

Kendali Robot Beroda Otonom dengan *Invers Kinematics*

Autonomous Wheeled Robot Control with Inverse Kinematics

MICHAEL JULYUS CHRISTOPHER MANULLANG¹, MEDRIA KUSUMA DEWI
HARDHIENATA¹, KARLISA PRIANDANA^{1*}

Abstrak

Penelitian ini dilakukan untuk mengembangkan robot otonom yang dikendalikan dengan pendekatan matematika *inverse kinematics*. Robot yang digunakan pada penelitian ini adalah robot beroda *nonholonomic differential drive*. Metode penelitian yang digunakan terdiri atas empat tahapan, yaitu: penentuan parameter robot beroda, pengembangan kontrol *inverse kinematics*, pembuatan data lintasan (*trajectory*) berupa garis lurus, dan pengujian sistem kendali. Pada pengujian, data *trajectory* yang dibangkitkan dibandingkan dengan pengukuran berdasarkan observasi di lapangan. Pengukuran data gerakan robot di lapangan dilakukan dengan dua alat, yaitu dengan *global positioning system* (GPS) yang terpasang pada robot dan GPS *smartphone*. Hasil pengujian menunjukkan bahwa robot beroda dapat dikendalikan dengan *inverse kinematics* dengan rata-rata nilai galat sebesar 0.9 meter.

Kata Kunci: *inverse kinematics*, kendali, otonom, robot beroda.

Abstract

This research aims to develop an autonomous robot that is controlled with a mathematical approach, inverse kinematics. The robot used in this research is a non-holonomic differential drive wheeled robot. The research method consists of four stages, namely: determination of the wheeled robot parameters, development of inverse kinematics control, generation of straight trajectory data, and control system testing. In testing, the generated trajectory data is compared with measurements based on observations in the field. The measurements is done by two equipment, namely the global positioning system (GPS) installed on the robot and the smartphone's GPS. The test results show that the wheeled robot can be controlled with inverse kinematics with an average error value of 0.9 meters.

Keywords: autonomous, inverse kinematics, control, wheeled robot

PENDAHULUAN

Robot beroda adalah robot yang dapat bergerak dan berpindah posisi menggunakan roda bermotor. Penelitian ini menggunakan jenis robot beroda karena secara luas dapat digunakan pada berbagai tugas otonom di bidang datar, misalnya untuk pengelolaan bencana tanah longsor (Liu dan Nejat 2013). Robot beroda yang digunakan yaitu robot beroda *nonholonomic*, di mana robot jenis ini tidak dapat bergerak ke segala arah, melainkan bergantung pada jalur yang dapat dicapainya (Laumond *et al.* 1998).

Sejak penemuannya, robot beroda telah dikembangkan dengan menggunakan berbagai sistem kontrol, baik pendekatan mekanik (Naveed dan Khan 2017) maupun matematika (Gu *et al.* 2017). Namun, hingga kini lingkungan robot yang kompleks dan non-linear tetap menjadi masalah utama. Metode kontrol konvensional, seperti *proportional-integral-derivative* (PID) dan *back-stepping controller* (Thao *et al.* 2010), tidak dapat memecahkan masalah ini karena

¹ Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor 16680.

* Penulis Korespondensi: Surel: karlisa@apps.ipb.ac.id

keterbatasannya dalam beradaptasi dengan karakteristik lingkungan yang berubah (Kusumoputro *et al.* 2016). Sistem kontrol adaptif dengan pendekatan matematika dapat menjadi solusi untuk tantangan ini, meskipun diperlukan penyesuaian pada saat diterapkan di dunia nyata (Almurib *et al.* 2011).

Penelitian ini menggunakan robot beroda dengan teknologi *differential drive* yang merupakan suatu teknik pergerakan robot dengan memanfaatkan perbedaan kecepatan dua roda. Sistem kendali yang diteliti adalah *inverse kinematics* yang merupakan sistem kendali berbasis pendekatan matematika. *Inverse kinematics* ini merupakan kebalikan dari pendekatan kinematika (*forward kinematics*) untuk memperoleh parameter berupa kecepatan masing-masing roda pada robot. Kecepatan roda diturunkan dari kecepatan robot pada koordinat global yang diperlukan untuk mencapai posisi akhir yang diinginkan. Sebelumnya, *inverse kinematics* sudah pernah digunakan untuk robot berkaki enam oleh Priandana *et al.* (2017). Akan tetapi, penelitian tersebut terbatas pada analisis akurasi waktu dan belum membahas akurasi perpindahan posisi robot. Selain itu, robot berkaki juga memiliki kelemahan yaitu lebih kompleks dikendalikan karena memiliki jumlah servo yang lebih banyak dibanding robot beroda. Sebelumnya, *inverse kinematics* juga sudah pernah digunakan oleh Maulana *et al.* (2014) pada robot beroda *differential drive*. Robot tersebut memiliki 4 roda (2 roda bermotor di bagian belakang, dan 2 roda bantu di bagian depan) dan *inverse kinematics* diimplementasikan untuk melacak jalur menggunakan sensor yang ada pada robot.

Penelitian ini dilakukan untuk mengendalikan robot beroda dengan pendekatan *inverse kinematics* serta menganalisis hasil pengendaliannya dari segi akurasi perubahan posisi robot. Pada penelitian ini, robot yang digunakan memiliki 3 roda (2 roda bermotor di bagian depan, dan 1 roda bantu di bagian belakang). *Inverse kinematics* diimplementasikan untuk lintasan buatan yang digunakan untuk menentukan arah pergerakan robot sehingga robot tersebut dapat berjalan secara otonom. Batasan dari penelitian ini adalah (a) fokus penelitian adalah penerapan algoritme *inverse kinematics* untuk mengendalikan robot beroda dengan parameter sesuai dengan robot yang digunakan, (b) lintasan (*trajectory*) yang dianalisis adalah *trajectory* garis lurus. Dengan kata lain, robot bergerak lurus dan tidak berbelok, dan (c) kecepatan motor yang dihasilkan oleh algoritme *inverse kinematics* tidak melebihi kecepatan maksimum motor yang digunakan, yaitu 50 rotasi per menit (RPM) atau 0.172 m/s.

