Jurnal Elektronika dan Telekomunikasi (JET), Vol. 25, No. 2, December 2025, pp. 71-79
Accredited by KEMDIKBUDRISTEK, Decree No: **158/E/KPT/2021**
doi: 10.55981/jet.717

Implementation of Internet of Things-Based Autofeeder to Maintain Koi Pond Water Quality

Helmy [a,*], Suko Tyas Pernanda [a], Septiantar Tebe Nursaputro [a], Mona Inayah Pratiwi [b], Brian Rahmaditya [a], and Clara Silvia Anggreini [a]

[a] *Electrical Engineering, School of Electrical Engineering*
Politeknik Negeri Semarang
Semarang, Indonesia
[b] *Business Administration*
Politeknik Negeri Semarang
Semarang, Indonesia

Abstract

Koi fish farming requires careful monitoring of water temperature and pH to prevent adverse impacts on the fish. This study presents a prototype IoT-based autofeeder that integrates real-time water quality monitoring and automatic feeding, controllable via both an Android application and local device buttons. The system allows users to configure feeding schedules, feed throw levels, and durations, as well as set pH thresholds. When the pH exceeds the safe range, the system automatically stops feeding and sends notifications, enabling the user to inspect and maintain pond water quality. The findings demonstrate that the dispensing level significantly influences the feed-throwing distance; higher dispensing levels result in longer distances. Small-sized feed (S) consistently produced the highest output, followed by medium-sized (M) and large-sized (L). Increasing the feeding duration enhanced the weight of the released feed. Additionally, the average delay in sensor data transmission to the database was recorded at 5.48 seconds. The data loss rate during the testing period was 1.72%, which is considered acceptable and does not adversely affect system operations. The data transmission system demonstrated good and stable performance with relatively low data loss.

Keywords: koi fish, autofeeder, temperature sensor, pH sensor.

I. INTRODUCTION

Koi fish are among the most sought-after ornamental fish species and are widely kept by the public. Their beauty in terms of shape and color makes them a popular choice for pond decoration [1]. In addition to their aesthetic value, koi fish also possess high economic value, with promising sales prospects that attract many people both as a hobby and as an investment [2].

Success in koi farming is greatly influenced by pond conditions, particularly water quality factors such as pH and temperature [3]. The ideal pH range for koi fish is between 6.5 and 8.0; excessive fluctuations can cause stress or even death [4], [5]. Furthermore, the optimal water temperature for koi growth is between 25°C and 27°C. Inappropriate temperatures can disrupt metabolism and harm fish health [6]. Therefore, maintaining stable pH and water temperature is crucial for ensuring the optimal health and development of koi fish. However, water quality management often faces various challenges, such as irregular monitoring, which can affect environmental stability and fish health [7]. Additionally, the frequency and amount of feeding should also be considered. Ideally, koi should be fed frequently but in small quantities to maintain nutritional balance and water quality. Overfeeding can increase ammonia levels, which are detrimental to fish health [8].

Previous studies have developed various solutions to address these challenges. Some studies focused on feeding systems, such as NodeMCU- and Blynk-based automatic feeders that provide scheduled and automated feeding [9]; however, these systems lack pH monitoring and feed quantity control. Fuzzy logic-based feeding systems have been introduced to optimize feeding based on temperature and turbidity [10], and energy-efficient automatic feeders have also been proposed [11], yet these systems rely on physical buttons and LCD modules, limiting their remote functionality.

Other studies have concentrated on water quality monitoring systems, including web-based platforms capable of measuring temperature, oxygen, and pH [12], but these only provide SMS notifications without Android-based automation. Smart aquarium systems that monitor pH and control feeding via smartphones [13] or IoT-based remote management applications [14] also exist, but most lack pH integration or fail to respond automatically to poor water conditions. Another study introduced an aquarium management system that monitors water quality, detects fish diseases, and prevents algae growth [15], but it lacks detailed field trials and does not describe strategies for handling poor water conditions.

Several integrated systems have attempted to combine feeding and monitoring. For instance, YOLO-based aquaponics systems can detect uneaten feed [16], and microcontroller-based automatic feeding systems

* Corresponding Author.
Email: helmy@polines.ac.id
Received: December 12, 2024 ; Revised : August 15, 2025
Accepted: August 22, 2025 ; Published: December 31, 2025

with water replacement have been developed [17]. However, these rely on limited parameters, such as turbidity and pH, without advanced control mechanisms. Smart koi farming prototypes can automatically feed and monitor water temperature [18], while IoT-based feeders using the System Usability Scale (SUS) [19] and koi-specific smart feeders [20] mostly focus on feeding or minimal monitoring, lacking comprehensive water quality integration.

Most existing studies focus either on automatic fish feeding or water quality monitoring, with only a few attempting to combine these features. However, the combined systems that do exist remain limited: they either monitor only basic parameters (such as temperature or turbidity) or lack real-time remote control capabilities. Previous studies have not fully integrated an IoT-based autofeeder with both pH and temperature monitoring through an Android application. To address this gap, this research proposes a novel IoT-based autofeeder system that:

1. Performs automatic fish feeding with configurable pH threshold, feed level, throw duration, and feeding schedule alarm;
2. Monitors water pH and temperature in real-time;
3. Integrates with an Android application for remote monitoring and control, including a configurable pH threshold that automatically pauses feeding and sends notifications when water conditions become unsafe, helping to prevent overfeeding and maintain pond water quality.

This integration enables koi pond managers to maintain water quality more effectively, receive early notifications to prevent overfeeding, and support sustainable koi farming practices.

