ILKOM Jurnal Ilmiah Vol. 13, No. 1, April 2021, pp. 1-8
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Object motion detection in home security system using the binary-image comparison method based on robot operating system 2 and Raspberry Pi

Abdul Jalil [a, 1, *]; **Matalangi Matalangi** [a, 2]
[a] *STMIK Handayani Makassar, Jl. Adhyaksa Baru No. 1, Makassar 90231, Indonesia*
[1] *abdul.jalil@handayani.ac.id;* [2] *matalangi@handayani.ac.id*
* *Corresponding author*

Article history: Received November 11, 2020; Revised April 16, 2021; Accepted April 16, 2021; Available online April 30, 2021

Abstract

This study aims is to build a home security system based on object motion detection using the Robot Operating System 2 (ROS2) and the Raspberry Pi. ROS2 in this study was used to read and process the camera data and to control the buzzer sound. At the same time, the Raspberry Pi hardware functioned to run ROS2 using the Linux Ubuntu 18.04 operating system. The camera function to read the image and video data could be developed for the input device to control the home security systems based on object motion detection. The method used to detect an object motion was Binary-Image Comparison (BIC). This method works by comparing the value of the binary image object with the binary master and using it as a decision-making algorithm when the camera detected the object movement based on the object colors. The object colors that were detected in this study were red, yellow, green, and blue. Each object color was processed using the OpenCV library in the ROS2 node service. After that, all of the nodes communicated through topics to communicating and exchanging the message data. This study has successfully developed a prototype that can generate a buzzer sound warning to the user when the camera detected the object motion based on the object color.

Keywords: Home Security; Object Motion; ROS2; Raspberry Pi

Introduction

The house is a primary need for humans to live and protect from the weather, natural disasters, and crimes. The rise of crime that occurs in homes has made PIR sensors and other sensors an early warning, when the system detects human movement, besides the use of cameras or CCTV as recording devices has been widely used for home security systems. The importance of security systems in homes has led researchers to develop technology that can be used as an early warning when detecting human movement. The use of RFID sensors as sensors for home security has been developed by [1] and [2]. In this study the researchers used the ATMega328 microcontroller to control the RFID sensor input, while the researchers [2] used Raspberry Pi and wireless ESP8266 as control media. Further, a prototype of a home security system using a combination of sensors and an SMS gateway has been developed by [3]. The researcher in this study has used a magnetic switch sensor as a sensor combination that can be used to detect human movement. Another study has focused on the use of relays as a home security system design which has been developed by [4]. In this study the relay has been used as an input medium to detect human movement when the door of the house is open.

The use of PIR sensors as input media for human motion detection in home security systems has been applied by [5][6] and [7]. In these studies, the researchers [5] and [6] have used Arduino Uno as the control center, while the researcher [7] using the NodeMCU as the control center to control the PIR sensor input. Next, the use of cameras as a home security system tool has been developed by [8] and [9] which has been used only to record video when the sensor detects human movement. Having discussed that, the current study aimed to build a home security system based on object motion detection using Robot Operating System 2 (ROS2) as control software and Raspberry Pi as control hardware. The utilization of cameras as media that can record video can be used as input media for home security control systems using image processing techniques. The image processing technique is to compare the binary image values based on the color of the object using the Binary-Image Comparison (BIC) method. The colors of object motion that can be detected in this study are red, yellow, green, and blue. Researchers have developed a BIC method for detecting object motion based on image processing [10]. In this study, the method was developed for a home

http://dx.doi.org/10.33096/ilkom.v13i1.686.1-8

security system where the output of the system featured a buzzer as an early warning when the system detects the movement of objects or humans.

Method

The home security control system in this study consisted of three core parts, namely input, process, and output. The input of the system used an RGB camera that functioned as a medium to take real-time video and then send the video data to the Raspberry Pi. The function of the Raspberry Pi was as a processing medium whose job is to read video data sent from the camera and then process the video using the OpenCV library and ROS2 software. Furthermore, the output of this system was a buzzer sound that would sound when the camera detected the movement of red, yellow, green and blue objects. **Figure 1** is the picture of the system architecture built in this study.