METODE

Kinematika Robot Beroda *Differential Drive*

Pada robot beroda, persamaan kinematika (*forward kinematics*) mendefinisikan beberapa bagian dari pergerakan robot, di antaranya \dot{x} , \dot{y} , dan $\dot{\theta}$. Resultan antara \dot{x} , \dot{y} adalah $v(t)$. Vektor pergerakan robot beroda ditunjukkan pada Gambar 1 dan 2 (Siegwart dan Nourbakhsh 2004, dengan modifikasi). Peubah v merepresentasikan kecepatan linear, dan ω merepresentasikan kecepatan sudut robot. Persamaan untuk postur robot dinyatakan dalam bentuk seperti pada persamaan (1).

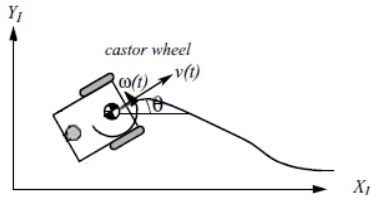
$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad (1)$$

Posisi P (pusat robot) pada koordinat global (G) dinyatakan dalam koordinat (x, y) dan perbedaan sudut antara koordinat global dengan koordinat robot (R) dinotasikan sebagai θ . Sementara itu, kecepatan robot beroda pada koordinat global dapat dinyatakan dalam bentuk seperti pada persamaan (2).

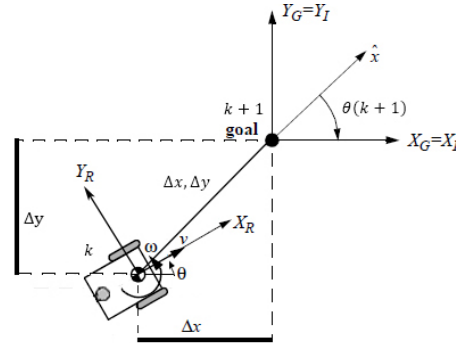
$$(2)$$

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix},$$

Peubah \dot{x} dan \dot{y} yang merupakan kecepatan robot pada sumbu x global (X_G) dan y global (Y_G), serta $\dot{\theta}$ yang merupakan kecepatan sudut robot.



Gambar 1 Vektor pergerakan robot beroda.



Gambar 2 Kinematika robot beroda dan lingkungannya dengan target tujuan.

Untuk mendeskripsikan gerakan robot dalam hal gerakan komponen gerakan di sepanjang kerangka referensi koordinat global (X_G, Y_G) untuk bergerak di sepanjang kerangka referensi koordinat lokal robot (X_R, Y_R) perlu dipetakan. Pemetaan ini dilakukan menggunakan matriks rotasi ortogonal ($R(\theta)$) seperti pada persamaan (3).

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Untuk mendapatkan kecepatan robot pada koordinat lokal robot ($\dot{\xi}_R$) dari kecepatan robot pada koordinat global ($\dot{\xi}_I$), persamaan (4) digunakan.

$$\dot{\xi}_R = R(\theta) \dot{\xi}_I. \tag{4}$$

Siegwart dan Nourbakhsh (2004) mendefinisikan dua asumsi pada roda robot yang digunakan untuk menurunkan persamaan kinematika dan *inverse kinematics*. Gambar 3 menunjukkan sebuah roda pada titik A yang terhubung dengan pusat (*origin*) robot P . Untuk satu roda, berlaku dua asumsi yang bergantung pada jenis roda yang digunakan. Untuk jenis roda standar tetap, kedua asumsi ini dinyatakan pada persamaan (5) dan (6).

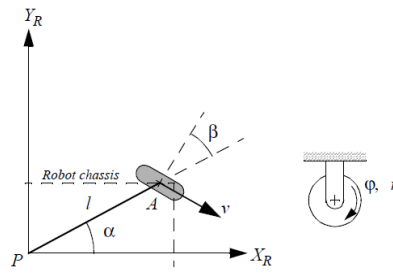
Rolling constraint (roda tidak slip)

$$[\sin(\alpha + \beta) \quad -\cos(\alpha + \beta) \quad (-l) \cos \beta] R(\theta) \dot{\xi}_I - r\dot{\phi} = 0. \tag{5}$$

No-sliding constraint (roda tidak lepas dari sumbunya)

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad l \sin \beta] R(\theta) \dot{\xi}_I = 0, \tag{6}$$

Peubah α adalah posisi (sudut) sumbu roda terhadap sumbu robot, β adalah posisi (sudut) setir terhadap sumbu roda, l adalah jarak antara pusat (*origin*) robot dengan pusat (*origin*) roda, r adalah jari-jari roda, dan $\dot{\phi}$ adalah kecepatan sudut roda.



