

# Implementasi ROS (Robot Operating System) Pada Sistem Kendali Jarak Jauh Robot Bergerak Jenis Non-holonomic

**Ajang Nurdin, MS Hendriyawan Achmad**

*Program Studi Teknik Teknik Elektro, Fakultas Informasi Teknologi dan Elektro  
Universitas Teknologi Yogyakarta  
Jl. Ringroad Utara Jombor Sleman Yogyakarta  
E-mail : [ajangnur7@gmail.com](mailto:ajangnur7@gmail.com)*

## ABSTRAK

Secara umum kegunaan robot adalah untuk menggantikan kerja manusia yang membutuhkan ketelitian yang tinggi atau mempunyai resiko yang sangat besar atau bahkan mengancam keselamatan manusia. Sebuah robot dapat saja dibuat untuk berbagai macam aktifitas, namun sebuah robot harus dibuat dengan tujuan untuk kebaikan manusia. Telah kita ketahui bahwa didalam menjalankan kehidupan sehari-harinya manusia tidak terlepas dari segudang pesawat elektronika, di zaman modern seperti sekarang inilah khususnya. Untuk berkomunikasi, mencari informasi, serta sebagai hiburan yang dapat ditampilkan secara audio maupun visual. Sifat keingintahuan manusia membuatnya terus mengembangkan ilmunya mempelajari lingkungan sekitarnya, tumbuhan, hewan-hewan. Tetapi pada saat keingintahuan itu terbentur pada kondisi, yaitu saat manusia tidak mungkin mendekati objek yang ingin ditelitinya. Maka dari itu tersirat sebuah pamikiran untuk membuat sebuah alat pengoprasian robot sistem IoT (Internet of Things) yang dapat dikontrol dari jarak jauh (teleoperasi). Sistem pengendalian robot jarak jauh berbasis ROS (Robot Operating System) pada layanan teleoperasi merupakan pengaplikasian terhadap perkembangan teknologi komunikasi bergerak. Robot ini dibangun untuk dapat memonitor suatu lokasi serta dapat memberikan informasi visual kepada user melalui layanan teleoperasi secara otomatis dan real time. Prinsip kerja sistem ini adalah server mengendalikan robot dengan sebuah inputan yaitu joystick dari server yang terhubung ke internet, lalu dari internet terhubung ke client sebagai module robot yang terhubung ke mikrokontroler Arduino, kemudian dari Arduino terhubung ke sistem penggerak untuk mengendalikan robot client yang dapat bergerak maju, mundur, belok kanan, dan belok kiri. Kemudian server juga akan menerima informasi visual audio-video. Pada penelitian ini dapat dibuktikan bahwa prototipe teleoperasi robot berbasis ROS (Robot Operating Sistem) dapat diimplementasikan untuk fungsi eksplorasi jarak jauh. Sistem yang dibangun, yakni sistem pengendalian robot jarak jauh yang terdiri dari robot, yang dilengkapi dengan informasi visual audio-video yang diterima server dari client (robot). Dari hasil percobaan ini, server dapat mengendalikan lebih dari satu fungsi sistem robot secara bersamaan, dalam waktu yang bersamaan, dan tidak dibatasi oleh jarak selama terkoneksi dengan internet.

**Kata kunci :** Robot, Teleoperasi, Arduino, IoT (Internet of things), ROS (Robot Operating System).

## 1. PENDAHULUAN

Sistem kendali telah berkembang dengan sangat luas, berbagai macam metoda pengendalian telah diciptakan dan dikembangkan. Tetapi pengendalian sistem jarak dari jauh masih dirasa kurang, misalkan ketika suatu perangkat keras atau boleh dikatakan mobil robot mencari suatu kejadian ditempat yang berbahaya mobil robot bisa dipantau oleh seorang operator dari jarak jauh untuk mengendalikan apa yang akan dilakukannya sehingga data yang didapat langsung dapat diterima oleh operator untuk selanjutnya dianalisa. Sistem kendali mobil robot bukan saja memudahkan dan meningkatkan prestasi kerja instansi, tetapi juga membantu manusia untuk

menggantikan sebagian dari tugasnya karena tidak mungkin atau tidak perlu dilakukannya sendiri.

Pengembangan robot tidak lebih karena dorongan kebutuhan manusia itu sendiri terhadap kehadiran sebuah individu yang mampu diandalkan dalam melakukan sebuah kegiatan yang rumit dan berbahaya, bahwa setiap manusia tidak pernah luput dari kesalahan. Baik kesalahan menghitung atau kesalahan lainnya yang dapat saja berakibat fatal. Belum lagi manusia memiliki kekuatan dan daya tahan tubuh yang sangat terbatas dan sulit diperhiutngkan.

