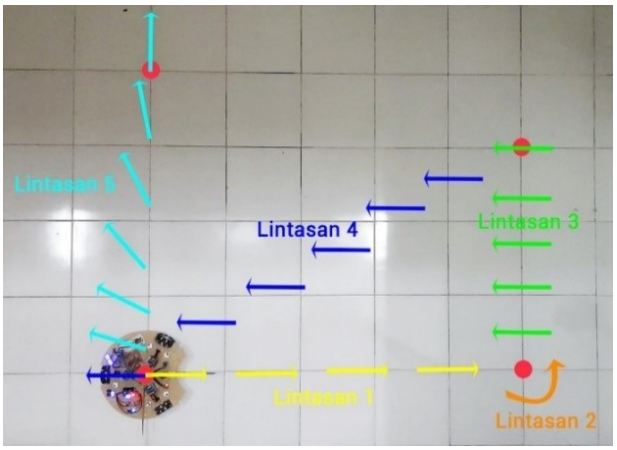
B. Eksperimen Pergerakan Robot

Hasil rancang bangun robot beroda omni secara fisik ditunjukkan Gambar 13 yang terlihat tampilan atas (a) dan tampilan bawah (b). Pada bagian atas digunakan untuk menyimpan komponen sedangkan bagian bawah digunakan unuk menyimpan baterai dan motor DC.

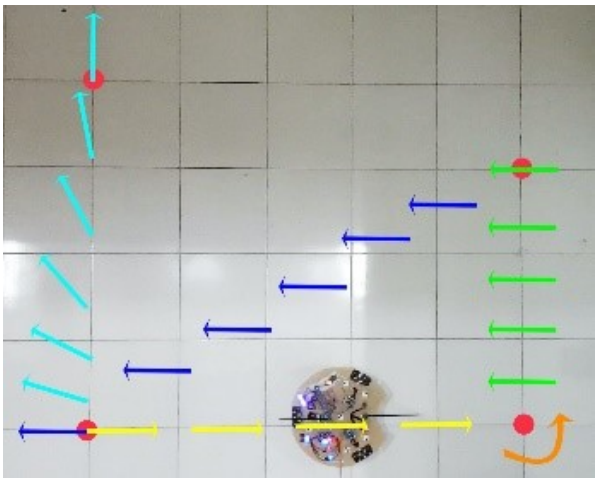


(a) (b)

Gambar 13. Robot beroda omni: (a) tampak atas, (b) tampak bawah



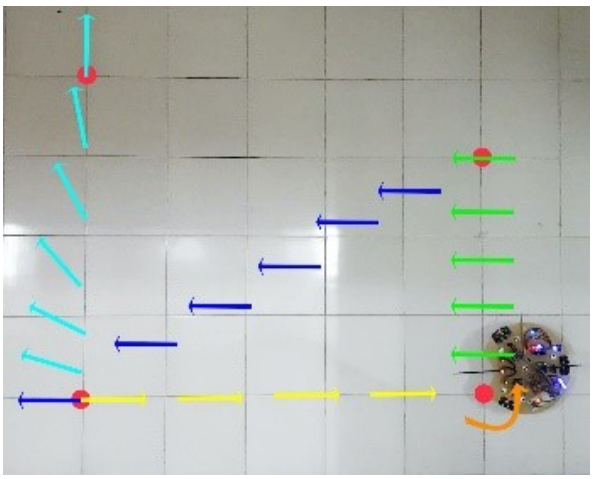
Gambar 14. Uji lintasan pergerakan robot



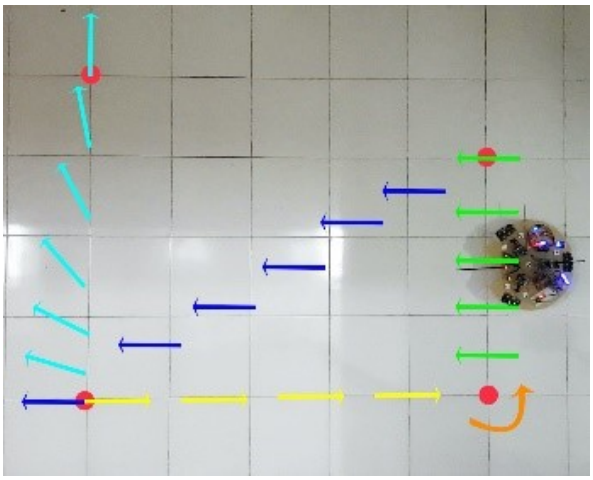
Gambar 15. Pergerakan robot pada lintasan pertama

Pengujian robot secara eksperimen dilakukan pada area tertutup, dengan lantai keramik licin (ubin), dan besar 1 kotak ubin adalah 40 cm. Robot akan menuju posisi atau *set-point* dan orientasi yang telah ditentukan. Posisi dan orientasi robot beroda omni disesuaikan dengan simulasi yang telah dilakukan sebelumnya. Pengujian dilakukan dengan pengambilan data kecepatan setiap roda dan pengamatan secara visual menggunakan kamera. Gambar 14 menunjukkan lintasan pergerakan robot beroda omni yang akan diuji.

