Vol. 2, No. 1, 2026 ISSN xxxx-xxxx (online)

Contents lists available at https://csi.unisi.my.id

Computer Science Insights

Journal Page is available to https://csi.unisi.my.id

Research article

# *Automatic Quail Coop System Based on IoT With Control of Temperature, Light, and Air Quality*

**M. Naofal Maulana Putra[1], M. Rizki Mahrifan[2], Singgih Yogo Pamungkas[3]*

*[1] Sistem Informasi, Universitas Islam Indragiri, Tembilahan, Riau*
*[2] Sistem Informasi, Universitas Islam Indragiri, Tembilahan, Riau*
*[3] Teknik Elektro, Universitas Negeri Yogyakarta, Purworejo, Jawa Tengah*
*email: [1]reifalsyptr7@gmail.com, [2] rzkimhrfn123@gmail.com, [3] singgihyogo26@gmail.com,*
** Correspondence*

**ARTICLE INFO**

***Article history:***
Received November 13, 2025
Revised December 23, 2025
Accepted June 1, 2026
Available June 2, 2026

***Keywords:***
Internet of Things
ESP32
Kandang Otomatis
DHT11
MQ135

**ABSTRAK**

Quail maintenance systems require stable environmental control to maintain livestock productivity and health. Factors such as temperature, light intensity, and air quality play an important role in the growth and egg production of quails. In practice, monitoring of cage conditions is often done manually, making it less efficient and prone to errors. This study aims to design and implement an automated quail cage system based on the Internet of Things (IoT) with temperature, light, and air quality control using the ESP32 microcontroller. A DHT11 sensor is used to measure temperature, an LDR sensor for light intensity, and an MQ135 sensor to detect air quality. The system is integrated with the Adafruit IO platform via the MQTT protocol, allowing users to monitor and control it in real-time from internet-connected devices. The test results show that the system is capable of detecting changes in environmental conditions and automatically activating actuators, such as turning on lights, opening cooling curtains, and operating waste cleaners. Thus, this system can improve quail maintenance efficiency, reduce manual intervention, and create more optimal cage conditions.

## 1. Introduction

Quail farming is a type of poultry that holds significant economic value, especially as a source of eggs [1]. The productivity of quails is greatly influenced by the conditions of their environment, including temperature, light intensity, and air quality [2]. An unstable cage environment can lead to stress in birds, reduce egg production, and increase the risk of diseases [3]. In conventional maintenance practices, the setup and monitoring of cage conditions are still carried out manually, which often leads to inefficiencies, requires more labor, and is less accurate in maintaining environmental stability [4].

The advancement of Internet of Things (IoT) technology offers opportunities to automate the monitoring and control of the environment in real-time via internet connectivity [5]. The Internet of Things (IoT) has been widely adopted in agriculture and livestock management to enhance efficiency, particularly in temperature control, humidity, and feeding systems. However, in some applications, air quality control, such as monitoring ammonia levels from poultry waste, is still often overlooked, even though this aspect directly impacts the health and comfort of quails [6].

Based on this, this research develops an automated quail cage system using ESP32, which is integrated with a temperature sensor (DHT11), a light sensor (LDR), and an air quality sensor (MQ135) [7]. This system is linked to the Adafruit IO IoT platform through the MQTT protocol, allowing users to oversee and manage devices from a distance. Besides tracking environmental conditions, this system can automatically trigger actuators like lights, ventilation curtains, and cage cleaning systems in response to variations in sensor data [8].

This research aims to design and implement an IoT-based automatic quail cage system that can maintain real-time stability in temperature, lighting, and air quality. Additionally, this study seeks to assess the effectiveness of the system in enhancing care efficiency and reducing reliance on manual intervention, thereby creating a more optimal cage environment that supports increased quail productivity.

## 2. Literature Review

The literature review includes an explanation of the concept of smart farming, the characteristics and functions of the ESP32 microcontroller, the working principles of IoT technology, the MQTT communication protocol, and the use of temperature, light, and air quality sensors. This review aims to provide a theoretical foundation that strengthens the system design and clarifies its connection to previous research.

