Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Physical Distancing Detection System using OpenCV Based on Raspberry Pi4

*Mohammad Rafiqul Farzan**

Department of Electronics Engineering/ Electronics Engineering Diploma Program, Cilacap State Polytechnic, Indonesia

Correspondence: E-mail: mohammadrafiqulfarzan@gmail.com

ABSTRACT

The Covid-19 pandemic is a disease that is ravaging all over the world, one of which is in our country, namely Indonesia. At this time, officers are still monitoring physical distancing by giving direct warnings to physical distancing violators. Monitoring system that aims to detect physical distancing violations using detection based on biometric identification using faces. The monitoring system uses a Mixio F10 webcam with 1080p resolution which is processed with a Raspberry Pi 4 microcomputer with specifications of 4 GB RAM and an SD Card to store the Raspbian operating system. The distance detection program between faces uses the Python programming language and requires the Open CV library. USB Speaker as an output medium in the form of sound as a warning against physical distancing violators. The tool detects the distance between adjacent faces equipped with a voice indicator if the distance between faces exceeds a predetermined distance with a maximum distance of 3 meters and the level of accuracy of face detection depends on the distance of the face to the camera.

ARTICLE INFO

Article History:
Submitted/Received 24 Jul 2022
First Revised 11 Aug 2022
Accepted 17 Sept 2022
First Available online 01 Oct 2022
Publication Date 01 Oct 2022

Keyword:
Covid-19,
Physical distancing,
Raspberry Pi 4,
USB Speaker,
Webcam Mixio F10.

1 INTRODUCTION

The implementation of physical distancing rules has become a necessity in various situations, especially to prevent the spread of dangerous and deadly viruses [1]. The global COVID-19 pandemic has demonstrated the importance of maintaining a safe distance between individuals to protect collective health and safety. However, in public places, there are often violations of the established physical distancing rules, resulting in the failure to achieve the intended objectives of these regulations. Individuals who disregard these rules increase the risk of virus transmission, threatening individual health and impeding pandemic control efforts.

To address the issue of physical distancing violations in public places, an effective and efficient surveillance method is needed. One solution that can be employed is the use of an automated physical distancing detection system. This system can help replace the role of personnel in monitoring public activities, reduce direct human involvement, and minimize the risk of virus transmission.

The physical distancing detection system is designed using a combination of a webcam and Raspberry Pi. By utilizing face detection algorithms like Haar cascade [2][3], the system can accurately detect the distance between individuals. This face detection method is based on visual pattern recognition, enabling the system to identify and measure the distance between detected faces. The system generates audio output through a speaker, indicating whether there is a physical distancing violation between individuals or not.

Figure 1. shows the example of the Raspberry Pi. The utilization of Raspberry Pi as the core component of this detection system provides advantages in terms of capability and flexibility. Raspberry Pi is a single-board microcomputer module with the size of a credit card. It features digital input and output ports similar to microcontroller boards, but with broader capabilities. Raspberry Pi can be used to run office programs, computer games, and serve as a media player, including high-resolution videos. Its advantages over other microcontroller boards include the ability to connect to screens such as TVs or PC monitors, as well as USB connectivity for keyboards and mice.

Figure 1. Raspberry Pi.

In addition to Raspberry Pi, the webcam plays a crucial role in the physical distancing detection system [4-13]. **Figure 2.** shows the example of the webcam. A webcam is an external device that captures images or video and optionally records audio through a microphone. The webcam is controlled by a computer or computer network. The images captured by the webcam are displayed on a monitor screen, enabling the quick and efficient use of the physical distancing detection system [14].

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

Figure 2. Webcam.

The Haar cascade algorithm used in the physical distancing detection system is part of OpenCV (Open Computer Vision), a widely used API library for computer image processing. OpenCV provides various algorithms and functions for object recognition, face detection, object tracking, and various other image processing applications [15-22]. By utilizing OpenCV, developers can easily integrate image processing capabilities into their programs or systems.

