Vol.02 No.02 (2026) 62-72

Sustainable in Energy Science and Technology

https://ojs.polmed.ac.id/index.php/SiEST

IoT-Based Automatic Chicken Feeding System Using ESP32 with Telegram Bot Notification and Web Dashboard Monitoring

Bintang Risky Deni[1*], Afritha Amelia[2], Roslina[3], Mazlina Abdul Majid[4],Samsul Bahri[5]

[1]*Program Study Electrical, Departement of Electrical Engineering, Politeknik Negeri Medan, Medan, 20155, Indonesia*

[2]*Program Study of Telecommunication Network Engineering and Technology, Departement of Electrical Engineering, Politeknik Negeri Medan, Medan, 20155, Indonesia*

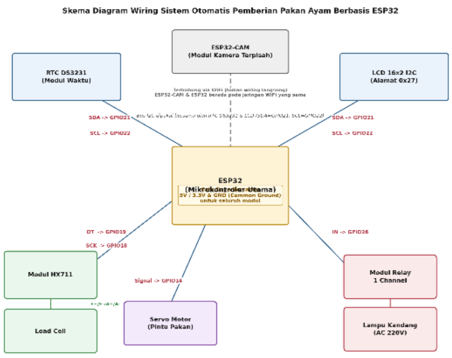
[3]*Program Study of Informatics Management, Departement of Computer and Informatics, Politeknik Negeri Medan, Medan, 20155, Indonesia*

[4]*Faculty of Computer- Centre for Artificial Intelligence & Data Science, Universiti Malaysia Al-Sultan Abdullah, Malaysia*

[5]*Program Study of Mechanical Engineering, Universitas Syiah Kuala, Banda Aceh, 23111, Indonesia*

Corresponding Author: bintangriskydeni@gmail.com

Graphical Abstract



Highlights

- An Esp32-based Iot system automates chicken feeding, coop lighting, and remote monitoring through a telegram bot and a web dashboard.
- The load cell (Hx711) achieved feed-weighing accuracy with an average error of approximately 1.0 %.
- The telegram bot responded to user commands within one minute, consistent with its 3 second polling design (theoretically under 5 Seconds).The system sustained continuous operation up to 18 hours 16 minutes between restarts and was field-tested on a physical coop for more than 3 days.

ARTICLE INFO

Keywords:
IoT; ESP32; automatic poultry feeder; Telegram Bot; smart farming

ABSTRACT

Manual feeding practices in small-scale poultry farming often result in inconsistent feeding schedules, feed waste, and inefficient labour use, particularly when farmers cannot be physically present at the coop. This study designs and implements an Internet of Things (IoT)-based automatic chicken feeding system using the ESP32 microcontroller. The system integrates a load cell with HX711 amplifier for feed-weight monitoring, a servo motor

Article history:

Received June 17, 2026

Revised July 29, 2026

Accepted August 17, 2026

Available online August 17, 2026

https://doi.org/10.51510/siest.v2i2.3373

for automated feed dispensing, a DS3231 real-time clock for scheduling, an ESP32-CAM for visual monitoring, a relay module for automated coop lighting, a web-based dashboard, and a Telegram Bot for two-way remote interaction. Testing consisted of functional tests of each component, system-level testing across ten operational scenarios, notification-timing tests, a 24-hour-plus stability log, and field testing on a physical coop for more than three days. Results show the load cell achieved an average error of approximately 1.0 %, the system achieved a 100 % success rate across ten scenarios, the Telegram Bot responded to commands within one minute (consistent with its 3-second polling design, theoretically under 5 seconds), and the system sustained continuous operation for up to 18 hours 16 minutes between manual restarts, with a total observation period exceeding 24 hours. Field testing on a physical coop for more than three days confirmed reliable operation outside laboratory conditions. These findings indicate that the proposed system offers a low-cost, integrated solution for automating feed scheduling, environmental monitoring, and remote interaction for small- to medium-scale poultry farms.

