Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 1 (2025) 51 – 61 ISSN Media Elektronik: 2580-0760

The Internet-of-Things-based Fishpond Security System Using NodeMCU ESP32-CAM Microcontroller

Arwin Datumaya Wahyudi Sumari[1], Ilyas Annurroni[2], Astika Ayuningtyas[3]
[1]Department of Electrical Engineering, State Polytechnic of Malang, Malang, Indonesia
[2,3]Faculty of Industrial Technology, Adisutjipto Institute of Aerospace Technology, Yogyakarta, Indonesia
[1]arwin.sumari@polinema.ac.id, [2]ilyasannurroni148@gmail.com, [3]astika@itda.ac.id

Abstract

Fish theft in ponds is a common problem, especially in freshwater fish farms. To solve this problem, a security system that can detect human movement and provide real-time notifications is needed. This research aims to design and implement an Internet of Things (IoT)-based fishpond security system using NodeMCU ESP32-CAM Microcontroller equipped with HB100 Radar Sensor to detect human entity movement with NodeMCU ESP32-CAM to take pictures of the approaching human entity, as well as Arduino Uno R3 to control system inputs and outputs. The system also sends real-time notifications and can be managed independently by a social media application. The results show that the system can detect human movement well, provide real-time notifications, and be handled easily. The test results show that the HB100 Radar Sensor can detect entities with a maximum distance of 9 meters with overall accuracy of 90%, the Buzzer performs well according to the human entity detected by the sensor, the Arduino Uno R3 successfully sends a trigger signal to NodeMCU ESP32-CAM to activate the OV2640 camera to capture the detected human entities with a maximum distance of up to 60 meters with an optimal distance between 1 to 9 meters. Integrated system test results show that all components of the fishpond security system perform 100% well and are prospective to protect a fishpond with an area of 7x7 meter squares. This system provides a solution to reduce fish theft in ponds and delivers real-time information to the fishpond owner.

Keywords: Arduino Uno R3; fishpond security system; Internet of Things (IoT); NodeMCU ESP32-CAM; real-time notification

How to Cite: A. D. W. Sumari, I. Annurroni, and A. Ayuningtyas, "The Internet-of-Things-based Fishpond Security System Using NodeMCU ESP32-CAM Microcontroller", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 1, pp. 51 - 61, Jan. 2025
DOI: https://doi.org/10.29207/resti.v9i1.6033

1. Introduction

Fish theft is a problem that often occurs in freshwater fish farming, which impacts both financial and material and non-material losses for fishpond owners [1]. Material losses can include loss of fish and damage to ponds and equipment. Non-material losses can occur in decreased pond cultivation productivity and loss of consumer confidence. This situation has gradually raised concerns among freshwater fishpond owners so that the location of the fishpond around the house or in the middle of the rice field, which was initially considered safe, turns out to require extra security from theft attempts [2], [3].

Surveillance of fishponds will be very easy in bright environmental conditions, from morning to evening. On the other hand, surveillance in dark environments presents a significant challenge because human sensing capabilities are limited, and fish thieves take advantage of this situation to commit fish theft. Therefore, the fishpond security system is urgently needed in the field, especially by freshwater fish farm owners. Remote surveillance has become necessary for businesses with production areas, such as freshwater fishpond owners. So far, remote monitoring technology in fishponds is mainly aimed at monitoring pond water quality [4], [5], monitoring pond environmental conditions [6], for feeding fish [7]-[9], and fishpond management [10].

Received: 06-09-2024 | Accepted: 11-11-2024 | Published Online: 24-02-2025

The availability of IoT technology brought by the wave of the Industrial Revolution 4.0 makes remote monitoring possible to implement [11], such as monitoring fisheries' water quality [12], [13]. However, IoT itself will not work without the availability of the Internet and its means of access, such as Wireless Fidelity (Wi-Fi), which is currently supported by various electronic equipment [14], down to the level of electronic components such as microcontrollers. NodeMCU is one of the microcontroller modules equipped with Wi-Fi facilities, making it widely used in projects to manufacture electronic systems [15]. One of the prospective types of NodeMCUs used for security systems with remote surveillance is the ESP32-CAM, which has a vision sensor in the form of an OV2640 camera and Wi-Fi and Bluetooth facilities. NodeMCU ESP32-CAM has been widely used to implement security systems, especially for homes [16]-[18], offices [19], [20], and laboratories [21], but no one has used it for fishpond security systems combined with IoT and other microcontrollers.

Answering this need, in this study, an Internet-of-Things (IoT)-based fishpond security system was designed and implemented using two types of microcontrollers that support each other, namely NodeMCU ESP32-CAM and Arduino Uno R3, along with two types of sensors to detect the presence of human entities around the fish pond, namely a vision sensor using an OV2640 camera and a motion sensor using an HB100 Radar sensor. The fishpond security system can be monitored remotely with the help of social media applications [22], Telegram via mobile phones with consideration of speed and data security [9], [22] to make it easier for fishpond owners to monitor from anywhere and anytime [23].

It can be operated on Android or iOS platforms. This fishpond security system application was created using the C++ programming language integrated with the Telegram application via Bot as a notification service against security access [18]. The selection of NodeMCU ESP32-CAM for our proposed fishpond security system because of some considerations, such as low price but equipped with an on-board built-in camera, internet connectivity that enables IoT implementation, various General-Purpose Input/Output (GPIO) [24]-[26], and has been used for various remote monitoring systems [27]. Moreover, with such low-price microcontrollers, fishpond farmers can afford our proposed security system.

2. Research Methods

The research method for designing and implementing this IoT-based fishpond security system includes direct field surveys, interviews, software application design, hardware design, software and hardware integration, and integrated system testing.

2.1. Field Surveys and Interviews

The field survey was conducted to Remboko Village-Owned Enterprises (Badan Usaha Milik Desa/ BUMDes) in Sumberrejo Village, Tempel District, Sleman, D.I. Yogyakarta. As conveyed by the BUMDes Manager, fish theft cases in 2023 have occurred until now.

For this reason, it is suggested that there is a unique strategy to overcome it to reduce fish theft, one of which is to install security equipment in fishponds, especially in areas prone to fish theft.

