Design and Performance Evaluation of an IoT-Enabled Feed Level Alert System for Smart Fish Feeders

Micko Tomas[1]*, Baharuddin Baharuddin[1], Illa Aryeni[2], Elsi Alfionita Syawal[3], and Muhammad Kodrat[4]

[1]Electrical Engineering Study Program, Electrical Engineering Department, Universitas Andalas, Padang, Indonesia
[2]Electronics Engineering Technology Study Program, Electrical Engineering Department, Politeknik Negeri Batam, Batam, Indonesia
[3]Electrical Engineering Study Program, Electrical and Informatics Engineering Department, Universitas Jambi, Jambi, Indonesia
[4]Electrical Engineering Study Program, Electrical Engineering Department, Universitas Telkom, Bandung, Indonesia

*Email: micko.tomas@eng.unand.ac.id

Received on 05-11-2025 | Revised on 03-06-2026 | Accepted on 26-06-2026

***Abstract*—The Feed Level Alert system on the Smart Fish Feeder was designed to monitor fish feed availability in real-time using an Internet of Things approach. Unlike previous research that focused solely on passive monitoring, this system presents a practical implementation equipped with automatic notifications when the fish feed level status reaches a set threshold. The HC-SR04 ultrasonic sensor is used to measure the distance between the tank lid and the feed surface, while the DHT-11 sensor monitors temperature and humidity to maintain feed quality. The ESP32 microcontroller is responsible for processing data and sending alert notifications via Telegram Bot with an average response time of 1,8 seconds. Test calibration results show a detection accuracy of 96,4% and a system reliability of 99,3%. The portable prototype design allows installation without significant mechanical modifications. With its integrative capabilities, this system shows high potential as an efficient, adaptive, and sustainable innovative aquaculture.**

Keywords: ESP32, Feed Level Alert, IoT, Smart Aquaculture, Ultrasonic Sensor.

I. INTRODUCTION

The development of Internet of Things (IoT) technology has had a significant impact on the automation of modern aquaculture systems. The integration of sensors, wireless networks, and microcontrollers allows for real-time, efficient, and adaptive monitoring of water conditions and fish feed availability [1], [2], [3]. In aquaculture, feeding is a crucial aspect in determining productivity and operational efficiency. Inaccurate feed management can lead to suboptimal fish growth, increased organic waste, and decreased water quality [4], [5]. Therefore, an intelligent system is needed that can accurately monitor feed levels and provide early warnings before feed stocks run out. Various approaches have been [6], [7] developed to detect feed levels, such as weight sensors (load cells), computer vision-based cameras, and ultrasonic sensors [8], [9], [10]. Load cell-based systems have limitations due to the need for regular calibration and sensitivity to humidity, while camera-based systems require stable lighting conditions and high computational requirements [11], [12]. Ultrasonic sensors are an efficient alternative because they can measure distance without direct contact with the feed material [13]. However, most developed monitoring systems are still passive and lack automatic warning mechanisms when the feed reaches the minimum limit. Furthermore, most studies have not integrated temperature and humidity monitoring, which impact feed quality [14].

Implementation of IoT in aquaculture automation shows excellent potential for realizing efficient, adaptive, and sustainable fish farming systems. Developed systems generally focus on water quality monitoring and automatic feed control, but are still limited in terms of real-time notification, communication performance analysis, and comprehensive system accuracy measurement.

Several studies have emphasized the importance of integrating reliable sensors and communication devices to support fish farming efficiency. However, systems that provide proactive notifications based on instant messaging platforms, such as Telegram, are still rare [15]. This study designed and evaluated an IoT-based Feed Level Alert system integrated with a Smart Fish Feeder.

Although numerous IoT-based aquaculture monitoring systems have been reported, several limitations remain unresolved. Previous studies mainly focused on automatic feeding mechanisms, water quality monitoring, or feed-level estimation using load cells and computer vision techniques. These approaches often require frequent calibration, substantial computational resources, stable lighting conditions, or additional hardware components, thereby increasing system complexity and operational costs.

Furthermore, most existing systems provide only passive monitoring via dashboards or mobile applications, requiring users to inspect feed conditions manually. Real-time proactive notification mechanisms integrated with lightweight messaging platforms remain limited. In addition, environmental parameters affecting feed storage conditions, such as temperature and humidity, are often not monitored simultaneously.

Therefore, a research gap exists in developing a low-cost, lightweight, and real-time feed monitoring system that integrates feed-level detection, environmental monitoring, and automatic notification within a single IoT architecture. This study addresses this gap by integrating an HC-SR04 ultrasonic sensor, a DHT-11 environmental sensor, an ESP32 microcontroller, and a Telegram Bot notification system.

The system utilizes an HC-SR04 ultrasonic sensor to detect the feed level, a DHT-11 sensor to monitor temperature and humidity, and an ESP32 microcontroller as the data processing and wireless communication hub. When the feed level reaches a critical threshold (10 cm from the bottom of the tank), the system automatically sends an alert via Telegram Bot to the user with an average response time of 1,8 seconds. Test results showed a detection accuracy of 96,4% and a system reliability of 99,3%, indicating stable and responsive performance.

The main contributions of this research include: (1) the development of an automated feed monitoring system with real-time IoT-based notifications, (2) the integration of temperature and humidity monitoring to maintain feed quality, (3) system performance evaluation based on accuracy, latency, and reliability parameters, and (4) a portable design that allows for application to various Smart Fish Feeder sizes without significant mechanical modifications. Thus, this system has the potential to enhance the implementation of innovative aquaculture, which is more efficient, adaptive, and sustainable.

II. METHOD

A. Study Research Design

This research uses a design and performance evaluation approach, focusing on the development of an Internet of Things (IoT)-based Feed Level Alert subsystem. This approach was chosen because the research aims not only to produce hardware and software designs but also to quantitatively measure system performance through parameters such as feed level detection accuracy, notification latency, and operational reliability.

The system design was carried out through several structured stages. The first stage is system design, which encompasses requirements identification, component selection, and the design of hardware and software architecture. The second stage is prototype implementation, where the system was developed using an ESP32 microcontroller as the central control unit connected to an HC-SR04 ultrasonic sensor and a DHT-11 environmental sensor. The third stage is testing and validation, which involves collecting data on feed surface distance, temperature, and humidity under various conditions to evaluate system reliability.

The final stage is results analysis, where all test data is evaluated to determine the overall level of system accuracy, latency, and reliability. Overall, this research focuses on the Feed Level Alert system, a stand-alone feed availability monitoring system that can be integrated with the more complex Smart Fish Feeder system. This subsystem is designed to provide automatic alerts to users via Telegram Bot when the feed level reaches a critical threshold (10 cm from the bottom of the tank). The portable and flexible design allows the system to be installed in various tank types and sizes without significant mechanical modifications.

