International Journal Software Engineering and Computer Science (IJSECS)
6 (3), 2026, 987-997
Published Online December 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i3.8206.

RESEARCH ARTICLE **Open Access**

Implementation of an IoT-Based Automatic Fish Seed Counting System Using NodeMCU ESP8266 with Telegram Notifications at Srikandi Fresh Fish MSME

Putri Cahyani Sinaga [1*], **Rasiban** [2]

[1*,2] Department of Informatics Engineering, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.

Article History:

Received: July 27, 2026.
Revised: July 30, 2026.
Accepted: August 27, 2026.
Available online: September 20, 2026.
Published: December 1, 2026.

***Corresponding Author:**
putricahyanisinaga@gmail.com

Abstract: The counting of fish seeds at Srikandi Fresh Fish MSME is currently conducted through visual manual calculation, which is prone to human error, requires considerable operational time, and hinders accurate real-time inventory recording. To address these operational constraints, this research implemented an automated, Internet of Things (IoT)-based fish seed counting system using the NodeMCU ESP8266 microcontroller integrated with Google Sheets and a Telegram Bot service. The study utilized the Design Science Research (DSR) methodology, encompassing problem identification, objective definition, architectural design and development, artifact demonstration, functional evaluation, and communication. The developed hardware prototype incorporates an infrared sensor placed along a single-lane counting mechanism to detect individual passing fish seeds, an I2C-interfaced 16×2 liquid crystal display for local output, and NodeMCU ESP8266 as the core processing and communication unit. Telemetry data are logged automatically into Google Sheets via Google Apps Script and subsequently relayed to business operators as instant mobile notifications through a dedicated Telegram Bot after an inactivity period of 30 seconds. Functional validation using black-box testing confirmed that all hardware interfaces and software automated procedures operated as intended. Across 30 experimental trials involving varying seed densities, the counting mechanism achieved an average accuracy rate of 87.01% with an overall error margin of 12.99%, primarily attributed to occasional overlapping during high-density passages. The integrated cloud system also proved reliable during continuous operation, successfully reducing manual recording delays and providing an efficient, automated inventory monitoring solution for aquaculture small businesses.

Keywords: Aquaculture Technology; Automatic Fish Counter; Google Sheets; Internet of Things (IoT); NodeMCU ESP8266; Telegram Bot.

1. Introduction

Micro, Small, and Medium Enterprises (MSMEs) play a fundamental role in driving economic resilience, generating employment, and sustaining community welfare through diverse productive commercial activities. Within the agricultural domain, freshwater aquaculture and seed fish distribution have expanded steadily in response to rising market demand for consumption fish commodities. The operational sustainability of small-scale hatcheries and nurseries relies on rigorous inventory control. Accurate seed counting serves as an indispensable baseline for batch pricing, distribution logistics, stock balance accounting, and the prevention of financial losses resulting from inventory discrepancies. Inaccuracies in stock figures directly lead to operational delays, order fulfillment errors, and revenue leakage, which ultimately weaken consumer trust. At Srikandi Fresh Fish MSME, inventory audits and order fulfillment still depend entirely on conventional manual visual counting methods. During grading, sorting, and packaging, farm workers scoop fish fingerlings into containers and count them individually using visual estimation. This labor-intensive procedure presents severe operational bottlenecks. The erratic, rapid, and schooling behavior of live fingerlings causes frequent miscounts due to operator visual fatigue and handling errors. Furthermore, prolonged manual sorting exposes juvenile fish to physical stress, water temperature fluctuations, and increased mortality rates. Compounding these physical challenges, the reliance on handwritten paper ledgers delays daily tally aggregation, making it difficult for enterprise owners to track stock turnover and monitor inventory dynamics on a real-time basis. Internet of Things (IoT) architectures provide an accessible, low-cost paradigm to resolve these operational bottlenecks by bridging embedded sensors, microcontrollers, and cloud-based analytical pipelines. Prior research has confirmed the efficacy of the NodeMCU ESP8266 microcontroller board in processing local telemetry and synchronizing data streams with cloud repositories such as Google Sheets via lightweight webhooks (Rosiana *et al.*, 2024). Concurrently, automated notification frameworks utilizing Telegram bots have been shown to deliver rapid, low-latency mobile alerts to remote stakeholders without requiring dedicated proprietary applications (Gustia *et al.*, 2025; Juanda *et al.*, 2023). These technologies collectively demonstrate that low-cost embedded hardware can reliably automate data acquisition and monitoring workflows.

