

ANALISA SISTEM KOMUNIKASI DATA PADA SIMULASI ROBOT SEPAK BOLA BERODA

Damar Dwiyanto¹, Nifty Fath²

1. Teknik Elektro: Universitas Budi Luhur
Jakarta, Indonesia
dwiyantodamar@gmail.com
2. Teknik Elektro: Universitas Budi Luhur
Jakarta, Indonesia
Nifty.fath@budiluhur.ac.id

ABSTRAK

Jaringan internet mampu menghubungkan setiap komputer yang terhubung di dalamnya, bukan hanya berbagi informasi tetapi juga dapat membuat setiap komputer untuk mengerjakan pekerjaan secara bersama-sama, sehingga memungkinkan pengguna untuk mengendalikan komputer lain. Pada robot sepak bola membutuhkan koordinasi yang baik antar robot dengan cara pengambilan keputusan atau kendali yang dilakukan oleh satu robot yang membutuhkan data kondisi lingkungan dari semua robot. Penelitian ini bertujuan untuk melakukan perancangan dan analisa sistem komunikasi data pada tim robot sepak bola. Metode yang digunakan adalah membangun sistem komunikasi antar klien dan server menggunakan protokol secure shell (ssh). Protokol secure shell (ssh) merupakan protokol jaringan yang berada di lapisan aplikasi, yang digunakan untuk mengendalikan komputer, mengirim file, dan lain-lain. Kualitas komunikasi data yang dirancang kemudian diuji menggunakan perangkat lunak wireshark. Adapun parameter-parameter Quality of Service yang akan diukur antara throughput, delay, packet loss, dan jitter yang mengacu pada standar TIPHON. Berdasarkan hasil pengujian sistem komunikasi data yang telah dilakukan dengan menempatkan machine berada dekat dengan access point diperoleh rata-rata throughput sebesar 28.02619 kbps (sangat bagus), rata-rata packet loss sebesar 0% (sangat bagus), delay sebesar 0.28651552697192 ms (sangat bagus), dan jitter sebesar 0.00000814 ms (bagus). Sementara itu pengujian dengan menempatkan machine berjarak 10 meter dengan access point diperoleh rata-rata throughput sebesar 15.44595 kbps (sangat bagus), rata-rata packet loss sebesar 0% (sangat bagus), delay sebesar 0.520748991128 ms (sangat bagus), dan jitter sebesar 0.0000144 ms (bagus). Dari pengujian yang dilakukan menunjukkan bahwa sistem komunikasi yang dibuat sudah sesuai dengan rencana awal dan sistem komunikasi tersebut dapat diandalkan.

Kata kunci: ROS, Gazebo, SSH, Robot Sepak Bola Beroda, QoS

ABSTRACT

The internet network is able to connect every computer connected to it, not only sharing information but also making each computer to do work together, thus allowing users to control other computers. In soccer robots that require good coordination between robots by means of decision making or control carried out by one robot which requires environmental condition data from all robots. This study aims to design and analyze data communication systems on a soccer robot team. The method used is to build a communication system between clients and servers using the secure shell (SSH) protocol. The secure shell (ssh) protocol is a network protocol located at the application layer, which is used to control computers, send files, and so on. The quality of the designed data communication is then tested using the Wireshark software. The Quality of Service parameters that will be measured includes throughput, delay, packet loss, and jitter which refers to the TIPHON standard. Based on the results of testing the data communication system that has been done by placing the machine close to the access point, it is obtained an average throughput of 28.02619 kbps (very good), an average packet loss of 0% (very good), a delay of 0.28651552697192 ms (very good), and a jitter of 0.00000814 ms (good). Meanwhile, testing by placing the machine 10 meters away from the access point obtained an average throughput of 15.44595 kbps (very good), an average packet loss of 0% (very good), a delay of 0.520748991128 ms (very good), and a jitter of 0.0000144 ms (good). From the examiners conducted, it shows that the communication system created is in accordance with the original plan and the communication system is reliable.

Keywords— ROS, Gazebo, SSH, Wheeled Football Robot, QoS

I. PENDAHULUAN

Jaringan internet mampu menghubungkan setiap komputer yang terhubung di dalamnya, bukan hanya berbagi informasi, tetapi juga dapat membuat setiap komputer untuk mengerjakan pekerjaan secara bersama-sama, sehingga memungkinkan pengguna untuk mengendalikan komputer lain. Pada robot sepak bola beroda membutuhkan koordinasi yang baik antar robot dengan cara pengambilan keputusan atau kendali yang dilakukan oleh satu robot yang membutuhkan data kondisi lingkungan dari semua robot. Oleh sebab itu dibutuhkan sistem komunikasi data yang handal. Sistem komunikasi yang akan dirancang yaitu *machine to machine* yang dimana sistem komunikasi ini dapat digunakan pada robot yang membutuhkan koordinasi antar robot tidak hanya dalam robot sepak bola beroda tetapi sistem komunikasi ini dapat digunakan pada *machine to machine*.

Pada penelitian [1] menjelaskan bahwa *middleware* robot adalah perangkat lunak yang digunakan untuk mengontrol sistem robot. *Middleware* berfungsi sebagai penghubung antara perangkat keras sensor, aktuator, dan komponen perangkat lunak. Beberapa *middleware* telah dikembangkan dan digunakan oleh para peneliti untuk mengontrol sistem robotika seperti Player/Stage, Miro, MRDS, ASEBA, Orca, Pyro, ROS, dan ROS2.

Pada penelitian [2] menjelaskan bahwa ROS adalah Sistem Operasi Robot bersifat *open source* yang didalamnya terdapat *library* dan *tools* untuk membuat perangkat lunak robot. ROS merupakan sebuah *middleware* robot yang dapat menghubungkan perangkat keras robot dengan sistem operasi komputer secara fleksibel. ROS bertujuan untuk memudahkan pengembangan robot dalam membuat perangkat lunaknya tanpa harus membuat kode sumber dari awal serta dapat dikembangkan bersama-sama.

Pada penelitian [2] menjelaskan bahwa Gazebo adalah sebuah aplikasi simulasi *open source* yang dapat mengolah perkembangan mesin dinamis yang dikembangkan oleh *Open Source Robotics Foundation*. Gazebo dapat mensimulasikan perancangan robot serta melatih sistem *artificial intelligence* dengan kemampuan simulasi yang akurat, perancangan simulasi dapat didesain baik didalam maupun diluar ruangan yang kompleks. Aplikasi gazebo beroperasi pada sistem operasi Linux dengan kemampuan grafis yang cukup tinggi.