1. Introduction

Poultry farming, particularly small- and medium-scale chicken rearing, remains an important economic activity in many regions of Indonesia. However, feeding is commonly still performed manually, which depends heavily on the farmer's presence and availability. Manual feeding is prone to delays, inconsistent scheduling, and difficulty in monitoring remaining feed stock, especially for farmers who must be away from the coop for extended periods. These limitations reduce farm efficiency and can negatively affect livestock growth consistency. Recent advances in the Internet of Things (IoT) have significantly transformed precision livestock farming by enabling intelligent sensing, real-time communication, and automated decision-making. IoT-based livestock systems not only reduce manual intervention but also support data-driven farm management capable of improving productivity, resource efficiency, and animal welfare [1].

Prior studies have explored automation in this domain from different angles. Agustina et al. developed an IoT-based smart feeder capable of automatically dispensing feed according to predefined schedules while monitoring feed availability inside the hopper. Although the proposed system effectively reduced feeding delays, it did not provide visual monitoring of poultry house conditions [2]. Setiawan et al. implemented a microcontroller-based automatic feeding system for poultry farms and reported improved operational efficiency compared to conventional feeding [3]. Surahman et al. developed an IoT-based automatic chicken feeder allowing farmers to monitor and schedule feeding remotely through a smartphone application

[4]. Heryanto automated feeding for laying-hen farms and found that automation improved time efficiency and feeding consistency relative to manual methods [5]. On the environmental-monitoring side, Rohman and Isnaini developed an IoT-based temperature and humidity control system for broiler coops [6], while Turesna et al. designed a monitoring system for coop temperature and humidity to improve broiler productivity [7]. For remote notification, Tisna et al. demonstrated that a Telegram-based chatbot integrated with ESP32 could achieve an average response time of about one second for water-quality monitoring and control [8], and Yulita and Afriansyah showed that ESP32-CAM could provide real-time visual monitoring for home security applications at low cost [9].Despite this body of work, most existing systems address feeding automation, environmental monitoring, visual monitoring, and remote notification as separate, single-purpose solutions. Few studies integrate feed scheduling, feed-weight monitoring, lighting control, camera-based monitoring, two-way chatbot interaction, and a web dashboard into a single coherent low-cost system, and even fewer report field validation on a physical coop over an extended period rather than laboratory-only testing.

This study addresses that gap by designing, implementing, and testing an integrated ESP32-based system that combines scheduled and manually triggered feeding, feed-weight monitoring, coop lighting control, ESP32-CAM-based visual monitoring, a web dashboard, and a Telegram Bot for two-way interaction and notification. The system was evaluated through component-level testing, system-level scenario testing, notification-timing analysis, an extended stability log, and field testing on a physical coop for more than three days, with the aim of providing a practical, low-cost, and reproducible reference design for small- to medium-scale poultry automation.

2. Materials and Methods

The system was developed using an ESP32 microcontroller as the central controller, selected for its integrated Wi-Fi capability, sufficient processing power, and low cost compared to alternative platforms. The complete hardware configuration consists of: (i) a load cell with an HX711 amplifier module for feed-weight measurement; (ii) an MG996R servo motor coupled to a feed-dispensing mechanism; (iii) a two-channel relay module for coop-lighting control; (iv) a DS3231 real-time clock module for schedule-keeping; (v) a 16x2 I2C LCD for local status display; (vi) an ESP32-CAM module for visual monitoring, connected to the main controller over the local Wi-Fi network rather than by direct GPIO connection; and (vii) a Wi-Fi network used both to host a local web dashboard and to communicate with the Telegram Bot API. Table 1 summarises the equipment used in this study.

Table 1. Equipment and components used in this study

No.	Component	Qty	Function / Notes
1	ESP32 DevKit V1	1	Main microcontroller (Wi-Fi enabled); central controller of the system

No.	Component	Qty	Function / Notes
2	ESP32-CAM	1	Visual monitoring module; connected to the main controller via the local Wi-Fi network (not a direct GPIO connection)
3	Load Cell 5 kg	1	Feed-weight sensing element
4	HX711	1	Load-cell amplifier/ADC; data line GPIO19, clock line GPIO18
5	RTC DS3231	1	Real-time clock for feed/lighting scheduling; I2C bus (SDA: GPIO21, SCL: GPIO22, shared with LCD)
6	Servo Motor MG996R	1	Actuator opening/closing the feed-dispensing channel; PWM signal on GPIO14
7	Relay 2-Channel	1	Channel 1 controls coop lighting (GPIO26); Channel 2 reserved, not used in the current program
8	LCD 16x2 I2C	1	Local status display; I2C bus (SDA: GPIO21, SCL: GPIO22, shared with RTC)
9	Adaptor 5V	1	Power supply for the system
10	Lampu LED (coop lighting)	1	Coop lighting, switched by relay Channel 1