Python programming language is the suitable choice for developing this physical distancing detection system. Python is an interactive computer programming language with versatile applications. The design philosophy of Python focuses on code readability or scripts. Compared to other programming languages, Python offers good capabilities, combining clarity and easily understandable syntax. Python programming language also comes with a functional and extensive standard library. In a short period, Python has become a popular and competitive programming language.

By combining all these components, the physical distancing detection system can become an effective and efficient solution for monitoring public activities in crowded places. In its implementation, the system can be deployed in various locations such as shopping centers, train stations, airports, or other public places with a high risk of crowds. The system will automatically detect the presence of individuals and measure the distance between them. If a physical distancing violation is detected, the system will produce an alert sound to inform individuals to maintain a safe distance.

In the current pandemic era, the use of technology is crucial in combating virus transmission. The utilization of the physical distancing detection system with Raspberry Pi, webcam, and Haar cascade algorithm is a tangible example of leveraging technology to ensure compliance with physical distancing rules. Thus, the objectives of these rules can be better achieved, and the risk of virus transmission can be reduced.

2 METHODS

2.1 Block Diagram

From this block diagram, the overall working principle of the circuit can be understood. It simplifies the process of designing and constructing the device, resulting in a system that aligns with the pre-designed specifications. The block diagram of the system can be seen in **Figure 3.** below:

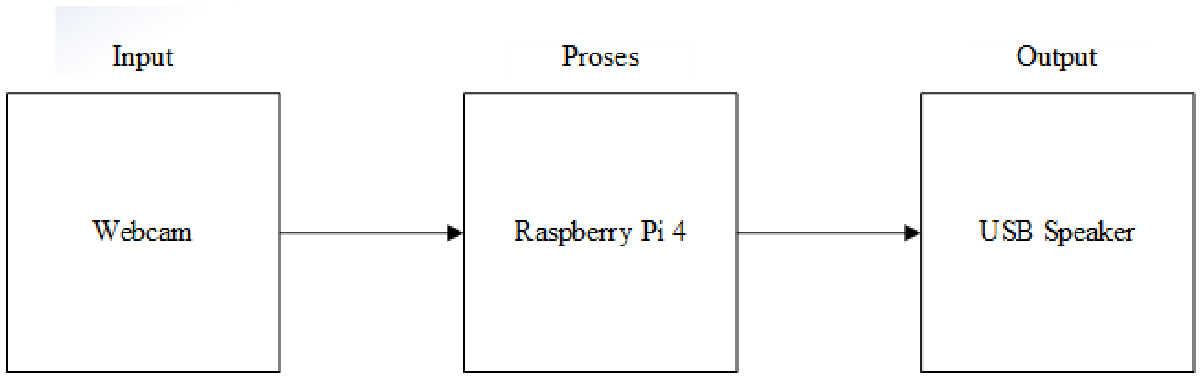

Figure 3. Block Diagram.

Based on the block diagram in **Figure 3.** it can be explained that this system utilizes Raspberry Pi 4 as the controller, Webcam as the input for detecting the distance between objects, and USB Speaker as the output that will provide audible warnings if the distance between objects is too close.

2.2 Flowchart

Flowchart is a standard way to depict a process. Each step in the system is represented by a symbol, and the flow of each step is indicated by lines with arrows. Flowcharts are created to simplify the recognition of the program flow embedded in the controller. In general, the flowchart consists of the controller initialization, where the controller recognizes the embedded program and its contents, followed by waiting for available commands to execute the program and provide instructions to the existing actuators. The flowchart of the system can be seen in **Figure 4**.

In **Figure 4**. the flowchart of the device's working system is illustrated, which can be explained as follows: When the system is running, the Webcam will initialize and capture an image (camera capture). The captured image will then go through the Haar-cascade process (object detection method). The output generated from the Haar-cascade process is in the form of coordinate points of an object. These coordinate points are automatically checked, and if the distance between the object coordinates is too close, a warning will be sounded through the USB Speaker.

