

PEMANFAATAN *MIDDLEWARE ROBOT OPERATING SYSTEM* (ROS) DALAM MENJAWAB TANTANGAN REVOLUSI INDUSTRI 4.0

Abdul Jalil
abdul.jalil.fw@gmail.com
STMIK Handayani Makassar

Abstrak

Revolusi industri 4.0 merupakan pengembangan industri berbasis internet, kontrol, robotika, dan jaringan. Penelitian ini memaparkan penerapan *middleware Robot Operating System* (ROS) dalam menjawab tantangan revolusi industri 4.0 untuk mengontrol peralatan elektronik industri dan rumah tangga menggunakan Raspberry Pi. ROS memiliki *node*, *topic* dan *message* yang dapat digunakan untuk mengontrol pin GPIO Raspberry Pi agar aktif *high* (1) atau aktif *low* (0). Hasil penelitian ini adalah sistem dapat mengontrol *on* dan *off* perangkat elektronik melalui pin GPIO Raspberry Pi dan modul relay berdasarkan perintah *message* dari ROS *node*. Perangkat elektronik yang dikontrol pada penelitian ini adalah kipas angin, lampu, pendingin ruangan, dan pemanas ruangan yang memiliki tegangan 110/220 Volt AC.

Kata Kunci: Revolusi industri 4.0, *Middleware Robot Operating System* (ROS), Raspberry Pi.

Abstract

The industrial revolution 4.0 is the development of industrial based on the internet, control, robotics, and networking. This study describes the usage of a Robot Operating System (ROS) middleware to answer the challenges of the industrial revolution 4.0 for control the industrial electronic devices and household electronic equipment using Raspberry Pi. ROS has nodes, topics and messages that can be used to manage the Raspberry Pi GPIO pins to be active high (1) or active low (0). The result of this study is the system be able to control the on and off electronic devices through the GPIO pins of Raspberry Pi and relay module based on message commands from the ROS node. The electronic devices had controlled in this study are a fan, light, air conditioner, and room heater that have 110/220 AC voltage.

Keyword: Industrial revolution 4.0, Robot Operating System (ROS) middleware, Raspberry Pi.

1. Pendahuluan

Pengembangan revolusi industri 4.0 menjadi tantangan oleh banyak kalangan seperti akademisi, industri, pemerintahan, dan masyarakat. Tantangan tersebut meliputi perubahan di bidang sosial [1],[2], pembelajaran berbasis teknologi [3], perkembangan riset [4], pengembangan keilmuan di bidang industri [5], dan revolusi mental [6]. Beberapa bidang yang menjadi topik dalam pengembangan revolusi industri 4.0 adalah *autonomous robot*, *smart machines*, *system integration*, *Internet of Things (IoT)*, *cyber security*, *cloud computing*, *additive manufacturing*, *augmented reality*, *3D printing*, *data analytic*, dan *big data processing* [7].

Kendala yang timbul dalam menjawab tantangan revolusi industri 4.0 diantaranya adalah kesiapan sumber daya manusia (SDM) dan penerapan teknologi [1],[3],[4],[5],[7]. Pentingnya dunia akademik dalam menyiapkan tenaga SDM yang terampil di bidang teknologi menjadi tolak ukur keberhasilan dalam menjawab tantangan revolusi industri 4.0 [3],[5],[8]. Keberhasilan dunia akademik dan penelitian telah menjadikan teknologi Programmable Logic Controller (PLC), mikrokontroler, mikrokomputer, dan teknik antarmuka komputer sebagai media untuk mengontrol perangkat elektronik yang dapat diaplikasikan pada teknik pengendalian industri.

Pemanfaatan PLC untuk mengontrol peralatan elektronik telah diaplikasikan untuk mengatur mesin pemotong kentang [9], mengontrol *on/off belt conveyor* [10], dan mengontrol peralatan *building* otomatis [11]. Penggunaan mikrokontroler untuk mengontrol perangkat elektronik telah diaplikasikan pada sistem kendali perangkat elektronik rumah menggunakan arduino [12], sistem pengendali peralatan elektronik rumah secara otomatis menggunakan mikrokontroler ATmega16 [13], dan perancangan alat untuk pengendali listrik rumah menggunakan mikrokontroler ATmega8 [14]. Pengontrolan alat elektronik menggunakan mikrokomputer telah diaplikasikan pada pengontrolan alat listrik menggunakan Raspberry Pi [15], dan aplikasi rumah pintar untuk pengendali peralatan elektronik rumah tangga menggunakan Raspberry Pi [16]. Selain itu penggunaa teknik *interface*



parallel port komputer untuk mengontrol peralatan elektronik telah di aplikasikan pada sistem kontrol peralatan elektronik jarak jauh berbasis web [17].