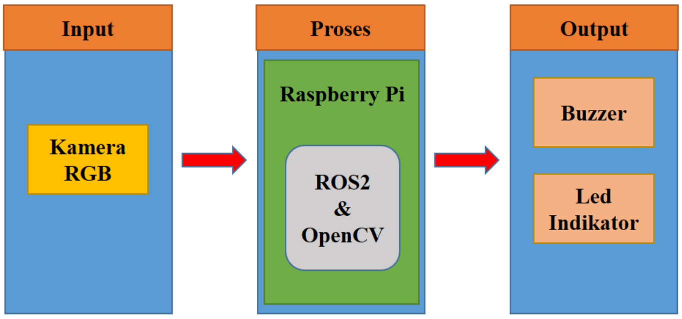


Figure 1. System Architecture

The method used to detect the motion of objects in this study was Binary-Image Comparison (BIC). BIC is a method used to compare two or more images in binary-image form [10]. The use of BIC in image processing has been applied by a study [11]. In this study, the researcher has developed the BIC method with local-dissimilarity quantification using the windowed Hausdorff distance technique. Furthermore, a study [12] has developed a comparison of binary images technique using the image edge line comparison method by modifying the Hausdroff distance which is called the censored Hausdorff distance. In the current study, the BIC method was used as a decision-making algorithm by comparing two binary image values, namely the color binary image value detected by the camera and the master binary image value. The binary image value of an image was processed using a thresholding image technique. In this technique the RGB value of the image was separated from its background so that it could produce a binary image value of 1 or 0 [13]. **Table 1** is an algorithm that runs on the BIC process.

Table 1. Pseudocode algorithm Binary-Image Comparison

Algorithm in the process of Binary-Image Comparison
Step 1. *Subscribe* data message image; Set the value of HSV color (*Low* dan *High*); Set the value of HSV master (*Low* dan *High*); **Step 2.** Read the data message image; *Threshold* the value of HSV color to the image; *Threshold* the value of HSV master; **Step 3. If** threshold_color = threshold_master **then**; Object is not detected; **Else** Object is detected; Buzzer sounds; **Return to step 2;**

Based on the program algorithm that runs on the BIC, it can be observed that there were three steps taken to detect the motion of an object. The first step was that BIC received data or subscribed to data messages sent from the camera node. Then we set the value of Hue, Saturation, Value (HSV) (low or high), to determine the color of the image value of the object, such as red, yellow, green, and blue. After that we set the master HSV value (low or high) which served as a comparison of the value between master binary image and the object binary image. The second step, BIC read the subscribed image data message and threshold the HSV value of the image color and master to be used as a binary image value. After BIC processing the HSV value of the image and the master into a binary image value, the next step was to compare the values of the binary image. In this third step the BIC process, if the value of the binary image of the object image was the same as the value of the master binary image, the system did not detect the object's motion. However, if the object's binary image value was not the same as the master binary image, the system detected the object's motion and then ordered the GPIO pin Raspberry Pi to turn on the buzzer sound as an early warning sign for the home security system.

The image thresholding process that run in this study was processed using the OpenCV library which runs on the ROS2 node process. ROS is an open-source robot meta-operating system which is a framework or middleware for making robot software [14][15]. Within the framework there are software tools, libraries, and packages that can be used to control robot hardware. ROS has been designed to communicate with each other between nodes, namely publisher nodes and subscriber nodes [15]. The ROS community has developed two types of ROS, namely ROS1 and ROS2. The basic difference between ROS1 and ROS2 is in the protocol used to communicate the data, where ROS1 uses TCPROS/UDPROS while ROS2 uses Data Distribution Service (DDS). The type of ROS2 used in this study is ROS2 Crystal Clemmys. ROS2 has many advantages over ROS1 including its real-time data communication system between publisher nodes and subscriber nodes. The publisher node is the node used to send data messages while the subscriber node is the node used to receive data messages sent from the publisher node. In the ROS communication system there are nodes, topics and messages. Node is a computational process that runs on ROS, while topic is the name of the path used to send and receive data messages between publisher nodes and subscriber nodes. The utilization of ROS as robot control software can be developed to control household electronic devices [16]. In this study, ROS2 was used as control software for home security systems that were processed using a Raspberry pi microcomputer.

