

SISTEM KENDALI PERANGKAT ELEKTRONIK JARAK JAUH BERBASIS JARINGAN NIRKABEL MENGGUNAKAN *SECURE SHELL* (SSH) DAN *ROBOT OPERATING SYSTEM* (ROS)

Abdul Jalil^{*1}

¹Sistem Komputer, STMIK Handayani Makassar

Email: ¹abdul.jalil.fw@gmail.com

*Penulis Korespondensi

(Naskah masuk: 19 November 2020, diterima untuk diterbitkan: 26 November 2020)

Abstrak

Salah satu tantangan di era revolusi industri 4.0 adalah pengembangan sistem kontrol secara jarak jauh menggunakan koneksi jaringan nirkabel. Tujuan penelitian ini adalah membangun sistem kontrol perangkat elektronik jarak jauh dengan memanfaatkan jaringan *wireless tethering* pada *smartphone* menggunakan topologi *Wireless Local Area Network* (WLAN) dan *Robot Operating System* (ROS) sebagai perangkat lunak kontrol. Pemanfaatan *wireless tethering smartphone* untuk berbagi koneksi internet dapat dimanfaatkan untuk mengontrol perangkat elektronik yang terkoneksi ke Raspberry Pi. Koneksi jaringan *wireless tethering* memiliki arsitektur jaringan yang cukup sederhana jika dibandingkan dengan arsitektur pengontrolan jarak jauh lainnya, serta memiliki jarak jangkauan koneksi yang cukup jauh dibandingkan dengan koneksi Bluetooth. Metodologi yang digunakan untuk mengontrol perangkat elektronik pada penelitian ini adalah menggunakan *Remote Access Control* (RAC) berbasis protokol SSH. Pemanfaatan *wireless tethering* dan aplikasi *mobile SSH* dapat digunakan untuk mengirim perintah ROS message dari *smartphone* ke Raspberry Pi untuk mengontrol pin GPIO Raspberry Pi agar aktif *high* atau aktif *low*. Pada saat ROS message mengirim perintah ke GPIO untuk aktif *high*, maka sistem akan memberikan instruksi kepada relay untuk menyalakan perangkat elektronik. Selanjutnya pada saat GPIO menerima perintah untuk aktif *low*, maka sistem akan memberikan instruksi kepada relay untuk mematikan perangkat elektronik. Hasil penelitian ini adalah *smartphone* android dapat digunakan untuk mengontrol perangkat elektronik seperti lampu, kipas angin, pemanas ruangan, dan *air conditioner* secara jarak jauh menggunakan jaringan WLAN berdasarkan perintah dari ROS message. Perangkat elektronik pada penelitian ini dapat di kontrol secara efektif pada jarak 20 meter di dalam ruangan dan 40 meter di area bebas hambatan.

Kata kunci: Kendali jarak jauh, *wireless tethering*, RAC, SSH, ROS.

ELECTRONIC DEVICES CONTROL SYSTEM BY REMOTELY BASED ON WIRELESS NETWORK USED SECURE SHELL (SSH) AND ROBOT OPERATING SYSTEM (ROS)

Abstract

One of the challenges in the Industrial Revolution 4.0 is the development of control systems by remotely using a wireless network connection. This study aims to build a control system for controlling the electronic devices by remotely with the utilization of wireless tethering network in the smartphone used Wireless Local Area Network (WLAN) topology and Robot Operating System (ROS) as software for the controller. Utilization of wireless tethering in the smartphone for share the internet connection can be used for control the electronic devices that connected to the Raspberry Pi. The connection of wireless tethering has a simple architecture when compared with the other architecture of the control system by remotely, it then has a long-range connection when compared to the Bluetooth connection. The methodology has used to manage the electronic devices in this study is used Remote Access Control (RAC) based on SSH protocol. The utilization of wireless tethering and mobile SSH can be used to sends ROS message command from smartphone to the Raspberry Pi to control the Raspberry Pi GPIO pin to active high or active low. When android smartphone send ROS message command to the Raspberry Pi to make the GPIO to active high, the system will instruct the relay to turn on the electronic devices. Then when GPIO accepts the instruction to active low, the system will instruct the relay to turn off the electronic devices. The result of this study is that android smartphone can be used to control the electronic devices such as a lamp, fan, heater, and air conditioner by remotely used WLAN network and command from ROS message. The electronic devices on this study can be controlled by effectively with the distance of 20 meters in the rooms and 40 meters at the outside area.

Keywords: Control by remotely, *wireless tethering*, RAC, SSH, ROS.

1. PENDAHULUAN

Pengontrolan perangkat elektronik secara jarak jauh menggunakan koneksi jaringan nirkabel telah menjadi perhatian utama dalam pengembangan sistem kontrol di era revolusi industri 4.0. Penggunaan media koneksi Bluetooth, jaringan WiFi, *mobile phone*, Internet, website, *Cloud Computing*, Web Server, dan *Internet of Things* (IoT) telah diaplikasikan untuk pengontrolan perangkat elektronik secara jarak jauh. Penggunaan Bluetooth sebagai *remote* kontrol telah diaplikasikan untuk mengontrol perangkat elektronik menggunakan Android dan Arduino (Rahmiati et al., 2014.), sistem tersebut dapat bekerja dengan optimal untuk mengendalikan perangkat elektronik pada jarak 320 cm. Perancangan sistem kendali lampu gedung jarak jauh telah diaplikasikan menggunakan android dan jaringan WiFi (Yuliana et al., 2018), pada penelitian tersebut peneliti telah menggunakan Android Studio dan mikrokontroler ATmega328 untuk mengendalikan lampu dan MySQL untuk penyimpanan informasi data pengontrolan lampu.