Sistem kendali mobil robot adalah salah satu pemanfaatan sistem kendali dengan menggunakan PC dimana dapat mengontrol perangkat keras yang dibuat dengan menggunakan PC, kemana gerak yang diinginkan kita mengontrol posisinya, penulis ingin membuktikan bahwa robot dapat mengendalikan lebih dari satu fungsi sistem robot secara bersamaan dari jarak jauh, dalam waktu yang bersamaan, dan tidak dibatasi oleh jarak selama terkoneksi dengan internet.

Salah satu solusinya adalah membuat perangkat keras dan perangkat lunak untuk pengendalian sistem robot, pengendalian ini dilakukan dengan pengendalian jarak jauh robot *non-holonomic* berbasis ROS (Robot Operating System) pada layanan teleoperasi merupakan pengaplikasian terhadap perkembangan teknologi komunikasi bergerak. Robot ini dibangun untuk dapat memonitor suatu lokasi serta dapat memberikan informasi visual *audio-video* kepada user melalui layanan teleoperasi secara otomatis dan real time.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian Tugas Akhir

Tahap awal penelitian ini adalah melakukan analisis tentang teleoprasi dengan program ROS (*Robot Operating System*), untuk dapat beroperasi secara jarak jauh dengan waktu yang bersamaan menggunakan sistem IoT (*Internet of Things*), selanjutnya melakukan pengambilan data pada lingkungan sekitar dengan menggunakan *audio-video* untuk mendapatkan hasil data secara maksimal.

Setelah mendapatkan data dari hasil analisis, tahap selanjutnya adalah membuat perancangan sistem ini, tahap perancangan meliputi:

1. Membuat diagram alur kerja sistem secara keseluruhan.
2. Perancangan perangkat keras yang meliputi:
3. Perancangan aplikasi komunikasi dan pengolahan data yang berbasis ROS.
4. Perancangan mobil robot yang menggunakan 2 buah motor DC yang dipasang ditengah-tengah yang digunakan untuk maju mundur, berputar, dan belok kanan kiri.

Pengujian dilakukan guna untuk mengetahui apakah sistem sudah bekerja sesuai dengan apa yang diinginkan. Pengujian ini dilakukan secara berulang-ulang agar sistem ini bisa berjalan dengan baik. Pengujian ini meliputi beberapa bagian antara lain:

1. Pengujian teleoperasi dengan program ROS (*Robot Operating System*), untuk dapat berinteraksi secara jarak jauh dengan waktu

yang bersamaan menggunakan system IoT (*Internet of Things*)

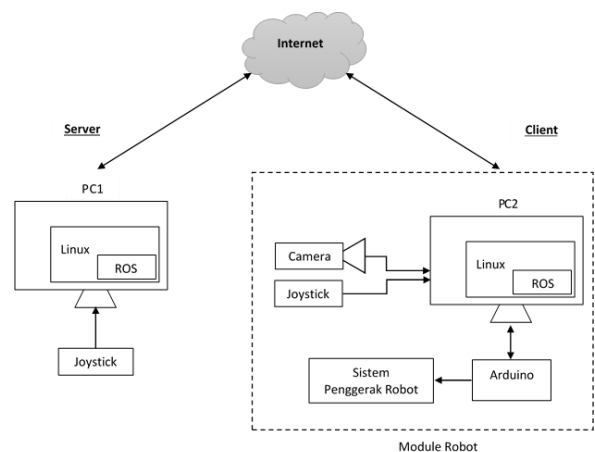
2. Pengujian kamera
3. Pengujian Driver motor
4. Pengujian keadaan mobil robot untuk bergerak maju, mundur, belok kanan, dan belok kiri.

### 2.2 Alat dan Bahan

- a. Alat Penelitian
  1. Laptop Acer Aspire One 722  
Laptop ini digunakan sebagai *client* yang dipasang pada module robot.
  2. Laptop Lenovo ThinkPad R60  
Laptop ini digunakan sebagai *server* yang dikendalikan oleh *user*.
  3. Sistem Operasi yang digunakan adalah Ubuntu 14.04.5 LTS
  4. Perangkat lunak yang digunakan
    - ROS (*Robot Operating System*)
    - Arduino IDE
- b. Bahan Penelitian
  1. Board Arduino Mega 2560
  2. Shield L293D
  3. USB Kamera
  4. *Joystick*
  5. Baterai Lipo 3 Cell
  6. Motor DC
  7. Kabel-kabel

### 2.3 Diagram Blok Sistem

Gambar 1 dibawah ini adalah diagram blok keseluruhan sistem untuk mengetahui representasi dari fungsi dari komponen didalam sistem pengendalian dan hubungannya antara satu komponen dengan komponen yang lain.

