Design and Build a Prayer Rak'ah Reminder Device for Elderly People with Pose Detection Using MediaPipe Based on Raspberry Pi

Luthfi Kukuh Raharjo[1], Abdul Rasyid[2], Moh. Abdullah Anshori [3]
[1,3] Digital Telecommunication Network Study Program, Department of Electrical Engineering,
State Polytechnic of Malang, 65141, Indonesia
[2]Telecommunication Engineering Study Program, Department of Electrical Engineering,
State Polytechnic of Malang, 65141, Indonesia

[1]luthfikukuh@gmail.com, [2] abdul.rasyid@polinema.ac.id, [3]moh.abdullah@polinema.ac.id

Abstract— **Establishing the five obligatory prayers is a necessity that Muslims must undertake. Problems often occur in people with memory problems, such as the elderly. Obstacles that often occur include forgetting the rak'ah and difficulty remembering the next pose to be performed. New technologies continue to emerge including digital imagery. Digital imagery can be used to help with these problems by utilizing pose detection using the MediaPipe library. MediaPipe is used to determine body parts visibility and joint angles captured by the webcam to detect performed pose. By detecting the pose, the output is then generated in an LED Matrix display namely the rak'ah and pose. The results of this study showed that the percentage of success in identifying ruku' is 93.73%, i'tidal is 94.12%, sujud is 92.55%, first tahiyah is 89.17%, final tahiyah is 82%, with the highest percentage of 98.04% in standing pose. The pose detection success percentages based on the distance between the performer and the webcam are from 150cm is 91.88% success percentage, at 200cm success percentage is 92.42%, and at a distance of 250cm is 93.75%, with the highest success percentage at the distance of 250cm. The system average delay for detecting poses is 1.028 seconds.**

Keywords— MediaPipe, Pose Detection, Prayer Poses, Prayer Rak'ah

I. INTRODUCTION

Establishing the five daily prayers according to the times specified is an obligation that must be carried out by Muslims. However, problems often occur in people who have memory problems or are senile, such as the elderly. Obstacles that often occur include forgetting the number of rak'ahs that have been done and difficulty knowing the prayer poses that have been done. So that it interferes with the course of prayer activities for senile people. These obstacles will certainly be a problem for Muslims who have memory problems establishing prayers. With the problem of forgetting the rak'ah of prayer, it is necessary to use technology as a reminder of the rak'ah of prayer.

Along with technological developments, new technologies continue to emerge including digital imagery. Digital imagery is a process to convert an image into another image representation. One technology that can be used to help with these problems is digital imagery by utilizing pose detection in MediaPipe Pose solution using the MediaPipe library. Inferring 33 3D landmarks and background segmentation masks across the body from RGB video frames, MediaPipe Pose is a machine-learning approach for high-fidelity body pose detection [1-3]. Unlike the common digital imagery to detect poses by training the program by inputting image samples, MediaPipe uses pre-trained a pose estimation model. From the model, the pose can be determined by utilizing the body part's visibility and the joint angles of the body.

The MAX7219/MAX7221 are integrated serial common-cathode input/output display drivers that link microprocessors (µPs) 64 individual LED displays, or 7-segment numeric LED displays with up to 8 digits. [4-8]. The MAX7219 will be used as the output driver for the LED Matrix 32*8 display.

The research entitled "Pose Estimated Yoga Monitoring System" was conducted by Ardra Anilkumar, et al. In this research, body pose estimation is utilized to determine where joints and body components of a human are located in the given image. Using the data acquired from the pre-trained pose estimation model, called MediaPipe, many applications can be created by using the data. Applications include gait analysis, motion capture, anomaly detection, sign language identification, and so on. Its main objective is to build a yoga tracking system that analyzes and tracks a user's motions and postures for faults in a yoga routine. The user is then alerted of his incorrect posture through a digital display or wireless speaker. Inaccurate user poses can be shown in real time so that users can correct their mistakes [9-12].