### 2.1 Smart Automated Quail Cage System

The smart automatic quail coop is a technology application that utilizes sensors and microcontrollers to monitor and control the environmental conditions of the coop automatically. The primary goal of this system is to create a stable environment, ensuring that the quails are in a comfortable and optimal condition for egg production. Environmental factors such as temperature, lighting, and air quality directly impact the stress levels and productivity of the quails. With the implementation of an automatic system, the maintenance process can be carried out more efficiently, reducing reliance on manual supervision and minimizing the risk of human error [9]. Additionally, the implementation of smart barns allows for remote monitoring via digital devices such as smartphones or computers, enabling farmers to check the barn's conditions at any time.

### 2.2 ESP32 Microcontroller

The ESP32 is a 32-bit microcontroller that comes with built-in Wi-Fi and Bluetooth modules. This microcontroller boasts high processing capabilities and supports various communication interfaces such as UART, I2C, and SPI, making it highly compatible for Internet of Things (IoT) systems [10]. In an automatic quail cage system, the ESP32 serves as the control center responsible for reading data from sensors, processing control logic, sending data to the cloud platform, and executing actuator control commands like lights and servo motors. With its WiFi connectivity support, the ESP32 is particularly effective for implementing remote monitoring and control systems.

### 2.3 Internet of Things (IoT)

The Internet of Things (IoT) is a concept where various physical devices can connect to the internet and communicate with each other, exchanging data without direct user interaction. In the context of farming, IoT plays a crucial role in enabling real-time monitoring of barn conditions and remote control of systems [11]. With IoT, all environmental conditions measured by sensors can be visualized in a digital dashboard, making it easier for farmers to make timely and accurate decisions.

### 2.4 MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a lightweight data communication protocol designed for devices with low processing power and unstable network connections. It employs a publish/subscribe architecture, where devices send data to a server (broker), and other devices can receive that data by subscribing.

In this study, the ESP32 acts as a client that sends sensor data to Adafruit IO, which serves as the cloud broker, while also receiving commands to control actuators. The MQTT protocol was chosen for its bandwidth efficiency, ease of implementation, and support for real-time communication.

### 2.5 Cage Sensors

Environmental conditions in the quail cage are measured using several sensors, namely the DHT11, LDR, and MQ135. The DHT11 sensor is used to measure air temperature and humidity. Excessively high temperatures can cause heat stress in quails, which may negatively affect their health and productivity. The Light Dependent Resistor (LDR) is utilized to measure light intensity inside the cage, as lighting influences feeding activity and egg production. The MQ135 sensor is employed to monitor air quality by detecting the presence of ammonia gas generated from quail waste. High ammonia concentrations can lead to respiratory problems and reduced overall health of the livestock [12].

This research employs a system design approach based on the Internet of Things (IoT) to automate environmental control in quail cages. The system is designed to monitor temperature, light intensity, and air quality in real-time, while also automatically activating actuators in response to changes in environmental conditions. The research phases include hardware design, software development, IoT integration, and system testing.

## 3. Research Methods

This research method includes stages of hardware design, software development, system integration, and testing processes. The system is designed to monitor temperature, light intensity, and air quality in real-time, as well as control actuators like curtains, lights, and dirt-cleaning servos both automatically and manually. Additionally, users can monitor and control the system through the Adafruit IO Internet of Things (IoT) platform, either automatically based on sensor logic or manually via an online dashboard.

### 3.1 Hardware Design

The system framework utilized in this research is illustrated in the following Figure 1.