Gambar 3 Roda standar tetap dan parameternya (Siegwart dan Nourbakhsh 2004)

Dari persamaan (5) dan (6) dapat diturunkan kinematika (*forward kinematics*) untuk robot beroda *differential drive* seperti yang ditunjukkan pada persamaan (7):

$$\dot{\xi}_1 = R(\theta)^{-1} \begin{bmatrix} \sin(\alpha+\beta) & -\cos(\alpha+\beta) & (-1) \cos \beta \\ \cos(\alpha+\beta) & \sin(\alpha+\beta) & 1 \sin \beta \end{bmatrix}^{-1} \begin{bmatrix} r\dot{\phi} \\ 0 \end{bmatrix}, \quad (7)$$

dengan $R(\theta)^{-1}$ adalah *inverse* dari matriks rotasi, $r\dot{\phi}_1$ adalah kecepatan translasi roda kanan, dan $r\dot{\phi}_2$ adalah kecepatan translasi roda kiri.

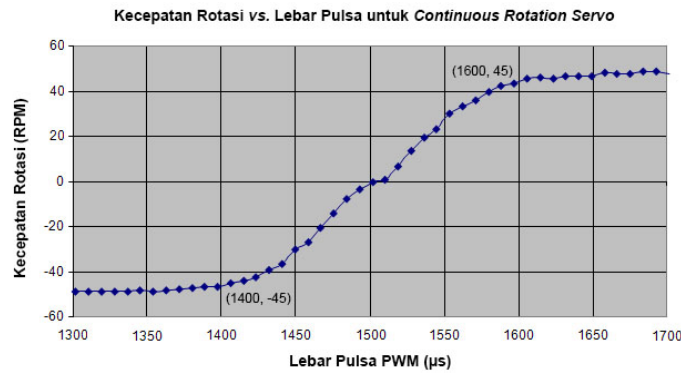
Persamaan (7) digunakan untuk mendapatkan kecepatan robot pada koordinat global dengan mengetahui kecepatan roda. Sebaliknya, kecepatan masing-masing roda dari kecepatan robot pada koordinat global dapat diperoleh dengan menggunakan persamaan *inverse kinematics*. Oleh karena itu, penelitian ini berkontribusi untuk menurunkan persamaan *inverse kinematics* untuk robot beroda *differential drive* yang berperan penting dalam pengendalian robot beroda tersebut.

Parallax Robot Shield with Arduino

Parallax Robot Shield with Arduino adalah robot beroda yang dapat diprogram menggunakan teknologi Arduino. Robot ini memiliki 2 roda standar tetap bermotor di bagian depan dan 1 roda bantu di bagian belakang. Kedua roda bermotor ini berguna untuk memberikan gaya dorong ke depan atau ke belakang. Perbedaan kecepatan kedua roda menciptakan torsi angular yang dimanfaatkan untuk berbelok (Ahmad *et al.* 2014).

Motor yang digunakan pada robot beroda ini berjenis *continuous rotation servo* yang dikendalikan dengan sinyal *pulse width modulation* (PWM). Hubungan lebar pulsa PWM dengan kecepatan dan arah rotasi motor servo ditunjukkan pada Gambar 4. Sumbu horizontal grafik menunjukkan lebar pulsa dalam mikrodetik (μs), dan sumbu vertikal menunjukkan respons servo dalam rotasi per menit (RPM). Untuk motor servo sebelah kanan, rotasi searah jarum jam ditunjukkan dengan nilai negatif, sedangkan arah berlawanan jarum jam ditunjukkan dengan nilai positif. Grafik servo (kurva transfer) ini berkisar dari sekitar -48 RPM hingga +48 RPM atau pada rentang lebar pulsa dari 1300 μs hingga 1700 μs . Motor servo berhenti berputar pada lebar pulsa 1500 μs . Ketika diberi pulsa PWM sebesar 1300 μs , motor servo sebelah kanan berputar searah jarum jam dengan kecepatan maksimum. Begitu juga untuk putaran berlawanan arah jarum jam pada lebar pulsa 1700 μs . Hal sebaliknya berlaku untuk motor servo sebelah kiri.

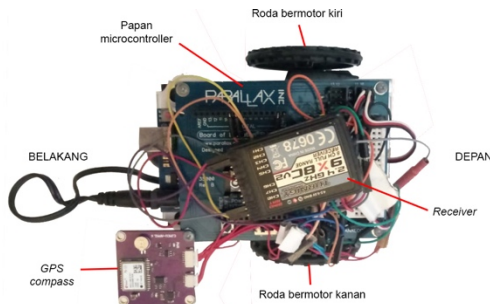
Kecepatan tidak banyak berubah pada lebar pulsa 1300 μs sampai 1400 μs dan pada lebar pulsa 1600 μs sampai 1700 μs . Sedangkan, pada lebar pulsa 1400 μs hingga 1600 μs , kontrol kecepatan berbentuk hampir linear. Dalam rentang ini, perubahan dalam lebar pulsa akan menghasilkan perubahan kecepatan yang sesuai. Oleh karena itu, lebar pulsa disarankan berada pada rentang 1400 μs hingga 1600 μs . Penelitian Abiyoga (2018) telah menggunakan robot beroda ini yang telah dilengkapi dengan GPS + sensor kompas (Gambar 5).



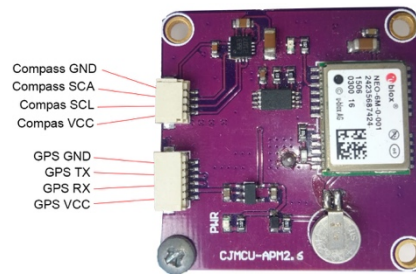
Gambar 4 Hubungan pulsa PWM dengan kecepatan servo (Lindsay 2016).