In this study, the past research will be the main reference for designing a prayer rak'ah reminder system, and prayer pose was carried out by utilizing the MediaPipe library which functions to read the prayer poses performed. MediaPipe will then be run in the Python programming language using the Raspberry Pi. The results of the prayer pose are then processed to determine the total of rak'ah and the next pose to be carried out with an output in the form of an LED dot matrix to help the user to know what rak'ah the user is at and what pose should be done next. The success of this device will be measured based on whether the display and the performed pose are corresponding [13-15].

II. METHOD

A. Stages of Research

The stages of the research to be carried out are stated in the flowchart shown in figure 1 below.

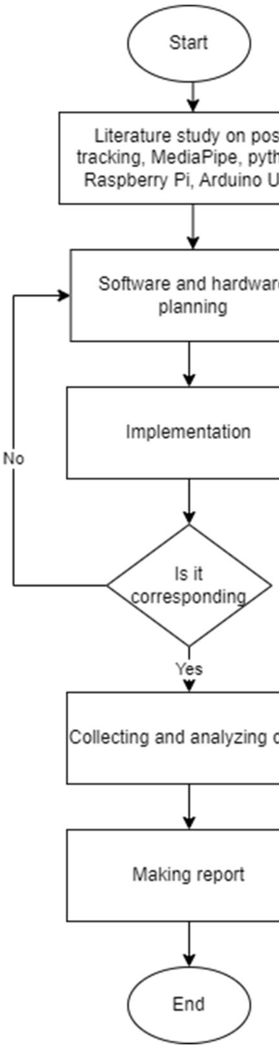

Start
Literature study on pose tracking, MediaPipe, python, Raspberry Pi, Arduino Uno
Software and hardware planning
Implementation
No
Is it corresponding
Yes
Collecting and analyzing data
Making report
End

Figure 1 Stages of Research Flowchart

The explanation of the research design flowchart is as follows:

1. Literature study
   The purpose of this literature review is to be able to accurately identify research objectives by making observations of the problem. Then a literature study was carried out by looking for references such as journals and studying supporting books on Pose tracking, MediaPIpe, Raspberry Pi, Arduino Uno, etc.
2. Software and hardware planning
   At this stage, the hardware of the device and the software needed for the device to work is planned.
3. Implementation
   At this stage, the software is written and the hardware design is assembled. If the result can be implemented then the stage will be continued. If the implementation fails the software and hardware will be replanned.
4. Collecting and analyzing data
   The implementation data is then collected then analyzed.
5. Making report
   At this stage, the result of the data collection and analysis will then be made into a report.

B. System Diagram Block

The design that will be made to facilitate system design requires a system block diagram in this study is shown in Figure 2 which shows the overall system of the device.

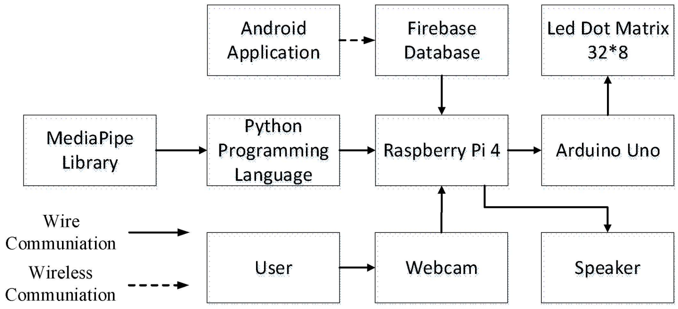


Figure 2 System Diagram Block

Based on Figure 2, each block in the figure can be explained as follows:

