

Internet-Connected Gamefowl Feed Control System with Weight Sensor

Alimin¹, Kariman Aqla²

^{1,2} Prodi Sistem dan Teknologi Informasi, Institut Teknologi dan Bisnis Yadika Pasuruan, Indonesia

Article Info

Article history:

Received February 15, 2026

Revised March 26, 2026

Accepted April 09, 2026

Keywords:

Android application

Arduino Uno

Cloud database

Microcontroller

Weight measurement

ABSTRACT

This study presents the design and implementation of an Internet of Things (IoT)-based feed control system for gamefowl (fighting chickens) to ensure consistent rations when the breeder is away. The system combines a load cell with an HX711 weight-sensor interface to measure dispensed feed, an Arduino Uno microcontroller to execute control logic, and an ESP8266 Wi-Fi module to exchange data with a cloud service. An Android application allows users to set the target feed mass in grams, start or stop dispensing, and monitor real-time weight feedback. A prototype development method was applied, covering requirement elicitation, hardware and software design, implementation, and functional testing. Performance was evaluated through 20 dispensing trials using a 10 g tolerance limit derived from breeder practice. The prototype achieved successful dosing in all trials, with measured outputs remaining within the defined tolerance and providing immediate feedback to the mobile interface. The results indicate that the proposed system can automate portioning, reduce the risk of overfeeding or underfeeding, and support remote feed management for small-scale poultry operations.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Alimin,

Institut Teknologi dan Bisnis Yadika, Jalan Salem No. 03 Kersikan Bangil, Pasuruan, 67153, Indonesia

alimin@itbyadika.ac.id

1. INTRODUCTION

The gamefowl breeding sector continues to grow in many Asian countries, including Indonesia, where animal performance and health are strongly influenced by daily husbandry practices. Feeding management is particularly critical because breeders often apply controlled rations to maintain stable body weight and conditioning. In small-scale settings, feeding is still commonly performed manually and depends on the breeder's presence; this approach can be inefficient and may lack precision in both timing and feed quantity, especially when the breeder is away [1].

Recent studies have explored Internet of Things (IoT) approaches for poultry monitoring and feeding automation, enabling remote supervision and basic control through networked devices [2]. However, a closer look at existing prototypes shows that many systems emphasize schedule-based feeding (e.g., time/RTC-driven dispensing) and/or open-loop actuation, where the feeder runs for a predefined duration or schedule rather than stopping based on the measured dispensed mass [3]. Even in systems that incorporate load cells, the dosing outcome can still deviate from the intended setpoint due to feed flow characteristics and mechanical inertia when weight feedback is not tightly used as a terminating control signal [4].

In parallel, precision-feeding research highlights the importance of aligning feed allocation with target body-weight trajectories, motivating practical implementations that can translate target masses into reliable, repeatable dispensing at the point of feeding [5]. For gamefowl management, this requirement becomes more specific: breeders need portioning in grams with consistent outcomes and the ability to supervise and intervene remotely.

Based on this context, this work addresses three concrete gaps in the current literature and practice: (1) a domain gap, where published IoT feeder implementations are largely reported for general poultry (broilers/layers/native chickens) and do not explicitly operationalize gamefowl rationing practices; (2) a control gap, where closed-loop, weight-feedback dosing that terminates dispensing at a gram-level target is not consistently implemented; and (3) an interface gap, where the dosing process is not fully integrated with a mobile interface that allows users to set a target mass, start/stop dispensing, and observe real-time weight feedback.

Therefore, this paper proposes a remote, closed-loop, weight-based feed control system tailored to gamefowl rationing. The system integrates a load cell and HX711 interface for mass measurement, an Arduino Uno controller for actuator control, and an ESP8266 module for cloud-connected communication. An Android application enables the user to set the target feed mass (grams), initiate or stop dispensing, and monitor the measured output in near real time. The main contribution is a practical prototype that combines closed-loop weight-based dosing with mobile-integrated remote control, validated through repeated dispensing trials under a breeder-informed tolerance limit.

2. METHOD

2.1. Research Design

This research describes the strategy used to design and build an IoT-based gamefowl feeding control system integrated with an Android application. The main goal is to enable remote feeding control and monitoring through an Internet-connected device. A prototype development model was selected to accelerate development and ensure the system meets user needs through iterative refinement [6] [7].

