

PEMETAAN LINGKUNGAN KERJA ROBOT BERODA DENGAN METODE SLAM GAPPING MENGGUNAKAN SENSOR LIDAR

Noeores Shobie Ahfan¹, Peby Wahyu Purnawan², Sujono³ Akhmad Musafa⁴, dan Indra Riyanto⁵
Program Studi Teknik Elektro, Fakultas Teknik Universitas Budi Luhur Jakarta, Indonesia
¹shobieahfan@gmail.com, ²pebywahyupurnawan, ³sujono, ⁴akhmad.musafa, ⁵indra.riyanto@budiluhur.ac.id

ABSTRAK

Robot beroda yang digunakan untuk mengantarkan dokumen antar ruangan harus mampu bergerak sesuai dengan kondisi lingkungan daerah kerjanya. Untuk itu robot harus memiliki pengetahuan tentang kondisi ruangan lingkungan kerja yang akan dilalui. Dalam penelitian ini dilakukan pemetaan lingkungan daerah kerja pada robot beroda. Pemetaan dilakukan dengan menggunakan metode Simultaneous Localization and Mapping (SLAM). Peralatan yang digunakan dalam pemetaan adalah sensor lidar. Sistem robot terdiri raspberry Pi 4 yang digunakan sebagai pengendali utama robot. Pada robot terdapat dua sensor. Sensor yang pertama sensor lidar, digunakan sebagai pendeteksi jarak objek yang ada di depan robot. Kemudian sensor IMU digunakan sebagai pendeteksi orientasi dan perubahan posisi robot. Pada robot terdapat driver motor yang digunakan sebagai pengolah sinyal kendali robot untuk menggerakkan motor DC. Pembuatan peta dilakukan dengan cara sensor lidar membaca lingkungan daerah kerja robot. Sinyal keluaran sensor lidar diproses menggunakan metode SLAM gmapping. Pada pengujian ini untuk mengetahui lingkungan daerah kerja robot dilakukan dengan menggunakan laser scanner untuk menghasilkan peta dua dimensi (2D), sementara untuk mengestimasi posisi robot pada peta menggunakan particle filter. Pemetaan secara bersamaan ini menggunakan algoritma Simultaneous Localization and Mapping (SLAM) Gmapping berbasis raspberry Pi 4. Hasil yang diperoleh berupa peta dalam skala abu-abu. Selain SLAM gmapping, artikel ini juga memperlihatkan arena pengujian 2D posisi robot ada satu sampai tiga.

Kata kunci : IMU ROS, SLAM, RPLIDAR, MAPPING, RASPBERRY PI 4

ABSTRACT

The wheeled robot used to deliver documents between rooms must be able to move according to the environmental conditions of the work area. For this reason, the robot must have knowledge of the conditions of the work environment to be passed. In this final project, the work area environment mapping on the wheeled robot is carried out. Mapping was done using the Simultaneous Localization and Mapping (SLAM) method. The equipment used in the mapping is a lidar sensor. The robot system consists of a raspberry Pi 4 which is used as the main controller of the robot. The robot has two sensors. The first sensor is a lidar sensor, which is used to detect the distance of the object in front of the robot. Then the IMU sensor is used to detect the robot's orientation and position changes. In the robot there is a motor driver which is used as a robot control signal processor to drive a DC motor. Map making is done by means of a lidar sensor reading the robot's working area environment. The lidar sensor output signal is processed using the SLAM gmapping method. In this test, to determine the environment of the robot's work area, using a laser scanner to produce a two-dimensional (2D) map, while estimating the position of the robot on the map using a particle filter. This simultaneous mapping uses the Simultaneous Localization and Mapping (SLAM) mapping algorithm based on Raspberry Pi 4. The results obtained are maps in grayscale. In addition to SLAM gmapping, this article also shows that there are one to three robot position 2D testing arenas.

Keywords: IMU ROS, SLAM, RPLIDAR, MAPPING, RASPBERRY PI 4

I. PENDAHULUAN

Rancangan pemetaan lingkungan kerja pada robot beroda menggunakan Rplidar. Robot harus dapat membentuk peta tempat beroperasi dan menganalisa rintangan dan jalan yang dapat dilalui menuju lokasi yang ditentukan dalam Lorong Universitas Budi Luhur unit 6 lantai 2. Dalam bentuk peta menggunakan slam gmapping yang merupakan 3D yang digunakan untuk mengenali lingkungan. Hasil pembacaan tersebut diproses oleh laptop dikonversi menjadi pembacaan laser scan untuk membentuk peta. Dengan algoritma simultaneous localization and mapping (SLAM) gmapping untuk mengkonversi gambar menjadi grip map yang menghasilkan data peta[1]. Pada penelitian ini mencoba mengambil data objek dilorong Universitas Budi Luhur unit 6 lantai 2 menggunakan sensor rpLidar 360 derajat yang diterapkan pada mobile robot untuk memperkirakan ukuran dilorong Universitas Budi Luhur unit 6 lantai 2. Namun, untuk mengukur sebuah dilorong Universitas Budi Luhur unit 6 lantai 2 secara penuh yang tidak terjangkau oleh sensor diperlukan pembacaan odometri. *Light Detection and Ranging* (LIDAR) merupakan sensor yang berkerja dengan memancarkan dan menangkap cahaya untuk memperoleh data berupa jarak dan informasi suatu objek[2]. Simultaneous Localization and Mapping (SLAM) salah satu bidang yang paling banyak diteliti dan kurang diteliti, robot yang ditempatkan dilokasi yang tidak diketahui, membangun dan mengeluarkan peta lingkungan, dan memetakan. Robot untuk pemetaan lingkungannya dan juga untuk pengukuran data. Jadi pada SLAM, robot menyelesaikan dua tugas dengan satu algoritma dengan pemetaan dan menentukan posisinya pada satu area yang ditemukan secara bersamaan[3]. Berdasarkan ini menajikan Teknik baru untuk robot yang berada tidak ketahui area tertentu. Kerangka kerja yang di sensor Rplidar untuk memindai lingkungan dan mengumpulkan informasi tentang robot[4]. Sehingga robot menggunakan sensor RP lidar untuk mendapatkan hasil area lingkungan dan mengirimkan informasi data ke raspberry. Aturan sin dan cos diperlukan untuk cari titik koordinat titik awal garis, sudut garis [5]. *Robot Operating system* (ROS) merupakan sebuah perangkat lunak pengoperasian robot. *Robot Operation system* bertujuan membuat perangkat lunak pengoprasian robot yang kuat dan bisa digunakan untuk segala jenis robot [6]. Integrasi Gazebo simulator dengan robot Operating system (ROS) system dibentuk dan dijalankan dalam perangkat lunak yaitu Gazebo Simulator yang diintegrasikan dengan Robot Operating system (ROS) [7]. Gazebo simulator merupakan sebuah perangkat lunak yang dikhususkan untuk melakukan simulasi robot. Gazebo simulator biasanya diintegrasikan dengan perangkat lunak lain yang berfungsi sebagai pengendali bagi robot [8]. Berdasarkan hasil

