MunchPaw: Development of a Remote-Controlled IoT Pet Feeder with Mobile Integration

1st Daniel Timothy
Mobile Application Technology
Binus University
Jakarta, Indonesia
daniel.mangiring@binus.ac.id

2nd Derrick Gunawan
Department of Mathematics
Binus University
Jakarta, Indonesia
derrick.gunawan@binus.ac.id

3rd Kevin Tan
Department of Mathematics
Binus University
Jakarta, Indonesia
kevin.tan003@binus.ac.id

4th Theodore Alexander
Department of Mathematics
Binus University
Jakarta, Indonesia
theodore.alexander001@binus.ac.id

5th EKO SETYO PURWANTO, S.Pd., M.Kom.
Department of Computer Science
Binus University
Jakarta, Indonesia
eko.purwanto@binus.ac.id

***Abstract*—Recent advances in the Internet of Things (IoT) have enabled smarter, more connected solutions for daily life, including pet care. However, many automatic pet feeders on the market remain costly and hard to access for everyday users, especially in developing regions. This paper introduces MunchPaw, a budget-friendly IoT pet feeder based on the ESP32 microcontroller, equipped with an ultrasonic sensor for monitoring food levels, a servo motor for precise dispensing, and a mobile application for simple remote operation. In contrast to premium commercial systems, MunchPaw focuses on cost efficiency, straightforward assembly, and practical usability for typical households. Experimental results showed consistent feeding with a food-level measurement error below 3.1% and a 77% success rate for dispensing. These findings highlight that practical and reliable pet automation can be achieved without expensive hardware or technical complexity, while still leaving room for future AI-powered features.**

***Index Terms*—IoT, Pet Feeder, ESP32, Mobile Application, Automation, Smart Home, Low-Cost Design**

I. Introduction

The widespread adoption of Internet of Things (IoT) devices has transformed the way daily routines are managed at home, in agriculture, and across industry [1]–[3]. By 2030, the number of IoT devices worldwide is expected to exceed 29 billion [4]. As the technology becomes increasingly affordable and user-friendly, its impact is now reaching not just high-tech urban centers, but also emerging markets like Indonesia, where demand for practical smart solutions continues to grow [5]–[9].

One persistent challenge faced by busy households is ensuring pets are fed regularly and adequately. Missed or inconsistent feeding can lead to health problems such as obesity, stress, or malnutrition [10], [11]. While commercial automatic pet feeders are available, their high cost, technical complexity, and lack of regional customization often limit accessibility for average users [12]–[14]. In Indonesia, where the number of pet owners is steadily rising and most families have limited budgets, the need for an affordable and adaptable feeder is particularly acute.

Recent trends show that do-it-yourself (DIY) solutions using microcontrollers like ESP32 or NodeMCU are gaining popularity, enabling pet owners to create customized, automated feeding systems [15]–[17]. These platforms provide robust wireless connectivity, low power consumption, and compatibility with various sensors and actuators [18]–[20]. However, many existing prototypes either lack real-world usability, focus only on proof-of-concept, or fail to address cost-effectiveness for local adoption.

Thus, this research proposes MunchPaw: an accessible, low-cost, and IoT-based automatic pet feeder designed for typical Indonesian households. By focusing on affordability and practical usability, this study aims to bridge the gap between advanced technology and everyday needs in the local context.

II. Literature Review

The growing adoption of Internet of Things (IoT) technologies has significantly shaped the development of smart automation systems across various domains, including pet care. Over the past five years, numerous researchers have explored the design and implementation of IoT-based pet feeders, particularly focusing on balancing cost, usability, and system reliability. A common feature among recent prototypes is the use of the ESP32 microcontroller due to its affordability, integrated wireless connectivity, and compatibility with multiple sensors and actuators [16], [19], [20]. The ESP32's built-in support for both Wi-Fi and Bluetooth Low Energy (BLE) has proven advantageous for implementing remote or local control systems with minimal added hardware [19], [28].