2.2. Prototype Development Stages

As illustrated in Figure 1, the prototype development model used in this study follows an iterative cycle consisting of Communication, Quick Planning, Quick Modelling, Construction, and Deployment.

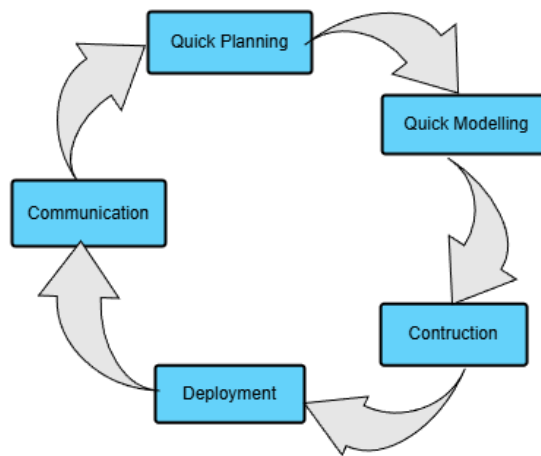


Figure 1. prototype development model

This cycle allows continuous refinement based on user feedback until the prototype meets the breeder's operational needs.

1. **Communication:** Requirement elicitation through interviews and discussions with the breeder to identify feeding routines, target portion ranges, tolerance limits, and practical constraints in the coop environment.
2. **Quick Planning:** Defining the development scope, selecting the main hardware and software components, and preparing a short development plan including test scenarios.
3. **Quick Modelling:** Producing an initial design of the system architecture, circuit layout, and Android interface mock-up as a basis for early validation.
4. **Construction:** Implementing and assembling the prototype, including embedded programming on Arduino and ESP8266, integrating the HX711-load-cell measurement, motor driver, and cloud/mobile communication.

5. Deployment: Installing and running the prototype in the coop environment and conducting functional and dosing trials. Findings from testing are then fed back into the next iteration of the cycle.

2.3. System Requirements

The system requirements consisted of hardware and software components. Hardware included an ESP8266 module, Arduino Uno, HX711 weight-sensor interface, load cell, L298N driver, DC motor, and supporting electronic components [13], [14]. Software tools included Android Studio for Android application development [15], [16], Arduino IDE for embedded programming and testing [17], [18], and Fritzing for circuit visualization [19], [20].

2.4. Research Site and Period

The research was conducted at a breeder's site in Tambaksari, Kraton District, Pasuruan Regency, Indonesia, from November 2024 to March 2025.

2.5. System Architecture and Workflow

The overall system architecture and data/control flow are shown in Figure 2. The system consists of an Android application, a cloud server, an ESP8266 Wi-Fi module, an Arduino Uno controller, an L298N motor driver, a DC motor as the feed dispenser actuator, and a weight measurement unit using a load cell with an HX711 interface.

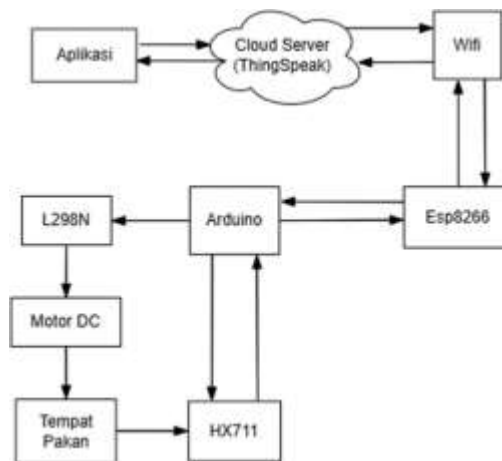


Figure 2. System Block Diagram

As depicted in Figure 2, the workflow begins when the user sets the target feed mass (grams) through the Android application. The target value is sent to the cloud server via the Internet. The ESP8266 retrieves the target value from the cloud and forwards it to the Arduino Uno. Arduino then activates the dispensing mechanism by sending a control signal to the L298N driver, which drives the DC motor to dispense feed from the container.