The ESP32 firmware was developed using the Arduino IDE. Feeding is triggered either automatically according to a fixed daily schedule (07:00, 12:00, 17:00, and 20:00) read from the DS3231 RTC, or manually through a command sent via the Telegram Bot. Coop lighting is switched on and off automatically at fixed times (18:30 and 05:30 respectively) through the relay module. The system continuously reads the load cell to report the remaining feed weight and sends a low-feed warning notification through Telegram when the weight falls below a defined threshold. Incoming Telegram commands are checked using a polling mechanism executed every 3 seconds, after which any new command is processed and a reply is sent through the Telegram Bot API; this design choice directly bounds the expected response latency of the system shown in Figure 1.

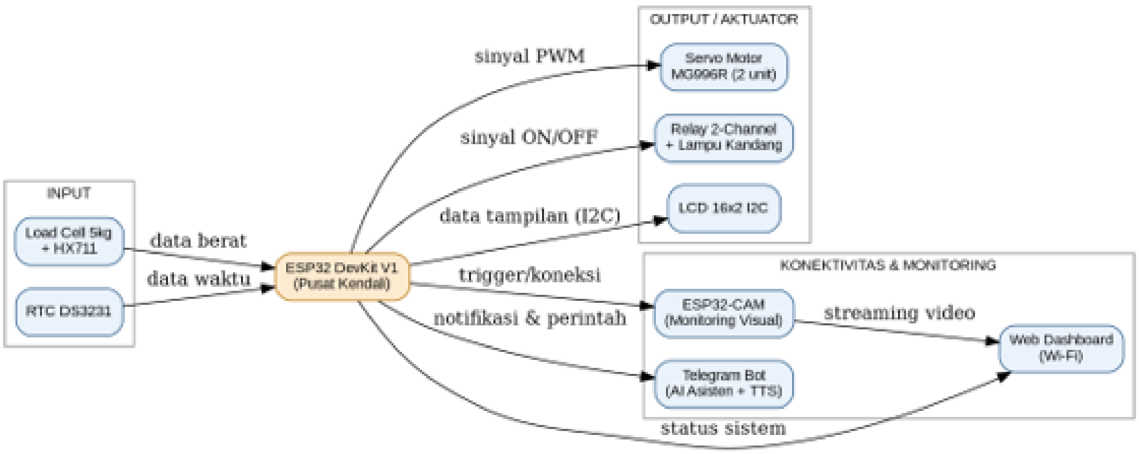


Figure 1. System block diagram showing input sensors, the ESP32 controller, output/actuators, and connectivity/monitoring modules

Testing was carried out in five stages: (1) component-level functional testing of the LCD, RTC, servo motor, relay module, Wi-Fi connectivity, and Telegram Bot; (2) system-level testing across ten operational scenarios covering scheduled feeding, manual feeding, mode switching, and lighting control; (3) notification-timing analysis based on Telegram Bot chat logs, comparing the scheduled time against the timestamp recorded in the RTC-based message content; (4) an extended system-stability log based on boot notifications recorded over a 29-30 June 2026 observation period; and (5) field testing in which the complete system was installed on a physical coop and operated continuously for more than three days to validate performance outside laboratory conditions.

3. Results and Discussion

This section presents the results obtained from component-level testing, system-level scenario testing, notification-timing analysis, the extended stability log, and field testing on a physical coop, followed by a discussion of their significance relative to prior related work.

