

Sistem Navigasi Kursi Roda Otonom dengan Sensor Laser

Navigation System of Autonomous Wheelchair with Laser Sensor

Muhammad Afan Aljafar¹, Angga Rusdinar², Irwan Purnama³

^{1,2,3} Universitas Telkom, Bandung

¹muhammadafanaljafar@student.telkomuniversity.ac.id,

²anggarusdinar@telkomuniversity.ac.id, ³irwanp@telkomuniversity.ac.id

Abstrak

Sistem navigasi menjadi bagian terpenting pada *autonomous mobile robot* agar mampu bergerak secara mandiri. Sistem navigasi pada *autonomous mobile robot* diartikan sebagai kemampuan untuk memandu pergerakan dari suatu lokasi ke lokasi lain yang dituju melalui penentuan lokasi dan arah gerakannya. Pada penelitian ini, kursi roda merupakan *autonomous mobile robot*-nya menggunakan *navigation stack package* pada *framework ROS* sebagai sistem navigasinya. *navigation stack* merupakan konsep program jaringan *mobile robot*, dan menggunakan sensor LiDAR untuk melakukan navigasi juga melakukan pemetaan ruangan terlebih dahulu, serta menggunakan sistem kendali adaptif untuk kecepatan motor dc. Terdapat juga sistem yang tergabung di dalam *navigation stack package* untuk mengendalikan posisi kursi roda yaitu metode kendali *odometry*, yang memanfaatkan pemodelan fisik *differential drive mobile robot* untuk mendapatkan kecepatan linier dan sudut dari masing-masing roda penggerak, kemudian dari perolehan kecepatan tersebut didapatkan jarak perpindahan dan sudut perpindahan dari kursi roda. kursi roda menggunakan sensor laser LiDAR. Kursi roda bergerak secara otonom dengan aktuator motor dc yang kecepatannya terkontrol dengan sistem kendali PID dengan nilai konstanta $P = 0.16$, $I = 5.5$, dan $D = 0$ pada masing-masing mikrokontroler yang terhubung pada motor dc kanan dan kiri. Dengan sistem *navigation stack package*, sensor LIDAR dan pengendali PID pada kecepatan motor dc Kursi roda mampu berjalan secara otonom dengan pola lurus dari titik (0,0) menuju (2.9 , 0), berjalan lurus dari titik (2.3, -1.3) menuju (2.3, -3.3), dan berjalan secara otonom dengan pola bolak-balik dari titik awal (0,0) menuju titik kedua (29,0) dan kembali ke titik awal, dengan rentang kesalahan sebesar 0.58% - 98.99%.

Kata Kunci : *Autonomous Navigation, LIDAR, ROS, Odometry, SLAM, PID.*

Abstract

The navigation system is the most important part of an autonomous mobile robot to be able to move independently. The navigation system on an autonomous mobile robot is defined as the ability to guide movement from one location to another by determining the location and direction of its movement. In this study, a wheelchair is an autonomous mobile robot using a navigation stack package in the ROS framework as a navigation system. The navigation stack is a mobile robot network program concept, and uses LiDAR sensors to navigate as well as to map the room first, and use an adaptive control system for dc motor speed. There is also a system incorporated in the navigation stack package to control the position of the wheelchair, namely the odometry control method, which utilizes the physical modeling of a mobile robot's differential drive to obtain the linear and angular velocity of each drive wheel, then from the speed gain, the displacement distance and angle are obtained. the displacement angle of the wheelchair. wheelchair using a LiDAR laser sensor. The wheelchair moves autonomously with a dc motor actuator whose speed is controlled by a PID control system with a constant value of $P = 0.16$, $I = 5.5$, and $D = 0$ on each of the microcontrollers connected to the right and left dc motors. With the navigation stack package system, LIDAR sensor, and PID controller at the speed of a dc motor, the wheelchair can run autonomously in a straight pattern from point (0.0) to (2.9, 0), running straight from point (2.3, -1.3) to (2.3, -3.3), and runs autonomously with a back and forth pattern from the starting point (0,0) to the second point (29.0) and back to the starting point, with an error range of 0.58% - 98.99%.

Keywords: *Autonomous Navigation, LIDAR, ROS, Odometry, SLAM, PID.*

1. Pendahuluan

Para penyandang disabilitas terkhusus yang terbatas kemampuan Bergeraknya sangat terbantu dengan adanya kursi roda. Seiring perkembangan zaman dan teknologi, perkembangan teknologi pada kursi roda pun semakin canggih. Setelah adanya kursi roda elektrik, Beberapa tahun terakhir banyak badan penelitian nasional atau swasta telah melakukan penelitian yang diaplikasikan pada kursi roda elektrik untuk semakin mempermudah penggunaannya.

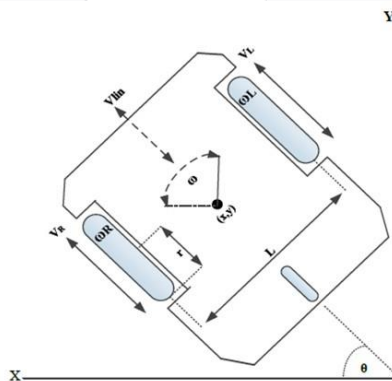
Perkembangan industri 4.0 saat ini mendorong adanya kemajuan teknologi dan ilmu pengetahuan yang diciptakan oleh manusia telah sampai pada zaman *autonomous mobile robot*. Pengaplikasian autonomus *mobile robot* akan diterapkan pada kursi roda, dengan tujuan kursi roda tersebut mampu berjalan sendiri kembali ke *charging station* untuk pengisian ulang daya, tentu saja dengan ketentuan dalam satu ruangan tertutup lokasi *charging station* sebagai tujuannya telah ditentukan. Sistem navigasi pada *autonomous mobile robot* dapat diartikan sebagai suatu kemampuan untuk memandu pergerakan dari suatu lokasi ke lokasi lain yang dituju melalui penentuan lokasi dan arah gerakannya. *Mobile robot* yang dibuat merupakan kursi roda dengan model kinematika *differential drive* dan dilengkapi dengan sensor *incremental rotary encoder* yang akan menghitung jumlah putaran roda dari kursi roda kemudian digunakan untuk menghitung posisi sekaligus perpindahan dari kursi roda, termasuk kecepatan serta percepatannya dengan kendali posisi menggunakan metode *odometry* dan kendali kecepatan adaptif untuk motor dc pada sistem *mobile robot* [2] dan sensor *Light Detection and Ranging* (LiDAR) yang mengukur juga mendeteksi keadaan lingkungan dengan rentang 360 derajat pada ketinggian yang sama dan juga menghasilkan pemetaan dua dimensi dari lingkungan sekitar kursi roda dengan data yang diperoleh sensor LiDAR dan digunakan untuk algoritma *Simultaneous Localization And Mapping* (SLAM) [3].