2.2. Software Design

Two applications support this fish pond security system, namely an application that handles data from the detection of approaching human entities by the HB100 Radar sensor, which is connected to the Arduino Uno R3 microcontroller and then sends an external signal to the NodeMCU ESP32-CAM, and an application that handles input from the Arduino Uno R3 on the NodeMCU ESP32-CAM and input data from the OV2640 camera and then sends a remote surveillance message to the owner of the fish pond through the Telegram application. The flow chart of the first application is shown in Figure 1, and the second application is shown in Figure 2.

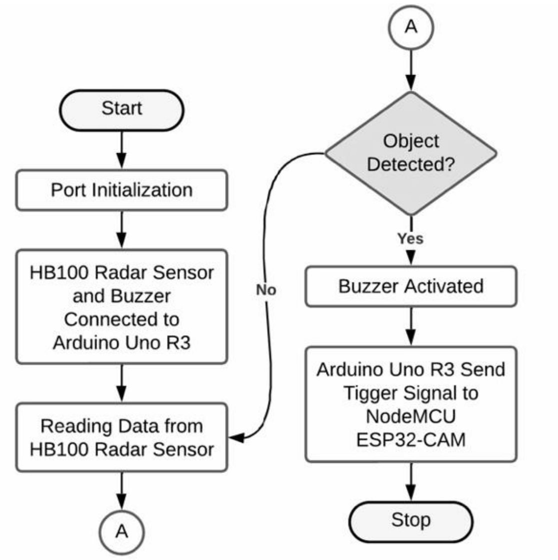

Figure 1. The first application flowchart handled by the Arduino Uno R3

The mechanism run by the first application in Figure 1 includes port initialization, namely setting pins on the Arduino Uno R3, which are inputs to receive data from the detection results of the HB100 Radar sensor and an output to activate the Buzzer according to the detection results of the HB100 Radar sensor. In addition, the Arduino Uno R3 also sends a trigger signal to the NodeMCU ESP32-CAM according to the detection results of the HB100 Radar sensor.

Based on the second application flow diagram shown in Figure 2, the mechanism that is executed includes port initialization, namely setting pins on the ESP32-CAM

NodeMCU, which will be used as input to receive trigger signals from the Arduino Uno R3 and input from image data from the OV2640 camera based on the results of detecting the presence of human entities from the HB100 radar sensor sent from the Arduino Uno R3 to the ESP32-CAM NodeMCU. For the results of the detection of approaching human entities in the form of notifications and image data captured by the camera sent to the mobile phone of the fishpond owner through the Telegram application, the NodeMCU ESP32-CAM will first connect itself to the internet network through the W-Fi Manager at the Internet Protocol (IP) address that has been prepared. The availability of an internet connection between the mobile phone and the NodeMCU ESP32-CAM allows the microcontroller to be controlled through the Telegram application.

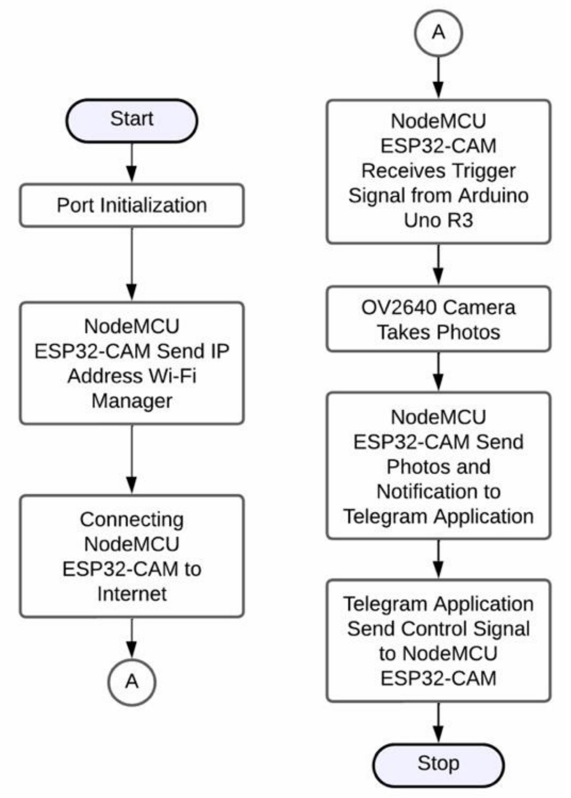

Figure 2. The second application flowchart handled by NodeMCU ESP32-CAM

2.3. Hardware Design

The hardware design begins with a schematic diagram of the interconnection between the components forming an IoT-based fish pond security system with the NodeMCU ESP32-CAM microcontroller, as shown in Figure 3. The primary power source of both the NodeMCU ESP32-CAM and Arduino Uno R3 is a Powerbank Solar Cell battery with an output voltage of 5 Volt Direct Current (VDC) with a current of 1 Ampere (A), which is channelled through the 5 VDC pin and GND pin, with an alternative via the Universal Serial Bus (USB) port on each microcontroller.

The two HB100 Radar sensors are connected to pin 2 and pin 3 of the Arduino Uno R3, while the Buzzer is connected via pin 4. Pin A0 (Analog0) is connected to the NodeMCU ESP32-CAM microcontroller. Both the sensor and the Buzzer are powered by the battery. Further information can be seen in Table 1.