The research flowchart is shown in Figure 1, which illustrates the process sequence from design to evaluation of results. The diagram in Figure 1 above shows a concise and focused workflow for the Smart Fish Feeder Feed Level Alert system research. The process begins with the system design phase, which includes identifying needs, selecting sensors, and designing the hardware and software architecture. The design results are then implemented in the system implementation phase, where the ESP32 microcontroller is programmed and integrated with the HC-SR04 ultrasonic sensor, the DHT-11 sensor, and the Telegram Bot platform for sending notifications.

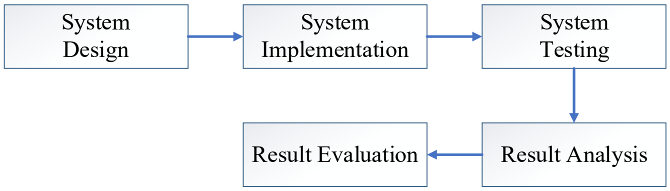

Figure 1. Research Flowchart

Next, system testing is conducted to ensure functionality and reliability, including measuring the distance to the feed surface, temperature, humidity, and message response time. The test data is analyzed in the results analysis phase to assess system accuracy, latency, and reliability. The final stage, the evaluation of the results, is used to determine the overall performance of the system and provide recommendations for improvements and potential further development. Through this process, the research ensures that the system is developed using a systematic and measurable scientific approach.

B. System Architecture

The Feed Level Alert system is an IoT-based subsystem of the Smart Fish Feeder designed for real-time monitoring of feed availability. The architecture integrates an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, a DHT-11 temperature and humidity sensor, a 5V power supply, and the Telegram Bot API for wireless communication.

The ESP32 functions as the central controller, processing data from the sensors. The HC-SR04 measures the distance between the tank lid and the feed surface to determine the remaining feed level, while the DHT-11 monitors ambient temperature and humidity that may affect feed quality. Sensor data are compared to a predefined threshold of 10 cm from the tank bottom.

Figure 2 illustrates the communication architecture of the Smart Fish Feeder IoT system, with a primary focus on the Feed Level Alert subsystem. This subsystem detects and informs users about feed levels in real-time via an Internet of Things (IoT) connection. On the right side of the diagram, the Feed Level Alert subsystem is a critical element directly integrated with the Smart Fish Feeder. This system works by reading data from ultrasonic and environmental sensors to monitor the level and condition of feed in the tank. When the system detects that the feed has reached the minimum threshold (approximately 10 cm from the bottom of the tank), the Feed Level Alert module automatically sends a warning signal. This process triggers a digital notification, which is then forwarded to the user via the Telegram Bot API platform. A low-feed condition is defined as a remaining feed height of ≤10 cm measured from the tank bottom. Since the total tank height is 95 cm, this condition corresponds to a measured sensor distance of ≥85 cm from the ultrasonic sensor.

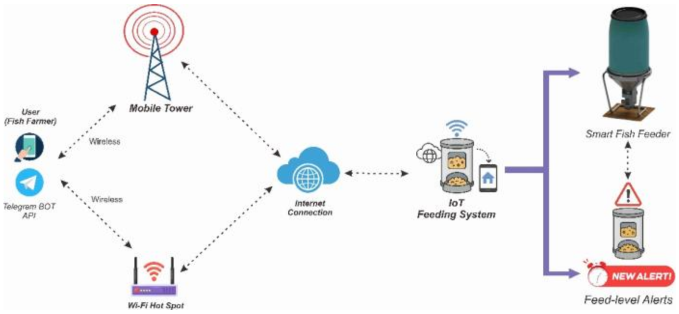

Figure 2. Feed Level Alert System Architecture

The measurement data from the Feed Level Alert is sent via a Wi-Fi router to the IoT Feeding System, which then forwards it to the Internet Cloud. From there, the information is communicated wirelessly to the user via the cellular network (Mobile Tower). This mechanism enables fish farmers to receive notifications directly through the Telegram app, eliminating the need for manual checks on their feed tanks. The main advantage of this architecture lies in the functional integration between the Feed Level Alert subsystem and the Smart Fish Feeder system. Feed Level Alert functions not only as a passive detector but also as an active early warning module, ensuring a continuous feed supply and reducing the risk of delayed fish feeding. Thus, this system improves feed management efficiency, maintains cultivation quality, and supports the implementation of intelligent, adaptive, and sustainable innovative aquaculture.

C. *Hardware Design*

1) *System Configuration and Component Connections*

The main hardware used consists of an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, and a DHT-11 temperature and humidity sensor. The ESP32 serves as the central processing and wireless communication hub. The HC-SR04 sensor measures the distance between the sensor position and the feed surface, while the DHT-11 sensor monitors environmental conditions in the feed storage tank.

The connections between the components are configured to ensure stable data communication with minimal power consumption. The pin configuration and connections between the elements are presented in Table 1.

TABLE I
CONFIGURATION OF ESP32 PINS WITH SENSORS

Component	Pin ESP32	Pin Component
Sensor HC-SR04	GPIO 14	Trig
Sensor HC-SR04	GPIO 27	Echo
Sensor DHT-11	GPIO 4	Data
Power Supply	VIN (5V)	VCC
Ground	GND	GND

Figure 3 illustrates the overall architecture of the IoT-based Feed Level Alert system designed for real-time monitoring of fish feed levels. The system integrates two main sensors connected to an ESP32 microcontroller, namely the HC-SR04 ultrasonic sensor and the DHT-11 temperature sensor. The ultrasonic sensor is mounted at the top of the feed container to measure the distance between the sensor and the surface of the remaining feed. When the feed height drops below the predetermined threshold of 10 cm, the system identifies a low-feed condition and triggers an alert. Meanwhile, the DHT-11 sensor continuously monitors the ambient temperature around the feeder to support environmental data logging.

The ESP32 serves as the central controller that processes the sensor data and transmits it wirelessly through a Wi-Fi connection to the Telegram Bot API, enabling the user (farmer) to receive instant notifications on their smartphone. The entire system is powered by a 5V DC supply, ensuring stable operation of the sensors and microcontroller. This configuration allows the smart fish feeder to provide real-time monitoring, automatic feed-level alerts, and improved efficiency in aquaculture feed management.