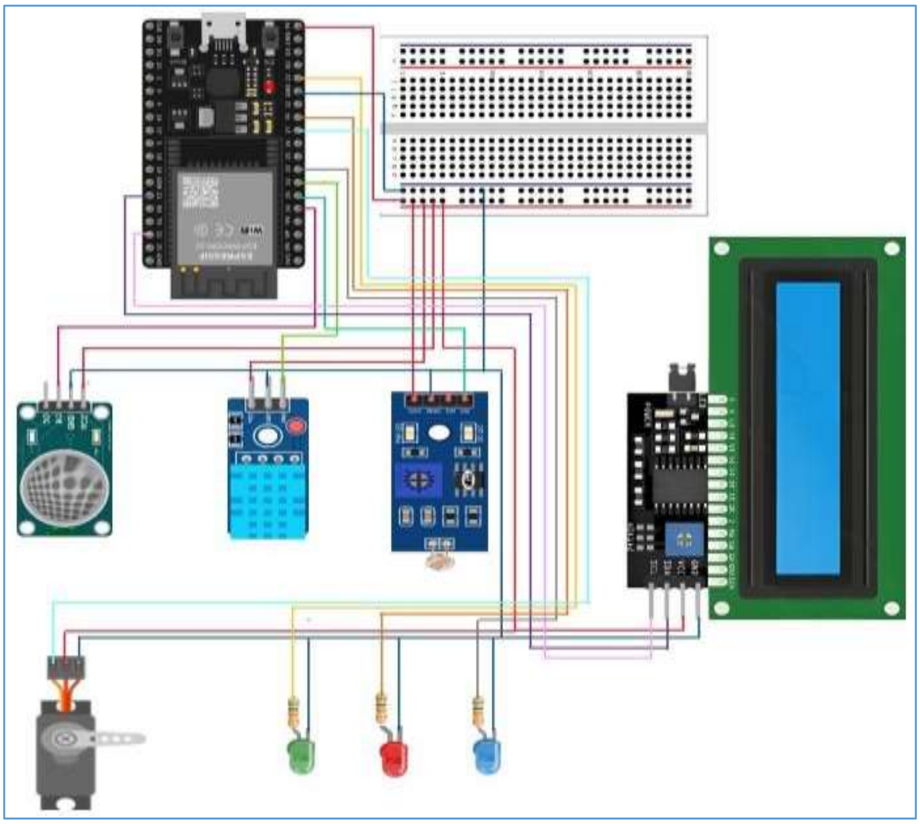

Figure 1. System Schematic Diagram

The ESP32 microcontroller serves as the control center for the system, equipped with a Wi-Fi module that facilitates communication with IoT platforms. The DHT11 sensor is utilized to measure temperature and humidity, while the LDR detects light intensity within the enclosure. To monitor air quality, the MQ135 sensor is employed, capable of detecting ammonia gas typically produced by quail droppings [13]. The actuators include LED lights for illumination and a servo motor to operate the waste disposal mechanism. Additionally, a 16x2 I2C LCD displays real-time information about the conditions in the enclosure, and a 5V adapter powers the system. With this combination of hardware, the system can autonomously detect environmental changes and adjust controls automatically according to predefined parameters [14].

### 3.2 Software Design

The system software is developed using the Arduino IDE to program the ESP32 microcontroller. The development process is supported by several libraries, including WiFi.h for network connectivity, AdafruitIO_WiFi.h for IoT integration, DHT.h for reading temperature and humidity sensors, ESP32Servo.h for controlling servo motors, and LiquidCrystal_I2C.h for displaying data on an LCD. The IoT system is managed through the Adafruit IO platform, which serves as a dashboard for monitoring sensor data and remote control. Additionally, the MQTT Panel app on smartphones is used as an alternative monitoring solution, making it easier for users to access data directly without needing to open the web dashboard.

### 3.3 Integration and Workflow of the System

The flowchart system illustrates the overall operational process of the system, which includes sensor readings, data transmission, and actuator control logic.