II. Materials and Methods

A. Autofeeder

An autofeeder is an automatic feeding device that allows users to set the frequency, schedule, and dosage of feeding. It provides convenience for pet owners by regulating the timing and portioning of food, thus ensuring pets receive their required intake throughout the day [21]. The use of autofeeders can improve efficiency by providing a consistent feeding schedule, reducing labor costs, and minimizing feed wastage compared to manual feeding methods. This can improve overall aquaculture productivity [22].

B. IoT

IoT (Internet of Things) is a system that connects various electronic devices and sensors through the internet, allowing them to communicate with each other and share data [23]. IoT is defined as a network of sensors, actuators, and real-world objects controlled by a central module such as a microcontroller or microprocessor. Sensors transmit data, which is then processed by the central module. Based on the program, the device activates the actuators to perform the required actions [14].

The working principle of IoT is that every object must have an Internet Protocol (IP) address, which serves as an identity in the network. With the IP address, the object can communicate with other devices on the same network. Once connected to the internet, the IP address also allows the device to be set up and controlled remotely [24].

C. pH Sensor

A pH sensor is a device designed to measure the acidity or basicity of a solution [25]. This sensor operates on the basic principle that acidic and alkaline solutions produce different voltages. The DFRobot pH sensor has a pH measurement range from 0 to 14 with an accuracy of 0.1 pH and can operate in temperatures between 0 and 60 ºC, making it suitable for various environments [26].

Physically, the sensor is equipped with an LED as a power indicator and a BNC connector, making it easy to connect with microcontrollers such as Arduino. By connecting this sensor through an analog cable, users can monitor changes in pH values in real time [27]. Under neutral conditions, this sensor reads zero voltage, while in acidic and alkaline solutions, the sensor detects positive and negative voltages, providing a fast and accurate response to pH changes [28].

D. Temperature Sensor

The DS18B20 is a digital temperature sensor from the Maxim IC series capable of measuring temperatures with 9 to 12-bit accuracy in the range of -50 °C to 125 °C with an accuracy of +/- 0.1 °C. Unlike temperature sensors in general that require ADC and several port pins on the microcontroller, the DS18B20 does not require ADC and only requires one cable for data communication [29]. This sensor uses a One-Wire interface, which eases installation and makes it more efficient. One of the advantages of the DS18B20 is its ability to be connected in parallel to a single pin of a microcontroller, such as an Arduino, so that multiple sensors can be connected on a single data line without the need for additional complex wiring. The sensor is also coated with an aluminium tube, making it ideal for temperature measurement in wet or liquid environments. This coating protects the sensor from potential short circuit risks that could damage the device or harm living things, so the DS18B20 is often used in safe and reliable water temperature control applications [30].

E. ESP32

The ESP32 is a Wi-Fi and Bluetooth-based microcontroller developed by Espressif Systems. It offers high performance, flexibility, and power efficiency, making it ideal for IoT projects [31]. As an advanced version of the ESP8266, the ESP32 has improvements in terms of processor speed, connectivity, and other additional features. In addition, this microcontroller can connect hardware and software via Wi-Fi and Bluetooth [32].

The ESP32 has advantages over other microcontrollers such as the Arduino Uno and ESP8266. It has a larger number of pins, including analog pins, which allows it to handle various applications more effectively. In addition, the ESP32 is equipped with a larger memory capacity, which supports more efficient table data storage and processing. Support for Bluetooth 4.0 Low Energy and Wi-Fi also provides ease of Internet deployment, making it a very attractive option for a variety of IoT applications that require optimal connectivity and performance [33].

F. RobotDyn R3 UNO+WI-FI

The RobotDyn R3 UNO+WI-FI, which is a classic Arduino Uno R3 board that has two processors: the Atmel ATmega328 microcontroller and the ESP8266 Wi-Fi chip. ESP8266 is an additional module that functions to connect Arduino with Wi-Fi internet networks and create TCP/IP connections (Wahyuni et al., 2021). The use of Arduino UNO+Wi-Fi can replace the Arduino Uno Rev3 and the ESP8266 module with a single board with equivalent functions. This board has 14 digital I/O pins and six analog I/O pins. In addition, this board also uses 802.11 b/g/n 2.4 GHz Wi-Fi for wireless connectivity and a micro USB port for wired communication and serial monitoring [34].

G. Autofeeder Design

The design of the autofeeder with koi pond water quality monitoring is shown in Figure 1. The device is equipped with various components to support its function of automatically managing water quality and fish feeding.

One of the main features is the sampling reservoir located on one side of the device. This reservoir serves to collect water from the koi pond. Inside the reservoir are pH sensors and temperature sensors that are used to monitor the water quality in real-time. The pH sensor measures the acidity of the water, while the temperature sensor measures the temperature of the pond water. Water from the pond will enter the reservoir through the inlet, then after being analyzed by the sensors, the water will exit back to the pond through the outlet, while excess water will be drained through the drain to prevent water spillage. At the bottom of the autofeeder, there is a feed output channel designed to automatically dispense fish feed according to predetermined settings. With this design, the autofeeder not only regulates feeding but also maintains the water quality of the koi pond.