U-blox NEO-6M GPS Integrated with Honeywell Compass HMC5883L

U-blox NEO-6M GPS integrated with Honeywell compass HMC5883L (Gambar 6) adalah perangkat GPS yang terintegrasi sensor kompas. GPS merupakan sistem penentuan letak suatu objek di permukaan bumi dengan bantuan penyelarasan sinyal satelit. Setidaknya 4 satelit harus terhubung dengan suatu penerima di permukaan bumi agar dapat menentukan perkiraan koordinat posisi suatu objek (Hofmann-Wellenhof *et al.* 2012). Sementara itu, kompas yang terintegrasi dengan GPS ini berfungsi untuk mengetahui arah gerak suatu objek.



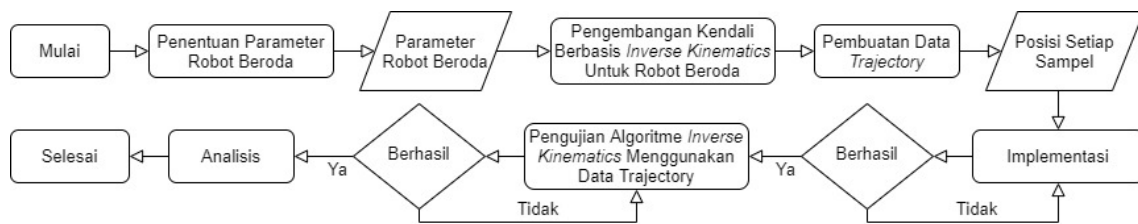
Gambar 11 Robot beroda yang digunakan.



Gambar 12 U-blox NEO-6M GPS integrated with Honeywell compass HMC5883L.

Tahapan Penelitian

Diagram alir tahapan penelitian ini ditunjukkan pada Gambar 7. Penjelasan masing-masing tahapan adalah sebagai berikut:

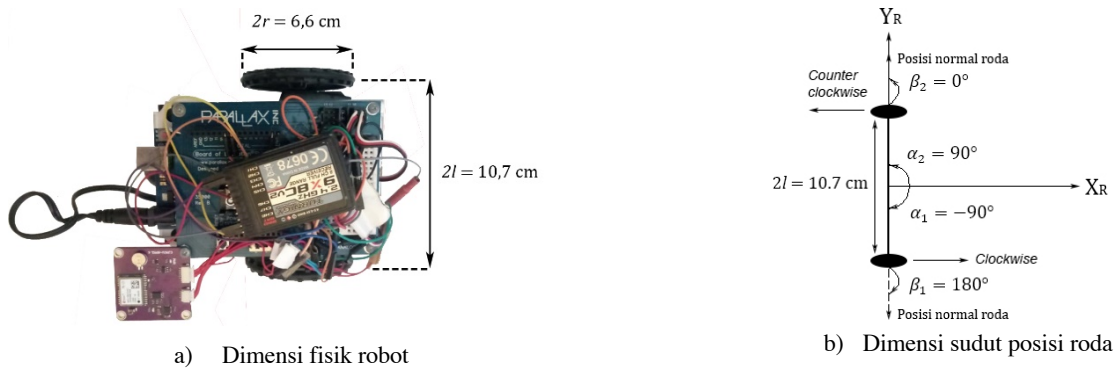


Gambar 7 Diagram alir tahapan penelitian.

Penentuan Parameter Robot Beroda

Tahapan ini dilakukan dengan cara mengukur parameter nyata dari robot beroda yang digunakan. Pada penelitian ini, nilai β untuk masing-masing roda pada robot tetap, karena robot yang digunakan tidak memiliki setir. Parameter robot dan ilustrasinya ditunjukkan pada Gambar 8, dengan penjelasan sebagai berikut:

- 1 jari-jari roda (r): 3.3 cm,
- 2 jarak antara pusat (*origin*) robot dengan pusat (*origin*) roda (l): 5.35 cm,
- 3 posisi (sudut) sumbu roda terhadap sumbu robot (α): $\alpha_1 = -90^\circ$, $\alpha_2 = 90^\circ$,
- 4 posisi (sudut) setir terhadap sumbu roda (β): $\beta_1 = 180^\circ$, $\beta_2 = 0^\circ$.



Gambar 13 Parameter robot beroda.

Pengembangan Kendali Berbasis Inverse Kinematics Untuk Robot Beroda

Nilai-nilai parameter robot yang telah diukur kemudian digunakan untuk menggantikan variabel pada persamaan (5) dan (6) sehingga dihasilkan persamaan (8).

$$\begin{bmatrix} \sin(-90^\circ+180^\circ) & -\cos(-90^\circ+180^\circ) & (-l) \cos 180^\circ \\ \sin(90^\circ+0^\circ) & -\cos(90^\circ+0^\circ) & (-l) \cos 0^\circ \\ \cos(-90^\circ+180^\circ) & \sin(-90^\circ+180^\circ) & l \sin 180^\circ \\ \cos(90^\circ+0^\circ) & \sin(90^\circ+0^\circ) & l \sin 0^\circ \end{bmatrix} R(\theta) \dot{\xi}_I = \begin{bmatrix} r\dot{\phi}_1 \\ r\dot{\phi}_2 \\ 0 \\ 0 \end{bmatrix}. \quad (8)$$

Berdasarkan persamaan (9) dan rumus kinematika untuk robot beroda *differential drive* pada persamaan (7), persamaan *inverse kinematics* untuk robot beroda dapat diturunkan seperti pada persamaan (10).