 ISSN xxxx-xxxx

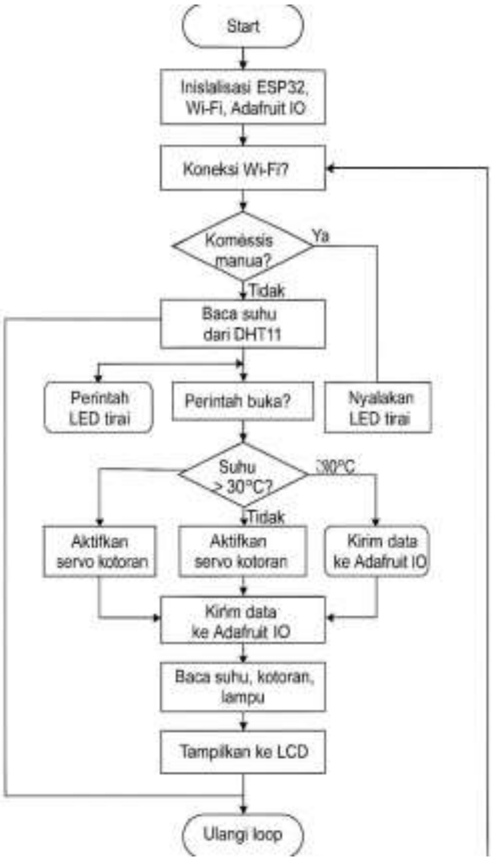


Figure 2. Flowchart of the IoT-Based Automatic Quail Coop System

The system's workflow begins with the initialization of the device. First, the ESP32 connects to the WiFi network and the Adafruit IO platform. Once the connection is successful, the DHT11, MQ135, and LDR sensors are initialized to be ready for environmental data readings. The actuators, which include a servo motor and an LED, are also configured, while an LCD screen is used to display the system's initial status. After the system is activated, the sensors periodically read the environmental conditions. The DHT11 sensor detects air temperature, the MQ135 sensor measures air quality in relation to ammonia gas levels, and the LDR sensor reads the light intensity inside the enclosure. The data collected from these readings is then sent to Adafruit IO every 15 seconds for real-time display and monitoring [15].

The system control logic operates automatically based on sensor values. If the temperature exceeds 30°C and the system is not in manual mode, the cooling curtain will activate, indicated by the curtain LED lighting up. However, if the user adjusts the curtain position via the Adafruit IO dashboard, the automatic mode will temporarily pause, and control will shift to the user. In terms of lighting settings, if the LDR value indicates darkness, the cage light will automatically turn on to maintain stable illumination. To ensure air quality, if the MQ135 sensor detects a value above the threshold (more than 4000), the servo motor will move to open the dirt cleaning mechanism for ten seconds before returning to its original position.

Additionally, the system is equipped with a feeding indicator in the form of an LED that blinks every nine seconds as a visual reminder. Information about the cage conditions, such as temperature, MQ135 value, curtain status, light intensity, and feeding indicator, is then displayed in rotation on the LCD screen every four seconds, allowing for direct monitoring without needing to open the IoT dashboard.

## 4. Results and Discussion

This section discusses the results of implementing an automatic quail cage system based on the Internet of Things (IoT), the integration of hardware with a cloud platform, and the analysis of the system's performance and efficiency. The implementation and testing processes were carried out to ensure that the system operates effectively according to the design outlined in the research methodology. The discussion results include the implementation of the tools, IoT integration and monitoring dashboard, testing outcomes, as well as an analysis of the system's accuracy, stability, and efficiency.

### 4.1 System Implementation

Implementation takes place after the hardware and software design phases are completed. The system comprises an ESP32 microcontroller, a DHT11 sensor, an LDR sensor, an MQ135 sensor, a servo motor, an LED light, and a 16x2 I2C LCD that displays temperature, light, and air quality data in real-time.

The system is assembled in such a way that the sensors can accurately capture environmental conditions. The DHT11 sensor is positioned in the center of the cage to read the temperature and humidity of the air. The LDR sensor is placed in an area exposed to light to detect the intensity of illumination. Meanwhile, the MQ135 sensor is installed near the source of waste to measure the ammonia gas levels produced by the quails. The actuator, in the form of a servo motor, is mounted on the waste disposal mechanism, while the LED light is used to automatically adjust the lighting based on light intensity. The following image shows the physical implementation results of the IoT-based automatic quail cage system that has been realized


Figure 3. Prototype of an IoT-Based Automatic Quail Cage System

The prototype shown in Figure 3 illustrates all the main components that are integrated into a single automated control system. The ESP32 acts as the primary controller, connecting all sensors and actuators. The sensors transmit environmental data to the microcontroller, which processes and executes commands based on the system's logic. The information from the sensor readings is displayed on an LCD screen, while users can also monitor and control the system remotely via the internet. The system's program code is developed using Arduino IDE to manage the functions of the sensors and actuators, as detailed in the following section.

```
#include <WiFi.h>
#include "AdafruitIO_WiFi.h" #include <DHT.h>
#include <ESP32Servo.h> #include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define IO_USERNAME    "singgih_8" #define IO_KEY
"aio_YEZZ49Pp59kz7jR4YXx3cgxKDoWd" #define WIFI_SSID"Wisma
Nusantara"
#define WIFI_PASS "bayardululah"

AdafruitIO_WiFi io(IO_USERNAME, IO_KEY, WIFI_SSID, WIFI_PASS);

AdafruitIO_Feed *suhuFeed = io.feed("SUHU");
```