1. User is the one who performed the prayer and gets its poses captured.
2. Webcam is used to capture the prayer poses performed.
3. MediaPipe is the library that provides a pre-trained pose estimation model to be used to estimate the position of human body parts and joints from the image captured by the webcam.
4. Python is used to run the MediaPipe library and to determine the prayer poses by concluding the angle of the joints and visibility from the estimated joints and body parts given by the MediaPipe library.
5. Raspberry Pi as the computer that runs the program.
6. Android application is used to give input of one of the five obligatory prayers.
7. Firebase Database is used to store the selection of the five obligatory prayers sent by the android application.
8. Arduino Uno is to control the LED dot matrix from the given output of the python program
9. LED dot matrix is used to display what rak'ah is being performed and also to display what the next pose should be performed.
10. Speaker is used as the sound output of the rak'ah

C. System Flowchart

The following is the flowchart of the system. There are two flowcharts in this system which consist of python system flowchart which is inside the Raspberry Pi, and the output flowchart which is installed in Arduino Uno. The system flowchart for the python program is shown in Figure 3 as follows.

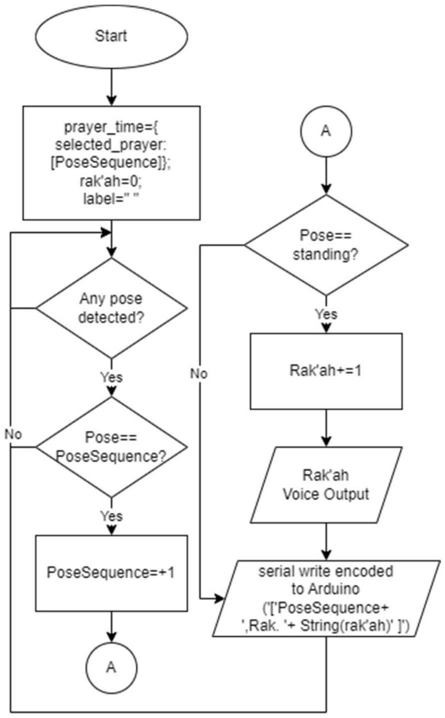


Figure 3 System Flowchart of The Python Program

After the program is starting, the first variable is set, namely, the selected prayer which is one of the five obligatory prayers that is input received from the firebase real-time database, and the array sequence of poses of each of the five obligatory prayer options. The program then detects the user's pose captured by the camera. If a pose is registered it will be checked whether the pose performed is the same as the sequence it should be. If not then re-detect the user's pose. If yes, then the pose sequence array continues to the next pose. If not then return to detecting pose if. The program then checks is the pose performed is standing. If yes then the previous amount of rak'ah is then added by one and then generating the voice output of the amount of the rak'ah. The python program then encodes the pose and rak'ah serial data to Arduino Uno. The following figure is the Arduino program which is shown in figure 4 as follows.

After the program is started, the serial communication and LED Matrix are initialized. After initializing the variable, the program then reads the encoded serial data from the raspberry pi. The program then checks whether the data is valid or not. The valid data means whether there is a complete array of data, which are the pose sequence data and the amount of rak'ah data. If yes, do the sorting of the pose and rak'ah. After that, the pose data and the rak'ah data are displayed shifting every 1,5s. If there's a change of data, the program then does the sorting again.

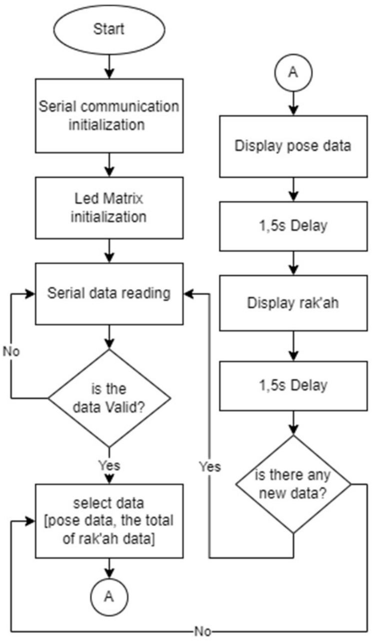


Figure 4 System Flowchart of The Arduino Program

D. Device Design

The device design shows the illustration of the placement and the use of each part of the device. The illustration of the device is shown in Figure 5 as follows.