3.1 Component and System Testing

Component-level testing confirmed that the LCD, RTC, servo motor, relay module, and Wi-Fi/web-dashboard functions operated as intended. The RTC maintained time accuracy with a deviation of less than 1 second and continued operating correctly after system restarts owing to its backup battery. The servo motor moved reliably between its closed and open positions and was triggered precisely at the scheduled feeding times. The relay-controlled lighting switched on and off at the scheduled times as expected. Feed-weight measurement using the load cell and HX711 module achieved an average error of approximately 1.0 %, which is comparable to the 98.97 % accuracy (1.03 % error tolerance) reported for a load-cell-based IoT weighing instrument in prior work [10], supporting the suitability of this sensor combination for feed-monitoring applications. System-level testing across ten operational scenarios achieved a 100 % success rate, as summarised in Table 2.

Table 2. System-level testing scenarios and results

No.	Test Scenario	Result
1	Scheduled feeding at 07:00, 12:00, 17:00, 20:00	Success
2	Manual feeding via Telegram command	Success
3	Mode switching (automatic/manual)	Success
4	Coop lighting ON/OFF on schedule	Success
5	Low-feed warning notification	Success

No.	Test Scenario	Result
6	RTC time accuracy check after power restart	Success
7	Telegram "status" command response	Success
8	ESP32-CAM visual monitoring stream to dashboard	Success
9	Web dashboard data update (weight, time, status)	Success
10	Wi-Fi reconnection after temporary signal loss	Success

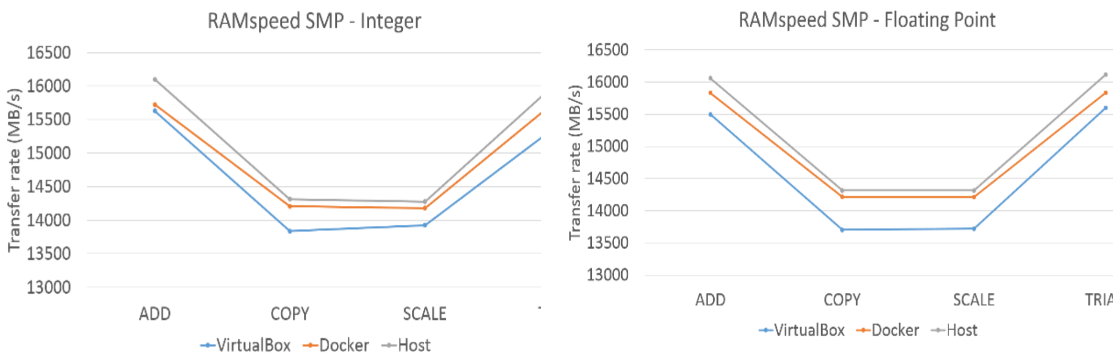
Overall, 10 of 10 test scenarios were completed successfully, giving a system-level success rate of 100 %. These results indicate that the individual subsystems function correctly both in isolation and when operating together within the complete system.

3.2 Notification Timing and System Stability

Notification-timing analysis based on Telegram Bot chat logs is summarised below, with representative examples illustrated in Figure 2. The lighting-OFF notification, for example, carried an RTC-based timestamp of 05:30:01 against a 05:30:00 schedule, a deviation of only about one second, while the scheduled feeding notification matched its 07:00 schedule exactly. The lighting-ON event recorded in the chat log, by contrast, carried a timestamp of 19:17:08 rather than the scheduled 18:30:00. Firmware inspection traced this deviation to the system's boot behaviour rather than to a scheduling fault: the relay output is deterministically forced OFF inside setup on every restart, and the in-memory lighting-state flag is reset accordingly; on the first loop() iteration after a restart, the firmware compares the current RTC time against the 18:30–05:30 lighting window and, if the ON condition is already satisfied, immediately switches the relay and sends the corresponding Telegram notification using the RTC time at that moment rather than the originally scheduled time. The chat log corroborates this mechanism: a "Sistem Pakan Ayam Aktif!" boot notification, which the firmware transmits only once inside setup(), was received within the same minute as the lighting-ON notification, indicating that the ESP32 had just restarted at approximately 19:15 and re-evaluated the lighting schedule immediately afterward, rather than the light having failed to switch on at 18:30 during continuous operation. This restart coincided with the firmware development and testing phase, during which the sketch was re-uploaded to the ESP32 multiple times, each upload causing an automatic reboot; the lighting-ON notification timestamp of 19:17:08 therefore reflects one such reboot rather than a fault occurring during steady-state operation. This behaviour is thus a boot-time re-evaluation artifact tied to the iterative testing process rather than an actual delay in the scheduling logic itself, which the relay-controlled lighting test in Table 2 (Scenario 4) and the field test in Section 3.3 confirmed to switch on and off correctly on schedule once the firmware was finalised. Commands such as "status", mode switching, and manual feeding were consistently answered within the same minute they were sent. Because the ESP32 checks for new Telegram messages every 3 seconds, the theoretical upper bound on response latency is approximately 3-5 seconds, which is consistent with both the recorded chat logs and with the sub-1-second to few-second response times reported for other Telegram-based ESP32 systems in the literature [6].

