Implementation On Smart Parking Using ESP32Cam In Telkom University Premises

1st Anissa Aulia Salsabillah
School of Electrical Engineering
Telkom University
Bandung, Indonesia
anissasalsabillah@student.telkomuniversity.ac.id

2nd Iman Hedi Santoso
School of Electrical Engineering
Telkom University
Bandung, Indonesia
imanhedis@telkomuniversity.ac.id

3rd Arief Indra Irawan
School of Electrical Engineering
Telkom University
Bandung, Indonesia
ariefirawan@telkomuniversity.ac.id

Abstract—The development of the academic community on the Telkom University campus is increasing and causing an increase in the number of riders on campus. The most important facility for a public area is a parking lot. No wonder the parking lot is generally always full, and not a few motorists have trouble finding a parking space. Finding a parking space can be difficult for some drivers as time spent attending lectures is wasted, which causes some drivers to drive around looking for affordable parking spaces. This smart parking uses a system development method, this method is run with an automatic tool that can detect motorized vehicles and send the data to a web server so that drivers can find out which parking spaces can be booked Research on smart parking has also been carried out previously but only uses ultrasonic sensors. The process of sending data from Esp32-cam to the database shows that the QoS on the system made is in the very good category with each average value for the first day, second day, and third-day scenarios, namely throughput 1859.27 KB/s, 4634, 84 KB/s and 3238.37 KB/s, and delay of 65.95 ms, 47.85 ms, and 42.31 ms.

Keywords— Internet of Things, Smart Parking, ESP32-CAM, Node MCU, and Parking.

I. INTRODUCTION

The development of the academic community on the Telkom University campus is increasing and causing an increase in the number of riders on campus. The most important facility for a public area is a parking lot. One of the campus parking problems that visitors cannot ascertain is the limited parking space, especially on days when many visitors arrive.

Finding a parking space can be difficult for some drivers as time spent attending lectures is wasted, which can cause some drivers to drive around looking for affordable parking spaces. This of course can cause traffic jams and long queues in the parking area. And can make some drivers feel disadvantaged in terms of time, and make the parking lot look messy, otherwise, it can make pedestrians uncomfortable.

In this study, smart parking uses the Esp32 Cam integration camera which can detect cars and is investigated in the Telkom University area. Esp32 Cam sends data from sensors to firebase so that data can be viewed in real-time. Implementation on smart parking using Esp32 Cam will provide information as well as detect cars provided for parking users.

II. THEORITICAL REVIEW

A. Internet of Things

Internet of Things or IoT has begun to be widely recognized by humans. In 1999, Kevin Ashton had the idea for the Internet of Things in one of his presentations. The Internet of Things was created to expand internet connectivity by connecting hardware such as machines, equipment, and other physical objects with sensors to obtain data[1].

B. Smart parking

Smart parking is a parking service system that provides information on the avail- ability of parking lots for drivers based on the internet of things. Smart parking is a part of the internet that uses sensors that communicate using remotely via the internet and share information using predefined communication protocols[2].

C. Esp32 Cam

The ESP32-CAM is a complete module with an integrated microcontroller that can be used with many Arduino projects. This module has WiFi and Bluetooth connectivity, an integrated video camera, and a micro sd for storage. [3]

D. Buzzer

The buzzer is an electronic component module that converts electrical signals into sound waves. The buzzer can also be used as an anti-theft alarm signal. The type of buzzer that is often used is piezoelectric, piezoelectric functions transmit data into progress blocks [4]

E. Nodemcu

The nodemcu is a hardware device in the form of an ESP8266 system on a chip made by an espressif system. This device also uses firmware and uses the Lua scripting programming language. The nodemcu is also an open-source platform[5].

F. USB To TTL

To communicate with external devices, a device that has a raw serial USB port is needed, for example in the RS-232 protocol, which has an increased signal voltage that is compatible with TTL. The voltage level used for the microcontroller UART is usually equivalent to 5 V[6].