During dispensing, the feed weight is measured continuously through the load cell–HX711 module. The HX711 output is read by the Arduino and used as feedback to compare the current measured mass with the target setpoint. When the measured mass reaches the target (within the defined tolerance used in testing), Arduino stops the motor to prevent over-dispensing. The measured weight data are also sent back through the ESP8266 to the cloud, allowing the Android application to display near real-time weight updates for remote monitoring and control.

2.6. Data Acquisition and Testing

Data acquisition was conducted from (1) interview results to determine practical feeding limits and constraints, and (2) experimental test records consisting of target feed mass and measured final mass per trial. System performance was evaluated through 20 dispensing trials, and each trial was classified as *successful* if the final dispensed amount met the defined target criteria. Accuracy was calculated as:

$$Accuracy(\%) = \frac{\text{Successful trials}}{\text{Total trials}} \times 100 \quad (1)$$

This testing structure aligns with common reporting practices for prototype validation and system performance evaluation [5]–[7].

3. RESULTS AND DISCUSSION

Recent advances in Internet of Things (IoT) have encouraged the adoption of automation in livestock farming to improve efficiency and feeding accuracy [1], [9], [21]. Manual feeding practices often cause inconsistent portions, which may lead to feed waste and unstable growth. Therefore, this study developed a cloud-connected, weight-based automatic feeder using Arduino Uno and ESP8266 with an HX711 weight sensor interface to measure and control dispensed feed in a measurable manner.

3.1. Breeder Requirement Data

Initial requirements were collected through interviews and discussions with a breeder in Tambaksari, Kraton, Pasuruan. The data include gamefowl type, feed type, typical feeding portions, and an acceptable excess portion. The summary is presented in Table 1.

Table 1. Breeder feeding data

No.	Gamefowl type	Feed type	Typical portion (g)	Allowed excess (g)	Note
1	Mangon	Ground corn, paddy grain	80–150	≤15	Portion depends on body weight
2	Birma	Ground corn, paddy grain	70–130	≤10	Portion depends on body weight
3	Bangkok	Pellets (<i>voer</i>), corn, red rice	100–180	≤15	Portion depends on body weight

Based on Table 1, the smallest allowed excess portion (10 g) was selected as the tolerance reference to form the acceptance range in system testing (e.g., a 100 g target is evaluated against a 100–110 g range).

3.2. Hardware Design

The hardware design of the proposed system is presented in Figure 3. The figure shows the wiring connections among the main modules that form the IoT-based feeder, namely the ESP8266 Wi-Fi module (1) for Internet/cloud connectivity, the Arduino Uno (2) as the primary controller, the HX711 module (3) as the load-cell weight sensor interface, the L298N motor driver (4) for actuator control, and the DC motor (5) as the feed-dispensing actuator.

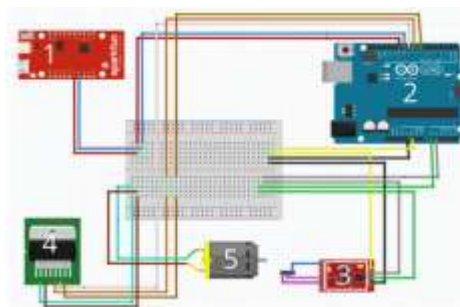


Figure 3. System Device Circuit Design

As illustrated in Figure 3, the hardware interaction works as follows. The ESP8266 (1) retrieves the user-defined target feed mass from the cloud and forwards the target value to the Arduino Uno (2) through a data communication link. The Arduino then activates the dispensing mechanism by sending control signals to the L298N driver (4), which drives the DC motor (5) to dispense feed from the container. During dispensing, the dispensed mass is measured by the load cell, whose signal is amplified and converted by the HX711 (3). The Arduino reads the HX711 output as feedback to monitor the incremental weight and to support weight-based dosing until the target is reached. All modules share a common ground (GND) to ensure stable sensing and actuation performance.

3.3. System Operation and Prototype Deployment

The operational workflow follows the block diagram described earlier and is demonstrated by the assembled prototype shown in Figure 4. As seen in Figure 4, the control electronics (Arduino-based controller, communication module, sensor interface, and motor driver) are placed inside a protective enclosure, with visible wiring connections and indicator LEDs showing that the system is powered and running. The dispensing path is directed toward a feed container/cup positioned below the outlet so that the dispensed feed can be collected and weighed during operation.