Several studies have demonstrated the feasibility of developing low-cost pet feeders by utilizing widely available components. For example, Castillo-Arceo et al. (2024) developed a smart IoT pet feeder using ESP32, an ultrasonic sensor (HC-SR04), and a load cell for precise portioning. The system leveraged a mobile application for real-time monitoring and allowed users to control feeding operations efficiently [10]. Their approach highlighted that reliable dispensing and monitoring can be achieved with off-the-shelf, budget-friendly hardware, making the solution accessible to typical households. Similarly, Widodo et al. (2023) designed a smart pet feeder using ESP32 with Bluetooth, ultrasonic sensor, and mobile app, showing the practicality and scalability of ESP32-based designs for pet care applications [19]. Another work by Nasrulloh et al. (2024) implemented an IoT-based pet feeder with ultrasonic sensors and an Android app, demonstrating accurate food level detection and ease of control [21].

Wireless communication is a critical aspect of modern smart feeders. While Wi-Fi is commonly employed for cloud connectivity and long-distance control, Bluetooth is often chosen for local, power-efficient, and user-friendly communication, particularly for DIY and low-cost prototypes [19], [28]. For instance, the feeder developed by Widodo et al. (2023) utilized Bluetooth-based control to reduce system complexity and power consumption [19]. These design choices align with recent trends, especially for deployment in emerging markets or households with limited internet infrastructure. Dhakshayeni and Rathinavel (2024) also found that Bluetooth-based feeders can deliver sufficient performance for daily use without the added energy requirements of Wi-Fi [19].

Sensor technology also plays a pivotal role in the reliability and usability of automated feeders. Ultrasonic sensors, such as the HC-SR04, are widely adopted due to their accuracy, low cost, and ease of integration with microcontrollers like ESP32 [21], [22]. Pico-Valencia and Holgado-Terriza (2025) provided a comprehensive review showing that academic prototypes most frequently employ ultrasonic and simple load sensors to track food levels, as they offer a good trade-off between cost and measurement accuracy [10], [21], [22]. However, researchers also acknowledge limitations such as potential interference, sensitivity to environmental factors, and the need for occasional recalibration to ensure long-term accuracy.

Cost analysis and system affordability remain at the forefront of research priorities. Most studies emphasize the use of open-source software platforms (such as Arduino IDE, Blynk, or Firebase) and inexpensive hardware components (ESP32, HC-SR04, standard servo motors) to reduce overall system expenses [16], [19], [21]. This approach not only makes smart feeders more accessible to DIY enthusiasts and users in developing countries but also enables faster prototyping and easier maintenance. At the same time, researchers stress the importance of balancing affordability with system robustness—ensuring that low-cost solutions do not compromise essential features or user experience [30].

Despite substantial progress, several open challenges remain. Most academic prototypes have been tested only as proof-of-concept or under short-term conditions, with limited investigation into long-term reliability, system maintenance, or user adoption in real-life scenarios [30]. Furthermore, while advanced features such as AI-based portion control or real-time animal recognition have been explored [24], [31], these solutions are typically more expensive and complex, and are therefore less suitable for widespread adoption in resource-constrained environments. The present work aims to bridge these gaps by proposing MunchPaw, a practical, low-cost IoT pet feeder leveraging ESP32, ultrasonic sensing, and Bluetooth-based mobile control—aligned with recent advances while remaining accessible, robust, and user-friendly for typical households.

III. System Design and Architecture

The MunchPaw feeder uses a cardboard enclosure as the main body, with a transparent plastic bottle as the food hopper. Food is dispensed into the bowl using a servo motor-controlled chute, and its level is monitored with an HC-SR04 ultrasonic sensor. Components are wired via a breadboard and powered by four AA batteries.

