Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 23, No. 2 September 2026 | DOI https://doi.org/10.33480/techno.v23i2.8613

# INTERNET OF THINGS-BASED ANIMAL MONITORING AND NOTIFICATION SYSTEM USING TELEGRAM BOT AND CLOUD SERVER

**M. Afif Rizaldi [1*]; Farida Idifitriani[2]**

Informatics Study Program[1,2]
Faculty of Systems Engineering[1,2]
Sumbawa University of Technology, Sumbawa, Indonesia[1,2]
farida.idifitriani@uts.ac.id

(*) Corresponding Author



The creation is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License.

***Abstract***— *Damage to agricultural land caused by livestock disturbance remains a common problem in agricultural areas in Sumbawa Regency. Manual monitoring makes it difficult for farmers to supervise remote agricultural areas continuously and in real time. This study implements an Internet of Things-based animal monitoring and notification system using ESP32-CAM, a Passive Infrared sensor, Telegram Bot, and a Virtual Private Server (VPS). When motion is detected, the ESP32-CAM captures an image and transmits it to the VPS through HTTP. The VPS performs object detection using a YOLOv8 model and sends the detection result and captured image to the user through Telegram Bot. The system also activates a local alarm through a relay so that the warning mechanism can operate when Internet communication is unavailable. The research stages include problem identification, literature review, system design, hardware and software implementation, testing, and analysis. Experimental results show that image transmission, server-side YOLOv8 inference, Telegram notification, and local alarm functions operate successfully. The measured end-to-end notification delay is approximately 5–10 seconds, depending on network conditions. The proposed architecture demonstrates the feasibility of combining edge image acquisition, cloud-based inference, and remote notification for agricultural field monitoring.*

***Keywords***: *Cloud Server, ESP32-CAM, Internet of Things, Telegram Bot, YOLOv8.*

**Intisari**—Kerusakan lahan pertanian akibat gangguan hewan ternak masih menjadi permasalahan yang sering terjadi pada area pertanian di Kabupaten Sumbawa. Proses monitoring lahan yang masih dilakukan secara manual menyebabkan petani mengalami kesulitan dalam melakukan pengawasan secara real-time, terutama pada area pertanian yang berada jauh dari permukiman. Penelitian ini bertujuan untuk mengimplementasikan sistem monitoring dan notifikasi deteksi hewan berbasis Internet of Things (IoT) menggunakan ESP32-CAM, Telegram Bot, dan cloud server berbasis Virtual Private Server (VPS). Sistem dirancang untuk melakukan monitoring area pertanian secara otomatis melalui sensor PIR dan ESP32-CAM. Ketika terdeteksi gerakan, ESP32-CAM akan mengambil gambar dan mengirimkannya ke cloud server menggunakan protokol HTTP. Selanjutnya server melakukan proses verifikasi objek menggunakan model YOLOv8 dan mengirimkan hasil monitoring ke Telegram Bot secara real-time. Tahapan penelitian diawali dengan proses identifikasi permasalahan, dilanjutkan dengan kajian pustaka, perancangan solusi, pengembangan sistem, pelaksanaan pengujian, serta evaluasi terhadap hasil yang diperoleh. Hasil penelitian menunjukkan bahwa sistem berhasil melakukan monitoring dan pengiriman notifikasi Telegram secara real-time dengan delay rata-rata sekitar 5–10 detik tergantung kualitas jaringan internet. Selain itu, cloud server mampu menjalankan proses verifikasi objek menggunakan YOLOv8 dengan baik sehingga mendukung sistem monitoring area pertanian berbasis IoT secara online.

***Kata Kunci***: *Server Cloud, ESP32-CAM, Internet of Things (IoT), Bot Telegram, YOLOv8.*

## INTRODUCTION

The agricultural sector continues to play a vital role in supporting rural economic development and ensuring national food security (Widyastutik et al., 2025). However, the sustainability of agricultural production largely

depends on farmers' ability to safeguard farmland from various threats, particularly damage caused by livestock and agricultural pests (Forqan, Muslim, & Aulia, 2024).

Sumbawa Regency is one of the regions with a substantial agricultural and livestock sector. Most of the local population relies on agricultural land as their primary source of livelihood, particularly through corn and rice cultivation (Sari et al., 2023). However, the free-range livestock farming system remains one of the major factors contributing to crop damage. Livestock, particularly cattle, frequently enter agricultural fields and damage cultivated crops, resulting in economic losses for farmers. Similar conditions have also been reported in several areas of Sumbawa Regency, where conflicts between farmers and livestock owners often arise due to crop damage caused by free-ranging livestock entering cultivated agricultural land (Rencani et al., 2024).