The Raspberry Pi is a small microcomputer that can perform computing processes like a computer. The type of Raspberry Pi used in this study was the Raspberry Pi 3 B+ which was operated using Linux Ubuntu 18.04 operating system. The function of the Raspberry Pi in this study was as a data processing center to process image/video data using OpenCV and ROS2. In the Raspberry Pi there is a GPIO pin that can be used to control the hardware by providing a logic high (1) or logic low (0) input. The hardware that was controlled via the GPIO pin of the Raspberry Pi in this study was a buzzer and this device was active when the logic high (1) was given from the GPIO pin. The logic was sent to the buzzer to activate high (1) when the home security control system detected the movement of objects or people.

Results and Discussion

The result of this study is in the form of a device or prototype that can be used to provide early warning when the device detects the movement of objects or humans based on their color in the home security control system. This system works based on image/video input sent from the RGB camera to the Raspberry Pi and then processed using communication services between ROS2 nodes. **Figure 2** and **Figure 3** is an image of the result of designing a home security control system based on object motion detection using ROS2 and Raspberry Pi.

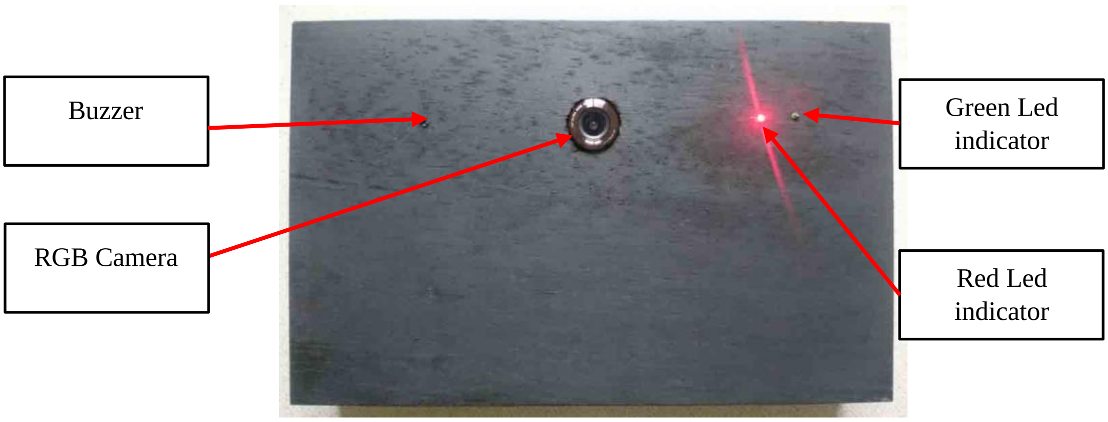


Figure 2. Hardware design results (front view)

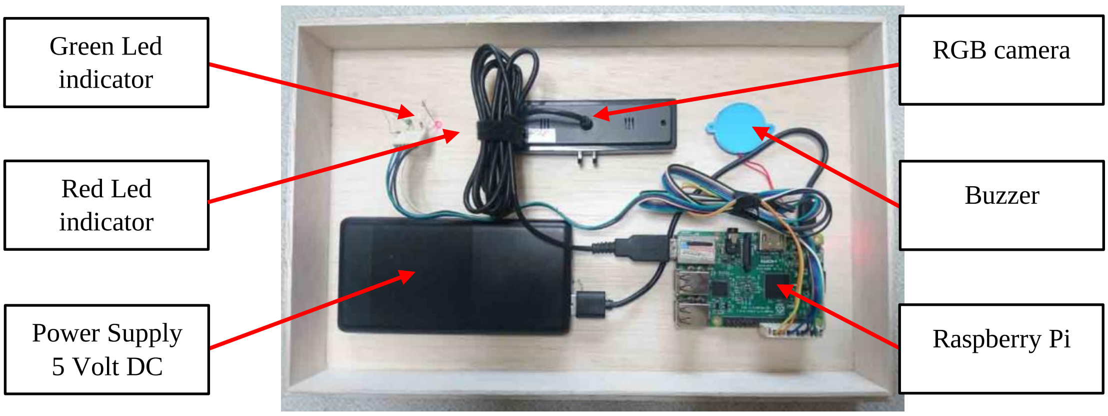


Figure 3. Hardware design results (inside view)

Based on the results of the hardware design, it can be seen that there were several hardware components that were used and connected to each other to build this home security control system device, namely an RGB camera, red and green LED indicators, buzzer, Raspberry Pi, and power supply. The camera the hardware functioned as a device used to detect object motion in the form of streaming video data. The video data was then sent to the Raspberry Pi to be processed using the OpenCV library and ROS2 control software. In this study, the function of the camera as a video recording device was developed as an input device for detecting object motion like a sensor so that it could be applied as a control system tool. The function of the led indicator on the device was to provide visual information when the system was on and when the system detected object motion. The green LED indicator lighted up when the system was activated, then the red indicator LED lighted up when the system detected object motion. Meanwhile, the buzzer function on the hardware was to provide early warning to users in the form of sound when the system detected the movement of objects, people, or objects.