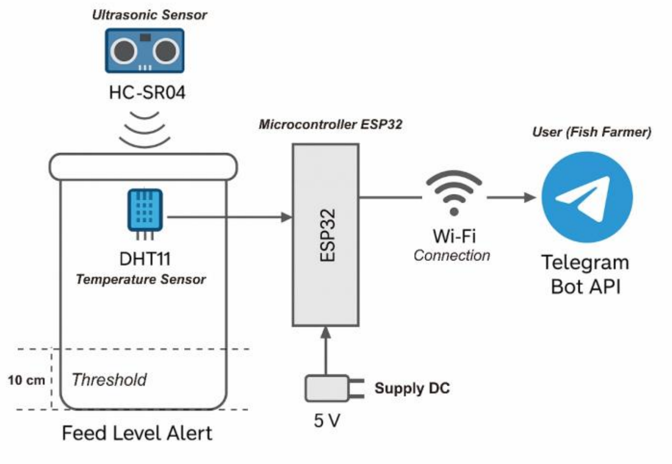

Figure 3. Feed Level Alert System Schematic

2) *Mechanical Design and Sensor Placement*

The system's mechanical design was focused on enhancing measurement efficiency and improving device portability. The HC-SR04 sensor was mounted on the feed tank lid at a total height of 95 cm. The feed threshold was set at a distance of 85 cm from the sensor, or approximately 10 cm from the tank bottom. This position was chosen to allow the system to detect low feed levels before the tank is empty, allowing the user time to refill.

The DHT-11 sensor was placed on the top side of the tank, with a small vent to allow for air circulation inside the tank, thereby enabling representative temperature and humidity measurements. Both sensors were connected to an ESP32 control board housed in a moisture-proof protective container near the tank. Detailed physical parameters for sensor placement are shown in Table 2.

Table 2 presents the primary physical parameters employed in sensor placement for the Feed Level Alert system. The feed tank height of 95 cm was set as the baseline measurement reference, while the detection threshold at 85 cm was used to indicate low feed levels (approximately 10 cm remaining from the tank bottom). The ±15° detection angle of the HC-SR04 sensor ensures that the ultrasonic wave reflection remains within the effective path without interference from the tank walls. Temperature and humidity parameters are measured by the DHT-11 sensor in the range of 20–40 °C and 30–80% RH to maintain the validity of environmental data that affect feed quality. Determining these parameters aims to obtain accurate, stable, and representative measurement results for the system's operational conditions.

TABLE II
PHYSICAL PARAMETERS OF SENSOR PLACEMENT

Parameter	Value	Information
Feed tank height	95 cm	From the bottom to the tank lid
Detection threshold	85 cm	Minimum feed limit (10 cm remaining)
HC-SR04 detection angle	±15°	Effective ultrasonic reflection angle
Operating temperature	20–40 °C	Optimal DHT-11 measurement range
Ambient humidity	30–80% RH	DHT-11 measurement range

3) Initial System Calibration

The calibration phase is conducted to ensure that the distance measurements from the HC-SR04 sensor accurately reflect the actual conditions inside the tank. The calibration procedure involves first measuring the distance from the empty tank to establish a reference of "0 cm" under conditions without feed. Afterward, feed is added in stages (20 cm, 40 cm, 60 cm, and 80 cm), and the sensor readings are compared with manual measurements using a digital ruler. The average difference is used to determine the calibration correction factor (Ck), as shown in the following equation:

$$C_k = \frac{|D_{actual} - D_{sensor}|}{D_{sensor}} x100\% \tag{1}$$

The calibration results showed an average error of less than 3%, which is still within the tolerance limits of an ultrasonic-based IoT system. After the calibration process was complete, the system was configured to send automatic notifications via the Telegram Bot API if the detection distance exceeded a predetermined threshold.

4) Hardware Component Specifications

Table 3 presents the primary technical specifications of the components utilized in the Feed Level Alert system. The ESP32 microcontroller serves as the data processing center and wireless communication interface. The HC-SR04 sensor is used to detect the distance between the sensor and the feed surface, while the DHT-11 sensor measures the temperature and humidity inside the tank. A 5V DC power supply provides a stable power source for the entire circuit, while a protective acrylic housing protects the device from moisture and splashes. This set of components forms an integrated, portable system suitable for IoT applications in fish farming environments.

TABLE III
SPECIFICATIONS OF COMPONENTS OF FEED LEVEL ALERT

Component	Technical Specifications
ESP32 Devkit v1	Dual-core MCU 240 MHz, 4 MB Flash, integrated Wi-Fi & Bluetooth, 3,3–5 V DC
HC-SR04	Operating voltage 5 V DC, measuring range 2–400 cm, accuracy ±3 mm, detection angle 15°
DHT11	Operating voltage 3,3–5 V DC, temperature range 0–50 °C, humidity 20–90% RH
DC Power Supply	Output 5 V, stable current 2 A
Protective Enclosure	3 mm acrylic material resistant to moisture and corrosion

5) Hardware Functionality Analysis

All components are assembled into a single, modularly connected system, making it easy to maintain and expand. The ESP32 microcontroller manages sensor data acquisition at specified intervals (e.g., every 10 seconds), then performs simple processing to compare the distance value to a threshold. The system is designed to remain stable in the presence of voltage fluctuations of ±5% and can automatically reset if the Wi-Fi connection is lost.

Overall, this hardware design meets the criteria for an IoT monitoring system: accuracy, reliability, power efficiency, and portability. Its modular design allows for further integration with the Smart Fish Feeder system to create a brilliant and sustainable feeding automation solution.

D. Software Design

1) Program Logic Flow

The program logic flow is a structured representation of the workflow of the Feed Level Alert system on the Smart Fish Feeder. These logical steps ensure that each process, from sensor readings to notification delivery, runs systematically and efficiently. The entire process is executed automatically using an ESP32 microcontroller, which serves as a control center for data acquisition, decision logic processing, and data communication over a Wi-Fi network.

The system's workflow begins with device initialization and network connection, during which the ESP32 performs the initial configuration of the sensor pins and verifies the availability of a Wi-Fi connection. Once the connection is successful, the system enters the sensor data reading phase, which includes capturing distance values from the HC-SR04 ultrasonic sensor and temperature and humidity data from the DHT-11 sensor.

These readings are processed to calculate the remaining feed level based on the following equation:

$$H_p = H_t - D \tag{2}$$

Where H_p denotes the actual feed height, H_t represents the total tank height (95 cm), and D is the distance measured by the HC-

SR04 sensor. If the calculation result indicates that H_p is below the threshold limit, the system identifies a low feed level condition and immediately executes the alert transmission process through the Telegram Bot API. The notification message is automatically sent to the user, informing them that the feed needs to be refilled. If the feed height remains above the threshold, the system returns to standby mode and performs periodic re-measurement at a defined interval (e.g., every 10 seconds). This cycle continues continuously as long as the system is active and connected to the network.