Pada penelitian ini akan memaparkan sebuah pengembangan sistem pengontrolan perangkat elektronik menggunakan *middleware Robot Operating System (ROS)* dan Raspberry Pi. ROS merupakan perangkat lunak kontrol yang digunakan untuk mengontrol sistem robotika, berkembangnya teknologi telah menjadikan ROS sebagai perangkat lunak kontrol yang terus dikembangkan oleh komunitas robot besar di dunia. Penggunaan ROS untuk mengontrol perangkat elektronik industri akan sangat efisien karena ROS bersifat *open source* serta mudah untuk dikembangkan, selain itu ROS dapat beroperasi pada sistem jaringan yang terdistribusi.

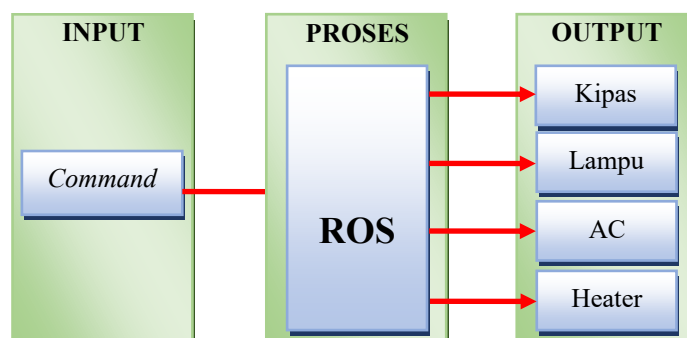
2. Metode

Middleware robot adalah perangkat lunak *tools* dan *library* yang digunakan untuk mengontrol sistem robotika. *Middleware* ini terletak di antara sistem operasi dan perangkat lunak aplikasi [18],[19] serta berfungsi sebagai penghubung antara perangkat keras sensor, aktuator, dan komponen perangkat lunak [19]. Beberapa *middleware* telah dikembangkan dan digunakan oleh para peneliti untuk mengontrol sistem robotika seperti Player/Stage [20], Miro, MRDS, ASEBA, Orca, Pyro, ROS, dan ROS2 [21],[22]. Pada penelitian ini akan memanfaatkan *middleware Robot Operating System (ROS)* dan Raspberry Pi untuk mengontrol *on/off* perangkat elektronik yang dapat diaplikasikan pada teknik pengontrolan perangkat elektronik industri.

ROS adalah *middleware* robot yang bersifat *open source* serta memiliki seperangkat *tools* dan *library* yang digunakan untuk membuat perangkat lunak robot [23],[24],[25],[26]. ROS telah diaplikasikan untuk mengontrol sistem robotika [27],[28] dan dapat digunakan untuk membuat simulasi robot [8],[29],[30]. ROS berjalan di atas sistem operasi Linux dan dapat dioperasikan pada Raspberry Pi. Raspberry Pi merupakan sebuah mikrokomputer yang berukuran kecil dan mempunyai fungsi sama seperti komputer serta dapat digunakan untuk pengontrolan perangkat keras.

Raspberry Pi telah digunakan oleh para peneliti untuk pengontrolan seperti mengontrol mesin penukaran uang kertas rupiah berbasis pengolahan citra dan raspberry pi [31], simulator *input-output* sistem kontrol menggunakan raspberry pi [32], *Monitor and Control of Remote Appliances using Raspberry Pi through IoT* [33], dan *Raspberry Pi Home Automation with Wireless Sensors using Smart Phone* [34]. Pemanfaatan pin GPIO Raspberry Pi pada penelitian ini digunakan untuk mengontrol on dan off perangkat elektronik berdasarkan perintah *message* dari ROS *node*.

Pada gambar 1 memperlihatkan arsitektur sistem yang telah dibangun, sistem tersebut terdiri dari tiga bagian yaitu *input*, proses dan *output*. Input sistem menggunakan *command* atau perintah *node* yang dikontrol dari *middleware* ROS. Proses sistem menggunakan Raspberry Pi yang dioperasikan menggunakan sistem operasi Linux Ubuntu Mate 16.04 dan *middleware* robot ROS Kinetic. Output sistem ini adalah mengontrol *on/off* perangkat elektronik yaitu kipas angin, lampu, pendingin ruangan (AC), dan pemanas ruangan. Dalam mengontrol perangkat elektronik dibutuhkan rangkaian modul relay yang dapat mengalirkan arus tegangan 110/220 Volt AC. Berikut adalah gambar arsitektur sistem yang dibangun dapat dilihat pada gambar 1.



