

Wea Aqiel Dhimas S

Universitas 17 Agustus 1945 Surabaya, weadhimas@gmail.com

Abstract

The development of robots in the era of the industrial revolution is increasing rapidly. Robots are now a necessity to help and even replace human work. Current robot control can be made easier by applying the ROS (Robot Operating System). ROS is open source so it can be developed freely and many developments throughout the world. The advantages of ROS inspire researchers to use it to control car robots. The ROS implementation requires Jetson Nano and Arduino hardware. The robot is driven by 4 wheels with a DC motor with L298N driver control. The research process includes designing tools and motor drives, coding, testing robot performance. Robot design consists of input, controller, output, and monitoring. The stages of implementing ROS consist of NVIDIA Jetson Nano Software Installation, Ubuntu Installation, NVIDIA Jetson Nano Software Configuration, ROS Installation, Arduino Installation and Rosserial Package. The results obtained by the mobile robot managed to move in 4 directions, namely forward, backward, right and left with teleoperation commands.

Keywords: Robot Operating System, Car Robot, Jetson Nano, Arduino

Abstrak

Perkembangan robot di era revolusi industri semakin pesat. Robot saat ini menjadi kebutuhan untuk membantu bahkan menggantikan pekerjaan manusia. Pengendalian robot saat ini dapat dipermudah dengan mengaplikasikan ROS (Robot Operating System). ROS bersifat open source sehingga dapat dikembangkan secara leluasa dan banyak pengembangannya diseluruh dunia. Kelebihan dari ROS menginspirasi peneliti untuk memanfaatkannya untuk kendali robot mobil. Implementasi ROS memerlukan perangkat keras Jetson Nano dan Arduino. Robot digerakkan 4 roda dengan motor DC dengan kendali driver L298N. Proses penelitian meliputi perancangan alat dan drive motor, coding, pengujian kinerja robot. Perancangan robot terdiri dari input, kontroler, output, dan monitoring. Tahapan penerapan ROS terdiri dari Instalasi Software NVIDIA Jetson Nano, Instalasi Ubuntu, Konfigurasi Software NVIDIA Jetson Nano, Instalasi ROS, Instalasi Arduino dan Rosserial Package. Hasil yang diperoleh robot mobil berhasil bergerak 4 arah yaitu maju, mundur, kanan, dan kiri dengan perintah teleoperasi.

Kata Kunci: Robot Operating System, Robot Mobil, Jetson Nano, Arduino

Pendahuluan

Dalam dunia robotik system kendali merupakan bagian terpenting dalam penerapannya dan telah berkembang sangat luas dan pesat seiring dengan teknologinya, berbagai macam sistem pengendalian telah diciptakan dan dikembangkan, namun pengendalian sistem jarak jauh dirasa masih kurang misalkan ada suatu kejadian yang mengharuskan suatu pekerjaan khusus dalam sebuah mobil robot tentunya keakuratan data akan selalu menjadi tolok ukur bagi sang operator yang pada akhirnya bisa menjadi analisa dan evaluasi dalam pengambilan keputusan.[1]