2) *Feed Level Alert Program Logic Flowchart*

The logic flowchart in Figure 4 systematically illustrates the working sequence of the Feed Level Alert system, implemented on an ESP32 microcontroller, to automatically and in real-time monitor fish feed levels. The process begins with system initialization and Wi-Fi connection, where the ESP32 configures the pins of the HC-SR04 ultrasonic sensor and the DHT-11 environmental sensor and ensures a wireless network connection for communication with the Telegram Bot API. Once the connection is successful, the system enters the sensor reading phase, where the HC-SR04 measures the distance between the sensor and the feed surface. At the same time, the DHT-11 records the temperature and humidity in the feed tank.

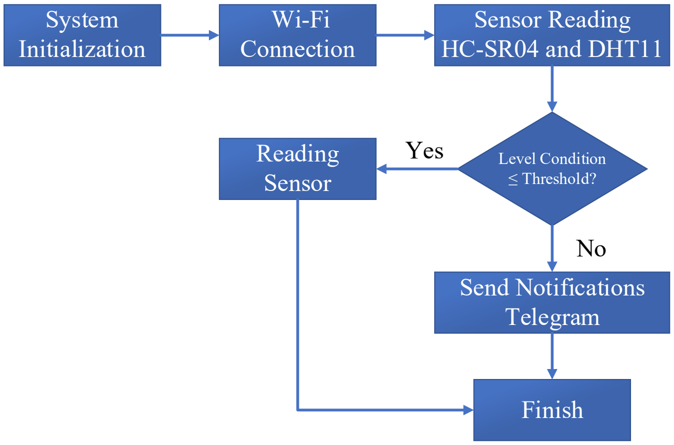

Figure 4. Feed Level Alert program logic flowchart

The data obtained from sensor readings are processed to calculate the actual feed height using the equation $H_p = H_t - D$, where $h_t$is the total tank height (95 cm) and D is the measured distance. The resulting feed height value is then compared with the threshold limit of 10 cm, which indicates a low-feed condition. If the calculation shows that $H_p < 10$ cm, the system automatically executes a digital notification command via Telegram, alerting the user that the feed needs to be refilled immediately. Conversely, if the feed level remains above the minimum threshold, no message is sent, and the system resumes the next reading cycle after a predefined delay. The entire process operates in a continuous periodic loop, ensuring uninterrupted monitoring without manual intervention.

3) *Measurement Algorithm and Decision Logic*

The system's main algorithm is based on digital signal processing from the HC-SR04 sensor. This sensor sends a trigger pulse for 10 µs and then receives a reflected pulse (echo) from the feed surface. The distance is calculated using the following equation:

$$D = \frac{t \times v}{2} \tag{3}$$

where:
D = distance between the sensor and the feed surface (cm)
t = time of ultrasonic wave reflection (µs)
v = speed of sound in air (340 m/s at 25 °C)

The variable D represents the distance between the sensor and the feed surface in centimeters (cm). This value is the primary output of the measurement algorithm and is a crucial parameter in the system's decision-making logic. In a feed level monitoring system, a small D value indicates that the feed surface is close to the sensor (still high stock), while a considerable D value indicates that the feed surface is farther from the sensor (low stock). The D value indicates that the feedstock is starting to decrease. A threshold value is set in the control system to trigger an alert or a refill notification when the distance exceeds a specified limit.

The variable t is the ultrasonic wave travel time, typically measured in microseconds (μs). This value represents the total time it takes for the ultrasonic pulse to travel from the sensor to the feed surface and return to the sensor as a reflected wave. A microcontroller (e.g., ESP32) measures this duration using a high-precision time measurement function. Because the measured time is a round-trip time (round-trip time), it covers twice the actual distance.

The variable v is the speed of sound in air. At 25°C, the speed of sound is generally assumed to be 340 m/s. This parameter is used to convert propagation time to physical distance. However, the speed of sound is affected by ambient temperature and can be calculated more accurately using the equation:

$$v = 331 + 0.6T \tag{4}$$

Where T is the air temperature in degrees Celsius; therefore, in systems requiring higher accuracy, temperature compensation can be applied to adjust the v value to environmental conditions dynamically. The divisor factor of 2 in the equation is used because the measured time t represents the round-trip travel time of the ultrasonic wave. By dividing by 2, the one-way distance between the sensor and the feed surface is obtained.

Overall, the combination of these variables forms the core of the measurement algorithm. The continuously calculated distance value D is compared with a threshold value in the decision logic block to determine system conditions, such as "Normal Level," "Feed Low," or "Refill Required," allowing the system to operate automatically and reliably in monitoring feed availability.

4) *Wi-Fi-Based IoT Communication Mechanism*

The Feed Level Alert system utilizes an Internet of Things (IoT)-based communication architecture, employing the HTTP protocol via the internal ESP32 Wi-Fi module. This communication mechanism sends measurement data from the sensor to the Telegram Bot API server in real-time, allowing users to receive automatic notifications when the feed reaches the minimum threshold.

The ESP32 acts as a client, initiating communication with an external server (Telegram API Server) over the internet. When the system detects low feed ($H_p \leq 10$ cm), the ESP32 generates an HTTP request in JSON (JavaScript Object Notation) format containing a digital notification message. This request is sent to the official Telegram endpoint using the address:
https://api.telegram.org/bot<token>/sendMessage

The message format is sent to the user's pre-registered chat ID, and the Telegram server responds with a success code indicating that the notification has been received. The ESP32 then returns to standby mode and periodically performs sensor readings without user intervention.

This communication process involves two main layers:

a. *The local communication layer*
   Which exchanges data between the HC-SR04 and DHT-11 sensors and the ESP32 microcontroller using digital signals (pulse width and serial data).
b. *The network communication layer*
   Which sends data from the ESP32 to the Telegram Bot Server using the HTTP protocol over a Wi-Fi connection.

To ensure communication stability, the system is equipped with an auto-reconnect mechanism, where the ESP32 automatically reconnects to the network if the Wi-Fi connection is lost. This allows the system to continue sending notifications even during momentary network disruptions.

During the data transmission process, each notification message contains three main elements:

a. the time of the event (timestamp),
b. the current feed level status, and
c. Environmental parameters (temperature and humidity).

This information is efficiently transmitted in JSON payload format, for example:

```
{
"chat_id": "123456789",
"text": "Feed Level Alert: Remaining feed below 10
cm.\nTemp: 29°C | Humidity: 67% | Time: 09:24:11"
}
```

This client-server communication approach ensures that all data can be accessed by users anywhere over the internet without the need for additional local servers. Furthermore, the use of the Telegram Bot API offers security benefits (SSL encryption) and ease of integration, as all connections are encrypted using the HTTPS protocol.