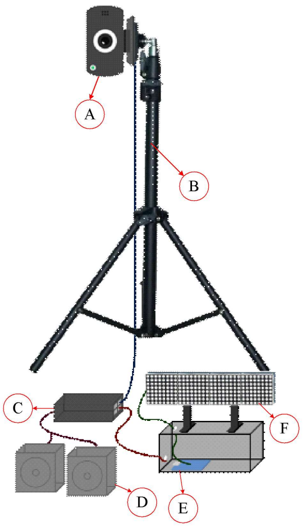


Figure 5 Device Design

The following is the description of Figure 5:

1. **A** Shows the camera placed at 90° angle for the user body pose capture device.

2. **B** Shows the tripod to place the webcam, the tripod height can also be adjusted.
3. **C** Shows the Raspberry Pi 4 as the main control center which is for pose detection and giving output.
4. **D** Shows the speaker for the voice output.
5. **E** Shows Arduino Uno board as the Led controler.
6. **F** Shows the LED dot matrix 32×8 as the display output for the rak'ah and pose data text.

The following is the circuit planning of the Raspberry Pi, Arduino Uno and the LED output shown in figure 6 below.

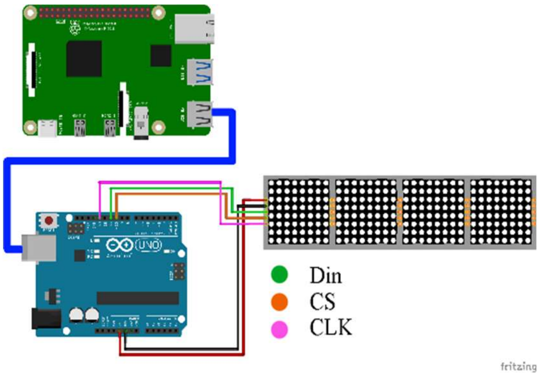


Figure 6 Circuit Design

The figure 6 shows the connection between the Raspberry Pi to the Arduino Uno and the LED driver. The Raspberry Pi connected to the Arduino Uno by serial communication through the the USB output. The USB output from Raspberry Pi is also distribute power to the Arduino Uno and Led display. On the Arduino Uno board there are three pin used which are pin 10 to the CS pin on the LED driver's pin, pin 11 to the data pin, and pin 13 to the CLK pin. Figure 7 below shows the implementation of the device design.

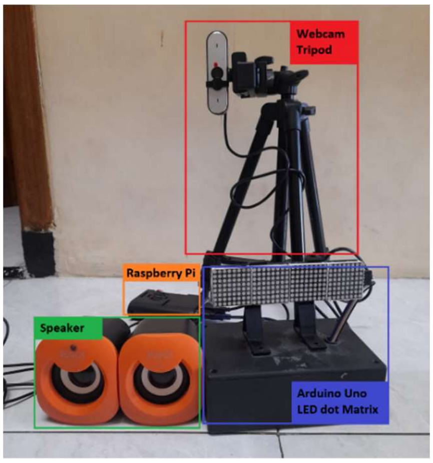
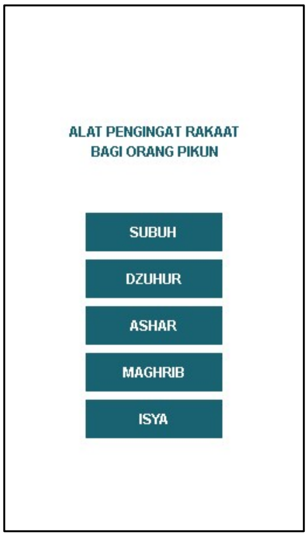