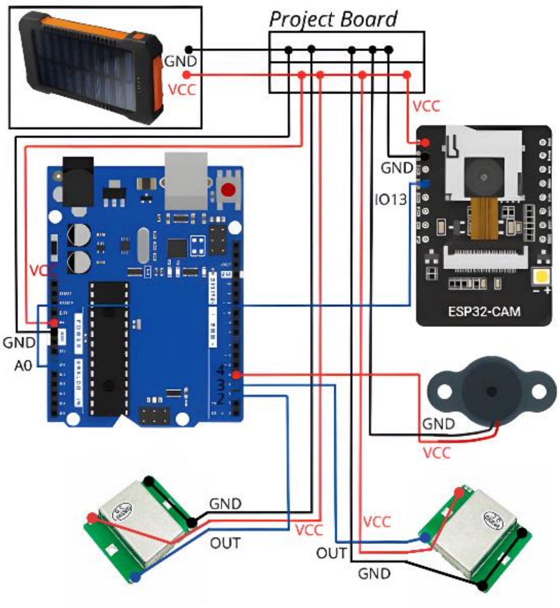

Figure 3. Schematic diagram of IoT-based fish pond security system and NodeMCU ESP32-CAM

Table 1. Connection between components of the fishpond security system

Microcontroller	Component
Pin 2 Digital Arduino Uno R3	Pin OUT HB100#1 Radar Sensor
Pin 3 Digital Arduino Uno R3	Pin OUT HB100#2 Radar Sensor
Pin 4 Digital Arduino Uno R3	Pin/Voltage VCC (+) Buzzer
Pin VCC Arduino Uno R3	VCC Voltage of Battery or Solar Cell Powerbank
Pin GND Arduino Uno R3	GND Voltage of Battery or Solar Cell Powerbank
Pin A0 (Analog) Arduino Uno R3	Pin GPIO13 NodeMCU ESP32-CAM
FPC Camera Connector ESP32-CAM	OV2640 Camera Module
All ESP32-CAM Pins	TTL CH340 USB
Micro USB Interface USB TTL CH340 ESP32-CAM	Solar Cell Powebank UBS

2.4. Software with Hardware Integration

Software and hardware integration is mandatory to ensure that the significant parts of the ESP32-CAM IoT-based fishpond security system and NodeMCU subsystems work as designed. To facilitate understanding of the collaboration of the two major parts, a subsystem integration diagram block is created, which is divided into three blocks: the Input block, the Process block, and the Output block, as shown in Figure 4.

The Input Block consists of the HB100 radar sensor, which detects the movement of the human entity, and the OV2640 camera module, which takes pictures of the human entity when the HB100 radar sensor detects the movement of the human entity. The Process Block consists of an Arduino Uno R3 microcontroller that functions to control the Buzzer, the HB100 Radar sensor, and will send a signal to the ESP32-CAM NodeMCU microcontroller when the HB100 Radar sensor detects the movement of a human entity.

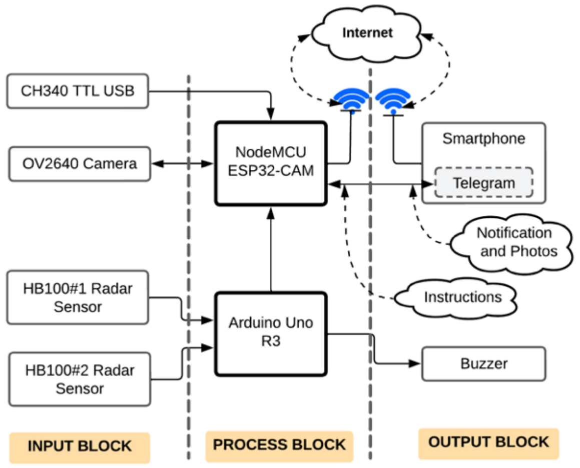

Figure 4. Subsystems integration block diagram of IoT-based fishpond security system and NodeMCU-ESP32

On the other hand, the ESP32-CAM NodeMCU Microcontroller oversees connecting the overall system to the internet network and sending images and notifications to the fishpond owner via the Telegram application. The Output Block consists of a buzzer that functions as an alarm when the system detects the movement of human entities around the fishpond and a smartphone that functions as a tool to receive messages from the NodeMCU ESP32-CAM microcontroller through the Telegram application.

2.5. *Fishpond Security System Placement Concept*

The fishpond used as the installation location of the fishpond security system measures 7x7 square meters, so a strategy for placing the system's components is needed. Figure 5 illustrates the fishpond situation and has been equipped with a sequence number for easy understanding.

The explanation of each number in Figure 5 is as follows: Fishpond with a size of 7x7 square meters; Location of the IoT-based fishpond security system and NodeMCU ESP32-CAM; The direction of the radar HB100#1 sensor beam is pointing East with a beam angle of 80^0; The beam direction of the OV2640 camera module is pointing East with a beam angle of 66^0; The direction of the radar HB100#2 sensor beam is pointing west with a beam angle of 80^0; Embankments or guardrails on the pond's West, South, and North sides.

The placement of the HB100#1 Radar sensor and the OV2640 camera vision sensor to the East is because, in this part, there are no obstacles in the form of embankments or guardrails, so it provides freedom of space for frequent fish theft. The HB100#2 radar sensor is directed to the West only to anticipate possible theft attempts from this section, even though there is already a barrier in the form of an embankment. Based on the survey results in the field, the actual location that makes it possible to place the NodeMCU ESP32-CAM and IoT-based fishpond security system is shown in Figure 6.

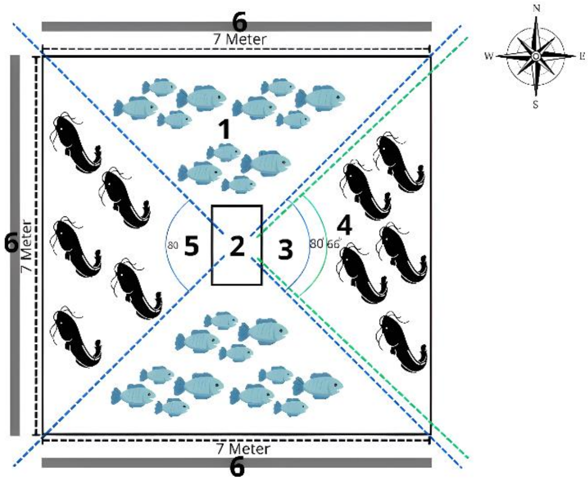

Figure 5. The concept of placing a fishpond security system according to actual conditions