Communication tests demonstrated that the system could send alert messages with an average latency of 1,8 seconds and a delivery success rate of 99,3%, indicating the system's reliability in real-time IoT data transmission. With this design, the Feed Level Alert communication mechanism is considered efficient, secure, and user-oriented, while also supporting the principles of sustainability in the implementation of an IoT-based innovative aquaculture system.

5) Software Evaluation

Software evaluation was conducted to assess the functional performance, measurement accuracy, data communication reliability, and algorithm efficiency of the Feed Level Alert system. The goal was to ensure that the developed software could perform consistently, accurately, and responsively under actual operational conditions. The evaluation was conducted through functional testing, performance testing, and communication reliability testing.

a. Testing Methods and Scenarios

The testing was conducted in an IoT laboratory with a system configuration identical to the field prototype. An ESP32 Devkit v1 microcontroller was used as the central controller, connected to an HC-SR04 sensor and a DHT-11 sensor. The test tank had a total height of 95 cm, with dry pellet feed, 2–3 mm in diameter, serving as the reflective test medium.

The testing was divided into three main scenarios to represent different field conditions. Scenario 1, Normal Level: Feed > 30 cm. The system only takes periodic readings without notification. Scenario 2, Alert Level: Feed 10–30 cm. The system begins to detect distance changes with high sensitivity, but it has not yet sent a notification. Scenario 3, Low Level: Feed ≤ 10 cm. The system must trigger an automatic Telegram alert.

Each scenario was run 30 times (N = 30) to obtain valid statistical data. Testing was conducted for 1 hour, with a 10-second reading interval, and the average message transmission time was measured from the moment of detection to the time the user received the notification.

b. Test Parameters and Procedures

This section outlines the parameters and procedures used in evaluating the Feed Level Alert system software. Testing was conducted to measure system performance across various aspects, including response speed, data communication reliability, sensor measurement accuracy, and hardware resource efficiency. Each test parameter was designed to objectively and measurably assess system performance in accordance with applicable standards for real-time IoT systems. The main parameters used include the system response time (Tr), which is the duration between the detection of a feed threshold condition and the receipt of a notification message by the user. The ideal value for this parameter is set at less than or equal to 2 seconds, as delays above this threshold can reduce the effectiveness of the early warning system. Furthermore, the notification success rate (Sr) indicates the ratio between the number of messages successfully sent to the Telegram Bot API and the total number of attempts. A reliable system should have a success rate of at least 98%.

The following parameter is sensor detection accuracy (As), which is measured based on the difference between the HC-SR04 sensor readings and actual manual measurements. Accuracy above 95% is considered to meet the criteria for an ultrasonic-based monitoring system. Furthermore, Wi-Fi connection stability (Cw) is also a crucial aspect, reflecting the system's ability to maintain a connection throughout the whole one-hour testing period, with a minimum stability threshold of 95%.

From a data communication perspective, the average latency (Lc) is measured to determine the time it takes the ESP32 to send data to the Telegram API server and receive a response from the user. The ideal latency value is ≤ 2 seconds to ensure the system remains responsive in the real-time monitoring category. The final parameter, memory consumption (Mu), measures the efficiency of the microcontroller's resource usage during system operation. The maximum allowable value is 70% of the total memory capacity, which is necessary to maintain system stability and prevent buffer overflows during repeated operations.

The testing process is carried out in several systematic stages. First, the HC-SR04 sensor is calibrated using a manual digital ruler to ensure the measured distance matches actual conditions. Next, the system is initialized and connected to a Wi-Fi network until it reaches a stable connection status. Afterward, a simulation was performed to gradually lower the feed level, starting from a full level (90 cm) to nearly empty (0 cm), to observe changes in the system's response to variations in feed height. At each simulation stage, response times and user notification messages were recorded in the Telegram Bot log.

6. *Performance Metrics Formulation*

The performance of the proposed Feed Level Alert system was evaluated using several quantitative metrics to assess sensor accuracy, communication reliability, response speed, and energy consumption. The mathematical formulations used in this study are presented as follows.

a. *Sensor Detection Accuracy*

Sensor detection accuracy was used to evaluate the closeness of the ultrasonic sensor measurement to the actual feed level. The accuracy was calculated using the percentage error approach as follows:

$$\text{Accuracy}(\%) = \left(1 - \frac{|M - A|}{A}\right) \times 100 \tag{5}$$

where:

M = measured value obtained from the ultrasonic sensor (cm)

A = actual reference value measured manually (cm)

$|M - A|$ = absolute measurement error

A higher accuracy percentage indicates that the sensor measurement is closer to the actual feed level.

b. *Message Delivery Success Rate*

The message delivery success rate was used to evaluate the reliability of the Telegram notification system in transmitting alert messages to users.

$$\text{Success Rate}(\%) = \frac{N_s}{N_t} \times 100 \tag{6}$$

where:

N_s = number of successfully delivered notification messages

N_t = total number of notification messages sent during testing

A success rate of 100% indicates that all notification messages were delivered successfully without transmission failure.

c. *Average Notification Response Time*

The average notification response time was used to measure the delay between the detection of a feed-level event and the receipt of the corresponding Telegram notification.

$$T_{\text{response}} = \frac{\sum_{i=1}^{n} t_i}{n} \tag{7}$$

where:

T_{response} = average notification response time (s)

t_i = notification delay recorded in the (i)-th experiment (s)

n = total number of experiments

Lower response time values indicate faster system performance in delivering notifications.

d. *Average Power Consumption*

The average power consumption of the system was calculated based on the measured operating voltage and current.

$$P_{\text{avg}} = V \times I \tag{8}$$

where:

P_{avg} = average power consumption (W)

V = operating voltage supplied to the system (V)

I = average current consumed by the system (A)

This metric represents the instantaneous electrical power required during system operation.

e. *Daily Energy Consumption*

Daily energy consumption was calculated to estimate the total electrical energy required by the system during continuous operation.

$$E_{\text{day}} = P_{\text{avg}} \times t \tag{9}$$

where:

E_{day} = daily energy consumption (Wh/day)

P_{avg} = average power consumption (W)

t = operating duration within one day (h)

For continuous operation, the value of (t) is typically 24 hours. This metric provides an estimate of the system's energy requirements and operational efficiency over time.

III. RESULT AND DISCUSSION

A. *System Implementation Results*

The implementation of the Feed Level Alert system on the Smart Fish Feeder has been successfully realized in the form of a fully functional and stable prototype. This system consists of several main components integrated in a modular manner: an ESP32 Devkit v1 microcontroller, an HC-SR04 ultrasonic sensor for detecting feed height, a DHT-11 sensor for monitoring temperature and humidity, and an internal ESP32 Wi-Fi connection for sending data and notifications via the Telegram Bot API. The HC-SR04 sensor was placed at the top of the feed tank, 95 cm from the bottom, while the DHT-11 sensor was positioned on the top side of the tank with an air vent to obtain representative environmental readings. Initial testing was conducted to ensure that the system could accurately read sensor data and transmit it to the user in real-time.