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} R(\theta) \dot{\xi}_I = \begin{bmatrix} r\dot{\phi}_1 \\ r\dot{\phi}_2 \\ 0 \\ 0 \end{bmatrix}. \quad (9)$$

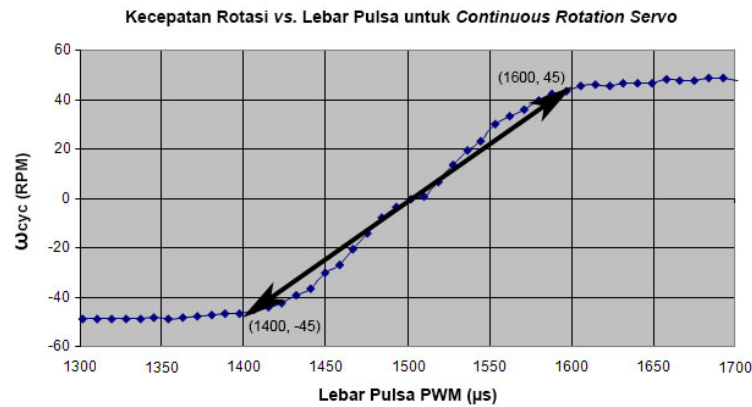
$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}, \quad (10)$$

dengan $r = 3.3$ cm dan $l = 5.35$ cm.

Hasil yang didapat dari persamaan (10) berupa kecepatan sudut roda kanan ($\dot{\phi}_1$) dan roda kiri ($\dot{\phi}_2$) dalam satuan radian per detik (rad/s). Hasil tersebut kemudian diubah menjadi kecepatan rotasi (ω_{cyc}) dengan satuan rotasi per menit (RPM) menggunakan persamaan (11):

$$\omega_{cyc}(\text{RPM}) = \frac{\dot{\phi} \text{ (rad/s)}}{2\pi} \times 60. \quad (11)$$

Setelah itu, kecepatan rotasi dikonversi kembali menjadi sinyal PWM menggunakan persamaan garis lurus dengan memperhatikan kecepatan motor yang digunakan pada Gambar 4 dengan mengasumsikan persamaan garis untuk 1400 μ s hingga 1600 μ s adalah linear dan ilustrasinya dapat dilihat pada Gambar 9 (Lindsay 2016, dengan modifikasi).



Gambar 14 Bentuk linear kecepatan motor.

Untuk memperoleh nilai pulsa dengan kecepatan yang sesuai, digunakan persamaan garis lurus yang diturunkan dari dua titik (x_1, y_1) dan (x_2, y_2) seperti persamaan (12):

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad (12)$$

Dari Gambar 9, maka $(x_1, y_1) = (1400, -45)$ dan $(x_2, y_2) = (1600, 45)$. Dengan melakukan substitusi pada persamaan (12) diperoleh persamaan (13):

$$x = \frac{y + 675}{0.45} \quad (13)$$

dengan x adalah parameter pulsa PWM dalam μs , dan y adalah kecepatan rotasi motor dalam RPM. Kemudian dari persamaan (13) dapat diturunkan persamaan (14) dan (15):

$$PWM_1 = \frac{(675 - \omega_{cyc})}{0.45} \quad (14)$$

$$PWM_2 = \frac{(\omega_{cyc} + 675)}{0.45} \quad (15)$$

dengan PWM_1 adalah sinyal kendali untuk roda kanan yang berputar searah jarum jam dan PWM_2 adalah sinyal kendali roda kiri yang berputar berlawanan arah jarum jam.

Pembuatan Data Trajectory

Langkah ini dilakukan dengan cara membuat data *trajectory* untuk menentukan arah gerak robot beroda. *Trajectory* yang dibangkitkan pada penelitian ini (Gambar 10) adalah *trajectory* garis lurus yang disesuaikan dengan kecepatan maksimum robot, dengan rincian sebagai berikut:

- 1 jumlah titik sampel: 10,
- 2 jarak tempuh setiap sampel: 0.71 meter pada sumbu x (Δx) dan 0.71 meter pada sumbu y (Δy),
- 3 orientasi robot pada setiap sampel (θ): 45° ,
- 4 *sampling time* (t): 7 detik.

Dari data *trajectory* tersebut, maka didapatkan kecepatan pada persamaan (16):

$$v = \sqrt{\frac{\Delta x^2 + \Delta y^2}{t}} = \sqrt{\frac{0.71^2 + 0.71^2}{7}} = 0.143 \text{ m/s.} \quad (16)$$

Nilai tersebut lebih rendah daripada kecepatan maksimum robot yaitu 0.172 m/s, artinya *trajectory* tersebut *feasible* untuk diterapkan. Berdasarkan *trajectory* tersebut, maka nilai-nilai \dot{x} , \dot{y} , dan $\dot{\theta}$ yang dimasukkan ke persamaan (10) adalah sebagai berikut:

$$1 \quad \dot{x} = \frac{\Delta x}{t} = \frac{0.71}{7} = 0.101 \text{ m/s,}$$

$$2 \quad \dot{y} = \frac{\Delta y}{t} = \frac{0.71}{7} = 0.101 \text{ m/s,}$$

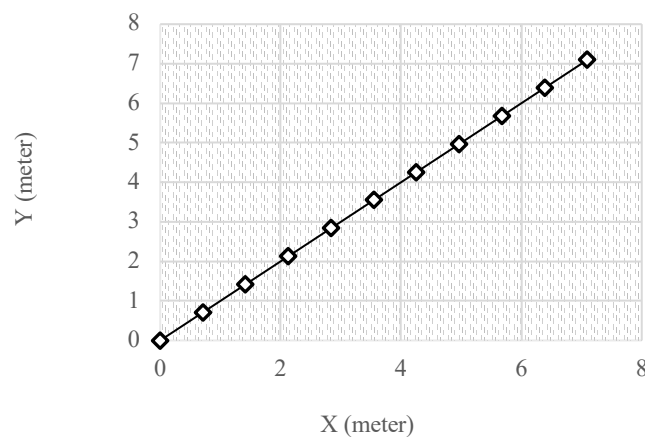
$$3 \quad \dot{\theta} = 0 \text{ rad/s, karena } \theta \text{ pada setiap sampel tetap.}$$

Implementasi

Tahapan ini dilakukan dengan cara menanamkan program kendali algoritme *inverse kinematics* untuk robot beroda ke mikrokontroler yang terpasang pada robot.