Figure 7 Device Design Implementation

E. Android Application Design

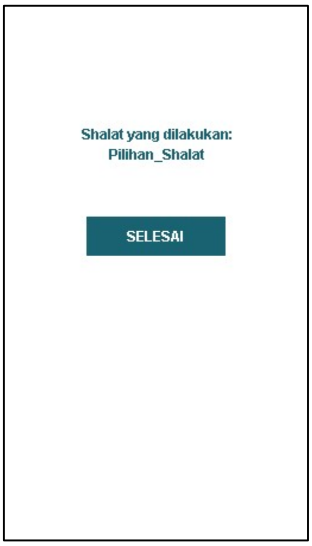
The following is the android application design. This application will later be used to input one of the five obligatory prayers. Figure 8 shows the prayer selection page of the application.



ALAT PENGINGAT RAKAAT
BAGI ORANG PIKUN

Figure 8 Prayer Selection Page Design

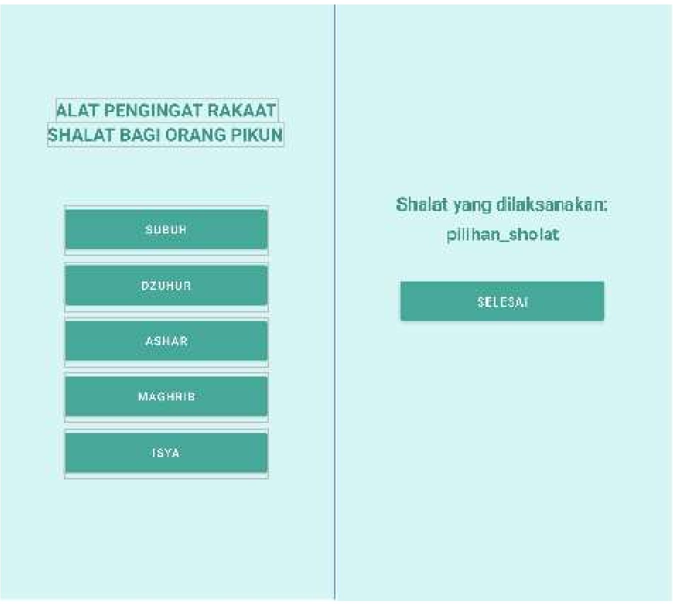
Figure 8 is the first display of the android application where the page is a selection of the five types of obligatory prayers. On this page, there are five kinds of buttons that correspond to the obligatory prayers which are user input according to which prayer will be performed. After that, the obligatory prayer button that is selected will then be sent to the firebase database. The data from the firebase is a prayer selection variable that is input for the program on the Raspberry Pi, the python program then starts counting rak'ah and detecting pose. Figure 9 shows the reset page of the application



Shalat yang dilakukan:
Pilihan_Shalat

Figure 9 Prayer Reset Page Design

Figure 9 is the second display of the android application where the page contains a button. The button on the page serves as the reset button for the firebase. When the button is pressed the selected prayer will be reset and the python program will reset the pose and rak'ah variable. Figure 10 shows the software design implementation


ALAT PENGINGAT RAKAAT SHALAT BAGI ORANG PIKUN

Shalat yang dilaksanakan: pilihan_sholat

Figure 10 Software Design Implementation

F. *Pose Identification Planning*

The following is the identification planning for the prayer poses. Figure 11 shows Pose Landmarks of the pre-trained pose estimation model for MediaPipe library.