Dalam pengaplikasiannya, *autonomous mobile robot* ini menggabungkan dua umpan balik yang diperoleh dari sensor *incremental rotary encoder* dan *Light Detection and Ranging* (LiDAR) yang kemudian diolah didalam ROS *navigation stack package* pada robot operating system (ROS). ROS adalah sistem operasi robot bersifat *open source* yang didalamnya terdapat *library* dan *tools* untuk membuat perangkat lunak robot [4]. Dengan data umpan balik yang didapatkan dari sensor dan kemudian diproses pada ROS, sebagai hasil keluarannya kursi roda elektrik dapat menuju *charging station* dengan sekali perintah menentukan titik tujuan.

2. Dasar Teori

2.1. Pemodelan Matematika dari Mobile Kinematics (kursi roda)

Pemodelan matematis kinematika *mobile robot* adalah salah satu peran utama sistem kontrol dalam desain robot. Pada gambar 2.1 menunjukkan pemodelan *mobile robot* penggerak diferensial dengan parameter yang dibutuhkan.



Gambar 2.1 Pemodelan kinematik *differential drive mobile robot* [2].

Parameter *mobile robot* berdasarkan model kinematik dapat ditulis sebagai berikut [2]:

V_R, V_L : Kecepatan roda kanan dan roda kiri masing-masing

X, Y : Posisi relatif robot dalam koordinat inersia frame

θ : *Pose relative* atau *relative heading robot* dalam bingkai inersia

V_{lin} : Kecepatan linier yang ditargetkan dari *mobile robot* (kursi roda)

ω : Kecepatan sudut yang ditargetkan dari *mobile robot* (kursi roda)

L : Jarak antara roda kiri dan kanan (*wheelbase*)

Model kinematik mengambil kecepatan robot dan mengubahnya menjadi vektor koordinat umum dengan persamaan berikut [2]:

$$\begin{aligned}x_t &= V_{lin} \cos \theta \\ \{y_t &= V_{lin} \sin \theta \\ \theta_t &= \omega\end{aligned}\tag{2.1}$$

$$\begin{aligned}x_t + \Delta &= x_t + \Delta V_{lin} \cos \theta \\ \{y_t + \Delta &= y_t + \Delta V_{lin} \sin \theta \\ \theta_t + \Delta &= \theta_t + \Delta \omega\end{aligned}\tag{2.2}$$

Kemudian diambil kecepatan robot dari masing-masing kecepatan roda dengan persamaan sebagai berikut:

$$V_{lin} = \frac{r(\omega_R + \omega_L)}{2}\tag{2.3}$$

$$\omega = \frac{r(\omega_R - \omega_L)}{L}\tag{2.4}$$

Parameter di atas dari *mobile robot* (kursi roda) differential drive berguna sebagai masukan dari model kinematika dasar robot [2].

2.2. ROS Navigation Stack Package

Navigation stack merupakan konsep program jaringan *mobile robot* untuk berpindah dari satu tempat ke tempat yang lain. *ROS Navigation stack* sangat tepat diimplementasikan pada *mobile robot* untuk berpindah dari satu tempat ke tempat lain dengan andal. Tugas *navigation stack* adalah menghasilkan jalur yang aman untuk dijalankan robot, dengan mengolah data dari odometri, sensor, dan peta lingkungan. Memaksimalkan kinerja *navigation stack* ini memerlukan beberapa penyetelan parameter yang baik, dan ini tidak sesederhana kelihatannya [5].

2.3. Sistem Pemetaan

Dalam membangun suatu sistem *autonomous mobile robot*. Pemetaan adalah sebuah komponen utama dengan sistem penggambaran atau pembacaan dari permukaan suatu objek yang diperkecil dengan menggunakan skala untuk mempresentasikan ukuran peta yang tergambar dengan ukuran peta sebenarnya [7]. Proses dalam pemetaan mencakup proses pengukuran, perhitungan dan penggambaran suatu objek yang dipetakan. Dalam pembuatan sebuah peta 2D ruangan dibutuhkan sensor yang mendukung untuk melakukan proses navigasi serta mikrokontroler untuk dihubungkan ke pc dan dalam pembuatan *source code* dibantu dengan *library SLAM (Simultaneous Localization and Mapping)* untuk pembacaan objek – objek yang ada di ruangan dan sebuah *framework* bernama *ROS (Robot Operating System)* untuk menampilkan peta yang dibuat [8].

2.4. Implementasi Sistem kendali PID Pada Kecepatan Motor DC

Kontroler PID digunakan untuk mengontrol kecepatan putar motor dc dengan bantuan nilai konstanta KP, KD dan KI. KP bergantung pada eror masukan, KI bergantung pada akumulasi eror masa lalu atau sebelumnya, dan KD adalah prediksi eror di masa mendatang. Oleh karena itu, nilai konstanta KP, KD dan KI harus diselaraskan secara tepat dengan respon sistem dari hasil pembacaan sensornya[2].