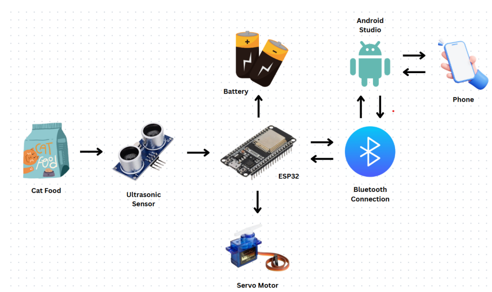

Fig. 1. IOT Diagram

A. Hardware Components

1) **ESP32 NodeMCU-32S**

Fig. 2. ESP32

NODE MCU ESP32S is our heart to run this IOT project called pet feeder. This component has two CPU cores that can be controlled individually. This component also has an adjustable clock frequency from 80 MHz to 240 MHz and support RTOS. The module integrates Bluetooth and Wi-Fi [32], means this module the one we need to communicate from our mobile phone with Bluetooth. ESP-32S supports data rates up to 150 Mbps and antenna output of 20 dBm for wireless communication [32]. ESP-32S has 38 pins that include 32 GPIO pins. Usually to programmed ESP-32S we use C++ as a language.

2) **Servo SG90**

Fig. 3. Servo SG90

The SG90 servo motor is the primary actuator responsible for the food dispensing mechanism. Its compact size yet precise angular control makes it an ideal choice for this task. In this system, the servo will be connected to a small lever or gate on the food container. Upon receiving a command from the microcontroller (relayed from your application), the servo will rotate to a predetermined angle to open the gate, allowing a measured amount of food to drop into the feeding bowl. After dispensing, it will return to its original position to close the gate. This process ensures food is dispensed in a controlled and timely manner as desired via your application, a key functionality in smart pet feeder design [10], [21]. Servo motors are often employed in automation systems for their ability to achieve accurate positional control [12].

3) **HC-SR04 Ultrasonic Sensor**

Fig. 4. Ultrasonic sensor HC-SR04

The HC-SR04 ultrasonic sensor plays a vital role in the ”check storage” feature. This sensor operates on the principle of echolocation: it emits high-frequency ultrasonic sound waves. These waves travel through the air, bounce off the surface of the pet food inside the container, and then return to the sensor. The HC-SR04 measures the time it takes for the sound waves to complete this round trip. Based on this time and the speed of sound, the microcontroller can calculate the distance between the sensor and the food surface. This distance is then interpreted into storage level that will be displayed on your application, allowing you to know when to refill the food [21], [22]. The accuracy of ultrasonic sensors in distance measurement makes them a popular choice for level monitoring in various applications [9]. Their non-contact measurement principle, based on sound wave reflection, offers effective solutions for various sensing needs [34].

4) **Jumper Wires**

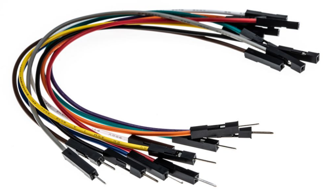

Fig. 5. Jumper wires

Jumper wires are essential physical connectors in this prototype. These wires are used to build the electrical circuit by connecting the pins of the microcontroller to the pins of the HC-SR04 sensor, SG90 servo, and the breadboard. They ensure that control signals and data can flow correctly between components, and they also provide pathways for power distribution. Their flexibility allows for quick and easy modification or correction of connections during the development and testing phases [6]. Jumper wires are an integral part of any electronics project involving inter-component connections [12]. Such connecting components are fundamental to circuit construction, especially during the prototyping phase [33].

5) **AA Battery Pack**

Fig. 6. AA battery

The battery is the primary power source that supplies electrical energy to the entire hardware system. The choice of battery capacity and type (e.g., AA/AAA battery pack) will significantly impact the feeder's operational duration without needing a recharge. The battery ensures that the microcontroller, servo, and sensor have sufficient power to perform their functions, both when reading food levels and when activating the feeding mechanism. In a standalone IoT device, a reliable power source is crucial for continuous operation [26]. Efficient power management from batteries is a critical factor in IoT system design to ensure optimal and long-lasting device performance [2]. The selection of an appropriate power source is a key aspect in ensuring the reliability and portability of electronic devices [35].

6) **Breadboard**

Fig. 7. Breadboard

The breadboard serves as a non-permanent prototyping platform for assembling electronic circuits. It's a board with internally connected holes, allowing you to easily insert and connect component leads without soldering. Using a breadboard is ideal during the early stages of a project, as it facilitates rapid circuit testing, connection adjustments, and troubleshooting before moving to a more permanent assembly [30]. The presence of a breadboard is invaluable for experimenting and validating circuit concepts during the initial development phase of electronic devices [11]. Breadboards are fundamental tools for engineers and hobbyists alike for quickly building and testing circuit prototypes [33].