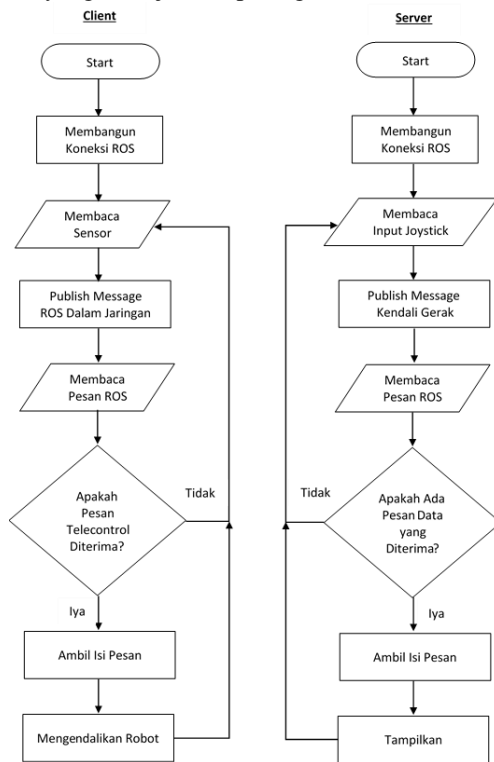


Gambar 1: Diagram Blok Sistem Robot

Prinsip kerja sistem ini adalah *server* mengendalikan robot dengan sebuah inputan yaitu *joystick* yang terhubung ke internet, lalu dari internet terhubung ke *client* sebagai module robot yang terhubung ke *microcontroller* Arduino, kemudian dari Arduino terhubung ke sistem penggerak untuk mengendalikan robot *client* yang dapat bergerak maju, mundur, belok kanan, dan belok kiri. Kemudian server juga akan menerima informasi visual *audio-video*.

## 2.4 Flowchart Sistem

Flowchart pada sistem ini terdapat dua bagian yaitu flowchart sebagai *client* dan flowchart sebagai *server* seperti yang ditunjukkan pada gambar 2 dibawah ini.



Gambar 2: Flowchart sebagai *client* dan flowchart sebagai *server*

Flowchart sebagai *client*, awal mulai adalah membangun koneksi ROS. Setelah itu proses pembacaan sensor, lalu *publish message* ROS dalam jaringan, setelah itu proses pembacaan ROS. Ketika pesan *telecontrol* diterima maka akan mengambil isi pesan dan diteruskan untuk mengendalikan robot, jika pesan *telecontrol* tidak diterima maka akan kembali untuk proses membaca sensor.

Sedangkan untuk flowchart sebagai *server*, awal mulai adalah membangun koneksi ROS. Setelah itu mengendalikan aplikasi *telecontrol*, lalu proses pembacaan pesan ROS. Ketika ada pesan, maka akan mengambil isi pesan dan akan menmpilkannya dilayar,

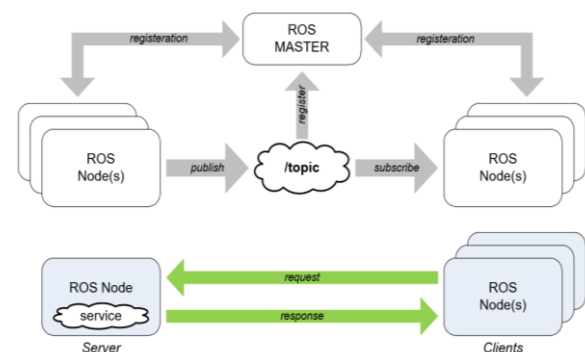
jika tidak maka akan kembali untuk mengendalikan aplikasi *telecontrol*.

## 2.5 Robot Operating System (ROS)

ROS adalah *framework* untuk membuat program robot yang fleksibel dan mudah digunakan. ROS berisi beragam peralatan (*tool*), *library*, *driver*, dan konvensi yang bertujuan untuk memudahkan pembuatan program yang kompleks dan handal pada berbagai *platform* robot. Sederhananya, dengan adanya ROS, setiap programmer menggunakan style yang sama dalam membuat sebuah program untuk robot jenis apapun.

*Robot Operating System* sebenarnya adalah *meta-operating system* atau *framework* yang bersifat *open source* yang dapat digunakan untuk robot. ROS mempunyai sebuah *service* seperti halnya sebuah sistem operasi pada umumnya, termasuk abstraksi perangkat keras, kendali perangkat tingkat bawah, implementasi dari fungsi-fungsi yang biasa digunakan, penyampaian pesan/data di antara proses serta management package. ROS juga menyediakan alat dan *library* yang memungkinkan kita untuk mendapatkan, membangun, memprogram hingga menjalankan program melalui banyak computer.

Kemampuan ROS multi-node dapat mengontrol banyak fungsi operasional robot. Robot yang memiliki tugas besar menjalankan banyak fungsi dan canggih, penerapan cara tradisional akan menyita waktu dan energi para peneliti. Dengan multi-node, properti ROS seperti yang ditunjukkan pada Gambar 3 akan memberikan kontrol aliran robot yang efisien. Selanjutnya, proses membangun perangkat lunak robot tidak butuh waktu lama karena dapat dilakukan secara tim.