Pada penelitian [3] menjelaskan bahwa SSH (*secure shell*) adalah protokol jaringan yang berada di lapisan aplikasi pada protokol TCP/IP, memfasilitasi sistem komunikasi yang aman diantara dua sistem yang menggunakan arsitektur klien server dengan menyediakan kerahasiaan dan integritas data melalui teknik enkripsi dan dekripsi yang dilakukan secara otomatis di dalam koneksinya, untuk menggunakan SSH dibutuhkan

otentifikasi user berupa kunci umum dan password yang terenkripsi.

Pada penelitian [4] menjelaskan bahwa *Quality of Service* (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat dari satu servis. *Quality of Service* (QoS) digunakan untuk mengukur sekumpulan atribut kinerja yang telah dispesifikasikan dan diasosiasikan dengan suatu servis.

Berdasarkan latar belakang dan studi pustaka yang telah diuraikan, maka dalam penelitian ini dirancang dan menganalisa sistem komunikasi data pada simulasi robot sepak bola beroda berdasarkan standar TIPHON untuk mengetahui apakah sistem komunikasi dapat diandalkan.

II. PERANCANGAN SISTEM

Diuraikan tentang diagram blok sistem, diagram alir sistem komunikasi data, dan juga perancangan sistem komunikasi data.

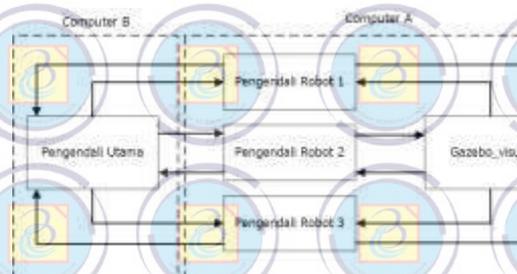
A. Diagram Blok Sistem

Diagram blok sistem komunikasi pada simulasi robot sepak bola beroda ditunjukkan pada Gambar 1 dan Gambar 2.



Gambar 1. Diagram Blok Pengiriman Data Pada PC

Diagram blok pada Gambar 1 menjelaskan bahwa komputer B sebagai *WORKSTATION* melakukan pengiriman data kepada komputer A sebagai *ROS MASTER* dengan menggunakan jaringan internet yang didapat dari *wireless access point* (router).



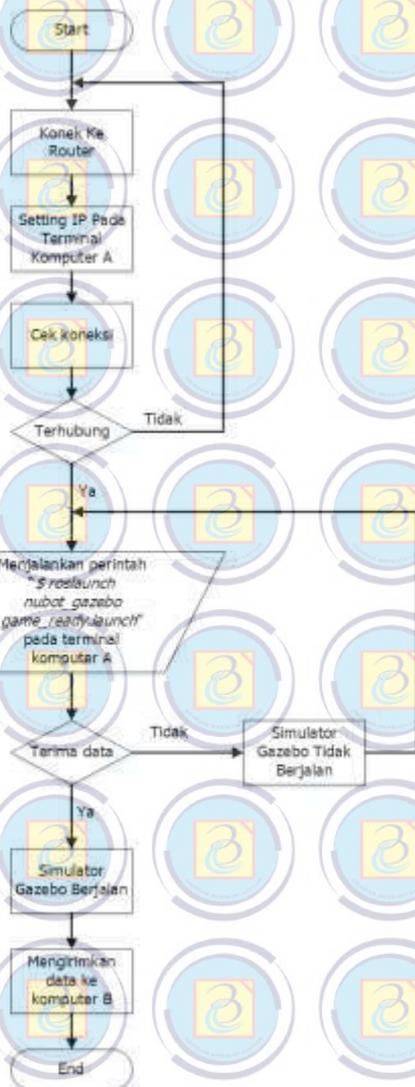
Gambar 2. Diagram Blok Sistem Komunikasi Data

Diagram blok pada Gambar 2 menjelaskan bahwa konfigurasi dilakukan dengan menjalankan simulasi *gazebo_visual* dengan dua komputer yang menjalankan *WORKSTATION* dan *ROS MASTER* secara terpisah. Komputer A sebagai *ROS MASTER* menjalankan *gazebo_visual* untuk menampilkan kondisi lingkungan dan pergerakan robot. Komputer B sebagai *WORKSTATION* menjalankan pengendali

utama untuk menghitung dan mengirim perintah gerakan ke pengendali robot yang berada pada komputer A. Komunikasi yang terjadi antar robot dalam simulasi yaitu *publish* (mengirim) dan *subscribe* (menerima) data. Komputer B atau dalam simulasi sebagai robot kiper yang berperan sebagai pengendali utama yang mempunyai tugas sebagai pembaca data lingkungan dan memberikan perintah kepada robot lain. Komputer A sebagai *ROS MASTER* mempunyai tugas untuk menampilkan gazebo simulator, data visual robot, beserta pergerakannya.

B. Diagram Alir Sistem Komunikasi Data

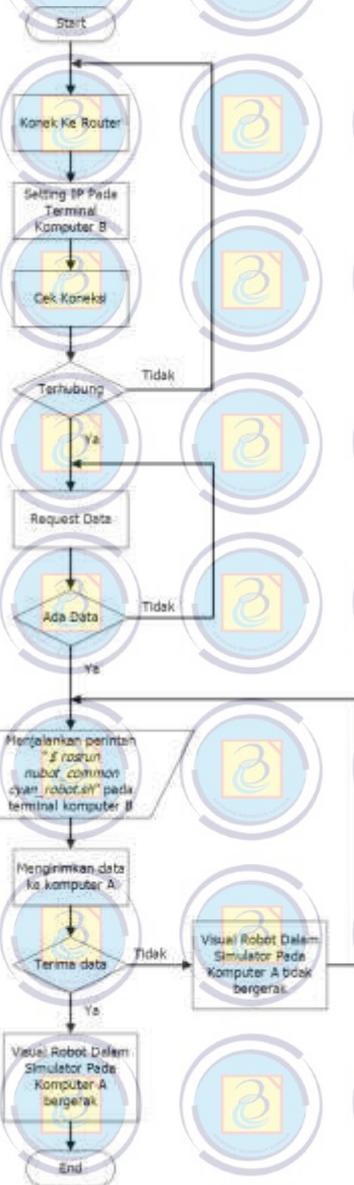
Diagram alir sistem komunikasi pada simulasi sepak bola beroda ditunjukkan pada Gambar 3 dan Gambar 4.