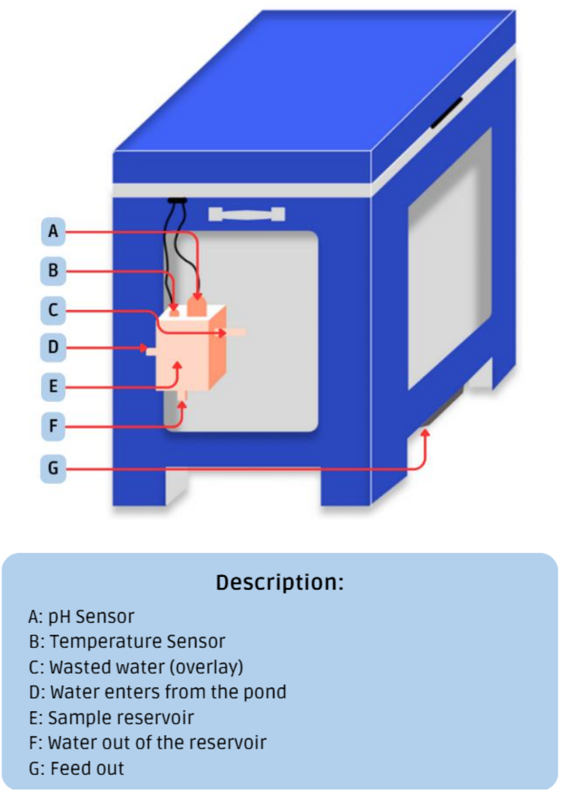

Figure 1. Design autofeeder.

H. System Design

The block diagram in Figure 2 illustrates the components and data flow of the autofeeder system. This system uses two main microcontrollers, namely the Arduino Uno Wi-Fi and ESP32, each with a distinct role in monitoring and controlling the autofeeder. For monitoring, the Arduino Uno Wi-Fi microcontroller is used. Its inputs include a temperature sensor and a pH sensor. The temperature sensor is digitally connected to the Arduino Uno Wi-Fi and is used to measure the pond water temperature, while the pH sensor is connected to the Arduino Uno Wi-Fi via an analog channel, requiring an analog isolator to separate the signals. The output of this monitoring system consists of water quality data sent periodically to a cloud server via the Wi-Fi module. This data can be accessed by the pond manager through an Android application, enabling remote water quality monitoring.

For the control section, the ESP32 is responsible for automatically managing fish feeding. This control component includes several settings, such as feed duration, feed level or distance, and feed frequency, which can be configured online through the Android application or offline directly on the autofeeder device. When configured online, the manager can access and control feeding parameters remotely via the application connected to the Wi-Fi network. However, if the system is in offline mode, settings can be adjusted directly on the device using buttons installed on the autofeeder. The result or information about the manual setting adjustments is displayed on an LCD screen connected to the ESP32, allowing the user to view the feeding status directly on the device. Inputs to the ESP32 in the control section come from the manual buttons and the time data from the RTC (Real-Time Clock) module, which schedules precise feeding times. Outputs from the ESP32 include a servo motor that drives the feeding mechanism, an LCD screen as an information display, and a motor driver that controls the feeding throw strength.

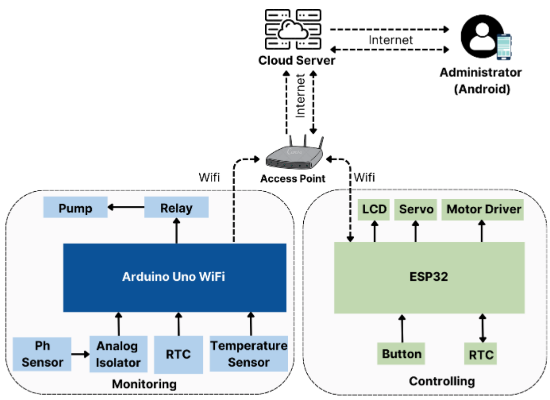

Figure 2. Block diagram of monitoring and control system.

The overall system architecture is illustrated in Figure 3, which shows the structure of the monitoring and control system for the autofeeder designed to maintain the water quality in koi ponds. This system is built with two primary functions: monitoring water quality parameters and automatically controlling fish feeding. Pond water temperature and pH data are monitored in real-time through sensors located in the autofeeder tank. To support data transmission, the autofeeder first connects to an access point (AP) via Wi-Fi, enabling the device to access the internet. The information from these sensors is sent to a cloud server using the REST API method, making the data accessible to the manager through an Android application.

On the control side, the system offers various configuration options, such as feeding time alarms, feeding duration, feed level or throw distance, and pH threshold settings. If water quality parameters fall outside optimal ranges, the manager will receive a notification through the application, allowing for timely preventive or corrective actions. All data communication between the autofeeder, cloud, and management application is conducted through the internet and cloud access, enabling remote monitoring and control.

The autofeeder system flowchart is shown in Figure 4. This system allows users to choose between two modes, online or offline, according to their needs. In offline mode, users can directly monitor and control the system through the device. There is a feature to set the desired alarms. Alarms can be configured for specific days or set to repeat daily, with a maximum of 12 alarms per day. Users can also set the feed duration, adjustable from 1 second to 30 seconds. Additionally, users can adjust the strength of the feed throw. In this setting, users can control the feed throw level, determining whether the feed will be dispensed far from or close to the autofeeder.

In online mode, users can configure all autofeeder settings through the available mobile application by logging in using the provided QR code. Through the application, users can monitor water conditions, set alarms, adjust feeding duration, control the feed throw level, and establish pH threshold limits. If the measured pH exceeds or falls below the set threshold, the system will automatically stop feeding to prevent water quality degradation and notify the user via the Android application.

The autofeeder also samples water condition data every 10 minutes to determine the current water condition. In every 10-minute interval, a small pump fills the pond water sampling tank. This sampling tank holds water, allowing the electronic sensors to read its condition. Parameters monitored include the pH level and water temperature—whether the water is too acidic or alkaline, and whether the temperature is too hot or cold. The data is then transmitted to the server.