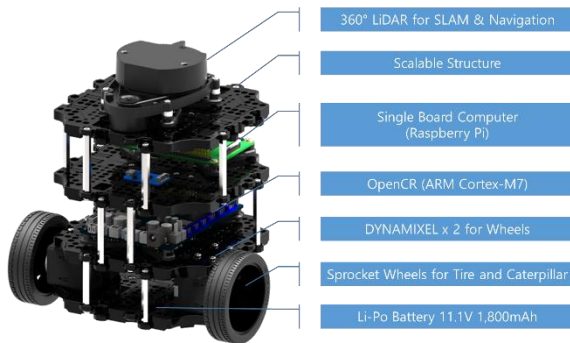
Gambar 3: Prinsip Kerja Kerangka Perangkat Lunak Berbasis ROS untuk Mengontrol Operasi Semua Sistem Robot Multi-Node.

## 2.6 Jenis Robot Kinematika Diferensial

Kinematika robot menggambarkan sifat gerak robot (penggerak) berdasarkan sistem penggerak roda diferensial. Sementara itu, struktur elektronik-mekanik membahas platform padat yang mendukung fungsi

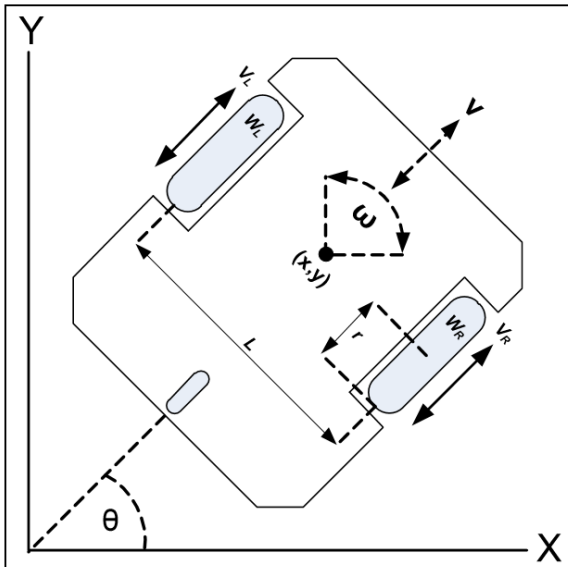
$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

robot dan sistem kontrol. Teknologi mobile robot yang digunakan dalam penelitian ini adalah tipe diferensial non-holonomis dengan sistem beroda. Proses simulasi menggunakan robot TurtleBot yang dikembangkan oleh Willow Garage, seperti yang ditunjukkan pada Gambar 4.



Gambar 4: TurtleBot V.3 Diproduksi oleh Willow Garage.

Robot TurtleBot yang digunakan selama percobaan simulasi memiliki model kinematik robot non-holonomik dan mengacu pada input kecepatan linier ( $v$ ) dan kecepatan sudut ( $\omega$ ). Gambar 5 menunjukkan model kinematik dari mobile robot tipe diferensial yang digunakan oleh TurtleBot. Berdasarkan model kinematik yang ditunjukkan pada Gambar 5, kita dapat mengekspresikan model matematika yang digunakan secara akurat menghitung nilai kecepatan diferensial roda kiri dan kanan untuk mencapai kecepatan linear dan sudut yang diinginkan. Persamaan (1) menjelaskan model kinematika maju dari robot TurtleBot yang mengubah nilai kecepatan ke posisi.



Gambar 5: Robot Model Kinematik dari Mobile Robot Diferensial Tipe Non-Holistik.

(1)

Output ( $x, y, \theta$ ) menunjukkan perbedaan posisi dari waktu ke waktu relatif terhadap koordinat awal berdasarkan input kecepatan linier ( $v$ ) dan kecepatan sudut ( $\omega$ ) yang disediakan oleh pengontrol utama berdasarkan jarak relatif robot ke posisi yang diinginkan (tujuan). Selain menghitung kinematika maju untuk mendapatkan posisi yang diperkirakan, TurtleBot juga mengubah nilai kecepatan input (linier dan sudut) menjadi kecepatan diferensial untuk penggerak roda kanan ( $v_R$ ) dan roda kiri ( $v_L$ ) seperti yang dijelaskan oleh (2) dan (3).

$$v_R = \frac{2v + \omega L}{2r} \quad (2)$$

$$v_L = \frac{2v - \omega L}{2r} \quad (3)$$

$r$  dan  $L$  adalah jari-jari roda robot dan jarak antara dua roda masing-masing, seperti yang ditunjukkan pada Gambar.7. Jika sistem penggerak roda kanan dan kiri dapat bergerak dengan nilai setpoint  $v_R$  dan  $v_L$  maka posisi yang diperkirakan  $[x, y, \theta]^T$  dapat memberikan estimasi yang akurat dari waktu ke waktu. Nilai setpoint  $v_R$  dan  $v_L$  harus dipertahankan oleh sistem kontrol gerak, untuk mendapatkan kecepatan gerak robot yang stabil. Salah satu metode untuk menjaga stabilitas sistem penggerak roda diferensial adalah dengan menggunakan metode PID (proporsional, turunan, integral).

