ATCSJ
Applied Technology and Computing Science Journal

Open Access

p-ISSN 2621-4458 | e-ISSN 2621-4474

Available online to journal2.unusa.ac.id

UNUSA

S4-Accredited – SK No.200/M/KPT/2021

Journal Page is available to https://journal2.unusa.ac.id/index.php/ATCSJ

ESPCam Control Using Telegram on ESP32 Microcontroller-Based Security Camera Systems

Aris Sudaryanto[1], Dimas Sasongko[2], Agung Kridoyono[3], Rizqi Putri Nourma Budiarti[4], Savilla Tifania Mahadewi[5], Feby Ade Arvianto[6]

[1,5]*Game Technology, Politeknik Elektronika Negeri Surabaya, Indonesia*
Raya ITS Sukolilo Kampus PENS, Surabaya
[1*]sudaryanto@pens.ac.id, [5]savillatifaniam@gt.student.pens.ac.id

[2]*Informatics Engineering, Universitas Muhammadiyah Magelang, Indonesia*
Tidar No. 21, Magelang
[2]dimassasongko@ummgl.ac.id

[3,6]*Informatics Engineering, Universitas 17 Agustus 1945 Surabaya, Indonesia*
Semolowaru No.45, Surabaya
[3]akridoyono@untag-sby.ac.id, [6]febyade98@gmail.com

[4]*Information Systems, Universitas Nahdlatul Ulama Surabaya, Indonesia*
Raya Jemursari 51-57, Surabaya
[4]rizqi.putri.nb@unusa.ac.id

Article history:

Received 30 December 2022
Revised 1 January 2023
Accepted 1 January 2022
Available online 2 January 2022

Keywords:

Security Camera,
ESPCam,
Telegram BOT,
ESP32 Microcontroller,
Camera System.

Abstract

A security camera must be easy to control because users need to be able to monitor and control it even when they are in another place. To accommodate this need, security cameras are generally controlled by a computer, then the data is sent to a web server that can be accessed by users. However, the system is quite complicated because it still requires a computer, as well as the additional cost of renting a website server. In this study, computer functions were replaced with the ESP32 microcontroller, and the website server as well as the application interface was replaced with the Telegram BOT. The ESP32 microcontroller was chosen because its function is almost the same as a computer, while the Telegram BOT can be used free of charge. This study focuses on testing the success rate of the system in responding to commands given via the Telegram BOT, with the type of command connecting the Telegram BOT with ESPCam, turning on or off the LED flash, taking pictures, and the combination. Based on the tests that have been carried out for all existing command combinations, with each test being repeated 25 times, it was found that the success rate of the system reached 84.67%.

1* Coressponding Author

ISSN 2621-4474 (online) 2621-4458 (print) © 2022 The Authors. Published by UNUSA
This is an open access article and licensed under a Creative Commons Attribution-Share Alike 4.0 International License
DOI: https://doi.org/10.33086/atcsj.v5i2.3760.

I. INTRODUCTION

Ease of control is an important aspect of a security camera system. This is because users want to be able to monitor and control security cameras, even if they are in another place. So, security cameras need to be connected to a web server, so that users can monitor and control security cameras. To be able to connect to a web server, a security camera is generally equipped with a computer device that functions to control the camera based on commands from the user, as well as sending images or video from the camera to the web server.

However, a security camera system consisting of a camera, computer and web server becomes less practical because of the large number of devices needed, even requiring an operator. So the ESP32 microcontroller can be used in security camera systems to make it more practical and still able to work according to its main function. The ESP32 microcontroller has the same function as a computer, but with a much smaller size. The ESP32 microcontroller is also equipped with Wi-Fi so that it can be connected to the internet network.

As an alternative to a web server, in this study Telegram BOT is used. The ESP32 microcontroller will receive an image from the security camera, then send it to the user's Telegram account. Users can also send commands to the security camera system via the Telegram BOT.