Rancang bangun prototype sistem kontrol peralatan elektronik jarak jauh telah diaplikasikan menggunakan ponsel Android (Bahri et al., 2015) dan Arduino (Kurniadi et al., 2018), pada penelitian tersebut perangkat elektronik dikendalikan dari jarak jauh menggunakan ponsel android yang terhubung pada jaringan internet. Perancangan dan penerapan sistem kontrol peralatan elektronik jarak jauh telah diaplikasikan menggunakan website (Alamsyah et al., 2015), pada penelitian tersebut perangkat elektronik yang di kontrol terhubung ke port parallel komputer kemudian dapat dikendalikan menggunakan teknologi website yang terhubung ke internet.

Pemanfaatan *cloud computing* sebagai sistem kendali jarak jauh telah diaplikasikan untuk mengontrol perangkat elektronik (Kurniadi et al., 2017), sistem pengontrolan tersebut dapat bekerja otomatis untuk menyalakan dan mematikan perangkat elektronik secara jarak jauh melalui jaringan internet. Pengaplikasian Web Server untuk mengontrol lampu jarak jauh telah diaplikasikan menggunakan Raspberry Pi (Prihatmoko, 2017), penelitian ini telah memanfaatkan Raspberry Pi untuk mengontrol jadwal *on* dan *off* lampu menggunakan Web Server. Penggunaan teknologi *Internet of Things* (IoT) untuk mengendalikan lampu jarak jauh menggunakan Raspberry Pi 3 telah diaplikasikan (Muzawi et al., 2018), sistem ini telah menerapkan pengendalian jaringan lokal melalui web yang disematkan pada perangkat *remote control*.

Perkembangan teknologi pengontrolan dan robotika telah menjadikan *Robot Operating System* (ROS) dan Raspberry Pi sebagai perangkat lunak dan perangkat keras yang digunakan untuk mengontrol robot. Peneliti telah memanfaatkan ROS dan Raspberry Pi untuk mengontrol peralatan elektronik secara manual melalui Raspberry Pi (Jalil, 2019),

pada penelitian tersebut perangkat elektronik di kontrol berdasarkan perintah ROS yang di kirim langsung melalui terminal Linux yang dioperasikan menggunakan Raspberry Pi.

Pada penelitian ini, peneliti akan mengembangkan penelitian untuk mengontrol perangkat elektronik rumah tangga secara jarak jauh dengan memanfaatkan jaringan *wireless tethering* yang ada pada *smartphone* android. Perangkat elektronik tersebut dapat di kontrol secara jarak jauh melalui perintah ROS *message* yang di kirim dari *smartphone* android ke Raspberry Pi menggunakan protokol SSH. Pemanfaatan *wireless tethering* sebagai tools untuk membagi koneksi internet dapat dimanfaatkan dan dikembangkan untuk mengontrol perangkat elektronik secara jarak jauh. Arsitektur jaringan WLAN pada koneksi *wireless tethering* memiliki arsitektur cukup sederhana jika dibandingkan dengan arsitektur pengontrolan jarak jauh lainnya seperti WiFi *access point*, *Internet of Things*, *Cloud Computing*, Internet, Web Server, dan Website. Selain itu, *wireless tethering* memiliki jarak jangkauan koneksi yang cukup jauh jika dibandingkan dengan Bluetooth yaitu *wireless tethering* memiliki jarak jangkauan koneksi hingga 40 meter (Kong et al., 2012) sedangkan Bluetooth hingga 20 meter (Hasan et al., 2018).

Penggunaan ROS sebagai perangkat lunak kontrol merupakan pengembangan teknologi baru dalam dunia pengontrolan, dimana ROS bukan hanya dapat digunakan untuk mengontrol perangkat keras Robot tetapi juga dapat dimanfaatkan untuk mengontrol perangkat elektronik rumah tangga. Sistem pengontrolan yang dibangun diharapkan dapat memberikan suatu hal baru dalam pengontrolan perangkat keras dengan memanfaatkan ROS sebagai perangkat lunak kontrol dan dapat diaplikasikan dalam dunia Revolusi Industri 4.0.

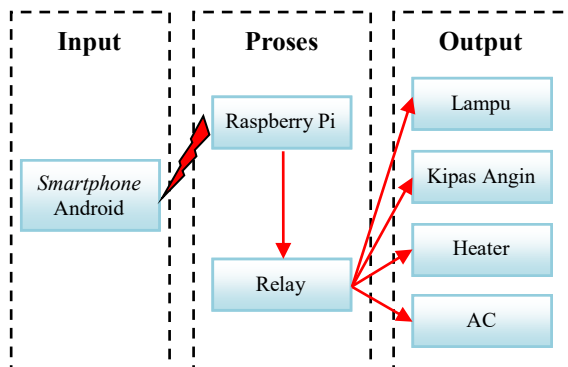
2. METODE PENELITIAN

Metode yang digunakan untuk mengontrol perangkat elektronik secara jarak jauh pada penelitian ini adalah *Remote Access Control* (RAC) berbasis protokol SSH. RAC adalah teknik pengontrolan yang memanfaatkan *Remote Access* untuk menghubungkan dua *resource* yang berbeda atau lebih dalam suatu jaringan. *Secure Shell* (SSH) merupakan protokol jaringan yang berada pada lapisan TCP/IP yang dapat menghubungkan dua sistem secara aman pada arsitektur *client-server*, SSH menyediakan kerahasiaan dan integritas data melalui teknik enkripsi dan dekripsi serta dibutuhkan autentikasi berupa *user* dan *password* yang terenkripsi pada saat ingin melakukan koneksi (Jusuf, H., 2015).