## 2.7 Telerobotic

Bruno Siciliano dan Oussama Khatib dalam buku yang berjudul Springer Handbook Of Robotic, 2008 menuliskan Guenter Niemeyer dkk (2008), menyatakan. Dibandingkan dengan sistem robot polos, di mana robot mengeksekusi gerakan atau program lain tanpa konsultasi lebih lanjut dari pengguna atau operator, sistem telerobotic memberikan informasi kepada dan memerlukan perintah dari pengguna. Arsitektur kendali mereka dapat digambarkan oleh gaya dan tingkat hubungan ini, diselenggarakan dispektrum, tiga kategori utama.

- Direct Control
- Shared Control
- Supervisory Control

Dalam prakteknya, bagaimanapun arsitektur kontrol sering termasuk bagian dari semua strategi.

a. *Direct control*

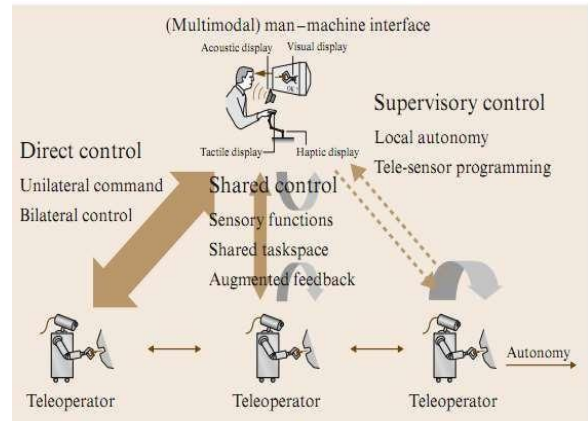
Kontrol langsung menyiratkan ada kecerdasan atau otonomi dalam sistem, sehingga semua gerakan robot langsung dikendalikan oleh pengguna melalui antarmuka master. Jika pelaksanaan tugas dibagi antara kontrol langsung dan umpan balik sensoris local dan otonomi, atau jika umpan balik pengguna ditambah dari penampakan maya yang nyata atau alat bantu otomatis lainnya, arsitektur dilambungkan sebagai kontrol bersama. Dalam kontrol pengawasan pengguna dan robot terhubung dengan baik dalam jaringan local yang bagus, operator memberikan perintah tingkat tinggi, yang kembali didefinisikan dan dieksekusi oleh telerobot tersebut.

b. *Shared Control*

Dalam pengaktifkan telepresence di jarak jauh atau diaplikasikan pada daerah yang berisiko seperti bekas ledakan nuklir (radio aktif), cukup dengan konsep shared control untuk melakuka pengontrolan melalui teleoperator akan lebih baik dan menjamin keselamatan user dalam menjalankan tugas. Maksud dari shared control didasarkan pada loop lokal sensorik umpan balik. Dimana perintah yang dilakukan dapat mengirim sinyal kembali atau dapat didefinisikan secara mandiri teleoperator menyediakan alat bantu sederhana berupa kecerdasan sensorik. Sehingga operator manusia yang melakukan control terhadap perangkat dapat menerima sinyal balik kondisi robot dan dapat membantu operator dalam melakukan tindakan berikutnya sehingga mendapat hasil yang maksimal.

c. *Supervisory Control*

Kontrol pengawasan, diperkenalkan oleh Ferell dan Sheridan pada tahun 1967, berasal dari analog mengawasi anggota staf bawahan manusia. Supervisor memberikan arahan tingkat tinggi untuk menerima dan ringkasan informasi, dalam hal ini, robot. Sheridan menggambarkan pendekatan ini dibandingkan dengan manual dan otomatis kontrol robot operator Manusia sebentar-sebentar pemrograman dan terus menerima informasi dari komputer itu sendiri menutup kontrol lingkaran otonom melalui efektor resmi dan sensor seperti yang ditunjukkan pada gambar 6 dibawah.



Gambar 6: Man-machine interface

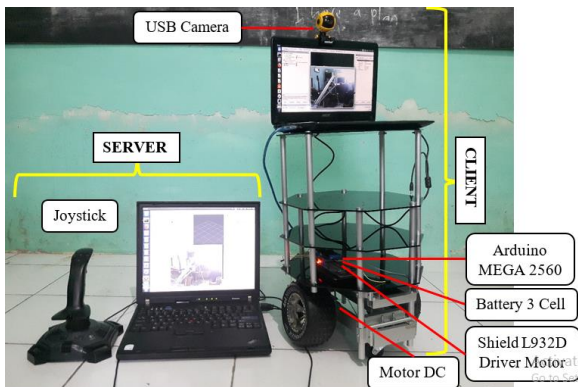
Saat ini loop kontrol otonom dapat dialihkan ke remote, dengan mengandalkan informasi dan model yang sedang dikirim ke situs operator. Operator mengawasi sistem telerobotic dan memutuskan bagaimana harus bertindak dan apa yang harus dilakukan. Sebuah implementasi khusus dari kontrol pengawasan adalah pendekatan pemrograman telesensor.