Furthermore, the Raspberry pi functioned as a data processing center microcomputer to process video data sent from the camera using ROS2 and to control the LED indicator light and buzzer via the Raspberry Pi's GPIO pin. The function of the power supply in this study was as a source of 5 Volt DC voltage for all hardware components. In the Raspberry Pi there was a USB port that was used to connect the RGB camera to the Raspberry Pi as well as a GPIO pin that functioned to connect the Raspberry Pi with output devices such as red and green indicator leds, and buzzers. **Table 2** shows the connection between hardware components and the Raspberry Pi.

Table 2. The connection between hardware components and the Raspberry Pi

Input / Output Component	Port / GPIO pin Raspberry Pi
RGB Camera	Port USB
Red led indicator	GPIO pin 14
Green led indicator	GPIO pin 15
Buzzer	GPIO pin 18

In the software that we built; the researcher used a communication system between ROS2 nodes to connect one node to another. In ROS there are nodes, topics, and messages. Node is a computational process that runs on the ROS process, while topic is the path used to connect between the publisher node and the subscriber node. The publisher node is the node used to send data messages while the subscriber node is the node used to receive data messages sent from the publisher node. Data message is data that is processed in the node computing process such as Integer, Float, Boolean and Image data. **Table 3** are the names of nodes, types of nodes, and their functions used in this study.

Table 3. Node names, node types and node functions

Node Names	Node Types	Node Functions
camera	Publisher	Detects objects in the form of video streams and then sends them in the form of message image data to the “red”, “yellow”, “green”, and “blue” nodes.
red	Subscriber / Publisher	Publisher Receives a data message from the camera node then processes it to detect the motion of a red object, the results of the process are then sent to the "hasil_process" (process output) node.
yellow	Subscriber / Publisher	Publisher Receives a data message from the camera node then processes it to detect the motion of a yellow object, the results of the process are then sent to the "hasil_process" node.
green	Subscriber / Publisher	Publisher Receives a data message from the camera node then processes it to detect the motion of a green object, the results of the process are then sent to the "hasil_process" node.
blue	Subscriber / Publisher	Publisher Receives a data message from the camera node then processes it to detect the motion of a blue object, the results of the process are then sent to the "hasil_process" node.
Process output	Subscriber	Subscriber Receives data messages from nodes “red”, “yellow”, “green”, and “blue” to activate the buzzer when the system detects object motion based on its color.

Based on the information in **Table 3**, it can be seen that there were six nodes used to process this home security control system, namely the “camera”, “red”, “yellow”, “green”, “blue”, and “hasil_process” (process output) nodes. All of these nodes were connected to each other and communicated via topics to send data messages. **Table 4** are the names of the types of nodes, topics, and data messages used in this study.

Table 4. Nama node, topic, dan data message

Node publisher	Topic “Data message”	Node subscriber / publisher	Topic “Data message”	Node subscriber
/camera	/image “sensor_msgs/Image”	/red	/hasil_merah “std_msgs/String”	/hasil_proses
		/yellow	/hasil_kuning “std_msgs/String”	
		/green	/hasil_hijau “std_msgs/String”	

Node publisher	Topic "Data message"	Node subscriber / publisher	Topic "Data message"	Node subscriber
		/blue	/hasil_biru "std_msgs/String"	

The main requirement for the publisher node and subscriber node to be able to connect and communicate with each other is that the topic name used to send data messages must be the same between the publisher node and the subscriber node. In table 3 it can be seen that the "camera" publisher node can send a "sensor_msgs/Image" data message to the "red", "yellow", "green", and "blue" subscriber nodes via a topic with the name "Image", as well as other nodes. ROS has a rqt graph service that serves to view the data message communication system between nodes through a graphical topic. **Figure 4** is a picture of the results of data communication between the publisher node and the subscriber node using the ROS2 rqt graph service.