Vol. 2, No. 1, 2026
(online)
ISSN xxxx-xxxx

```
AdafruitIO_Feed *kotoranFeed = io.feed("KOTORAN");
AdafruitIO_Feed *kontrolServoFeed = io.feed("KONTROL-SERVO");
AdafruitIO_Feed *kontrolTiraiFeed = io.feed("KONTROL-TIRAI");

#define DHTPIN 32
#define DHTTYPE DHT11
#define MQ135_ANALOG_PIN 34
#define LDR_PIN 35
#define LED_TIRAI_PIN 33
#define SERVO_KOTORAN_PIN 27
#define LED_LAMPU_PIN 14
#define LED_PAKAN_PIN 13
DHT dht(DHTPIN, DHTTYPE);
Servo servoKotoran; LiquidCrystal_I2C lcd(0x27, 16, 2);
 bool sedangBuangKotoran = false;
 unsigned long waktuMulaiBuang = 0; const unsigned long durasiBuang =
 10000;
 bool servoAktif = false; bool tiraiManual = false; bool
 tiraiStatus = false;
 unsigned long previousMillisPakan
 = 0;
 const unsigned long intervalPakan
 = 9000;
 bool ledPakanState = false;
 unsigned long lastSendTime = 0;
 const unsigned long sendInterval = 15000;
 unsigned long previousLcdMillis = 0;
 const unsigned long intervalLcd = 4000;
 int lcdStage = 0; void
 handleServoMessage(AdafruitIO_Data
 *data) {
   int perintah = data->toInt(); if (perintah == 1) {
     Serial.println("Perintah servo manual diterima!");
     servoKotoran.write(90); sedangBuangKotoran = true; waktuMulaiBuang =
     millis(); servoAktif = true;
   }
}
  } else if (perintah == 0) { tiraiManual = true; tiraiStatus =
     false;
     digitalWrite(LED_TIRAI_PIN, LOW);
     Serial.println("Tirai manual: MATI");
   } else {
     tiraiManual = false;
     Serial.println("Tirai kembali ke mode otomatis");
   }
```

```
}
void setup() {
  Serial.begin(115200); dht.begin();
  Wire.begin(21, 22); lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Koneksi WiFi...");
  pinMode(LED_TIRAI_PIN,  OUTPUT);   pinMode(LED_LAMPU_PIN,   OUTPUT);
  pinMode(LED_PAKAN_PIN, OUTPUT);
servoKotoran.attach(SERVO_KOTORAN_ PIN);
  servoKotoran.write(0);
  digitalWrite(LED_TIRAI_PIN, LOW);
  digitalWrite(LED_LAMPU_PIN, LOW);
  digitalWrite(LED_PAKAN_PIN, LOW);
io.connect();
kontrolServoFeed->onMessage(handle ServoMessage);
kontrolTiraiFeed->onMessage(handle TiraiMessage);
  while (io.status() < AIO_CONNECTED) {
Serial.print("."); delay(500);
Serial.println("\nTerhubung ke Adafruit IO");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("WiFi & IO OK!"); delay(2000);
  lcd.clear();
}
void loop() { io.run();
  float suhu =
dht.readTemperature(); int mq135Value =
analogRead(MQ135_ANALOG_PIN); int ldrValue =
analogRead(LDR_PIN);
  unsigned long currentMillis = millis();
  if (currentMillis - lastSendTime
>= sendInterval) {
    lastSendTime = currentMillis; if (!isnan(suhu))
suhuFeed->save(suhu);
    kotoranFeed->save(servoAktif ?
1 : 0);
  }
  if (!tiraiManual) { if (suhu > 30) {
      digitalWrite(LED_TIRAI_PIN, HIGH);
      tiraiStatus = true;
    } else {
      digitalWrite(LED_TIRAI_PIN,
LOW);
```