The main problem addressed in this study is the difficulty of continuously monitoring agricultural fields located far from residential areas and identifying whether detected movement is caused by livestock or other objects. Conventional monitoring methods require farmers to visit or patrol the field directly, which is inefficient for continuous surveillance. A PIR sensor can detect movement but cannot determine the type of object responsible for the movement. Therefore, a monitoring system is required that can combine motion detection, image acquisition, object verification, remote notification, and local warning mechanisms. Based on this problem, this study proposes an IoT-based monitoring system integrating PIR, ESP32-CAM, YOLOv8, a VPS-based cloud server, Telegram Bot, and a relay-controlled alarm.

This problem becomes increasingly difficult to manage because most agricultural land is located in areas far from residential settlements. Consequently, farmers are unable to continuously monitor their fields. Conventional security practices, including night patrols, simple fencing, and direct field observation, are considered ineffective in providing adequate protection against animal disturbances in agricultural areas (Limbong & Harsanto, 2025).

Recent advances in Internet of Things (IoT) technology have opened new opportunities for developing automated monitoring systems capable of supporting remote surveillance of agricultural areas. IoT enables interconnected devices to communicate and exchange data over the Internet, allowing real-time monitoring and information transmission (Rofianto et al., 2026). In agriculture, IoT has been widely adopted for environmental monitoring, farmland security, and the supervision of agricultural activities, thereby improving operational efficiency and supporting precision farming practices (Gyamfi et al., 2024).

One of the IoT devices widely used in monitoring systems is the ESP32-CAM, as it integrates a camera module and Wi-Fi connectivity into a single compact device. The ESP32-CAM can capture images and transmit data to a server via the Internet in real time (Humam & Triawan, 2024). According to (Arrahma and Mukhaiyar 2023), the ESP32-CAM is an effective platform for IoT-based monitoring applications because it combines image acquisition capabilities and wireless network connectivity within a single module. Furthermore, the use of a cloud server enables centralized data processing, allowing monitoring systems to be accessed remotely from different locations (Nurachman et al., 2026). In addition, cloud servers function as data storage platforms and communication intermediaries between IoT devices and monitoring applications. (Arun et al. 2025) stated that cloud-based communication enables centralized data exchange and computational processing, thereby improving the performance, scalability, and reliability of IoT-based monitoring systems.

Telegram Bot has emerged as an effective monitoring platform for Internet of Things (IoT) applications because it provides rapid notification services and can be easily integrated using the Telegram Bot API. Through Telegram Bot, users are able to receive monitoring results, including text messages and captured images, directly on their smartphones in real time (Tisna et al., 2024). In addition, Telegram supports remote device control through predefined commands, allowing users to perform system testing and monitor IoT devices efficiently from any location. Rosita et al. (2022) reported that Telegram Bot is an effective monitoring platform as it automatically delivers information to users via the Internet, thereby enhancing the responsiveness and reliability of IoT-based monitoring systems.

Recent advances in deep learning-based object detection have led to the development of various algorithms capable of detecting objects in real time. Among these, YOLOv8 has become one of the most widely adopted object detection algorithms due to its high inference speed and superior detection accuracy compared with earlier YOLO versions (Ramdan & Asriyanik, 2024). Consequently, YOLOv8 has been extensively applied in computer vision-based monitoring systems to enable automatic object recognition and real-time decision-making. According to (Wijayanti et al. 2025), YOLOv8 achieves high detection accuracy while maintaining real-time performance, making it

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 23, No. 2 September 2026 | DOI https://doi.org/10.33480/techno.v23i2.8613

highly suitable for intelligent monitoring applications.

To address these challenges, this study develops an Internet of Things (IoT)-based animal detection and monitoring system that integrates an ESP32-CAM, a Telegram Bot, and a Virtual Private Server (VPS)-based cloud server. The proposed system automatically monitors agricultural fields by capturing images whenever motion is detected and transmitting them to the cloud server, where object verification is performed using the YOLOv8 algorithm (Kalesaran, 2024). The detection results are then sent to users via Telegram in real time, allowing farmers to remotely monitor the condition of their agricultural fields more effectively and respond promptly to potential threats.