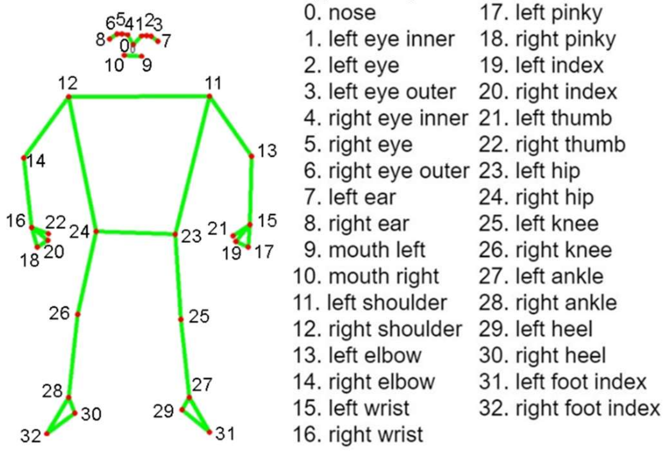


Figure 11 Pose Landmarks

MediaPipe pose is used to determine the angles of the body joints such as the angles of the elbows, armpits, hips, and knees. The program will later conclude from all of the body joints and the landmarks visibility to determine the pose performed by the user. Using a detector, the MediaPipe pipeline first places the individual/pose region-of-interest (ROI) in the frame. Utilizing ROI trimmed frames as input, the tracker predicts posture landmarks as well as segmentation masks in ROI. MediaPipe poses is a subsystem of MediaPipe that focuses on monitoring body poses. The landmarks of the pose identification for the prayer are shown in the following table 1.

TABLE I
BODY POSE LANDMARK IDENTIFICATION PLANNING

No	Body pose identification	Display
1	Left and right ankle visibility above 85%. Left and right knee visibility is above 85% The left knee angle is between 170°-190°. The right knee angle is 170°-190°. Left hip angle 140°-180°. Right hip angle 165°-210°. Left armpit angle 15°-55°. Right armpit angle 15°-55°. Left elbow angle 30°-80°. Right elbow angle 275°-320°	Standing
2	Left and right ankle visibility above 85%. Left and right knee visibility above 85%. The left knee angle is between 170°-190°. The right knee angle is 170°-190°. The left hip angle is 120°-175°. Right hip angle 200°-255°. Left armpit angle 0°-50°. Right armpit angle 0°-50°. Left elbow angle 165°-240°. Right elbow angle 110°-185°	Ruku'
3	Visibility of left ankle and right foot above 85%. Visibility of left knee and right knee above 85%. The left knee angle is between 170°-190°. The right knee angle is 170°-190°. Left hip angle 140°-180°. Right hip angle 165°-210°. Large left armpit angle 15°-55°. Right armpit angle 15°-55°. Left elbow angle 165°-240° Right elbow angle 110°-185°	'I'tidal
4	Visibility of the left ankle and right ankle is above 85%. Visibility of left knee and right knee is above 85%	Sujud
5	Besides of sujud pose if the visibility of the left knee and right knee is below 35%	Ifitrasy/Tahiyah (depending on the pose sequence)

III. RESULTS AND DISCUSSION

A. Distance Test Result

The distance test is carried out to determine whether the pose can be detected or not depending on the distance of the user to the webcam. To start the test, first, the user chooses one of the five obligatory prayers on the android application. The program then initializes the sequence of the pose of the prayer chosen. For example, if the user chooses the maghrib prayer the pose sequence will start from standing to ruku' then to i'tidal up until the final tahiyah in the third rak'ah. the program will start reading the pose when the user performs the first pose sequence which is the standing pose. As for the amount of rak'ah, it will be added by one by the program every time the pose performed by the user is standing. The test then will be conducted by observing whether the pose done by the user is corresponding with the displayed output, including the displayed rak'ah with the actual rak'ah. figure 12 below shows some samples from the conducted tests.