penelitian yang sudah diuraikan, penelitian ini dirancang pengambilan data pemetaan pada robot menggunakan slam gmapping, pengambilan data pemetaan robot hanya diam dan ada tiga posisi robot mobile yang dilakukan untuk mengambil pemetaan.

II. LANDASAN TEORI

Platform robot yang digunakan dalam artikel ini adalah robot mobile. Robot ini memiliki 4 roda dan penggerakan serta sebuah penyangga. Untuk melakukan manuver, robot ini memvariasikan kecepatan putar motornya. Mekanisme penggerakan dengan sistem ini dikenal dengan *differential drive* [9]. Modul software menggunakan *Robot Operating Sistem* (ROS) sebagai robot *middleware*. ROS merupakan *middleware* dalam penelitian robotika. ROS menyediakan modul-modul yang dapat diimplementasikan langsung oleh peneliti robotika ataupun komunitas dan memungkinkan pengembangan modul-modul baru. Untuk menyelesaikan permasalahan SLAM, pada artikel ini digunakan teknik *partikel filter* dan diimplementasikan dengan ROS dalam modul SLAM *gmapping* [10]. Robot dilengkapi dengan laser *scanner* sebagai sensor untuk pembuatan peta denah arena indoor yang dijelaskan lebih rinci pada bagian berikutnya. ROS diinstal pada *Single Board Computer* (SBC) odroid XU-4 sebagai pengganti Raspberry Pi 4 yang merupakan SBC standar yang terpasang pada robot mobile.

• ROS

Robot Operating Sistem (ROS) merupakan suatu *framework* berbasis open source yang berisi *tools* dan *library* yang berguna dalam mengembangkan program atau aplikasi untuk sistem robot[11]. Struktur ROS bersifat modul ar sehingga memudahkan proses integrasi antara program yang telah dikembangkan sebelumnya dengan program yang sedang dikembangkan. Komunikasi antar proses dalam ROS dilakukan dengan menghubungkan program atau yang disebut dengan *node* yang saling berkomunikasi via *topic*. *Node* merupakan program yang ditulis untuk melakukan berbagai fungsi atau tugas tertentu. Sedangkan *topic* adalah *bus* yang digunakan dalam transmisi pesan (*message*) antar *node*. *Node* dapat mem-*publish* data (misanya data laser *scanner*) yang disebut dengan *publisher* dan *node* dapat pula berfungsi menerima dan menampilkan data yang disebut dengan *subscriber*. Streaming data (*message*) via *topic* tersebut dilakukan dalam jaringan lokal TCP yang disebut dengan jaringan ROS atau ROS *network*. Kumpulan dari *nodes* yang saling berkomunikasi diorganisir menjadi paket atau *package*. *Package* terdiri dari *node* dan *library* lainnya. Kumpulan dari *nodes* yang saling berkomunikasi diorganisir menjadi paket atau *package*. *Package* terdiri dari *node* dan *library* lainnya.

• Laser Scanner

Laser *scanner* atau sering juga disebut dengan LiDAR (*Light Detection Ranging*) merupakan komponen penting dalam robot mobile. Lidar

menggunakan laser untuk memindai halangan atau dinding agar robot dapat bermanuver dengan aman. Penggunaan lidar sangat populer dewasa ini untuk berbagai keperluan terutama semakin populer setelah semakin banyak digunakan mobil mengemudi otomatis (*self driving cars*) [12]. Cara kerja sensor ini adalah dengan memproyeksikan sinar laser atau sinyal sumber kemudian menghitung lamanya (waktu) sinyal pantul yang diterima oleh receiver lidar [13]. Pada lidar 360 derajat, proses ini berlangsung secara berkesinambungan. Hasil scan atau pemindaian lidar ini berupa titik-titik awan (*point clouds*) dalam bidang 2 dimensi, karena sensor ini hanya menindai bidang horizontal maka *point clouds* yang dihasilkan berupa ruang 3 dimensi.

- **Algoritma SLAM**

SLAM merupakan algoritma komputasi untuk membuat peta lingkungan sekitar robot dan memperbarui peta berdasarkan informasi dari sensor serta disaat yang sama merekam jejak lintasan robot. Ada berbagai macam teknik yang bisa digunakan untuk menerapkan algoritma SLAM antara lain *hector_slam*, *gmapping*, dan *cartographer* [14]. Semua algoritma SLAM ini tersedia dalam paket standar ROS. Dalam artikel ini, penulis menggunakan *SLAM gmapping* untuk melakukan proses lokalisasi dan pemetaan. Metode SLAM dengan *gmapping* mirip dengan metode SLAM lainnya seperti *hector_slam* dimana metode ini dapat membuat peta meskipun tidak ada informasi odometry pada robot dan mengimplementasikan algoritma lingkaran terbuka (*open loop algoritma*) dan sebagai konsekuensinya eror akibat ketidakpastian sensor akan bertambah besar selama proses pemetaan dilakukan. *Gmapping* disini lain menggunakan pendekatan lingkaran tertutup (*closed loop approach*) dimana jika robot kembali pada posisi awal lagi, peta yang dibentuk akan diperbarui [10]. *Gmapping* relative lebih ringan dan membutuhkan sumber daya komputasi lebih sedikit dibandingkan dengan *cartographer*. Sehingga metode ini cocok diterapkan pada robot mobile. Metode *gmapping* membutuhkan informasi dari lidar dan data odometry robot. Lokalisasi pada *gmapping* menggunakan metode *Monte Carlo Localization (MCL)* atau sebut juga dengan filter partikel (*particle filter*). Pemetaan dengan *gmapping* menggunakan teknik *occupancy grid map*. Peta ini merupakan peta dua dimensi (2D) yang sering digunakan pada ROS. *Occupancy grid map* merupakan peta atau citra *greyscale* yang direpresentasikan dalam rentang nilai 0 sampai dengan 225. Nilai ini diperoleh dari probabilitas posterior menggunakan teorema Bayes. Peta hasil algoritma SLAM dengan *gmapping*. Warna putih pada peta adalah area dimana robot bisa bergerak bebas. Warna hitam merepresentasikan dinding atau area okupansi dimana robot tidak bergerak. Dan terakhir warna abu-abu merupakan daerah yang dieksplorasi oleh robot. Peta ini nantinya digunakan sebagai informasi bagi robot untuk bermanuver dan bernavigasi. Pembuatan peta 2D *occupancy grid*