Gambar 3. Diagram Alir Komputer A

Berdasarkan Gambar 3 proses pertama yang dilakukan komputer A adalah mengkoneksi ke *router access point*, setelah dikoneksikan melakukan konfigurasi IP pada terminal yang ada pada komputer kemudian dilakukan pengecekan apakah sudah terhubung. Jika belum terhubung

maka hubungkan kembali ke *router*, jika sudah terhubung kemudian menjalankan perintah “*\$ roslaunch nubot_gazebo game_ready.launch*” pada terminal komputer A untuk menjalankan simulator gazebo. Jika simulator belum berjalan maka menjalankan perintah lagi pada terminal komputer A, jika simulator sudah berjalan maka komputer A akan mengirimkan data ke komputer B.



Gambar 4. Diagram Alir Komputer B

Berdasarkan Gambar 4 proses pertama yang dilakukan komputer B adalah mengkoneksi ke *router access point*, setelah dikoneksikan melakukan konfigurasi IP pada terminal yang ada pada komputer kemudian dilakukan pengecekan apakah sudah terhubung. Jika belum terhubung maka hubungkan kembali ke *router*, jika sudah terhubung kemudian menunggu pengiriman data dari komputer A yang sedang menjalankan perintah “*\$ roslaunch nubot_gazebo game_ready.launch*” untuk menjalankan simulator gazebo. Jika data pengiriman dari komputer A sudah diterima

kemudian menjalankan perintah “\$ *roslaunch nubot_common cyan_robot.sh*” pada terminal komputer B untuk menjalankan pergerakan visual robot pada simulator gazebo. Kemudian mengirimkan data ke komputer A. Jika data belum diterima komputer A maka visual robot dalam simulator pada komputer A tidak bergerak maka menjalankan lagi perintah pada terminal komputer B, jika data sudah diterima komputer A maka visual robot dalam simulator pada komputer A akan bergerak.

C. Wireless Access Point (Router)

Router wireless access point digunakan sebagai penghubung antara komputer A dan komputer B dengan alamat yang dibagikan dari *router wireless access point* kepada masing-masing komputer secara unik. Komputer A dan B harus terkoneksi dengan *access point* agar dapat terkoneksi dalam komunikasi data. Gambar konfigurasi router wireless access point sebagai media pengiriman data dapat dilihat pada Gambar 5.



Gambar 5. Router Wireless Access Point

D. Komunikasi Multiple Machine

Komunikasi *multiple machine* adalah komunikasi yang dilakukan dengan menggunakan lebih dari satu komputer secara bersamaan saat bernavigasi sebuah robot [5]. Komunikasi yang terjadi antar robot dalam simulasi yaitu *publish* (mengirim) dan *subscribe* (menerima) data. Komputer B atau dalam simulasi sebagai robot kiper berperan sebagai pengendali utama yang mempunyai tugas sebagai pembaca data lingkungan dan memberikan perintah kepada robot lain. Komputer A sebagai *ROS MASTER* mempunyai tugas untuk menampilkan gazebo simulator, data visual robot, beserta pergerakannya.

Pada perancangan *multiple machine* dibutuhkan protokol sistem komunikasi yang mampu menyediakan layanan komunikasi antar dua atau lebih titik komputer. *Secure shell* (ssh) merupakan protokol jaringan yang memungkinkan pertukaran data secara aman antara dua komputer. Protokol *secure shell* (ssh) berada pada lapisan aplikasi dalam *osi layer*. Lapisan aplikasi merupakan interaksi antarmuka yang digunakan oleh *host* di jaringan komunikasi yang berisi protokol komunikasi dan metode antarmuka yang digunakan dalam komunikasi klien-server. Untuk membuat sistem bekerja dengan menentukan nama

untuk kedua mesin di file */etc/hosts* pada sistem. Dalam menentukan nama pada file */etc/hosts* mesin harus mempunyai alamat IP untuk berkomunikasi di jaringan LAN dan namanya adalah sebagai berikut:

1) User Authentication

Otentikasi atau juga disebut sebagai identitas pengguna merupakan sarana yang wajib diketahui oleh sistem agar dapat diverifikasi oleh sistem atau server. Dengan mengatur alamat IP pada masing-masing komputer. Kemudian menjadikan komputer A sebagai *ROS MASTER* atau disebut juga sebagai ssh server dan menjadikan komputer B sebagai *WORKSTATION* atau disebut juga sebagai ssh klien. Otentikasi identitas pengguna ditunjukkan pada Gambar 6.



Gambar 6. User Authentication

2) Host Authentication

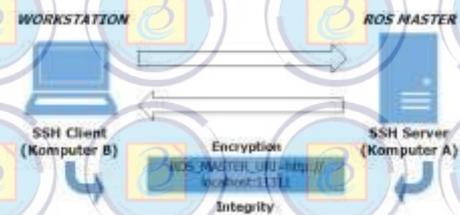
Host digunakan oleh server untuk membuktikan identitasnya kepada klien dan untuk memverifikasi dikenal oleh server. Dengan mengatur *hostname* pada masing-masing komputer. Kemudian menjadikan komputer A sebagai *ROS MASTER* atau disebut juga sebagai ssh server dan menjadikan komputer B sebagai *WORKSTATION* atau disebut juga sebagai ssh klien. Otentikasi host ditunjukkan pada Gambar 7.



Gambar 7. Host Authentication

3) Data Encryption & Integrity

Menjadikan komputer A sebagai server dengan mengatur *ROS_MASTER_URI* pada komputer A “ \$ *export ROS_MASTER_URI = http://localhost:11311*” pada komputer B harus menggunakan host komputer A dengan “ \$ *export ROS_MASTER_URI = http://mastermachine:11311*”. Mengatur *localhost* pada komputer A ditunjukkan pada Gambar 8.



Gambar 8. Set Localhost Pada Komputer A

Konfigurasi *multiple machine* dilakukan dengan cara mengkonfigurasi alamat IP dari masing-

masing komputer [6]. Mengkonfigurasi *ROS MASTER* dan *WORKSTATION* dengan cara menjalankan perintah “\$ *gedit .bashrc*” pada terminal linux [7].