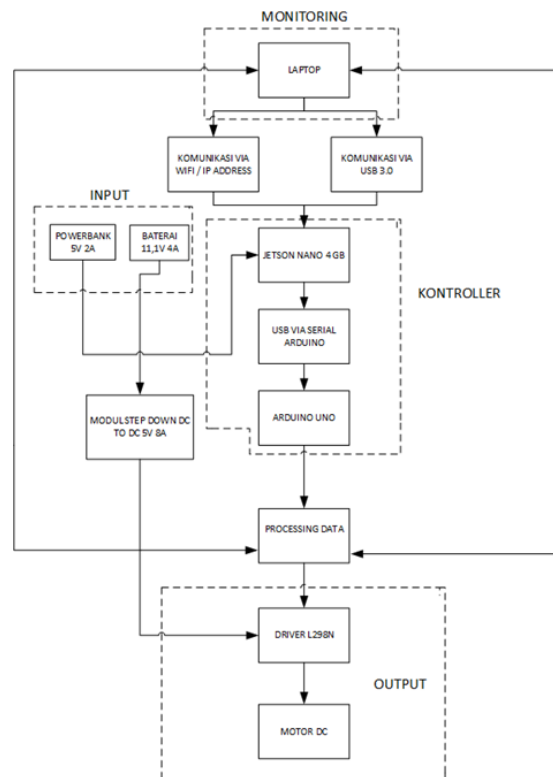
Dalam perkembangannya robot tidak lebih dari dorongan kebutuhan manusia itu sendiri secara kodrat, di era industri 4.0 yang semakin pesat dan semakin kompleksitas maka jawabannya adalah semua pekerjaan diberikan solusi kemudahan dengan memanfaatkan kinerja robot. Adapun tujuan dalam penelitian ini adalah untuk pembuatan prototype modul dalam kinerja mobil robot dengan menggunakan perangkat lunak yaitu Arduino uno yang dapat diprogram dengan bahasa C++ dan dikendalikan lewat komputer atau laptop. Modul bisa sebagai sarana pembelajaran bagi pelajar menengah, tingkat serta kalangan mahasiswa.

Metode

Proses penelitian meliputi 3 tahapan yaitu:

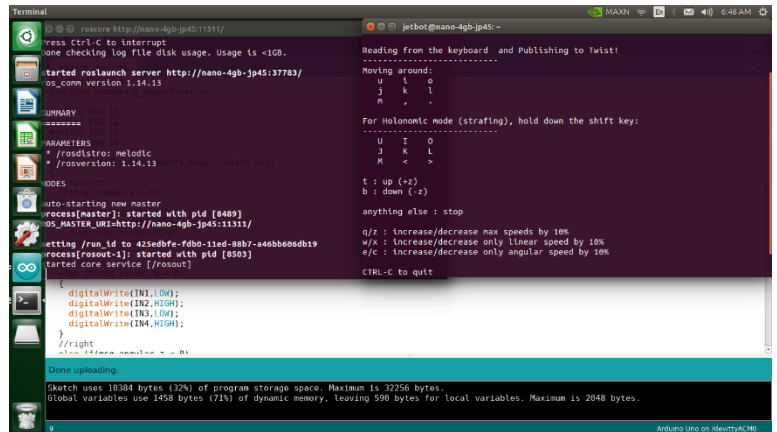
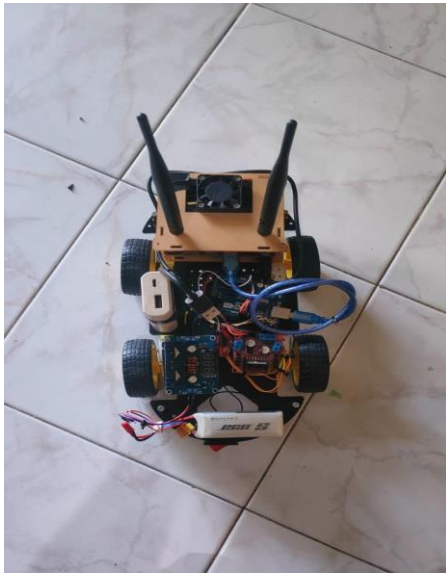
- a. Perancangan Alat & Driver motor L298N yaitu meliputi Jetson NANO 4GB , Arduino UNO & Driver motor L298N yang berfungsi sebagai alat mikrokontroler yang fungsinya untuk berkomunikasi menggerakkan motor DC.[2]