## MATERIALS AND METHODS

a. ESP32-CAM

The ESP32-CAM is an ESP32-based microcontroller module that integrates an OV2640 camera and built-in Wi-Fi connectivity into a compact platform. In this study, it functions as the primary hardware device for capturing images and transmitting data to a cloud server over the Internet. A key advantage of the ESP32-CAM is its ability to combine image acquisition and wireless data communication within a single low-power device, making it highly suitable for real-time Internet of Things (IoT)-based monitoring applications (Arrahma & Mukhaiyar, 2023). The ESP32-CAM used in this study is presented in Figure 1.




Source: (Research Result, 2026)
Figure 1. ESP32-CAM

b. Telegram Bot

Telegram Bot serves as a real-time monitoring and notification platform within the proposed IoT system. By leveraging the Telegram Bot API, the server can automatically transmit text messages, captured images, and object detection results to users. This functionality enables users to remotely monitor system activities through their smartphones while receiving timely notifications whenever an object is detected. The Telegram Bot monitoring interface used in this study is presented in Figure 2.

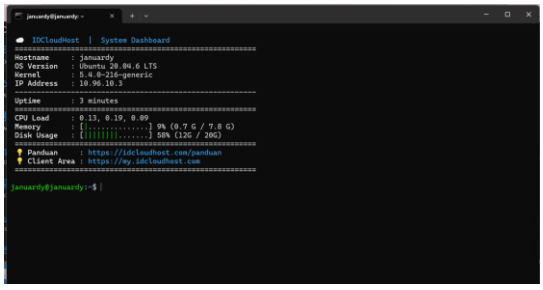



Source: (Research Result, 2026)
Figure 2. Telegram bot monitoring

Telegram Bot provides a Bot API that facilitates real-time automated communication between the server and end users, making it an ideal platform for Internet of Things (IoT)-based monitoring applications. Through the Telegram Bot API, monitoring information and system notifications can be delivered automatically without requiring direct user interaction. (Rosita et al. 2022) and (Kurniawan et al. 2025) reported that Telegram Bot effectively supports remote monitoring by automatically transmitting messages and notifications to users over the Internet, thereby improving the responsiveness and accessibility of IoT-based monitoring systems.

c. Virtual Private Server

A Virtual Private Server (VPS) is utilized as the cloud computing platform to receive image data transmitted from the ESP32-CAM and perform object detection using the YOLOv8 model. The server environment is built on the Ubuntu Linux operating system and implemented using the Flask web framework, which manages communication between IoT devices and the server while executing the object detection inference process. The architecture of the VPS-based cloud server is presented in Figure 3.

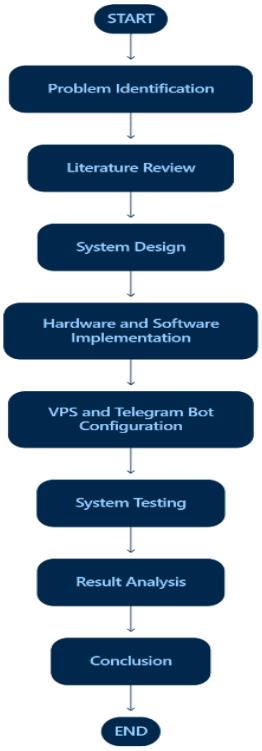



Source: (Research Result, 2026)
Figure 3. Cloud Server Architecture

The use of a Virtual Private Server (VPS) enables computational processes to be centralized, thereby improving the efficiency, scalability, and reliability of the monitoring system while supporting cloud-based object detection. According to (Arun et al. 2025), cloud server utilization provides a more efficient data processing environment than executing computational tasks directly on resource-constrained IoT devices. By off loading computational workloads to the cloud, IoT devices can operate with lower resource consumption while maintaining real-time system performance.

d. Research Flowchart

The research was conducted through a systematic sequence of stages to ensure that the proposed monitoring system was developed, implemented, and evaluated in a structured manner. The process began with problem identification and a comprehensive literature review to analyze existing approaches and determine the research objectives.