E. Konfigurasi ROS Master Pada Komputer A dan Komputer B

Konfigurasi *ROS MASTER* dilakukan dengan cara menjalankan perintah “\$ *gedit .bashrc*” pada terminal seperti yang ditunjukkan pada Gambar 9.



```
gedit ~/.bashrc
# Enable password authentication
# If you want to put all your ssh keys in a separate file like
# ~/.ssh/authorized_keys, please uncomment these lines:
# Set the path to the authorized_keys file
# if [ -f ~/.ssh/authorized_keys ]; then
#     ssh -i ~/.ssh/authorized_keys
# fi

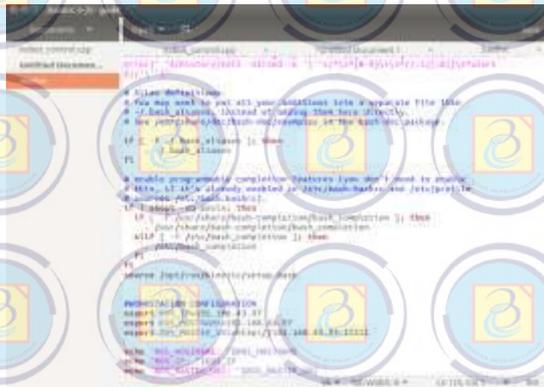
# Enable program completion for bash. You must first
# enable completion for the shell. You can do this by
# running the following command:
# sudo apt-get install bash-completion
# If you have bash-completion installed, you must first
# run the following command:
# source /usr/share/bash-completion/bash_completion

# Enable ROS configuration
# The location of the ROS_IP address for the master node
export ROS_IP=$(hostname -i)
# The IP address for the workstation
export ROS_HOSTNAME=$(hostname -i)

# Enable ROS configuration
# The location of the ROS_IP address for the master node
export ROS_IP=$(hostname -i)
# The IP address for the workstation
export ROS_HOSTNAME=$(hostname -i)
```

Gambar 9. Konfigurasi ROS MASTER Komputer A

Konfigurasi *ROS MASTER* pada komputer A digunakan untuk menentukan komputer A sebagai *ROS MASTER* yang mempunyai tugas untuk menampilkan data visual lingkungan serta pergerakan robot yang ada pada Gazebo. Langkah-langkah dalam melakukan konfigurasi yaitu seperti pada Gambar 10.



```
gedit ~/.bashrc
# Enable password authentication
# If you want to put all your ssh keys in a separate file like
# ~/.ssh/authorized_keys, please uncomment these lines:
# Set the path to the authorized_keys file
# if [ -f ~/.ssh/authorized_keys ]; then
#     ssh -i ~/.ssh/authorized_keys
# fi

# Enable program completion for bash. You must first
# enable completion for the shell. You can do this by
# running the following command:
# sudo apt-get install bash-completion
# If you have bash-completion installed, you must first
# run the following command:
# source /usr/share/bash-completion/bash_completion

# Enable ROS configuration
# The location of the ROS_IP address for the master node
export ROS_IP=$(hostname -i)
# The IP address for the workstation
export ROS_HOSTNAME=$(hostname -i)

# Enable ROS configuration
# The location of the ROS_IP address for the master node
export ROS_IP=$(hostname -i)
# The IP address for the workstation
export ROS_HOSTNAME=$(hostname -i)
```

Gambar 10. Konfigurasi WORKSTATION Komputer B

Konfigurasi *WORKSTATION* pada komputer B digunakan untuk menentukan komputer B sebagai *WORKSTATION* yang mempunyai tugas sebagai pengendali utama untuk mengirimkan perintah data ke komputer A sebagai *ROS MASTER*. Langkah-langkah dalam melakukan konfigurasi yaitu seperti gambar 10.

F. Sistem Menggunakan Simulator Gazebo

Perancangan sistem simulasi dilakukan dengan mempelajari sistem simulasi yang dibuat dan didistribusikan secara *open source* oleh tim Nubot

yang berasal dari National University of Defense Technology [8]. Sistem simulasi terintegrasi oleh Gazebo simulator sebagai kondisi visual robot beserta lingkungan dan jalannya proses pengendalian robot dilaksanakan oleh *Robot Operating System (ROS)*.

Modul simulator ini berfungsi sebagai simulator yang menampilkan hasil dari menjalankan perintah “\$ *roslaunch nubot_gazebo game_ready.launch*” pada terminal. Hasil yang ditampilkan dari *package* ini berupa tampilan *world model* dari kondisi visual robot beserta lingkungan. Visual robot dan lingkungan pada simulator gazebo ditunjukkan pada Gambar 11.



Gambar 11. Tampilan World Model Simulasi Gazebo

III. PENGUJIAN DAN ANALISA

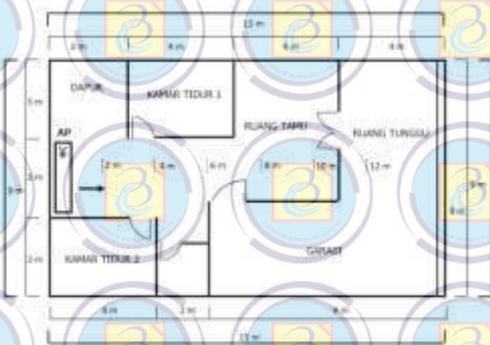
Akan dibahas tentang pengujian berdasarkan perancangan sistem yang telah dibuat. Pengujian ini dilakukan untuk mengetahui kinerja dari sistem dan untuk mengetahui apakah sudah sesuai dengan perencanaan. Pada pengujian menggunakan perangkat lunak wireshark untuk menganalisis jaringan yang digunakan, skenario pengujian yang dilakukan yaitu pengujian kuat sinyal WiFi, pengujian koneksi internet pada *ROS MASTER* dan *WORKSTATION*, pengujian konfigurasi komunikasi *multiple machine*, pengujian simulasi, dan analisa *Quality of Service*.

A. Kuat Sinyal Wifi

Pengujian kuat sinyal dilakukan dengan menghubungkan komputer A (server) dan komputer B (klien) dengan *access point* router. Pengukuran dilakukan didalam ruangan atau *indoor* sebanyak 5 kali pengukuran pada masing-masing komputer dengan rentang jarak pengukuran 0 meter sampai dengan 12 meter. Lokasi pengukuran beserta proses pengukuran disajikan pada Gambar 12 dan Gambar 13.