Gambar 1. Arsitektur sistem

Command pada ROS merupakan perintah berbasis *command line interface* yang dikirim ke pin GPIO Raspberry Pi. Perintah-perintah tersebut dapat dieksekusi setelah *middleware* ROS Kinetic terinstall pada Raspberry Pi. ROS memiliki 3 level konsep, yaitu level *filesystem*, level grafik komputasi dan level komunitas [23],[24]. Pada ROS terdapat *node* yang saling terhubung menggunakan *topic*

untuk dapat saling mengirim *message* dalam bentuk tipe data. Perangkat lunak pemrograman yang digunakan pada penelitian ini adalah menggunakan bahasa pemrograman Python yang di proses menggunakan *middleware* ROS.

Raspberry Pi merupakan salah satu jenis mikrokomputer yang mempunyai fungsi dan kegunaan sama seperti komputer pada umumnya. Kelebihan Raspberry Pi dalam mengolah data komputer dan mengontrol perangkat elektronik membuat mikrokomputer ini banyak digunakan oleh para peneliti sehingga mudah untuk dikembangkan. Raspberry Pi mempunyai GPIO pin yang dapat digunakan untuk menerima data *input* dari sensor dan mengirim data *output* untuk mengontrol perangkat keras elektronik. Adapun jenis Raspberry Pi yang digunakan pada penelitian ini adalah Raspberry Pi 3 Model B. Pada Raspberry Pi 3 model B terdapat 26 pin GPIO yang dapat digunakan untuk mengontrol perangkat *interface*. Dalam mengontrol perangkat elektronik tegangan 110/220 Volt AC dibutuhkan rangkaian pemacu relay. Relay merupakan sebuah saklar elektronik yang dapat dikendalikan menggunakan arus [35]. Pada pengontrolan ini digunakan empat buah relay 5 Volt DC yang digerakkan menggunakan seperangkat rangkaian elektronik dan dipicu berdasarkan input *high* (1) atau *low* (0) dari pin GPIO Raspberry Pi. Berikut adalah gambar Raspberry Pi dan modul relay yang digunakan pada penelitian ini dapat dilihat pada gambar 2 dan gambar 3.