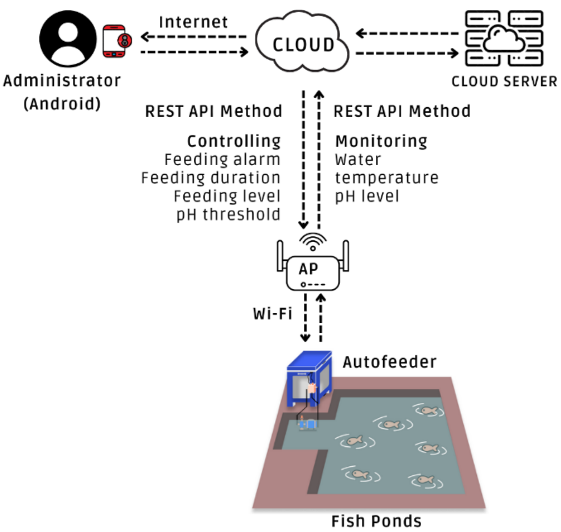

Figure 3. System architecture of monitoring and control system.

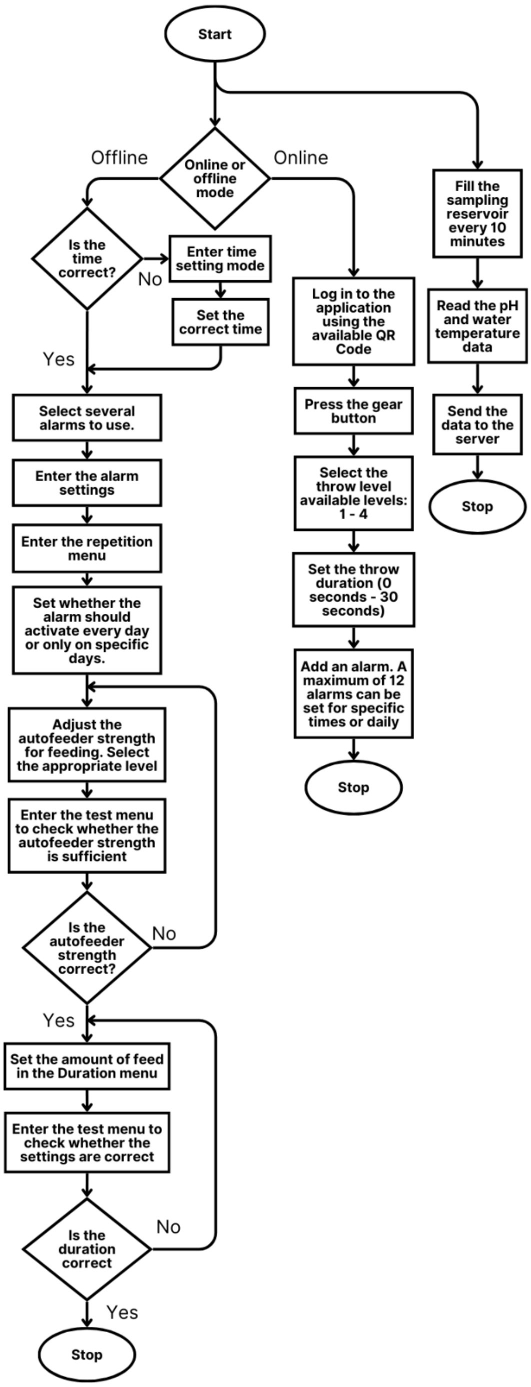

Figure 4. Autofeeder system flowchart.

I. Android App

The Autofeeder Android Application is displayed in Figure 5. It is designed to help pond managers monitor water quality and control feeding in real-time. This application ensures that water quality parameters, such as pH and temperature, remain within optimal conditions and allows the manager to efficiently and accurately control the feeding process via the autofeeder device.

1) Initial Screen of the Application

On the initial screen of the application, users are provided with a scanning feature that allows the manager to add a new autofeeder device by scanning the QR code or barcode on the device. This feature simplifies the process of inputting the device quickly and conveniently. Additionally, on this screen, users can select the autofeeder device to monitor and control, enabling the manager to monitor multiple devices simultaneously.

2) Application Dashboard

After selecting the autofeeder device, the manager is directed to the dashboard, which displays information about the water quality in the pond, such as the current pH value and water temperature. This dashboard provides real-time information about the pond's environmental conditions. Furthermore, this section includes an export data feature, which allows the manager to download data related to pH and temperature in a file format for further analysis or documentation purposes.

3) Settings Feature

The settings section of the application provides four main features that can be adjusted by the manager:

a) *pH Threshold*: The manager can set the desired pH range for the pond water. The system will provide an alert if the pH value exceeds the set threshold.

b) *Feed Throw Level*: This feature allows the manager to set the dispensing power level of the autofeeder device, which determines the force applied when dispensing feed. The system offers four power levels: Level 1 represents the lowest dispensing power, producing the shortest throw distance, while Level 4 is the highest power, delivering the farthest throw. This setting enables the feed to be distributed optimally within the pond area according to its size and stocking density.

c) *Feed Duration*: The manager can adjust the feeding duration for each session, ensuring that the amount of feed dispensed meets the pond's requirements, from 0 to 30 seconds.

d) *Alarm*: The alarm feature is used to set the time and recurrence days for feeding. It can be programmed for up to 12 alarms per day. The system will provide an alert if the pH value exceeds the set threshold.

Figure 5. Autofeeder Android Application.