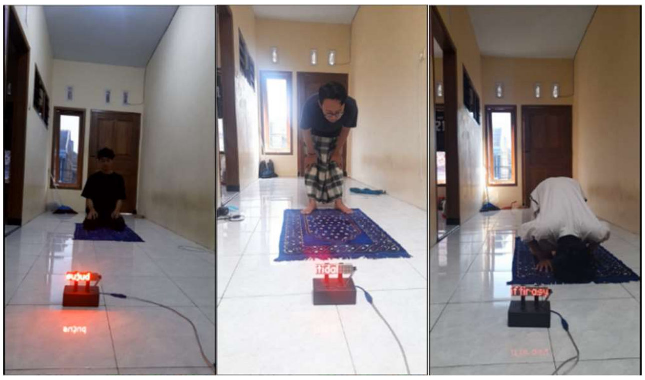


Figure 12 Pose Detection Samples

Figure 12 shows some samples of the conducted test. The samples were taken by performing each of the five obligatory prayers. The test is carried out by performing the prayer with ten different models, each at three different distances between the webcam and the models (150cm, 200cm, and 250cm). The following graph in figure 13 shows the result of the test conducted.

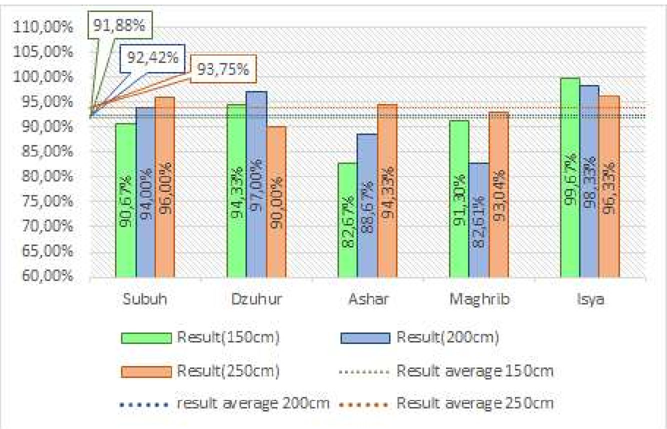


Figure 13 Result Graph

The graph in figure 13 shows the percentage results from each of the three distances test of all detected poses including the rak'ahs. Based on the result, the success percentage for a distance of 150cm is 91,88%, for a distance of 200cm is 92,42, and the success percentage from a 250cm distance is 93,75%. With the highest percentage of success performed at a distance of 250cm.

B. Pose Reading Result

In this section, the results of the pose data in the previous test will be used to observe the program's ability to detect each prayer pose. The first tahiyah and final tahiyah pose will follow the pose identification of the iftirasy movements, the program will display the output of the first tahiyah and final tahiyah depending on the pose sequence of the prayer performed. The success percentages of each pose is shown in table 2 below.

TABLE II
SUCCESS RATE OF EACH POSE

No	Pose	Number of Trials	Number of detected	Success Percentages
1	Standing	510	500	98,04%
2	Ruku'	510	478	93,73%
3	I'tidal	510	480	94,12%
4	Sujud	1020	921	90,29%
5	Iftirrasy	510	472	92,55%
6	First Tahiyah	120	107	89,17%
7	Final Tahiyah	150	123	82,00%

From the test results shown in table 2, the results of testing the success rate of the standing pose is 98.04%, the ruku' pose is 93.73%, the itidal pose is 94.12%, the sujud pose is 92.55%, the first tahiyah pose is 89 .17%, the final tahiyah pose is 82%. From the results obtained, the highest success rate of the test obtained is the standing pose and the lowest success rate of the test is obtained by the final tahiyah pose.

C. System Delay Test

The system delay test is carried out to determine the delay between the real-time pose performed and the output displayed. This test was carried out by using a timer to calculate the time difference between the performed pose and its output. The result of this test is shown in table 3 below.

TABLE III
SYSTEM'S DELAY

No	Pose	Delay
1	Standing	0,96s
2	Ruku'	0,93s
3	I'tidal	1,06s
4	Sujud	0,89s
5	Iftirrasy	1s
6	Sujud	1,08s
7	Standing	0,94s
8	Ruku'	1,06s

No	Pose	Delay
9	I'tidal	1,13s
10	Sujud	0,98s
11	Iftirasy	1,19s
12	Sujud	1,11s
Average delay		1,028

Based on the test result shown in table 3, the system's delay of the MediaPipe library implementation on Raspberry pi 4 is obtained with the average delay from the twelve tests of 1,028s.