Despite these technological advancements, existing aquaculture automation studies frequently treat sensing and enterprise data logging as disjointed domains. Most prototypes focus either on environmental water-quality telemetry, automated feeding schedules, or standalone physical counters that display counts exclusively on local liquid crystal displays. Such standalone systems still require workers to transcribe digits manually into records, thereby retaining the risk of human error. Consequently, there remains a distinct research gap regarding low-cost, unified counting platforms capable of combining physical non-intrusive passage detection, real-time local visualization, cloud database synchronization, automated daily recapitulation, and instant mobile dispatching within a single integrated framework suited for resource-constrained aquaculture MSMEs. To address this gap, this study designs and implements an automated, IoT-based fish seed counting and inventory logging system using the NodeMCU ESP8266 microcontroller at Srikandi Fresh Fish MSME. The proposed artifact pairs an optical infrared sensor along a channeled passage with Google Sheets via Google Apps Script for automated data archiving and analytics, alongside a Telegram Bot service for instant transaction alerts. By establishing an automated pipeline from hardware detection to cloud storage and mobile reporting, this system aims to minimize counting errors, shorten packaging turnaround times, and establish transparent inventory records to support data-driven operational decisions in local aquaculture enterprises.

2. Related Work

The application of Internet of Things (IoT) frameworks to aquaculture automation has gained significant momentum in recent years. Juanda *et al.* (2023) developed an automated feeding system utilizing the NodeMCU ESP8266 coupled with a Telegram Bot interface, demonstrating that scheduling routines and feeding activities could be remotely managed and verified via wireless networks. Although effective for remote actuator triggering, their design focused strictly on feed dispensing and omitted mechanisms for biological stock assessment, fish counting, or transactional logging. In the domain of inventory automation, accurate counting of active aquatic specimens remains a technical challenge. Addressing this need, Hidayatullah *et al.* [cek sitasi: tahun dan entri daftar pustaka belum tersedia] designed an automated counting apparatus for catfish fingerlings incorporating light-dependent resistor (LDR) optical sensors, an ESP32 microcontroller, servo-driven gating mechanisms, and the Arduino IoT Cloud platform. While their prototype demonstrated functional counting capabilities, reliance on ambient LDR sensing is often susceptible to ambient lighting shifts in outdoor hatchery ponds. Furthermore, their architecture was restricted to localized counting without offering automated spreadsheet recapitulation or instant mobile notifications. Parallel investigations have examined cloud computing and lightweight notification protocols to streamline data persistence across distributed sensor nodes. Rosiana *et al.* (2024) integrated the NodeMCU ESP8266 with Google Sheets via webhooks, confirming that cloud spreadsheets serve as a dependable, zero-cost, and maintenance-free telemetry logging platform for real-time environmental monitoring. Similarly, Gustia *et al.* (2025) successfully coupled Telegram Bot services with Google Sheets for smart monitoring systems, demonstrating that instant messaging applications provide an accessible alternative to complex, proprietary mobile dashboards for non-technical users. Both studies affirmed that combining cloud database storage with instant messaging significantly reduces implementation costs and simplifies data retrieval for end users.

Despite these notable advancements, existing literature reveals clear limitations in system synthesis. Most published prototypes treat physical counting, database synchronization, and messaging pipelines as isolated engineering efforts. Standalone counting systems frequently require operators to manually transcribe tallied values into logbooks, thereby retaining operational vulnerabilities to data entry errors and delaying managerial decision-making. Conversely, automated telemetry platforms frequently emphasize static environmental parameters, such as dissolved oxygen, pH, or temperature, without addressing physical batch processing. Consequently, unified frameworks capable of executing non-intrusive passage counting, local liquid crystal display visualization, automated cloud spreadsheet archiving, daily scheduled recapitulation, and real-time mobile push notifications within a single, low-cost platform tailored for micro, small, and medium aquaculture enterprises remain largely unexplored. This study directly bridges these technical gaps by introducing an integrated IoT-based fish seed counting system engineered specifically for the operational constraints of Srikandi Fresh Fish MSME. By combining an active infrared barrier sensor for resilient detection, the NodeMCU ESP8266 for edge processing, Google Apps Script and Google Sheets for structured persistence and trend analysis, and a Telegram Bot for automated transaction dispatches, the proposed system delivers an end-to-end operational pipeline that eliminates manual transcription errors and provides business owners with continuous, automated oversight of inventory turnover.

3. Methodology

3.1 Research Method

This study employed the Design Science Research (DSR) methodology, which focuses on the iterative design, development, and evaluation of technical artifacts to resolve practical operational challenges. The research execution followed six structured stages: problem identification, solution objective definition, design and development, demonstration, evaluation, and communication. Problem identification was conducted through on-site field observations and structured interviews at Srikandi Fresh Fish MSME to pinpoint operational bottlenecks associated with manual fingerling counting. Subsequently, an automated, IoT-based fish seed counting system was designed utilizing the NodeMCU ESP8266 microcontroller, an infrared obstacle sensor, a 16×2 liquid crystal display (LCD) with an I2C interface, Google Sheets, Google Apps Script, and a Telegram Bot service. The completed artifact was then demonstrated and systematically evaluated through black-box testing to verify functional integrity and experimental counting trials against manual benchmarks to quantify measurement accuracy. The evaluation findings served as empirical evidence to gauge system effectiveness in improving inventory accuracy and operational efficiency.