 ISSN xxxx-xxxx

```
tiraiStatus = false;
    }
  }
if (ldrValue < 400) {
    digitalWrite(LED_LAMPU_PIN, LOW);
  } else {
digitalWrite(LED_LAMPU_PIN,
if (!sedangBuangKotoran && mq135Value >= 4000) {
    servoKotoran.write(90); sedangBuangKotoran = true; waktuMulaiBuang =
currentMillis; servoAktif = true;
  }
  if (sedangBuangKotoran && currentMillis - waktuMulaiBuang >=
durasiBuang) {
    servoKotoran.write(0); sedangBuangKotoran = false; servoAktif =
    false;
  }
  if (currentMillis - previousMillisPakan >= intervalPakan) {
    previousMillisPakan = currentMillis;
    ledPakanState =
!ledPakanState;
    digitalWrite(LED_PAKAN_PIN, ledPakanState ? HIGH : LOW);
  }
  if (currentMillis - previousLcdMillis >= intervalLcd)
{
    previousLcdMillis = currentMillis;
    lcd.clear();
    switch (lcdStage) { case 0:
        lcd.setCursor(0, 0); lcd.print("SUHU : "); lcd.print(suhu);
        lcd.print((char)223); lcd.print("C");
        lcd.setCursor(0, 1);
lcd.print("KOTORAN: ");
       lcd.print(mq135Value); break;
case 1:
        lcd.setCursor(0, 0);
        lcd.print("TIRAI : "); lcd.print(tiraiStatus ?
"MENYALA" : "MATI");
        lcd.setCursor(0, 1);
        lcd.print("LAMPU : "); lcd.print(ldrValue);
        break; case 2:
        lcd.setCursor(0, 0);
        lcd.print("PAKAN : ");
        lcd.print(ledPakanState ? "MENYALA" : "MATI");
        break;}
    lcdStage = (lcdStage + 1) % 3;}
  delay(1000);}
```

### 4.2 Integration of IoT and Monitoring Dashboard

The system is developed with the integration of the Adafruit IO platform as an MQTT broker, enabling two-way communication between the ESP32 microcontroller and the cloud server. This platform displays sensor reading data in real-time and provides an interface for users to perform manual control.

Before sending data to Adafruit IO, the ESP32 first connects to the WiFi network and initializes all sensors. Temperature, humidity, air quality, and light intensity values are sent to the server every 15 seconds for real-time monitoring. Additionally, the system receives control commands from users through the "CONTROL-CURTAIN" and "CONTROL-SERVO" feeds. The Adafruit IO dashboard display is shown in Figure 4 below:

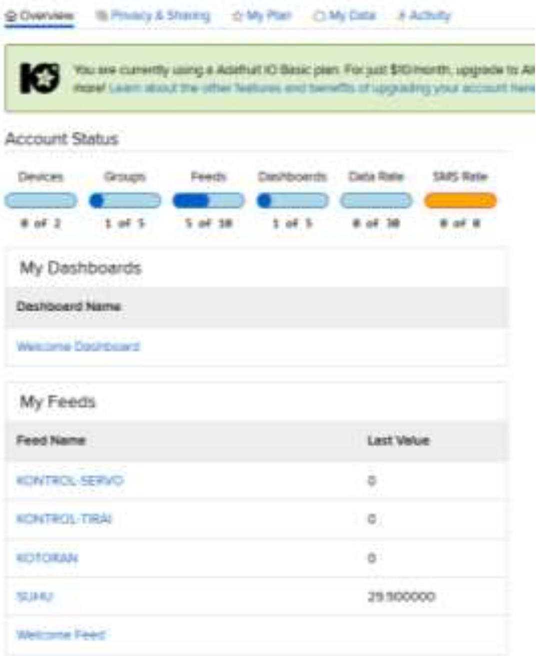


Figure 4. Adafruit IO Dashboard for Monitoring and System Control

Figure 4 shows the dashboard interface that includes feeds for temperature, light, gas, as well as controls for curtains and servos. Each sensor parameter is visually represented in both numerical and graphical formats. Users can switch between automatic and manual modes using the provided control buttons. In addition to the web dashboard, the system is also tested using the MQTT Panel app on Android devices, which offers easy access for quick monitoring and control without needing to open the Adafruit IO site.