2.5. Metode Odometry Untuk Kendali Posisi

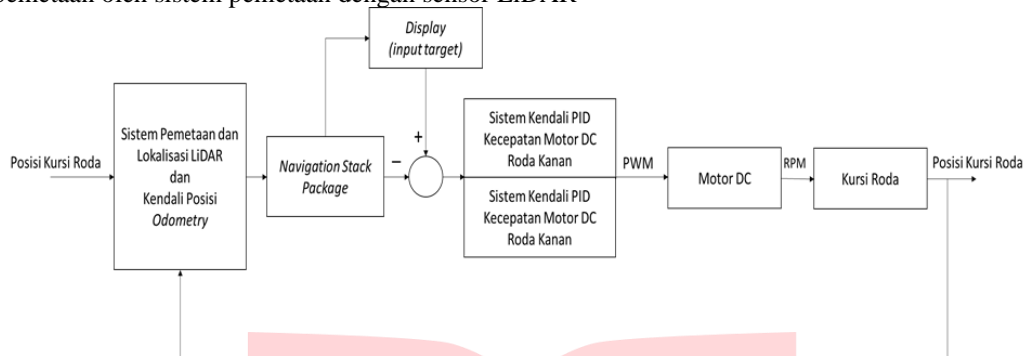
Odometry adalah penggunaan data dari pergerakan aktuator untuk memperkirakan perubahan posisi dari waktu ke waktu. *odometry* digunakan untuk memperkirakan posisi relatif terhadap posisi awal [13]. *Odometry* didasarkan pada persamaan sederhana yang mudah diterapkan dan menggunakan data dari *wheel encoders*. Namun, *odometry* juga didasarkan pada asumsi bahwa putaran roda dapat diterjemahkan ke dalam perpindahan linier relatif terhadap lantai. Asumsi ini hanya validitas terbatas. Salah satu contoh ekstrem adalah selip roda: jika satu roda tergelincir, katakanlah, tumpahan minyak, maka *encoder* terkait akan mencatat putaran roda meskipun putaran ini tidak sesuai dengan perpindahan linier roda [2].

3. Perancangan Sistem

3.1. Diagram Blok

Sistem dimulai dengan keadaan kursi roda diam di satu titik posisi dengan pemetaan yang sudah selesai, peta 2D keadaan ruangan telah disimpan dan dijadikan masukan untuk sistem navigasi dan peta 2D ditampilkan pada *display monitor* yang juga menunjukkan perkiraan posisi kursi roda pada *layout 2D*. Kemudian sistem navigasi diberikan masukan target perkiraan titik tujuan dengan menggunakan fitur interaksi pada *ROS Vizualitation (RViz)*. Setelah target perkiraan titik tujuan dimasukan, sistem navigasi memprosesnya merencanakan jalur navigasi (*path following*) dan

memberikan keluaran nilai kecepatan sudut motor dc untuk dijadikan nilai masukan setpoint kendali adaptif kecepatan motor dc dan diubah menjadi nilai PWM untuk *driver* motor dc dan menjadi kecepatan sudut motor dc, masing-masing motor dc kanan dan kiri yang dikendalikan oleh pengendali adaptif yang disesuaikan dan sambil berjalan melakukan navigasi, kursi roda juga memperbarui data posisinya dengan kendali posisi *odometry* bersamaan dengan pembaruan pemetaan oleh sistem pemetaan dengan sensor LiDAR



Gambar 3.1 Blok Diagram

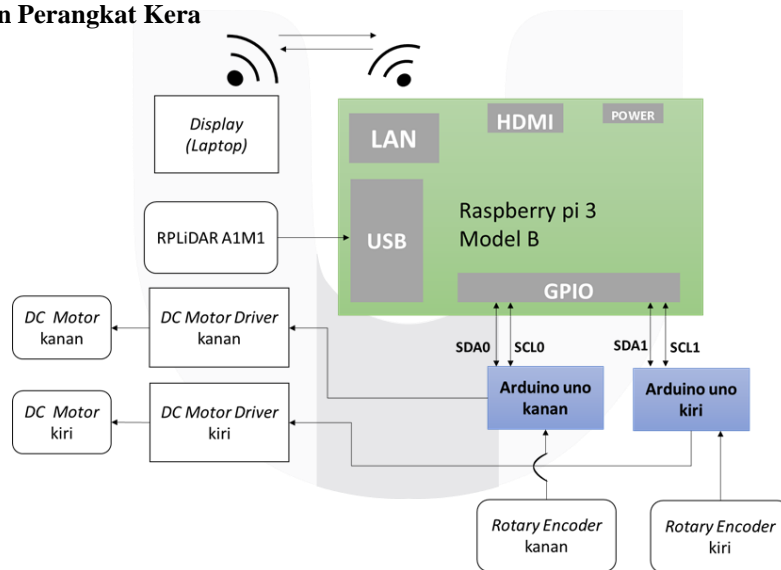
3.2. Kebutuhan Sistem

Program yang digunakan pada pembuatan sistem navigasi kursi roda otonom adalah sebagai berikut:

Sistem Operasi : Ubuntu 18.04

Software : framework Robot Operating System (ROS)
Version distro Melodic

3.3. Desain Perangkat Keras



Gambar 3.2. Perancangan Rangkaian Perangkat keras

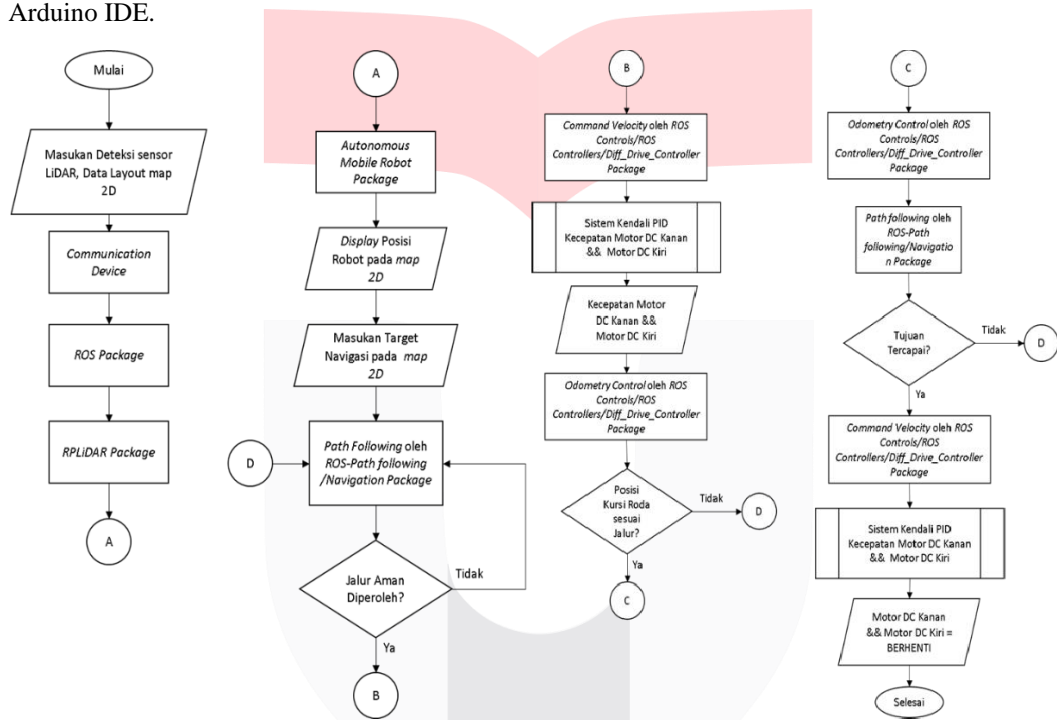
Dapat dilihat pada gambar 3.3 adalah diagram blok perancangan perangkat keras sistem navigasi kursi roda otonom. RPLiDAR A1M1 dihubungkan ke port USB Raspberry pi, Dengan USB adapter untuk RPLiDAR A1M1. Untuk melakukan komunikasi antara Raspberry pi 3 model B dan Arduino uno dengan protokol komunikasi *Inter Integrated Circuit* (I2C), maka pin SDA dan SCL pada Arduino kiri dihubungkan dengan pin SDA1 dan SCL1 pada Raspberry pi, dan juga pin SDA dan SCL pada Arduino kanan dihubungkan dengan pin SDA0 dan SCL0 pada Raspberry pi. Sensor rotary encoder dihubungkan ke Arduino uno dengan menghubungkan kabel hitam(out A) rotary encoder dengan pin 2(interrupt 0) Arduino, kabel putih(out B) dengan pin 4, kabel coklat(+VCC) dengan pin 5V, kabel biru(0V/GND) dengan pin GND(Ground). Hal tersebut dilakukan masing-masing untuk Arduino kanan dengan sensor rotary encoder kanan dan Arduino kiri dengan sensor rotary encoder kiri. Untuk menggerakkan motor dc, maka Arduino uno dihubungkan ke motor driver H-Bridge IBT2 dengan menghubungkan pin R_EN dan L_EN pada motor driver ke pin 5V pada

Arduino, pin RPWM ke pin 9, pin LPWM ke pin 10, pin VCC ke pin 5V, dan pin GND ke pin GND(Ground). Hal tersebut dilakukan masing-masing untuk Arduino kanan dengan motor driver H-Bridge IBT2 kanan dan Arduino kiri dengan motor driver H-Bridge IBT2 kiri. Arduino dihubungkan ke motor driver H-Bridge IBT2 untuk mengirimkan besar sinyal PWM yang telah diproses dengan sistem kendali adaptif yang disesuaikan untuk menggerakkan motor dc. Motor driver H-Bridge IBT2 dihubungkan dengan motor dc dengan menghubungkan kabel merah(Vin) motor dc dengan port M+ pada port terminal motor driver H-Bridge IBT2, dan kabel hitam (Ground) dengan port M-. Hal tersebut dilakukan masing-masing untuk motor driver H-Bridge IBT2 kanan dengan motor dc kanan dan motor driver H-Bridge IBT2 kiri dengan motor dc kiri.

Catatan: keterangan kanan dan kiri menunjukkan hal tersebut terhubung untuk roda bagian kanan dan kiri, masing-masing motor dc kanan dan kiri terpasang ke roda dengan gear box sesuai dengan setelan awal yang diberikan oleh perusahaan kursi roda listrik, dan sensor rotary encoder dipasang pada titik pusat atau center roda.

3.4. Desain Perangkat Lunak

Gambar 3.6 merupakan diagram alir dari algoritma program. Pada alat yang telah dirancang digunakan aplikasi untuk merealisasi perancangan tersebut menggunakan aplikasi ROS Melodic dan Arduino IDE.



Gambar 3.3. Diagram Alir

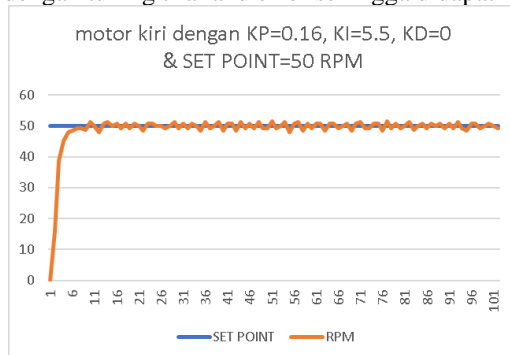
Program dimulai dengan pemetaan menggunakan sensor LiDAR untuk melakukan pemetaan ruangan dan membuat peta ruangan *layout 2D* dengan data jarak serta sudut yang sesuai dengan keadaan ruangan. Data peta ruangan berupa *layout 2D* disimpan untuk dimasukkan dan digunakan di program *navigation stack package*. Program *navigation stack package* dijalankan untuk memproses data peta ruangan dan mengintegrasikannya dengan data kecepatan kursi roda dari program kendali adaptif kecepatan motor dc (dengan nilai PID yang telah ditetapkan melalui proses tuning) yang kemudian mendapatkan data untuk kendali posisi *odometry* kursi roda. Navigasi kursi roda berjalan dengan memberikan masukan target berupa menentukan titik tujuan pada display interkasi RViz, kemudian program *navigation stack* melakukan path following untuk menentukan rute yang akan dilewati. Setelah rute diperoleh, kursi roda pun berjalan sesuai dengan rute tersebut dan dengan kecepatan yang terkendali. Selagi kursi roda berjalan program terus memproses data posisi *odometry* kursi roda. Keseluruhan program terus berjalan secara berulang hingga kursi roda sampai pada titik tujuan.