Pemrograman telesensor (TSP) adalah konsep pendekatan otonomi bersama yang mendistribusikan kecerdasan antara manusia dan mesin. Menganggap bahwa informasi yang mencakup tentang lingkungan yang sebenarnya tersedia dari sistem sensor, tugas parsial dapat dijalankan secara independen pada tingkat mesin.

### 3. HASIL DAN PEMBAHASAN

#### 4.1 Purwarupa

Keseluruhan sistem dalam pembuatan purwarupa penelitian ini terdapat dua bagian yaitu sebagai server dan sebagai client. Komponen yang terdapat pada server yaitu joystick sebagai pengendali robot. Sedangkan komponen yang terdapat pada client yaitu USB Camera untuk menampilkan informasi visual audio-video, Arduino MEGA 2560 sebagai mikrokontroler, baterai 3 cell sebagai power supply motor DC, Shield L932D sebagai driver motor DC, dan motor DC sebagai penggerak robot. Keseluruhan sistem purwarupa ditunjukkan pada gambar 7 dibawah ini.



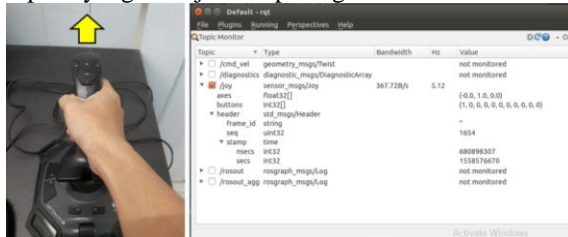
Gambar 7: Keseluruhan Sistem Purwarupa

#### 4.1 Pengujian dan Pembahasan

a. Pengujian hasil nilai data posisi pada *joystick*  
 Pengujian ini akan menampilkan hasil nilai data posisi *joystick*. Hasil nilai data akan tampil dengan menuliskan `rtt` pada terminal.

##### 1. *Joystick* posisi maju

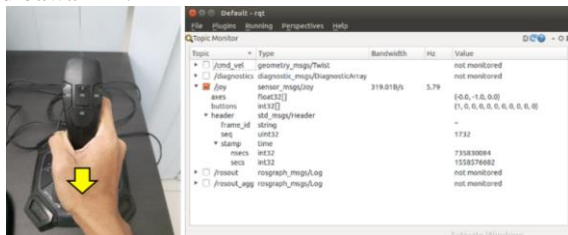
Hasil nilai data yang didapatkan ketika *joystick* posisi maju. Pada *angular velocity* (-0.0) dan pada *linier velocity* (1.0), maka roda kiri dan kanan bergerak maju. Seperti yang ditunjukkan pada gambar 8 dibawah ini.



Gambar 8: Hasil Nilai Data yang Ditampilkan Ketika *Joystick* Posisi Maju

##### 2. *Joystick* posisi Mundur

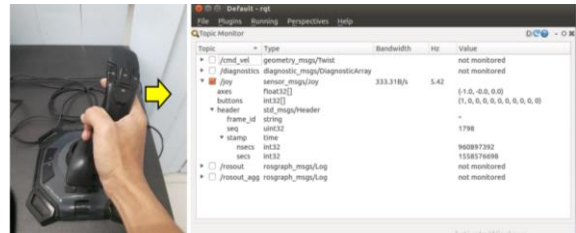
Hasil nilai data yang didapatkan ketika *joystick* posisi maju. Pada *angular velocity* (-0.0) dan pada *linier velocity* (-1.0), maka roda kiri dan kanan bergerak mundur. Seperti yang ditunjukkan pada gambar 9 dibawah ini.



Gambar 9: Hasil Nilai Data yang Ditampilkan Ketika *Joystick* Posisi Mundur

##### 3. *Joystick* posisi ke kanan

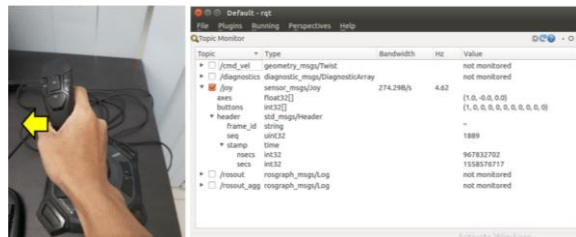
Hasil nilai data yang didapatkan ketika *joystick* posisi ke kanan. Pada *angular velocity* (-1.0) dan pada *linier velocity* (-0.0), maka roda kiri dan kanan tidak bergerak. Seperti yang ditunjukkan pada gambar 10 dibawah ini.