III. Result and Discussion

A. Dispensing level and feed size on feed throwing distance

Tests on the distance of dispensing feed were conducted using three different feed sizes, namely Small (S), Medium (M), and Large (L), at four different levels (Level 1, Level 2, Level 3, and Level 4). Measurements were taken to see the effect of feed size on feed-throwing distance at each level. The feed used for testing consisted of three pellet sizes from the brand "Hokky Fish Food" with approximate diameters of 3 mm (S), 5 mm (M), and 7 mm (L) (Figure 6). Data on the results of feed throwing distance are shown in Table 1.

From Table 1, it is evident that the feed-throwing distance increases as the level increases. Higher levels apply greater dispensing power or use a more optimized dispensing angle, both of which contribute to longer feed-throwing distances. This indicates that the level setting significantly influences the effectiveness of feed distribution.

In addition, feed size also affects the feed-throwing distance, with larger pellets consistently producing longer distances than smaller ones at each level. This is because larger pellets have greater mass, resulting in a higher momentum-to-drag ratio, which allows them to maintain velocity longer and overcome air resistance more effectively. Overall, the results of this feed-dispensing test demonstrate a clear relationship between feed size, dispensing level, and the resulting distance.

Figure 6. Feed pellet sizes used in the experiment: (A) Small, (B) Medium, (C) Large.

Table 1
Feed Throw Distance

Feed Throw Level	Throw Distance (meters)		
	Small-sized feed	Medium-sized feed	Large-sized feed
Level 1	1.25	1.5	1.6
Level 2	1.55	1.65	1.8
Level 3	1.93	2.15	2.37
Level 4	2.3	2.6	2.86

B. Dispensing level, feed size, and feed-throwing distance duration on feed output weight in grams

This test was conducted to measure the effect of dispensing level, feed size, and duration of feed-throwing on the weight of feed released by the autofeeder. The feed used in this test was the same as in the feed-throwing distance experiment, namely pellets from the brand “Hokky Fish Food” in three different sizes: small (S) with an approximate diameter of 3 mm, medium (M) with an approximate diameter of 5 mm, and large (L) with an approximate diameter of 7 mm, as shown in Figure 6. The test was conducted by measuring the feed output from the autofeeder set at four different levels (Level 1 to Level 4) with the duration of the feeding time ranging from second 1 to second 30. Each level has a different dispensing power setting to adjust the feed distribution needs in the pond area. The purpose of this test is to determine the amount of feed released in grams at each combination of level and feed size so that users can adjust the time and level of dispensing as needed.

To measure feeding accuracy, the feed output at each combination of feed throw level and duration was collected and weighed using a digital scale with 0.1 g precision. Each measurement was repeated three times, and the average value was used as the final feeding accuracy. This procedure ensures that the reported feed weight is representative and minimizes random measurement errors.

1) *Level 1 testing*

The graph in Figure 7 shows that at Level 1, the feed output weight increases gradually with time (seconds 1 to 30). The small feed produced the highest output weight compared to the medium and large. At 30 seconds, the small feed reached about 50 g, while the medium and large were below it, respectively. This shows that at Level 1, small feed tends to produce a larger output weight within the same duration.

2) *Level 2 testing*

In Figure 8, the Level 2 graph shows a similar pattern to Level 1, but with a higher increase in output weight. At 30 seconds, the small-sized feed reaches a weight of about 80 g, while the medium and large sizes are below it by a significant margin.

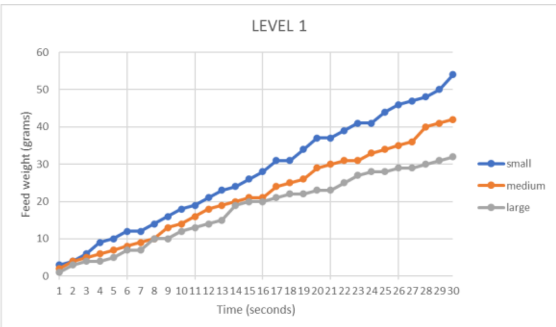

Figure 7. Level 1.

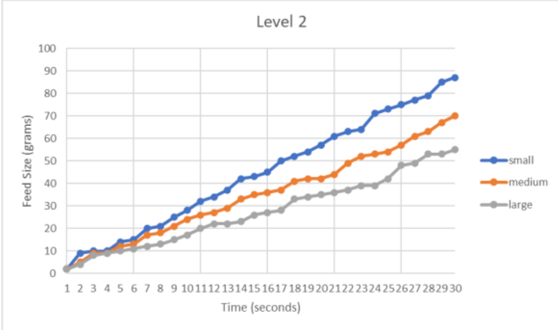

Figure 8. Level 2.

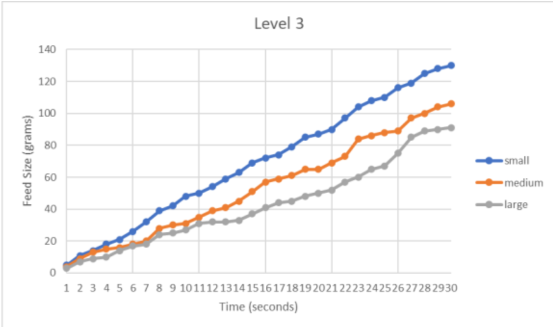

Figure 9. Level 3.

3) *Level 3 testing*

The Level 3 graph in Figure 9 shows a more significant increase in feed output weight compared to Level 1 and Level 2. At 30 seconds, the small feed reaches a weight of about 130 g, with the medium and large sizes falling below. At this level, the increase in feed output weight occurs faster, indicating that the dispensing power setting at Level 3 is higher and allows for greater output weight.