2.3 Mechanical Design/Hardware Design

The materials used in the construction of the physical distancing violation detection device include a camera and a USB Speaker. **Figure 5.** represents the form of the mechanical design framework created as follows:

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

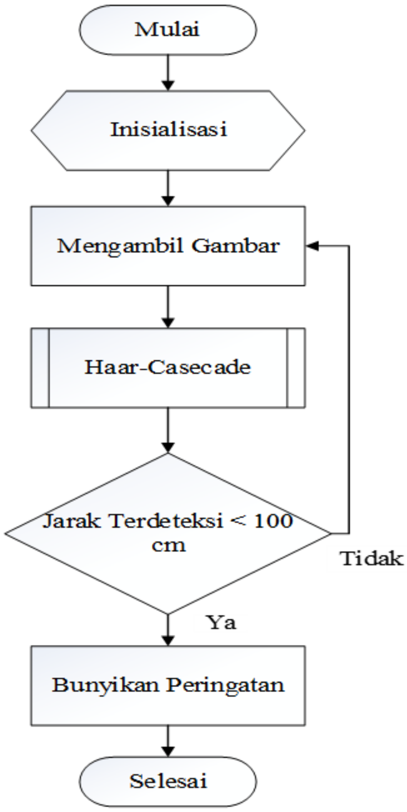

Figure 4. Flowchart.

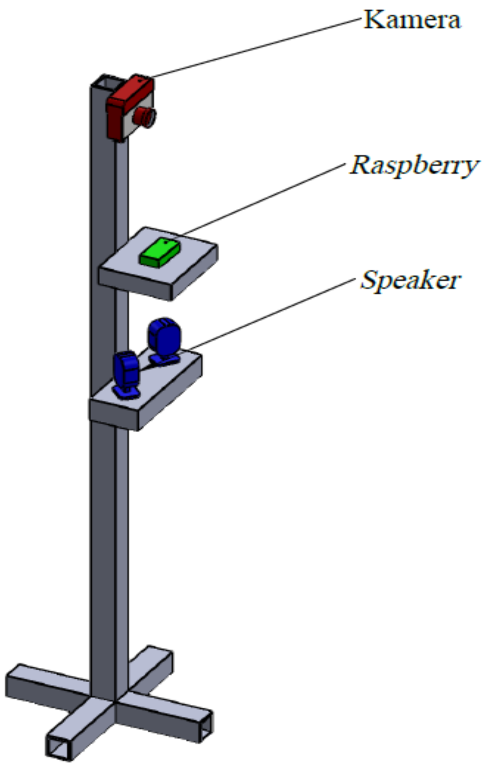

Figure 5. Mechanical Design.

The design of the device includes the following specifications: the overall height from the base to the camera is 150 cm, the height from the base to the Raspberry Pi is 110 cm, and the height from the base to the Speaker is 80 cm. The device is constructed using Holo Iron,

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

ensuring durability and stability. These measurements and materials are carefully chosen to create a robust and functional structure for the physical distancing violation detection tool. The use of Holo Iron provides strength and support to the components, ensuring the device's stability during operation. This mechanical design plays a crucial role in the overall effectiveness and reliability of the device, allowing it to effectively detect and address instances of physical distancing violations.

2.4 Electrical Circuit Planning

The electrical design is carried out using the Fritzing software. The design encompasses the entire measurement information system from input to output, and it can be viewed in **Figure 6.** as follows:

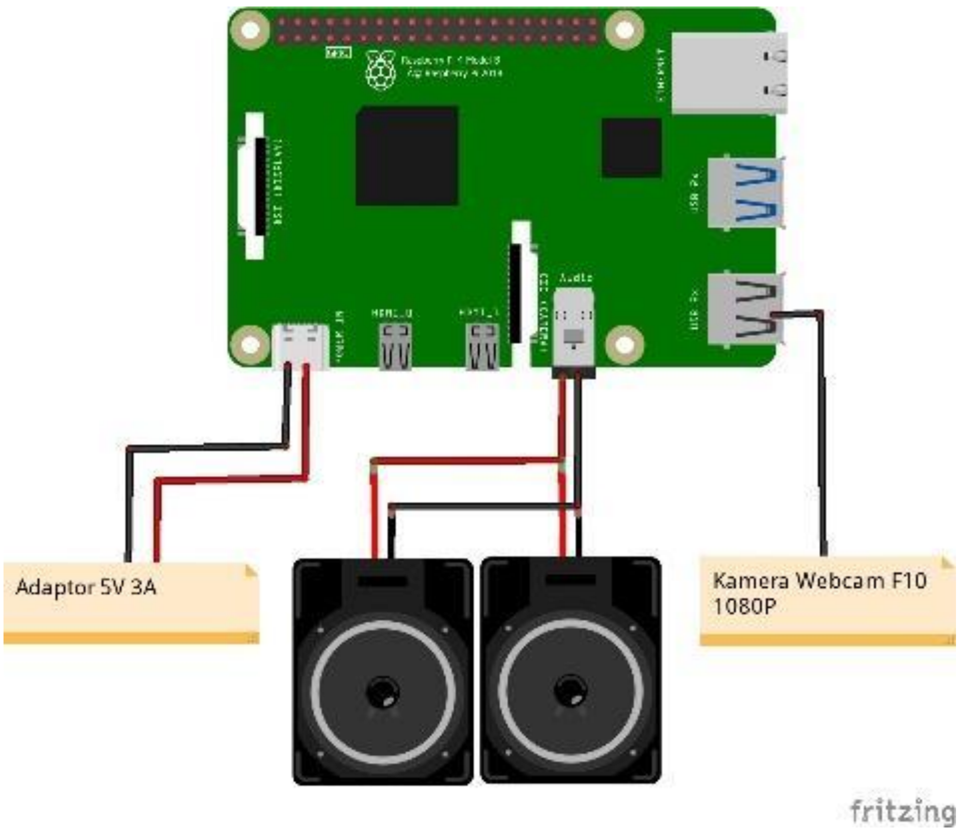

Figure 6. Electrical Circuit.

Overall System Design involves using the Raspberry Pi 4 controller to process the captured images from the camera. The images are then processed for their coordinate points, and if the coordinate points of objects or images are close to each other, the USB Speaker will emit a warning sound.

2.5 Design of Face Distance Detection Program

The programming for face distance detection within the Raspberry Pi is depicted in the program shown in **Figure 7**. as follows:

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

Distance

```
def distance(x1, y1, x2, y2):
        return int(np.sqrt(((y1-y2) ** 2) + ((x1-x2) ** 2)))

    for a in range(len(results)) :
        for b in range(a + 1, len(results)) :
            jarak = Calculation.distance(results[a][2], results[a][3], results[b][2], results[b][3])
            if jarak > 100 :
                cv2.line(frame, (results[a][0], results[a][1]), (results[b][0], results[b][1]), (0,255,0), 2)
            else :
                if time.time() > lastPlay + 3 :
                    subprocess.Popen(['cvlc', '2.mp3'])
                    lastPlay = time.time()
                cv2.line(frame, (results[a][0], results[a][1]), (results[b][0], results[b][1]), (0,0,255), 2)
```

Figure 7. Face Distance Detection Program

From the sample acquisition experiment, the actual distance results can be observed in centimeters (cm) and in pixel measurements, indicating the camera's viewing range as shown in **Figure 8**.

Total lebar jangkauan kamera

Total Lebar Terukur (cm)	Ukuran Wajah Dalam (pixel)
50	221
60	198
70	167
80	149
90	135
100	128
110	114
120	106
130	100
140	93
150	89
160	82
170	81
180	75
190	74
200	69
210	66
220	65
230	61
240	59
250	59
260	55
270	54
280	53
290	51
300	50