This research specifically tests the control of the security camera system through the Telegram BOT. Testing is carried out by sending commands to security cameras via Telegram BOT, then observing the suitability of the security camera's response to the commands given. The commands that can be given consist of commands to connect the Telegram BOT with a security camera system, commands to take photos, commands to turn on or turn off the LED flash, and their combinations. Each test combination was repeated 25 times.

II. RELATED WORKS

The security camera system designed in this study consists of three main parts, namely the ESPCam as the camera, the ESP32 microcontroller as the controller, and the Telegram BOT as the interface.

ESPCam or ESP cameras can be used as security cameras, surveillance cameras, detection devices, or other things that require visual images. One that is quite interesting is the use of ESPCam in an emergency tool for women's safety. The tool consists of an ESP32 module, ESPCam, and a GPS module, as well as an internet connection. When in an emergency, the user can press a button that will trigger the device to activate GPS position data, take pictures using ESPCam, and send them to the nearest security (police) [1]. ESPCam can also be used as a surveillance camera, for example in an automatic sprinkler system [2] or as a surveillance camera placed on a mobile device [3]. Apart from that, ESPCam can also distinguish images with certain characteristics, for example, the image of wet tobacco with green characteristics and the image of dry tobacco with brown characteristics [4].

Internet Of Things projects that work in real-time for monitoring or supervision purposes must inevitably be accompanied by an adequate interface. At least a real-time interface is required and can be connected wirelessly to the device. Apart from that, good data storage support is also needed, considering that Internet of Things project data is generally real-time and continuous. Several types of interfaces or data storage methods that may be used include storing in a local database [5] or using a website server that can function as an interface as well as data storage [6] [7], other options can also utilize Internet supporting applications. Of Things like Blynk [8] [9] although with a long delay. Another alternative is to create your own interface design which can also be used as a data storage system, for example using Visual Basic [9], or using other applications [10]. However, in this study, BOT Telegram was used as an interface as well as storage, given its non-paid nature and also the delay which was not too long.

III. METHODS

The whole system can be categorized into three parts, namely ESPCam, ESP32 microcontroller and Telegram BOT. Telegram BOT in this study functioned as an interface for users. Through the Telegram BOT, users can send commands and at the same time see the results of the system's response to the

commands given. For example, if the user gives the command to turn on the LED flash and take a picture, then the user will be able to see the picture taken by ESPCam on the Telegram BOT. Commands sent by users via BOT Telegram are sent wirelessly to the ESP32 microcontroller.

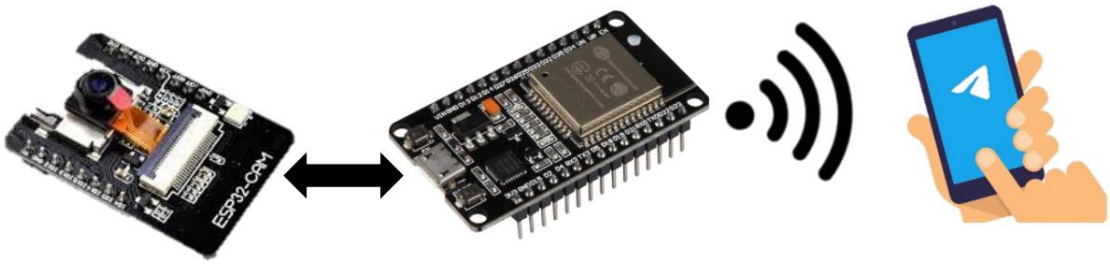

Figure 1. Overall system overview

The ESP32 microcontroller functions as an ESPCam controller and also communicates with the Telegram BOT. ESP32 will receive commands from BOT Telegram, then execute those commands, then send the results back to BOT Telegram wirelessly. For example, if a user sends a command to take a picture, ESP32 will receive the command and then forward it to ESPCam. Images obtained from ESPCam are then sent to ESP32 to be sent back to BOT Telegram.

The third part is the Telegram BOT which functions as the application interface. Through this Telegram BOT users can send commands to the system and can also see the results of the execution of these commands. For example, the user sends a command to the system to take a picture, then after the image has been taken by ESPCam it is then sent to the Telegram BOT so that the user can see the results of the image.