4. Hasil dan Analisis

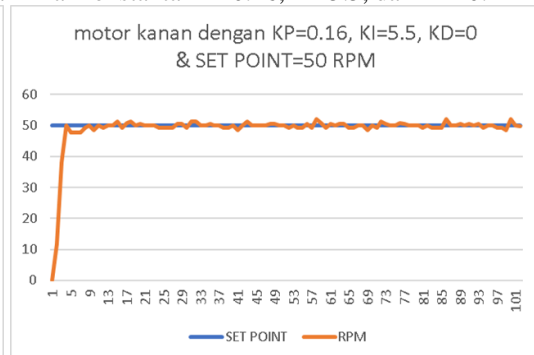
4.1. Tuning Sistem Kendali PID

Pada kendali PID kecepatan motor dc, tuning untuk nilai konstanta Proportional (P), Integral (I), Derivative (D) harus dilakukan dan ditentukan terlebih dahulu untuk mendapatkan respon sistem

yang sesuai dengan yang diinginkan. Nilai konstanta P, I, dan D pada penelitian ini didapatkan dengan tuning trial and error sehingga didapatkan nilai konstanta $P = 0.16$, $I = 5.5$, dan $D = 0$.



Gambar 4.1. Respon Sistem Hasil Tuning Kendali Kecepatan Motor DC kiri dengan setpoint 50 RPM



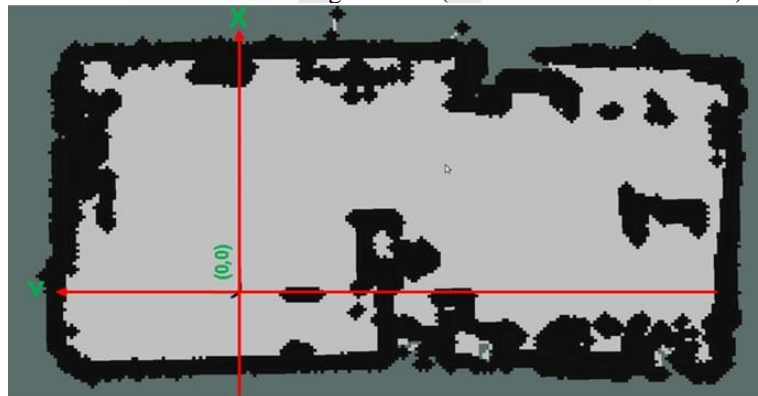
Gambar 4.2. Respon Sistem Hasil Tuning Kendali Kecepatan Motor DC kanan dengan setpoint 50 RPM

4.2. Pengujian dan Pengambilan Layout Peta

Pengujian ini dilakukan untuk mendapatkan *layout* peta ruangan 2D yang akan digunakan pada sistem navigasi robot sekaligus mendapatkan titik referensi untuk lokalisasi kursi roda pada ruang lingkup dalam ruangan N315 (INACOS LABORATORY). *Layout* map ruangan diinterpretasikan ke dalam koordinat kartesius (x,y). hasil pengukuran luas ruangan didapatkan adalah 10 meter x 5.9 meter atau 59 m².



Gambar 4.3. Keadaan Ruangan N315 (INACOS LABORATORY)



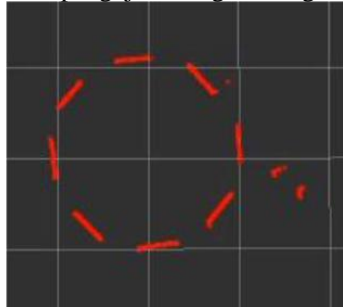
Gambar 4.4. *Layout Map* hasil *mapping* sensor LiDAR.

dan hasil pengukuran dari *layout map* dengan nilai terdekat daripada hasil pengukuran manual yang didapat dengan proses *mapping* sensor LiDAR dengan metode *mapping* menggunakan *library package* Hector SLAM, ukuran luas ruangan didapatkan 10,3 meter x 5,4 meter atau 55,6 meter². didapat dan ditentukan titik referensi untuk lokalisasi kursi roda adalah titik (0,0) seperti yang tertera pada gambar 4.5 *layout map* hasil *mapping* sensor LiDAR. Perbandingan hasil ukur luas ruangan dengan pengukuran manual dibandingkan dengan pengukuran luas ruangan dengan program ROS Rviz yang didapatkan tidak bersifat linier, oleh karena itu *layout map* yang dipakai

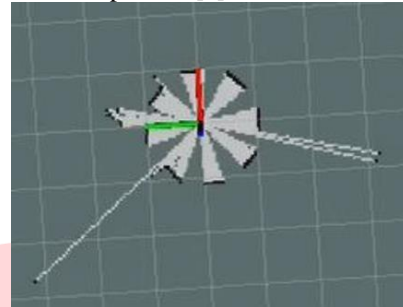
untuk melakukan navigasi kursi roda menggunakan hasil *layout map* dengan nilai pengukuran luas yang paling mendekati dengan hasil pengukuran luas ruangan manual atau nyata.

4.3. Pengujian dan Kalibrasi Sensor LiDAR

Pengujian ini bertujuan untuk mengetahui hasil pembacaan jarak 1 dan 2 meter pada jangkauan sudut tertentu. Pengujian ini dilakukan di Gedung Serba Guna Telkom University. Pengujian pengukuran jarak dilakukan sebanyak 10 kali pada masing – masing sudut. Beberapa sudut untuk pengujian jarak 1 m diantaranya $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 325^\circ, 360^\circ$. Tahapan dalam melakukan pengujian dengan mengukur jarak 1 dan 2 meter disetiap sudut [8].

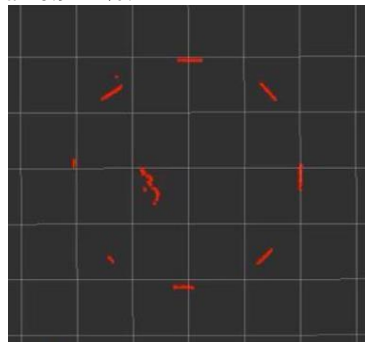


Gambar 4.5. Hasil Pembacaan Sensor Pada Jarak 1 m.