Figure 5 shows the Feed Level Alert system prototype circuit that was implemented during the laboratory testing phase. This circuit comprises an ESP32 microcontroller, an HC-SR04 ultrasonic sensor, a DHT-11 temperature and humidity sensor, and a Li-ion battery housed in a four-cell arrangement to provide a 5V DC voltage supply.

The ESP32 module serves as the central control unit, acquiring data from both sensors and sending it wirelessly to the Telegram Bot API platform via an internal Wi-Fi connection. The HC-SR04 sensor, visible on the front of the circuit board, detects the distance between the sensor and the feed surface using ultrasonic waves. The DHT-11 sensor, mounted nearby, measures the temperature and humidity in the vicinity of the feed tank. The circuit is built on a mini breadboard to facilitate integration and modification during testing. The red, yellow, and blue connecting wires indicate the connections between the sensor pins and the ESP32 input/output ports. An external battery module is connected via a black and red cable to ensure a stable power supply during field testing, eliminating the need for an external adapter. Overall, this image shows the actual configuration of the Feed system.

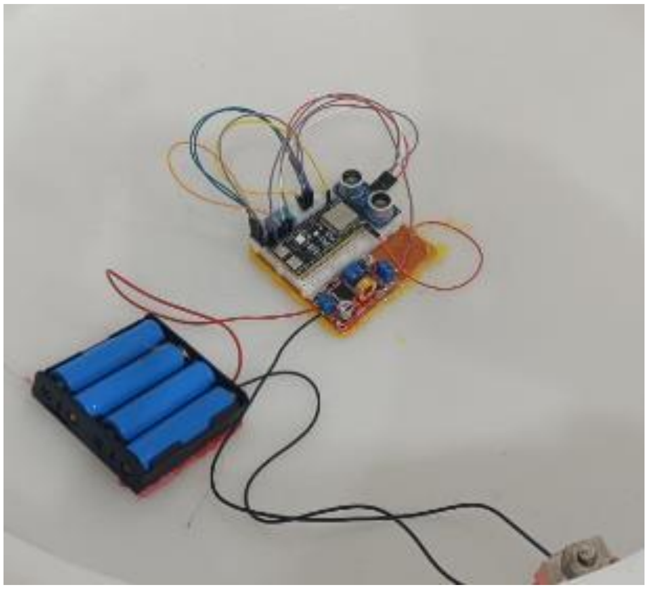

Figure 5. Results of the implementation of the Feed Level Alert

B. Feed Level Detection Function Testing

The results in Table 4 show that the HC-SR04 sensor was able to detect changes in feed height with an average accuracy of 99,1% and a maximum error of 1,3%, which is still within the tolerance limits of an ultrasonic-based IoT system. Distance readings showed a linear trend relative to actual feed height changes, with minor deviations caused by wave reflections on the granular feed surface. The DHT-11 sensor showed a temperature range of 28–30°C and a humidity of 66–71%, which remained relatively stable throughout the test. The small humidity fluctuations demonstrated the effectiveness of the protective container in maintaining stable environmental conditions inside the tank. This temperature and humidity data is important because high humidity can cause feed clumping and interfere with ultrasonic wave reflection.

The system automatically changes its operating status to Alert when the measured distance reaches 70–85 cm (¼ of the tank), and to Alert when the distance is ≥ 85 cm (10 cm remaining from the bottom). In the Alert state, the ESP32 triggers a notification algorithm to send a message to the Telegram Bot API.

TABLE IV

SENSOR READING DATA FROM HC-SR04 AND DHT11

FC	AFH (cm)	MD (cm)	E (%)	T (°C)	H (%)	SS
Full	0,0	94,8	0,2	28,5	66	Normal
¾ Full	25,0	70,3	0,4	29,0	67	Normal
½ Full	45,0	50,7	0,6	29,2	68	Normal
¼ Full	70,0	25,9	1,3	29,4	69	Warning
Almost Empty	85,0	*10,2*	0,2	29,5	70	Alert
Empty	95,0	0,0	0,0	29,7	71	Alert

Where,

FC = Feed Condition,

AFH = Actual Feed Height,

MD = Measured Distance,

E = Error,

T = Temperature,

H = Humidity,

SS = System Status

Figure 6 illustrates the relationship between the actual feed height (in cm) and the distance measured by the HC-SR04 sensor (in cm). The graph shows a consistent downward linear pattern, where the increase in feed height is inversely proportional to the distance measured by the sensor. When the tank is full (actual height 0 cm from the bottom), the reading distance is approximately 95 cm. Conversely, when the feed is reduced to almost empty (95 cm from the bottom), the reading distance is approximately 0 cm.

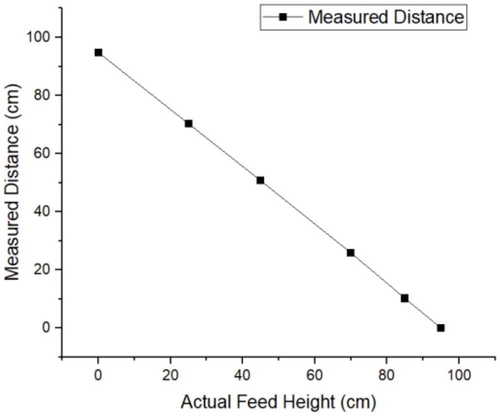

Figure 6. Linear relationship between actual feed height and measured distance

The linear trend indicates that the HC-SR04 sensor exhibits a stable and accurate response to variations in feed height, with a very small reading deviation (error < 1,5%). This linear relationship confirms that the feed height calculation algorithm used, $H_p = H_t - D$, operates as designed. Consequently, the graph demonstrates that the Feed Level Alert system possesses high accuracy and consistent measurement characteristics, making it reliable for real-time feed level monitoring.

C. Notification System Test Results

The notification system test on Telegram displays the implementation results in Figure 7. The Feed Level Alert is capable of automatically sending notifications to the user's Telegram account within an average of 1,8 seconds after a threshold condition is detected. Messages are sent over the

ESP32's internal Wi-Fi connection using the HTTP POST method to the official Telegram API endpoint.