Arsitektur jaringan WLAN untuk menghubungkan antara *smartphone* android ke Raspberry Pi menggunakan *wireless tethering* memiliki arsitektur yang sangat sederhana, dimana untuk mengkoneksikan Raspberry Pi ke *smartphone*

android hanya dengan memanfaatkan fasilitas *tools* yang ada pada *wireless tethering smartphone*. WLAN merupakan jenis jaringan nirkabel berbasis lokal yang memanfaatkan sinyal gelombang elektromagnetik untuk mengirim dan menerima data (Sharon et al., 2014). ROS merupakan perangkat lunak kontrol yang dapat beroperasi dalam sebuah koneksi jaringan WLAN, sehingga perangkat lunak kontrol ini akan sangat efektif jika dikembangkan untuk mengontrol perangkat elektronik berbasis jaringan nirkabel.

Pemanfaatan *smartphone* android dan aplikasi mobile SSH dapat digunakan untuk mengirim perintah ROS message dari *smartphone* android ke Raspberry Pi. Aplikasi mobile SSH telah menggunakan protokol SSH untuk menghubungkan antara perangkat *smartphone* ke Raspberry Pi melalui jaringan *wireless tethering* sehingga perangkat tersebut dapat saling berkomunikasi. Berikut adalah gambar arsitektur sistem yang dibangun pada penelitian ini.



Gambar 1. Arsitektur sistem

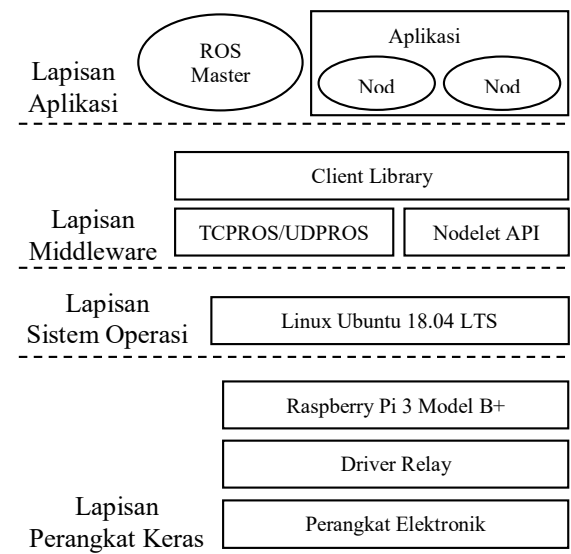
Berdasarkan gambar 1 dapat dilihat bahwa sistem yang dibangun terdiri dari tiga bagian yaitu *input*, *proses*, dan *output*. *Input* pada sistem ini yaitu *input command* atau perintah dari ROS message yang dikirim dari *smartphone* android ke Raspberry Pi. *Smartphone* android dapat terhubung ke Raspberry Pi dengan memanfaatkan jaringan *wireless tethering* yang ada pada *smartphone* android, kemudian menggunakan aplikasi *mobile SSH* untuk mengirim perintah ROS message dari *smartphone* android ke Raspberry Pi.

Media proses pada sistem ini menggunakan Raspberry Pi 3 model B+ dan sebuah *driver relay*. Fungsi Raspberry Pi pada sistem ini adalah sebagai pusat untuk mengolah data proses berupa perintah ROS message yang dikirim dari *smartphone* android kemudian memberikan instruksi ke GPIO pin Raspberry Pi untuk aktif *high* (1) atau aktif *low* (0). Raspberry Pi merupakan komputer berukuran kecil seperti kartu kredit yang dapat digunakan untuk mengolah data, pemrograman, dan pengontrolan (Shadiq et al., 2014). Pada saat ROS message memberikan instruksi ke GPIO pin untuk aktif *high* maka sistem akan memerintahkan relay untuk *on* dan menyalakan perangkat elektronik, dan pada saat

GPIO pin aktif *low* maka sistem akan memerintahkan relay untuk *off* dan mematikan perangkat elektronik.

Output dari sistem ini adalah pengontrolan empat buah perangkat elektronik bertegangan 110V/220V yaitu lampu, kipas angin, pemanas ruangan (*heater*), dan *air conditioner* (AC). Perangkat elektronik akan menyala pada saat *driver relay* dalam kondisi *on*, sehingga arus listrik tegangan 110V/220VAC akan mengalir dari sumber listrik ke perangkat elektronik. Relay merupakan saklar elektronik yang dapat mengalirkan arus listrik kecil dan listrik bertegangan tinggi menggunakan prinsip elektromagnetik (Saleh et al., 2017).