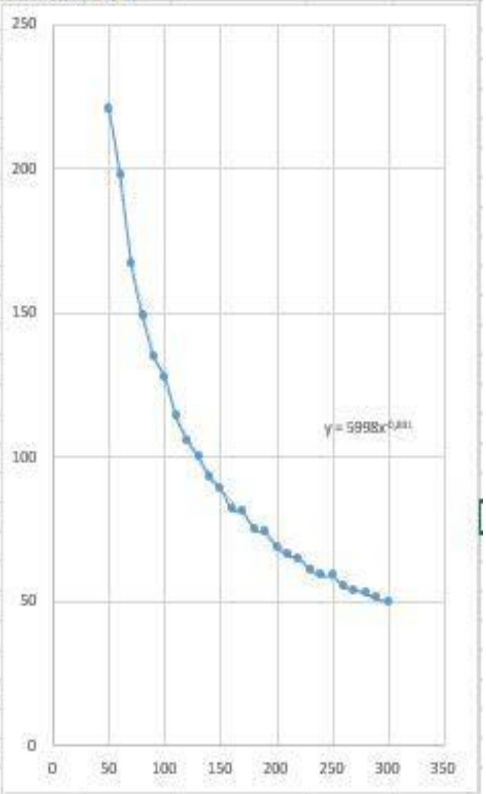

Figure 8. Face Distance Detection Program.

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

3 RESULTS AND DISCUSSION

3.1 Testing Face Distance Identification to the Camera

Face identification testing is performed by displaying a video on the display and detecting the distance between the face and the camera that corresponds to the real distance. **Table 1.** is based on the results of the face distance detection accuracy testing of the system by comparing the camera readings with actual measurements

Table 1. Results of Face-to-Camera Distance Experiment.

No	Jarak Real (cm)	Percobaan ke-1 (cm)	Percobaan ke-2 (cm)	Percobaan ke-3 (cm)
1	50	51	50	51
2	100	99	99	99
3	150	148	147	147
4	200	196	196	196
5	250	244	245	244

3.2 Face Distance Identification Testing

Next, distance identification testing was conducted between faces. It can be observed that the system detects distances between faces less than 100 cm. In such cases, the system displays a red line, while if the distance between faces is greater than 100 cm, the line appears green.

Below is **Figure 9**. showing the testing of face distance identification within a range of less than 100 cm:

Figure 9. Display of Face Distance Identification Less Than 100 cm.

In **Figure 9**. it is indicated that the distance between two individuals is 89 cm, the distance between the face of a person wearing a black shirt and the camera is 116 cm, and the distance between the face of a person wearing a red shirt and the camera is 125 cm. In the image, it can be seen that the distance between individuals is marked with a red line, indicating that the distance between them is unsafe.

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

Below is **Figure 10**. illustrating the testing of face distance exceeding 100 cm:

Figure 10. Testing Of Face Distance Exceeding 100 cm.

In **Figure 10**. it can be explained that the distance between the individuals is 128 cm, and the distance between the face of the person wearing black clothing and the camera is 112 cm. In the image, it can be observed that the distance between the individuals is indicated by a green line, indicating that the distance between them is considered safe.

3.3 Testing with 5 Objects

In **Figure 11**. it can be explained that the distance between the second and third person is 79 cm, indicated by the red line, indicating an unsafe distance. However, the first, fourth, and fifth persons are not detected due to lighting intensity.

Based on **Table 2**. according to the system testing results, it can provide a warning if two or more people are detected to be in close proximity with a distance less than 1 meter.

Table 2. Results of Testing with 5 Objects

No	Number of People	Result
1	1	Detected
2	2	Detected
3	3	Detected
4	4	Detected
5	5	Detected