IV. RESULTS AND DISCUSSIONS

This study focuses on testing the appropriateness of the response from ESPCam to commands given by users via BOT Telegram. The commands that can be given consist of commands to connect Telegram BOT with ESPCam, commands to turn on or turn off the LED flash, as well as commands to take pictures, as well as combinations of commands to take pictures and commands to turn on or turn off the flash LED.

The first test is the Telegram BOT connection test to ESPCam. The test is carried out by connecting Telegram BOT with ESPCam wirelessly, then observing whether the Telegram BOT with ESPCam is successfully connected or not. If the Telegram BOT is successfully connected to ESPCam, it will be given a success statement, otherwise, if the Telegram BOT cannot be successfully connected to ESPCam, it will be given a failed statement. The test was carried out 25 times.

Table 1. Testing Telegram BOT connection with ESPCam

No	Expected	Results Obtained Results	Information
1	Connected	Connected	Success
2	Connected	Connected	Success
3	Connected	Connected	Success
4	Connected	Connected	Success
5	Connected	Not Connected	Failed

E-ISSN:*2621-4474, ISSN:2621-4458.*

6	Connected	Connected	Success
7	Connected	Connected	Success
8	Connected	Connected	Success
9	Connected	Connected	Success
10	Connected	Connected	Success
11	Connected	Connected	Success
12	Connected	Connected	Success
13	Connected	Connected	Success
14	Connected	Not Connected	Failed
15	Connected	Connected	Success
16	Connected	Connected	Success
17	Connected	Connected	Success
18	Connected	Connected	Success
19	Connected	Connected	Success
20	Connected	Connected	Success
21	Connected	Connected	Success
22	Connected	Not Connected	Failed
23	Connected	Not Connected	Failed
24	Connected	Connected	Success
25	Connected	Connected	Success
	average percentage of success is		84%

Table 1 displays the results of testing the con Telegram BOT extension to ESPCam. It can be seen that the 25 tests, the results were successful 21 times and failed 4 times in the 5th, 14th, 22nd and 23rd tests. The failed results were obtained because the Telegram BOT was not connected to ESPCam. But overall, the percentage of successful BOT Telegram connections to ESPCam is 84%. The second test is to test the command to take pictures. The test is carried out when the Telegram BOT is connected to ESPCam, then the user gives commands to ESPCam to take pictures (and send them) via the Telegram BOT. Then analyzed the response from ESPCam to the command. If the ESPCam is successful in fetching and sending the image, then the status is success. Conversely, if the ESPCam fails to take a picture and send it, the status is failed. The test was carried out 25 times.

Table 2. Testing the command to take pictures

No	Expected	Results Obtained Results	Information

1	Photos taken	Photos taken	Success
2	Photos taken	Photos taken	Success
3	Photos taken	Photos taken	Success
4	Photos taken	Photos not taken	Failed
5	Photos taken	Photos taken	Success
6	Photos taken	Photos Taken	Success
7	Photos taken	Photos taken	Success
8	Photos taken	Photos taken	Success
9	Photos taken	Photos taken	Success
10	Photos taken	Photos taken	Success
11	Photos taken	Photos taken	Success
12	Photos taken	Photos taken	Success
13	photos taken	Photos taken	Success
14	photos taken	Photos taken	Success
15	Photos taken	Photos taken	Success
16	Photos taken	Photos not taken	Fail
17	Photos taken	Photos taken	Success
18	Photos taken	Photos taken	Success
19	Photos taken	Photos not taken	Fail
20	Photos taken	Photos taken	Success
21	Photos taken	Photos taken	Success
22	Photos taken	Photos taken	Success
23	Photos taken	Photos taken	Success
24	Photos taken	Photos taken	Success
25	Photos taken	Photos taken	Success
	average percentage of success is		88%

Table 2 displays the results of testing the command to take pictures on ESPCam. It can be seen that of the 25 tests, the majority are successful (22 tests). However, the failed results also occurred in tests 4, 16, and 19, the failed status occurred because ESPCam failed to carry out the appropriate command given by the user via BOT Telegram, namely the command to take pictures. Based on the whole test, the average percentage of success reaches 88%.