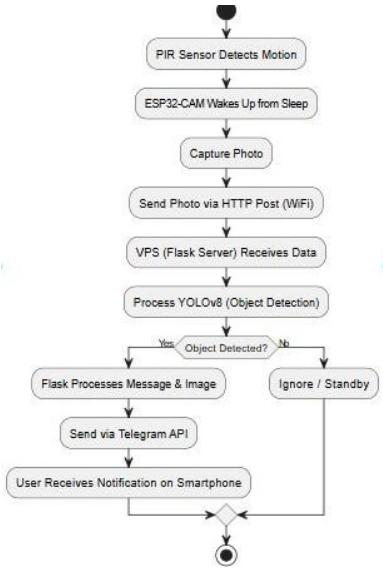



Source: (Research Result, 2026)
Figure 4. Research Flowchart

Subsequently, the system architecture was designed, followed by the implementation of the ESP32-CAM, VPS-based cloud server, Telegram Bot, and the YOLOv8 object detection model. After the integration of both hardware and software components, the proposed system was deployed and evaluated through a series of functional and performance tests, including object detection accuracy, notification delivery, and system response time. Finally, the experimental results were analyzed to assess the effectiveness of the proposed system and to formulate the conclusions of this study. The overall research procedure is illustrated in Figure 4.

e. Monitoring System Design

The system design aimed to develop an Internet of Things (IoT)-based monitoring and notification system capable of delivering real-time information regarding the condition of agricultural fields. The proposed architecture enables automated data acquisition, processing, and transmission, thereby supporting continuous remote monitoring and timely notifications for users. The system architecture is presented in Figure 5.




Source: (Research Result, 2026)
Figure 5. Monitoring System Design

Within the proposed monitoring system, a Passive Infrared (PIR) sensor is employed to detect motion events in the monitored area. Upon detecting movement, the ESP32-CAM automatically captures an image and uploads it to the VPS-based cloud server. The server then performs object detection using the YOLOv8 model, and the resulting information is transmitted to users via Telegram for real-time monitoring and notification purposes.

f. System Implementation

The implementation phase focused on integrating all hardware and software components into a unified IoT-based monitoring

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 23, No. 2 September 2026 | DOI https://doi.org/10.33480/techno.v23i2.8613

system. The integrated system operates over the Internet, enabling continuous communication between devices, real-time data processing, and automatic notification delivery to end users. The implemented monitoring system is presented in Figure 6.



Source: (Research Result, 2026)
Figure 6. Implementation of Monitoring System

The main hardware and software components used in the proposed system are summarized in Table 1.

Table 1. System Device Specifications

| No | Component | Specification |
|---|---|---|
| 1 | ESP32-CAM | OV2640 + WiFi |
| 2 | Sensor PIR | HC-SR502 |
| 3 | VPS Server | Ubuntu Linux |
| 4 | Framework | Flask |
| 5 | Telegram Bot | Bot API |
| 6 | Model AI | YOLOv8 |
| 7 | Relay | 1 Channel 5V |
| 8 | Alarm | Sirine 12V |
| 9 | Solar Panel | 20 Wp |
| 10 | Battery | 12 V, 3.5 Ah |
| 11 | Solar Charge Controller | PWM, 12 V-10 A |

Source: (Research Result, 2026)

The monitoring system was designed for deployment in agricultural areas with limited access to conventional electrical power. Therefore, the prototype uses a 20 Wp solar panel connected to a PWM 12 V–10 A solar charge controller and a 12 V 3.5 Ah battery. The nominal energy capacity of the battery is approximately 42 Wh, calculated from the battery voltage and rated capacity. This energy value represents the nominal storage capacity before considering conversion losses, battery discharge limits, charging efficiency, and variations in environmental conditions.

g. YOLOv8 Dataset and Training

The YOLOv8 model was trained using a dataset consisting of 438 images collected directly from the agricultural environment. The images were annotated using Roboflow, with the documented target object class being "cow" or cattle. The dataset was used to train the object detection model for identifying livestock entering the monitored agricultural area. The training process used 300 epochs. The available training record reports a precision of 93.58%, recall of 94.23%, and mAP@0.50 of 97.36%.

The YOLOv8 training process was conducted using the documented training workflow managed through Roboflow. After training, the resulting model weights were used for server-side inference on the VPS. The ESP32-CAM was not used to perform YOLOv8 inference; it functioned as the image acquisition and transmission device. The available research record does not document a separate workstation-based training procedure, so no separate workstation is claimed in this study.

The available training record confirms the use of training, validation, and testing subsets. However, the exact percentage or number of images assigned to each subset was not documented in the available research record. Therefore, no specific split ratio is reported in this study to avoid introducing unsupported information. The main YOLOv8 training information is summarized in Table 2.