G. Sd Card

Secure Digital (SD) Card is a flash memory card format used in portable media such as PDAs, digital cameras or smartphones[7]. SD Card has a very good security system and has a size of 32 mm x 24 mm x 2.1 mm. SD Cards come in two smaller sizes, both known as MiniSD and MicroSD or Trans Flash (T-Flash).

H. Firebase

Firebase is a service provided by Google that has various features to develop cloud computing-based infrastructure with very small latency. In addition, firebase is also a real-time database that can store and synchronize data directly using a NoSQL cloud database[8].

I. Wi-Fi

Wi-Fi is an abbreviation of Wireless Fidelity, which has the meaning of a set of standards used for wireless local networks (Wiress Local Area Network-WLAN). Initially Wi-Fi was intended for the use of wireless devices and Local Area Networks (LAN), but now it is more widely used to access the internet. This allows a person with a computer with a wireless card (wireless card) or personal digital assistant (PDA) to connect to the internet using the nearest access point (also known as a hotspot)[9].

J. Wireshark

Wireshark is one of the free as well as open source packet analyzers. This tool is used by Network Administrators as a network problem solver, analysis, software and communication protocol development. Wireshark is also one of the network analyzer applications that are widely used by Network Administrators to analyze network performance and control data traffic on the network[10].

K. Quality Of Service

Quality of service is a method of measuring bandwidth, delay, jitter, and packet loss to show the quality of service in the network[15]. This measurement test is carried out using the Wireshark application and follows the TIPHONE and ITU-T G1010 standards[16]. Quality of service has the objective of influencing one of these four parameters

1. Delay

Delay is the time it takes for data to travel a distance from origin to destination.

2. Packet loss

Packet Loss is a condition that shows how many packets have been lost reaching their destination.

3. Throughput

Throughput is the effective data transfer rate measured in bits per second.

L. C++

C++ programming language is a high-level computer programming language, but C++ is also possible to write low-level programming languages. In coding, C++ is an extension of the C programming language which is classified as a middle-level programming language, which means the C++ programming language has all the features and advantages, one of the advantages is that we can use Assembly programming language in C++ coding and also provides facilities for manipulating level-level memory low [12]

M. Python

Python is a dynamic object-oriented programming language and can be used for a variety of software development. Python has strong support for integration with other programming languages and other tools[11].

N. Yolov3

You Only Look Once (YOLO) is a family of Convolutional Neural Networks (CNN) that achieves optimal results with a single-ended model that can perform object detection in real-time. YOLO v3 is an object detector that is able to detect objects in real-time with a relatively good level of accuracy[14].

III. METHOD

A. Design System

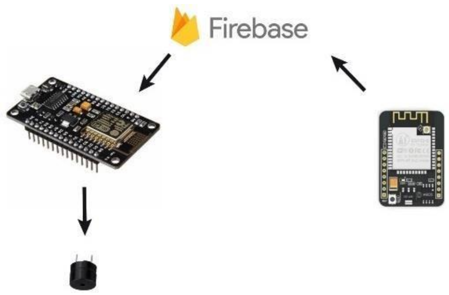

FIGURE 1
DESIGN SYSTEM

This chapter will discuss the design of a smart parking system tool, this system is equipped with an Esp32 Cam which functions to integrate the presence of vehicles through an integrated video camera. Esp32 cam retrieves data using the image classification Yolov3 method, then the data will be sent through the firebase database in real-time, after which the Node MCU will receive data from the firebase and issue an output in the form of a buzzer.

B. Hardware System

The picture below shows the hardware that will be used in completing this final project. The hardware used consists of Esp32 Cam, Node MCU microcontroller, USB To TTL, USB Cable, Buzzer, and Laptop/PC

FIGURE 2
Hardware System

C. Block Diagram of System

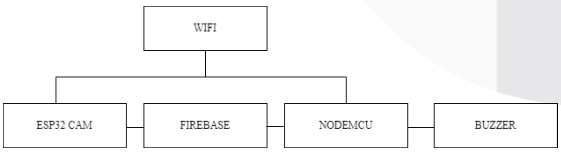

FIGURE 3
Block Diagram

A block diagram of a public parking reservation system. This system consists of several stages, in the first stage the Esp32 Cam and Node MCU must be connected to WiFi to generate an IP address to access the streaming server. Esp32 Cam can send data through the firebase database and the result will be read by Node MCU. Besides being able to read Esp32 Cam, Node MCU can also generate buzzer alarms.