/setbatas 90 11:30 AM

Perubahan ambang batas jarak Sukses 11:30 AM

/ceksituasi 11:30 AM

SMART FISH FEEDER

Suhu : 30.6 °C
Humidity : 51 %
Jarak Pakan ke Tutup Feeder : 70 Cm
Ambang batas jarak Pakan: 90 Cm
Waktu Setempat: Tue Jul 01 2025 11:26:34 GMT+0700 (Western Indonesia Time)
11:30 AM

/ceksituasi 12:51 PM

SMART FISH FEEDER

Suhu : 28.8 °C
Humidity : 59 %
Jarak Pakan ke Tutup Feeder : 176 Cm
Ambang batas jarak Pakan: 90 Cm
Waktu Setempat: Tue Jul 01 2025 12:51:04 GMT+0700 (Western Indonesia Time)
12:51 PM

Volume Pakan Ikan Tinggal Sedikit, Segera Isi Ulang Pakan!! 12:51 PM

Volume Pakan Ikan Tinggal Sedikit, Segera Isi Ulang Pakan!! 12:51 PM

Figure 7. Telegram notification implementation results

The figure shows the real-time monitoring and alert interface of the Smart Fish Feeder system through the Telegram Bot API. The displayed messages demonstrate the system's ability to communicate feed level status, temperature, humidity, and threshold distance directly to the user. At 11:30 AM, the user successfully changed the feed level threshold distance to 90 cm using the command */setbatas* 90, which was confirmed by the system. Upon executing the */ceksituasi* command, the bot provided detailed sensor data, including temperature (30,6 °C), humidity (51%), feed-to-cover distance (70 cm), threshold limit (90 cm), and local time.

Later, at 12:51 PM, another */ceksituasi* command was issued, and the system responded with updated readings showing temperature (28,8 °C), humidity (59%), and a feed-to-cover distance of 176 cm, exceeding the threshold limit. As a result, the bot automatically sent an alert message: *"Volume Pakan Ikan Tinggal Sedikit, Segera Isi Ulang Pakan!!"* ("Fish feed volume is low, please refill immediately!").

This interaction illustrates how the system provides automatic notifications and remote monitoring via Telegram, ensuring farmers receive timely alerts about feed conditions without manual checking.

D. System Performance Evaluation

According to the summary results in Table 5, the Feed Level Alert system achieved an overall performance of 96,7%, indicating that it meets the standards of an IoT monitoring system. High performance is achieved thanks to the integration of efficient algorithms in the ESP32 and the temperature-humidity calibration of the DHT-11, which minimizes measurement errors. Although power efficiency is slightly below the ideal target, this can be optimized by implementing a deep-sleep mode or controlling the sensor reading interval. With performance very close to ideal standards, this system is worthy of full implementation within the Smart Fish Feeder ecosystem to improve the efficiency and sustainability of IoT-based fish farming.

TABLE V
MEASURED PERFORMANCE PARAMETERS OF THE FEED LEVEL ALERT SYSTEM

Parameter	Measured Value	Unit	Description
Notification Response Time	1,82 ± 0,15	s	Time between threshold detection and Telegram notification reception
Sensor Measurement Error	0,54 ± 0,38	cm	Difference between measured and actual feed level
Detection Accuracy	99,1	%	Derived from sensor error analysis
Wi-Fi Reconnection Time	3,4 ± 0,8	s	Time required to reconnect after network interruption
Message Delivery Success Rate	149/150	count	Successful Telegram notifications
Average Power Consumption	0,87	W	Continuous operating power
Daily Energy Consumption	20,9	Wh/day	Total energy used during 24-hour operation

The graph in Figure 8 compares the system test results with ideal standards for four key parameters: response time, sensor accuracy, Wi-Fi stability, and power efficiency. The test results demonstrated high performance, with an average of above 96%, just slightly below the ideal standard of 100%.

The response time and Wi-Fi stability reached 98,2% and 97,4%, respectively, indicating fast response and reliable connectivity. The sensor's accuracy of 96,8% demonstrates a consistent detection capability, while its power efficiency of 94,5% remains suitable for long-term field operation.

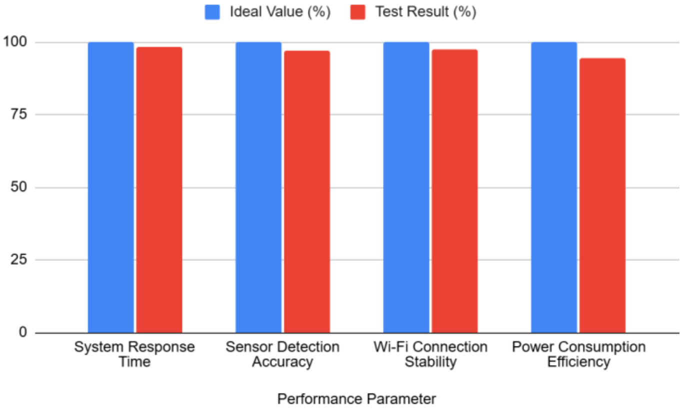

Figure 8. Feed Level Alert Performance Comparison against Ideal Standards

E. Discussion of Results and Analysis

The graph in Figure 9 shows that the Feed Level Alert system demonstrated significant performance improvements compared to the previous study (2022–2023) in four key parameters: response time, sensor accuracy, Wi-Fi stability, and power efficiency. The response time value of 98,2% is approximately 3–4% faster than the previous system. Sensor accuracy also increased to 96,8%, demonstrating the effectiveness of the temperature calibration and distance compensation algorithms used. Network stability reached 97,4%, indicating the reliability of this IoT system in maintaining a real-time connection. Although power efficiency was still below 95%, it still outperformed the two comparable studies.