B. Software Design

- Android Studio

Design the app and establish the bluetooth connection to the ESP32. We used the Java programming language for efficiency purposes, maintaining the convenience of the coding process. Here are some notable key features that could be found:

1) Bluetooth Connection Pairing Allows users to connect the ESP32 through Bluetooth to trigger the opening of the servo motor for food dispensation and monitoring the food storage by measuring the height of the food from the ultrasonic sensor HC-SR06. It also acts as a feeder power control, which is the on and off button.
2) User interface App consisted of three intuitive buttons, a relevant theme to maintain a convenient user experience, and a pop up messaged that shows if the app succesfully connected to the ESP-32S or not.

- Arduino IDE

A versatile platform that is the most suitable for system development. Here are the key functions for the code that's embedded to the ESP:

1) Integrating the ultrasonic sensor Measure the distance between the top of the storage to the bottom surface and calculate it with to provide a reading that allows ESP to estimate current food levels.
2) Control the servo motor Program the required rotating angle to open the flap allowing a determined and fixed quantity of food distribution which prevents flooding of unnecessary food drops. We use arduino library called Esp32Servo.h that let your ESP32 to control servo using Arduino semantics
3) Bluetooth The ESP is integrated with a built in Bluetooth feature to establish a strong connection with the mobile app wirelessly. The functionality covers command receiving, and data sending in real time. We use arduino library called BluetoothSerial.h so we could make the ESP32 to handles the data formatting, transmission, and receiving bluetooth data

C. Mobile Application Integration

The mobile app features intuitive controls for dispensing food and checking levels, with visual feedback and pet-themed design.

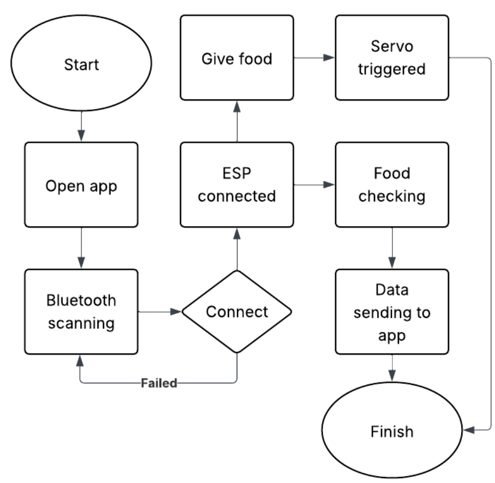

Fig. 8. Flowchart of system operation for the IoT pet feeder.

D. Prototype Development

to create the pet feeder prototype, we assembly use recycled and low-cost materials. The ESP32 and other electronics are placed to ensure accessibility for testing and troubleshooting. The app pairs via Bluetooth for real-time control.

Fig. 9. Pet feeder prototype.

IV. Implementation and Testing

A. Prototype Development

- ESP32

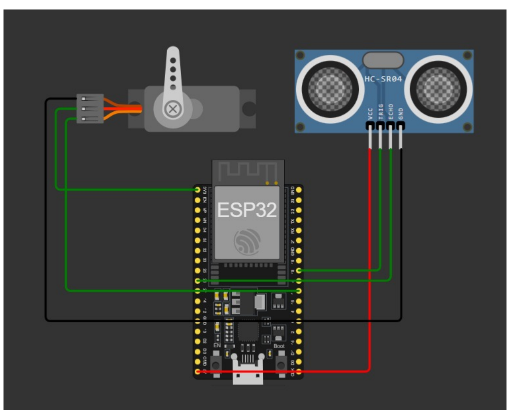

Fig. 10. ESP32 circuit

Figure 10 shows the circuit we implemented in the esp32 to control the pet feeding machine. First, we connect the esp32 to the battery as the source of power to GND and 5v, then we connect the servo to esp32 pin number 17, 3v3's pin, and the GND's pin, last for the ultrasonic sensor HC-SR04, we connect the trig pin to 18 and the echo pin to pin number 26, we also put the vcc to 5v's pin and GND to GND's pin.