map didasarkan informasi dari odometry robot, modul transformasi koordinat (TF), dan informasi dari sensor lidar (*laser scan*) Ketika robot bergerak. Data ini dapat divisualisasikan pada ROS dengan RViz. Peta 2D yang dihasilkan disimpan oleh *map_server* dalam *map.pgm* dan *map.yaml*. Peta ini digunakan oleh robot untuk mendeteksi rintangan atau objek di sekitar robot sehingga robot dapat bermanuver dengan aman. Alur proses algoritma *gmapping SLAM* dan *node-node ROS* yang diperlukan diilustrasikan.

- Lingkungan pengujian pada arena buatan
Lingkungan pengujian dalam pembuatan peta *indoor* ruangan oleh robot dilakukan pada arena. lokasi arena peta yang akan dibuat adalah lingkungan dilorong Universitas Budi Luhur Unit 6 Lantai 2.

III. PERANCANGAN SISTEM

- Diagram Blok Sistem

Gambar 3.1 menjelaskan diagram blok dari system robot yang digunakan. Dari gambar tersebut, tampak ada beberapa komponen utama yang digunakan. Raspberry pi digunakan sebagai komponen pengolah data, perintah, dan pengendali utama robot [15]. Sebagai sensor, robot menggunakan dua jenis sensor. Sensor yang pertama adalah sensor Rplidar. Sensor ini digunakan sebagai pendeteksi jarak objek yang ada di depan robot [13]. Kemudian sensor IMU digunakan sebagai pendeteksi orientasi dan perubahan posisi robot [16]. Sedangkan driver motor digunakan sebagai pengolah sinyal kendali robot untuk menggerakkan motor DC [17]. Sehingga posisi robot akan dapat berubah (bergerak).



Gambar 3. 1 diagram blok sistem

Diagram yang ditunjukkan oleh gambar 3.1 merupakan diagram blok system yang berjalan secara otomatis. Dimana robot berjalan tanpa kendali manusia dengan bantuan apapun termasuk kendali remot [18]. Tetapi, keadaan ini baru bisa terpenuhi jika robot sudah memiliki peta dari area yang dilalui oleh robot tersebut. Maka untuk mencapai kondisi sepenuhnya system otomatis, diperlukan satu langkah penting yaitu membuat peta lingkungan robot. Menjelaskan bahwa robot beroda untuk pemetaan perlu mencari IP Robot dan IP Laptop. Pada pemetaan robot mobile perlu ada IP Robot dan IP Laptop untuk bisa melakukan pemetaan yang ada di robot beroda yang digunakan pada sensor Rplidar.

- Rangkaian Sensor Rplidar

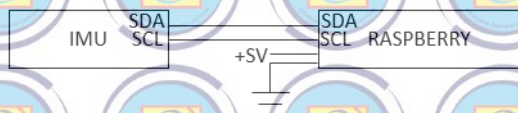


Gambar 3.1.1 Rangkaian Sensor Rplidar

Gambar 3.1.2 bahwa sensor Rplidar untuk menyambungkan bisa berkoneksi USB diraspberry.

- Rangkaian Sensor IMU

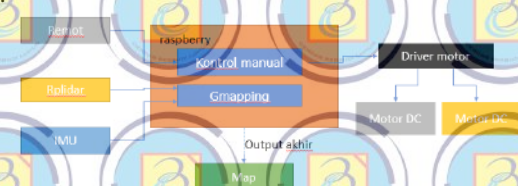
Gambar 3.1.3 bahwa sensor IMU untuk menyambungkan diraspberry. Kemudian raspberry dan sensor IMU berhubungan dengan secara menggunakan I2C, kemudian I2C berkomunikasi serial dua arah menggunakan dua saluran yang khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya.



Gambar 3.1.2 Rangkaian Sensor IMU

- Sistem Pembuatan Peta

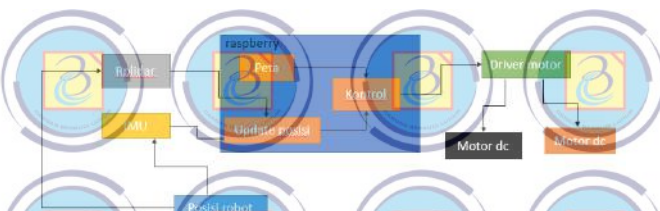
Gambar 3.2 menampilkan diagram blok ketika proses pembuatan peta. Proses ini ditujukan agar system memiliki peta dari area lingkungan operasi robot. Sehingga ketika beroperasi sepenuhnya, maka system dijkstra akan dapat dijalankan. Proses ini menggunakan dua sensor utama. Sensor Rplidar digunakan memancarkan sinar laser ke target di permukaan bumi, kemudian sinar laser tersebut dipantulkan kembali ke sensor[13]. Kemudian, sensor IMU berperan untuk merekam posisi robot saat terakhir[16]. Disaat yang bersamaan dengan robot yang dikendalikan secara manual, robot menjalankan program untuk merekam peta dengan metode *SLAM gmapping*[3]. Sehingga dapat menghasilkan peta yang digunakan oleh robot ini. Dimana lokasi peta yang akan dibuat adalah lingkungan di Universitas Budi Luhur Unit 6 Lantai 2.