4) *Level 4 testing*

Figure 10 shows that the Level 4 graph produces the highest feed output weight among all levels. At 30 seconds, the small feed reached approximately 180 g, while the medium and large sizes are below it. The same pattern is seen, where the small feed consistently produces the highest feed output weight.

Overall, the graphs indicate that the feed output weight increases as both the dispensing level and the feed-throwing duration increase. At the same duration, a higher dispensing level produces a greater feed output

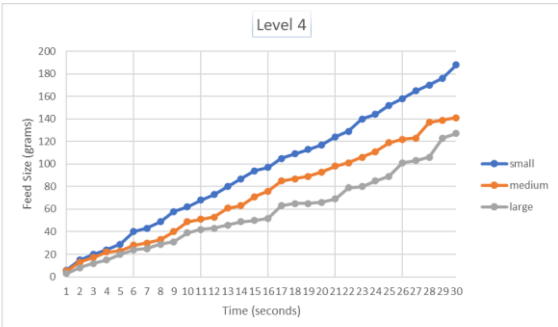

Figure 10. Level 4.

TABLE 2
DELAY TESTING

Date	Delay (s)
11/30/2024	4.94
12/01/2024	4.87
12/02/2024	5.55
12/03/2024	5.82
12/04/2024	5.68
12/05/2024	6.05
12/06/2024	5.45
12/07/2024	5.46
Average Delay (s)	5.48

weight. This occurs because a higher level applies greater force to the feeding mechanism, which increases the feed discharge rate.

In addition, at the same duration, small feed consistently produces the highest output weight, followed by medium and large feed sizes. This phenomenon occurs because smaller feed pellets can more densely cover the feeder outlet, allowing more pellets to be discharged per second. Consequently, the total feed weight released is higher compared to medium or large feed for the same duration. The "feed weight" shown in the graphs represents the cumulative total feed dispensed from the start of the feeding session up to the indicated time. These results show that dispensing level, feed size, and feed-throwing distance duration significantly affect feed output weight.

C. Delay Testing

Based on Table 2, the testing of delay for sensor data transmission to the database was conducted from November 30 to December 7, 2024, with data being transmitted every 10 minutes. The data in the table shows the daily average delay, with an average recorded during this period of 5.48 seconds. The shortest delay occurred on December 1, 2024, with an average of 4.87 seconds, while the longest delay was recorded on December 5, 2024, at 6.05 seconds. This delay is primarily caused by a slight discrepancy between the Real-Time Clock (RTC) and actual system time, which contributes approximately 4 seconds to the total delay. Despite this, a 5–6 second delay is acceptable for IoT-based aquaculture, as feeding and water-quality variations do not require instantaneous response. Therefore, the system's latency is considered reliable for practical operation.

D. Data Loss

Table 3 shows the data loss during the sensor data transmission process to the database conducted from November 30 to December 7, 2024. This table includes the daily average data transmitted (Data Transmitted) and the daily average data successfully received (Data Received), along with the difference between the two. These data provide an insight into the system's efficiency in managing data transmission during the testing period.

During this period, data was transmitted every 10 minutes, which means that every 10 minutes produced 1 piece of data. Therefore, from November 30 at 07:53 to December 7, 2024, at 23:59, the total number of data generated was 1105.

TABLE 3
DATA LOSS

Date	Data Transmitted	Data Received	The Difference between Transmitted and Received Data
11/30/2024	97	97	0
12/01/2024	144	133	11
12/02/2024	144	142	2
12/03/2024	144	142	2
12/04/2024	144	143	1
12/05/2024	144	144	0
12/06/2024	144	143	1
12/07/2024	144	142	2
Total	1105	1086	19
Data Loss (%)	**1.72**		

Based on the table, the total data sent during this period was 1105 data, while the successfully received data was 1086 data, resulting in a total data loss of 19 data. This corresponds to a data loss rate of 1.72%. The highest data loss occurred on December 1, 2024, with an 11-data difference between sent and received data, due to a power outage that temporarily deactivated the system, preventing the tool from sending data during that time. On November 30, 2024, and December 5, 2024, there was no data loss (data loss = 0%).

According to the ETSI (European Telecommunications Standards Institute) guidelines, this system is classified as highly reliable because its data loss rate of 1.72% remains within the acceptable range of 0–2%. This indicates that the system's data transmission is stable and dependable for aquaculture monitoring and feeding operations.

IV. CONCLUSION

This study demonstrates that the developed autofeeder is capable of automatically feeding fish while simultaneously maintaining koi pond water quality through real-time monitoring of pH and water temperature. The system automatically stops feeding when the pH exceeds the safe threshold to prevent overfeeding and sends notifications to the user for immediate pond inspection.

In addition, the results show that the dispensing level significantly influences the feed-throwing distance; higher dispensing levels result in longer distances. Small-sized feed (S) consistently produced the highest output, followed by medium-sized (M) and large-sized (L). Increasing duration can enhance the weight of released feed. Additionally, the average sensor data transmission delay was measured at 5.48 seconds, which is acceptable for IoT-based aquaculture applications. The system recorded a data loss rate of 1.72%, which falls within the 0–2% range recommended by ETSI, indicating that the data transmission is highly reliable and stable for feeding and water-quality monitoring operations.

DECLARATIONS

Conflict of Interest

The authors have declared that no competing interests exist.