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

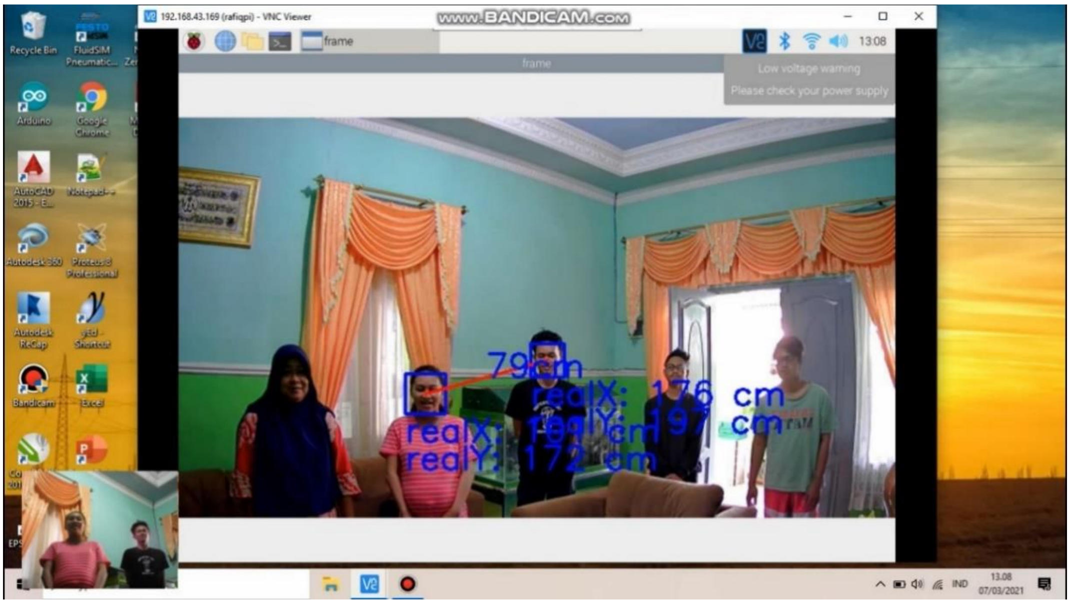

Figure 11. Display of Testing with 5 Objects.

3.4 USB Speaker Testing

The testing was conducted by trying out at specific distances, which can be seen in **Table 3**.

From the testing, it can be observed that there is an unreadable condition where if the distance is more than 100 cm. And when the distance is less than 100 cm, the USB Speaker lights up.

Table 3. USB Speaker Testing

No	Distance Between Faces (cm)	USB Speaker
1	30	On
2	60	On
3	90	On
4	120	Off
5	150	Off
6	180	Off
7	210	Off
8	240	Off
9	270	Off
10	300	Off

4 CONCLUSION

Based on the testing results and data collected during the research, several conclusions can be drawn. Firstly, the system for detecting the distance between the camera and the face showed a relatively lower level of accuracy. As the distance between the camera and the face increased, the error in the testing results also became more significant. Secondly, the system for detecting the distance between faces proved capable of identifying more than two faces and issuing warnings when faces were in close proximity to each other. However, it should be noted that the system's performance was affected by lighting conditions. Objects became harder to detect under excessively bright or dark lighting. Additionally, the system had a maximum range of 300 cm. Finally, the USB Speaker served as an audio output that activated when the distance between faces was less than 100 cm.

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

Based on the testing results and data collected during the research, several recommendations can be made. Firstly, considering the use of a higher-specification camera would improve the accuracy of distance readings and reduce errors. Secondly, careful consideration should be given to the resource utilization of the Raspberry Pi to ensure optimal performance. Lastly, it is important to acknowledge that this device focuses solely on detecting the distance between faces, which may limit its overall effectiveness in certain scenarios.

5 AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. Authors confirmed that the paper was free of plagiarism.

6 REFERENCES

[1] Juaningsih, I. N., Consuello, Y., Tarmidzi, A., dan NurIrfan, D. (2020). Optimalisasi kebijakan pemerintah dalam penanganan COVID-19 terhadap masyarakat Indonesia. *SALAM: Jurnal Sosial dan Budaya Syar-i*, *7*(6), 509-518.

[2] Choi, C. H., Kim, J., Hyun, J., Kim, Y., and Moon, B. (2022). Face detection using haar cascade classifiers based on vertical component calibration. *Human-centric Computing and Information Sciences*, *12*(11), 1-17.

[3] Minu, M. S., Arun, K., Tiwari, A., and Rampuria, P. (2020). Face recognition system based on haar cascade classifier. *International Journal of Advanced Science and Technology*, *29*(5), 3799-3805.

[4] Paramitha, I. A. P. I., Djuni, I. D., dan Setiawan, W. (2020). Rancang bangun prototipe sistem pendeteksi asap rokok berbasis mikrokontroler menggunakan sensor MQ-2 dilengkapi dengan exhaust fan. *Jurnal SPEKTRUM*, *7*(3), 69-75.

[5] Syafruddin, R., Ramady, G. D., dan Hudaya, R. R. (2021). Rancang bangun sistem proteksi daya listrik menggunakan sensor arus dan tegangan berbasis arduino. *Jurnal Online Sekolah Tinggi Teknologi Mandala*, *16*(1), 36-43.