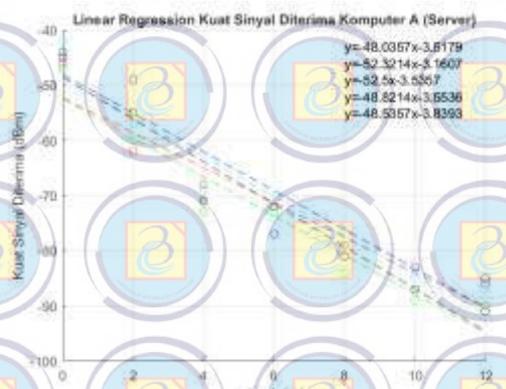


Gambar 12. Proses Pengambilan Data Kuat Sinyal Pada Komputer



Gambar 13. Proses Pengambilan Data Menjauhi Titik Access Point

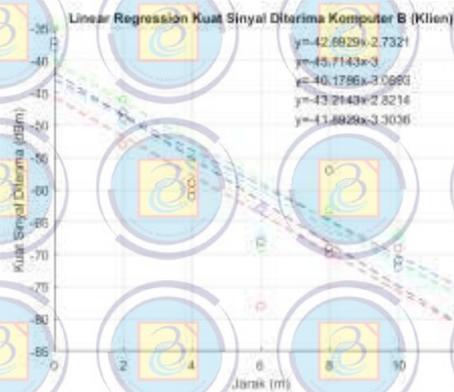
Sesuai dengan data hasil observasi pengukuran kuat sinyal diterima dilakukan dengan cara mengumpulkan data menggunakan *wifi analyzer* pada masing-masing komputer dengan menentukan jarak untuk mendapatkan kuat sinyal sampai tidak ada sinyal yang didapat. Kemudian didapat hasil pengukuran yang akan disajikan dalam bentuk grafik regresi linier. Hasil pengukuran kuat sinyal diterima disajikan pada Gambar 14 dan Gambar 15.



Gambar 14. Grafik Hasil Pengukuran Kuat Sinyal Diterima Komputer A

Gambar 14 menjelaskan grafik regresi linier yang digunakan untuk melakukan prediksi berdasarkan data-data yang telah dimiliki. Pada

grafik regresi linier terdapat titik atau lingkaran kecil yaitu merupakan data yang diperoleh pada saat pengukuran sementara garis putus-putus menunjukkan bahwa terjadi perubahan rata-rata kuat sinyal diterima ketika pengukuran yang dilakukan menjauhi *access point* router. Hasil pengukuran kuat sinyal yang diterima menjelaskan bahwa didapat nilai total rata-rata sebesar -71.17142857 dBm dan mendapat kategori sangat baik dengan rentang nilai 0 sampai -74 dBm yang mengacu berdasarkan *key performance indicator* yang telah ditentukan.

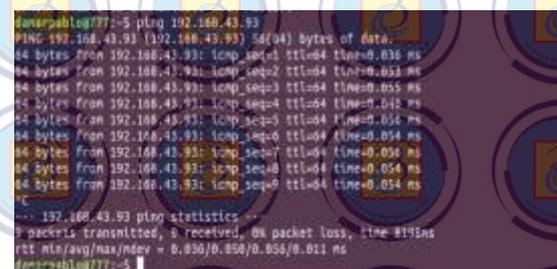


Gambar 15. Grafik Hasil Pengukuran Kuat Sinyal Diterima Komputer B

Gambar 15 menjelaskan grafik regresi linier yang digunakan untuk melakukan prediksi berdasarkan data-data yang telah dimiliki. Pada grafik regresi linier terdapat titik atau lingkaran kecil yaitu merupakan data yang diperoleh pada saat pengukuran sementara garis putus-putus menunjukkan bahwa terjadi perubahan rata-rata kuat sinyal diterima ketika pengukuran yang dilakukan menjauhi *access point* router. Hasil pengukuran kuat sinyal yang diterima menjelaskan bahwa didapat nilai total rata-rata sebesar -60.71428571 dBm dan mendapat kategori sangat baik dengan rentang nilai 0 sampai -74 dBm yang mengacu berdasarkan *key performance indicator* yang telah ditentukan.

B. Pengujian Koneksi Internet ROS Master dan WORKSTATION

Pada pengujian koneksi internet ini dilakukan dengan cara memberikan PING pada *ROS MASTER* agar dapat terlihat sudah saling terkoneksi dan dapat menerima atau mengirimkan data. Berikut ini merupakan gambar pengujian yang ditunjukkan pada Gambar 16 dan Gambar 17.



Gambar 16. Pengujian PING ROS MASTER

daftar topik list yang aktif pada *roscore* saat *ROS MASTER* menjalankan perintah “\$ *roslaunch nubot_gazebo game_ready.launch*”.

D. Pengujian Simulasi

Pengujian simulasi ini dilakukan dengan memberi perintah “\$ *source devel /setup.bash*” untuk menambahkan beberapa variabel lingkungan agar ROS dapat berfungsi, kemudian menjalankan “\$ *roslaunch nubot_gazebo game_ready.launch*” pada komputer A sebagai *ROS MASTER* untuk menampilkan kondisi visual lingkungan serta pergerakan robot pada Gazebo. Setelah Gazebo berhasil menampilkan kondisi lingkungan serta pergerakan robot, lalu menjalankan perintah “\$ *roslaunch nubot_common cyan_robot.sh*” pada komputer B sebagai *WORKSTATION* yang berperan sebagai pengendali utama yang memberikan data perintah pergerakan robot. Pengujian dapat dilihat pada Gambar 20.



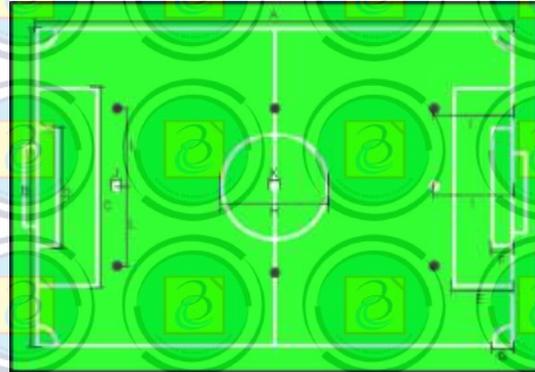
Gambar 20. Tampilan Pengujian Pengiriman Data Simulasi Menggunakan Wireshark

Gambar 20 menjelaskan komunikasi yang terjadi antar robot dalam simulasi yaitu *publish* (mengirim) dan *subscribe* (menerima) data. Komputer B atau dalam simulasi sebagai robot kiper yang berperan sebagai pengendali utama yang mempunyai tugas sebagai pembaca data lingkungan dan memberikan perintah data. Beberapa data yang dikirim berupa *ballsholding* merupakan data perintah robot untuk memegang bola, *velcmd* merupakan data perintah kecepatan robot, *actioncmd* merupakan data perintah pergerakan robot, dan *strategy* merupakan data perintah yang didapat ketika *ballsholding* sudah menjalankan perintahnya kemudian melakukan operan bola yang dilakukan antara robot 2 dan robot 3. Komputer A sebagai *ROS MASTER* mempunyai tugas untuk menampilkan gazebo simulator, data visual robot, beserta pergerakannya. Kemudian seluruh pengiriman data yang terjadi antara komputer A sebagai *ROS MASTER* dan komputer B sebagai *WORKSTATION* akan ter-capture pada software wireshark sebagai perangkat lunak untuk menganalisa komunikasi pengiriman yang terjadi.