Gambar 3. 2 Sistem Pembuatan Peta

- Sistem Navigasi

Gambar 3.3 menampilkan system navigasi otomatis robot. Dari gambar tersebut, ada beberapa komponen utama yang digunakan. Peta digunakan untuk berjalannya robot yang akan dilalui. Sedangkan control akan pengolah sinyal kendali robot untuk menggerakkan driver motor. Sedangkan robot, menggunakan dua jenis sensor. Pertama ada sensor Rplidar. Sensor ini digunakan pendeteksi jarak untuk mendapatkan objek yang ada di depan robot[13]. Kemudian sensor IMU digunakan sebagai perubahan posisi robot akan di update posisi[16].

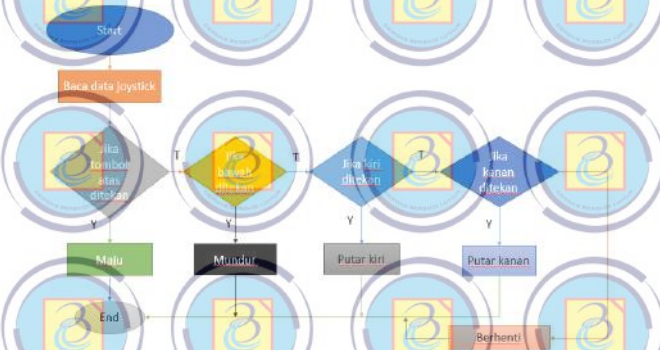


Gambar 3. 3 Sistem Navigasi

System navigasi yang ditunjukkan gambar 3.3 merupakan system robot yang sudah dapat berjalan otomatis. Pada keadaan ini, robot sudah memiliki peta dari area yang dilalui oleh robot tersebut. Sehingga algoritma dijkstra dapat menunjukkan perannya dalam mencari rute terpendek untuk berpindah dari satu titik lokasi ke titik lokasi lainnya[19]. Sehingga system navigasi ini dapat menunjukkan siklus yang tertutup dari masuknya input lokasi tujuan dan lokasi awal sampai berpindahanya lokasi robot ke titik yang ditentukan atau lokasi tujuan.

- Diagram alir driver motor

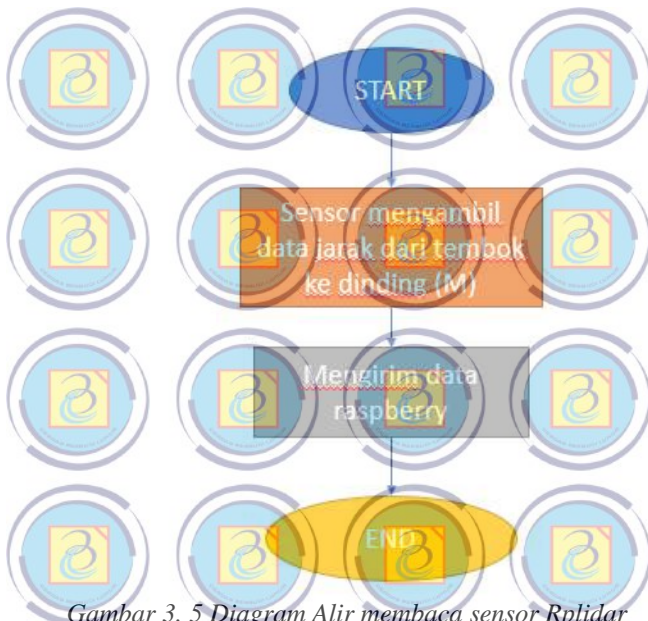
Gambar 3.4 menjelaskan diagram alir Driver motor yang digunakan. Pertama akan menyalakan robot mobile. Kemudian akan membaca data joystick. Jika tombol ditekan atas kondisi robot maju. Jika tombol ditekan kanan kondisi robot kanan. Jika tombol ditekan bawah kondisi robot mundur. Jika tombol ditekan kiri kondisi robot kiri.



Gambar 3. 4 Diagram Alir Driver motor

- Diagram alir membaca sensor Rplidar

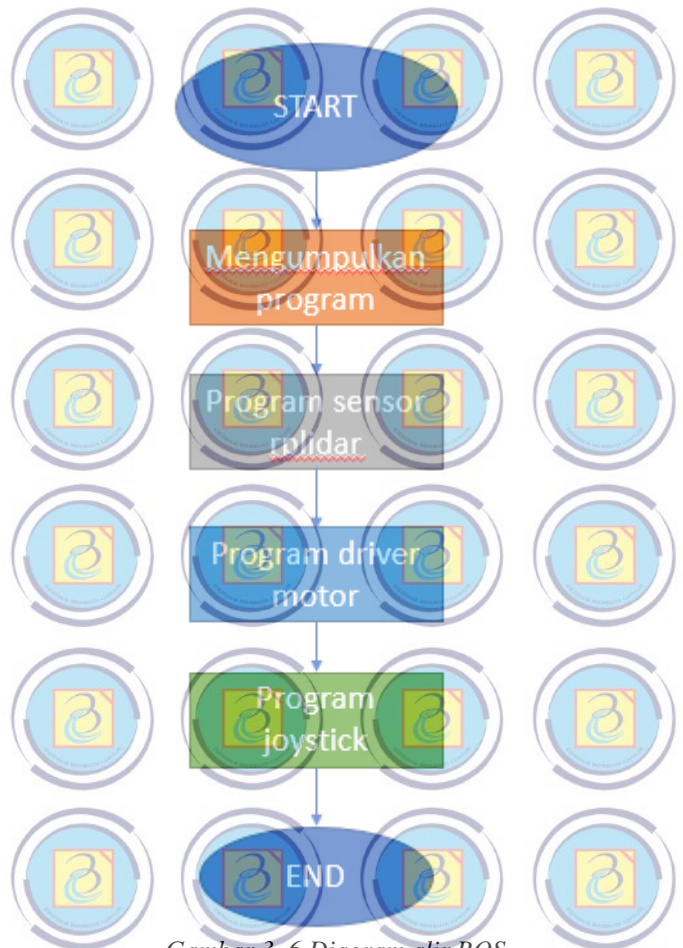
Gambar 3.5 menjelaskan diagram alir membaca sensor rplidar yang digunakan. Pertama menyalakan robot mobile. Kemudian sensor mengambil data jarak dari tembok ke dinding (M). Selanjutnya akan mengirim data ke raspberry. Diagram alir membaca sensor IMU.