- b. Perancangan Coding berupa python dan Sinkronisasi yang bertugas pembuatan coding python yang dibuat di JETSON NANO & Arduino yang akan disinkronisasi/diterima di Driver motor L298N.[3]
 - c. Pengujian & Hasil dari Pergerakan coding tersebut berfungsi tahap finalisasi/bergeraknya dari coding tersebut dan menghasilkan sebuah pergerakan pada mobil.[4]
- Perancangan sistem kendali robot mobil berbasis ROS dipaparkan pada Gambar 1. Setiap proses dari sistem kendali dipaparkan sebagai berikut.
- a. Input merupakan elemen sistem yang bertugas memberikan data masukan yang diperlukan oleh kontroller untuk diolah lebih lanjut sesuai keperluan sistem. Adapun inputan yang digunakan berupa *Powerbank 5v 2A* & *Baterai 11,1 V 4A*. [5]
 - b. Kontroler merupakan elemen inti dari sistem yang mampu mengolah data dan mengeksekusi data dengan perangkat lunak yang telah dirancang untuk kebutuhan sistem. Dalam hal ini kontroler yang dipakai berupa *Arduino Uno* karena merupakan mikrokontroler yang didesain khusus untuk mempermudah perancangan sistem elektronika karena untuk mempelajari bahasa pemrograman *Arduino Uno* mudah dan ketersediaan library yang lengkap. [6]
 - c. Output merupakan elemen akhir sistem yang menerima data yang telah diolah oleh kontroler untuk dikerjakan sistem keluaran sesuai kebutuhan sistem yang diinginkan. Untuk itu output sistem yang digunakan berupa *Motor DC* yang menggunakan driver *L298N*. [7]
 - d. Monitoring merupakan elemen yang digunakan untuk mengambil data maupun mengirim data melalui komunikasi serial. Untuk monitoring sistem menggunakan laptop. [8]



Gambar 1. Blok Diagram Sistem Kontroler Robot Mobil berbasis ROS

Hasil Dan Pembahasan



Gambar 2. Prototype robot mobil berbasis ROS

Gambar 3. Perintah Arah untuk menjalankan robot melalui ROS

Sesuai tujuan penelitian untuk dapat mengendalikan robot mobil jarak jauh berbasis ROS, hasil percobaan dipaparkan pada Tabel 1[9]. Sistem yang dirancang dapat menghasilkan gerakan robot mobil bergerak maju, mundur, kanan, dan kiri. Robot dikendalikan melalui *software* dengan input tombol pada keyboard sebagai penentu arah gerak dari robot. Berdasarkan hasil pengujian, robot mobil berhasil bergerak sesuai desain kendali yang diharapkan.[10]

Tabel 1. Hasil Pengujian Kendali Robot

No	Arah	Tombol Keyboard	Pergerakan Roda	Keterangan (Berhasil / Tidak)
1	Maju	I	Semua roda bergerak searah jarum jam	Berhasil
2	Mundur	M	Semua roda bergerak berlawanan arah jarum jam	Berhasil
3	Kanan	L	2 roda sebelah kiri bergerak searah jarum jam, 2 roda sebelah kanan tidak bergerak	Berhasil
4	Kiri	J	2 roda sebelah kanan bergerak searah jarum jam, 2 roda sebelah kiri tidak bergerak	Berhasil

Kesimpulan

Berdasarkan hasil percobaan pada robot tersebut, untuk beberapa arah masih belum maksimal dikarenakan masih menggunakan motor saja tanpa encoder. Dengan jarak komunikasi yang terbilang lumayan jauh robot ini lebih disarankan di area indoor karena

cakupan jarak komunikasinya yang sangat memadai daripada di area outdoor. Kecepatan yang bisa diatur sesuai keinginan menggunakan step down DC to DC bahwa robot ini sangat fleksibel dan mudah untuk dioperasikan bila ingin mengembangkan lebih detail lagi bisa ditambahkan perangkat IoT / Neural Network + encoder agar saat melaju lebih halus dan lebih mutakhir. pengembangan citra digital seperti face recognition data, pengenalan objek yang semakin luas dan dapat terverifikasi dengan cepat, Penambahan sensor encoder yang kedepannya bisa memperhalus gerak arah robot tersebut. Penambahan joystick penambahan visualisasi secara Real-time.