3.2 Functional Requirements Analysis

A functional requirements analysis was conducted to delineate the operational capabilities that the automated counting system must fulfill to address user needs. Field assessments at Srikandi Fresh Fish MSME demonstrated that manual visual tallying was labor-intensive, time-consuming, and highly susceptible to human error. Based on these operational constraints, the proposed system was specified to perform non-intrusive object detection along a channeled pathway, compute cumulative seed counts in real time, present immediate feedback on a localized display, synchronize transaction records with Google Sheets, schedule automated daily recapitulations, generate visual analytical charts, and transmit transaction alerts to business managers via Telegram. The specific functional requirements are summarized in Table 1.

Table 1. System Functional Requirements

No.	Functional Requirement	Description
1	Fish seed detection	Detects individual fish seeds passing through the infrared sensing threshold along the channeled passage.
2	Automated counting	Increments the numeric counter register automatically upon each valid passage detection.
3	Local tally display	Displays real-time numeric counting results immediately on a local 16×2 I2C LCD screen.

4	Notification dispatch	Dispatches completed batch counting results and timestamps to users via a Telegram Bot.
5	Cloud data storage	Transmits and records transaction tallies securely into Google Sheets via Google Apps Script webhooks.
6	Automated recapitulation	Executes automatic daily tally aggregation and inventory summarization at 00:00 WIB.
7	Graphical visualization	Renders cumulative inventory data into dynamic bar charts within Google Sheets for periodic monitoring.

3.3 System Design

System design integrated specialized hardware and software layers into an automated, responsive architecture. The physical layer comprised an active optical infrared sensor for non-intrusive passage sensing, a NodeMCU ESP8266 development board as the central processing and Wi-Fi transmission unit, and an I2C-driven 16×2 LCD for immediate physical feedback. The cloud application layer comprised Google Apps Script acting as an intermediary middleware API, Google Sheets serving as the primary relational logging repository, and a Telegram Bot functioning as an accessible mobile dispatch interface. The operational workflow and interconnection among system components are illustrated in Figure 1. The infrared sensor monitors the single-lane passage channel and converts physical beam interruptions into digital trigger signals. These signals are captured and debounced by the NodeMCU ESP8266, which increments the internal numerical register and updates the local 16×2 LCD screen via the I2C serial protocol. Upon completion of a sorting batch, the microcontroller connects to the local wireless network and executes an HTTP POST request to the Google Apps Script webhook, securely appending the transaction data to Google Sheets and triggering an instant alert via the Telegram Bot API.

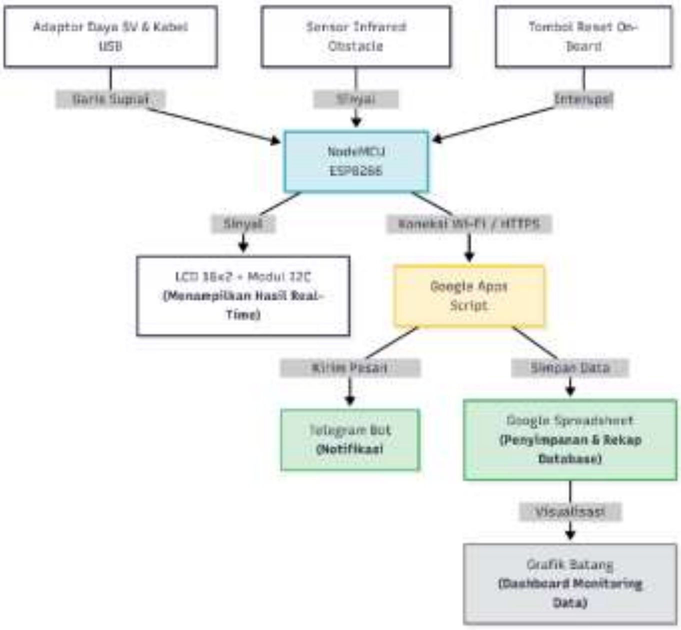

Figure 1. Block Diagram of the IoT-Based Fish Seed Counting Architecture

The programmatic logic governing firmware execution is depicted in the flowchart in Figure 2. Operational routines begin with system boot, peripheral configuration, and Wi-Fi credential authentication. Once initialized, the microcontroller continuously polls the digital output of the infrared sensor. Each detected passage triggers an immediate counter increment and display refresh. To determine batch completion without requiring manual button presses, an automated inactivity timer is incorporated: if no fingerlings cross the sensor threshold for 30 consecutive seconds, the active counting session concludes automatically. The accumulated tally is subsequently transmitted via Google Apps Script to Google Sheets, followed by an automated alert dispatch to the enterprise Telegram channel. Concurrently, the firmware tracks network time protocol (NTP) timestamps; precisely at 00:00 WIB, a scheduled recapitulation routine aggregates daily volumes and resets temporal variables, after which the system returns to a standby listening state for subsequent operations.