Pengujian terdiri dari lima *set-point* dan orientasi tertentu yang ditandai dengan tanda panah



Gambar 16. Pergerakan robot pada lintasan kedua

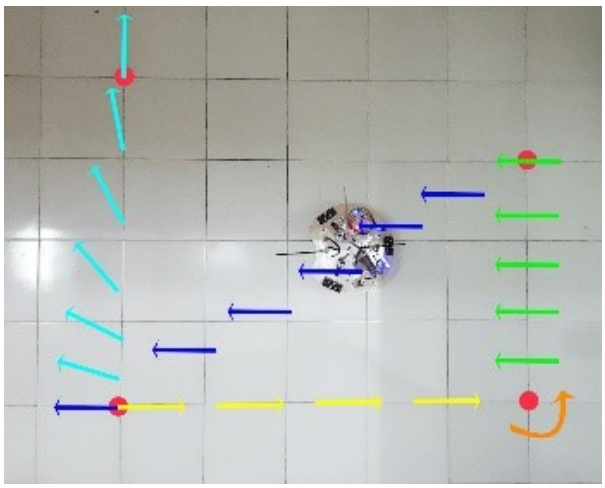


Gambar 17. Pergerakan robot pada lintasan ketiga

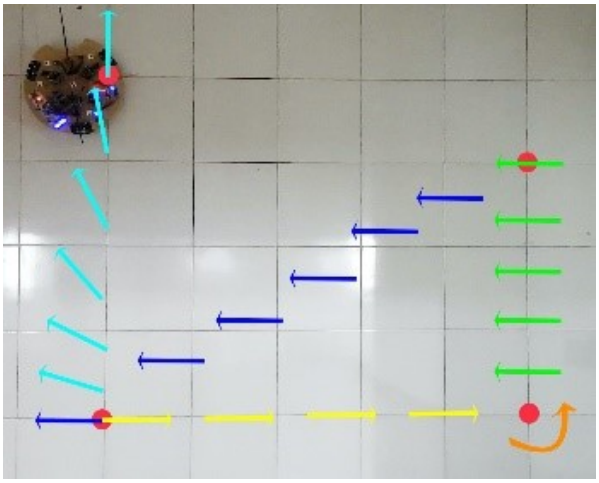
berwarna kuning merupakan lintasan robot yang pertama, lintasan robot kedua berwarna jingga (robot berotasi pada porosnya), lintasan ketiga berwarna hijau, lintasan keempat berwarna biru tua, dan lintasan kelima berwarna biru muda. Pada *set-point* pertama robot berada di posisi awal yaitu koordinat global X = 0, Y = 0 dan menghadap ke sumbu X positif global ($\theta = 0°$ atau 0 radian). Robot akan menuju ke *set-point* kedua yaitu koordinat global X = 2, Y = 0, dengan orientasi ($\theta = 0$) tetap menghadap ke sumbu X positif global.

Pada Gambar 15 terdapat titik-titik merah yang merupakan koordinat tujuan robot. Untuk mencapai koordinat X = 2 meter, maka robot perlu maju sejauh 5 buah ubin (panjang sisi ubin 40 cm) yang ditandai panah berwarna kuning merupakan lintasan robot yang pertama.

Lintasan kedua (jingga) adalah robot berada di koordinat global X = 2, Y = 0 dan menghadap ke sumbu X positif global ($\theta = 0°$ atau 0 radian). Robot akan berputar pada porosnya sejauh 1,5 putaran berlawanan arah jarum jam, sehingga posisi



Gambar 18. Pergerakan robot pada lintasan keempat



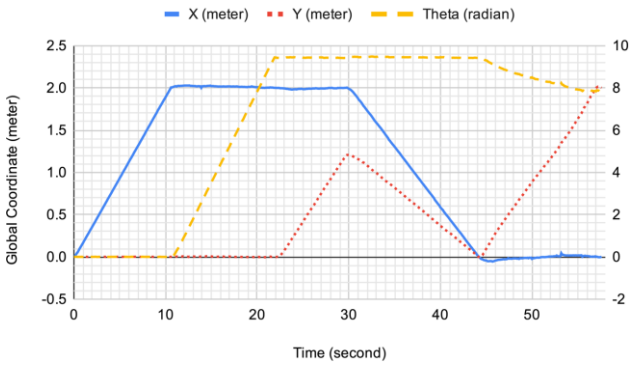
Gambar 19. Pergerakan robot pada lintasan kelima

akhir menghadap ke sumbu X negatif global ($\theta = 540°$ atau 9,42 radian). Hasil pergerakan robot pada lintasan kedua dapat dilihat pada Gambar 16.