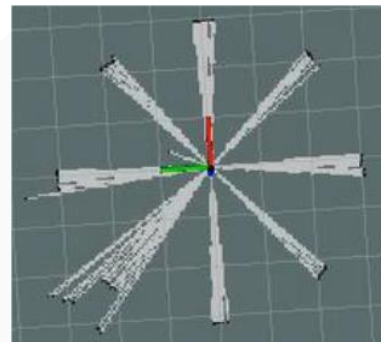


Gambar 4.6. Hasil Visualisasi Pemetaan Pada Jarak 1 m Melalui Rviz.

Pada pengujian sensor LiDAR terhadap jarak 1 meter pada sudut tertentu diantaranya $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 325^\circ, 360^\circ$, hasil yang didapatkan dari perhitungan kesalahan pengukuran sensor LiDAR dan pengukuran aktual dengan menggunakan alat ukur satuan panjang (meteran) didapatkan kesalahan pengukuran dari sepuluh kali pengujian yaitu total rata-rata kesalahan 0.941 %.



Gambar 4.7. Hasil Pembacaan Sensor Pada Jarak 2 m.

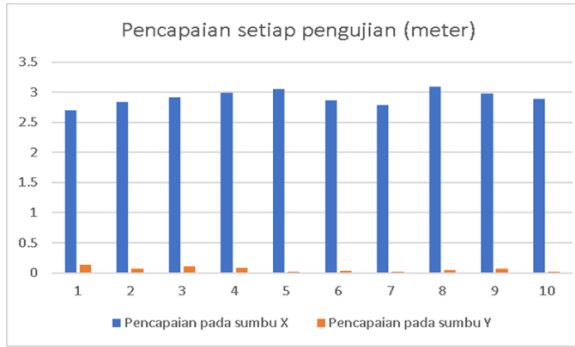


Gambar 4.8. Hasil Visualisasi Pemetaan Pada Jarak 2 m.

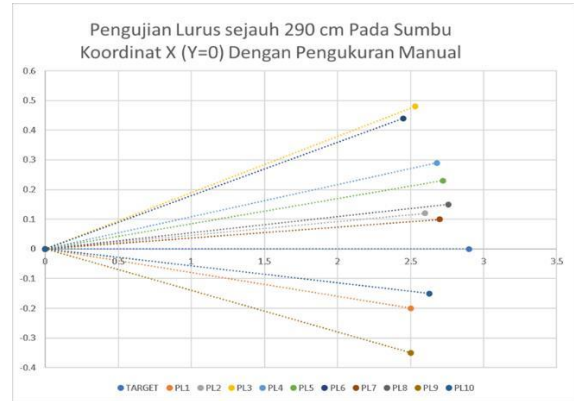
Pada pengujian sensor LiDAR terhadap jarak 2 meter pada sudut tertentu diantaranya $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 325^\circ, 360^\circ$, hasil yang didapatkan dari perhitungan kesalahan pengukuran sensor LiDAR dan pengukuran aktual dengan menggunakan alat ukur satuan panjang (meteran) didapatkan kesalahan pengukuran dari sepuluh kali pengujian yaitu total rata-rata kesalahan 1.483 %.

4.4. Pengujian Navigasi Lurus Kursi Roda

Pengujian ini dilakukan untuk mendapatkan besar kesalahan dari perpindahan kursi roda, berdasarkan dengan membandingkan nilai dari hasil keluaran koordinat (x,y) yang didapat dari program pada ROS dengan nilai hasil pengukuran dengan alat ukur manual ketika kursi roda melakukan navigasi. Pengujian navigasi kursi roda dilakukan dengan pola lurus dari titik (0,0) menuju (2.9 , 0). Pengujian ini dilakukan dan ditempuh berulang oleh kursi roda sebanyak sepuluh kali, sehingga didapat rata-rata kesalahan navigasi dari sepuluh kali pengujian yaitu 10.31 % pada sumbu x dan 76.77% pada sumbu y, juga persentase kesalahan terkecil yaitu 2.9% pada sumbu x, 15.57% pada sumbu y, persentase kesalahan terbesar yaitu 17.94% pada sumbu x dan 98.24% pada sumbu y dan pose perhentian akhir kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.76 , 0.15).



Gambar. 4.9. Grafik Pencapaian Titik Akhir Kursi Roda Dari Setiap Percobaan dengan titik awal (0,0)

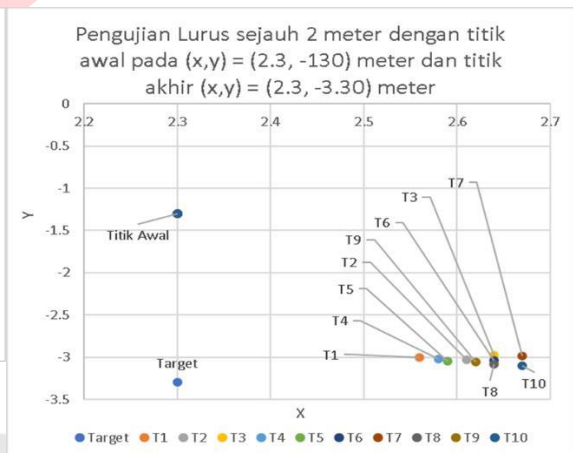


Gambar 4.10 Plot Grafik Pose Titik Awal (0,0) dan Titik Akhir Kursi Roda Setiap Pengujian.

Pengujian navigasi kursi roda dilakukan dengan pola lurus dari titik (2.3, -1.3) menuju ke titik (2.3, -3.3). Pengujian ini dilakukan dan ditempuh berulang oleh kursi roda sebanyak sepuluh kali, sehingga didapat persentase rata-rata kesalahan navigasi dari sepuluh kali pengujian yaitu 8,53% pada sumbu x dan 11.03% pada sumbu y, juga persentase kesalahan terkecil yaitu 0.58% pada sumbu x, 4.87% pada sumbu y, persentase kesalahan terbesar yaitu 14.18% pada sumbu x dan 25.12% pada sumbu y dan pose perhentian akhir kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.56, 3.1).