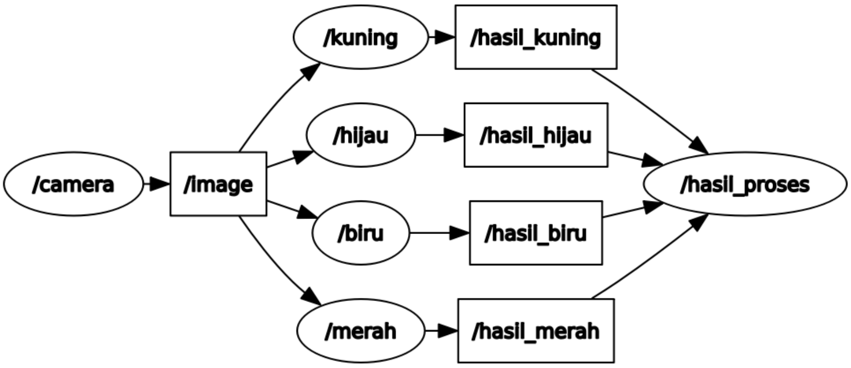


Figure 4. Image display of the ROS node communication system using the rqt graph tool

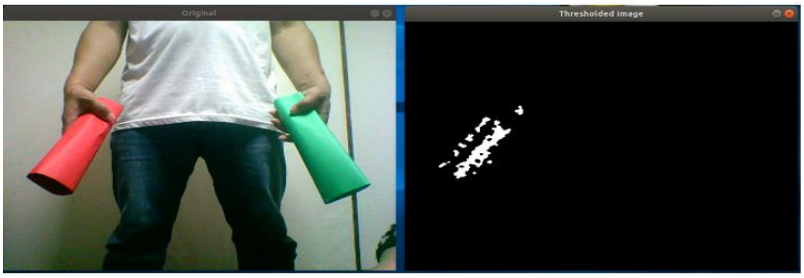
The object motion detection system based on color in this study was processed in the "yellow" node to detect the motion of a yellow object, the "green" node to detect the motion of a green object, the "blue" node to detect the motion of a blue object, and the "red" node to detect the motion of a red object. In each process of these nodes, there was an OpenCV library service that functioned to convert message image data sent from the "camera" node into a binary image, which then was processed using the BIC method as a decision-making algorithm when the system detected object motion. If the value of the detected color binary image was the same as the master binary image value, then the system did not detect the motion of the object, but if the binary image value was not the same as the master binary image, then each color object detection node would send a String data message to the "hasil_process" node to turn on the buzzer and led indicator.

The key to the image processing process to detect object motion in this study was to determine the HSV high and HSV low values to detect object motion based on color and for high and low HSV values in the master. **Table 5** shows a list of HSV high and HSV low configuration values for each color and the master in the image processing process to convert the RGB color values of the image into a binary image.

Table 5. The configuration of HSV high dan HSV low values for each color and master

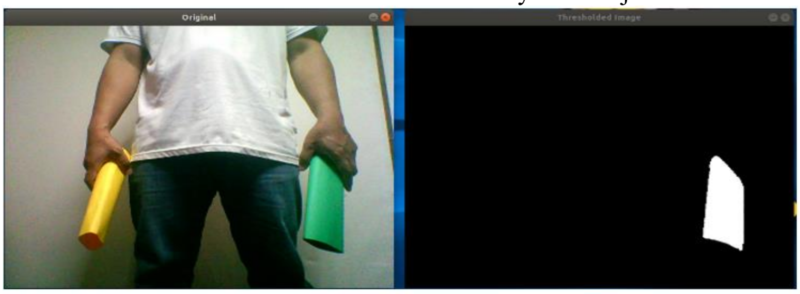
Color	HSV *high* dan *low*	Threshold Value
Red	LowH	170
	HighH	179
	LowS	150
	HighS	255
	LowV	60
	HighV	255
Yellow	LowH	9
	HighH	47
	LowS	153
	HighS	255
	LowV	114
	HighV	255
Green	LowH	60
	HighH	93
	LowS	103
	HighS	255
	LowV	92
	HighV	255

Based on the information in **Table 5**, it can be seen that each detected color had different HSV high and low HSV values. The HSV values were then be processed on the detected image using the image thresholding technique. The results of the thresholding image process then produced binary image values of 0 and 1 in detecting the color of the object's motion. The threshold value shown in **Table 5** was the threshold value used to determine the binary image value. The following is an image of the result of object motion detection based on color using the image thresholding technique.