E. Analisa Quality of Service

Pada pengujian ini akan dibahas mengenai analisis tentang performansi jaringan *quality of service* yang mencakup pengamatan paket,

pengukuran *throughput*, pengukuran *packet loss*, pengukuran *delay* dan pengukuran *jitter*. Jarak pengujian *analisis quality of service* dilakukan berdasarkan persyaratan ukuran lapangan kontes robot sepakbola Indonesia beroda 2020 [9]. Ukuran lapangan akan disajikan pada Gambar 21 dan Tabel 4.



Gambar 21. Ukuran Lapangan KRSBI

Tabel 2. Ukuran Lapangan (satuan : meter)

A	9	G	0,375
B	6	H	2
C	Lebar gawang + 2,25	I	1,5
D	Lebar gawang + 0,75	J	0,05
E	1,125	K	0,075
F	0,375	L	1,5

F. Pengukuran Throughput

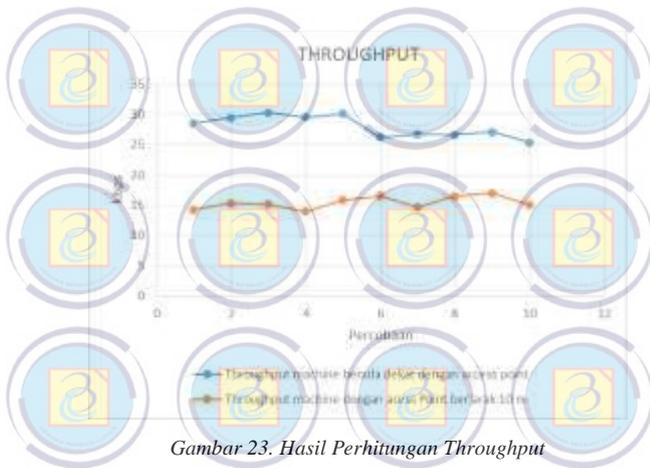
Pengukuran throughput adalah jumlah total kedatangan paket selama pengamatan berlangsung. Aspek utama keberhasilan throughput adalah ketersediaan bandwidth yang cukup dan minim *packet loss*. Dalam pengujian ini dilakukan sebanyak 5 kali percobaan. Menangkap pengukuran *throughput* pada software wireshark dapat dilihat pada Gambar 22.



Gambar 22. Pembacaan Throughput Pada Software Wireshark

Hasil pengujian didapatkan dengan menggunakan persamaan perhitungan throughput:

$$\begin{aligned}
 \text{Throughput} &= \frac{\text{Paket yang diterima}}{\text{Lama pengamatan}} \\
 &= \frac{147072 \text{ Byte}}{41.231 \text{ s}} \\
 &= 3567.024811428294 \\
 &\times 8 \text{ bit} \\
 &= 28536.19849142635 \text{ bps} \\
 &= 28.53619849142647 \text{ kbps}
 \end{aligned}$$



Gambar 23. Hasil Perhitungan Throughput

Gambar 23 hasil perhitungan pengujian *throughput* dari masing-masing percobaan. Garis berwarna biru menandakan nilai *throughput* berdasarkan *machine* berada dekat dengan *access point* sementara garis berwarna jingga menandakan nilai *throughput* berdasarkan *machine* dengan *access point* berjarak 10 m.

G. Pengukuran Packet Loss

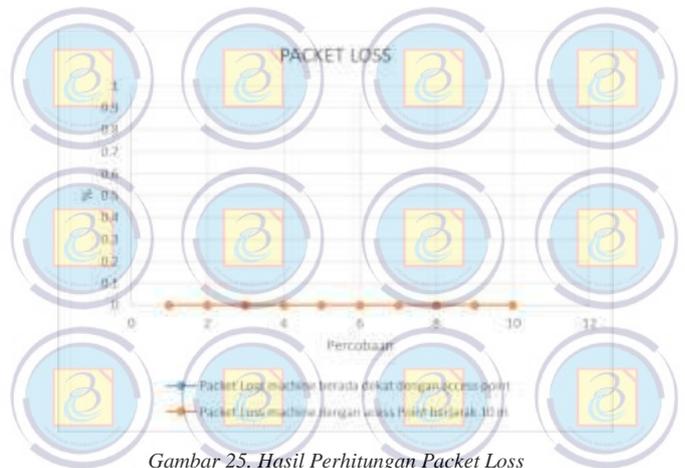
Packet loss merupakan suatu parameter yang menggambarkan jumlah total paket yang hilang, dapat terjadi karena *collision* (tabrakan antar paket-paket yang dikirimkan oleh 2 pengguna secara bersamaan) dan *congestion* (kemacetan transmisi paket akibat padatnya *traffic* yang harus dilayani). Dalam pengujian ini dilakukan sebanyak 5 kali percobaan. Dalam *software* wireshark untuk mencari nilai *packet loss* dengan mem-filter paket dengan menuliskan *tcp.analysis.lost_segment*, akan menangkap *packet loss* dengan *software* wireshark seperti Gambar 24.



Gambar 24. Menangkap Packet Loss Dengan Software Wireshark

Hasil pengujian didapatkan dengan menggunakan persamaan perhitungan *packet loss*:

$$\text{Packet Loss} = \frac{(\text{Paket Data Dikirim} - \text{Paket Data Diterima})}{\text{Paket Data Dikirim}} \times 100\% = \frac{147072 - 147072}{147072} \times 100\% = 0\%$$



Gambar 25. Hasil Perhitungan Packet Loss

Gambar 25 hasil perhitungan pengujian *packet loss* dari masing-masing percobaan. Garis berwarna biru menandakan nilai *packet loss* berdasarkan *machine* berada dekat dengan *access point* sementara garis berwarna jingga menandakan nilai *packet loss* berdasarkan *machine* dengan *access point* berjarak 10 m.