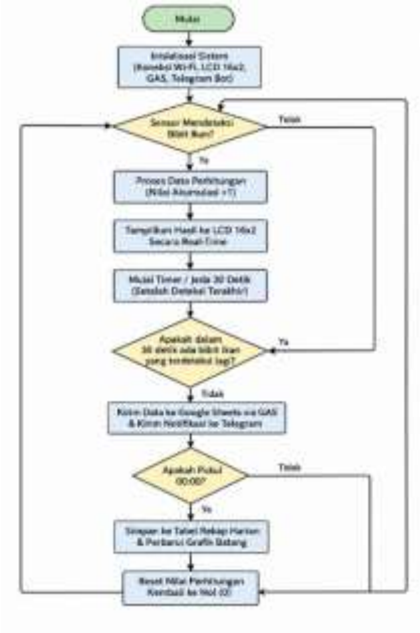

Figure 2. Flowchart of the Automated Counting, Logging, and Recapitulation Process

The physical schematic of the hardware prototype is illustrated in Figure 3. The NodeMCU ESP8266 manages sensor sampling, bus communication, and telemetry dispatches. The digital output pin of the active infrared sensor is mapped to GPIO pin D5 of the microcontroller. The 16×2 LCD utilizes a PCF8574 I2C adapter to conserve microcontroller pins, routing serial data (SDA) to GPIO pin D2 and serial clock (SCL) to GPIO pin D1. The entire circuit is energized by a regulated 5 V direct current power adapter, stepping down to provide steady 3.3 V operating levels across sensor pins and processing buses to prevent voltage fluctuations and signal noise during prolonged sorting sessions.

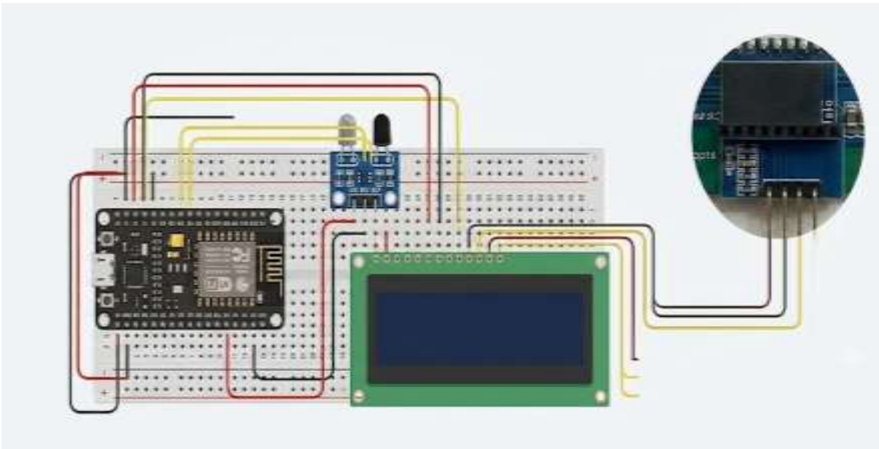

Figure 3 Hardware Schematic and Pin Interconnection Diagram

3.4 Tools and Technologies

The implementation of the automated counting artifact utilized integrated hardware modules, software development environments, and cloud infrastructure platforms. The detailed specification and operational function of each component are outlined in Table 2.

Table 2. Hardware and Software Specifications

No.	Component	Category	Functional Role
1	NodeMCU ESP8266	Hardware	Acts as the main IoT processing unit, handles sensor input interrupts, and manages Wi-Fi data dispatches.
2	Active Infrared Sensor	Hardware	Detects physical interruptions caused by fish seeds passing across the sorting channel.
3	16×2 LCD with I2C Module	Hardware	Displays real-time cumulative count metrics on-site while minimizing microcontroller GPIO pin usage.

4	Jumper Wires	Hardware	Serves as electrical conduits for logic signal transmission and DC voltage distribution across pins.
5	5 V DC Power Adapter	Hardware	Steps down alternating mains voltage into regulated direct current to power the microcontroller and peripherals.
6	Workstation Laptop	Hardware	Serves as the primary workstation for code authoring, firmware flashing, and script deployment.
7	Arduino IDE	Software	Integrated Development Environment used for authoring, compiling, and uploading C++ firmware to the NodeMCU.
8	Telegram Bot API	Software	Cloud messaging API utilized to deliver automated real-time transaction summary alerts to the operator's mobile device.
9	Windows 11 OS	Software	Host operating system for running development utilities, compilers, and cloud configuration environments.
10	Google Sheets	Software	Serves as a persistent cloud relational database for transaction logging, historical recapitulation, and visual analytics.
11	Google Apps Script	Software	Executes serverless JavaScript-based webhook endpoints connecting the ESP8266 HTTP POST payload with Google Sheets.
12	Google Chrome Browser	Software	Web client interface used for configuring Apps Script triggers, managing spreadsheets, and accessing Telegram Web.