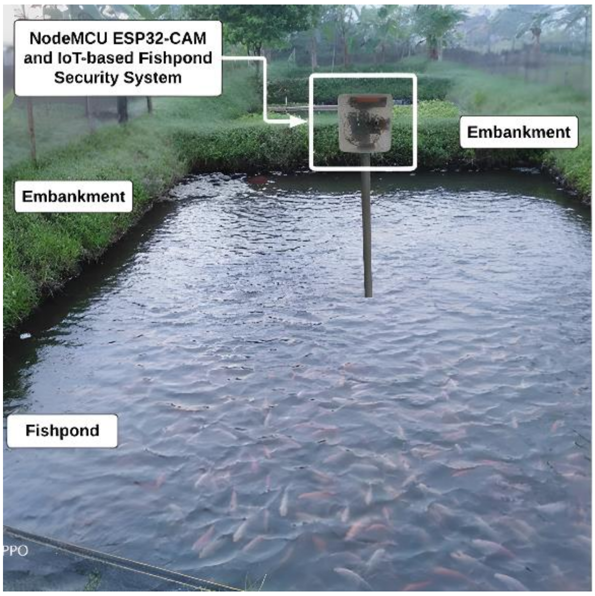

Figure 6. The actual location of a fishpond security system placement. The system's installation is just an illustration of where it is supposed to be located.

2.6. *Fishpond Security System Testing*

The test of the fishpond security system is carried out partially and in an integrated manner with the stages, as shown in Figure 7. The system is tested to ensure its key components function according to their intended purpose. Partial testing includes operational testing of HB100 radar sensors, Buzzer, OV2640 camera, and NodeMCU ESP32-CAM microcontrollers, while integrated testing is testing the system's work from receiving input to producing final outputs in the form of notifications to smartphones through the Telegram application.

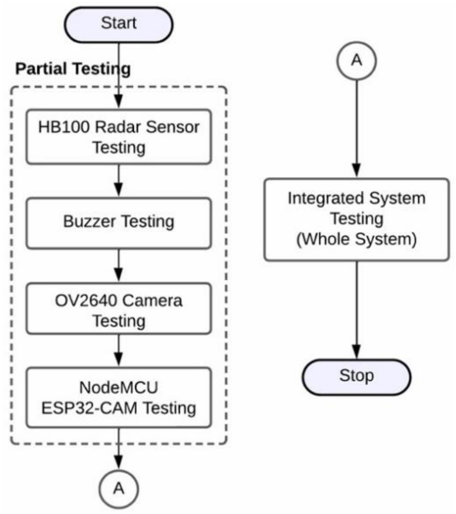

Figure 7. Partial and integrated system testing flowchart

3. Results and Discussions

3.1. Complete Circuit of Fishpond Security System

The complete circuit of the IoT-based fishpond security system and NodeMCU ESP32-CAM is shown in Figure 8.

3.2. Partial Testing – HB100 Radar Sensor Test Results

There are two HB100 radar sensors, and both have been tested for their respective functions. The test aimed to obtain data on the sensor's maximum range when detecting a human entity approaching a fishpond. The test results can be seen in Table 2, and only the test results on the HB100#1 radar sensor are presented. **Remarks: D means entity Successfully Detect, and F means entity Fail to Detect**.

Table 2. HB100#1 Radar Sensor Test Results

Entity	i^{th} Test Results				
Distance (m)	1	2	3	4	5
1	D (Detect)	D	D	D	D
2	D	D	D	D	D
3	D	D	D	D	D
4	D	D	D	D	D
5	D	D	D	D	D
6	D	D	D	D	D
7	D	D	D	D	D
8	D	D	D	D	D
9	D	D	D	D	D
10	F (Fail)	F	F	F	F
11	F	F	F	F	F

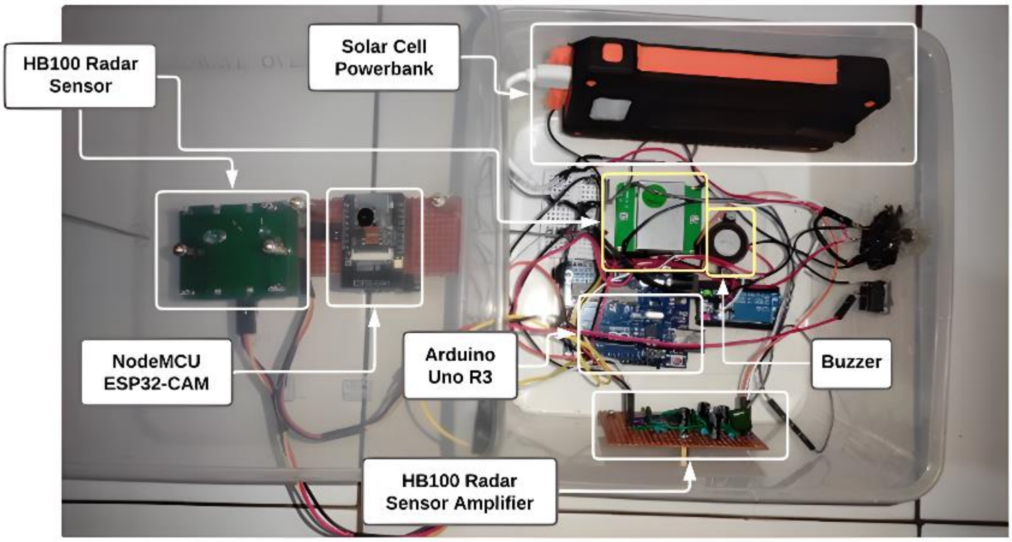

Figure 8. NodeMCU ESP32-CAM and IoT-based fish pond security system circuit

The test results show that in five tests, the HB100#1 radar sensor shows consistency in the ability to detect human entity motion at 1 to 9 meters. These results are inconsistent with the technical specifications of the sensor, which can detect entities at 8 to 30 feet or 2.6 to 10 meters because, at 10 meters, the sensor fails to detect the movement of a human entity. However, with **a detection accuracy of up to 90%**, the HB100 Radar sensor is feasible to maintain the safety of fishponds measuring 7x7 square meters.

3.3. Partial Testing – Buzzer Test Results

Buzzers are used to provide audible sound alarms up to a certain distance as a real-time notification to the owner of the fishpond that the fishpond security system detects the presence of human entities that are indicated to intend to commit theft. The Buzzer produces sound at a frequency of 1000 Hz with a duration of on for 500 ms and a duration of off for 500 ms. Testing was only carried out on the HB100#1 Radar sensor as a human entity motion detector, assuming the HB100#2 Radar sensor performs similarly to the HB100#1 Radar sensor. **Remarks: Act means Successfully Activated, and NAct means Not Activated**.