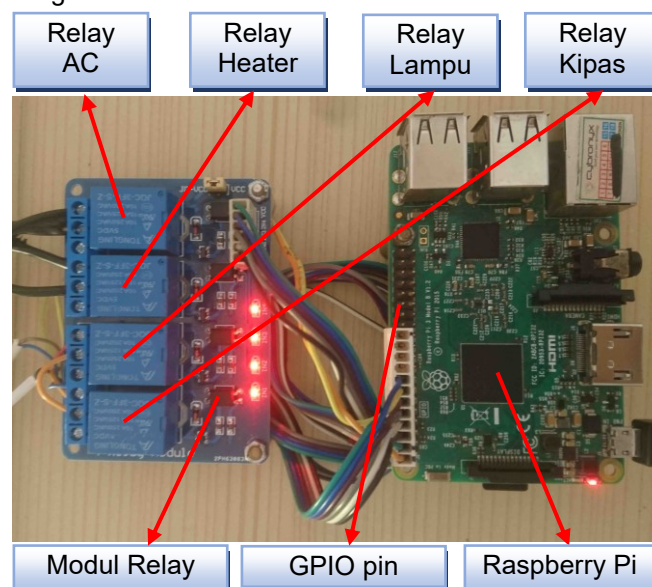
Gambar 2. Raspberry Pi 3 Model B



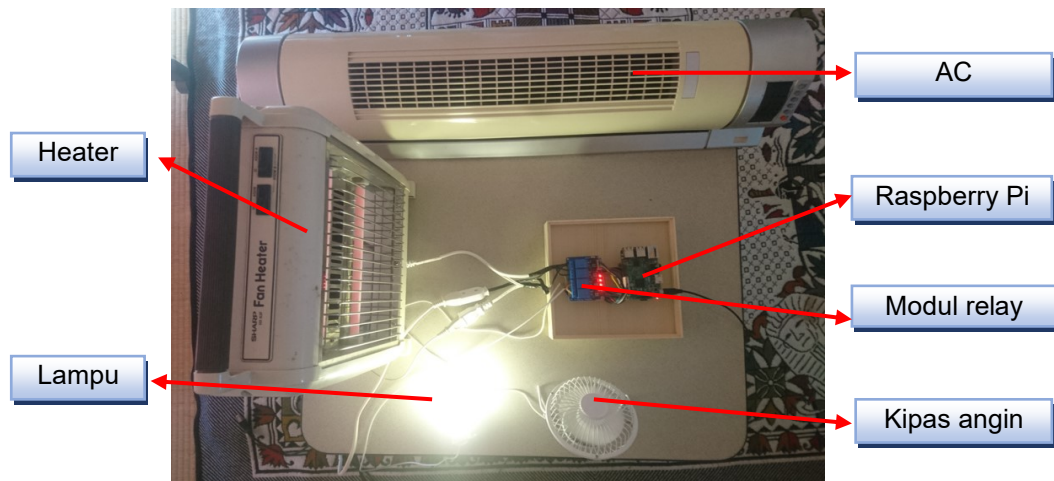
Gambar 3. Rangkaian modul relay

3. Hasil dan Pembahasan

Teknik pengontrolan perangkat elektronik berupa kipas, lampu, pendingin ruangan (AC), dan pemanas ruangan (heater) telah diaplikasikan pada penelitian ini. Terdapat dua tahapan perancangan pada sistem ini yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras dilakukan dengan cara merakit dan menghubungkan semua komponen antara Raspberry Pi, modul relay, dan perangkat elektronik. Pada perancangan perangkat lunak dilakukan dengan menginstal *middleware* ROS Kinetic pada Raspberry Pi kemudian membuat perangkat lunak kontrol menggunakan bahasa pemrograman Python. Hasil dari perancangan tersebut kemudian dapat diaplikasikan untuk membangun sebuah sistem pengontrolan perangkat elektronik dengan menggunakan *middleware* ROS dan Raspberry Pi. Pada gambar 4 dan 5 memperlihatkan hasil dari penelitian yang telah dibangun.



Gambar 4. Hasil perancangan perangkat keras



Gambar 5. Hasil perancangan sistem

Pada Raspberry Pi telah digunakan 4 pin GPIO untuk mengontrol perangkat elektronik. Berikut adalah tabel penggunaan GPIO pin Raspberry Pi yang terhubung ke relay dan peralatan elektronik dapat dilihat pada tabel 1.

Tabel 1. Koneksi GPIO pin dengan perangkat elektronik

GPIO Pin	Perangkat Elektronik
Pin 8	Kipas angin
Pin 10	Lampu
Pin 12	Pendingin ruangan (AC)
Pin 16	Pemanas ruangan (<i>Heater</i>)

Input pada sistem ini adalah berupa perintah node atau *command* yang dikirim dari *middleware* ROS untuk mengontrol GPIO pin Raspberry Pi. Pada ROS terdapat *node*, *topic*, dan *message* yang saling terhubung dan berkomunikasi antara satu dengan lainnya. Informasi komunikasi data antara *node*, *topic*, dan *message* dapat dilihat pada tabel 2, setiap *node*, *topic*, dan *message* tersebut mempunyai nama yang berbeda-beda. Pada sistem komunikasi data ROS, setiap *node* dapat saling terhubung dengan syarat harus mempunyai nama *topic* yang sama. Komunikasi data yang diterapkan pada sistem ini adalah sistem *publish* dan *subscribe node* dalam mengirim data *message* untuk mengontrol kondisi *high* (1) dan *low* (0) pin GPIO Raspberry Pi dapat dilihat pada tabel 2.

Tabel 2. *Node*, *topic*, dan *message* ROS

<i>Node Publish</i>	<i>Message</i>	<i>Topic</i>	<i>Node Subscribe</i>
/kipas_on	"kipas_on"		
/kipas_off	"kipas_off"		
/lampu_on	"lampu_on"		
/lampu_off	"lampu_off"		
/ac_on	"ac_on"	/industri	/gpio_pin
/ac_off	"ac_off"		
/heater_on	"heater_on"		
/heater_off	"heater_off"		
/semua_on	"semua_on"		
/semua_off	"semua_off"		

Perangkat elektronik yang dikontrol pada penelitian ini adalah perangkat yang dapat aktif pada tegangan 110/220 Volt AC, untuk mengaktifkan perangkat elektronik tersebut dibutuhkan rangkaian modul relay sebagai saklar elektronik yang dapat aktif jika mendapat perintah dari GPIO pin berupa kondisi 1 (*on*) dan kondisi 0 (*off*). Jika GPIO berlogika 1 maka GPIO pin akan mengaktifkan rangkaian relay agar dapat menyalakan perangkat elektronik, jika GPIO berlogika 0 maka GPIO pin

memerintahkan relay untuk mematikan perangkat elektronik. Berikut adalah tabel relasi antara *message* ROS, GPIO pin dan perangkat elektronik dapat dilihat pada tabel 3.