Gambar 5. Motion detection for a red object

Gambar 6. Motion detection for a yellow object



Gambar 7. Motion detection for a green object

Gambar 8. Motion detection for a blue object

The results of the motion of the red, yellow, green and blue objects shown in **Figures 5 to 8** have been successfully detected by using image processing techniques, where the calibration of the HSV value greatly affected the accuracy of the system when detecting object motion based on color. When the system successfully detected the object's motion based on the color of the object, the system then instructed the buzzer to sound which indicated that the device built had detected an object's movement and provided an early warning to the user as a tool for the home security system. The following are the results of testing the accuracy of the system when detecting object motion between the camera and the object.

Table 5. System Testing Result

Distance (Meter)	Object Color	Result	Buzzer
1	Merah	Detected	On
	Kuning	Detected	On

Distance (Meter)	Object Color	Result	Buzzer
	Hijau	Detected	On
	Biru	Detected	On
	Ungu	Undetected	Off
3	Merah	Detected	On
	Kuning	Detected	On
	Hijau	Detected	On
	Biru	Detected	On
	Ungu	Undetected	Off
5	Merah	Detected	On
	Kuning	Detected	On
	Hijau	Detected	On
	Biru	Detected	On
	Ungu	Undetected	Off
7	Merah	Undetected	Off
	Kuning	Detected	On
	Hijau	Detected	On
	Biru	Detected	On
	Ungu	Undetected	Off
9	Merah	Undetected	Off
	Kuning	Detected	On
	Hijau	Undetected	Off
	Biru	Undetected	Off
	Ungu	Undetected	Off
11	Merah	Undetected	Off
	Kuning	Undetected	Off
	Hijau	Undetected	Off
	Biru	Undetected	Off
	Ungu	Undetected	Off

Based on the results of the accuracy of the testing system in table 5, it can be seen that the device built could detect object motion with an accuracy of up to a distance of 7 meters, where the accuracy of this system was very influential with the configuration of the high and low HSV values set in the software. In addition, the intensity of the light reflected by the object greatly affected the camera when it detected the motion of the object. This influence gave noise to the object when the camera detected an image which was then processed using binary image processing techniques. In table 5 it can also be observed that when the detected object was purple, the system did not detect any object movement. This happened because the HSV high and HSV low values for purple were not entered into the software system making the device could not detect the movement of purple objects.

Conclusion

Detecting object motion in a home security system using Robot Operating System 2 (ROS2) and Raspberry Pi has been successfully applied in this study, where the device built could provide a warning in the form of a buzzer sound when the system detected the movement of objects or humans based on their color. The color motion of objects detected in this study was red, yellow, green, and blue. The color of the object could be detected based on the results of binary image processing techniques that were processed using OpenCV and communication services between ROS2 nodes. The application of the Binary-Image Comparison (BIC) method has been successfully applied as an algorithm for decision makers when the system detected object motion. The device that was built could detect object motion up to 7 meters, where the accuracy of the system greatly affected the configuration of the high and low HSV values, and the intensity of the light reflected by the object. This current study suggests that the motion of objects detected for home security systems can be developed not only in terms of object color but can be developed to detect objects based on the shape of the object pattern. In addition, the setting of HSV high and HSV low values can be dynamically configured to adapt to light noise when objects reflect light.

Acknowledgement

We would like to thank the Ministry of Research, Technology and Higher Education, STMIK Handayani Makassar, and the entire community for their support so that we could complete this Beginner Lecturer Research (PDP) and carried it out properly.

Reference

[1] A. Moborak, I. Sofyan, A. A. Rismayadi, and I. Najiyah, “Sistem Keamanan Rumah Menggunakan RFID, Sensor PIR dan Modul GSM Berbasis Mikrokontroler,” [Home Security System Using RFID, PIR Sensor and GSM Module Based on Microcontroller] Jurnal INFORMATIKA, Vol. 5, No. 1, pp. 137–144, April 2018.

[2] R. L. Singgeta, P. D.K. Manembu, and M. D. Rembet, “Sistem Pengamanan Pintu Rumah Dengan RFID Berbasis Wireless ESP8266,” [Home Door Security System With RFID Based Wireless ESP8266] Seminar Nasional Riset dan Teknologi Terapan 2018 (RITEKTRA 2018), pp. 87–97, August 2018.