CRediT Authorship Contribution

Helmy: Methodology, Visualization, Investigation, Writing – Original Draft Preparation, Writing – Reviewing and Editing; Suko Tyas Pernanda: Software Development, Writing – Reviewing and Editing; Septiantar Tebe Nursaputroa: Hardware Development, Writing – Reviewing and Editing; Mona Inayah Pratiwi: Funding Acquisition, Writing – Reviewing and Editing; Brian Rahmaditya: Hardware Development , Writing – Reviewing and Editing; Clara Silvia Anggreini: Software Development, Writing – Reviewing and Editing.

Funding

The research reported in this publication was supported by DIPA POLINES 2024.

REFERENCES

[1] A. K. Nurdina, A. P. Sasmito, and N. Vendyansyah, "Implementation of an Internet of Things (IoT)–based monitoring and control system for koi fry care using a web platform," (in Indonesian), *Jurnal Mahasiswa Teknik Informatika,* vol. 2, no. 2, pp. 1115-1122, 2022, doi: https://doi.org/10.36040/jati.v6i2.5349.

[2] A. Hendriana, F. Ridwansyah, A. Iskandar, A. S. Munawar, and D. Lugina, "Koi fish (Cyprinus carpio) breeding methods for producing high-quality fry at Mizumi Koi Farm, Sukabumi Regency, West Java," (in Indonesian), *Jurnal Penelitian Pertanian Terapan*, vol. 2, pp. 17–26, 2021, doi: 10.25181/peranan.v2i1.2203.

[3] B. D. T. Bodaragama, E. H. A. D. M. Miyurangana, Y. T. W. S. L. Jayakod, D. M. H. D. Vipulasiri, U. U. S. Rajapaksha, and J. Krishara, "IoT-based solution for fish disease detection and controlling a fish tank through a mobile application," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Apr. 2024, pp. 1–6. doi: 10.1109/I2CT61223.2024.10543840.

[4] S. P. Sari, J. M. Amelia, and G. I. Setiabudi, "Effect of different temperatures on the growth rate and survival of koi fish (Cyprinus carpio) fry," (in Indonesian), *Jurnal Perikanan Unram*, vol. 12, no. 3, pp. 346–354, Sep. 2022, doi: 10.29303/jp.v12i3.328.

[5] A. K. P. M. Daud, N. A. Sulaiman, Y. W. M. Yusof, and M. Kassim, "An IoT-based smart aquarium monitoring system," in *Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Apr. 2020, pp. 277–282. doi: 10.1109/ISCAIE47305.2020.9108823.

[6] S. Indriyanto, F. Titan Syifa, and H. Aditya Permana, "Water temperature monitoring system in koi fish fry ponds based on the Internet of Things," (in Indonesian) *TELKA*, vol. 6, no. 1, pp. 10–19, 2020.

[7] P. M. Jacob, J. Moni, A. Joy, G. Gopan, A. Abraham, and A. O, "An internet of things-based mobile application for remote monitoring and management of aquariums," in *2024 1st International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, Apr. 2024, pp. 1–6. doi: 10.1109/ICTEST60614.2024.10576150.

[8] N. W. Sari, I. Lukistyowati, and N. Aryani, "Effect of administering temulawak (Curcuma xanthorrhiza Roxb.) on the survival of common carp (Cyprinus carpio L.) after infection with Aeromonas hydrophila," (in Indonesian), *Jurnal Perikanan dan Kelautan*, vol. 17, no. 2, 2012.

[9] K. Jadhav, G. Vaidya, A. Mali, V. Bankar, M. Mhetre, and J. Gaikwad, "IoT-based automated fish feeder," in *2020 International Conference on Industry 4.0 Technology (I4Tech)*, Feb. 2020, pp. 90–93. doi: 10.1109/I4Tech48345.2020.9102682.

[10] N. H. Harani, A. S. Sadiah, and A. Nurbasari, "Smart fish feeder using Arduino Uno with fuzzy logic controller," in *2019 5th International Conference on Computing Engineering and Design (ICCED)*, Apr. 2019, pp. 1–6. doi: 10.1109/ICCED46541.2019.9161114.

[11] N. Busaeri, N. Hiron, A. Andang, and I. Taufiqurrahman, "Design and prototyping the automatic fish feeder machine for low energy," in *2019 International Conference on Sustainable Engineering and Creative Computing (ICSECC)*, Aug. 2019, pp. 9–13. doi: 10.1109/ICSECC.2019.8907077.

[12] R. D. Clemente, "Development of online hatchery monitoring and feeding management system for Nile tilapia," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Nov. 2019, pp. 1–4. doi: 10.1109/HNICEM48295.2019.9072727.

[13] A. K. Pasha Mohd Daud, N. A. Sulaiman, Y. W. Mohamad Yusof, and M. Kassim, "An IoT-based smart aquarium monitoring system," in *Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Apr. 2020, pp. 277–282. doi: 10.1109/ISCAIE47305.2020.9108823.

[14] P. M. Jacob, J. Moni, A. Joy, G. Gopan, A. Abraham, and A. O, "An internet of things-based mobile application for remote monitoring and management of aquariums," in *International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, Apr. 2024, pp. 1–6. doi: 10.1109/ICTEST60614.2024.10576150.

[15] B. D. T. Bodaragama, E. H. A. D. M. Miyurangana, Y. T. W. S. L. Jayakod, D. M. H. D. Vipulasiri, U. U. S. Rajapaksha, and J. Krishara, "IoT-based solution for fish disease detection and controlling a fish tank through a mobile application," in *International Conference for Convergence in Technology (I2CT)*, Apr. 2024, pp. 1–6. doi: 10.1109/I2CT61223.2024.10543840.