Perangkat lunak yang digunakan untuk mengontrol perangkat keras adalah *meta-operating system* ROS Melodic Morenia yang dapat beroperasi di atas sistem operasi Linux Ubuntu 18.04 LTS. ROS merupakan sistem operasi robot bersifat *open source* yang di dalamnya terdapat *tools* dan *library* yang digunakan untuk mengontrol perangkat keras robotika (Jalil., 2018). Pada penelitian ini ROS dimanfaatkan untuk mengontrol perangkat elektronik sehingga pemanfaatan ROS bukan hanya untuk robot saja tetapi dapat digunakan untuk mengontrol perangkat elektronik rumah tangga atau industri. Berikut adalah gambar lapisan arsitektur perangkat keras dan perangkat lunak yang digunakan pada penelitian ini



Gambar 2. Arsitektur perangkat keras dan perangkat lunak

Pada gambar 2 dapat dilihat bahwa terdapat empat lapisan perangkat lunak dan perangkat keras yang digunakan yaitu lapisan aplikasi, lapisan *middleware*, lapisan sistem operasi, dan lapisan perangkat keras. Lapisan aplikasi adalah lapisan dimana aplikasi ROS bekerja, ROS *master* berfungsi untuk meregistrasi node yang beroperasi kemudian ROS node merupakan tempat untuk membuat perangkat lunak kontrol menggunakan bahasa pemrograman Python atau C++. Dalam node terdapat *topic* dan *message*, dimana *topic* adalah nama unik yang digunakan untuk menghubungkan beberapa

node agar saling terkoneksi dan *message* adalah paket data yang dikirim antar *node* melalui *topic*.

Lapisan *middleware* adalah inti dari sistem pengoperasian ROS, dimana pada lapisan *middleware* terdapat TCPROS/UDPROS yang merupakan protokol yang digunakan untuk mengkomunikasikan data pada ROS (Maruyama et al., 2016). Pada bagian *client library* berfungsi untuk meng-*compile* dan menjalankan perangkat lunak yang dibangun pada ROS yaitu *roscpp* untuk C++ dan *rospy* untuk Python, sedangkan *Nodelet API* merupakan layanan yang digunakan untuk menjalankan perintah-perintah ROS.

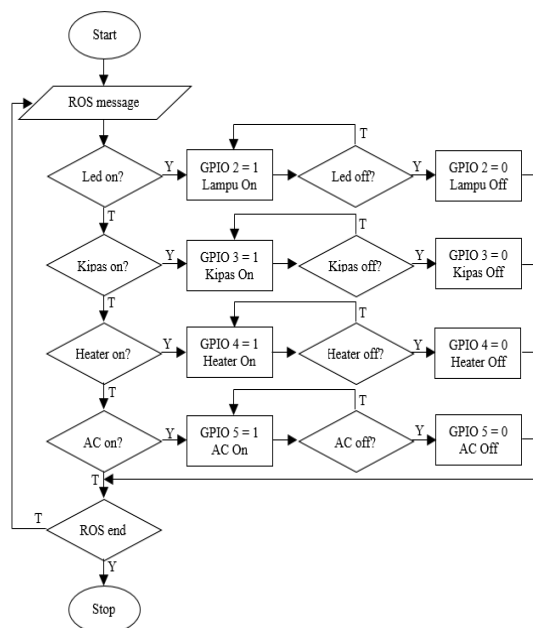
Lapisan sistem operasi merupakan lapisan perangkat lunak sistem yang berfungsi untuk menghubungkan antara perangkat lunak ROS dengan perangkat keras. Sistem operasi yang digunakan untuk mengoperasikan ROS melodic dan menjalankan Raspberry Pi adalah Linux Ubuntu 18.04 LTS. Pada lapisan perangkat keras terdapat Raspberry Pi yang berfungsi sebagai pusat pemrosesan data dan terdapat GPIO pin yang dapat aktif *high* (1) atau aktif *low* (0). Relay pada lapisan ini berfungsi untuk mengalirkan arus AC dari sumber tegangan ke perangkat elektronik, dimana jika GPIO pin menerima instruksi untuk aktif *high* maka relay akan *on* dan menyalakan perangkat elektronik, kemudian jika GPIO pin aktif *low* maka sistem akan memerintahkan relay untuk *off* dan mematikan perangkat elektronik. Berikut adalah *flowchart* sistem yang digunakan untuk mengontrol perangkat elektronik rumah tangga.

3. HASIL DAN PEMBAHASAN

Pengontrolan perangkat elektronik jarak jauh menggunakan protokol SSH dan ROS telah diaplikasikan pada penelitian ini. Perintah ROS dapat dikirim dari *smartphone* android ke Raspberry Pi secara *wireless* menggunakan koneksi protokol SSH. Pada ROS terdapat *Workspace*, *Package*, *Node*, *Topic*, dan *Message*. Fungsi *Workspace* adalah *folder* atau direktori untuk menyimpan dan menjalankan *package* secara lokal atau global. Pada *package* terdapat *source code* program, *tools*, dan *library* yang digunakan untuk mengontrol perangkat keras.

Node merupakan bagian dari *package* yang dapat berjalan pada saat dieksekusi oleh perintah *roslaunch*. Pada *node* terdapat *topic* yang berfungsi untuk menghubungkan antara satu *node* dengan *node* lainnya, dimana ROS *node* dapat saling berkomunikasi menggunakan protokol TCPROS/UDPROS.