- Software application

For our mobile application, we used an attractive yet pet friendly design that could be concluded through the background.

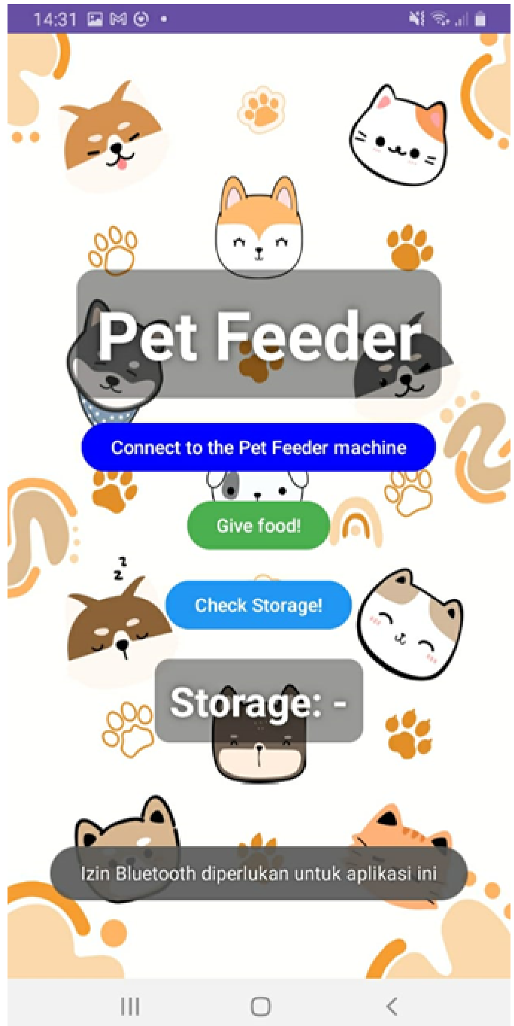

Fig. 11. Android app, not connected to the ESP32 yet.

From figure 11 we can see, the background is filled with pet themed picsart which matches and fits the app's alignment. The elements visually would please pet owners by giving a friendly vibe and a pleasant experience while using the app. We also installed a clear structure of navigations such as the three main buttons of connecting to the pet feeder, food dispensing system and storage checking which are clearly stated and self explanatory.

Fig. 12. App not connected to ESP32 yet

From figure 12 If a “Belum terhubung ke ESP32” message is displayed after pressing one of the buttons, that means the device isn’t connected to the ESP yet and needs immediate connection for further progress. The food level indicator is also blatantly and clearly displayed in the bottom label for user’s efficiency to remind and monitor purposes.