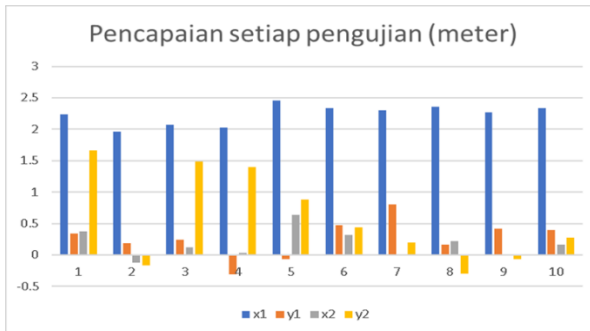
Gambar 4.11. Grafik Pencapaian Titik Akhir Kursi Roda Dari Setiap Percobaan, titik awal (2.3, -1.3) meter.



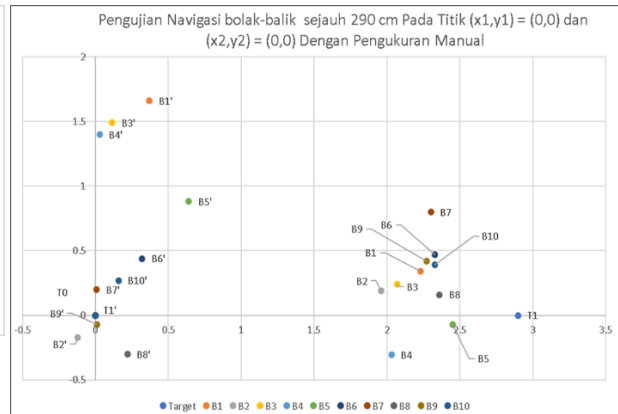
Gambar 4.12. Plot Grafik Pose Titik Awal dan Titik Akhir Kursi Roda Setiap Pengujian, titik awal (2.3, -1.3) meter.

4.5. Pengujian Navigasi Bolak-balik Pada Titik (0,0) menuju (2.9,0) Kembali Ke Titik (0,0)

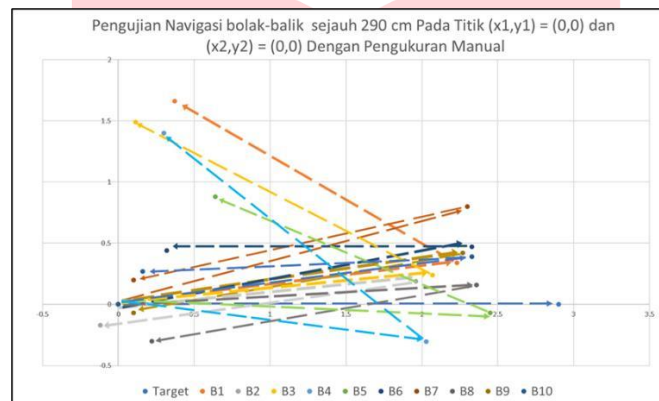
Pengujian ini dilakukan untuk mendapatkan besar kesalahan dari hasil nilai pencapaian kursi roda ketika melakukan navigasi bolak-balik dengan titik awal (0,0) menuju titik kedua (2.9,0) dan kembali ke titik awal dengan berdasarkan membandingkan nilai hasil keluaran koordinat (x,y) yang didapat dari program pada ROS dengan nilai hasil pengukuran dengan alat ukur manual ketika kursi roda melakukan navigasi. Pengujian ini dilakukan dan ditempuh berulang oleh kursi roda sebanyak sepuluh kali, sehingga didapat persentase rata-rata kesalahan navigasi dari sepuluh kali pengujian yaitu 31,13% pada sumbu x dan 49.38% pada sumbu y untuk perhentian awal, 77.67% pada sumbu x dan 85.62% pada sumbu y untuk perhentian akhir (Kembali ke titik awal), juga persentase kesalahan terkecil perhentian awal yaitu 16.06% pada sumbu x, 27.32% pada sumbu y, persentase kesalahan terbesar perhentian awal yaitu 50.71% pada sumbu x dan 73.31% pada sumbu y, persentase kesalahan terkecil perhentian akhir yaitu 48.17% pada sumbu x, 66.76% pada sumbu y, persentase kesalahan terbesar perhentian akhir yaitu 95.88% pada sumbu x dan 98.99% pada sumbu y dan pose perhentian awal kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.45, -0.07) dan (0.009, 0.02) untuk perhentian akhir.



Gambar 4.13. Grafik Pencapaian Titik Akhir Kursi Roda Dari Setiap Pengujian Navigasi bolak-balik.



Gambar 4.14. Plot Grafik Pose Titik Awal, Titik Perhentian awal dan Pose titik perhentian akhir (Kembali ke pose titik awal) Kursi Roda Setiap Pengujian.



Gambar 4.15. Plot Grafik Pose Titik Awal, Titik Perhentian awal dan Pose titik perhentian akhir (Kembali ke pose titik awal) Kursi Roda Setiap Pengujian (dengan panah).

5. Kesimpulan

Berdasarkan hasil penelitian, pengujian dan analisis sistem navigasi kursi roda otonom dengan sensor LiDAR, maka didapatkan kesimpulan sebagai berikut:

1. Penggunaan kendali kecepatan motor DC menggunakan sistem kendali PID dengan nilai konstanta $P = 0.16$, $I = 5.5$, dan $D = 0$ pada masing-masing mikrokontroler yang terhubung pada motor dc kanan dan kiri, nilai tersebut didapat dengan melakukan *trial and error* sampai didapat nilai RPM yang stabil dan dengan respon sistem yang cepat.
2. sistem pemetaan dengan sensor LiDAR untuk navigasi kursi roda, hasil perbandingan dari pengukuran nyata luas ruangan dengan pengukuran pada *layout map* program Rviz ROS didapatkan hasil perbandingannya tidaklah bersifat linier, dari pengukuran nyata dan *layout map* program Rviz ROS. Sehingga kesalahan pada navigasi kursi roda masih bernilai besar.
3. Besar persentase rata-rata kesalahan navigasi dengan pola lurus dari titik (0,0) menuju (2.9, 0) sebanyak sepuluh kali pengujian yaitu 10.31 % pada sumbu x dan 76.77% pada sumbu y, juga persentase kesalahan terkecil yaitu 2.9% pada sumbu x, 15.57% pada sumbu y, persentase kesalahan terbesar yaitu 17.94% pada sumbu x dan 98.24% pada sumbu y dan pose perhentian akhir kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.76, 0.15).
4. Besar persentase rata-rata kesalahan navigasi dilakukan dengan pola lurus dari titik (2.3, -1.3) menuju (2.3, -3.3) sebanyak sepuluh kali pengujian yaitu 8,53% pada sumbu x dan 11.03% pada sumbu y, juga persentase kesalahan terkecil yaitu 0.58% pada sumbu x, 4.87% pada sumbu y, persentase kesalahan terbesar yaitu 14.18% pada sumbu x dan 25.12% pada sumbu y dan pose perhentian akhir kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.56, 3.1).
5. Nilai kesalahan pada pengujian lurus menjadi evaluasi dan perbaikan untuk pengujian navigasi bolak-balik dengan balik dengan titik awal (0,0) menuju titik kedua (29,0) dan kembali ke titik