Gambar 3. 5 Diagram Alir membaca sensor Rplidar

- Diagram alir ROS

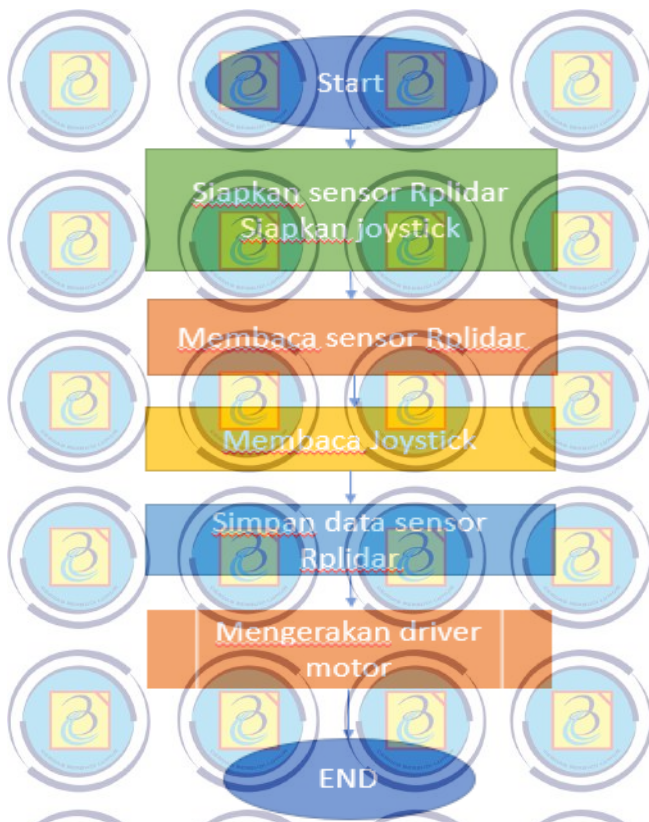
Dari Gambar 3.6 dapat dijelaskan diagram alir *Robot Operating System* (ROS) yaitu perangkat lunak yang digunakan sebagai pengumpulan program modul dan sensor. Program sensor rplidar, program *driver* motor L298N, dan program joystick yang telah dibuat akan disimpan pada ROS. Tugas dari ROS yaitu mengumpulkan data serta program yang telah disimpan untuk gabungan lalu menjalankan program secara bersamaan.



Gambar 3. 6 Diagram alir ROS

- Diagram alir pembuatan peta

Gambar 3.7 menjelaskan diagram alir pembuatan peta yang digunakan. Pertama akan menyalakan robot mobile. Kemudian menyiapkan sensor Rplidar, menyiapkan Joystick. Selanjutnya membaca sensor Rplidar. Selanjutnya membaca Joystick. Kemudian akan simpan data sensor Rplidar. Kemudian driver motor akan bergerak.



Gambar 3. 7 Diagram alir pembuatan peta

- Desain Robot Mobile

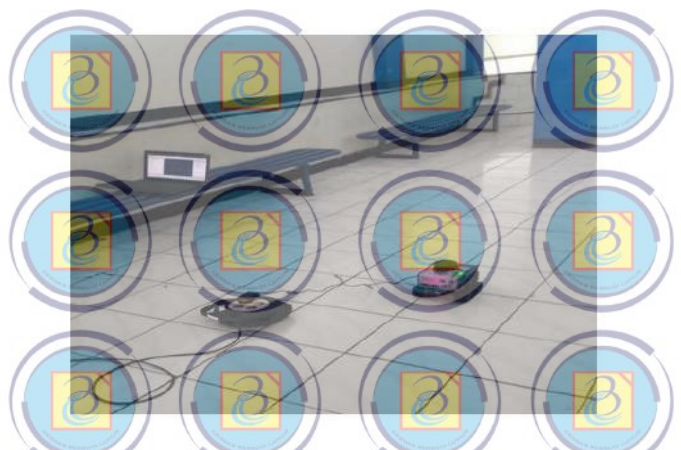
Robot mobile berikut ini adalah robot bergerak menggunakan empat pasang roda. Keempat roda ini digerakkan oleh dua buah motor DC, dimana satu buah menggerakkan 2 buah roda. Tata letak serta bentuk dari keempat pasang roda ini dibuat sedemikian rupa sehingga hanya dengan dua motor dapat menggerakkan roda tersebut.



Gambar 3. 8 Desain Robot Mobile

IV. PENGUJIAN PERILAKU ROBOT

- Pengujian sensor lidar
Pengujian sensor lidar ini untuk membuat peta area lingkungan di Universitas Budi Luhur unit 6 lantai 2.



Gambar 4. 1 area pengujian posisi satu



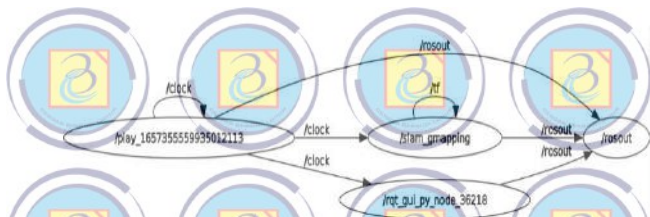
Gambar 4. 2 area pengujian posisi dua



Gambar 4. 3 area pengujian posisi tiga

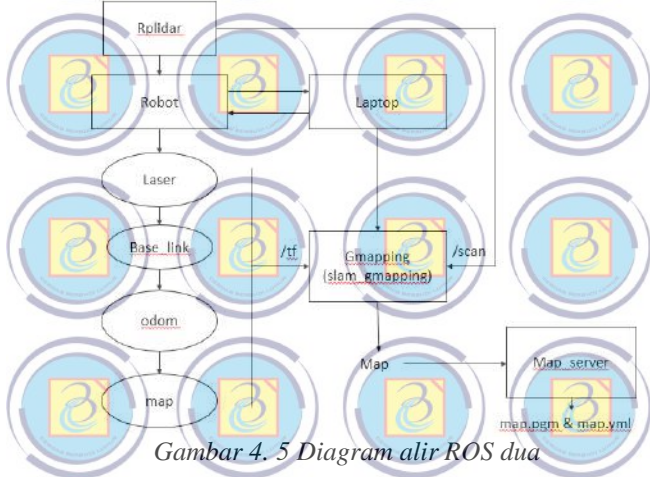
Terdapat lidar Al AIM8 360, robot mobile, baterai lipo, antena wifi. Didalam kotak warna pintu adalah raspberry Ip 4 dimana sistem ROS terinstal. ROS yang digunakan adalah versi *kinetic kame* dan Linux versi 16.04 LTS. Selain itu untuk menjalankan simulator gazebo diperlukan PC desktop karena membutuhkan kartu grafis dengan ukuran 4 GB agar simulasi dengan gazebo berjalan mulus.