Figure 4. Assembled hardware prototype of the feeder system

In operation, the user inputs a target feed mass (grams) through the Android application. The target value is transmitted via the cloud to the device, where the controller activates the motor-driven dispensing mechanism. During dispensing, the load-cell measurement (via HX711) is continuously read to provide weight feedback. The controller uses this feedback to monitor the accumulated dispensed mass and stop the motor when the target is reached (within the defined acceptance tolerance used in testing). The measured weight values are then made available for monitoring through the mobile interface.

To evaluate real-world applicability, the prototype was installed in the breeder's coop environment as shown in Figure 5.



Figure 5. Prototype installed in the coop environment

Figure 5 illustrates the device mounted in the feeding area, allowing the dispensing outlet and the receiving container to remain in a fixed position during dosing. This deployment demonstrates that the system can function as a practical remote-feeding tool, enabling the breeder to control dispensing and observe weight feedback without being physically present at the coop.

3.4 Accuracy Testing Results

System testing was conducted to ensure correct functionality and dosing performance. Two representative test evidences are shown in Figure 6 and Figure 7.

```

-> Received from ESP8266: TARGET:80
-> Same target weight received again. Ignored.
-> Current weight from Load Cell: 72.27
-> Sending current weight to ESP8266: WEIGHT:72
-> Motor is currently running...
-> Current weight from Load Cell: 84.82
-> Motor stopped (target reached).

```

Figure 6. Test Evidence Samples 1

```

-> Received from ESP8266: TARGET:125
-> Same target weight received again. Ignored.
-> Current weight from Load Cell: 99.06
-> Sending current weight to ESP8266: WEIGHT:99
-> Motor is currently running...
-> Current weight from Load Cell: 129.93
-> Motor stopped (target reached).

```

Figure 7. Test Evidence Samples 2

Table 2 reports, for each trial, the target mass (g), the acceptance range based on the breeder-informed tolerance, the feed mixture used, and the final measured dispensed mass.

Table 2. Dispensing test results (20 trials)

No.	Target (g)	Acceptance Range (g)	Feed mixture	Measured (g)	Status
1	100	100-110	P94	106,02	Within range
2	125	125-135	P94	129,93	Within range
3	90	90-100	P94, JG, BM	94,29	Within range
4	120	120-130	P94, JG, BM	126,1	Within range
5	80	80-90	P94, JG	84,82	Within range
6	150	150-160	P94, JG	150,81	Within range
7	130	130-140	P94, JG, Gb	134,3	Within range
8	105	105-115	P94, JG, Gb	114,41	Within range
9	145	145-155	P94, Gb	149,83	Within range
10	95	95-105	P94, Gb	98,02	Within range
11	115	115-125	JG, Gb	115,48	Within range
12	70	70-80	JG, Gb	73,79	Within range
13	160	160-170	JG, BM	162,98	Within range
14	155	155-165	JG, BM	158,67	Within range
15	110	110-120	JG, Gb	116,29	Within range
16	75	75-85	JG, Gb	80,87	Within range
17	130	130-140	P94, BM, Gb	138,63	Within range
18	60	60-70	P94, BM, Gb	64,32	Within range
19	100	100-110	JG	100	Within range
20	115	115-125	JG	115,95	Within range

As shown in Table 2, all 20 trials produced measured outputs that remained within the predefined acceptance ranges (20/20 trials), meaning the system achieved an acceptance-based accuracy of 100% under the 10 g tolerance criterion. Across all trials, the measured values deviated only slightly from the targets; the mean absolute error was approximately 4.28 g, and the maximum deviation was 9.41 g (Trial 8), which still satisfied the acceptance rule. These deviations are consistent with practical dispensing behavior, where variations in feed flow and motor inertia can cause small overshoots. Future improvements may reduce these deviations by applying finer motor control (e.g., shorter actuation intervals or PWM-based speed control) and optimizing the mechanical outlet design for more stable feed flow.

4. CONCLUSION

This study presented a cloud-connected, weight-based feeder that enables remote portion control and monitoring through a mobile interface. The main contribution is the integration of closed-loop weight feedback with remote dosing control to support more consistent rationing practices for gamefowl management. Based on 20 dispensing trials using a breeder-informed 10 g acceptance criterion, the prototype achieved 100% acceptance, indicating that the system can deliver repeatable portions within the defined tolerance and reduce the risk of overfeeding or underfeeding during unattended operation.