The results of the Buzzer test in Table 3 show consistency with the performance of the HB100#1 Radar sensor. The sensor detects the movement of human entities at 1 to 9 meters and automatically activates the Buzzer alarm with a duration of 500 ms. At a distance above 9 meters, the sensor cannot detect the movement of human entities, so the Buzzer will also not be activated. These results **also show that the Buzzer performance** is 100% consistent with the performance of the HB100 Radar sensor.

Table 3. Buzzer Testing Results for HB100#1 Radar Sensor

Entity Distance (m)	i^{th} Test Result 1	2	3	4	5
1	Act (Activated)	Act	Act	Act	Act
2	Act	Act	Act	Act	Act
3	Act	Act	Act	Act	Act
4	Act	Act	Act	Act	Act
5	Act	Act	Act	Act	Act
6	Act	Act	Act	Act	Act
7	Act	Act	Act	Act	Act
8	Act	Act	Act	Act	Act
9	Act	Act	Act	Act	Act
10	NAct (Not)	NAct	NAct	NAct	NAct
11	NAct	NAct	NAct	NAct	NAct

3.4. Partial Testing – OV2640 Camera Module Test Results

The OV2604 camera module was tested on the NodeMCU ESP32-CAM microcontroller to ensure its performance in this IoT-based fishpond security system. The OV2604 camera module can capture images with a resolution of up to 2 megapixels in image formats such as JPG and BMP. With Wi-Fi connectivity support, the ESP32-CAM can wirelessly communicate with the Telegram application and process data in real-time. The tests are shown in Figure 9 to Figure 17, and the test results are shown in Table 4. **Remarks: C means entity Capture, and NC means Entity Not Captured.**

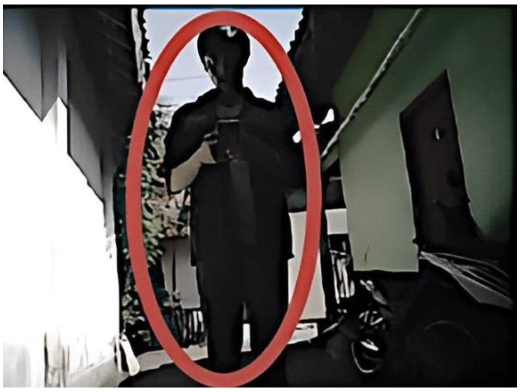

Figure 9. Capturing a human entity at a distance of 1 meter

Figure 10. Capturing a human entity at a distance of 5 meters

Figure 11. Capturing a human entity at a distance of 10 meters

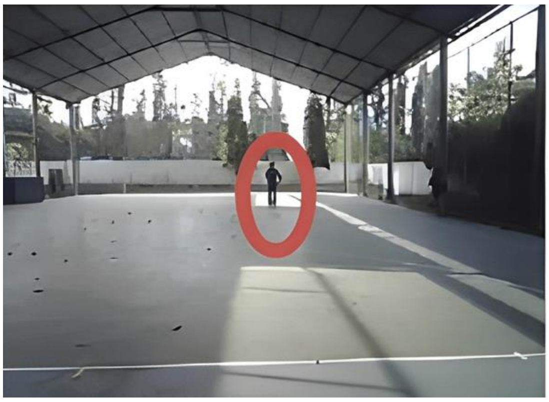

Figure 12. Capturing a human entity at a distance of 20 meters

Figure 13. Capturing a human entity at a distance of 30 meters

Figure 14. Capturing a human entity at a distance of 40 meters

Figure 15. Capturing a human entity at a distance of 50 meters

Figure 16. Capturing a human entity at a distance of 60 meters

Figure 17. Capturing a human entity at a distance of 70 meters

Table 4. OV2604 Camera Test Results

Entity Distance (m)	Information
1	C (Captured)
5	C
10	C
20	C
30	C
40	C
50	C
60	C
70	NC (Not Captured)

The OV2640 camera was tested by installing it on a 1.7-meter-high pole because it can detect human entities and adequately take pictures of them at that height. The test was carried out by taking pictures of human entities from 1 to 70 meters. The camera can detect and draw images of entities at 1 to 60 meters, with the image size getting smaller as the distance increases. At 70 meters, the camera cannot detect and take pictures of entities.

On the other hand, the image quality resulting from the camera capture highly depends on the light intensity conditions at the time of shooting. It is challenging when the camera takes pictures in limited lighting conditions, such as at night. The results of these tests also provide new knowledge about the potential use of the OV2640 camera module in IoT-based fishpond security systems, including its limitations.

The area of the protected fishpond is 7x7 square meters, and the OV2640 camera module can capture images of human entities at 1 to 60 meters, with **optimal capture at between 1 and 10 meters**. Paying attention to this, the OV2640 camera module is an alternative for fishpond security systems.

3.5. Partial Testing – NodeMCU ESP32-CAM Microcontroller Test Results

The tests carried out include testing the connection of the NodeMCU ESP32-CAM to the Internet via Wi-Fi and the connection of the NodeMCU ESP32-CAM with the Telegram application. In the first type of test, the NodeMCU ESP32-CAM will bring up a list of available Wi-Fi networks to choose from to log in to the Wi-Fi Manager Web IP Address to connect the NodeMCU ESP32-CAM to an available Wi-Fi network. The display on the Wi-Fi Manager Web IP Address can be seen in Figures 18 and 19.