- Diagram ROS pada robot dan PC
Penelitian ini membahas tentang bagaimana membuat peta disekitar robot menggunakan sensor lidar. Peta 2D menghasilkan area pengujian sebenarnya, peta dihasilkan di publish oleh ROS sebagai informasi bagi *topic node*.



Gambar 4. 4 Diagram ROS satu

Hal ini diperlukan karena kita dapat mendistribusikan implementasi tugas dalam ROS node pada kedelapan core dari CPU, terutama untuk menjalankan algoritma SLAM pada robot mobile ini yang membutuhkan komputasi intensif dan diharapkan algoritma SLAM dapat berjalan mulus.

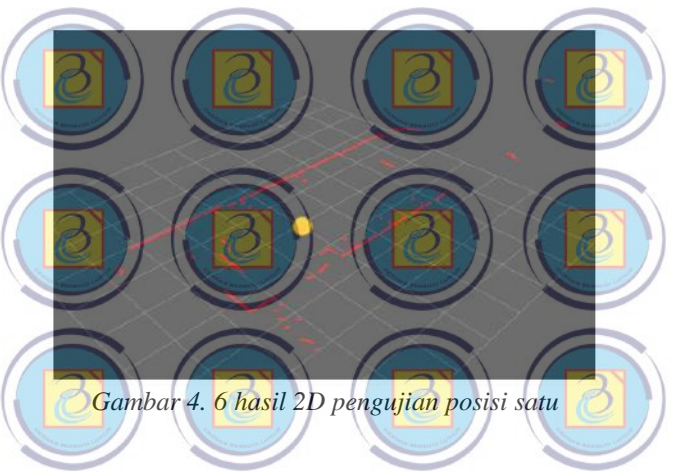


Gambar 4. 5 Diagram alir ROS dua

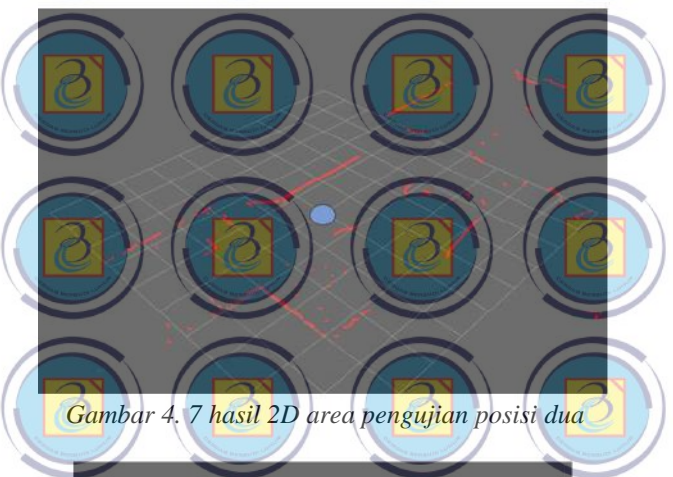
Node dan topic yang dieksekusi pada robot mobile seperti gambar 4.4. Package pada robot mobile untuk menjalankan algoritma SLAM gmapping antara lain adalah Rplidar, /tf, teleoperation, dan gmapping. Node gmapping memperoleh informasi lingkungan dari sensor lidar kemudian diolah menjadi peta probabilistik 2D *occupancy grid map*. Peta 2D disimpan oleh package map_server. Untuk melihat hasilnya pada PC desktop, perlu dijalankan RViz. Untuk melihat hasilnya pada PC desktop, perlu dijalankan RViz. Sementara untuk keperluan simulasi dengan gazebo, langkah yang harus dijalankan adalah dengan menjalankan gazebo, teleoperation, gmapping dan RViz.

- Hasil pemetaan LIDAR rviz

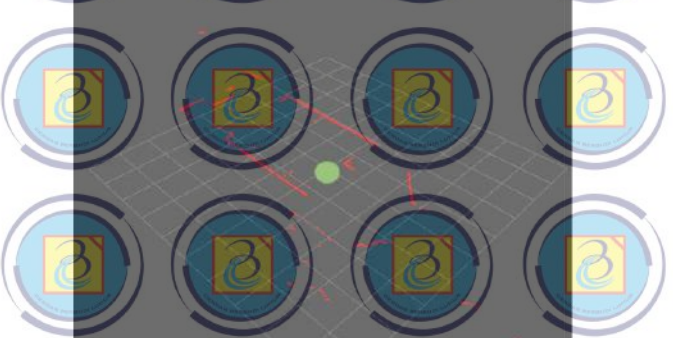
Hasil pemetaan ini membaca oleh sensor lidar yang berada area lingkungan Universitas Budi Luhur unit 6 lantai 2.



Gambar 4. 6 hasil 2D pengujian posisi satu



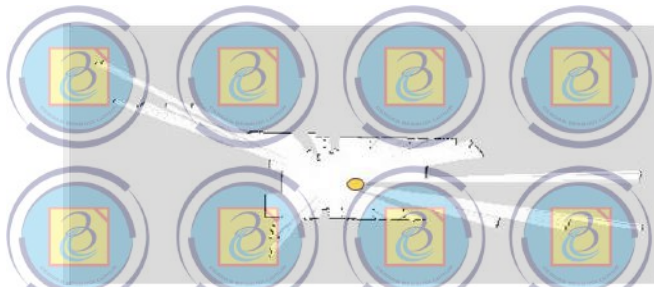
Gambar 4. 7 hasil 2D area pengujian posisi dua



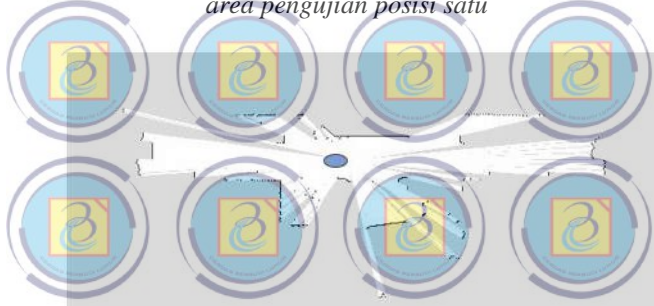
Gambar 4. 8 hasil 2D area pengujian posisi tiga

Menjelaskan area pengujian dengan rviz. Dari gambar tersebut ada beberapa warna yaitu, berdasarkan berwarna garis merah adalah penghalang yang ada di depan robot tersebut. Titik warna kuning, biru, hijau itu titik lokasi robot berada.