Several limitations should be noted. First, the evaluation was conducted in a controlled short-term setting with a limited number of trials, so long-term reliability (e.g., sensor drift, mechanical wear, and environmental effects such as dust and humidity) was not fully assessed. Second, the system was not systematically tested under dynamic or disturbed conditions such as sudden movement/vibration during weighing, changes in feed flow characteristics across a wider range of textures, or dynamic load variations while dispensing. Third, network robustness was not evaluated; performance under Wi-Fi instability, delayed cloud updates, or temporary disconnections remains to be validated.

Future work should address these limitations by conducting extended field trials over longer periods, improving mechanical design and motor control (e.g., finer actuation steps) to further reduce overshoot, adding power backup (battery/solar) for operational continuity, and implementing network-fault handling (offline queuing, local fallback control, and reconnection logic) to maintain safe operation during connectivity disruptions.

ACKNOWLEDGEMENTS

The author would like to thank the gamefowl breeder in Tambaksari, Kraton District, Pasuruan Regency, Indonesia, for providing access, practical feeding information, and support during prototype installation and testing. The author also thanks the colleagues and laboratory staff at Institut Teknologi dan Bisnis (ITB) Yadika Pasuruan for technical assistance and the use of facilities.

REFERENCES

- [1] G. Musyaha, I. Prasetyo, R. K. D. Septiady, and A. Y. Dewi, "Perancangan Dispenser Pakan Ayam Otomatis Memanfaatkan Sistem Pembangkit Listrik Tenaga Surya (PLTS)," *Jurnal Ilmiah Surya Teknik*, vol. 9, no. 2, 2025, doi: 10.48144/suryateknika.v9i2.2169.
- [2] Ade Surahman, Bobi Aditama, Muhammad Bakri, and Rasna Rasna, "Sistem pakan ayam otomatis berbasis Internet of Things," *Jurnal Teknologi dan Sistem Ternam*, vol. 2, no. 1, p. 13, 2021, doi: 10.33365/jtst.v2i1.1025.
- [3] T. Setiadi and T. I. Arifiandi, "Development of an IoT-Enabled Automatic Poultry Feeder and Cage Temperature Monitoring System Using Microcontroller Technology," *TIERS Information Technology Journal*, vol. 5, no. 2, pp. 129–140, 2024, doi: 10.38043/tiers.v5i2.5631.
- [4] T. Kornelius, M. Dahlan, and N. Y. D. Setyaningsih, "Alat Feeding Ayam Otomatis Berbasis Arduino," *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, vol. 2, no. 5, pp. 19–41, 2024, doi: 10.61132/mars.v2i5.335.
- [5] C.-L. Chang and R.-Y. Xu, "Optimizing Feeding Schedule and Live-Weight Prediction for Native Chicken Based on Machine Learning," *Animals (Basel)*, vol. 16, no. 1, p. 75, 2025, doi: 10.3390/ani16010075.
- [6] A. Aziz, Winarno, and T. Haryanti, "Rancang bangun sistem pakan ternak otomatis berbasis Arduino dan load cell," *J. Ilm. Comput. Insight*, vol. 2, no. 1, pp. 1-8, 2020. [Online]. Available: <https://journal.um-surabaya.ac.id/index.php/CI/article/view/5895>.
- [7] Arifin, Jauhari, Leni Natalia Zulita, and Hermawansyah Hermawansyah, "Perancangan muottal otomatis menggunakan mikrokontroler Arduino Mega 2560," *Jurnal Media Infotama*, vol. 12, no. 1, pp. 89-98, 2016, doi: 10.37676/jmi.v12i1.276.
- [8] Fitra Ramadhan, Ido, M. Irwan Bustami, and Willy Riyadi, "Perancangan smart system ternak ayam berbasis IoT menggunakan Arduino UNO," *Jurnal Informatika dan Rekayasa Komputer (JAKAKOM)*, vol. 3, no. 1, pp. 511-521, 2023, doi: 10.33998/jakakom.2023.3.1.814.
- [9] Indra Gunawan, Hamzan Ahmadi, and Muhammad Ramdhani Said, "Rancang bangun sistem monitoring dan pemberi pakan otomatis ayam anakan berbasis Internet of Things (IoT)," *Infotek: Jurnal Informatika dan Teknologi*, vol. 4, no. 2, pp. 151-162, 2021, doi: 10.29408/jit.v4i2.3562.
- [10] Hajarul Ihsan, Ikhsan Ikhsan, and Rini Asmara, "Smart home berbasis Internet of Things dan mobile application pada Pustaka Galeri Mandiri Padang," *Jurnal Pustaka Robot Sister (Jurnal Pusat Akses Kajian Robotika, Sistem Ternam, dan Sistem Terdistribusi)*, vol. 1, no. 1, pp. 6-10, 2023, doi: 10.55382/jurnalpustakarobotsister.v1i1.331.
- [11] Irkham, Iwan Setiawan, and Agung Nugroho, "Perancangan boost converter sebagai suplai inverter menggunakan dsPIC30F4011 dengan metode kontrol proporsional integral," *Transient*, vol. 7, no. 3, pp. 737-744, 2019, doi: 10.14710/transient.7.3.737-744.
- [12] Abdul Jalil Ikromi Radana and Rahmad Zainul Abidin, "Prototype sistem pemberi pakan ternak ayam otomatis menggunakan ESP8266," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 6, pp. 3732-3738, 2024, doi: 10.36040/jati.v7i6.7301.
- [13] Karim Amrulloh, Mohammad Syafa', I Komang Somawirata, and Mohammad Ibrahim Ashari, "Design sistem pengendalian kecepatan dan pengereman pada kursi roda elektrik untuk kondisi jalanan menurun dan menanjak," *Prosiding SENIATI*, vol. 7, no. 2, pp. 243-248, 2023, doi: 10.36040/seniati.v7i2.8019.
- [14] Abdul Haris Mubarak, Moh Afandy, and Muhammad Khaidir, "Rancang bangun sistem kontrol miniatur alat pemindah material pada proses distribusi bijih nikel menggunakan PLC," *Jambura Physics Journal*, vol. 5, no. 1, pp. 1-9, 2023, doi: 10.34312/jpp.v5i1.18466.
- [15] Renny Eka Putri, "Pengembangan sistem pemberi pakan ayam cerdas berbasis Internet of Things (IoT)," 2020, p. 12.