ROS *node* dapat saling berkomunikasi dan bertukar data *message* melalui *topic*, dimana *message* merupakan paket data ROS dalam bentuk String, Integer, Float, Image, dan lain sebagainya. Berikut adalah tabel nama *Workspace*, *Package*, *Node*, *Topic*, dan *Message* yang digunakan pada penelitian ini.



Gambar 3. Flowchart sistem pengontrolan perangkat elektronik

Tabel 1. Nama *Workspace*, *Package*, *Node*, *Topic*, dan *Message*

Workspace	Package	Node	Topic	Message
project_ws	my_package	led_on	led	"led_on"
		led_off		"led_off"
		kendali_led	kipas	"kipas_on"
		kipas_on		"kipas_off"
		kipas_off	heater	"heater_on"
		kendali_kipas		"heater_off"
		heater_on	ac	"ac_on"
		heater_off		"ac_off"
		kendali_heater	kendali_ac	
		ac_on		
		ac_off		
		kendali_ac		

Sistem komunikasi data pada ROS adalah menggunakan teknik *publisher* dan *subscriber*. *Publisher* pada sistem ini adalah *node* yang dijalankan pada *smartphone* android untuk mengirim message ke *subscriber* pada Raspberry Pi. *Node* pada *publisher* dapat terhubung ke *node subscriber* melalui *topic*, dimana data yang dikirim melalui *topic* adalah berupa data *message* dengan tipe data string. Berikut adalah tabel komunikasi *publisher* dan *subscriber* ROS pada penelitian ini.

Langkah awal yang dilakukan untuk menjalankan sistem ini adalah dengan mengeksekusi *node subscriber* pada Raspberry Pi. *Node subscriber* dapat di eksekusi menggunakan perintah ROS yang dijalankan pada terminal Linux Ubuntu.

Tabel 2. Komunikasi data *publisher* dan *subscriber*

Node Publisher	Topic	Message	Node Subscriber
/led_on	/led	"led_on"	/kendali_led
/led_off	/led	"led_off"	
/kipas_on	/kipas	"kipas_on"	/kendali_kipas
/kipas_off	/kipas	"kipas_off"	
/heater_on	/heater	"heater_on"	/kendali_heater
/heater_off	/heater	"heater_off"	
/ac_on	/ac	"ac_on"	/kendali_ac
/ac_off	/ac	"ac_off"	

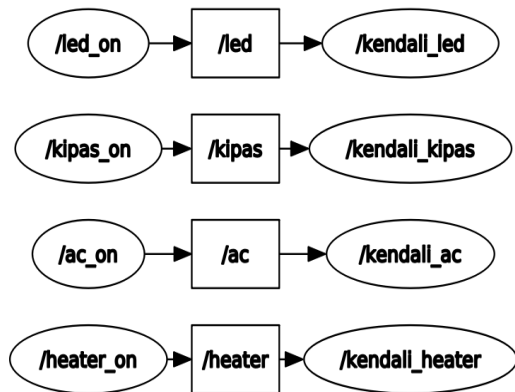
Berikut adalah perintah ROS yang digunakan untuk menjalankan *node subscriber* di Raspberry Pi.

Perintah ROS menjalankan *node /kendali_led*:
`$roslaunch my_package kendali_led.py`
 Perintah ROS menjalankan *node /kendali_kipas*:
`$roslaunch my_package kendali_kipas.py`
 Perintah ROS menjalankan *node /kendali_heater*:
`$roslaunch my_package kendali_heater.py`
 Perintah ROS menjalankan *node /kendali_ac*:
`$roslaunch my_package kendali_ac.py`

Langkah selanjutnya adalah mengaktifkan *tools wireless tethering* pada *smartphone* android kemudian mengkoneksikan Raspberry Pi ke *smartphone* android menggunakan jaringan nirkabel. Setelah Raspberry Pi terkoneksi dengan *smartphone*, langkah selanjutnya adalah menjalankan aplikasi *Mobile SSH* pada *smartphone* kemudian koneksikan ke Raspberry Pi dengan cara memasukkan alamat IP *address* Raspberry Pi ke *tools Mobile SSH*. Setelah terkoneksi maka *smartphone* sudah siap digunakan untuk mengontrol perangkat elektronik secara jarak jauh menggunakan jaringan nirkabel. Berikut adalah perintah ROS yang digunakan untuk menyalakan dan mematikan perangkat elektronik melalui aplikasi *Mobile SSH*.

Perintah ROS menyalakan lampu:
`$roslaunch my_package led_on.py`
 Perintah ROS mematikan lampu:
`$roslaunch my_package led_off.py`
 Perintah ROS menyalakan kipas:
`$roslaunch my_package kipas_on.py`
 Perintah ROS mematikan kipas:
`$roslaunch my_package kipas_off.py`
 Perintah ROS menyalakan heater:
`$roslaunch my_package heater_on.py`
 Perintah ROS mematikan heater:
`$roslaunch my_package heater_off.py`
 Perintah ROS menyalakan AC:
`$roslaunch my_package ac_on.py`
 Perintah ROS mematikan AC:
`$roslaunch my_package ac_off.py`

Pada saat *smartphone* mengirim perintah ROS *node publisher* ke Raspberry Pi untuk menyalakan atau mematikan perangkat elektronik, maka ROS *node subscriber* akan menerima data *message* dari *topic* untuk memberikan instruksi ke GPIO pin agar aktif *high* (1) atau aktif *low* (0). Data *message* dari *node publisher* akan terus dikirim ke *subscriber* hingga pengguna menghentikannya melalui aplikasi *mobile SSH*. Komunikasi data antara *node publisher* dan *node subscriber* dapat dimonitor dengan menggunakan *tools rqt* yang disediakan oleh ROS. Berikut adalah gambar komunikasi data antara *node publisher* dan *node subscriber*.