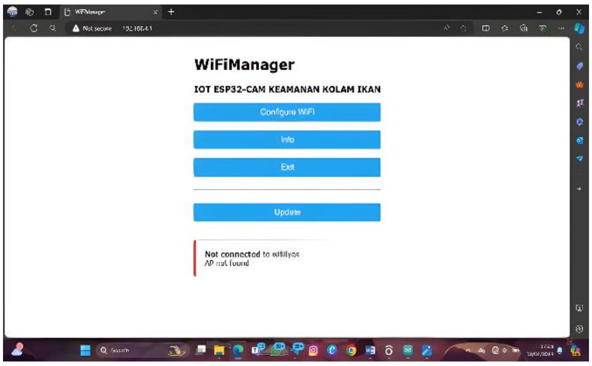

Figure 18. Wi-Fi Manager IP Address web view

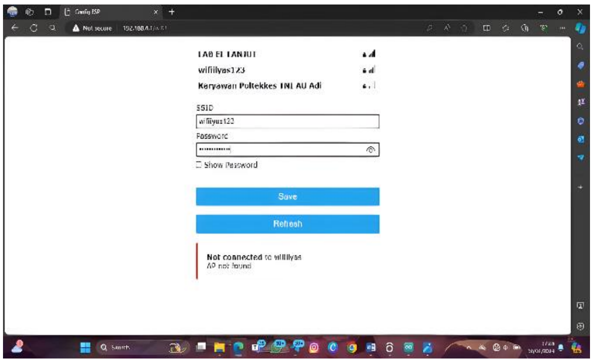

Figure 19. IP Address Web view for connection to Wi-Fi

The Configure Wi-Fi menu will appear after successfully logging in to the Web IP Address. Continue by entering the selected Wi-Fi name along with the Password and then select the save menu so that the NodeMCU ESP32-CAM is connected to the Internet to work with the IoT mechanism. Once the NodeMCU ESP32-CAM is connected to the Internet, a "Successfully Connected to Wi-Fi Network" message will appear on the Arduino IDE monitor. On the other hand, the NodeMCU ESP32-CAM will also send a "Successfully Connected to Wi-Fi Network" message to the Telegram Bot, as shown in Figure 20. This message is shown on the Telegram display on the smartphone.

Table 5. Connection between components of the fishpond security system

Instruction	Remarks
`/start`	Start the system so that it can be controlled through the Telegram application
`/photo`	Take pictures around the fishpond
`/flashon`	Turning on the Light Emitting Device (LED) flash
`/flashoff`	Turn off the LED flash

Figure 20. Telegram view to control the system

This Telegram Bot application will receive messages and images from the NodeMCU ESP32-CAM if the HB100 Radar sensor detects the movement of human entities around the fishpond. Telegram Bot also controls or commands the fishpond security system to take pictures. Figure 21 shows how to give commands to turn the NodeMCU ESP32-CAM Light Emitting Device (LED) Flash on or off, while the list of commands is shown in Table 5. LED Flash helps the camera capture better photos, especially in a dark environment.

3.6. *Integrated Test Results*

Integrated testing of the fishpond security system is carried out on all system components, ranging from the input block and process block to the output block, and the results are checked. The test began with the HB100 Radar sensor that detects the movement of human entities; the Buzzer sounds the alarm, the OV2604 camera takes photos, the NodeMCU ESP32-CAM communicates with the Telegram Bot to send messages and photos, Telegram receives the message, the user requests a photo until the user accepts the photo. Testing also ensures that communication between the two microcontrollers is running as designed.

In the test, one of our research members, shown in Figure 20, was placed as a human entity approaching a fishpond intending to commit an act of fish theft. When this human entity is within range, the HB100 radar sensor detects its presence and automatically triggers Buzzer. At the same time, the Arduino Uno R3 sends a trigger signal to the NodeMCU ESP32-CAM to activate the OV2640 camera module and take a picture of the human entity detected by the HB100 radar sensor.

The HB100 radar sensor carried out the test conditions with a maximum distance of 9 meters for human entities. The test results show that this security system can operate automatically and respond according to commands, such as sending messages and images through the Telegram application. The results of the integrated system testing are shown in Table 6. **Remarks for each part of the system are as follows.**

Sensor Radar where D means entity Successfully Detect and F means entity Failed to Detect; Buzzer where Act means Successfully Activated, and NAct means Not Activated; Camera where A means Activated and NA means Not Activated; Telegram Notification where DRx means Successfully Received and NDRx means Not Received; Photo Notification where PRx means Photo Successfully Requested and Received, and NPRx means Photo Not Successfully Requested and Received Delivered; Flash, where Working (W) means Instruction Successfully Delivered to NodeMCU ESP32-CAM, which causes the Flash to be activated, and Now Working (NW) means Instruction Not Successfully Delivered to NodeMCU ESP32-CAM, which causes the Flash not to be activated.

Figure 21. Telegram view to control the system on the integrated system test

The results of the tests in Table 6 show that the implemented IoT-based fishpond security system and NodeMCU ESP32-CAM have been operating as designed. The HB100 Radar sensor successfully detects the movement of human entities at its maximum distance; the NodeMCU ESP32-CAM connects the system to the internet network and sends images or photos and notification messages to the Telegram Bot App. From the results in Table 6, it can be concluded that **all components of the fishpond security system performed 100% well up to 9 meters, which is enough to protect the fishpond with an area of 7x7 meter squares** as recapitulated in Figure 22. On the other hand, even though at a distance above 9 meters the performance of this security system generally reaches 88%, but notifications will not be sent to the Telegram application because the Radar sensor cannot detect the presence of human entities at that distance.