Gambar 10: Hasil Nilai Data yang Ditampilkan Ketika *Joystick* Posisi ke Kanan

##### 4. *Joystick* posisi ke kiri

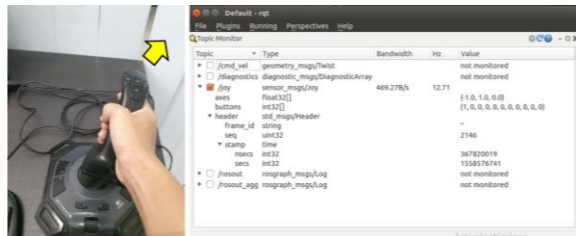
Hasil nilai data yang didapatkan ketika *joystick* posisi ke kiri. Pada *angular velocity* (1.0) dan pada *linier velocity* (-0.0), maka roda kiri dan kanan tidak bergerak. Seperti yang ditunjukkan pada gambar 11 dibawah ini.



Gambar 11: Hasil Nilai Data yang Ditampilkan Ketika *Joystick* Posisi ke Kiri

##### 5. *Joystick* posisi serong kanan arah depan

Hasil nilai data yang didapatkan ketika *joystick* posisi serong kanan arah depan. Pada *angular velocity* (-1.0) dan pada *linier velocity* (1.0), maka roda kanan lebih cepat daripada roda kiri dan rah putaran roda kedepan (maju). Seperti yang ditunjukkan pada gambar 12 dibawah ini.

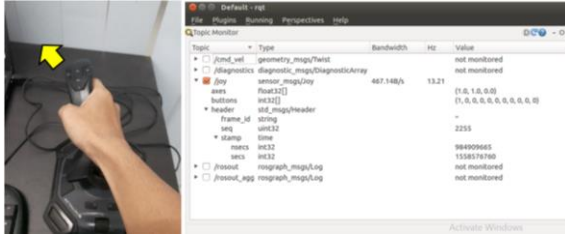


Gambar 12: Hasil Nilai Data yang Ditampilkan Ketika *Joystick* Posisi Serong Kanan Arah Depan

##### 6. *Joystick* posisi serong kiri arah depan

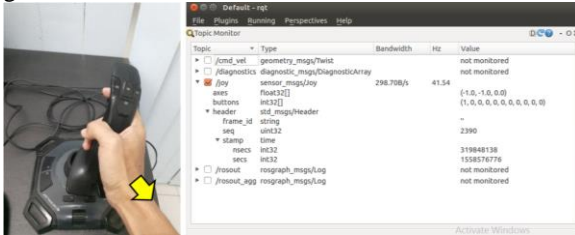
Hasil nilai data yang didapatkan ketika *joystick* posisi serong kiri arah depan. Pada *angular velocity* (1.0) dan

pada *linier velocity* (1.0), maka roda kiri lebih cepat daripada roda kanan dan arah putaran roda kedepan (maju). Seperti yang ditunjukkan pada gambar 13 dibawah ini.



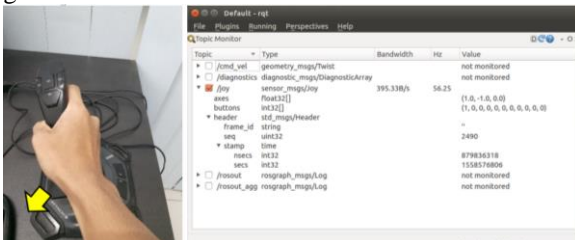
Gambar 13: Hasil Nilai Data yang Ditampilkan Ketika Joystick Posisi Serong Kiri Arah Depan

7. *Joystick* posisi serong kanan arah belakang Hasil nilai data yang didapatkan ketika *joystick* posisi serong kanan arah belakang. Pada *angular velocity* (-1.0) dan pada *linier velocity* (-1.0), maka roda kiri lebih cepat daripada roda kanan dan arah putaran roda kebelakang (mundur). Seperti yang ditunjukkan pada gambar 14 dibawah ini.



Gambar 14: Hasil Nilai Data yang Ditampilkan Ketika Joystick Posisi Serong Kanan Arah Belakang

8. *Joystick* posisi serong kiri arah belakang Hasil nilai data yang didapatkan ketika *joystick* posisi serong kiri arah belakang. Pada *angular velocity* (1.0) dan pada *linier velocity* (-1.0), maka roda kanan lebih cepat daripada roda kiri dan arah putaran roda kebelakang (mundur). Seperti yang ditunjukkan pada gambar 15 dibawah ini.



Gambar 15: Hasil Nilai Data yang Ditampilkan Ketika Joystick Posisi Serong Kanan Arah Belakang

## b. Pengujian Gerak Kendali Pada Robot

Pengujian ini akan dilakukan kendali gerak maju, mundur, belok ke kiri dan ke kanan.

### 1. Posisi gerak maju

Ketika *joystick* di gerakan posisi kedepan (maju), maka robot akan bergerak maju kedepan. Seperti yang ditunjukkan pada gambar 16 dibawah ini.



Gambar 16: Posisi Robot Bergerak Maju Kedepan

### 2. Posisi gerak mundur

Ketika *joystick* di gerakan posisi kebelakang (mundur), maka robot akan bergerak mundur kebelakang. Seperti yang ditunjukkan pada gambar 17 dibawah ini.