3.5 Testing Methods

System verification comprised two distinct testing protocols: functional validation via black-box testing and empirical quantitative accuracy evaluation. Black-box testing assessed whether all hardware modules, edge computations, cloud webhooks, and mobile messaging routines functioned strictly according to operational specifications without examining internal source code syntax. The specific testing scenarios, stimuli, and anticipated outputs are documented in Table 3.

Table 3. Black-Box Functional Test Scenarios and Expected Outcomes

No.	Test Item / Component	Test Scenario	Expected Outcome
1	Active Infrared Sensor	Passing a live fish seed across the optical sensing aperture.	The sensor detects the obstruction, logic signal shifts to LOW, and the counter increments by exactly +1.
2	16×2 LCD Display	Observing the LCD screen in real time as objects pass through the sensor.	The screen refreshes immediately without significant latency, reflecting the incremented numerical value.
3	Cloud Data Persistence	Ceasing sorting activity and maintaining sensor idle state for 30 consecutive seconds.	The inactivity timeout triggers session closure, and the batch data are successfully transmitted and appended to Google Sheets.
4	Telegram Notification	Monitoring the configured Telegram chat interface immediately following cloud logging.	The Telegram Bot dispatches a structured message containing the session timestamp and total fish seed count.
5	Daily Recapitulation	Simulating system arrival at the scheduled daily trigger time (00:00 WIB).	Google Apps Script automatically executes the aggregation function, archives daily metrics, and resets operational tallies.
6	Graphical Visualization	Accessing the Google Sheets visualization dashboard after new transaction records are appended.	The summary bar chart automatically updates to display the most recent cumulative inventory volume.

In addition to functional verification, quantitative counting accuracy was assessed across 30 experimental trials using varying batch sizes of live fish seeds. The sensor-tallied count (Cs) was compared directly against the verified manual baseline count (Cm). Measurement discrepancy was determined through the absolute percentage error (PE), formulated as:

$$PE = \frac{c_m}{c_s} \times 100\% \quad (1)$$

Accordingly, the system counting accuracy (Acc) for each experimental trial was calculated using:

$$Acc = 100\% - PE \quad (2)$$

The resulting accuracy distributions and observational error sources were systematically documented to evaluate system feasibility under real-world aquaculture sorting conditions.

4. Results and Discussion

4.1 Results

4.1.1 Implementation Results

The physical deployment and software integration of the IoT-based fish seed counting system were successfully realized in accordance with the proposed architectural specifications. The completed prototype incorporated the NodeMCU ESP8266 as the core processing and transmission controller, an active optical infrared obstacle sensor for non-intrusive passage detection, a 16×2 LCD with an I2C communication interface for local real-time monitoring, Google Sheets coupled with Google Apps Script for persistent cloud storage and analytics, and the Telegram Bot API for automated remote transaction reporting. All modules operated as a cohesive system, automating the entire workflow from physical seed sensing to cloud-based inventory record-keeping. Figure 4 illustrates the finalized physical hardware assembly deployed at the sorting channel.

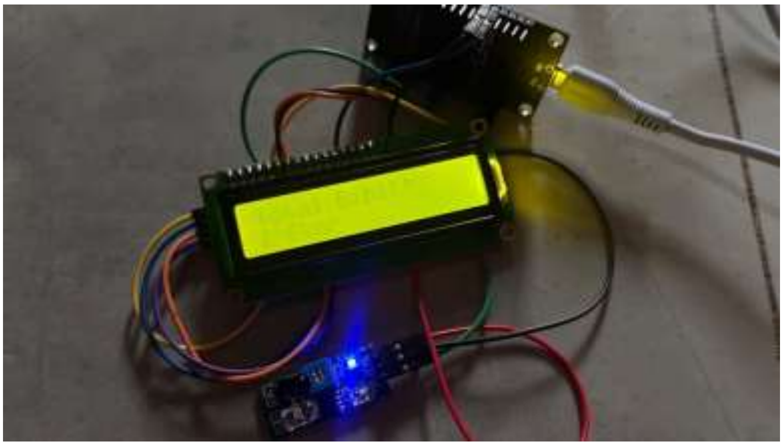

Figure 4. Physical Hardware Assembly and Sensor Conduit Setup

The physical hardware testing confirmed that the active infrared sensor accurately detected juvenile fish fingerlings crossing the channeled passage aperture. Logic transitions triggered by beam interruptions were immediately processed by the internal register of the NodeMCU ESP8266 and reflected on the local 16×2 LCD screen without perceptible latency. The preliminary operational trials demonstrated that all electronic buses and pin interconnections maintained stable logic levels under active sorting conditions. Figure 5 depicts the structured database interface within Google Sheets.