Gambar 4. Komunikasi *node publisher* ke *node subscriber*

ROS menyediakan layanan untuk melihat hasil *output* komunikasi data yang dikirim dari *node publisher* ke *node subscriber* menggunakan *script* `ROS_INFO`, sehingga pada saat sistem dijalankan maka akan tampil pesan atau data yang dikirim dari *publisher* ke *subscriber*. Tabel 3 berikut ini memperlihatkan hasil *output* komunikasi antara *node publisher* yang ditampilkan pada *smartphone* dan *node subscriber* yang ditampilkan pada Raspberry Pi.

Tabel 3. Hasil *output* komunikasi data antara *node publisher* dan *node subscriber*

Node Publisher	Node Subscriber
"led_on"	"Lampu Menyala"
"led_off"	"Lampu Mati"
"kipas_on"	"Kipas Menyala"
"kipas_off"	"Kipas Mati"
"heater_on"	"Pemanas Menyala"
"heater_off"	"Pemanas Mati"
"ac_on"	"AC Menyala"
"ac_off"	"AC Mati"

Pada saat *node subscriber* memberikan instruksi kepada GPIO pin untuk aktif *high* atau *low*, maka relay yang berfungsi sebagai saklar elektronik akan menyalakan atau mematikan perangkat elektronik. Berikut adalah tabel GPIO pin yang digunakan untuk mengontrol perangkat elektronik.

Tabel 4. Penggunaan GPIO pin Raspberry Pi

GPIO pin	Perangkat elektronik
GPIO 2	Lampu
GPIO 3	Kipas
GPIO 4	Pemanas Ruangan (Heater)
GPIO 5	Air Conditioner (AC)

Data *message* yang dikirim dari *node publisher* ke *node subscriber* merupakan inti dari komunikasi sistem ini, dimana pada saat data *message* sampai ke *node subscriber* dan sesuai dengan instruksi, maka sistem akan memerintahkan untuk menyalakan atau mematikan perangkat elektronik. Berikut adalah tabel hubungan antara data *message*, GPIO pin, dan perangkat elektronik yang di kontrol.

Tabel 5. Relasi data *message*, GPIO pin, dan perangkat elektronik yang di kontrol

Message	GPIO pin	Kondisi GPIO	Perangkat Elektronik	Kondisi Perangkat
"led_on"	2	High	Lampu	Menyala
"led_off"	2	Low	Lampu	Mati
"kipas_on"	3	High	Kipas	Menyala
"kipas_off"	3	Low	Kipas	Mati
"heater_on"	4	High	Heater	Menyala
"heater_off"	4	Low	Heater	Mati
"ac_on"	5	High	AC	Menyala
"ac_off"	5	Low	AC	Mati

Berdasarkan informasi tabel 5 dapat dilihat bahwa pada saat data *message* memproses data string "on" maka GPIO pin akan aktif *high* (1) kemudian perangkat elektronik akan menyala, selanjutnya pada saat data *message* memproses data string "off" maka GPIO pin akan aktif *low* (0) kemudian perangkat elektronik akan mati. Berikut adalah tabel hasil pengukuran tegangan fisis ketika perangkat elektronik dalam keadaan menyala dan mati berdasarkan perintah dari *message*.

Tabel 6. Tegangan fisis yang dihasilkan pada saat mengontrol perangkat elektronik

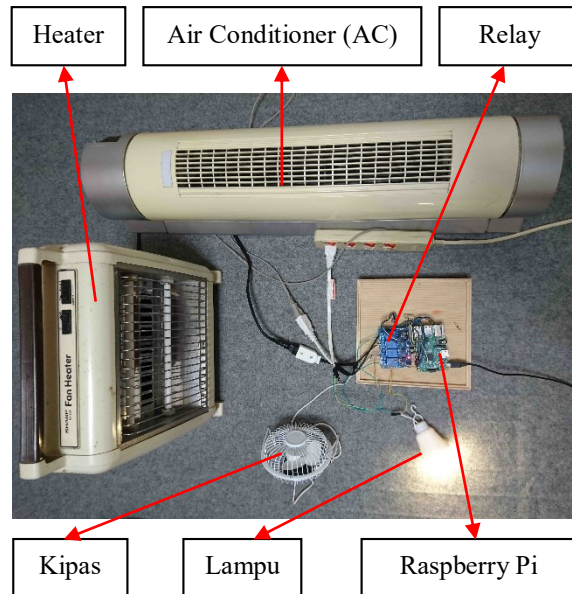
Message	GPIO pin	Kondisi GPIO	Tegangan GPIO	Tegangan Listrik
"led_on"	2	High	5 Volt DC	220 Volt AC
"led_off"	2	Low	0 Volt DC	Idle
"kipas_on"	3	High	5 Volt DC	220 Volt AC
"kipas_off"	3	Low	0 Volt DC	Idle
"heater_on"	4	High	5 Volt DC	220 Volt AC
"heater_off"	4	Low	0 Volt DC	Idle
"ac_on"	5	High	5 Volt DC	220 Volt AC
"ac_off"	5	Low	0 Volt DC	Idle

Perangkat elektronik rumah tangga yang di kontrol pada penelitian ini adalah lampu, kipas, pemanas ruangan, dan AC. Berikut adalah gambar perangkat elektronik yang di kontrol secara jarak jauh menggunakan SSH dan ROS.