Table 3. Notification timing accuracy (scheduled vs. actual timestamp)

Notification Event	**Scheduled Time**	**Actual Timestamp (RTC)**	**Deviation**
Scheduled feeding	07:00:00	07:00:00	0 s (exact match)
Coop lighting OFF	05:30:00	05:30:01	~1 s
Coop lighting ON	18:30:00	19:17:08 (after restart)	~47 min (boot re-evaluation artifact; see text)



(a) (b)

Figure 2. Web dashboard (a) and Telegram Bot conversation (b) captured during testing

3.3 Extended Stability and Field Testing

System-stability was evaluated using boot notifications recorded between 29 and 30 June 2026, summarised in Table 4. Because the boot message is only transmitted once during the ESP32 setup routine, its recurrence indicates a restart event. Across the observation window, four boot events were recorded, corresponding to one initial boot and three subsequent manual restarts performed for testing purposes. The longest continuous operating interval between restarts was 18 hours 16 minutes, and the total observation period exceeded 24 hours. Beyond laboratory and prototype-level testing, the complete system was installed on a physical coop and operated continuously for more than three days. Throughout this field test, the feed dispenser, coop lighting, ESP32-CAM monitoring, and Telegram Bot notifications all continued to function without significant issues, confirming that the system performs reliably outside controlled laboratory conditions. Photographs of the installed system are shown in Figure 2.

Table 4. Boot/restart log summary (29–30 June 2026 observation window)

Boot Event	Type	Date	Interval Since Previous Event
1	Initial boot	29 June 2026	— (baseline)
2	Restart	29–30 June 2026	~Beberapa menit setelah Boot 1
3	Restart	29–30 June 2026	~Beberapa menit setelah Boot 2
4	Restart	30 June 2026	18 jam 16 menit setelah Boot 3 (interval terpanjang)

3.4 Comparison with Related Work

Compared with prior single-purpose systems such as automated feeders [1,2], environmental-monitoring systems [4,5], and Telegram-notification systems [6] reported separately in earlier studies, the present work integrates feed scheduling, feed-weight monitoring, lighting control, camera-based monitoring, two-way chatbot interaction, and a web dashboard within a single low-cost ESP32-based platform, and additionally reports field validation on a physical coop rather than laboratory testing in Figure 3 (a-c).

Figure 3. Installed system on a physical coop: (a) front view; (b) side view; (c) rear view

4. Conclusion

An IoT-based automatic chicken feeding system using an ESP32 microcontroller, integrating feed-weight monitoring, scheduled and manual feeding, lighting control, camera-based monitoring, a web dashboard, and a Telegram Bot, has been successfully designed, implemented, and tested. The load cell achieved an average weighing error of approximately

1.0%, the system achieved a 100% success rate across ten operational test scenarios, notifications matched their scheduled RTC times to within about one second, and the Telegram Bot consistently responded to commands within one minute, consistent with its 3-second polling design. The system sustained continuous operation for up to 18 hours 16 minutes between restarts across an observation period exceeding 24 hours, and field testing on a physical coop for more than three days confirmed reliable operation outside laboratory conditions. These results demonstrate that the proposed integrated, low-cost system is a practical solution for automating feeding, monitoring, and remote interaction in small- to medium-scale poultry farms. Future work may extend the system with temperature and humidity sensing, computer-vision-based health monitoring, solar power integration, larger feed-storage capacity with automatic refilling, and deployment at larger farm scales with historical data analytics.