Gambar 17: Posisi Robot Bergerak Mundur Kebelakang

### 3. Posisi gerak belok ke kanan

Ketika *joystick* di gerakan posisi ke kanan, maka robot akan bergerak belok ke kanan. Seperti yang ditunjukkan pada gambar 18 dibawah ini.



Gambar 18: Posisi Robot Bergerak Belok ke Kanan

### 4. Posisi gerak belok ke kiri

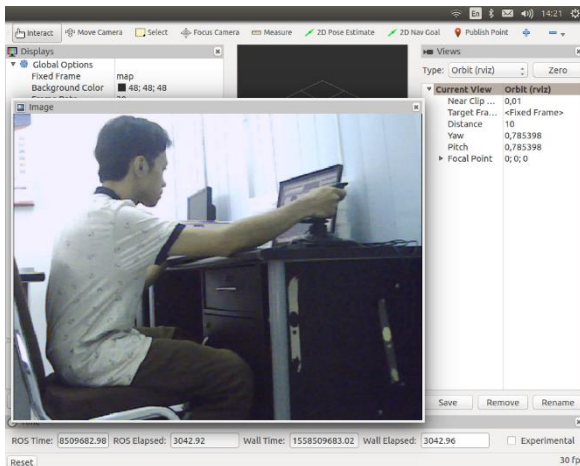
Ketika *joystick* di gerakan posisi ke kiri, maka robot akan bergerak belok ke kiri. Seperti yang ditunjukkan pada gambar 19 dibawah ini.



Gambar 19: Posisi Robot Bergerak Belok ke Kiri

### c. Pengujian Audio-Video

Pengujian ini akan memberikan informasi visual audio-video dari robot kepada server melalui layanan teleoperasi secara otomatis dan real time. Seperti yang ditunjukkan pada gambar 20 dibawah ini.



Gambar 20: Informasi Visual Audio-Video Dari Robot Kepada Server

## 4. PENUTUP

### 5.1. Kesimpulan

Pada penelitian ini dapat dibuktikan bahwa prototipe teleoperasi robot berbasis ROS (*Robot Operating System*) dapat diimplementasikan untuk fungsi eksplorasi jarak jauh. Sistem yang dibangun, yakni sistem pengendalian robot jarak jauh yang terdiri dari robot, yang dilengkapi dengan informasi visual *audio-video* yang diterima *server* dari *client* (robot). Dari hasil percobaan ini, *server* dapat mengendalikan lebih dari satu robot secara bersamaan, dalam waktu yang bersamaan, dan tidak dibatasi oleh jarak selama terkoneksi dengan internet.

### 5.2. Saran

Dalam penggunaan robot pengoprasian secara jarak jauh memiliki banyak manfaat berbasis ROS (*Robot Operating System*), sehingga untuk selanjutnya perlu dikembangkan lebih luas lagi. Dalam pengembangan selanjutnya diharapkan robot yang dibangun dapat melalui beberapa rintangan luar dan dapat menangani berbagai masalah manusia.

## DAFTAR PUSTAKA

- [1] Sayuti Achmad. (2015). *Perancangan Sistem Monitoring Suhu Menggunakan Raspberry Pi Berbasis Web dan Android Pada Ruang Server Universitas Darma Persada*. Jakarta.
- [2] Daniel S Pamungkas. (2013). *Sinyal Elektrik untuk Memperkaya Pengendalian Robot Jarak Jauh*. Batam. Jurnal Rekayasa Elektrika.
- [3] Walid Wisnu Wardhana. (2013). *Sistem Kendali Jarakjauh Mobil Robot Menggunakan Personal Computer*. Bandung.
- [4] MS Hendriyawan A, S.T., M.Eng. (2018). *ROS-based 2-D Mapping Using Non-holonomic Differential Mobile Robot*. Yogyakarta. Jurnal Infotel.
- [5] Moh. Khairudin, Ph.D., Totok Heru T, M.Pd., dan Rustam Asnawi, Ph.D. (2014). *Sistem Kendali Pid Jarak Jauh Robot Manipulator Menggunakan Jaringan Internet Berbasis Matlab*. Yogyakarta. 22 Mei 2019. <http://staffnew.uny.ac.id/upload/132161227/penelitian/sistem-kendali-berbasis-weblaporane.pdf>
- [6] Bima Bisma Baskara. (2014). *Mengenal Lebih Dalam Teknologi TelePresence*. 28 April 2018. Tangerang. Ilmu Teknologi Informasi (ilmuti.org). <https://fdokumen.com/document/mengenal-lebih-dalam-teknologi-telepresence-jepang-hingga-speaker-itulah.html>
- [7] Agung Tullah. *Sistem Kontrol Nirkabel Robot Mobile*. <https://www.slideshare.net/agungajhaib/makalah-sistem-kontrol-nirkabel-robot-mobile>. Diakses Tanggal 29 April 2018, Jam 19.21 WIB.