Lintasan ketiga (hijau) adalah robot berada di koordinat global X = 2, Y = 0 dan menghadap ke sumbu X negatif global ($\theta = 540°$ atau 9,42 radian). Robot akan menuju ke koordinat X = 2, Y = 1,2, dengan orientasi tidak berubah tetap menghadap ke sumbu X negatif global ($\theta = 540°$atau 9,42 radian). Hasil pergerakan robot pada lintasan ketiga dapat dilihat pada Gambar 17.

Lintasan keempat (biru tua) adalah robot berada di koordinat global X = 2, Y = 1,2 dan menghadap ke sumbu X negatif global ($\theta = 540°$atau 9,42 radian). Robot akan menuju ke koordinat X = 0, Y = 0, dengan orientasi tidak berubah tetap menghadap ke sumbu X negatif global ($\theta = 540°$atau 9,42 radian). Hasil pergerakan robot pada lintasan keempat dapat dilihat pada Gambar 18.

Lintasan kelima (biru muda) adalah robot berada di koordinat global X = 0, Y = 0 dan menghadap ke sumbu X negatif global ($\theta = 540°$ atau 9,42 radian). Robot akan menuju ke koordinat X = 0, Y = 1, 6, $\theta = 540°$ atau 7,85 radian. Robot akan bergerak menuju titik tujuan dengan orientasi berubah menghadap ke titik tujuan selama perjalanan. Hasil pergerakan robot pada lintasan kelima dapat dilihat pada Gambar 19.



Gambar 20. Data posisi dan orientasi robot berdasarkan data *encoder*

Tabel 2. *Error* dari posisi (*X, Y*) dan orientasi (θ) robot berdasarkan pengamatan visual

Set-point	*Error X* (%)	*Error Y* (%)	*Error θ* (%)	Rata-rata (%)
1	12,94	7,79	0	6,91
2	12,85	15,7	0,78	9,776
3	15,93	18,69	1,48	12,03
4	24,5	15,3	3,88	14,56
5	1,82	8,35	3,95	4,71

Setelah dilakukan pengamatan secara visual, selanjutnya adalah dilakukan analisis dari hasil pembacaan sensor *encoder.* Dalam penelitian ini, pengamatan visual menggunakan kamera tidak memberikan *feedback* terhadap robot itu sendiri. Gambar 20 menunjukkan grafik informasi posisi dan orientasi robot berdasarkan pembacaan dari sensor *encoder*. Garis tidak putus (—) berwarna biru menandakan posisi robot di koordinat X global dalam satuan meter, garis titik-titik (. . .) berwarna merah menandakan posisi robot di koordinat Y global dalam satuan meter, dan garis putus-putus (- - -) berwarna jingga menandakan orientasi muka robot terhadap sumbu X positif global (θ) dalam satuan radian. Data ini didapatkan dari hasil pembacaan sensor *encoder* yang diubah menjadi nilai kecepatan setiap motor di dalam mikrokontroler. Setelah itu, nilai kecepatan tersebut diolah menggunakan persamaan kinematika (2) untuk menentukan kecepatan robot dan diterapkan persamaan matrik rotasi lokal ke global sehingga menampilkan grafik tersebut.

Tujuan *set-point* pertama adalah robot berjalan pada sumbu X sejauh 2 meter (X = 2, Y = 0, θ = 0) yang dapat dicapai dalam waktu 12 detik (0-12),

set-point kedua (X = 2, Y = 0, θ = 9,42) dapat dicapai dalam waktu 11 detik (12-23), *set-point* ketiga (X = 2, Y = 1, 2, θ = 9,42) dicapai dalam waktu 7 detik (23- 30), *set-point* keempat (X = 0, Y = 0, θ = 9,42) dicapai dalam waktu 15 detik (30-45) dan *set-point* terakhir (X = 1, 6, Y = 0, θ = 7,85) dicapai dalam waktu 10 detik (45-55).

Dari hasil grafik tersebut, kecepatan setiap motor DC dapat dikendalikan dengan baik dan sesuai dengan masukan yang diberikan. Namun, jika melihat pergerakan robot secara visual, terdapat pegerakan yang melenceng dari jalur yang seharusnya. Untuk mengetahui *error* antara posisi aktual secara visual dan posisi yang diinginkan maka digunakan (6).

$$error = \frac{nilai_{visual} - nilai_{target}}{nilai_{target}} 100\% \quad (6)$$

Dari Tabel 2, didapatkan nilai rata-rata *error* keseluruhan *set-point* adalah 9,6%. Hal ini terdapat beberapa kemungkinan, diantaranya faktor lapangan yang licin dan juga nat keramik yang tidak rata dan juga belum terdapat umpan balik posisi dan orientasi dari kerangka koordinat global sehingga pergerakan robot sedikit bergeser dari semestinya. Namun secara umum, robot beroda omni yang telah dirancang secara simulasi dan dilakukan secara eksperimen dapat berfungsi dengan baik dan bergerak menuju posisi dan orientasi yang telah ditentukan.