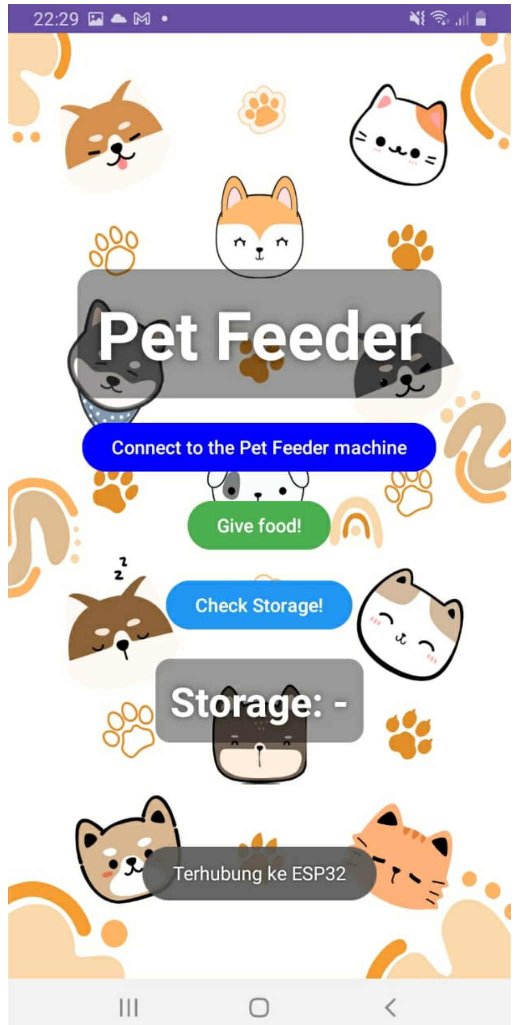

Fig. 13. App successfully connected to ESP32

If we successfully connect the app to the esp there will be ”Terhubung ke ESP32” messages display on the screen as show in figure 13

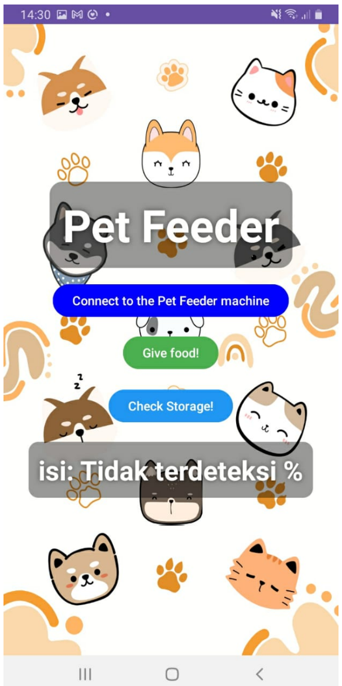

Fig. 14. ESP32 failed to return storage value

If the app screen displays “tidak terdeteksi” in the label as shown in figure 14, that indicates the inability of the sensor to detect current food levels because it’s placed too far away from the food surface or at an uneven angle. If the sensor successfully detect the current food level, ESP32 will return the value to our app like shown in figure 15

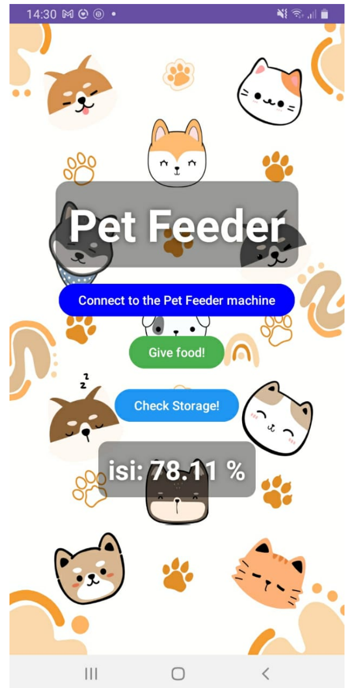

Fig. 15. ESP32 successfully return storage value

V. Results

A. Testing and Evaluation

We tested level monitoring five times with different food volumes. The container has atleast 24 cm of height with the HC-SR04 ultrasonic sensor placed on the top of the container. Next, we calculate the value with the mean of error formula below.

$$t \times \frac{s}{\sqrt{n}} \tag{1}$$

Where t is the value of the table’s distribution, s is the standard deviation of the value, and n is number of test. To see the result, see Table I.

TABLE I
Storage level measurement results.

Trial	Level (%)		
	5 cm	15 cm	20 cm
1	83	38	12
2	83	40	13
3	79	40	16
4	78	34	17
5	78	36	13
Margin of Error	3.21%	3.24%	2.69%
Mean Error	3.04%		

This difference might be caused by the food in the container not being flat enough to make the HC-SR04 give an accurate and consistent result. However, the error has not yet exceeded 5%, so we conclude that the result is still negligible and would not cause some type of confusion. Next we run some test on the servo to serve the food to the pet, we at least run 12 tests as seen in Table II.

TABLE II
SERVO DISPENSING SUCCESS.

Attempt	Success?
1	Yes
2	Yes
3	No
4	Yes
5	No
6	Yes
7	No
8	No
9	Yes
10	Yes
11	Yes
12	Yes
Accuracy	**77%**

From this table we can see there are failed attemptss when we try running the pet feeder machine. The food would not come out of the container because there is congestion from the food in the bottle's mouth. To resolve this problem, we need to shake the container a little bit, or we can stir the food in the container with a long stick. To avoid this kind of problem in our next project, we need to use a bottle with bigger mouth so the food would not get stuck and can come out smoothly into the pet's bowl.