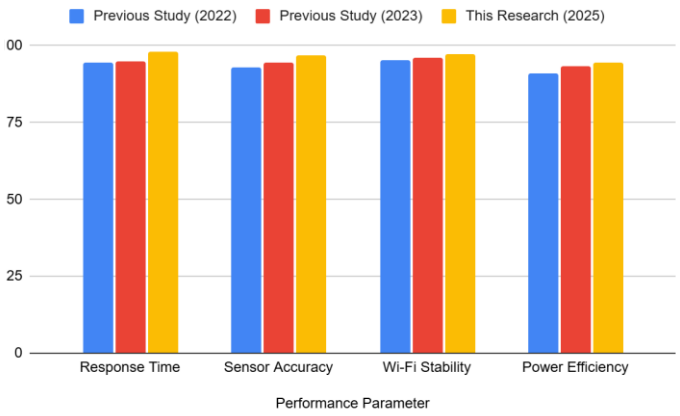

Figure 9. Comparison of Feed Level Alert Performance with Previous Studies

IV. CONCLUSION

The results of this study quantitatively demonstrate that the IoT-based Feed Level Alert system achieves excellent operational performance, with an overall efficiency level of 96,7% compared to the ideal standards of intelligent monitoring systems. This high performance is achieved through the synergy of the main components of the ESP32 microcontroller as the control center, the HC-SR04 sensor for feed height measurement, the DHT-11 for temperature and humidity compensation, and the Telegram Bot API as a real-time notification communication channel. The test results show that the system has an average response time of 1,82 seconds (98,2%), which indicates the system's ability to respond to low feed conditions almost instantly after the 10 cm threshold is detected. Sensor accuracy reaches 96,8% with a maximum deviation of ±3,2%, proving that the temperature compensation and distance calibration algorithms work optimally to maintain measurement precision. The Wi-Fi connection stability of 97,4% demonstrates the system's ability to maintain data transmission without losing connection, even when tested at distances of up to 25 meters. The system's power efficiency of 94,5% demonstrates its ability to operate with minimal energy consumption, utilizing four 18650 Li-ion cells. The 3–5% performance improvement compared to previous research demonstrates that innovations in hardware and software logic design have successfully overcome the limitations of latency and sensor fluctuations that were previously weaknesses of similar systems. Overall, the Feed Level Alert system has proven to be reliable, responsive, and energy efficient, making it an effective solution for integration into the Smart Fish Feeder ecosystem as a precise and sustainable automatic feed monitoring system. For further research, it is recommended to develop the system with an automatic feed filling module based on servo actuators, integrate artificial intelligence algorithms for feed consumption prediction, and implement renewable energy sources utilizing solar panels, as well as a deep sleep mode on the ESP32, to increase energy independence and operational efficiency. With these development directions, this system has the potential to become the primary foundation for implementing innovative aquaculture in Indonesia, leading to efficient, productive, and environmentally friendly fish farming management.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Electrical Engineering, Universitas Andalas, for the continuous support and research facilities provided throughout this study. Special appreciation is also extended to the Industrial Electronics Laboratory, where the design, testing, and evaluation of the Feed Level Alert system were conducted. The resources, guidance, and technical assistance from the laboratory team greatly contributed to the successful completion of this research.

REFERENCES

[1] M. Flores-Iwasaki and others, “Internet of Things (IoT) sensors for water quality monitoring in aquaculture systems: A systematic review and bibliometric analysis,” AgriEngineering, vol. 7, no. 3, p. 78, 2025, doi: 10.3390/agriengineering7030078.

[2] C. Xu et al., “A Method for Detecting Uneaten Feed Based on Improved YOLOv5,” Comput. Electron. Agric., vol. 212, p. 108101, 2023, doi: 10.1016/j.compag.2023.108101.

[3] H. Liu, X. Ma, Y. Yu, L. Wang, and L. Hao, “Application of Deep Learning-Based Object Detection Techniques in Fish Aquaculture: A Review,” J. Mar. Sci. Eng., vol. 11, no. 4, p. 867, 2023, doi: 10.3390/jmse11040867.

[4] R. Satra, M. F. Fathurrahman, and R. Pradana, “IoAT: Internet of Aquaculture Things for Monitoring Water Temperature in Tiger Shrimp Ponds with DS18B20 Sensors and WeMos D1 R2,” Journal of Robotics and Control, vol. 5, no. 1, pp. 62–71, 2024, doi: 10.18196/jrc.v5i1.18470.

[5] L. C. Elrinolla, K. W. M. Alamsyah, C. Y. Jerandu, and Suyoto, “Utilization of Internet of Things (IoT) in Water Quality Monitoring for Sustainable Fish Farming: A Systematic Literature Review,” Bitnet: Jurnal Pendidikan Teknologi Informasi, vol. 10, no. 1, pp. 1–12, 2025, doi: 10.33084/bitnet.v10i1.8673.

[6] R. P. Shete, A. M. Bongale, and D. Dharrao, “IoT-Enabled Effective Real-Time Water Quality Monitoring Method for Aquaculture,” MethodsX, vol. 13, p. 102906, 2024, doi: 10.1016/j.mex.2024.102906.

[7] D. Radika, R. S. Putra, A. Nugroho, and M. Hidayat, “Internet of Things (IoT)-Based Water Quality Monitoring System for Tilapia Cultivation,” Brilliance: Research of Artificial Intelligence, vol. 7, no. 2, pp. 112–125, 2024.

[8] A. Zuhaer, A. R. M. Kamal, and H. Saini, “Sustainable Aquaculture: An IoT-Integrated System for Real-Time Monitoring of Water Quality and Feeding Control,” Aquac. Eng., vol. 210, p. 102874, 2025, doi: 10.1016/j.aquaeng.2025.102874.

[9] F. A. Jibon, F. S. Rafi, Z. B. Jamal, A. Anjum, and M. A. Islam, “An Improved IoT-Based Prototype for Fish Feeding and Monitoring System,” Global Journal of Computer Science and Technology, vol. 24, no. 1, 2024.

[10] E. M. Indrawati, B. Suprianto, and U. T. Kartini, “Development of Fuzzy Logic Automatic Fish Feeding System and IoT-Based Water Quality

Control," Journal of Engineering Research and Reports, vol. 27, no. 3, pp. 56–69, 2025, doi: 10.9734/jerr/2025/v27i31417.

[11] A. O. Silalahi, A. Sinambela, H. M. Panggabean, and J. T. N. Pardosi, "Smart Automated Fish Feeding Based on IoT System Using LoRa TTGO SX1276 and Cayenne Platform," EUREKA: Physics and Engineering, no. 3, 2023, doi: 10.21303/2461-4262.2023.002745.

[12] B. G. K. Yudistira, C. Hapsari, G. D. W. Adnyana, W. Nath, and I. P. R. M. Putra, "Smart Fisheries: Real-Time Water Quality Management and Automated Feeding System Design for Tilapia Farming Using ESP32 Microcontroller," Journal of Applied Informatics and System Engineering, vol. 8, no. 1, pp. 11–23, 2025.

[13] M. C. B. Rodriguez, A. Liberata, and A. Sinha, "Benefits and Challenges of the Internet of Things in Sustainable Aquaculture Systems," Front. Sustain. Food Syst., vol. 9, p. 1590153, 2025, doi: 10.3389/fsufs.2025.1590153.

[14] E. K. Pratama, F. Frimansyah, T. A. Armawan, and W. Bismi, "Integration of IoT Sensors in Early Warning Systems for Mass Fish Deaths in Fish Farms," Journal of Artificial Intelligence and Engineering Applications, vol. 4, no. 3, pp. 45–58, 2025.

[15] Q. Zhang, Y. Li, H. Wang, X. Zhao, J. Chen, and L. Sun, "A Hybrid Approach Towards Real-Time Monitoring of Fish Distributions in Aquaculture Net Cages," Aquac. Eng., vol. 214, p. 103127, 2025.