The third test is testing the command to turn on the flash LED. Testing is carried out when the LED flash is off, then the user gives the command to ESPCam to turn on the LED flash via the Telegram BOT. ESPCam's response to user commands is then analyzed. If the ESPCam successfully fires the LED flash then the status is success. Conversely, if the ESPCam fails to turn on the flash LED, then the status is failed. The test was carried out 25 times.

Table 3. Testing the command to turn on the flash LED

No	Result Expected	Results Obtained	Information
1	Flash On	Flash On	Success
2	Flash On	Flash On	Success
3	Flash On	Flash On	Success
4	Flash On	Flash On	Success
5	Flash On	Flash On	Success
6	Flash On	FlashOn	Success
7	Flash On	Flash On	Success
8	Flash On	Flash On	Success
9	Flash On	Flash On	Success
10	Flash On	Flash Not On	Fail
11	Flash On	Flash On	Success
12	Flash On	Flash Not On	Fail
13	Flash On	Flash On	Success
14	Flash On	FlashOn	Success
15	Flash On	Flash On	Success
16	Flash On	Flash On	Success
17	Flash On	Flash On	Success
18	Flash On	Flash Not On	Fail
19	Flash On	Flash On	Success
20	Flash On	Flash On	Success
21	Flash On	Flash On	Success
22	Flash On	Flash	Success
23	Flash On	Flash Not On	Failed
24	Flash Running	Flash Flashing	Success
25	Flash Flashing	Flash Flashing	Success

average percentage of success is	84%

Table 3 shows the results of testing the command to flash the LED on the ESPCam. It can be seen that of the 25 tests, the majority were successful (21 tests), but failed results also occurred in the 10th, 12th, 18th and 23rd tests. The failed status occurred because ESPCam was unable to carry out the appropriate command given by the user via BOT Telegram, namely the command turn on the LED flash. Based on the whole test, the average percentage of success reaches 84%.

The fourth test is testing the command to extinguish the flash LED. Testing is carried out when the LED flash is on, then the user gives the command to ESPCam to turn off the LED flash via Telegram BOT. ESPCam's response to user commands is then analyzed. If the ESPCam manages to extinguish the flash LED then the status is success. Conversely, if the ESPCam fails to extinguish the flash LED, then the status is failed. The test was carried out 25 times.

Table 4. Testing the command to extinguish the flash LED

No.	Expected	results. Obtained results.	Information
1.	Flash Off	Flash Off	Success
2	Flash Off	Flash Off	Success
3	Flash Off,	Flash Off	Success
4	Flash Go On,	Flash Go On	Success
5	Flash Go On	, Flash Does Not Go On	Fail
6	Flash Go On	Flash Unsuccess	7
Flash	Unsuccessful	Flash Unsuccessful	8
Flash	Unsuccessful	Flash Unsuccessful	9
Flash	Unsuccessful	Flash Unsuccessful	10
Flash	Unsuccessful	Flash Unsuccessful	11
Flash	Unsuccessful	Flash Unsuccessful	12
Flash	Unsuccessful	Flash	Flash
Flash	13s	Unsuccessful	14
Extinguishes	Unsuccessful	Flash	Success
15	Flash Off	Flash Off	Success
16	Flash Off	Flash Off	Success
17	Flash Go On	Flash Go On	Success
18	Flash Go On	Flash Go On	Success
19	Flash Go On	Flash Go On	Fail
20	Flash Go On	Flash Go On	Fail

 E-ISSN:*2621-4474, ISSN:2621-4458.*

21	Flash Go	Flash Go	Success
22	Flash Go	Flash Go	Success
23	Flash Padam	Flash Padam	Success
24	Flash Padam	Flash Padam	Success
25	Flash Padam	Flash Padam	Success
	The average percentage of success is		88%

Table 4 shows the results of testing the command to extinguish the LED flash on the ESPCam. It can be seen that of the 25 tests, the majority were successful (22 tests), but failed results also occurred in the 5th, 19th and 20th tests. The failed status occurred because ESPCam was unable to carry out the appropriate command given by the user via BOT Telegram, namely the command to extinguish flash LEDs. Based on the whole test, the average percentage of success reaches 88%.