Pengujian Algoritme Inverse Kinematics menggunakan Data Trajectory

Pada tahap ini, program kendali dengan algoritme *inverse kinematics* yang telah dibuat ditanamkan pada robot beroda menggunakan Arduino. Program kendali ini diujikan untuk *trajectory* garis lurus yang ditunjukkan pada Gambar 10. Dengan demikian, robot beroda yang digunakan dapat berjalan secara otonom. Pengujian ini dilakukan sebanyak 15 kali di lapangan terbuka, yaitu 8 kali (pengujian 1-8) pada saat cuaca cerah dan 7 kali (pengujian 9-15) pada saat cuaca mendung. Untuk mengetahui pergerakan robot yang diujikan, posisi robot direkam dengan menggunakan tiga metode, yaitu (1) observasi di lapangan secara manual menggunakan meteran, (2) observasi menggunakan perangkat GPS yang terpasang pada robot, dan (3) observasi menggunakan GPS *smartphone*. Pada awalnya, posisi robot direkam hanya menggunakan GPS robot. Namun, karena galat yang dihasilkan dari perangkat tersebut cukup besar, GPS *smartphone* juga digunakan. Setelah dilakukan percobaan, ternyata GPS *smartphone* juga menghasilkan galat yang besar. Oleh karena itu, dilakukan pula pengukuran secara manual dengan cara mengukur posisi robot pada sumbu x dan y global menggunakan meteran pada setiap sampelnya (X_G, Y_G).



Gambar 15 Trajectory garis lurus

Analisis

Analisis dilakukan terhadap hasil pengujian yang mencakup perhitungan akurasi seberapa tepat robot bergerak mengikuti *trajectory* garis lurus yang ditunjukkan pada Gambar 10. Oleh karena itu, harus diketahui terlebih dahulu perubahan jarak robot dari titik awal dengan melakukan konversi pada data posisi robot yang telah didapatkan dari perangkat GPS robot, dan GPS *smartphone*. Data *longitude* dan *latitude* dari GPS ini diolah menjadi data posisi dengan menggunakan konvensi *east north up* (ENU), yaitu arah sumbu *East* yang bersesuaian dengan penambahan posisi *longitude* adalah arah sumbu x positif, arah sumbu *North* yang

bersesuaian dengan penambahan posisi *latitude* adalah sumbu *y* positif, dan arah ke atas (*Up*) adalah sumbu *z* positif (Priandana 2017). Posisi robot pada sumbu *x* dan *y* global diperoleh dengan asumsi bahwa posisi awal robot $(x_R(0), y_R(0))$ adalah pada koordinat (0, 0), dengan persamaan (17) dan (18):

$$x_R(k) = (\text{longitude}(k) - \text{longitude}(0)) \times 111000,$$

$$y_R(k) = (\text{latitude}(k) - \text{latitude}(0)) \times 111000,$$

dengan *k* adalah indeks untuk sampel data yang diambil. Dengan persamaan tersebut akan diketahui perubahan jarak robot dari posisi awal dalam meter.

Metode yang digunakan untuk menghitung akurasi hasil pengujian yaitu *mean squared error* (MSE) yang dapat dianalogikan sebagai varian ditambah dengan kuadrat bias dari suatu model. Persamaan dari metode MSE ditunjukkan pada persamaan (19):

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (19)$$

Dengan Y_i = data *trajectory* pada sampel ke-*i*, \hat{Y}_i = data hasil pengujian sebenarnya pada sampel ke-*i*, dan *n* = banyaknya sampel data. Kinerja kendali semakin baik jika nilai MSE yang dihasilkan semakin mendekati nol.

HASIL DAN PEMBAHASAN

Kinerja kendali berdasarkan nilai MSE-nya ditunjukkan pada Tabel 1 dan perbandingannya ditunjukkan pada Gambar 11. Pada Tabel 1, pengujian 1-8 dilakukan pada saat cuaca cerah dan pengujian 9-15 dilakukan pada saat cuaca mendung. Nilai MSE menunjukkan galat antara posisi yang ditetapkan oleh *trajectory* pada Gambar 10 dengan posisi robot yang terukur, baik dari pengukuran berdasarkan observasi manual, perangkat GPS robot, maupun GPS *smartphone*. Hasil pergerakan robot dalam satuan meter pada setiap sampelnya dapat dilihat pada Gambar 12-14.

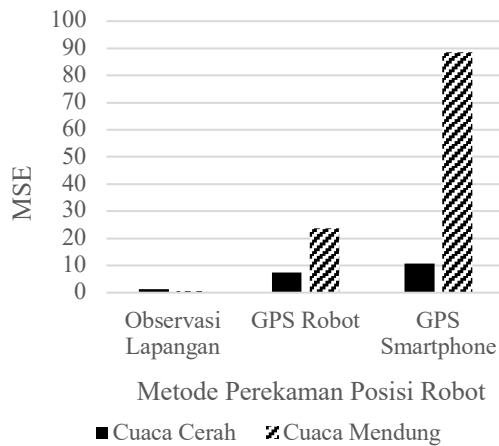
Berdasarkan hasil tersebut, terlihat bahwa *error* yang didapatkan dari pengukuran manual di lapangan lebih rendah dibandingkan dengan *error* yang didapatkan dari perangkat GPS. Selain itu, terlihat juga bahwa robot pada awalnya bergerak lurus mengikuti *trajectory*, namun pada akhirnya bergerak sedikit berbelok. Hal ini dapat terjadi karena adanya gesekan antara roda robot dengan kondisi tanah yang tidak rata sehingga gerakan robot menjadi sedikit berbelok. Sementara itu, untuk hasil yang diperoleh dari perangkat GPS, baik GPS robot maupun GPS *smartphone*, terlihat bahwa galat yang didapatkan pada cuaca cerah cukup rendah dibandingkan dengan galat pada cuaca mendung. Dengan demikian, dapat disimpulkan bahwa perangkat GPS sangat sensitif dengan cuaca. Perbedaan posisi yang terekam oleh kedua perangkat GPS ini mungkin disebabkan oleh perbedaan jenis GPS yang digunakan.