awal dilakukan pengujian sebanyak sepuluh kali sehingga didapatkan besar persentase rata-rata kesalahan yaitu 31,13% pada sumbu x dan 49,38% pada sumbu y untuk perhentian awal, 77,67% pada sumbu x dan 85,62% pada sumbu y untuk perhentian akhir (Kembali ke titik awal), juga persentase kesalahan terkecil perhentian awal yaitu 16,06% pada sumbu x, 27,32% pada sumbu y, persentase kesalahan terbesar perhentian awal yaitu 50,71% pada sumbu x dan 73,31% pada sumbu y, persentase kesalahan terkecil perhentian akhir yaitu 48,17% pada sumbu x, 66,76% pada sumbu y, persentase kesalahan terbesar perhentian akhir yaitu 95,88% pada sumbu x dan 98,99% pada sumbu y dan pose perhentian awal kursi roda dengan nilai koordinat yang paling mendekati ada pada titik (2.45 , -0.07) dan (0.009 , 0.02) untuk perhentian akhir.

REFERENSI

- [1] Bellis Mary, "History of the wheelchair," ThoughtCo, Feb. 11, 2020. thoughtco.com/history-of-the-wheelchair-1992670. (accessed Apr. 21, 2020).
- [2] C. Myint and N. N. Win, "Position and Velocity control for Two-Wheel Differential Drive Mobile Robot," *Int. J. Sci. Eng. Technol. Res.*, vol. 5, no. 9, pp. 2278–7798, 2016.
- [3] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an Autonomous Mobile Robot Based on ROS," *IDAP 2017 - Int. Artif. Intell. Data Process. Symp.*, 2017, doi: 10.1109/IDAP.2017.8090199.
- [4] A. Jalil, "Robot Operating System (Ros) Dan Gazebo Sebagai Media Pembelajaran Robot Interaktif," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 284–289, 2018, doi: 10.33096/ilkom.v10i3.365.284-289.
- [5] K. Zheng, "ROS Navigation Tuning Guide," vol. 2019, pp. 1–23, 2017, [Online]. Available: <http://arxiv.org/abs/1706.09068>.
- [6] J. S. Gill, "Setup and Configuration of the Navigation Stack on a Robot," 2018. <http://wiki.ros.org/navigation/Tutorials/RobotSetup>.
- [7] D. Erlangga, E. D. R. H S, S. Sunarto, K. Rahardjo T.S, and F. G, "Sistem Navigasi Mobile Robot Dalam Ruang Berbasis Autonomous Navigation," *J. Mech. Eng. Mechatronics*, vol. 4, no. 2, p. 78, 2019, doi: 10.33021/jmem.v4i2.823.
- [8] R. SEPTISARI, "SISTEM PEMETAAN PADA KURSI RODA AUTONOMOUS MENGGUNAKAN SENSOR LiDAR A1M1," *Univ. Telkom, S1 Tek. Elektro*, p. 83, 2021.
- [9] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *Int. J. Rob. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007, doi: 10.1177/0278364907081229.
- [10] T. Bräunl, "Localization and Navigation," *Embed. Robot.*, pp. 241–269, 2008, doi: 10.1007/978-3-540-70534-5_16.
- [11] [www.mathworks.com](https://www.mathworks.com/discovery/slam.html), "SLAM (Simultaneous Localization and Mapping)." <https://www.mathworks.com/discovery/slam.html>.
- [12] A. Ma'Arif, Iswanto, N. M. Raharja, P. A. Rosyady, A. R. C. Baswara, and A. A. Nuryono, "Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation," *Proceeding - 2020 2nd Int. Conf. Ind. Electr. Electron. ICIEE 2020*, pp. 74–78, 2020, doi: 10.1109/ICIEE49813.2020.9277258.
- [13] R. Haq and E. Rahmawati, "Kendali Posisi Mobile Robot Menggunakan Sistem Proportional Integral Derivative (Pid) Dengan Metode Odometry," *Inov. Fis. Indones.*, vol. 6, no. 3, pp. 63–67, 2017.
- [14] M. A. Hidayatullah, "MONITORING AREA PARKIR JURUSAN TEKNIK KOMPUTER MENGGUNAKAN CCTV BERBASIS RASPBERRY PI," *Politek. NEGERI SRIWIJAYA, Jur. Tek. Komput. Politek.*, vol. 8, no. 5, p. 55, 2019.
- [15] Arduino, "Arduino Uno Rev3," arduino.cc, 2020. .
- [16] RoboPeak Team, "RPLIDAR Low Cost 360 Degree Laser Scanner (LIDAR) System Development Kit User Manual," 2013. <http://www.RoboPeak.com> (accessed Apr. 29, 2020).
- [17] D. P. Wicaksa, "SISTEM PENGENALAN POSISI PADA LINGKUNGAN TERTUTUP MENGGUNAKAN METODE FEATURE DETECTION BERDASARKAN SENSOR RP-LIDAR 3600 LASER SCANNER A1M1," *Univ. Telkom, S1 Tek. Elektro*, 2017.
- [18] Dynapar, "Quadrature Encoder Overview," 2021. https://www.dynapar.com/technology/encoder_basics/quadrature_encoder/.