D. Work Flow of The System

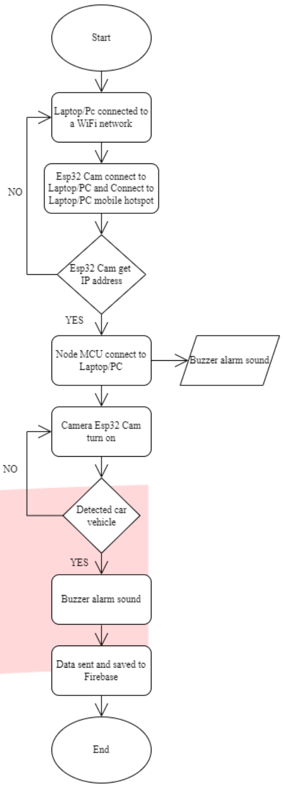

FIGURE 4
Flow Diagram

A flow diagram in system design. The system work process starts with the WiFi network used is already connected to the laptop. Esp32 Cam is connected to the laptop and connected to the laptop's mobile hotspot WiFi connection to generate an IP address. If the IP address cannot be generated then we have to re-assure the WiFi network used. If the IP address has been obtained, connect the Node MCU to the Laptop using a USB cable until the buzzer alarm sounds indicating that the Node MCU can be connected. After that, the program can be run until the Esp32 Cam turns on. Esp32 Cam can detect when there is a car on the land and generate a buzzer alarm. If there is no car vehicle then Esp32 Cam cannot detect and continue the next process.

IV RESULT AND DISCUSSION

A. Tools Design Result

The hardware used in this final project consists of Esp32 Cam, a Node MCU microcontroller, USB To TTL, a USB cable, and Buzzer. The Esp32 Cam is pointed at the vehicle to detect the car's vehicle.

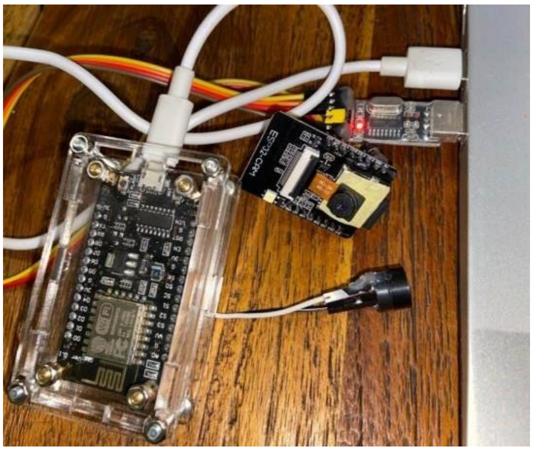

FIGURE 5
TOOLS DESIGN RESULT

B. Monitoring System Workflow using Esp32 Cam

Esp32 Cam monitoring system workflow as a control and detection system for upcoming vehicles/cars. In this process, the workflow of the tool is explained starting from connecting the Esp32 Cam and Node MCU with the same Wi-Fi as the device.