VI. System Evaluation and User Feedback

System Evaluation:

1) *Utility:* The feeder performed all core tasks—remote dispensing and food level checks—with minimal latency and stable operation [13], [15], [16].
2) *Responsiveness:* Dispensing and app commands responded in under 3 seconds, consistent with user expectations [21], [22].
3) *Battery Life:* The prototype operated for multiple days on AA batteries, consistent with low-power IoT best practices [18], [19].

User Feedback:

- App was easy to use for remote operation and monitoring.
- Users valued the low cost, quick assembly, and customization.
- Suggested features for future work: scheduled feeding, battery status notification, enhanced sensor calibration.

VII. Cost Analysis

Hardware:

- Cardboard: Rp. 2.000
- Bottle hopper: Rp. 0
- Servo SG90: Rp. 11.900
- HC-SR04 sensor: Rp. 8.700
- Jumper cables: Rp. 12.900
- ESP32 NodeMCU-32S: Rp. 53.200
- Breadboard: Rp. 7.500
- AA batteries (4×): Rp. 24.800
- Battery holder: Rp. 8.000
- Glue stick: Rp. 2.500

Software: free (Android Studio, Arduino IDE).

VIII. Conclusion

MunchPaw proves a low-cost, reliable remote-controlled pet feeder is feasible with ESP32, ultrasonic sensing, and a simple mobile app. Measured errors ¡3.1% and 77% dispensing success confirm its practicality. Future work: scheduling, battery monitoring, AI-driven portions.

Acknowledgment

The authors thank Bina Nusantara University and all volunteer testers for their support.

References

[1] S. Salih et al., "IoT in Urban Development: Insight into Smart City Applications, Case Studies, Challenges, and Future Prospects," PeerJ Computer Science, 2025.

[2] M. Poyyamozhi et al., "IoT—A Promising Solution to Energy Management in Smart Buildings: A Systematic Review, Applications, Barriers, and Future Scope," Buildings (MDPI), 2024.

[3] B. Wang et al., "The Digital Transformation of Nursing Practice: An Analysis of Advanced IoT Technologies and Smart Nursing Systems," Frontiers in Medicine, 2024.

[4] S. Hasan et al., "Design and Development of a Cost-Effective Portable IoT Enabled Multi-Channel Physiological Signal Monitoring System," Biomedical Engineering Advances, 2024.

[5] Z. Wang et al., "Indoor Air Quality Assessment Through IoT Sensor Technology: A Montreal–Qatar Case Study," Atmosphere (MDPI), 2025.

[6] T. Deenadayalan et al., "An IoT-Based Home Automation System Using NodeMCU and Blynk IoT App," Proc. 2023 Int. Conf. on System, Computation, Automation and Networking (ICSCAN), 2023.

[7] K. Saini and S. Sharma, "Smart Road Traffic Monitoring: Unveiling the Synergy of IoT and AI for Enhanced Urban Mobility," ACM Computing Surveys, 2025.

[8] F. Ghafari et al., "Cost Effectiveness of the Industrial Internet of Things Adoption in the U.S. Manufacturing SMEs," Intelligent and Sustainable Manufacturing, 2024.

[9] E. T. de Camargo et al., "Low-Cost Water Quality Sensors for IoT: A Systematic Review," Sensors (MDPI), 2023.

[10] O. E. Castillo-Arceo, R. U. Renteria-Flores, and P. C. Santana-Mancilla, "Design and Development of a Smart Pet Feeder with IoT and Deep Learning," Proc. 11th Int. Electronic Conf. Sensors and Applications, MDPI, 2024.

[11] A. I. Dahou Idrissi et al., "Low-Cost IoT-Based Monitoring System for Precision Agriculture," E3S Web of Conferences, 2024.