[6] Zanofa, A. P., Arrahman, R., Bakri, M., dan Budiman, A. (2020). Pintu gerbang otomatis berbasis mikrokontroler arduino uno r3. *Jurnal Teknik dan Sistem Komputer*, *1*(1), 22-27.

[7] Khalifa, A. F., Badr, E., and Elmahdy, H. N. (2019). A survey on human detection surveillance systems for raspberry pi. *Image and Vision Computing*, *85*(1), 1-13.

[8] Sari, I. P., Al-Khowarizmi, A. K., dan Batubara, I. H. (2021). Analisa sistem kendali pemanfaatan raspberry pi sebagai server web untuk pengontrol arus listrik jarak jauh. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, *6*(1), 99-103.

[9] Chellappa, A., Reddy, M. S., Ezhilarasie, R. R., S. Suguna S. K., and Umamakeswari, A. (2018). Fatigue detection using raspberry pi 3. *International Journal of Engineering and Technology, 7*(2), 29-32.

[10] Efendi, Y. (2018). Internet of things (IoT) sistem pengendalian lampu menggunakan raspberry pi berbasis mobile. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, *4*(2), 21-27.

[11] Ristyawan, E. F., Yuniarno, E. M., dan Kurniawan, A. (2018). Digital signage berbasis raspberry pi 3. *Jurnal Teknik ITS*, *7*(1), A171-A174.

[12] Sajjad, M., Nasir, M., Muhammad, K., Khan, S., Jan, Z., Sangaiah, A. K., Elhoseny, M., and Baik, S. W. (2020). Raspberry pi assisted face recognition framework for

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149

enhanced law-enforcement services in smart cities. *Future Generation Computer Systems*, *108*(1), 995-1007.

[13] Pasumarti, P., and Sekhar, P. P. (2018). Classroom attendance using face detection and raspberry pi. *International Research Journal of Engineering and Technology (IRJET)*, *5*(3), 167-171.

[14] Aiyub, F. F., and Munawir, M. (2019). Kontrol mouse menggunakan webcam berdasarkan deteksi warna. *JTIM: Jurnal Teknologi Informasi dan Multimedia*, *1*(1), 73-77.

[15] A. Zein., (2018). Pendeteksian kantuk secara real time menggunakan pustaka openCV dan DLIB python, *Jurnal Penelitian dan Pengkajian Sains dan Teknologi*, *28*(2), 22-26.

[16] Juneja, A., Juneja, S., Soneja, A., and Jain, S. (2021). Real time object detection using CNN based single shot detector model. *Journal of Information Technology Management*, *13*(1), 62-80.

[17] Zulkhaidi, T. C. A. S., Maria, A., dan Yulianto, Y. (2019). Pengenalan pola bentuk wajah dengan openCV, *Teknologi Rekayasa Perangkat Lunak*, *3*(2), 181-186.

[18] Ismael, K. D., and Irina, S. (2020). Face recognition using viola-jones depending on python. *Indonesian Journal of Electrical Engineering and Computer Science*, *20*(3), 1513-1521.

[19] Lazaro, J. L. A., Buliali, dan Amaliah. B. (2017). Deteksi jenis kendaraan di jalan menggunakan openCV, *Jurnal Teknik ITS*, *6*(2), 293-299.

[20] Wajdi, M. F., dan Sugiantara, J. (2018). Pemanfaatan teknik pengenalan wajah berbasis openCV untuk sistem informasi pencatatan kehadiran dosen. *Infotek: Jurnal Informatika dan Teknologi*, *1*(2), 96-106.

[21] Dhawle, T., Ukey, U., and Choudante, R. (2020). Face detection and recognition using openCV and python. *International Research Journal of Engineering and Technology (IRJET)*, 7(10), 1269-1271.

[22] Hasan, R. T., and Sallow, A. B. (2021). Face detection and recognition using openCV. *Journal of Soft Computing and Data Mining*, *2*(2), 86-97.

DOI: https://doi.org/10.17509/coelite.v1i2.59702
p- ISSN 2829-4157 e- ISSN 2829-4149