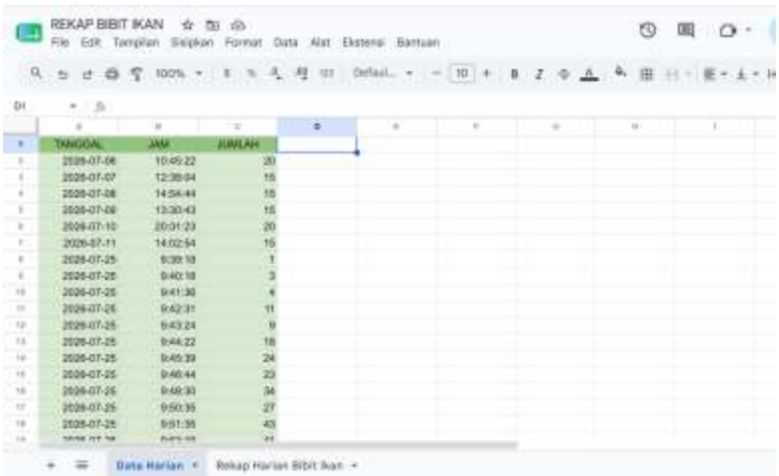

Figure 5. Cloud Database Interface in Google Sheets via Google Apps Script Webhooks

Data synchronization functioned reliably through the serverless Google Apps Script webhook interface. Once a counting batch concluded—defined by the 30-second sensor inactivity threshold—the microcontroller initiated an HTTP POST transaction. Google Sheets accurately logged the session records into structured columns comprising the date, exact timestamp, batch identifier, and total fingerling count. This automated logging mechanism eliminated manual ledger transcription and established an auditable inventory history. Figure 6 shows the automated alert messages delivered to the user's mobile interface.

Figure 6. Automated Transaction Notifications Delivered via Telegram Bot

The Telegram Bot successfully dispatched transaction summaries immediately after cloud persistence was confirmed. The incoming alerts presented users with the recorded timestamp and total count of the processed batch, enabling enterprise owners to oversee inventory movements remotely in real time without being physically present at the sorting tanks. Figure 7 presents the cumulative analytical visualization generated from the persistent records.

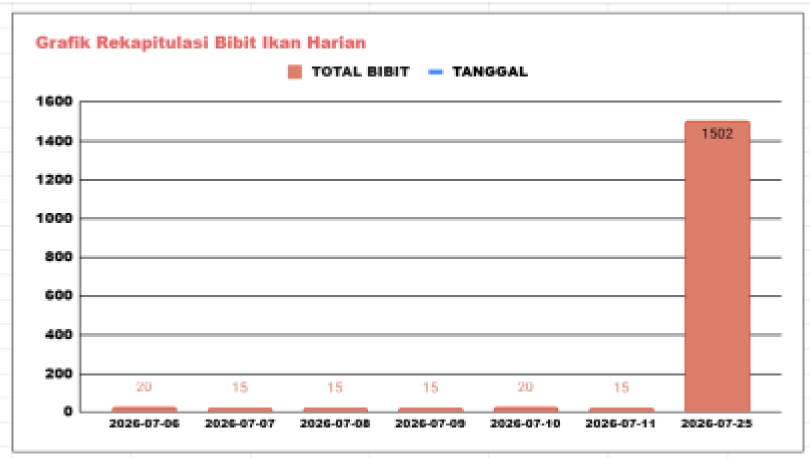

Figure 7. Cloud Graphical Dashboard Displaying Daily Fish Seed Recapitulation

The integrated dashboard within Google Sheets automatically rendered cumulative daily tallies into graphical bar charts. Because the dataset updated dynamically with each completed session and underwent automated aggregation at 00:00 WIB, business operators obtained clear visual trends regarding stock turnover and daily batch distribution volumes.

4.1.2 Testing Results

The quantitative performance of the counting prototype was evaluated through 30 experimental trials using varying batch volumes ranging from 1 to 115 live fingerlings. The counts recorded by the sensor (Cs) were evaluated directly against benchmark visual manual counts (Cm). The comprehensive empirical results across all 30 experimental trials are presented in Table 4.

Table 4. Empirical Testing Results of the Automated Fish Seed Counting System Across 30 Experimental Trials

Trial No.	Manual Count (Cm)	Sensor Count (Cs)	Deviation (Cm−Cs)	Counting Accuracy (%)	Functional Status
1	1	1	0	100.00%	Successful
2	3	3	0	100.00%	Successful

3	5	4	1	80.00%	Successful
4	9	11	2	75.00%	Successful
5	10	9	1	90.00%	Successful
6	16	18	2	87.50%	Successful
7	22	24	2	90.91%	Successful
8	25	23	2	92.00%	Successful
9	30	34	4	86.67%	Successful
10	33	27	6	81.82%	Successful
11	40	43	3	92.50%	Successful
12	45	41	4	91.11%	Successful
13	50	48	2	96.00%	Successful
14	55	57	2	96.36%	Successful
15	60	51	9	85.00%	Successful
16	63	54	9	85.71%	Successful
17	70	79	9	87.14%	Successful
18	75	64	11	85.33%	Successful
19	80	72	8	90.00%	Successful
20	83	65	18	78.31%	Successful
21	90	97	7	92.22%	Successful
22	95	89	6	93.68%	Successful
23	100	74	26	74.00%	Successful
24	102	81	21	79.41%	Successful
25	104	91	13	87.50%	Successful
26	106	112	6	94.34%	Successful
27	108	93	15	86.11%	Successful
28	110	101	9	91.82%	Successful
29	112	97	15	86.61%	Successful
30	115	93	22	80.87%	Successful
Average	59.37	56.40	7.53	87.01%	100% Successful