ROS (*Robot Operating System*) terbukti dapat diaplikasikan untuk pengendalian robot mobil secara jarak jauh. Robot berhasil bergerak maju, mundur, kanan, dan kiri. Perangkat keras untuk dapat memanfaatkan ROS untuk kendali robot mobil berupa Jetson Nano, Arduino, driver L298N, dan baterai. Hasil pengujian ini dapat dijadikan literasi oleh dunia industri bahwa kendali robot sudah semakin canggih dan aplikasinya dapat digunakan untuk kendaraan tanpa awak yang harus melalui area-area berbahaya seperti area gas beracun atau yang lainnya.

Ucapan Terima Kasih

Terima Kasih kepada pihak Universitas 17 Agustus 1945 Surabaya yang sudah menyediakan wadah untuk mempublikasi jurnal. Saya harap jurnal ini dapat dimanfaatkan sebaik - baiknya.

Daftar Pustaka

- [1] S. Pramod Thale, M. Mangesh Prabhu, P. Vinod Thakur, and P. Kadam, "ROS based SLAM implementation for Autonomous navigation using Turtlebot," *ITM Web Conf.*, vol. 32, p. 01011, 2020, doi: 10.1051/itmconf/20203201011.
- [2] Muhammad Luqman Bukhori and Erwan Eko Prasetyo, "Sistem Deteksi Masker Berbasis Jetson Nano dengan Deep Learning Framework TensorFlow," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 12, no. 1, 2023, doi: 10.22146/jnteti.v12i1.5472.
- [3] A. Saphala and P. I. Tanaya, "IMPLEMENTATION AND RECONFIGURATION OF ROBOT OPERATING SYSTEM ON HUMAN," vol. 9, no. 2, pp. 59-65, 2015.
- [4] M. Kiftiyah, Santoso, and Munsyi, "Robot Pendeteksi Warna," *J. Sains dan Inform.*, vol. Vol.1 No., no. 2, pp. 38-47, 2015.
- [5] H. A. M. ari, A. N. Jati, and C. Setianingsih, "Realisasi Multi Robot Simultaneous Localization And Mapping Berbasis Sensor Kamera," *eProceedings Eng.*, vol. 6, no. 2, 2019.
- [6] S. Santoso and S. Mursyid, "Kontrol Proportional Integral (Pi) Pada Robot Line Follower," *J. Sains dan Inform.*, vol. 1, 2015, [Online]. Available: <http://jsi.politala.ac.id/index.php/JSI/article/view/73%0Ahttps://jsi.politala.ac.id/index.php/JSI/article/download/73/40>
- [7] R. Hartayu, S. Santoso, A. O. U. Kaleka, and M. K. Musakhol, "Desain Simulasi Robot Kesetimbangan Dua Roda Dengan Kecerdasan Buatan," *J. Sains dan Inform.*, vol. 6, no. 2, pp. 175-182, 2020, doi: 10.34128/jsi.v6i2.232.
- [8] S. Siti Fatimah, "Robot Pendeteksi Api," *sains dan Teknol.*, vol. 1, no. 24600-173X, pp. 71-

- 80, 2015, [Online]. Available: <https://jsi.politala.ac.id/index.php/JSI/article/view/99>
- [9] K. L. Besseghieur, R. Trębiński, W. Kaczmarek, and J. Panasiuk, "Trajectory tracking control for a nonholonomic mobile robot under ROS," *J. Phys. Conf. Ser.*, vol. 1016, no. 1, 2018, doi: 10.1088/1742-6596/1016/1/012008.
- [10] A. Jalil, "PEMANFAATAN MIDDLEWARE ROBOT OPERATING SYSTEM (ROS) DALAM MENJAWAB TANTANGAN REVOLUSI INDUSTRI 4.0," *Ilk. J. Ilm.*, vol. 11, no. 1, 2019, doi: 10.33096/ilkom.v11i1.412.45-52.