- [16] Program Mahasiswa, Tesar Kornelius, Mohammad Dahlan, Noor Yulita, Dwi Setyaningsih, et al., "Alat feeding ayam otomatis berbasis Arduino," no. 5, 2024.
- [17] Ade Septiansyah Rasiban, Septi Hasanah, Veren Nita Permatasari, and Agistia Yuliawati, "Sistem informasi otomatisasi pelaporan data penjualan Toko Buku Nazwa yang masuk dan yang keluar," *Informatika*, vol. 8, no. 1, pp. 283-284, 2024.
- [18] Dezetty Monika Riandini, Lingga Ghufira, Yani Haryani, Britantyo Wicaksono, and Dian Fig, "Seminar Nasional Teknik Elektro Tahun 2018," pp. 278-281, 2018.
- [19] Ahmad Rufa'i and Zakaria Ansori, "Prototype alat pemberian pakan ayam otomatis menggunakan Arduino dan Internet of Things untuk notifikasi ketersediaan pakan," *Journal of Innovation and Future Technology (IFTECH)*, vol. 4, no. 2, pp. 19-26, 2022, doi: 10.47080/iftech.v4i2.2172.
- [20] Sondang Sibuea, Mohammad Ikhsan Saputro, Agie Annan, and Yohanes Bowo Widodo, "Aplikasi mobile collection berbasis Android pada PT. Suzuki Finance Indonesia," *Jurnal Informatika dan Teknologi Komputer (JITEK)*, vol. 2, no. 1, pp. 31-42, 2022, doi: 10.55606/jitek.v2i1.185.
- [22] Agustinus Zalukhu, Purba Swingly, and Dedi Darma, "Perangkat lunak aplikasi pembelajaran flowchart," *Jurnal Teknologi, Informasi dan Industri*, vol. 4, no. 1, pp. 61-70, 2023.