H. Pengukuran Delay

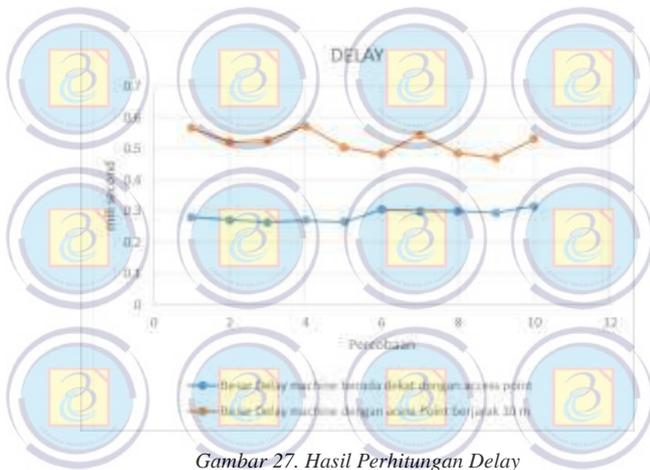
Pada pengukuran *delay* pada sistem komunikasi data simulasi dilakukan dengan menguji pengiriman data dari *WORKSTATION* pada komputer B ke *ROS MASTER* pada komputer A dan mengamati waktu pengiriman data tersebut. Pada Gambar 26 menunjukkan nilai total paket dan waktu saat pengamatan pada *software* wireshark.



Gambar 26. Total Paket dan Waktu Saat Pengamatan

Hasil pengujian didapatkan dengan menggunakan persamaan perhitungan *delay*:

$$\text{Delay} = \frac{\text{Total Waktu}}{\text{Total Paket Diterima}} = \frac{41.231 \text{ s}}{147072} = 0.00028034568103 \text{ sec} = 0.28034568103 \text{ ms}$$



Gambar 27. Hasil Perhitungan Delay

Gambar 27 hasil perhitungan *delay* dari masing-masing percobaan. Garis berwarna biru menandakan nilai *delay* berdasarkan *machine* berada dekat dengan *access point* sementara garis berwarna jingga menandakan nilai *delay* berdasarkan *machine* dengan *access point* berjarak 10 m.

I. Pengukuran Jitter

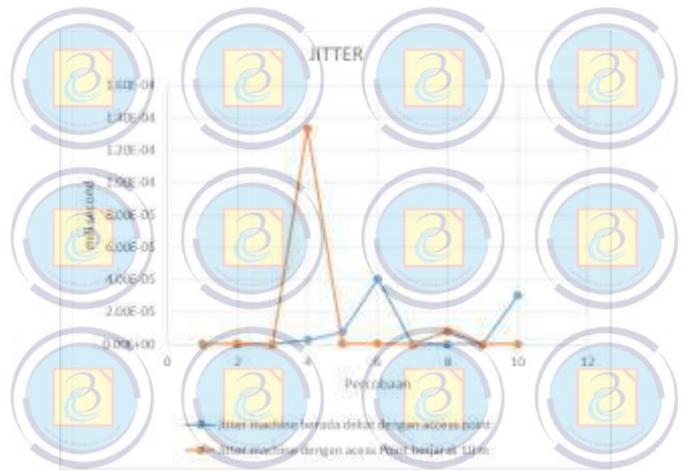
Pengukuran *jitter* didapatkan dari total variasi *delay* dibagi dengan total paket diterima. Untuk melakukan perhitungan dengan melakukan *export packet dissections* ke dalam file csv. Gambar 28 menunjukkan perhitungan *jitter*.

41.22854513	41.228890	0.238E-05	3.88E-05	8.298E-05	2.234E-05
41.22854892	41.228845	8.894E-05	4.88E-05	3.8804E-05	4.2889E-05
41.22855318	41.228849	2.6334E-05	0.000945	4.6054E-05	0.00088835
41.22855977	41.228855	0.00094553	1.99E-05	0.00094551	0.00094562
41.22856153	41.22885	1.1704E-05	1.58E-05	0.11764E-05	3.079E-05
41.22856997	41.22892	1.1439E-05	1.32E-05	0.5459E-05	-2.313E-05
41.22858410	41.228937	1.3336E-05	7.79E-05	1.3226E-05	-5.487E-05
41.22858794	41.228958	7.778E-05	3.21E-05	7.750E-05	-5.3E-05
41.22859215	41.228988	7.305E-05	4.95E-05	7.309E-05	1.0378E-05
41.22859473	41.228995	4.8587E-05	1.46E-05	4.8587E-05	-1.1979E-05
41.22859834	41.22902	3.4632E-05	0.001738	1.4612E-05	0.00169325
41.22859821	41.229079	0.00123864	1.27E-05	0.00177864	-0.00170667
41.22860946	41.231404	2.2347E-05	1.90E-05	2.2367E-05	-2.012E-05
41.23109809	41.231626	1.8634E-05	0.83E-05	1.9034E-05	1.2804E-05
41.23109207	41.231436	0.822E-05			
total delay		41.23145283		total variasi delay	1.8007E-05

Gambar 28. Perhitungan Jitter

Hasil pengujian didapatkan dengan menggunakan persamaan perhitungan *jitter*:

$$\begin{aligned}
 \text{Jitter} &= \frac{\text{Total Variasi Delay}}{\text{Total Paket Diterima} - 1} \\
 &= \frac{1.8027 \times 10^{-5}}{14702 - 1} \\
 &= 1.22573 \times 10^{-10} \text{ sec} \\
 &= 1.22573 \times 10^{-7} \text{ ms}
 \end{aligned}$$

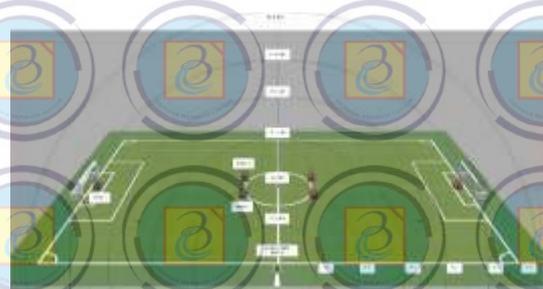


Gambar 29. Hasil Perhitungan Jitter

Gambar 29 hasil perhitungan pengujian *jitter* dari masing-masing percobaan. Garis berwarna biru menandakan nilai *jitter* berdasarkan *machine* berada dekat dengan *access point* sementara garis berwarna jingga menandakan nilai *jitter* berdasarkan *machine* dengan *access point* berjarak 10 m.