Pada penelitian ini, uji coba pengontrolan perangkat elektronik secara jarak jauh telah di terapkan untuk mengetahui kualitas dari komunikasi data yang di kirim dari *smartphone* android ke Raspberry Pi. Berikut adalah tabel hasil uji coba sistem pengontrolan di dalam rumah dan di area bebas hambatan.

Tabel 7. Uji coba sistem pengontrolan jarak jauh (dalam rumah)

Message	Jarak (Meter)	Kondisi Perangkat	Kualitas Koneksi	Error Koneksi
"led_on"	2	Lampu On	Baik	0 %
	10	Lampu On	Baik	2 %
	15	Lampu On	Baik	5 %
"led_off"	5	Lampu Off	Baik	0 %
	18	Lampu Off	Baik	9 %
	25	Lampu Off	Kurang	35 %
"kipas_on"	8	Kipas On	Baik	1 %
	20	Kipas On	Baik	10 %
	30	Idle	Buruk	80 %
"kipas_off"	12	Kipas Off	Baik	3 %
	23	Kipas Off	Kurang	28 %
	28	Idle	Buruk	55 %



Gambar 5. Pengontrolan perangkat elektronik

Tabel 8. Uji coba sistem pengontrolan jarak jauh (area bebas hambatan)

Message	Jarak (Meter)	Kondisi Perangkat	Kualitas Koneksi	Error Koneksi
"heater_on"	5	Heater On	Baik	0 %
	20	Heater On	Baik	5 %
	30	Heater On	Baik	7 %
"heater_off"	10	Heater Off	Baik	1 %
	35	Heater Off	Baik	8 %
	55	Heater Off	Kurang	38 %
"ac_on"	15	AC On	Baik	2 %
	25	AC On	Baik	6 %
	52	AC On	Kurang	30 %
"ac_off"	38	AC Off	Baik	10 %
	57	Idle	Buruk	44 %
	60	Idle	Buruk	70 %

Berdasarkan informasi tabel 7 dan 8, dapat dilihat bahwa sistem dapat bekerja dengan baik di dalam ruangan dengan jarak maksimum 20 meter dan jarak 40 meter di area bebas hambatan. Kualitas koneksi dalam kondisi baik jika *smartphone* dapat mengirim perintah ROS *message* ke Raspberry Pi dengan persentase error komunikasi data antara 0% hingga 10%, kualitas kurang baik jika persentase error komunikasi data antara 10% hingga 40%, dan kualitas buruk jika jika persentase error komunikasi data di atas 40%.

Pengontrolan perangkat elektronik secara jarak jauh menggunakan *wireless tethering* sangat efektif diaplikasikan jika dibandingkan dengan beberapa teknologi nirkabel lainnya berbasis koneksi lokal, seperti Bluetooth yang hanya mampu terkoneksi hingga 20 meter dalam ruang terbuka (Sadewo et al., 2017), dan *remote control* yang mampu mengontrol hingga 15 meter (Abdullah., 2019). Selain itu, arsitektur jaringan yang cukup sederhana menjadikan sistem ini sangat efektif karena tidak membutuhkan perangkat tambahan untuk saling terkoneksi seperti WiFi 2,4 GHz (Masinambow et al., 2014), koneksi

internet (Hendrawati et al., 2018), dan *Cloud Computing* (Iksan et al., 2015).

4. KESIMPULAN

Pengontrolan perangkat elektronik secara jarak jauh berbasis jaringan nirkabel menggunakan protokol *Secure Shell* (SSH) dan perangkat lunak *Robot Operating System* (ROS) telah diaplikasikan pada penelitian ini. Pemanfaatan *tools wireless tethering* pada *smartphone* yang terhubung ke Raspberry Pi dapat diaplikasikan untuk mengontrol *on* dan *off* perangkat elektronik. Metode yang digunakan untuk mengontrol perangkat elektronik secara jarak jauh adalah menggunakan metode *Remote Access Control* (RAC) berbasis protokol SSH. Perangkat elektronik akan menyala pada saat *smartphone* android mengirim ROS *message* ke Raspberry Pi melalui aplikasi *Mobile SSH* untuk memberikan instruksi kepada GPIO pin agar aktif *high*, selanjutnya perangkat elektronik akan mati pada saat GPIO pin menerima instruksi dari ROS *message* agar aktif *low*. Perangkat elektronik berupa lampu, kipas angin, pemanas ruangan (*heater*) dan *air conditioner* (AC) dapat di kontrol dengan baik pada jarak maksimum 20 meter di dalam ruangan dan jarak 40 meter pada area bebas hambatan.