Acknowledgements

The author would like to express sincere gratitude to Prof. Ir. Arridina Susan Silitonga, S.T., M.Eng., Ph.D.,IPM., for her invaluable guidance and supervision as the academic supervisor throughout this research. The author would like to the institution for their guidance and support throughout this research.

CRediT Authorship Contribution Statement

Bintang Risky Deni: Conceptualization, Methodology, Software, Hardware Implementation, Testing, Writing – Original Draft, Writing – Review & Editing. **Afritha Amelia**: Conceptualization, Methodology, Writing-Review & Editing. **Roslina**: System Validation, Writing-Review& Editing. **Mazlina Abdul Majid**: Formal Analysis, Wriring-Review& Editing. **Samsul Bahri:** Writing-Review& Editing

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Goyal V, Yadav A, Mukherjee R. A Literature Review on the Role of Internet of Things, Computer Vision, and Sound Analysis in a Smart Poultry Farm. ACS Agricultural Science & Technology 2024;4:368–88. https://doi.org/10.1021/acsagscitech.3c00467.

[2] Agustina EB, Rachman DA, Nofillah R, Fitri LI. Desain Sistem Smart Feeder Ayam Berbasis Internet of Things (IoT) Guna Mencegah Keterlambatan Pemberian Pakan. Jurnal Fisika Unand 2024;13:297–302. https://doi.org/10.25077/jfu.13.2.297-302.2024.

[3] Setiawan A, Nugroho Y, Prasetyo E. Implementasi sistem pemberian pakan otomatis berbasis mikrokontroler pada peternakan ayam. Jurnal Teknologi Dan Sistem Komputer 2020;8:85–92.

[4] Surahman A, Aditama B, Bakri M. Sistem Pakan Ayam Otomatis Berbasis Internet Of Things. Jurnal Teknologi Dan Sistem Tertanam 2021;02:13–20. https://doi.org/https://doi.org/10.33365/jtst.v2i1.1025.

[5] Heryanto MA, Septian FM, Rahmawati GA, Putra HGS, Wijayanto ND, Pambudi AD. Otomatisasi Pemberian Pakan Ayam Pada Peternakan Ayam Petelur. JTERA (Jurnal Teknologi Rekayasa) 2025;10:33–40. https://doi.org/10.31544/jtera.v10.i1.2025.33-40.

[6] Rohman GA, Isnaini AR. Otomatisasi Pengendalian Suhu dan Kelembaban Berbasis Internet of Things pada Kandang Ayam Potong. MALCOM: Indonesian Journal of Machine Learning and Computer Science 2025;5:558–65. https://doi.org/10.57152/malcom.v5i2.1686.

[7] Turesna G, Andriana A, Abdul Rahman S, Syarip MRN. Perancangan dan Pembuatan Sistem Monitoring Suhu Ayam, Suhu dan Kelembaban Kandang untuk Meningkatkan Produktifitas Ayam Broiler. Jurnal TIARSIE 2020;17:33. https://doi.org/10.32816/tiarsie.v17i1.67.

[8] Tisna DR, Maharani T, Nugroho KT. Pemanfaatan Chatbot Telegram Untuk Monitoring Dan Kontrol Kualitas Air Menggunakan Esp32. JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika) 2024;9:1292–306. https://doi.org/10.29100/jipi.v9i3.5329.

[9] Yulita W, Afriansyah A, Terusan Ryacudu J, Huwi W, Jati Agung K, Lampung Selatan K. Alat Pemantau Keamanan Rumah Berbasis Esp32-Cam. Jurnal Teknologi Dan Sistem Tertanam 2022;03:23–31. https://doi.org/https://doi.org/10.33365/jtst.v3i2.2197.

[10] Supriyanto G, Kumara Jurusan Teknik Pertanian A, Teknologi Pertanian F, Pertanian Stiper Yogyakarta Jl I. Rancang Bangun Timbangan Menggunakan Sensor Load Cell dan Mikrokontroler Berbasis Internet of Things (IoT). Agricultural Engineering Innovation Journal 2024;02:62–8. https://doi.org/10.55180/aei.v2i1.1024.