[3] F. P. Juniawan, and D. Y. Sylfani, “Prototipe Sistem Keamanan Rumah Menggunakan Kombinasi Sensor dan SMS Gateway,” [Prototype of a Home Security System Using a Combination of Sensors and SMS Gateway] Jurnal TEKNOINFO, Vol. 13, No. 2, pp. 78–83, 2019.

[4] M. Saleh, “Rancang Bangun Sistem Keamanan Rumah Menggunakan Relay,” [Design of Home Security Systems Using Relays] Jurnal Teknologi Elektro, Vol. 8, No. 2, pp. 87–94, May 2017.

[5] M. D. Payana, and N. Husna, “Rancang Bangun Sistem Keamanan Pada Pintu Rumah Dengan Menggunakan Sensor PIR dan Peringatan Dini Melalui SMS Berbasis Mikrokontroler,” [Design of a Security System at Home Doors Using PIR Sensors and Microcontroller-Based SMS Early Warning] Journal of Informatics and Computer Science, Vol. 4, No. 1, pp. 1–5, April 2018.

[6] A. N. Ginting, and M. Amin, “Keamanan Rumah Menggunakan Sensor PIR dan Modul GSM Arduino,” [Home Security Using PIR Sensors and Arduino GSM Modules] Jurnal Teknovasi, Vol. 05, No. 01, pp. 46–53, 2018.

[7] I. Ramadhan, and K. Prawiroredjo, “Prototipe Otomatisasi Pengaman Rumah Berbasis Arduino dan Android,” [Prototype of Arduino and Android-Based Home Security Automation] JETri, Vol. 16, No. 1, pp. 71–86, August 2018.

[8] D. Setiawan, J. E. Candra, and C. E. Suharyanto, “Perancangan Sistem Pengontrol Keamanan Rumah Dengan Smart CCTV Menggunakan Arduino Berbasis Telegram,” [Designing a Home Security Control System With Smart CCTV Using Arduino Based on Telegram] InfoTekjar: National Journal of Informatics and Network Technology, Vol. 4, No. 1, pp. 185–190, 2019.

[9] H. A. Syahidulhaq, Hafiddudin, and S. Aulia, “Sistem Keamanan Berbasis Alarm IP Camera Dengan Passive Infrared Receiver (PIR) Sensor dan SMS Gateway,” [IP Camera Alarm Based Security System with Passive Infrared Receiver (PIR) Sensor and SMS Gateway] Journal of Applied Electrical and Telecommunication, Vol. 3, No. 2, pp. 312–320, December 2016.

[10] Matalangi, and A. Jalil, “Deteksi Gerak Objek Berbasis Pengolahan Citra Menggunakan Metode Binary-Image Comparison,” [Object Motion Detection Based on Image Processing Using the Binary-Image Comparison Method] Luceat Electro Journal, Vol. 6, No. 1, July 2020.

[11] A. Baudrier, G. Millon, F. Nicolier, and S. Ruan, “A fast binary-image comparison method with local-dissimilarity quantification,” International Conference on Pattern Recognition (ICPR'06), 2006. DOI: 10.1109/ICPR.2006.63.

[12] J. Paumard, “Robust comparison of binary images,” ELSEVIER, Pattern Recognition Letters, 1997.

[13] E. Murniyasih, and L. Suryani, “Penerapan Metode Learning Vector Quantization Untuk Identifikasi Penyakit Padi Berdasarkan Bentuk Bercak Daun,” [Application of Learning Vector Quantization Methods for Identification of Rice Diseases Based on Leaf Spots] Luceat Electro Journal, Vol. 6, No. 1, July 2020.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, “ROS: an open-source Robot Operating System ,", 2009.

[15] Y. Maruyama, S. Kato, and T. Azumi, “Exploring the Performance of ROS2,” EMSOFT 16, 2016. DOI: http://dx.doi.org/10.1145/2968478.2968502.

[16] A. Jalil, “Pemanfaatan middleware Robot Operating System (ROS) Dalam Menjawab Tantangan Revolusi Industri 4.0,” [Utilization of Robot Operating System (ROS) middleware in Responding to the Challenges of the Industrial Revolution 4.0] ILKOM Scientific Journal, Vol. 11, No. 1, 45–52, April 2019. DOI: https://doi.org/10.33096/ilkom.v11i1.412.45-52.