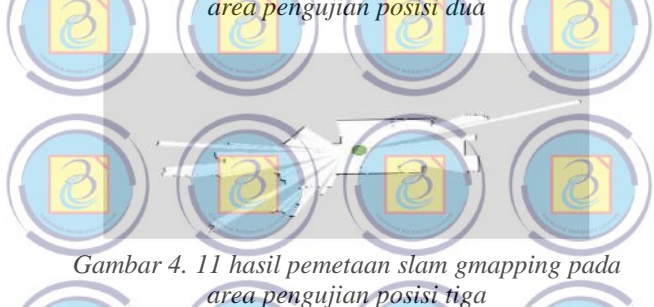
- Hasil pemetaan SLAM Gmapping
Hasil peta SLAM gmapping berupa 2D dalam skala hitam, putih, abu-abu, yang titik kuning, biru, hijau menandakan titik lokasi robot berada.



Gambar 4. 9 hasil pemetaan slam gmapping pada area pengujian posisi satu



Gambar 4. 10 hasil pemetaan slam gmapping pada area pengujian posisi dua



Gambar 4. 11 hasil pemetaan slam gmapping pada area pengujian posisi tiga

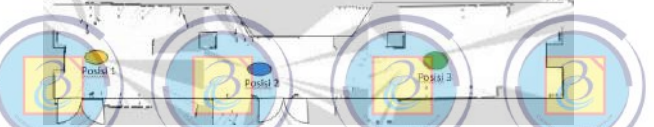
Dari hasil pembahasan diatas dapat diperoleh beberapa hal yang dapat didiskusikan. Melihat hasil peta 2D dengan occupancy grip map yang dihasilkan dari algoritma SLAM gmapping terlihat bahwa hasilnya dapat merepresentasikan area pengujian. Hal ini terlihat dari segmentasi warna dalam skala abu-abu grey scale bahwa warna putih merupakan area terbuka, warna hitam merupakan dinding-dinding arena, dan warna abu-abu merupakan area yang tidak dieksplorasi oleh robot mobile, kemudian warna kuning, biru, hijau itu titik lokasi robot mobile. Peta 2D yang dihasilkan cukup akurat simulasi dengan gazebo menunjukkan bahwa simulator ini cukup baik untuk melakukan simulasi.

Menjelaskan bahwa untuk mendapatkan data seperti gambar 4.9, 4.10, 4.11. Kemudian menjalankan (gedit.bashrc) untuk IP ROS dilaptop untuk mendapatkan hasil gambar 4.9, 4.10, 4.11 berkomentar pada di export ROS, selanjutnya adalah merekam perubahan data-data ROS menjalankan (rosbag record -a-o posisi1.a.bag -duration=90) untuk mendapatkan data hasil pemetaan merubah nama file (posisi1.a.bag), setelah itu menjalankan (roscore), selanjutkannya menjalankan (roscparam set use_sim_time true), menjalankan (roscrun gmapping slam_gmapping scan=scan), menjalankan (rosbag play -clock posisi1.a.bag), setelah itu menjalankan (roscrun map_server map_saver -f posisi1.a.bag), kemudian sudah menjalankan perintahnya untuk

mendapatkan hasil data seperti gambar 4.9, 4.10, 4.11.

- Hasil peta atau layout

Gambar 4.12 menjelaskan bahwa area pengujian lorong Universitas Budi Luhur unit 6 lantai 2 yang digunakan untuk robot. Kemudian peta atau layout dilorong Universitas Budi Luhur unit 6 lantai 2 ini untuk menjalankan pemetaan SLAM gmapping. Setelah itu ada beberapa halangan yang ada didepan robot mobile tersebut. Kemudian warna kuning posisi satu robot, warna biru posisi kedua robot, dan warna hijau posisi tiga robot. Bahwa didepan sensor lidar tidak dapat objek atau memantulkan sinar laser kepada sensor lidar jadi berwarna abu-abu. Bahwa penjelasan posisi satu sebab tidak dapat objek atau memantulkan sinar laser kepada sensor lidar, posisi dua sebab ada penghalang dari depan sensor lidar dan belakang sensor lidar karena ada tembok, posisi tiga sebab tidak dapat objek atau memantulkan sinar laser kepada sensor lidar.



Gambar 4. 12 area pengujian lorong Universitas Budi Luhur unit 6 lantai 2

Hasil analisis dari SLAM gmapping menggunakan RViz untuk mendapatkan area pengujian dan SLAM gmapping. Hasil peta SLAM gmapping berupa 2D dalam skala abu-abu yang dibedakan menjadi tiga kategori yaitu: warna hitam adalah dinding, warna putih adalah area kosong atau bebas penghalang, warna abu-abu merupakan area yang belum dieksplorasi oleh robot.

V. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa:

- Hasil pengujian untuk mendapatkan area lingkungan berupa 2D dalam skala ada warna, berwarna garis merah pengalng yang ada didepan robot
- Hasil pengujian dari algoritma SLAM gmapping berupa 2D sudah menjadi satu dena, dena ada tiga posisi robot beroda hasilnya dalam skala abu-abu yang dibedakan menjadi enam kategori yaitu: warna hitam adalah dinding, warna putih adalah area kosong atau bebas penghalang, warna abu-abu merupakan area yang belum dieksplorasi oleh robot, warna kuning posisi satu robot, warna biru posisi dua robot, warna hijau posisi tiga robot
- Hasil analisis dari SLAM gmapping menggunakan rviz untuk mendapatkan area pengujian dan SLAM gmapping. Hasil Hasil peta SLAM gmapping berupa 2D dalam skala abu-abu yang dibedakan menjadi tiga kategori yaitu: warna hitam adalah dinding, warna putih adalah area kosong atau bebas penghalang, warna abu-abu merupakan area yang belum dieksplorasi oleh robot.