J. Radius Kuat Sinyal Access Point

Setelah dilakukan pengujian kuat sinyal beserta perhitungan regresi linier yang telah dilakukan untuk mengetahui prediksi kuat sinyal berdasarkan data-data yang telah dimiliki didapat gambaran berapa kuat sinyal pada titik terjauh dari *access point* pada lapangan yang ditunjukkan pada Gambar 30.



Gambar 30. Radius Kuat Sinyal Access Point

Gambar 30 menjelaskan bahwa *access point* berada pada sisi luar tengah lapangan yang dapat menjangkau seluruh luas lapangan berdasarkan ukuran lapangan Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda. Titik terjauh lapangan yang dapat dijangkau oleh *access point* yaitu sejauh 7.5 meter dengan kuat sinyal sebesar -60 dBm sampai dengan -80 dBm berdasarkan hasil pengujian regresi linier pada kuat sinyal dan mendapat kategori bagus berdasarkan tabel rentang nilai RSRP.

IV. KESIMPULAN

Berdasarkan hasil percobaan dan pengujian yang telah dilakukan, didapat beberapa hal yang perlu dicatat dan diambil sebagai kesimpulan untuk menjawab tujuan dari penelitian. Hal-hal itu antara lain adalah sebagai berikut:

1. Dari pengujian perancangan komunikasi *multiple machine* yang menggunakan protokol *secure shell* (ssh) berhasil terkoneksi dan bekerja sangat baik. Dapat dikatakan andal dalam mengelola pertukaran data dengan menggunakan model jaringan klien-server.
2. Dari pengujian simulasi yang dilakukan semua pengiriman data yang terjadi antara komputer A sebagai *ROS MASTER* dan komputer B sebagai *WORKSTATION* berfungsi dengan baik dan dapat mengirimkan data ke tujuan.
3. Dari pengujian kuat sinyal yang dilakukan bahwa *access point* berada pada sisi luar tengah lapangan yang dapat menjangkau seluruh luas lapangan berdasarkan ukuran lapangan Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda. Titik terjauh lapangan yang dapat dijangkau oleh *access point* yaitu sejauh 7.5 meter dengan kuat sinyal sebesar -60 dBm sampai dengan -80 dBm berdasarkan hasil pengujian regresi linier pada kuat sinyal dan mendapat kategori bagus.
4. Berdasarkan hasil pengujian *Quality of Service* (QoS) yang telah dilakukan didapat hasil pengujian sebagai berikut, pada pengujian dengan menempatkan *machine* berada dekat dengan *access point* didapat hasil pengujian dari masing-masing parameter diperoleh rata-rata *throughput* sebesar 28.02619 kbps dan mendapat kategori (**sangat bagus**), pengujian dari parameter *packet loss* diperoleh rata-rata *packet loss* sebesar 0% ini menandakan tidak ada paket yang hilang selama pengiriman berlangsung dan mendapat kategori (**sangat bagus**), pengujian dari parameter *delay* diperoleh rata-rata *delay* sebesar 0.28651552697192 ms dan mendapat kategori (**sangat bagus**), dari parameter *jitter* diperoleh rata-rata *jitter* sebesar 0.00000814 ms dan mendapat kategori (**bagus**). Sementara itu pada pengujian dengan menempatkan *machine* dengan *access point* berjarak 10 meter didapat hasil pengujian dari masing-masing parameter diperoleh rata-rata *throughput* sebesar 15.44595 kbps dan mendapat kategori (**sangat bagus**), pengujian dari parameter *packet loss* diperoleh rata-rata *packet loss* sebesar 0% ini menandakan tidak ada paket yang hilang selama pengiriman berlangsung dan mendapat kategori (**sangat bagus**), pengujian dari parameter *delay* diperoleh rata-rata *delay* sebesar 0.520748991128 ms dan mendapat kategori (**sangat bagus**), dari parameter *jitter* diperoleh rata-rata *jitter* sebesar 0.0000144 ms dan mendapat kategori (**bagus**). Dari pengujian yang dilakukan menunjukkan bahwa sistem komunikasi yang dibuat sudah sesuai dengan rencana awal dan sistem komunikasi tersebut dapat diandalkan.
5. Hasil perbandingan pengujian dengan menempatkan *machine* berada dekat dengan *access point* dan menempatkan *machine* dengan *access point* berjarak 10 meter menunjukkan terjadinya penurunan yang tidak terlalu signifikan.

V. REFERENSI

- [1] A. Jalil, "Pemanfaatan Middleware Robot Operating System (Ros) Dalam Menjawab Tantangan Revolusi Industri 4.0," *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 45–52, 2019.
- [2] A. Jalil, "Robot Operating System (Ros) Dan Gazebo Sebagai Media Pembelajaran Robot Interaktif," *Ilk. J. Ilm.*, vol. 10, no. 3, pp. 284–289, 2018.
- [3] H. Jusuf, "Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online," *Bina Insa. Ict J.*, vol. 2, no. 2, pp. 75–84, 2015.
- [4] R. Wulandari, "Analisis Qos (Quality Of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – LIPI)," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016.
- [5] R. L. Guimarães, A. S. de Oliveira, J. A. Fabro, T. Becker, and V. A. Brenner, "ROS navigation: Concepts and tutorial," *Stud. Comput. Intell.*, vol. 625, pp. 121–160, Feb. 2016.
- [6] ROS, "ROS/Tutorials/MultipleMachines - ROS Wiki," *wiki.ros.org*, 2018. [Online]. Available: <http://wiki.ros.org/ROS/Tutorials/MultipleMachines>. [Accessed: 03-Jul-2020].
- [7] ROS, "ROS/NetworkSetup - ROS Wiki," *wiki.ros.org*, 2018. [Online]. Available: <http://wiki.ros.org/ROS/NetworkSetup>. [Accessed: 03-Jul-2020].
- [8] W. Yao, W. Dai, J. Xiao, H. Lu, and Z. Zheng, "A Simulation System Based On Ros And Gazebo For Robocup Middle Size League," in *Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on*, 2015, pp. 54–59.
- [9] Kementerian Pendidikan dan Kebudayaan, *Petunjuk Pelaksanaan Kontes Robot Indonesia (KRI)*. 2020.