DAFTAR PUSTAKA

- ABDULLAH, MH., 2019. Rancang Bangun Sistem Kontrol Lampu Listrik Menggunakan Remote Berbasis Mikrokontroler ATmega 8535. *Jurnal Ilmiah ILKOMINFO – Jurnal Ilmu Komputer dan Informatika*, Vol.2, No.1, pp.40-47.
- ALAMSYAH, A., dan FAISAL, M.N., 2015. Perancangan dan Penerapan Sistem Kontrol Peralatan Elektronik Jarak Jauh Berbasis Web. *Jurnal Mekanikal*, Vol.6, No.2, pp.577-584.
- BAHRI, S., dan SUDRAJAT, A., 2015. Rancang Bangun Prototype Sistem Kontrol Jarak Jauh Berbasis Ponsel Android. *Symposium Nasional Teknologi Terapan (SNTT)* 3. ISSN: 2339-028X.
- HASAN, M., dan TAHTAWI, ARA., 2018. Detektor Dini Kebakaran Multisensor Terintegrasi Android Menggunakan Komunikasi Bluetooth. *Jurnal Teknologi dan Sistem Komputer*, Vol.6, No.2, pp.64-70.
- HENDRAWATI, TD., WICAKSONO, YD., dan ANDIKA, E., 2018. Internet of Things: Sistem Kontrol-Monitoring Daya Perangkat Elektronik. *JTERA – Jurnal Teknologi Rekayasa*, Vol.3, No.2, pp.177-184.
- IKSAN, N., dan ARFRIANDI, A., 2015. Pengendali Listrik Rumah Berbasis Cloud Computing. *Jurnal Infotel* Vol.7 No.1, pp.53-60.
- JALIL, A., 2018. Robot Operating System (ROS) dan Gazebo Sebagai Media Pembelajaran Robot Interaktif. *ILKOM Jurnal Ilmiah*, Vol.10, No.3, pp.284-289.
- JALIL, A., 2019. Pemanfaatan Middleware Robot Operating System (ROS) Dalam Menjawab Tantangan Revolusi Industri 4.0. *ILKOM Jurnal Ilmiah*, Vol.11, No.1, pp.45-52.
- JUSUF, H., 2015. Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online. *BINA INSANI ICT JOURNAL*, Vol.2, No.2, pp.75-84.
- KONG, G., dan KAM, J., 2012. Using Mobile Tethering for Sharing Data Across Devices: Application in Rural Eye Screening. *JOURNAL OF MOBILE TECHNOLOGY IN MEDICINE*, VOL.1, ISSUE 3, pp.40-45.
- KURNIADI, D., dan AMELIA, L., 2018. Sistem Kendali Perangkat Elektronik Rumah Berbasis Android dan Arduino. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut (STT-Garut)*, Vol.15, No.2, pp.1-6.
- KURNIADI, D., dan FITRIYANI., 2017. Sistem Kendali Jarak Jauh Perangkat Elektronik Rumah Berbasis Cloud Computing. *Jurnal Algoritma Sekolah Tinggi Teknologi Garut (STT-Garut)*, Vol.4, No.2, pp.205-214.
- MASINAMBOW, V., NAJOAN, MEI., dan LUMENTA, ASM., 2014. Pengendali Saklar Listrik Melalui Ponsel Pintar Android. *E-journal Teknik Elektro dan Komputer*, pp.1-9, ISSN 2301-8402.
- MARUYAMA, Y., KATO, S., dan AZUMI, T., 2016. Exploring the Performance of ROS2. *EMSOFT'16, Pittsburgh, PA, USA*. ISBN 978-1-4503-4485-2/16/10.
- MUZAWI, R., EFENDI, Y., dan SAHRUN, N., 2018. Prototype Pengendalian Lampu Jarak Jauh dengan Jaringan Internet Berbasis Internet of Things(IoT) Menggunakan Raspberry Pi 3. *INFORM: Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi* Vol.3 No.1, pp.46-50.
- PRIHATMOKO, D., 2017. Pemanfaatan Raspberry Pi Sebagai Server Web Untuk Penjadwalan Kontrol Lampu Jarak Jauh. *Jurnal Infotel*, Vol.9, No.1, pp. 84-91.
- RAHMIATI, P., FIRDAUS, G., dan FATHORRAHMAN, N., 2014. Implementasi Sistem Bluetooth menggunakan Android dan Arduino untuk Kendali Peralatan Elektronik. *Jurnal ELKOMIKA Itenas*, Vol.2, No.1, pp.1-14.
- SADEWO, ADB., WIDASARI, ER., dan MUTTAQIN, A., 2017. Perancangan Pengendali Rumah menggunakan Smartphone Android dengan Konektivitas Bluetooth. *Jurnal Pengembangan Teknologi*

Informasi dan Ilmu Komputer, Vol.1, No.5, pp. 415-425.

- SALEH, M., dan HARYANTI, M., 2017. Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay. *Jurnal Teknologi Elektro, Universitas Mercu Buana, Vol.8 No.3, pp.181-186.*
- SHADIQ, HM., SUDJADI., dan DARJAT., 2014. Perancangan Kamera Pemantau Nirkabel Menggunakan Raspberry Pi Model B. *TRANSIENT, Vol.3, No.4, pp.546-551.*
- SHARON, D., SAPRI., dan SUPARDI, R., 2014. Membangun Jaringan Wireless Local Area Network (WLAN) Pada CV.Biq Bengkulu. *Jurnal Media Infotama Vol.10 No.1, pp.35-41.*
- YULIANA, G., NASEER, M., dan FEBRIYA, F., 2018. Rancang Bangun Sistem Kendali Lampu Gedung Berbasis Android Melalui Jaringan Wifi. *Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang, pp.496-501.*