Tabel 3. Relasi antara *message* ROS, kondisi GPIO pin, dan kontrol perangkat elektronik

Message ROS	Kondisi GPIO Pin	Perangkat Elektronik
"kipas_on"	Pin 8 = 1	Kipas On
"lampu_on"	Pin 10 = 1	Lampu On
"heater_on"	Pin 12 = 1	AC On
"ac_on"	Pin 16 = 1	Heater On
"semua_on"	Pin 8,16,12,16 = 1	Semuanya On
"kipas_off"	Pin 8 = 0	Kipas Off
"lampu_off"	Pin 10 = 0	Lampu Off
"heater_off"	Pin 12 = 0	AC Off
"ac_off"	Pin 16 = 0	Heater Off
"semua_off"	Pin 8,16,12,16 = 0	Semuanya Off

Pada *middleware* ROS terdapat *workspace* dan *package* untuk menyimpan *node-node* yang akan dijalankan pada sistem, adapun nama *package* yang digunakan pada sistem ini adalah */industri_40*. ROS juga memiliki *command* atau perintah-perintah yang digunakan untuk menjalankan ROS *master* dan *node*. Perintah tersebut di eksekusi menggunakan *command line interface* pada terminal Linux. Berikut adalah *command* atau perintah ROS yang digunakan pada penelitian ini:

Perintah untuk menjalankan master ROS:

```
$roscore
```

Perintah untuk menyalakan kipas:

```
$rosrn industri_40 kipas_on.py
```

Perintah untuk mematikan kipas:

```
$rosrn industri_40 kipas_off.py
```

Perintah untuk menyalakan lampu:

```
$rosrn industri_40 lampu_on.py
```

Perintah untuk mematikan lampu:

```
$rosrn industri_40 lampu_off.py
```

Perintah untuk menyalakan pendingin ruangan:

```
$rosrn industri_40 ac_on.py
```

Perintah untuk mematikan pendingin ruangan:

```
$rosrn industri_40 ac_off.py
```

Perintah untuk menyalakan pemanas ruangan:

```
$rosrn industri_40 heater_on.py
```

Perintah untuk mematikan pemanas ruangan:

```
$rosrn industri_40 heater_off.py
```

Perintah untuk menyalakan semua perangkat elektronik:

```
$rosrn industri_40 semua_on.py
```

Perintah untuk mematikan semua perangkat elektronik:

```
$rosrn industri_40 semua_off.py
```

Perintah untuk membaca *message* node dan mengontrol GPIO:

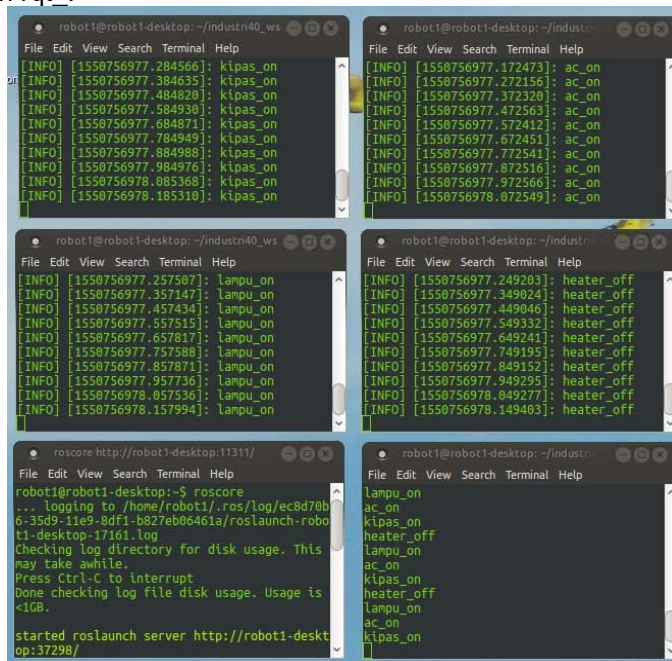
```
$rosrn industri_40 gpio_pin.py
```

Berikut adalah tabel penjelasan perintah ROS yang digunakan pada penelitian ini dapat dilihat pada tabel 4.