Table 2. YOLOv8 Training Results

| Metric | Result |
|---|---|
| Dataset images | 438 |
| Target class | Cow |
| Training epochs | 300 |
| Precision | 93.58% |
| Recall | 94.23% |
| mAP@0.50 | 97.36% |

Source: (Research Result, 2026)

## RESULTS AND DISCUSSION

1. Implementation of Monitoring System

The proposed monitoring system was deployed in an agricultural field by positioning the ESP32-CAM and Passive Infrared (PIR) sensor at the designated surveillance location. The implemented system successfully established communication with the VPS-based cloud server and Telegram Bot, enabling real-time image transmission, object detection, and automated notification delivery to end users.

The solar power subsystem supports autonomous operation of the monitoring prototype in agricultural areas where grid electricity may not be readily available. The system uses a 20 Wp solar panel, a PWM 12 V–10 A charge controller, and a 12 V 3.5 Ah battery. The nominal battery energy is approximately 42 Wh. Field measurements in the supporting implementation study recorded approximately 14.1 V at the battery terminal in the morning, 19.7 V at the solar-panel terminal around midday, and 12.8 V at the battery terminal in the afternoon. However, component-level current consumption measurements for the ESP32-CAM, PIR sensor, relay, and alarm were not documented. Therefore, the exact battery operating duration and daily energy efficiency cannot be calculated reliably from the available measurements. Longer-duration field testing with component-level current measurements is required to determine battery autonomy and overall energy efficiency. The implementation of the monitoring system in the agricultural field is presented in Figure 7.



Source: (Research Result, 2026)

Figure 7. System Implementation in Agricultural Areas

2. Telegram Monitoring Results

System testing was performed to evaluate the ability of the proposed monitoring system to deliver real-time notifications. The evaluation assessed whether the system could reliably transmit object detection results from the VPS-based cloud server to users via Telegram with minimal delay, thereby ensuring effective remote monitoring. The Telegram monitoring result is presented in Figure 8.




Source: (Research Result, 2026)

Figure 8. Telegram Monitoring Result

The Telegram Bot testing results are summarized in Table 3.

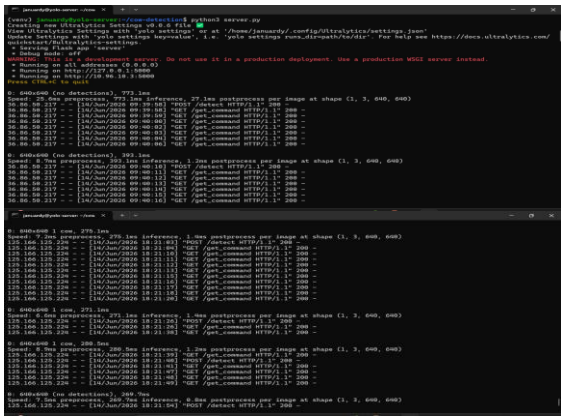
Table 3. Telegram test result

| No | Testing | Result |
|---|---|---|
| 1 | Sending photos | Succeed |
| 2 | Message delivery | Succeed |
| 3 | Real-time notification | Succeed |
| 4 | Command/test | Succeed |

Source: (Research Result, 2026)

3. Cloud Server Testing

The VPS performance was evaluated to assess its capability to receive image data transmitted from the ESP32-CAM and execute object detection using the YOLOv8 model. The evaluation focused on verifying the server's ability to process incoming image data accurately and efficiently while maintaining reliable communication with IoT devices in a cloud-based environment. The VPS detection result is presented in Figure 9.



Source: (Research Result, 2026)

Figure 9. VPS Detection Result

Techno Nusa Mandiri: Journal of Computing and Information Technology
Vol. 23, No. 2 September 2026 | DOI https://doi.org/10.33480/techno.v23i2.8613

The VPS functional testing results are summarized in Table 4.

Table 4. VPS Server test result

| No | Testing | Result |
|---|---|---|
| 1 | Image reception | Succeed |
| 2 | Flask running | Succeed |
| 3 | YOLOv8 detection | Succeed |
| 4 | Telegram notification | Succeed |

Source: (Research Result, 2026)

Based on the experimental results, all functions of the VPS server operated successfully, including receiving image data from the ESP32-CAM, executing object detection inference using the YOLOv8 model, and transmitting the detection results to users via Telegram. These findings demonstrate that the cloud server effectively serves as the centralized platform for data processing and communication within the proposed IoT-based monitoring system. The successful execution of all server-side functions confirms the feasibility of deploying cloud-based object detection to support reliable real-time monitoring applications.