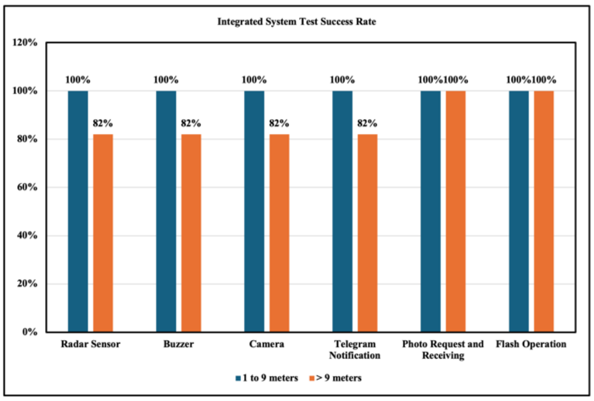

Figure 22. Telegram view to control the system on the integrated system

Table 6. Results of Integrated System Tests

Entity Distance (m)	Radar Sensor	Buzzer	Camera	Telegram Notification	Photo Request and Receiving	Flash Operation
1	Detect (D)	Activated (Act)	Activated (A)	Received (DRx)	Requested and received (PRx)	Working (W)
2	Detect	Activated	Activated	Received	Requested and received	Working
3	Detect	Activated	Activated	Received	Requested and received	Working
4	Detect	Activated	Activated	Received	Requested and received	Working
5	Detect	Activated	Activated	Received	Requested and received	Working
6	Detect	Activated	Activated	Received	Requested and received	Working
7	Detect	Activated	Activated	Received	Requested and received	Working
8	Detect	Activated	Activated	Received	Requested and received	Working
9	Detect	Activated	Activated	Received	Requested and received	Working
10	Failed to Detect (F)	Not Activated (NAct)	Not Activated (NA)	Not Received (NDRx)	Requested and received	Working
11	Failed to Detect	Not Activated	Not Activated	Not Received	Requested and received	Working

On the other hand, the Arduino Uno R3 successfully processed the data of the detection results of the HB100 Radar sensor and gave a command to turn on the Buzzer based on the detection results. At the same time, the Arduino Uno R3 successfully sends a trigger signal to the NodeMCU ESP32-CAM to give commands to the OV2640 camera module to take a picture or photo of the detected human entity and send a notification to the fishpond owner via the Telegram application in real-time as shown in Figures 20 and 21.

Figure 21 and Figure 23 also show that the NodeMCU ESP32-CAM successfully connects with the Arduino Uno R3. This is shown from the information received about the detection of human entities and the information that the Buzzer has been activated, which is displayed on the Telegram display. In addition to these two pieces of information, the trigger signal from the Arduino Uno R3 activates the NodeMCU ESP32-CAM camera to capture images of the human entity. To improve the quality of the camera capture, the `/flashon` command will activate the LED flash to make the image captured by the camera of better quality. To deactivate the LED flash, just write the the `/flashoff` command.

Figure 23. Telegram view to control the system on the integrated system test

4. Conclusions

The IoT-based fishpond security system and NodeMCU ESP32-CAM microcontroller designed and implemented in this study can be an alternative means of helping fishpond owners conduct remote surveillance to prevent and reduce fish theft, especially at night. This security system is quite simple but can significantly reduce both material and financial losses for fishpond owners. In our case, it performs 100% well up to 9 meters, enough to protect the fishpond with an area of 7x7 meter squares. For a freshwater fishpond of 7x7 square meters, the HB100 radar motion sensor and vision sensor in the form of an OV2640 camera have been able to cover the entire fishpond area. The NodeMCU ESP32-CAM microcontroller, which is the primary controller of this system, is supported by an Arduino Uno R3 microcontroller that handles two HB100 radar sensors and a buzzer, which is the right combination so that all system functions can operate as conceptualized. The availability of Wi-Fi facilities on the NodeMCU ESP32-CAM provides remote monitoring capabilities through smartphones using the Telegram application. The possibility of fish theft is carried out from different directions, such as from the North and South, even though there are embankments; this fishpond security system can be developed by adding three HB100 radar sensors so that the coverage of detecting human entity movements will be full 360^0 where there will be overlap coverage between one sensor and another. For the vision sensor, three more cameras can be added to connect to an additional microcontroller to ease the task of the NodeMCU ESP32-CAM as the central controller. Another feature that can be added is video captures that are recorded or live-streamed directly to the fishpond owner and can be evidence of theft that occurred.

References

[1] I. R. N. Adnan S, H. J. Wantasen, L. Pangemanan, N. P. L. Sinjal, S. Lantu, and D. Wungkana. Salindeho, “Strategi Pengembangan Budidaya Perairan di Kecamatan Ratahan Timur, Kabupaten Minahasa Tenggara,” *Budidaya Perairan 2022*, vol. 10, no. 1, pp. 31–41, 2022.

[2] K. Sidoarjo, J. Santoso, F. Rachmatullah, and M. Dicky, “Pembangunan Kolam Ikan Koi di Desa Simoketawang, Kecamatan Wonoayu, Kabupaten Sidoarjo,” in *Seminar Nasional Pengabdian kepada Masyarakat 2022*, 2022, pp. 189–193. [Online]. Available: https://conferences.unusa.ac.id/index.php/snpm/article/view/960

[3] D. Sanjaya and I. Badarina, “Pembuatan Kolam Pembudidayaan Lele Untuk Memotivasi Masyarakat Guna Meningkatkan Perekonomian yang Terhambat Akibat Pandemi Covid-19 di RT 1 RW 3 Desa Bukit Tinggi,” *Tribute: Journal of Community Services*, vol. 2, no. 2, pp. 99–107, 2021, doi: 10.33369/tribute.v2i2.18701.

[4] S. A. Obado, "IoT Based Real-time Fish Pond Water Quality Monitoring Model,” 2020. [Online]. Available: https://su-plus.strathmore.edu/handle/11071/6710

[5] F. Rizal, G. S. Santyadiputra, and G. A. Pradnyana, “Prototype of Water Quality Monitoring for Grouper Fish Pond Based on Microcontroller Arduino,” *International Journal of Natural Science and Engineering*, vol. 5, no. 2, pp. 77–86, Sep. 2021, doi: 10.23887/ijnse.v5i2.32702.

[6] I. A. Permana, T. Handayani, and B. G. Pratama, “Sistem Pengendalian dan Pemantauan Kolam Bioflok Berbasis IoT (Internet of Things) dengan Google Firebase,” *Jurnal Mahasiswa Teknik Elektro*, vol. 22, no. 1, pp. 1–12, 2024, [Online]. Available: https://journal.itny.ac.id/index.php/JMTE/article/view/4868

[7] R. A. Riantama and T. Fatimah, “Sistem Monitoring dan Pemberian Pakan Ikan Otomatis Menggunakan ESP32Cam Berbasis Web,” *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, no. September, pp. 724–733, 2022.