The next test, namely tests 5 and 6, is a test for a combination of the command to take pictures and turn on or turn off the LED flash. The fifth test is testing the command combination to take a picture and turn on the flash LED. Testing is carried out when the ESPCam and Telegram BOT are connected and the LED flash is off, then the user gives the command to ESPCam to turn on the LED flash and take pictures through the Telegram BOT. ESPCam's response to user commands is then analyzed. If the ESPCam successfully fires the LED flash and takes a picture, then the status is success. Conversely, if the ESPCam fails to turn on the LED flash and take a picture, the status is failed. The test was carried out 25 times.

Table 5. Testing the command to turn on the flash LED and take pictures

No.	Command		Hope	Result	Status
	LED	**ESPCam**			
1	On	Take picture	LED on, picture taken	Picture taken and Flash Not On	Fail
2	On	Take picture	LED on, picture taken	Picture taken and Flash On	Success
3	On	Take picture	LED on, picture taken	Picture taken and Flash Flashes	Success
4	Flashes	Take picture	LED lights up, picture is taken	Picture is not taken and Flash Flashes	Fails
5	Lights	Take picture	LED lights up, picture is taken	Picture is taken and Flash Flashes	Success
6	Lights	Take picture	LED lights up, picture is taken	Picture is taken and Flash Flashes	Success
7	Flashes	Take picture	LED is on, picture is taken	Picture is taken and Flash is On	Success
8	Flashes	Take picture	LED is on, picture is taken	Picture is taken and Flash is	Successful
9	Lights	Take picture	LED is on, picture is taken	Picture is taken and Flash is	Successful
10	Lights	Take picture	LED is on, captured image Captured	image and Flash On	Success
11	On	Capture g Picture	LED is on, picture is taken	Picture is taken and Flash is On	Success
12	Flashes	Take picture	LED is on, picture is taken	Picture is not taken and Flash is	Failed
13	Flashes	Take picture	LED is on, picture is taken	Picture is taken and Flash is	Successful
14	Lights	Take picture	LED is on, picture taken	Picture taken and Flash Flashes	Success
15	Lights	Take picture	LED is on, picture is taken	Picture taken and Flash Flashes	Success
16	Picture	Take picture	LED is on, picture is taken	is taken and Flash Goes	Success
17	Flashes	Take picture	LED is on, picture is taken	Picture is taken and Flash On	Success
18	On	Take picture	LED is on, picture is taken	Picture is taken and Flash is On	19
Flashes	Take	picture	LED is on, picture is taken	taken and Flash is On	Success
20	Success	Take picture	LED is on, picture is taken	Picture	Successful

21	Flashes	Take a picture	LED lights up,	and Flash On	Success
22	On	Take picture	LED is on, picture is taken	Picture is taken and Flash is Not Flashing	Failed
23	Flashes	Take picture	LED is on, picture is taken	Picture is taken and Flash is On	24
Flashes	Success	Take picture	LED is on, picture is taken	Picture	Successful
25	Turns on	Take a picture	LED lights up, the picture is taken	The picture is taken and the Flash Flashes	Successfully
	Total success percentage is				84%

Table 5 shows the results of testing the command combination to turn on the flash LED and take pictures on ESPCam. It can be seen that of the 25 tests, the majority were successful (21 tests), but failed results also occurred in tests 1, 4, 12 and 22. The failed status occurred because ESPCam was unable to carry out the appropriate command given by the user via BOT Telegram, namely the command turn on the LED flash and take a picture. Based on the whole test, the average percentage of success reaches 84%.