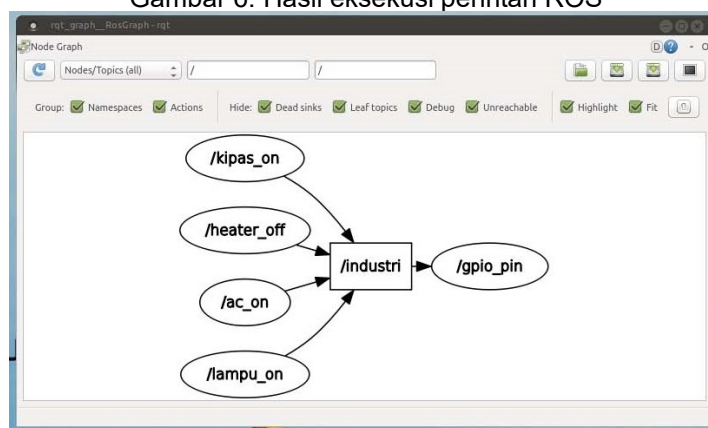
Tabel 4. Fungsi perintah ROS

ROS Command	Keterangan
roscore	ROS <i>Master</i>
rosrn	Perintah menjalankan ROS <i>node</i>
industri_40	Nama <i>package</i>
kipas_on.py	Nama <i>node</i>

Berdasarkan informasi perintah dan tabel 4 dapat dilihat bahwa untuk menjalankan ROS *node* maka pertama yang harus dilakukan adalah menjalankan ROS *master* (*roscore*). ROS *node* dapat saling berkomunikasi antara satu dengan lainnya menggunakan protokol TCPROS/UDPROS, sehingga sistem ini dapat digunakan pada jaringan terdistribusi. Perintah *roscore* adalah command ROS untuk menjalankan *node*, kemudian memanggil nama *package* yaitu *industri_40*, kemudian memanggil nama *node* yaitu *kipas_on.py*. Setelah perintah tersebut dieksekusi kemudian setiap *node* akan saling terhubung sesuai dengan nama *topic* yaitu */industri*. *Node* yang terhubung kemudian akan saling mengirim *message* berupa tipe data string dari *node publish* ke *node subscribe*. Pada gambar 6 memperlihatkan hasil eksekusi perintah ROS pada setiap terminal Linux, dan gambar 7 memperlihatkan komunikasi antar *node publish* ke *node subscribe* melalui *topic /industri* menggunakan perintah *rqt*.



Gambar 6. Hasil eksekusi perintah ROS



Gambar 7. Komunikasi *node publish* dan *node subscribe*

Konsep pengontrolan menggunakan *middleware* ROS dan Raspberry pi yang telah dipaparkan pada penelitian ini sangat efisien digunakan untuk pengembangan sistem kontrol industri serta dapat menjawab tantangan revolusi industri 4.0. Kelebihan yang ada pada konsep sistem kontrol ini dibandingkan dengan sistem kontrol lainnya adalah pada pengembangan perangkat lunak kontrolnya. *Middleware* ROS terus dikembangkan oleh komunitas robot di dunia sehingga selalu ada pembaruan sistem, selain itu perangkat lunak ini bersifat *open source* sehingga sangat mudah untuk dikembangkan dan bebas untuk digunakan sesuai dengan standard lisensi perangkat lunak ROS.



4. Kesimpulan dan Saran

Pengembangan sistem kontrol peralatan elektronik menggunakan *middleware Robot Operating System (ROS)* dan Raspberry Pi telah diaplikasikan pada penelitian ini. Pada ROS terdapat *node*, *topic*, dan *message* yang saling terhubung menggunakan komunikasi protokol TCPROS/UDPROS. *Middleware ROS* dapat mengontrol kondisi pin GPIO untuk aktif *high* (1) atau *low* (0) berdasarkan *command* atau perintah *message* dari ROS *node* yang dijalankan pada sistem operasi Linux dan Raspberry Pi. Perangkat elektronik yang dikontrol pada penelitian ini adalah kipas, lampu, pendingin ruangan, dan pemanas ruangan yang memiliki tegangan 110/220 Volt AC. Pada saat ROS *node* memerintahkan pin GPIO untuk aktif *high* (1) maka modul relay akan mengaktifkan perangkat elektronik, dan saat ROS *node* mengirim sinyal aktif *low* (0) pada pin GPIO maka modul relay akan menonaktifkan perangkat elektronik. Konsep pengontrolan perangkat elektronik ini dapat digunakan dan dikembangkan untuk mengontrol perangkat elektronik pada dunia industri sehingga dapat menjawab tantangan revolusi industri 4.0 di bidang kontrol dan robotika. Saran untuk penelitian kedepannya adalah perangkat elektronik yang dikontrol bisa lebih banyak sesuai dengan jumlah pin GPIO Raspberry Pi yaitu maksimum 26 perangkat elektronik.