Berdasarkan pengujian yang telah dilakukan, hasil terbaik yang didapatkan yaitu pada pengujian ke-7, yaitu ketika pergerakan robot yang diperoleh dari pengukuran berdasarkan observasi di lapangan menghasilkan nilai galat relatif rendah, yaitu 0.2 meter dibanding observasi lainnya. Hasil terbaik juga diperoleh dari pengukuran berdasarkan observasi di lapangan dengan rata-rata nilai galat cukup rendah, yaitu 0.9 meter. Sementara itu, untuk hasil dari perangkat GPS, ternyata GPS robot lebih baik dibandingkan GPS *smartphone* dengan rata-rata nilai galat yaitu 15.0 meter untuk GPS robot, dan 47.0 meter untuk GPS *smartphone*.

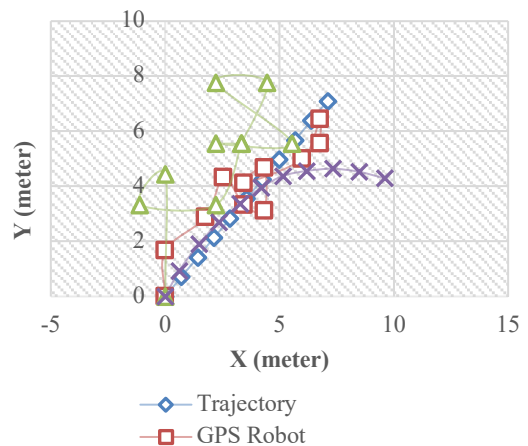
Tabel 2 MSE posisi robot

Pengujian	MSE Posisi pada Sumbu-X			MSE Posisi pada Sumbu-Y			MSE		
	A	B	C	A	B	C	A	B	C
1	2.0	10.1	7.0	1.9	3.7	6.3	2.0	6.9	6.6
2	1.4	10.3	4.4	2.9	21.1	7.8	2.2	15.7	6.1
3	1.1	1.9	3.3	1.5	8.2	2.8	1.3	5.0	3.0
4	1.1	2.5	2.5	1.0	5.2	1.1	1.0	3.9	1.8
5	1.1	0.4	3.5	0.6	1.0	5.6	0.8	0.7	4.6
6	0.5	2.4	15.8	0.4	1.3	28.1	0.4	1.8	21.9
7	0.3	21.7	21.8	0.1	11.3	18.5	0.2	16.5	20.1
8	1.3	1.9	20.3	2.0	15.8	22.1	1.7	8.8	21.2
9	1.0	4.8	62.6	1.9	3.2	45.5	1.5	4.0	54.0
10	0.8	2.6	12.0	0.4	10.8	205.3	0.6	6.7	108.6
11	0.2	2.8	47.1	0.3	0.4	39.0	0.3	1.6	43.0
12	0.2	11.9	49.1	0.3	5.0	167.6	0.3	8.5	108.3
13	0.1	1.9	82.9	0.3	5.5	18.5	0.2	3.7	50.7
14	0.1	5.1	52.5	0.3	237.1	32.0	0.2	121.1	42.2
15	0.2	38.0	364.8	0.3	2.7	60.7	0.2	20.4	212.8
Rata-rata	0.8	7.9	50.0	0.9	22.1	44.1	0.9	15.0	47.0
Standar deviasi	0.5	10.0	90.6	0.8	59.73	60.6	0.6	29.9	57.4

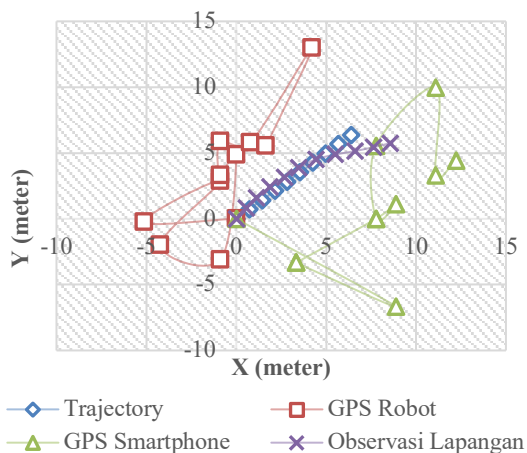
^aA adalah hasil yang diperoleh berdasarkan observasi di lapangan, B adalah hasil yang diperoleh dari GPS robot, C adalah hasil yang diperoleh dari GPS *smartphone*.