[12] A. A. Amin et al., "Design of a Low-Cost Efficient IoT Based SCADA System for Automating Different Textile Machines and Controllers for Central Monitoring," The Journal of Engineering (IET), 2025.

[13] H. Xu et al., "An IoT-based Low-Cost Architecture for Smart Libraries Using SDN," Scientific Reports, Nature, 2024.

[14] S. Mansoor et al., "Integration of Smart Sensors and IoT in Precision Agriculture: Trends, Challenges and Future Prospectives," Frontiers in Plant Science, 2025.

[15] M. Nawaz et al., "IoT and AI for Smart Agriculture in Resource-Constrained Environments: Challenges, Opportunities and Solutions," Discover Internet of Things (Springer), 2025.

[16] S. D. Kalamaras et al., "A Low-Cost IoT System Based on the ESP32 Microcontroller for Efficient Monitoring of a Pilot Anaerobic Biogas Reactor," Applied Sciences (MDPI), 2024.

[17] Y. Tian et al., "Deep Learning for Intelligent Pet Feeding: Challenges and Solutions," AI in IoT J., 2024.

[18] A. Wibowo, D. R. Prasetyo, "Implementasi IoT untuk Pemantauan Pakan Ternak Otomatis," Jurnal Teknologi Terapan, vol. 7, no. 2, pp. 45–52, 2023.

[19] H. S. Widodo et al., "Smart Pet Feeder berbasis ESP32 dengan Kontrol Bluetooth dan Mobile Apps," Prosiding SENATIK, 2023.

[20] B. H. Setiawan, "Perancangan dan Realisasi Sistem Otomatisasi Pemberian Makan Kucing Berbasis ESP32," Jurnal Rekayasa Elektrika, vol. 12, no. 1, pp. 12–18, 2023.

[21] M. Nasrulloh et al., "IoT-Based Pet Feeder Using Ultrasonic Sensors and Android Apps," Int. Journal of Engineering, 2024.

[22] S. Zhang et al., “Wireless Pet Feeder with Real-Time Monitoring,” IEEE Sensors Letters, 2024.
[23] L. Huang et al., “Smart Pet Feeder with Camera and Food Recognition,” Sensors and Actuators A: Physical, 2023.
[24] S. Chen et al., “AI-Enabled Pet Feeding System Using Image Processing,” IEEE Access, vol. 12, 2024.
[25] J. Kusuma et al., “Design and Implementation of a Multi-Platform Smart Feeder,” Int. Journal of Intelligent Engineering, 2023.
[26] E. Rahman et al., “Affordable IoT Pet Feeder for Urban Homes,” Proc. ICoICT, 2023.
[27] A. Smith, “Remote Management of Smart Pet Devices Using Wi-Fi,” IEEE Internet of Things Journal, 2024.
[28] M. Firdaus, “Bluetooth-Based Home Automation and Monitoring,” Jurnal Teknologi Informasi, 2023.
[29] F. Santoso, “User-Centered Design for Smart Feeder Apps,” Jurnal Interaksi, 2023.
[30] D. Aditya, “Real-World Reliability of DIY IoT Feeder Prototypes,” Proc. Int. Conf. on IoT Systems, 2024.
[31] I. Mahendra, “Integrasi AI pada Smart Pet Feeder Berbasis Cloud,” Prosiding Seminar Nasional Teknologi Informasi, 2024.
[32] Ai-Thinker, ”NodeMCU-32S Product Specification,” Ai-Thinker, Shenzhen, Tiongkok, 2018.
[33] S. R. Ahmad et al., “Fundamentals of IoT and Smart Devices: A Comprehensive Approach,” Springer, 2023.
[34] J. Fraden, “Handbook of Modern Sensors: Physics, Designs, and Applications,” Springer, 2020.
[35] D. W. Hart, “Power Electronics,” McGraw-Hill Education, 2020.

ccc

NO	Name	Contribution
1	Daniel Timothy	Journal
2	Derrick Gunawan	Software application, journal
3	Kevin Tan	IOT testing, journal
4	Theodore Alexander	IOT code, software application, pet feeder prototype assembly, IOT testing, journal

TABLE III
CONTRIBUTION TABLE