Daftar Pustaka

- [1] B. Prasetyo dan U. Trisyanti, "Revolusi Industri 4.0 dan Tantangan Perubahan Sosial," *Prosiding SEMATEKSOS 3, Strategi Pembangunan Nasional Menghadapi Revolusi Industri 4.0*, 2018.
- [2] Morrar, R., Arman, H., & Mousa, S. "The Fourth Industrial Revolution (Industri 4.0): A Social Innovation Perspective," *Technology Innovation Management Review*, 7(11): 12–20. 2017. DOI: <http://doi.org/10.22215/timreview/1117>.
- [3] Syamsuar dan Reflianto, "Pendidikan dan Tantangan Pembelajaran Berbasis Teknologi Informasi di Era Revolusi Industri 4.0," *Jurnal Ilmiah Teknologi Pendidikan (E-TECH)* Vol 2, No 2, 2018. DOI: 10.24036/et.v2i2.101343.
- [4] H. Prasetyo dan W. Sutopo, "Industri 4.0: Telaah Klasifikasi Aspek dan Arah Perkembangan Riset," *Jurnal Teknik Industri*, Vol. 13, No. 1, Januari 2018. [Online]. DOI: 10.14710/jati.13.1.17-26.
- [5] H. Prasetyo dan W. Sutopo, "Perkembangan Keilmuan Teknik Industri Menuju Era Industri 4.0," Seminar dan Konferensi Nasional IDEC 2017. ISSN: 2579-6429.
- [6] H. Suardana, "Revolusi Industri 4. 0 Berbasis Revolusi Mental," JATI UNIK, 2017, Vol.1, No.2, Hal. 102-110. ISSN : 2597-7946.
- [7] Liao, Y., Loures, E. R., Deschamps, F., Brezinski, G., & Venâncio, A. "The impact of the fourth industrial revolution: a cross-country/region comparison," *Production*, 28, e20180061, 2018. DOI: 10.1590/0103-6513.20180061.
- [8] A. Jalil, "Robot Operating System (ROS) dan Gazebo Sebagai Media Pembelajaran Robot Interaktif," *ILKOM Jurnal Ilmiah Volume 10 Nomor 3 Desember 2018*. DOI: 10.33096/ilkom.v10i3.365.284-289.
- [9] DW. Nugraha, "Sistem Pengaturan Mesin Pemotong Kentang Berbasis Programmable Logic Controller," *Jurnal SMARTek*, Vol. 8 No. 4. Nopember 2010: 270 – 279.
- [10] Taufik, W. Putri, "Perancangan Prototype Early Warning System Pada Kontrol On/Off Belt Conveyor Menggunakan PLC Siemens S7-300," *Jurnal Optimasi Sistem Industri*, Vol. 14 No. 1, April 2015: 116-137.
- [11] D. Yuhendri, "Penggunaan PLC Sebagai Pengontrol Peralatan Building Otomatis," *Journal of Electrical Technology*, Vol. 3, No.3, Oktober 2018.
- [12] D. Kurniadi, L. Amelia, "Sistem Kendali Perangkat Elektronik Rumah Berbasis Android dan Arduino," *Jurnal STT-Garut* Vol. 15 No. 2, 2018.
- [13] E. Desyantoro, AF. Rochim, KT. Martono, "Sistem Pengendali Peralatan Elektronik dalam Rumah secara Otomatis Menggunakan Sensor PIR, Sensor LM35, dan Sensor LDR," *Jurnal Teknologi dan Sistem Komputer*, Vol 3, No. 3, Agustus 2015.
- [14] A.Rosadi, "Perancangan Alat Pengendali Listrik Rumah Berbasis Mikrokontroler ATmega8," *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic*, J Pikel 2(2) : 61- 68 (2014).
- [15] MD. Nanda, Susandri, "Sistem Pengontrolan Peralatan Listrik secara Online Menggunakan Sensor Gerak (Studi Kasus PT.Capella Dinamik Nusantara Riau)," *SATIN - Sains dan Teknologi Informasi*, Vol. 1, No. 2, Desember 2015.