1. The user connects the Esp32 Cam and Node MCU to the same wi-fi as the laptop so that they can be connected

FIGURE 6
WIFI

2. Open visual code, and wait until task execution appears. Then copy the IP Address.

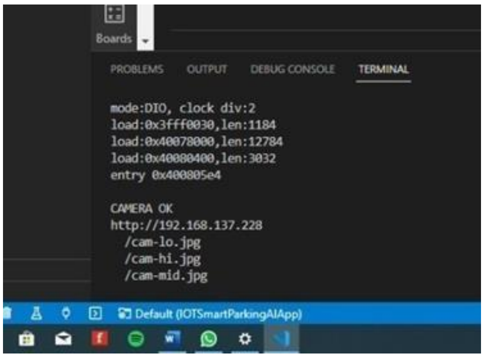

FIGURE 7
IP ADRESS

3. After task execution appears press the button on Esp32 Cam until 'camera OK'

FIGURE 8
ESP32 CAM

4. Open IDLE, enter the copied IP address.

FIGURE 9
IDLE

5. Connect the Nodemcu USB cable to the device

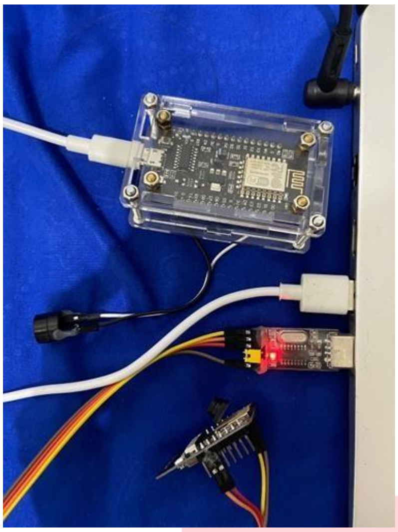

FIGURE 10
ESP32 CAM AND NODE MCU PUT TOLAPTOP

6. Esp32 Cam can detect objects well, if it is detected that there is a vehicle / car that enters the buzzer alarm will sound.

FIGURE 11
DETECTION CAR

B. Hardware Test

No.	Hardware Function	Status
1.	Esp32 Cam is connect andcan send data to database	OK
2.	NodeMCU is connect and can receive data from database	OK
3.	Buzzer can work when thecar is detected or caught on camera	OK
4.	USB To TTL can connected Esp32 Cam to device	OK

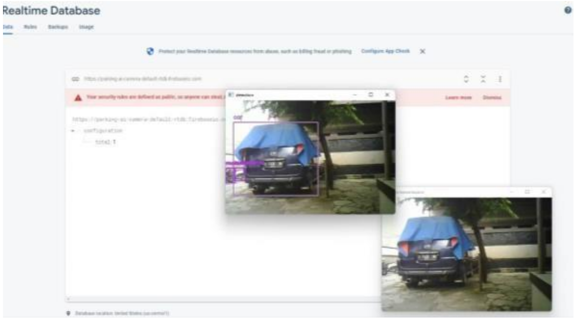

FIGURE 12
REAL-TIME DATABASE

C. Test Function/Performance detection Esp32 Cam

This test is carried out whether the system is successfully used to detect vehicles that appear in several conditions. This test is carried out both in hardware and in software.

1. Test Result in Dim Conditions

The results of this test are carried out in dim conditions, to be able to find out when the conditions are dim and whether the tool can work properly. This test has been carried out and it is stated that when the light is low, the Esp32 Cam cannot detect objects properly, the intensity of the lack of light is one of the problems that can affect object detection. The following are the results of the appearance of the car vehicle test object when the condition is dim

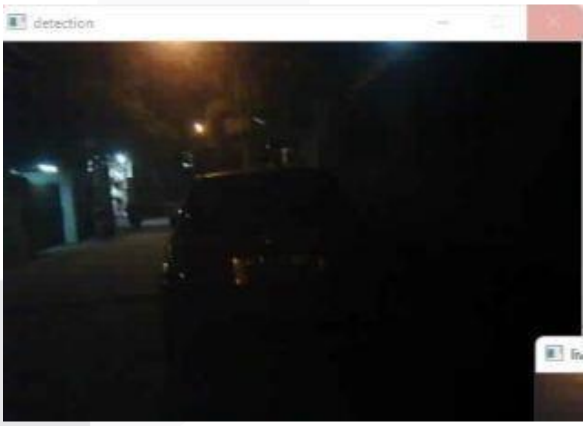

FIGURE 13
DIM CONDITION

2. Test result car speed comes and stop

The results of this test are carried out when the car comes and goes, when the car arrives it takes 05.55 seconds to detect a car and the buzzer alarm sounds. when the car leaves the place it takes 05.14 seconds for the alarm to stop sounding. The results of this test were carried out using a stopwatch. Here is a picture of the results from testing the car coming and going or stopping.