The sixth test is testing the command combination to take pictures and extinguish the LED flash. Testing is carried out when the ESPCam and Telegram BOT are connected and the LED flash is on, then the user gives the command to ESPCam to turn off the LED flash and take pictures through the Telegram BOT. ESPCam's response to user commands is then analyzed. If the ESPCam manages to extinguish the LED flash and take a picture, then the status is success. Conversely, if the ESPCam fails to extinguish the LED flash and take a picture, the status is failed. The test was carried out 25 times.

Table 6. Testing the command to extinguish the LED flash and take pictures

No.	Command		Hope	Result	Status
	LED	ESPCam			
1	Off	Take picture	LED goes off and picture is taken	LED goes off and picture is taken	Success
2	Off	Take picture	LED goes off and picture is taken	LED goes off and picture is taken	Success
3	Off	Take picture	LED goes off and picture takes LED goes off and picture	takes	Success
4	Off	Take picture	LED goes out and picture is taken	Not LED goes off and Picture is taken	Fail
5	Off	Take picture	LED goes off and picture is taken	LED Goes off and Picture is not taken	Fail
6	Off	Take picture	LED goes off and picture is taken	LED Off and Picture taken	Success
7	Off	Take a picture of the	LED off and the picture taken of the	LED Off and the picture taken	Success
8	Off	Take the picture of the	LED off and the picture	taken	Success
10	Off	Take the picture of the	LED off and the picture	taken	Success
10	Off	Take the picture of the	LED off and picture taken	LED Off and Picture taken	Success
11	Off	Take picture	LED off and picture taken	LED Off and GaCaptured	Success
12	Off	Take Picture	LED goes out and Picture Takes	LED Off and Picture Takes	Success
13	Off	Take Picture	LED goes out and Picture Takes	LED Goes Off and Picture Takes	Success
14	Off	Take Picture	LED goes out and Picture Takes	LED Off and Picture Takes	Success
15	Off	Take picture of	LED off and picture taken of	LED Off and Picture taken	Success
16	Off	Take picture of	LED off and picture taken of	LED Not Off and Picture taken	Failed
17	Off	Take picture of	LED turn off and picture taken of	LED Off and Picture taken	Success
18	Off	Take picture of	LED turn off and picture taken	LED Off and Picture taken	Success

 E-ISSN:*2621-4474, ISSN:2621-4458.*

19	Off	Take picture	LED goes out and picture	LED Off and Picture Not taken	Fail
20	Off	Take picture	LED goes out and picture taken	LED Goes off and Picture taken	Success
21	Off	Take picture	LED turns off and picture taken	LED Off and Picture Captured	Success
22	Off	Take picture	LED is off and picture taken	is LED is Not Off and Picture Captured	Failed
23	Off	Take picture of	LED is off and picture is taken	LED is off and picture is taken	Success
24	Off	Take picture	is LED is off and picture is taken	LED is off and picture is taken	Success
25	Off	Take picture	is LED off and picture is taken	LED is off and picture is taken	Success
	Percentage of total success				80%

Table 6 shows the results of testing the command combination to turn off the LED flash and take pictures on the ESPCam. It can be seen that of the 25 tests, the majority were successful (20 tests), but failed results also occurred in tests 4, 5, 16, 19 and 22. The failed status occurred because ESPCam was unable to carry out the appropriate command given by the user via Telegram BOT i.e., the command extinguishes the LED flash and takes a picture. Based on the whole test, the average percentage of success reaches 80%.

Based on all the tests carried out, namely tests 1, 2, 3, 4, 5 and 6, each of which produces an average success rate of 84%, 88%, 84%, 88%, 84%, 80%, the average percentage overall success is 84.67%.

V. CONCLUSIONS AND RECOMMENDATIONS

Based on the tests that have been carried out, it can be concluded that the security camera control system using BOT Telegram is already able to work properly based on all the tests carried out, namely tests 1, 2, 3, 4, 5, and 6, each of which produces an average success rate of 84%, 88%, 84%, 88%, 84%, 80%. So, the overall average success rate reaches 84.67%. The Telegram BOT connection function to ESPCam also works well with a success rate of 84%. The functions for taking pictures and turning on or off the LED flash work well with success rates ranging from 80% to 88%.