- [16] F. Masykur, F. Prasetyowati, "Aplikasi Rumah Pintar (Smart Home) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, Vol. 3, No. 1, Maret 2016.
- [17] Alamsyah, A. Amir, MN. Faisal, "Perancangan Dan Penerapan Sistem Kontrol Peralatan Elektronik Jarak Jauh Berbasis Web," *Jurnal Mekanikal*, Vol. 6 No. 2: Juli 2015: 577-584.
- [18] A. Elkady and T. Sobh, "Robotics Middleware: A Comprehensive Literature Survey and Attribute-Based Bibliography," *Journal of Robotics*, Volume 2012, Article ID 959013, 15 pages, 2012. DOI:10.1155/2012/959013.
- [19] S. Ceriani dan M. Migliavacca, "Middleware in Robotics," Internal report for "Advanced Methods of Information Technology for Autonomous Robotics", Prof. G. Gini. [Online, Diakses 30 Desember 2018].
- [20] K.S. Senthilkumar dan K. K. Bharadwaj, "Player-Stage Based Simulator for Simultaneous Multi-Robot Exploration and Terrain Coverage Problem," *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 2, No. 4, 2011. DOI: 10.5121/ijaia.2011.2410.
- [21] N. Mohamed, J. Al-Jaroodi, dan I. Jawhar, "Middleware for Robotics: A Survey," In Proc. of The IEEE Intl. Conf. on Robotics, Automation, and Mechatronics (RAM 2008), pp. 736-742, Sep. 2008.
- [22] SG. Chitic, J. Ponge, dan O. Simonin, "Are Middlewares Ready for Multi-robots Systems?," 4th International Conference, SIMPAR 2014, LNAI 8810, pp. 279–290, 2014.
- [23] A. Martinez and E. Fernandez, "Learning ROS for Robotics Programming," Published by Packt Publishing Ltd. Livery Place, Birmingham B3 2PB, UK. ISBN 978-1-78216-144-8, 2013.
- [24] E. Fernandez, LS. Crespo, Anil Mahtani and Aaron Martinez, "Learning ROS for Robotics Programming second edition," Published by Packt Publishing Ltd. Livery Place, Birmingham B3 2PB, UK. ISBN 978-1-78398-758-0, 2015.
- [25] <http://www.ros.org/>. [online], diakses pukul 20.05, tanggal 3 Januari 2019.
- [26] J. M. O'Kane, "A Gentle Introduction to ROS," Independently published, available at <http://www.cse.sc.edu/~jokane/agitr/>, 2013.
- [27] E. Munera, et.al, "Distributed Real-time Control Architecture for ROS-based Modular Robots," IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. IFAC PapersOnLine 50-1 (2017) 11233–11238. <https://doi.org/10.1016/j.ifacol.2017.08.1600>.
- [28] L. Forero, Leonardo & M. Yáñez, José & Ruiz-del-Solar, Javier, "Integration of the ROS framework in soccer robotics: The NAO case," Springer-Verlag Berlin Heidelberg 2014, RoboCup 2013, LNAI 8371, pp. 664–671, 2014. DOI: 10.1007/978-3-662-44468-9_63.
- [29] E. Uslu, F. Cakmak, N. Altuntas, S. Marangoz, M.F. Amasyali, dan S. Yavuz, "An Architecture for Multi-Robot Localization and Mapping in the Gazebo/Robot Operating System Simulation Environment," *Simulation: Transactions of the Society for Modeling and Simulation International*. p 1-10, 2017. DOI: 10.1177/0037549717710098.
- [30] M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev dan E. Magid, "3d Modelling and Simulation of a Crawler Robot in Ros/Gazebo," *MATEC Web of Conferences*, Conference Paper December 2016. DOI: 10.1145/3029610.3029641.
- [31] A. Jalil, "Sistem Kontrol Mesin Penukaran Uang Kertas Rupiah Berbasis Pengolahan Citra Dan Raspberry Pi," *ILKOM Jurnal Ilmiah Volume 10 Nomor 2 Agustus 2018*. DOI: 10.33096/ilkom.v10i2.272.128-135.
- [32] Z. Bachrudin, C.E. Widodo dan K. Adi, "Simulator Input-Output Sistem Kontrol Menggunakan Raspberry Pi," *Youngster Physics Journal*, ISSN: 2302 – 7371. Vol. 6, No. 3, Hal. 272-279, Juli 2017.
- [33] B. Viswanadh dan A. Singh, "Monitor and Control of Remote Appliances using Raspberry Pi through IoT," *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181, Vol. 5 Issue 11, November-2016.
- [34] P Bhaskar Rao dan S.K. Uma, "Raspberry Pi Home Automation with Wireless Sensors using Smart Phone," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, Vol. 4, Issue. 5, May 2015, pg.797 – 803.
- [35] DAO. Turang, "Pengembangan Sistem Relay Pengendalian dan Penghematan Pemakaian Lampu Berbasis Mobile," *Seminar Nasional Informatika 2015 (semnasIF 2015)*, UPN "Veteran" Yogyakarta, 14 November 2015. ISSN: 1979-2328.