Test results on the car that came

FIGURE 14
TEST RESULT CAR COME

Test results on cars that go

FIGURE 15
TEST RESULT CAR GO

3. Maximum Distance Test Results

The results of this test are carried out at the farthest point of the object with the Esp32 Cam camera, the maximum distance of the object that can be traveled is about 29 meters. At a distance of 30 meters, Esp32 Cam cannot detect cars. This affects the detection so that at a distance the object/car vehicle cannot be detected properly. The following is a picture of the maximum distance test results

FIGURE 16
MAXIMUM DISTANCE TEST RESULT

Distance	Description
3.5 Meter	Detected
4 Meter	Detected
4.5 Meter	Detected
5 Meter	Detected
5.5 Meter	Detected
6.5 Meter	Detected
7 Meter	Detected
8 Meter	Detected
9 Meter	Detected
9.5 Meter	Detected
20 Meter	Detected
30 Meter	Not Detected

4. Esp32 Cam to device test result

The results of the Esp32 cam to device test are carried out when USB To TTL is connected to the device it takes 06.79 seconds to connect to wi-fi, this can also be caused by the stability of the wi-fi network connection used. Therefore, it takes a stable wi-fi network to get maximum result.

FIGURE 17
ESP32 CAM TO LAPTOP RESULT

D. Parameter Testing Quality of Service

1. Throughput Test

Throughput testing is carried out with 3 scenarios, the first and second scenarios are carried out on weekdays. while the third scenario is carried out on weekends. Each scenario experimented with 30 repetitions.

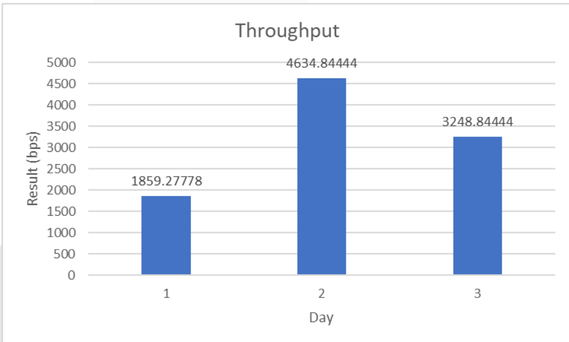

FIGURE 18
THROUGHPUT TEST

In throughput testing using a WiFi network, on the first day the results were 1859.27 bp/s. On the second day, the result was 4634.84 bp/s, and on the third day, the result was 3248.37 bp/s. From the results of the throughput test for 3 days, the value obtained is in the very good category, according to the ITU-T G1010 standard. The throughput results are variable and can be caused by the quality of the network, the number of network users, and the type of network used.

2. Delay Test

The delay test is carried out using 3 scenarios, the first and second scenarios are carried out on weekdays. while the third scenario is carried out on weekends. Each scenario experimented with 30 repetitions.

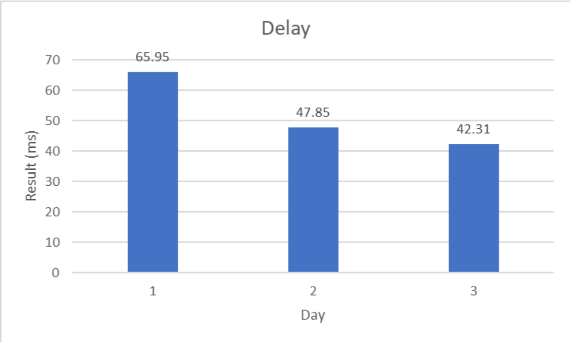

FIGURE 19
DELAY TEST

In the Delay test using a Wi-Fi network, on the first day the results were 65.95 ms. On the second day, the result was 47.85 ms, and on the third day, the result was 42.31 ms. From the results of the delay test for 3 days, the value obtained is in the very good category, according to the ITU-T G1010 standard. Throughput results are variable and can be caused by network quality, and protocol performance which takes time to process posts and requests between the device and the server.