Suggestions for development for future research can be carried out in terms of image processing automation, with the hope that the system will be able to automatically detect motion in the future, or detect faces, or recognize faces from users or permitted parties. Methods that may be used include the weighted neighbor method [11] or the pattern recognition method [12] or the matrix mode method [13] or using the color sorter method [14] or the deep learning method [15] where all of these methods exist. possibility to be used in the process of motion detection or detection of the user's face. However, if image improvement is needed first, several methods may be used as references, such as the Simple Method, Adaptive-Gaussian, and Otsu Binarization Thresholding [16].

The next suggestion is considering that this system really needs a good network so that ESPCam can communicate with the Telegram BOT (both communication for receiving/sending commands and for receiving/sending data), it is very important to ensure the network on the system is properly managed. One thing that can be done is to simplify or make it easier for users to make network settings [17], or automate the network configuration backup system so that when the system turns off and starts up again there is no need to repeat the system configuration, but automatically uses the network configuration that was last used [18]. In addition, it is also necessary to first analyze the type of network that will be used, for example VPN or OpenVPN if you have started to focus on network security [19], considering that in several fields, VPN is proven to be implemented [20].

Lastly, considering that this security camera system really needs continuous monitoring, it is very important to use a display that supports it. The display used can be of various kinds, for example in the form of a desktop application designed using Visual Basic [21], or displayed on a website page [22] [23]

[24] or through a display on a mobile-based application, or not even displayed but still stored in an offline database [5].

VI. REFERENCES

[1] S. R. Jadhav, P. S. Patil, V. H. Thigale, M. Andhare, and T. B. Kute, "A ESP based Smart Device For Women Safety Using IOT," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 2, no. 9, 2020.

[2] A. Shukla and R. Diwan, "IOT Based load Automation with Remote Access Surveillance Using ESP 32 CAMand ESP 8266 Module," *Ann. Rom. Soc. Cell Biol.*, pp. 6904–6914, 2021.

[3] B. Fandidarma, R. D. Laksono, and K. W. B. Pamungkas, "Rancang Bangun Mobil Remote Control Pemantau Area berbasis IoT menggunakan ESP 32 Cam," *J. ELECTRA Electr. Eng. Artic.*, vol. 2, no. 1, 2021.

[4] N. R. Setyawan, "RANCANG BANGUN SISTEM PENGERING TEMBAKAU MENGGUNAKAN ESP-32 CAM BERBASIS PENGOLAHAN CITRA," University of Technology Yogyakarta, 2021.

[5] A. A. D. Haris, A. Sudaryanto, and D. H. Sulistyawati, "Uji Fungsional Sistem PengukurSuhu Tubuh Berbasis Arduino Dengan Metode Blackbox Testing," *Informatics, Electr. Electron. Eng.*, vol. 1, no. 1, pp. 31–35, 2021.

[6] A. Sudaryanto, A. E. Wahyudianto, and A. Rizaldi, "Pengujian Stop Kontak Pintar Menggunakan ESP 32," *J. Teknol. Inf. Dan Komun.*, vol. 11, no. 2, pp. 27–30, 2020.

[7] M. F. Yordani and A. Sudaryanto, "Pengujian Sistem Monitoring Listrik Berbasis NodeMCU Menggunakan Blackbox Testing," *Informatics, Electr. Electron. Eng.*, vol. 1, no. 2, pp. 50–55, 2021.

[8] M. S. Aris Sudaryanto, Muhammad Rois Udin, Agung Kridoyono, "DESAIN SISTEM MONITORING SISA PAKAN MENGGUNAKAN SENSOR ULTRASONIK PADA ALAT PEMBERI MAKAN IKAN OTOMATIS," *Jur. Tek. Elektro Fak. Tek. Univ. Halu Oleo*, vol. 7, no. 1, pp. 18–23, 2022.