4. Object Detection Testing Using YOLOv8
The YOLOv8 object detection results at different testing distances are presented in Table 5.

Table 5. YOLOv8 test result

| Object Distance | Detection Status | Confidence |
|---|---|---|
| 2 Meter | Succeed | 95% |
| 4 Meter | Succeed | 93% |
| 6 Meter | Succeed | 90% |
| 8 Meter | Succeed | 88% |
| 10 Meter | Succeed | 82% |

Source: (Research Result, 2026)

The experimental results demonstrated that the YOLOv8 model achieved reliable object detection performance across various testing distances. These findings are consistent with those reported by (Wijayanti et al. 2025), who concluded that YOLOv8 delivers high detection accuracy and robust real-time performance, making it well suited for computer vision-based monitoring applications. The consistency between the present study and previous research further confirms the effectiveness of YOLOv8 for intelligent real-time monitoring systems.

The experimental results demonstrated that the YOLOv8 model achieved reliable object detection across a testing distance range of 2–10 meters. Although the detection confidence score gradually decreased as the distance between the object and the camera increased, the model maintained accurate object identification throughout the evaluated range. These findings demonstrate the robustness of the YOLOv8 model for real-time agricultural field monitoring and confirm its suitability for intelligent IoT-based monitoring applications.

Environmental Limitations. Although the proposed system demonstrated successful object detection under the tested field conditions, its performance may be affected by environmental conditions that were not systematically evaluated in this study. Under low-light or nighttime conditions, the ESP32-CAM may produce images with reduced brightness, noise, and lower visual detail, which can decrease the confidence of YOLOv8 detection. Similarly, rain, fog, and water droplets on the camera lens may reduce image clarity and visibility (Kumar & Muhammad, 2023). These conditions may cause the PIR sensor to detect movement while the captured image is insufficient for reliable object verification. Therefore, the current results should not be interpreted as representing performance under all environmental conditions. Future field testing should specifically evaluate the system during nighttime, rain, and fog conditions and consider additional illumination or camera protection mechanisms.

5. System Delay Testing
The end-to-end system delay measurements are summarized in Table 6.

Table 6. Delay test results

| Proses | Delay |
|---|---|
| PIR → ESP32-CAM | 1-2 second |
| ESP32-CAM → VPS | 2-4 second |
| VPS → Telegram | 3-5 second |
| Total Delay | 5-10 second |

Source: (Research Result, 2026)

The experimental evaluation showed that the proposed system required an average of 5–10 seconds to deliver notifications from the moment an object was detected until the notification was received by the user (Afrin et al., 2025). This end-to-end response time indicates that the system is capable of supporting real-time monitoring by providing timely notifications while maintaining reliable communication between the ESP32-CAM, VPS-based cloud server, and Telegram Bot.

6. Online and Offline System Testing
The online and offline system testing results are summarized in Table 7.

Table 7. Online and Offline system testing

| Mode | Telegram | Alarm | VPS |
|---|---|---|---|
| Online | Active | Active | Active |
| Offline | Not active | Active | Not active |

Source: (Research Result, 2026)

The offline mode remained functional by generating audible alarm notifications even when the Internet connection was lost. This capability ensures that local warning mechanisms continue to operate independently of network availability, allowing the system to provide immediate alerts despite communication failures. Consequently, the proposed monitoring system maintains operational reliability under both online and offline conditions.

7. Comparative Analysis of YOLOv8
YOLOv8 was selected as the object detection model because its real-time detection capability is suitable for the proposed cloud-assisted monitoring architecture. Previous comparative studies have investigated different versions of the YOLO family and reported that model performance varies according to the model version, tracking configuration, dataset, and computational environment (Alkandary et al. 2025).
YOLOv8 has also been applied to camera-based object detection under challenging environmental conditions, demonstrating that detection performance can be affected by weather and image quality (Kumar & Muhammad, 2023). In this study, YOLOv8 was selected as a practical server-side detector because the computationally intensive inference process can be performed on the VPS rather than on the resource-constrained ESP32-CAM. However, this study does not claim that YOLOv8 is universally superior to other detectors because no direct same-dataset comparison experiment was conducted.