E. Subjective Testing (Quality os Experience/QoE)

Subjective testing was carried out to pass a poll of respondents regarding the assessment of the quality of live streaming video on a web server. This assessment is in the form of the success rate of live streaming video to the quality of live streaming video on a web server. Here are the test results:

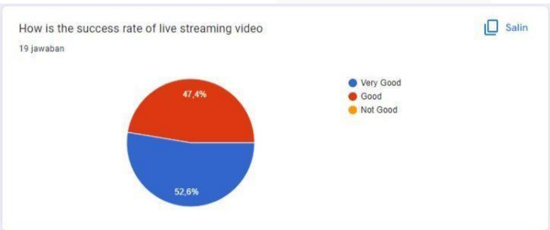

FIGURE 20
THE SUCCESS OF VIDEO STREAMING

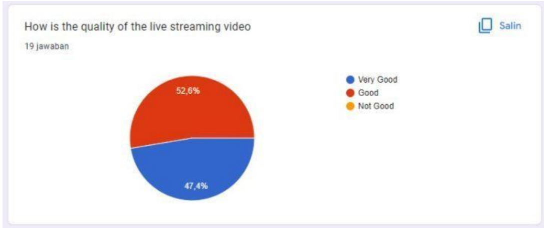

FIGURE 21 The quality of live streaming

The assessment that has been made on this Google form is an assessment of the success rate of live streaming video to the quality of live streaming video on a web server. This assessment was followed by 19 respondents. Proving that the tool is feasible to use. It can be seen that the best assessment is as many as 10 respondents out of 19 with a very good assessment followed by 9 respondents who chose well. It can be concluded that this smart parking deserves acceptance in the surrounding Telkom University area to be produced as a whole.

V. CONCLUSION

A. Conclusion

In testing and analyzing the Esp32 Cam monitoring, the following conclusions and suggestions were obtained:

1. The implementation of the Esp32-Cam for Smart Parking has succeeded in detecting objects according to the plan with test results in both bright and dim conditions.
2. This Esp32-Cam can detect car objects up to a distance of <9.5 meters.
3. Tests conducted on the speed of the car coming and stopping takes 05.55 seconds to detect an oncoming vehicle, and if the vehicle leaves the camera object, it takes 05.14 seconds for the alarm to stop sounding.
4. Tests conducted when the Esp32-Cam is connected to the device take 06.79 seconds to connect to wifi, this can be due to the stability of wifi connectivity and requires a large internet connection.
5. Assessment on the success rate of video streaming subjectively has 10 respondents from 19 respondents with 52.6% in the very good category, and on the quality of video streaming has 9 respondents from 19 respondents with 52.6% in the good category.
6. Throughput testing using a Wi-Fi network, on the first day the results were 1859.27 ms. On the second day, the result was 4634.84 ms, and on the third day, the result was 3248.37 ms. From the results of the throughput test for 3 days, the value obtained is in the very good category, according to the ITU-T G1010 standard.

In the Delay test using a Wi-Fi network, on the first day the results were 65.95 ms. On the second day, the result was 47.85 ms, and on the third day, the result was 42.31 ms. From the results of the delay test for 3 days, the value obtained is in the very good category, according to the ITU-T G1010 standard

B. Suggestion

From the results of this test, it can be suggested that this research can be further developed as follows:

1. To improve the quality of the video can be done using Raspberry pi.
2. To improve the quality of detection can be added using other sensors, such as ultrasonic sensors and so on.