[9] A. D. Harianto, A. Sudaryanto, A. Kridoyono, and M. Sidqon, "Rancang Bangun Alat Pelindung Jemuran Berbasis Arduino Dengan Sensor Hujan Dan Sensor Cahaya," *Informatics, Electr. Electron. Eng.*, vol. 2, no. 1, pp. 1–5, 2022.

[10] F. Ariani, R. Y. Endra, E. Erlangga, Y. Aprilinda, and A. R. Bahan, "Sistem Monitoring Suhu dan Pencahayaan Berbasis Internet of Thing (IoT) untuk Penetasan Telur Ayam," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 10, no. 2, pp. 36–41, 2020.

[11] A. Sudaryanto, I. K. E. Purnama, and E. M. Yuniarno, "3D Hole Filling Based on Voxel with Weighted Neighbours Interpolation," *Int. J. Eng. \& Technol.*, vol. 8, no. 1.9, pp. 381–386, 2019.

[12] E. Ronando and A. Sudaryanto, "Sistem Pengenalan Pola Huruf Braille Berbasis Audio Menggunakan Metode Na{\"\i}ve Bayes," *J. Comput. Sci. Vis. Commun. Des.*, vol. 3, no. 1, pp. 43–52, 2018.

[13] A. Sudaryanto, "Inpanting Citra Biner Menggunakan Metode Modus Matrik," *JHP17 J. Has. Penelit.*, vol. 3, no. 01, 2018.

[14] A. P. Sanjoto, "RANCANG BANGUN ALAT PENYORTIR BUAH KOPI MENGGUNAKAN RASPBERRY DENGAN METODE COLOUR SORTER," Untag 1945 Surabaya, 2019.

[15] A. A. Rahmadi and A. Sudaryanto, "Visual recognition of graphical user interface components using deep learning technique," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 1, pp. 35–45, 2020.

[16] D. T. Anggraeni, "Perbaikan Citra Dokumen Hasil Pindai Menggunakan Metode Simple, Adaptive-Gaussian, dan Otsu Binarization Thresholding," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 2, pp. 71–77, 2021.

[17] K. D. Perkasa, A. Sudaryanto, and E. D. Hartono, "Pengujian Bandwidth Pada Sistem Setting Bonding Mikrotik Otomatis Menggunakan Library Paramiko," *Informatics, Electr. Electron. Eng.*, vol. 1, no. 1, pp. 1–5, 2021.

[18] M. Afrianto, A. Darwanto, and A. Sudaryanto, "Sistem Backup Konfigurasi Router Secara Otomatis Dengan Shell Script (Studi Kasus: PT Nettocyber Indonesia)." KONVERGENSI, 2019.

[19] A. Sudaryanto and others, "Analisa Konektivitas Jaringan IPSEC Dan OpenVPN Pada Jaringan Berbasis IP Dinamis," *Informatics, Electr. Electron. Eng.*, vol. 1, no. 2, pp. 56–65, 2021.

[20] S. Sumarna and A. Maulana, "Implementasi Virtual Private Network Menggunakan L2TP/IPsec pada BBPK Jakarta," *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 11, no. 2, pp. 90–97, 2021.

[21] A. B. Yunanda, A. Darwanto, and A. Sudaryanto, "PENGENDALI VOLUME AIR UNTUK RUMAH KOS BERBASIS ARDUINO," *J. KONVERGENSI*, vol. 14, no. 2, pp. 54–59, 2018.

[22] R. R. P. Rhamadhan, "Penyiraman Tanaman Tomat Berdasarkan Kelembaban Tanah, Kelembaban Udara dan Suhu Lingkungan Menggunakan Fuzzy Logic Berbasis IOT," Untag Surabaya, 2019.

[23] A. Iga, "Sistem Monitoring Banjir Secara Real Time Berbasis Website," Universitas 17 Agustus 1945 Surabaya, 2020.

[24] R. D. Irsansyah, "Sistem Monitoring Tanaman Cabai Menggunakan Fuzzy Logic Sugeno Dengan Konsep Internet of Things (IoT)," Universitas 17 Agustus 1945 Surabaya, 2020.