8. Discussion
The cloud server performance evaluation demonstrated that the VPS reliably received image data transmitted from the ESP32-CAM, executed the Flask-based application, performed object detection inference using the YOLOv8 model, and consistently delivered the detection results to users via Telegram. These findings are consistent with those reported by (Arun et al. 2025), who demonstrated that cloud-based communication enables centralized computational processing, thereby improving the efficiency, scalability, and reliability of IoT-based monitoring systems. The successful execution of all server-side functions confirms the effectiveness of the proposed cloud architecture for supporting real-time object detection and remote agricultural monitoring.
The delay evaluation demonstrated that the proposed system required an average end-to-end response time of 5–10 seconds, measured from the moment an object was detected until the notification was successfully received by the user. This response time is considered adequate for real-time agricultural field monitoring applications. The results indicate that the integration of the ESP32-CAM, Telegram Bot, a VPS-based cloud server, and the YOLOv8 object detection model provides an effective and reliable IoT-based monitoring solution for agricultural environments.

## CONCLUSION

The proposed Internet of Things-based animal monitoring and notification system successfully integrates PIR-based motion detection, ESP32-CAM image acquisition, YOLOv8 object verification, VPS-based processing, Telegram notification, relay-controlled alarm, and solar-powered operation. The available YOLOv8 training record used 438 images with the documented target class of cattle and reported 93.58% precision, 94.23% recall, and 97.36% mAP@0.50 after 300 training epochs. The field prototype was supported by a 20 Wp solar panel, a 12 V 3.5 Ah battery, and a PWM 12 V–10 A charge controller. System testing demonstrated successful image transmission, cloud-based object detection, Telegram notification, and local alarm operation, with an end-to-end notification delay of approximately 5–10 seconds under the tested network conditions. However, the system has limitations under low-light, nighttime, rain, and fog conditions, while component-level power measurements were not available to determine exact battery autonomy. In addition, the exact training, validation, and testing dataset split was not documented in the available training record. Future work should evaluate the system under different environmental conditions, document the complete dataset split and additional evaluation metrics, compare YOLO versions using the same dataset, and conduct long-duration power consumption measurements to determine battery autonomy and energy efficiency.

## REFERENCE

Widyastutik, Hotsawadi, Setyawati, D., Amaliah, S., & Hermawan, I. (2025). PERSEPSI PELAKU USAHATANI TERHADAP KEBIJAKAN DAN PROGRAM INVESTASI PUBLIK DI SEKTOR

PERTANIAN. *RISALAH KEBIJAKAN PERTANIAN DAN LINGKUNGAN Rumusan Kajian Strategis Bidang Pertanian Dan Lingkungan* , *12* (1), 59-68. doi: 10.29244/jkebijakan.v12i1.62166

Forqan, B. N., Muslim, A., Aulia, R., Pertiwi, R. A., & Maulana, A. (2024). Dampak digitalisasi terhadap tingkat ekonomi petani di Desa Karang Indah sebagai percontohan desa cerdas. Jurnal Agrimansion, 25(2), 436–443. https://doi.org/10.29303/agrimansion.v25i2.1680

Sari, U. K., Dipokusumo, B., & Septiadi, D. (2023). Analisis tingkat kesejahteraan ekonomi rumah tangga petani jagung lahan kering di Kecamatan Alas Barat Kabupaten Sumbawa. *Jurnal Agrimansion, 24*(2), 295–309. https://doi.org/10.29303/agrimansion.v34i2.1457

Rencani, N. K., Jemarut, W., Sumantri, I. N., & Rizal, P. (2024). *Strategi penyelesaian konflik kerusakan lahan pertanian akibat hewan ternak (Studi kasus di Lampok–Sumbawa)*. *Jurnal Ilmu Hukum, Humaniora dan Politik, 5*(2), 1546–1554. https://doi.org/10.38035/jihhp.v5i2.3516

Limbong, A. F., & Harsanto, B. (2025). Analisis Karakteristik dan Peran Aktor dalam Rantai Pasok Pertanian Berkelanjutan di Indonesia. Jurnal Pertanian Terpadu, 13(2). https://doi.org/10.36084/jpt.v13i2.642

Humam, F., & Triawan, M. A. (2024). Sistem keamanan ruangan menggunakan ESP32CAM dan sensor gerak berbasis IoT. *Infotek: Jurnal Informatika dan Teknologi, 7*(2), 575–584. https://doi.org/10.29408/jit.v7i2.26109

Yavvi Adi Nurachman, YZ Maulana, Gunawan Wibisono, dan Luqman Hakim, "Rancang Bangun Sistem Monitoring Kelembapan Tanah dan Suhu Udara Berbasis IoT", *J. SINTA: Sist. Inf. dan Teknol.Komputasi* , vol. 3, tidak. 1, hlm. 16–23, Desember 2025. https://doi.org/10.61124/sinta.v3i1.133