IV. KESIMPULAN

Pada penelitian ini telah dirancang dan dibangun sebuah kendali kecepatan robot beroda omni dengan kemampuan dapat menuju posisi dan orientasi yang diinginkan secara simulasi dan eksperimen berbasis pengendali PID. Berdasarkan hasil PID *tuner* didapatkan nilai Kp sebesar 0,48, nilai Ki sebesar 11,16 dan nilai Kd sebesar 0. Hasil pengujian secara simulasi menunjukkan bahwa robot beroda omni dapat bermanuver berdasarkan kinematika yang telah dirancang sebelumnya dan dapat menuju ke *set-point* dan orientasi yang telah ditentukan. Sementara itu, hasil pengujian secara eksperimental berdasarkan data pembacaan *encoder* menunjukkan robot beroda omni telah berhasil menuju posisi dan orientasi yang diinginkan, namun berdasarkan pengamatan secara visual terdapat *error* sebesar 9,6% dari keseluruhan *set-point* yang disebabkan oleh faktor teknis lapangan. Saran untuk pengembangan selanjutnya adalah memperkaya dengan beberapa algoritma dan menggunakan kamera sebagai umpan balik posisi robot dalam kerangka koordinat global serta membuat lapangan se-ideal mungkin untuk menghindari *error* yang disebabkan masalah teknis.

REFERENSI

[1] K. V. Ignatiev, M. M. Kopichev, and A. V Putov, "Autonomous Omni-Wheeled Mobile Robots," in *2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2016, pp. 0–3.

[2] A. Dwi, R. Aljabar, and M. Rivai, "Rancang Bangun Robot Omni untuk Mencari Sumber Gas Menggunakan Kontrol Logika Fuzzy Berbasis Mikrokontroler ARM STM-32," *J. Tek. ITS*, vol. 7, no. 1, pp. 2–7, 2018.

[3] Z. Abidin, "Omni-Wheeled Robot with Rapidly-exploring Random Tree (RRT) Algorithm for Path Planning," in *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, 2019, pp. 288–292.

[4] R. Risfendra, A. Asfinaldi, H. Habibullah, and J. Julisardi, "Sistem Pergerakan Robot Kiper Beroda Menggunakan Metode Wall Follower Berbasis Image Processing," *Elkha*, vol. 12, no. 1, pp. 1, 2020.

[5] P. I. Chang, Y. S. Shi, and C. Y. Lan, "SLAM Configuration from Video Images for Remote Omni- direction Vehicle Platform," *2020 Int. Conf. Adv. Robot. Intell. Syst.*, pp. 4–7, 2020.

[6] A. Birari, A. Kharat, P. Joshi, R. Pakhare, U. Datar, and V. Khotre, "Velocity Control of Omni Drive Robot using PID Controller and Dual Feedback," in *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, 2016, pp. 295–299.

[7] A. Priyambudi and B. Firman, "Kendali Kecepatan Motor Pada Robot Dengan Empat Roda Omni Menggunakan Metode PID," *J. Teknol. Technoscientia*, vol. 10, pp. 54–63, 2018.

[8] X. Liu, H. Chen, C. Wang, and F. Hu, "MPC Control and Path Planning of Omni-directional Mobile Robot with Potential Field Method," *2018 IEEE Int. Conf. Real-time Comput. Robot.*, pp. 634–638, 2018.

[9] M. Gavani, D. Tanpure, and P. Falake, "Path Planning of Three Wheeled Omni-Directional Robot Using Bezier Curve Tracing Technique and PID control Algorithm," *2019 IEEE Pune Sect. Int. Conf. PuneCon 2019*, pp. 1–6, 2019.

[10] H. Muhammad and F. Al, "Rancang Bangun Sistem Positioning Mobile Robot Omnidirectional Wheel Menggunakan STM32 Berbasis Fuzzy Logic Controller," *J. Tek. Elektro*, vol. 10, pp. 547–555, 2021.

[11] O. I. Al-agha, "Robot Arm based on (ANFIS + PID) controller," in *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018, pp. 1–5.

[12] D. A. Maulana, "Penerapan Model Predictive Control (MPC) pada Desain Pengendalian Robot Mobil Beroda Empat," *Zeta - Math J.*, vol. 3, no. 2, pp. 46–51, 2017.

[13] G. Commutation, *DC-Micromotors Graphite Commutation Series 3863CR*. 2011.