[8] I. Sayekti *et al.*, “Penerapan Teknologi Pemberi Pakan Ikan Berbasis Internet of Things (IoT) pada Usaha Budidaya Ikan di Desa Nyatnyono,” *Prosiding Seminar Hasil Penelitian dan Pengabdian Masyarakat*, vol. 4, no. 1, pp. 897–911, 2022.

[9] R. Maulana, K. Kusnadi, and M. Asfi, “Sistem Monitoring dan Controlling Kualitas Air Serta Pemberian Pakan Pada Budidaya Ikan Lele Menggunakan Metode Fuzzy, NodeMCU dan Telegram,” *ITEJ (Information Technology Engineering Journals)*, vol. 6, no. 1, pp. 53–64, 2021, doi: 10.24235/itej.v6i1.57.

[10] A. Roza and P. Jaya, “Penerapan Teknologi Berbasis Internet of Things (IoT) Untuk Pengelola Peternakan Ikan Air

Tawar," *Voteteknika (Vocational Teknik Elektronika dan Informatika)*, vol. 11, no. 1, p. 71, Mar. 2023, doi: 10.24036/voteteknika.v11i1.121214.

[11] Sumarno, H. S. Setiawan, V. H. Valentino, and A. S. Putra, "Penerapan Internet of Think (IoT) pada Transportasi Cerdas," *Seminar Nasional Pengaplikasian Telematika (SINAPTIKA)*, vol. 1, no. 1, pp. 94–98, 2021.

[12] N. Ya'acob, N. N. S. N. Dzulkefli, A. L. Yusof, M. Kassim, N. F. Naim, and S. S. M. Aris, "Water Quality Monitoring System for Fisheries using Internet of Things (IoT)," *IOP Conf Ser Mater Sci Eng*, vol. 1176, no. 1, p. 012016, 2021, doi: 10.1088/1757-899x/1176/1/012016.

[13] F. P. E. Putra, U. Ubaidi, R. N. Saputra, F. M. Haris, and S. N. R. Barokah, "Application of Internet of Things Technology in Monitoring Water Quality in Fishponds," *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 1, pp. 356–361, 2024, doi: 10.47709/brilliance.v4i1.4231.

[14] D. Listya, D. Fitria, E. Triyana, and N. Asni, "Pemanfaatan Wireless Fidelity (WiFi) di Perguruan Tinggi sebagai Fasilitas Kegiatan Akademik Mahasiswa," *Biodik*, vol. 9, no. 1, pp. 31–39, 2023, doi: 10.22437/bio.v9i1.19534.

[15] Tri Sulistyorini, Nelly Sofi, and Erma Sova, "Pemanfaatan Nodemcu ESP8266 Berbasis Android (Blynk) sebagai Alat Alat Mematikan dan Menghidupkan Lampu," *Jurnal Ilmiah Teknik*, vol. 1, no. 3, pp. 40–53, 2022, doi: 10.56127/juit.v1i3.334.

[16] W. Yulita and A. Afriansyah, "Alat Pemantau Keamanan Rumah Berbasis ESP32-Cam," *Jurnal Teknologi dan Sistem Tertanam*, vol. 3, no. 2, pp. 2–10, 2022, doi: 10.33365/jtst.v3i2.2197.

[17] H. A. Kusuma, S. B. Wijaya, and D. Nusyirwan, "Sistem Keamanan Rumah Berbasis ESP32-Cam dan Telegram sebagai Notifikasi," *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 8, no. 1, p. 30, 2023, doi: 10.32897/infotronik.2023.8.1.2291.

[18] D. Danang, E. Fredyan, and I. S. Suasana, "Prototype Alat Keamanan Rumah Internet of Things (Iot) Berbasis Nodemcu ESP8266 dengan ESP32 Cam dan Kombinasi Sensor Menggunakan Telegram," *Unitech*, vol. 1, no. 1, pp. 1–9, 2022.

[19] A. Syuhada, N. Zairi, A. A. Basari, and M. Noor, "Control and Monitoring of Office Electrical Appliances Using IoT and an ESP32 Microcontroller," 2023, pp. 1–5.

[20] Y.- Yanti, "Sistem Monitoring Keamanan Kantor Berbasis Android Menggunakan Modul Wireless ESP-01," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3734.

[21] I. Tantowi, Sunardi, and F. Santoso, "Sistem Keamanan Laboratorium Berbasis IoT dengan ESP32 di SMP N 1 Jangkar," *G-Tech: Jurnal Teknologi Terapan*, vol. 7, no. 4, pp. 1254–1261, 2023, doi: 10.33379/gtech.v7i4.3030.

[22] N. Citra Dewi, T. Sutabri, and F. Putrawansyah, "Analisis Penyadapan pada Telegram dengan Network Forensic," *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 2, p. 183, 2023, doi: 10.26798/jiko.v7i2.789.

[23] B. Dewantara, I. Sulistiyowati, and J. Jamaaluddin, "Automatic Fish Feeder and Telegram Based Aquarium Water Level Monitoring," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 1, pp. 98–107, 2023, doi: 10.12928/biste.v5i1.7575.

[24] S. Suwarmiyati, "Development Of A Patient Control And Monitor System Using Esp32-Cam And Android," *Jurnal Teknologi Elekterika*, vol. 21, no. 1, pp. 9–17, 2024, doi: 10.31963/elekterika.v21i1.4777.

[25] M. R. Fachri and I. Firja, "Development of Home Security System Using Motion Sensor and Magnetic Sensor with ESP-32 Cam Based on Telegram," in *The 4 Education, Sciences and Technology International Conference 2024*, 2024, pp. 345–373.

[26] W. S. Putra and A. Setyawan, "Room Security System Design using ESP32 CAM with Fuzzy Algorithm," *Mobile and Forensics*, vol. 3, no. 2, pp. 66–74, 2021, doi: 10.12928/mf.v3i2.5554.

[27] W. S. Nuankaew, B. Tupaso, P. Nasa-Ngium, and P. Nuankaew, "Internet of Things for Monitoring Fish Cage Water Quality," *International Journal of Engineering Trends and Technology*, vol. 71, no. 6, pp. 212–220, Jun. 2023, doi: 10.14445/22315381/IJETT-V71I6P222.