REFERENCE

[1] INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU MENGGUNAKAN RASPBERRY PI BERBASIS MOBILE , Prodi Teknik Sipil , Fakultas Teknik , Universitas Udayana , 2015. Avail- able: https://media.neliti.com/media/publications/283803-internet-of-things- iot-sistem-pengendali-c98bdddd.pdf

[2] Hernoko, M. G., Wibowo, S. A., & Vendyansyah, N. (2021). PENERAPAN IoT (Internet of Things) SMART PARKING SYSTEM DAN PENDETEKSI KEBAKARAN DENGAN FITUR MONITORING. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *5*(1), 261-267.

[3] Riskyapriliani, N., Darlis, D., & Hartaman, A. (2021). Perancangan Pendeteksi Golongan Darah Dan Rhesus Berbasistensorflow Menggunakan Esp32-cam. *eProceedings of Applied Science*, *7*(6).

[4] Nduru, S., Al Hafiz, A., & Pane, D. H. (2022). Implementasi Metode Fuzzy Berbasis Internet Of Things (IoT) Untuk Peringatan Dini Banjir. Jurnal Sistem Komputer Triguna Dharma (JURSIK TGD), 1(1), 26-33.

[5] Nurhuda, A., Harpad, B., & Mubarak, M. S. A. (2019). Kendali lampu menggunakan perintah suara berbasis node mcu. Sebatik, 23(1), 77-83.

[6] Yosephat Suryo Susilo, Hartono Pranjoto, Albert Gunadhi, "Sistem Pelacakan Dan Pengamanan Kendaraan Berbasis GPS Dengan Menggunakan Komunikasi GPRS", Jurnal I lmiah Widya Teknik, Vol. 13 --- No. 1 --- 2014

[7] Pristian, D. H., & Mulyana, D. I. (2022). Optimasi Pengunduhan Anime Jepang Bersubtitle Indonesia dengan Metode Restful API dan Firebase Cloud Messaging. Jurnal Pendidikan dan Konseling, 4(4), 4877-4885.

[8] Priantama, R. (2015). Efektivitas wifi dalam menunjang proses pendidikan bagi lembaga perguruan tinggi (Studi kasus terhadap mahasiswa pengguna di lingkungan Universitas Kuningan). Cloud Information, 1(1).

[9] Gunawan, I. (2021). Analisis Keamanan Wifi Menggunakan Wireshark. JES (Jurnal Elektro Smart), 1(1), 7-9.

[10] Arifin, J., Zulita, L. N., & Hermawansyah, H. (2016). Perancangan Murottal Otomatis Menggunakan Mikrokontroller Arduino Mega 2560. Jurnal Media Infotama, 12(1).

[11] Fitri, K. R. R., Rahmansyah, A., & Darwin, W. (2017). Penggunaan Bahasa Pemrograman Python Sebagai Pusat Kendali Pada Robot 10-D. In 5th Indonesian Symposium on Robotic System and Control (pp. 23-26).

[12] Ramadhana, I., & Sujatmiko, B. (2018). PENGEMBANGAN APLIKASI KAMUS BAHASA PEMROGRAMAN C++ BERBASIS ANDROID UNTUK MENINGKATKAN KOMPETENSI KOGNITIF MATA KULIAH STRUKTUR DATA. IT-Edu: Jurnal Information Technology and Education, 3(01).

[13] Dewi, L. J. E. (2010). Media Pembelajaran Bahasa Pemrograman C++. Jurnal Pendidikan Teknologi dan Kejuruan, 7(1).

[14] Riyadi, A. S., Wardhani, I. P., Wulandari, M. S., & Widayati, S. (2022). Perbandingan Metode ResNet, YoloV3, dan TinyYoloV3 pada Deteksi Citra dengan Pemrograman Python. PETIR, 15(1), 135-144.

[15] Muharram, A. A. Analisis Quality Of Service Jaringan Wireless Virtual Local Area Network Pada UIN Syarif Hidayatullah Jakarta (Bachelor's thesis, Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).

[16] Bramantya, A. W. U. (2015). TA: Implementasi VPLS pada Jaringan MPLS Berbasis Mikrotik (Doctoral dissertation, Institut Bisnis dan Informatika Stikom Surabaya).