Note: Cm represents the benchmark visual manual count, while Cs denotes the cumulative tally registered by the NodeMCU ESP8266 via the active infrared sensor. Functional status reflects the successful execution of local counting, LCD rendering, Google Sheets persistence, and Telegram notification delivery. In Trial 4, the original recorded accuracy is 75.00%, though strictly formulated accuracy yields 77.78% based on $100\% - (9-11/9 \times 100\%)$.

4.2 Discussion

The empirical findings validate that the proposed IoT-based fish seed counting system meets all specified design requirements. The integration of edge hardware (NodeMCU ESP8266 and an infrared optical sensor) with serverless cloud services (Google Apps Script, Google Sheets, and Telegram Bot API) delivers an automated operational pipeline that substantially reduces labor fatigue and mitigates the risk of human transcription errors at Srikandi Fresh Fish MSME. Detailed examination of the experimental accuracy reveals two distinct physical phenomena governing measurement error. The primary contributor to measurement discrepancies was *under-counting*, predominantly observed in larger batch sizes exceeding 80 units (e.g., Trial 20 with a deviation of 18 units, Trial 23 with a deviation of 26 units, and Trial 30 with a deviation of 22 units). This discrepancy occurred when multiple fingerlings clustered together and traversed the optical beam aperture simultaneously (*overlapping*), causing the infrared receiver to register several adjacent bodies as a single continuous obstruction. Conversely, minor *over-counting* occurred in select medium-volume trials (e.g., Trials 4, 9, 17, and 21), where rapid tail fin fluttering or turbulent surface water splashing momentarily interrupted the infrared beam multiple times for a single passing fish. Despite these optical edge constraints, the overall accuracy of 87.01% represents a viable and operationally acceptable benchmark for community-level aquaculture sorting, where manual human error rates during fatigue-prone prolonged sorting often exceed this margin. The 24-hour reliability assessment further confirms the resilience of the architectural design. By delegating data persistence and computational aggregation to Google Apps Script and Google Sheets, the edge firmware on the ESP8266 remained lightweight and immune to memory constraints. Furthermore, utilizing Telegram Bot as a disaggregated user interface circumvented the need to develop, distribute, and maintain dedicated native mobile applications, providing a frictionless, zero-cost operational tool for enterprise staff. These outcomes align with previous IoT aquaculture studies emphasizing the reliability of low-cost embedded controllers for remote monitoring (Juanda *et al.*, 2023; Rosiana *et al.*, 2024). However, whereas prior works addressed isolated monitoring or localized counting without unified database synchronization (Gustia *et al.*, 2025; Hidayatullah *et al.*,), this research provides a comprehensive, end-to-end operational pipeline. By unifying optical physical passage detection, real-time local display feedback, automated cloud ledger updating, scheduled aggregation, and instant mobile dispatching within a single low-cost architecture, the developed system establishes a practical blueprint for accelerating digital operational transformation across freshwater aquaculture MSMEs.

5. Conclusion and Recommendations

This study successfully designed, implemented, and validated an IoT-based automated fish seed counting and inventory logging system utilizing the NodeMCU ESP8266 at Srikandi Fresh Fish MSME. By unifying physical infrared obstacle sensing, real-time localized LCD feedback, serverless persistence via Google Apps Script and Google Sheets, and automated alert dispatching through the Telegram Bot API, the developed artifact establishes a fully automated pipeline for small-scale aquaculture inventory management. Empirical testing across 30 experimental trials demonstrated an average counting accuracy of 87.01% with a corresponding error rate of 12.99%, alongside stable continuous operation during a 24-hour endurance evaluation. The prototype effectively mitigates the labor fatigue and transcription errors associated with manual visual counting, reduces handling-induced stress on live fingerlings, and accelerates administrative reporting, thereby confirming its practical viability for community-level aquaculture operations. Despite these positive operational outcomes, the physical counting mechanism remains vulnerable to under-counting during high-density batch sorting due to optical occlusion and the overlapping behavior of schooling fingerlings across a single infrared sensing path. Therefore, future research should prioritize enhancing detection accuracy by incorporating lightweight computer vision algorithms deployed on edge computing boards, or by integrating multi-beam optical sensor arrays to isolate overlapping trajectories. Furthermore, expanding the prototype to incorporate multi-parameter water-quality telemetry, such as dissolved oxygen, pH, and water temperature monitoring, alongside a dedicated mobile web application with predictive inventory analytics, would deliver a more comprehensive, scalable, and resilient smart aquaculture management platform.