Tisna, D. R., Maharani, T., & Nugroho, K. T. (2024). Pemanfaatan chatbot Telegram untuk monitoring dan kontrol kualitas air menggunakan ESP32. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 9*(3), 1292–1306. https://doi.org/10.29100/jipi.v9i3.5329

Ramdan, A., & Asriyanik. (2024). Implementasi deteksi objek real-time sebagai media edukasi dengan algoritma YOLOv8 pada objek sampah. *Jurnal Saintekom: Sains, Teknologi, Komputer dan Manajemen, 14*(2), 142–153. https://doi.org/10.33020/saintekom.v14i2.638

Kalesaran, R. D. B., William, C., Redja, C., & Setyaningsih, E. (2025). Implementation of ESP32-CAM and Telegram bot for automatic customer arrival detection system. *TESLA: Jurnal Teknik Elektro, 26*(2), 178–185. https://doi.org/10.24912/tesla.v26i2.32950

Arrahma, S. A., & Mukhaiyar, R. (2023). Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32. *JTEIN: Jurnal Teknik Elektro Indonesia*, *4*(1), 60-66. https://doi.org/10.24036/jtein.v4i1.347

Rofianto, D., Fitra, J., Fathoni, H., Khoerunnisa, T. K., & Sari, H. P. (2026). IoT-Based Smart Irrigation and Fertilization System with Realtime Cloud Integration. Jurnal Nasional Teknik Elektro dan Teknologi Informasi, 15(1), 34–43. https://doi.org/10.22146/jnteti.v15i1.23124

Rosita, I., Jamal, J., Pamungkas, W. H., Atma, Y. D., Idris, N. B., & Daffa, A. (2022). Monitoring Ketinggian Permukaan Air Menggunkan Telegram Bot Berbasis NODEMCU ESP8266. *METIK JURNAL (AKREDITASI SINTA 3)*, *6*(2), 123-132. https://doi.org/10.47002/metik.v6i2.384

Kurniawan, N. A., Ruswanti, D., & Charolina, A. (2025). Implementasi Telegram Bot sebagai media monitoring gangguan pelanggan Telkom dengan pembaruan berkala berbasis mobile. *INSOLOGI: Jurnal Sains dan Teknologi, 4*(5), 1226–1240. https://doi.org/10.55123/insologi.v4i5.6234

K. Wijayanti, GA Mutiara, B. Suryawardani, E. Ervina, dan GP Kusuma, "Pemantauan Kerumunan Wisatawan Real-Time Non-Intrusif untuk Mitigasi Overtourism menggunakan Deteksi dan Pelacakan Kepala Berbasis YOLOv8", *J Robot Control (JRC)* , vol. 6, no. 4, hlm. 1985–2004, Juli 2025. https://doi.org/10.18196/jrc.v6i4.26396

Gyamfi, E. K., ElSayed, Z., Kropczynski, J., Yakubu, M. A., & Elsayed, N. (2024, April). Agricultural 4.0 leveraging on technological solutions: Study for smart farming sector. In *2024 IEEE 3rd international conference on computing and machine intelligence (ICMI)* (pp. 1-9). IEEE. https://doi.org/10.48550/arXiv.2401.00814

Arun, A. N., Lee, B., Castiblanco, F. A., Buckmaster, D. R., Wang, C. C., Love, D. J., ... & Ghosh, A. (2025). Ambient IoT: Communications enabling precision agriculture. *IEEE Communications Magazine*, *63*(4), 137-143. https://doi.org/10.1109/MCOM.003.2400508

Afrin, S., Rafa, SJ, Kabir, M., Farah, T., Alam, MSB, Lameesa, A., ... & Gandomi, AH (2025). Industrial Internet of Things: Implementasi, tantangan, dan solusi potensial di berbagai industri. *Komputer dalam Industri* , *170* , 104317.https://doi.org/10.1016/j.compind.2025.104317

Alkandary, K., Yildiz, A. S., & Meng, H. (2025). A comparative study of YOLO series (v3–v10) with DeepSORT and StrongSORT: A real-time tracking performance study. Electronics, 14(5), 876. https://doi.org/10.3390/electronics14050876

Kumar, D., & Muhammad, N. (2023). Object detection in adverse weather for autonomous driving through data merging and YOLOv8. Sensors, 23(20), 8471. https://doi.org/10.3390/s23208471