SARAN

Adapun saran yang dapat diberikan untuk perbaikan penelitian ini dimasa yang akan datang sebagai berikut:

- Untuk menghitung posisi robot menambahkan sensor encoder agar bisa lakukan.

REFERENCES

- [1] D. Erlangga, E. D. R. H S, S. Sunarto, K. Rahardjo T.S, dan F. G, "Sistem Navigasi Mobile Robot Dalam Ruang Berbasis Autonomous Navigation," *J. Mech. Eng. Mechatronics*, vol. 4, no. 2, hal. 78, 2019, doi: 10.33021/jmem.v4i2.823.
- [2] A. N. Ranaminanta, D. Nurcipto, dan M. A. Heryanto, "Pemetaan Ruang Dua Dimensi Menggunakan Sensor Lidar 360 Derajat Pada Mobile Robot," vol. 4, hal. 30–38, 2022.
- [3] Werede Gunaza Teame, Dr. Yanan Yu, dan Professor Wang Zhongmin, "Optimization of SLAM Gmapping based on Simulation," *Int. J. Eng. Res.*, vol. V9, no. 04, hal. 74–81, 2020, doi: 10.17577/ijertv9is040107.
- [4] B. A. Issa, "RP Lidar Sensor for Multi-Robot Localization using Leader Follower Algorithm," vol. 15, no. 2, 2019.
- [5] M. A. Maulyda, E. Hidayanto, dan S. Rahardjo, "Representation of Trigonometry Graph Function Colage Students Using GeoGebra," *Int. J. Trends Math. Educ. Res.*, vol. 2, no. 4, hal. 1–7, 2019, doi: 10.33122/ijtmer.v2i4.
- [6] T. Lane, "A short history of robotic surgery," *Ann. R. Coll. Surg. Engl.*, vol. 100, hal. 5–7, 2018, doi: 10.1308/rcsann.suppl1.5.
- [7] S. Pietrzik dan B. Chandrasekaran, "Setting up and Using ROS-Kinetic and Gazebo for Educational Robotic Projects and Learning," *J. Phys. Conf. Ser.*, vol. 1207, no. 1, 2019, doi: 10.1088/1742-6596/1207/1/012019.
- [8] K. Shabalina, A. Sagitov, K.-L. Su, K.-H. Hsia, dan E. Magid, "Avrora Unior Car-like Robot in Gazebo Environment," *Proc. Int. Conf. Artif. Life Robot.*, vol. 24, hal. 116–119, 2019, doi: 10.5954/icarob.2019.os4-3.
- [9] M. Kim, J. Cho, S. Lee, dan Y. Jung, "Imu sensor-based hand gesture recognition for human-machine interfaces," *Sensors (Switzerland)*, vol. 19, no. 18, hal. 1–13, 2019, doi: 10.3390/s19183827.
- [10] G. Grisetti, C. Stachniss, dan W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, hal. 34–46, 2007, doi: 10.1109/TRO.2006.889486.
- [11] H. Yoshida, H. Fujimoto, D. Kawano, Y. Goto, M. Tsuchimoto, dan K. Sato, "Range extension autonomous driving for electric vehicles based on optimal velocity trajectory and driving braking force distribution considering road gradient information," *IECON 2015 - 41st Annu. Conf. IEEE Ind. Electron. Soc.*, no. Figure 1, hal. 4754–4759, 2015, doi: 10.1109/IECON.2015.7392843.
- [12] E. Yurisever, J. Lambert, A. Carballo, dan K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, hal. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.
- [13] M. Mirdanies dan R. P. Saputra, "Experimental review of distance sensors for indoor mapping," *J. Mechatronics, Electr. Power, Veh. Technol.*, vol. 8, no. 2, hal. 85–94, 2017, doi: 10.14203/j.mev.2017.v8.85-94.
- [14] M. Teichmann dan B. Mishra, "Probabilistic algorithms for efficient grasping and fixturing," *Algorithmica (New York)*, vol. 26, no. 3–4, hal. 345–363, 2000, doi: 10.1007/s004539910017.
- [15] A. Priyambudi, B. Firman, dan S. Kristiyana, "Kendali Kecepatan Motor Pada Robot Dengan Empat Roda Omni Menggunakan Metode," *J. Teknol. Technoscintia*, vol. 10, no. 2, hal. 209–217, 2018, [Daring]. Tersedia pada: [https://ejournal.akprind.ac.id/index.php/technoscintia/a](https://ejournal.akprind.ac.id/index.php/technoscintia/article/view/104)

rticle/view/104.

- [16] I. Arun Faisal, T. Waluyo Purboyo, dan A. Siswo Raharjo Ansori, "A Review of Accelerometer Sensor and Gyroscope Sensor in IMU Sensors on Motion Capture," *J. Eng. Appl. Sci.*, vol. 15, no. 3, hal. 826–829, 2019, doi: 10.36478/jeasci.2020.826.829.
- [17] M. N. Masrukhan, M. P. Mulyo, D. Ajatmo, dan M. Ali, "Optimasi Kecepatan Motor DC Menggunakan PID dengan Tuning Ant Colony Optimization (ACO) Controller," *Pros. SENTIA-2016*, vol. 8, no. 1, hal. B49–B52, 2016.
- [18] M. Du, R. F. Ribeiro, dan J. Yuen-Zhou, "Remote Control of Chemistry in Optical Cavities," *Chem*, vol. 5, no. 5, hal. 1167–1181, 2019, doi: 10.1016/j.chempr.2019.02.009.
- [19] Syaiful Ahdan dan Setiawansyah, "Pembangunan Sistem Informasi Geografis Untuk Pendonor Darah dengan Algoritma Dijkstra berbasis Android," *J. Sains dan Inform.*, vol. 6, no. 2, hal. 67–77, 2020, [Daring]. Tersedia pada: <http://doi.org/10.22216/jsi.v6i2.5573>.