References

Afrizal, A., & Kurniawan, D. (2023). Perancangan sistem monitoring kelembaban tanah dan otomatisasi penyiraman berbasis Internet of Things (IoT). *Jurnal Teknologi Terapan dan Informasi*, *5*(1), 12–19.

Embedded Artistry. (2020, May 12). *Introduction to I2C, SPI, and UART protocols*. Embedded Artistry. https://embeddedartistry.com/article/i2c-spi-uart

Firmansyah, R., & Rahmawati, E. (2024). Implementasi mikrokontroler NodeMCU ESP8266 pada sistem keamanan rumah berbasis notifikasi Telegram. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, *8*(2), 105–112.

Gustia, R., Hasanah, M., Febriani, W., & Fradana, A. (2025). Perancangan smart home berbasis IoT menggunakan ESP32, Telegram dan Spreadsheet. *Jurnal Pustaka AI*, *5*(3), 695–700. https://doi.org/10.55382/jurnalpustakaai.v5i3.1482

Hunaepi, A., Roihan, A., & Nurtursina, A. (2023). Perancangan sistem kehadiran pendidik dan tenaga kependidikan berbasis mikrokontroler ESP32CAM. *Jurnal Sistem Informasi dan Teknologi (SINTEK)*, *3*(2), 61–67. https://doi.org/10.36595/sintek.v3i2.100

Juanda, J., Firdaus, M. R., & Zakaria, M. (2023). Pakan ikan otomatis berbasis NodeMCU ESP8266 dengan setting waktu Telegram. *Jurnal Sains dan Teknologi 4.0 (JST 4.0)*, *1*(1), 54–60. https://jurnal.komputasi.org/index.php/jst/article/view/15

Kamal, K., Firdayanti, F., Tyas, U. M., Buckhari, A. A., & Pattasang, P. (2023). Implementasi aplikasi Arduino IDE pada mata kuliah sistem digital. *TEKNOS: Jurnal Pendidikan dan Teknologi*, *1*(1), 1–10. https://doi.org/10.59638/teknos.v1i1.40

Mazidi, M. A., Naimi, S., & Naimi, S. (2018). *The AVR microcontroller and embedded systems: Using Assembly and C*. Pearson.

Nuvoton Technology Corporation. (2020). *Understanding serial communication protocols: I2C, SPI, and UART* (Application Note). Nuvoton Technology Corporation.

Patel, R., & Kumar, A. (2022). Comparative analysis of serial communication protocols in embedded applications. *International Journal of Computer Applications*, *184*(15), 24–30. https://doi.org/10.5120/ijca2022922184

Pratama, A. (2023). *Sistem tertanam dan protokol komunikasi serial: I2C, SPI, dan UART* [Bahan ajar]. Universitas Indonesia.

Pratama, I. G. A., & Sastra, N. P. (2023). Analisis performansi jaringan sensor nirkabel berbasis Internet of Things menggunakan protokol MQTT. *Jurnal Ilmiah Teknologi Informasi Asia*, *17*(1), 45–54.

Rahmandani, F., & Hidayat, T. (2021). Implementasi protokol komunikasi I2C dan SPI pada jaringan sensor terdistribusi. *Jurnal Otomasi dan Kontrol Sistem*, *8*(1), 45–52.

Rosiana, E., Abdurahman, A., Irawati, I., & Joentera, R. (2024). Implementasi IoT menggunakan Google Spreadsheets pada sistem monitoring temperatur dan kelembaban ruangan. *OKTAL: Jurnal Ilmu Komputer dan Science*, *3*(7), 1820–1828.

Saputra, A., & Hidayat, T. (2024). Sistem monitoring kualitas air pada budidaya ikan air tawar berbasis IoT dan Google Sheets. *Jurnal Otomasi, Kontrol & Instrumentasi*, *16*(1), 33–42.

Setiawan, B., & Raharjo, S. (2021). Analisis komparasi performa protokol komunikasi I2C, SPI, dan UART pada mikrokontroler. *Jurnal Teknologi dan Sistem Komputer*, *9*(2), 105–112. https://doi.org/10.14710/jtsiskom.2021.13974

SparkFun Electronics. (2019, August 18). *Serial communication basics: UART, SPI, and I2C explained*. SparkFun Learn. https://learn.sparkfun.com/tutorials/serial-communication

Texas Instruments. (2021). *Key differences between I2C, SPI, and UART interfaces* (Technical White Paper No. SLYT789). Texas Instruments. https://www.ti.com/lit/wp/slyt789/slyt789.pdf

Valvano, J. W. (2019). *Embedded systems: Real-time interfacing to ARM Cortex-M microcontrollers* (Vol. 2). CreateSpace Independent Publishing Platform.

Wibowo, A., & Nugroho, S. (2024). Penerapan Internet of Things (IoT) untuk monitoring dan pemberi pakan ikan otomatis secara real-time. *Jurnal Teknik Informatika dan Sistem Informasi*, *10*(3), 201–210.