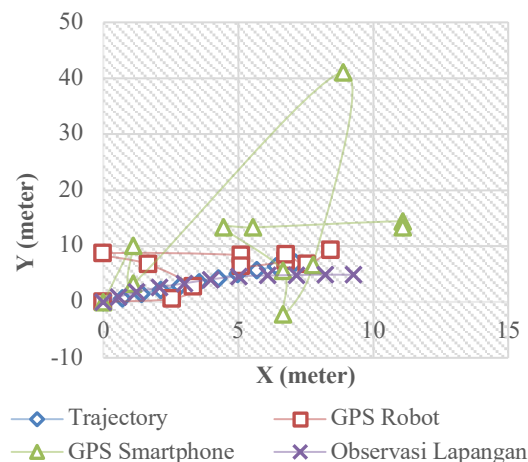
Gambar 16 Perbandingan MSE



Gambar 17 Contoh hasil pengujian ke-5 menggunakan data *trajectory* garis lurus



Gambar 18 Contoh hasil pengujian ke-7 menggunakan data *trajectory* garis lurus.



Gambar 19 Contoh hasil pengujian ke-10 menggunakan data *trajectory* garis lurus.

SIMPULAN

Penelitian mengenai kendali robot beroda otonom dengan *inverse kinematics* ini dikerjakan menggunakan metode yang terdiri atas enam tahapan, yaitu penentuan parameter robot beroda, pengembangan kendali berbasis *inverse kinematics* untuk robot beroda, pembuatan data *trajectory*, implementasi, pengujian algoritme *inverse kinematics* menggunakan data *trajectory*, dan analisis. Setelah sistem kendali ini dibuat dan diterapkan pada robot beroda, pengujian dilakukan menggunakan *trajectory* garis lurus yang kemudian direkam data pergerakannya dari pengukuran berdasarkan observasi di lapangan, GPS yang terpasang pada robot, dan GPS *smartphone*. Hasil pengujian menunjukkan bahwa robot beroda dapat berjalan mengikuti *trajectory* dengan rata-rata nilai galat yaitu 0.9 meter untuk pengukuran berdasarkan observasi di lapangan, 15.0 meter untuk GPS robot, dan 47.0 meter untuk GPS *smartphone*. Dengan demikian, dapat disimpulkan bahwa robot beroda yang digunakan dapat dikendalikan dengan *inverse kinematics* pada *trajectory* yang diujikan dengan galat relatif rendah berdasarkan hasil yang diperoleh dari pengukuran sesuai observasi di lapangan.

SARAN

Penelitian ini dapat diperbaiki atau dikembangkan dengan cara menggunakan teknik lokalisasi robot selain GPS agar diperoleh data posisi robot yang lebih akurat pada saat melakukan pengujian.

DAFTAR PUSTAKA

- Abiyoga Iqbal. 2018. Pengembangan sistem kendali berbasis kecerdasan komputasional dengan metode *backpropagation neural network* untuk robot beroda [skripsi]. Bogor (ID): Institut Pertanian Bogor
- Ahmad Omar, Ullah Irfan, Iqbal Jamshed. 2014. A multi-robot educational and research network. *International Journal of Academic Research*. 6(2):217-222.doi:10.7813/2075-4124.2014/6-2/ A.32
- Almurib HAF, Isa AAM, Al-Assadi HMAA. 2011. Direct neural network control via inverse modelling: Application on induction motors. *Artificial Neural Networks - Industrial and Control Engineering Applications*. doi:10.5772/16052.
- Gu J, Ma X, Liu F, Wang Y, Ren H. 2017. Mathematical modeling and intelligent algorithm for multi-robot path planning. *Mathematical Problems in Engineering*. doi:10.1155/2017/1465158.
- Hofmann-Wellenhof B, Lichtenegger H, Collins J. 2001. Global positioning system: theory and practice. Ed ke-5. New York (US): Springer-Verlag Wien.
- Kusumoputro B, Suprijono H, Heryanto MA, Suprpto BY. 2016. Development of an attitude control system of a heavy-lift hexacopter using Elman recurrent neural networks. *2016 22nd International Conference on Automation and Computing (ICAC)*. Colchester (UK): IEEE.
- Laumond JP, Sekhavat S, Lamiroux F. 1998. Guidelines in nonholonomic motion planning for mobile robots. Di dalam: Robot Motion Planning and Control, Laumond JP, editor. Volume ke-229. *Lecture Notes in Control and Information Sciences*. London (GB): Springer-Verlag Berlin Heidelberg. hlm 1-53.
- Lindsay Andy. 2016. Robotics with the BOE Shield-bot for Arduino. Versi ke-1.1. California (US): Parallax Inc.

- Liu Y, Nejat G. 2013. Robotic urban search and rescue: A survei from the control perspective. *Journal of Intelligent and Robotic Systems*. 72(2):147-165.doi:10.1007/s10846-013-9822-x.
- Maulana E, Muslim MA, Zainuri A. Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot. *2014 Electrical Power, Electronics, Communications, Controls, and Informatics Seminar (EECCIS)*. Malang (ID): IEEE.
- Naveed K, Khan ZH. 2017. Adaptive path tracking control design for a wheeled mobile robot. *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*. Beijing (CN): IEEE.
- Priandana K. 2017. Sistem kendali kapal-pengejar-nirawak berbasis *neural networks* menggunakan *ensemble kalman filter* sebagai prediktor informasi radar [disertasi]. Depok (ID): Universitas Indonesia.
- Priandana K, Buono A, Wulandari. 2017. Hexapod leg coordination using simple geometrical tripod-gait and inverse kinematics approach. *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. Bali (ID): IEEE.
- Siegwart R, Nourbakhsh IR. 2004. Introduction to autonomous mobile robots. London (GB). MIT Press.
- Thao NGM, Nghia DH, Phuc NH. 2010. A PID back-stepping controller for two-wheeled self-balancing robot. *International Forum on Strategic Technology 2010*. Ulsan (KR): IEEE.