IV. CONCLUSION

After conducting the research and analyzing the result we can conclude that first for the pose detection the percentage of success of each pose, for the standing pose is 98.04%, ruku' pose is 93.73%, i'tidal pose is 94.12%, sujud pose is 92.55%, first tahiyat pose is 89.17%, final tahiyat pose is 82%, with the highest success percentage being the standing pose. The result of distance, the success percentages of the pose detection based on the distance of the prayer performed are from 150cm the success percentage is 91.88%, at a 200cm distance the percentage is 92.42%, and at the distance of 250cm is 93.75%, with the highest success percentages being at the distance of 250cm. The delay between the real-time poses performed and the pose reading results when running the MediaPipe library in raspberry pi 4 is in the average of 1,028s.

REFERENCES

[1] A. Anilkumar, A. K.T., S. Sajan, and S. K.A., "Pose Estimated Yoga Monitoring System," SSRN Electron. J., no. Icicnis, pp. 1–8, 2021, doi: 10.2139/ssrn.3882498.

[2] A. Halder and A. Tayade, "Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning," Int. J. Res. Publ. Rev., no. 2, pp. 9–17, 2021, [Online]. Available: www.ijrpr.com

[3] B. Lanza, M. Lancini, C. Nuzzi, and S. Pasinetti, "Deep Learning for Gesture Recognition in Gym Training Performed By a Vision-Based Augmented Reality Smart Mirror," pp. 363–366, 2022.

[4] H. Moetia Putri and W. Fuadi, "Pendeteksian Bahasa Isyarat Indonesia Secara Real-Time Menggunakan Long Short-Term Memory (Lstm)".

[5] Indriani, M. Harris, and A. S. Agoes, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," Proc. 2nd Int. Semin. Sci. Appl. Technol. (ISSAT 2021), vol. 207, no. Issat, pp. 101–108, 2021, doi: 10.2991/aer.k.211106.017.

[6] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," 2019, [Online]. Available: http://arxiv.org/abs/1906.08172

[7] Google LLC, "MediaPipe Pose," GitHub, 2020. https://google.github.io/mediapipe/solutions/pose (accessed Feb. 20, 2022).

[8] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, First. Sebastopol: O'Reilly Media, Inc, 2008.

[9] G. Zaccone, M. R. Karim, and A. Menshawy, Deep Learning with TensorFlow, First. Mumbai: Packt Publishing Ltd., 2017.

[10] Raspberry Pi Foundation, "What is a Raspberry Pi?," Raspberry Pi Foundation, 2014. https://www.raspberrypi.org/help/what- is-a-raspberry-pi/#:~:text=The Raspberry Pi is a,languages like Scratch and Python. (accessed Feb. 20, 2022).

[11] H. A. Dharmawan, Mikrokontroler: Konsep Dasar dan Praktis. Malang: UB Press, 2017.

[12] A. S. Nataprawira, A. Rizal, and A. S. Wibowo, "Perancangan Display Led Dot Matrix Via Wi-Fi Menggunakan Aplikasi Mobile Android," Intech, vol. 1, no. 1, pp. 1–7, 2020.

[13] A. Sonita and R. F. Fardianitama, "Aplikasi E-Order Menggunakan Firebase dan Algoritme Knuth Morris Pratt Berbasis Android," Pseudocode, vol. 5, no. 2, pp. 38–45, 2018, doi: 10.33369/pseudocode.5.2.38-45.

[14] Herlinah and M. KH, Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition. Jakarta: Elex Media Komputindo, 2019.

[15] Sri Haryati, "(R & D) Sebagai Salah Satu Model Penelitian Dalam Bidang Pendidikan," Academia, vol. 37, no. 1, p. 13, 2012.