[16] J. John and M. P. R, "Automated fish feed detection in IoT based aquaponics system," in *2021 8th International Conference on Smart Computing and Communications (ICSCC)*, Jul. 2021, pp. 286–290. doi: 10.1109/ICSCC51209.2021.9528186.

[17] A. Akhriana, I. Intan, N. Tamsir, N. Nirwana, R. W. Rahmi, and R. Rahmadani, "Microcontroller application in feeding fish using an Android mobile," in International Conference on Cybernetics and Intelligent Systems *(ICORIS)*, Oct. 2021, pp. 1–6. doi: 10.1109/ICORIS52787.2021.9649453.

[18] P. Ginanjar, S. Opipah, D. Rusmana, Muhlas, M. R. Effendi, and E. A. Z. Hamidi, "Prototype smart fish farm in koi fish farming," in *2021 7th International Conference on Wireless and Telematics (ICWT)*, Aug. 2021, pp. 1–6. doi: 10.1109/ICWT52862.2021.9678208.

[19] U. M. R. Herdian, H. H. Nuha, and R. G. Utomo, "Smart fish feeder design and analysis at Sekemala Integrated Farming (SEINFarm)," in *2022 2nd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, Dec. 2022, pp. 53–58. doi: 10.1109/ICICyTA57421.2022.10038102.

[20] R. G. Baldovino, F. N. Magallanes, E. J. C. Santos, K. D. Conti, and P. D. L. Sia, "Smart IoT-based feeder system for koi fish (Cyprinus rubrofuscus) aquaculture," in *2024 9th International Conference on Mechatronics Engineering (ICOM)*, Aug. 2024, pp. 240–244. doi: 10.1109/ICOM61675.2024.10652414.

[21] M. Ibrahim, H. Zakaria, and E. W. Xian, "Pet food autofeeder by using Arduino," *IOP Conf Ser Mater Sci Eng*, vol. 670, no. 1, p. 012069, Nov. 2019, doi: 10.1088/1757-899X/670/1/012069.

[22] S. J. Yeoh, F. S. Taip, J. Bin Endan, R. A. Talib, and M. K. S. Mazlina, "Development of automatic feeding machine for aquaculture industry," *Pertanika J Sci Technol*, vol. 18, pp. 105–110, 2010, [Online]. Available: https://api.semanticscholar.org/CorpusID:54074167

[23] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *J Big Data*, vol. 6, no. 1, p. 111, Dec. 2019, doi: 10.1186/s40537-019-0268-2.

[24] W. Wilianto and A. Kurniawan, "History, working principles, and benefits of the Internet of Things," (in Indonesian),

Matrix : Jurnal Manajemen Teknologi dan Informatika, vol. 8, no. 2, p. 36, Jul. 2018, doi: 10.31940/matrix.v8i2.818.

[25] D. Desmira, D. Ariwibowo, and R. Pratama, "Application of pH sensors in the electrolyzer area at PT Sulfindo Adiusaha," (in Indonesian), *Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, vol. 5, no. 1, Mar. 2018.

[26] G. Marques, D. Aleixo, and R. Pitarma, "Enhanced hydroponic agriculture environmental monitoring: an internet of things approach," 2019, pp. 658–669. doi: 10.1007/978-3-030-22744-9_51.

[27] D. R. Wati and W. Sholihah, "pH and nutrient controller for lettuce plants in an NFT hydroponic system based on Arduino," (in Indonesian), *MULTINETICS*, vol. 7, no. 1, pp. 12–20, Mar. 2021, doi: 10.32722/multinetics.v7i1.3504.

[28] K. O. Flores, I. M. Butaslac, J. E. M. Gonzales, S. M. G. Dumlao, and R. S. J. Reyes, "Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server," in *2016 IEEE Region 10 Conference (TENCON)*, Nov. 2016, pp. 3018–3021. doi: 10.1109/TENCON.2016.7848600.

[29] A. F. Y. Saputro and D. A. Prasetyo, "Rancang bangun thermopen sebagai pengukur suhu menggunakan sensor ds18b20 dalam internet of things ," *Emitor: Jurnal Teknik Elektro*, vol. 22, no. 1, pp. 26–33, Mar. 2022, doi: 10.23917/emitor.v22i1.14928.

[30] M. Imam, E. Apriaskar, and Djuniadi, "Design and construction of a thermopen as a temperature measurement device using the DS18B20 sensor in the Internet of Things," (in Indonesian), 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:213556203

[31] Mrs. K. Anusha, "Internet of Things (IoT) based home automation using ESP32," *Int J Res Appl Sci Eng Technol*, vol. 12, no. 5, pp. 1672–1676, May 2024, doi: 10.22214/ijraset.2024.61912.

[32] A. B. Lasera and I. H. Wahyudi, "Development of a prototype IoT-based ESP32 electric power control system for a smart home system," (in Indonesian), *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 5, no. 2, pp. 112–120, Dec. 2020, doi: 10.21831/elinvo.v5i2.34261.

[33] Muliadi, Imran Al, and Rasul Muh, "Development of a smart trash bin using ESP32,", (in Indonesian), *Jurnal Media Elektrik*, vol. 17, no. 2, May 2020, doi: 10.22214/ijraset.2024.61912.

[34] W. Shalannanda, I. Zakia, F. Fahmi, and E. Sutanto, "Implementation of the hardware module of IoT-based infant incubator monitoring system," in *2020 14th International Conference on Telecommunication Systems, Services, and Applications (TSSA*, Nov. 2020, pp. 1–5. doi: 10.1109/TSSA51342.2020.9310901.