### 4.3 System Testing Results

System testing was conducted to ensure that all sensors and actuators functioned according to the designed specifications. The tests were carried out by applying various environmental conditions, including temperature, light intensity, and gas concentration, and observing the system's automatic responses. The test results are presented in Table 1, which illustrates the relationship between sensors, actuators, and their active and inactive conditions.

**Table 1. Results of Sensor and Actuator Testing**

| Sensor | Actuator | Active Condition | Inactive Condition |
|---|---|---|---|
| **Temperature Sensor** | Curtain | Temperature > 30 °C | Temperature < 30 °C |
| **Light Sensor** | Lamp | Analog value < 80 | Analog value > 4000 |
| **Gas Sensor** | Waste Servo | Analog value > 3900 | Analog value < 1000 |
| **-** | Feed Indicator | Every 9 seconds | - |

The testing results indicate that the system is capable of responding to environmental changes quickly and in accordance with the predefined logic. When the temperature exceeds 30 °C, the cooling curtain is activated to reduce the temperature inside the cage. When light intensity is low, the lamp turns on automatically. If the

 ISSN xxxx-xxxx

ammonia gas level exceeds the threshold value of 3900, the servo motor is activated to operate the waste disposal mechanism for 10 seconds. In addition, the feed indicator LED blinks every 9 seconds to signal the automatic feeding schedule. Furthermore, the I2C LCD displays temperature data, air quality levels, curtain status, and feeding conditions alternately every 4 seconds. This feature allows users to directly monitor cage conditions without accessing the IoT dashboard.

### 4.4 Analysis of System Performance and Stability

This section discusses the evaluation results of the implemented automated quail cage system. The analysis aims to assess how well the sensors, actuators, and IoT connectivity operate in accordance with the system design. The evaluated aspects include sensor data accuracy, MQTT-based data communication stability, system efficiency in real-time cage monitoring, as well as system limitations and potential future improvements.

**a. Accuracy of Sensor Data Readings**

The system utilizes DHT11, MQ135, and LDR sensors to monitor temperature, air quality (ammonia gas), and light intensity, respectively. Sensor data are periodically read and transmitted to Adafruit IO via a WiFi connection using the MQTT protocol.

- The DHT11 sensor has a tolerance of ±2°C. Although it is less accurate compared to industrial-grade sensors such as the DHT22, the measurement results are sufficiently stable and suitable for monitoring cage temperature.
- The MQ135 sensor is used to detect ammonia gas generated from quail waste. When the analog value exceeds 4000, the system activates the servo motor to dispose of waste, indicating that the control logic operates correctly according to the predefined threshold.
- The LDR sensor detects light intensity, enabling the system to automatically turn on the lamp when low light conditions are detected (analog value < 400).

In addition, sensor data are displayed on a 16×2 I2C LCD, allowing users to directly monitor cage conditions on-site without accessing the IoT application.

**b. Data Communication Stability (MQTT)**

The system relies on the MQTT protocol through the Adafruit IO platform for bidirectional communication. The ESP32 transmits sensor data to the "SUHU" and "KOTORAN" feeds and receives control commands from the "KONTROL-SERVO" and "KONTROL-TIRAI" feeds. The connection between the ESP32 and Adafruit IO is maintained using the io.connect() function during initialization and the io.run() function within the main loop to ensure continuous communication. Visual feedback is provided through the LCD and serial monitor, displaying the message "WiFi & IO OK!" when the connection is successfully established. Based on testing results, the data communication remained stable without noticeable interruptions, even when the system operated continuously for extended periods.

**c. System Efficiency in Real-Time Monitoring**

Although the system does not directly measure electrical power consumption, its operational efficiency can be analyzed based on the automated control mechanisms:

- The curtain actuator is activated only when the temperature exceeds 30°C.
- The lighting system operates only when the LDR sensor detects low light conditions.
- The waste disposal servo is activated only when excessive gas levels are detected by the MQ135 sensor.

The data transmission interval is set to 15 seconds (sendInterval = 15000) to balance communication efficiency and ESP32 power consumption. As a result, the system remains energy-efficient while still being responsive to real-time changes in cage environmental conditions.

**d. System Limitations and Potential Improvements**

To enhance system performance in future developments, several limitations and improvement recommendations are summarized in Table 2.

Table 2. System Limitations and Proposed Improvements

| System Limitations | Proposed Improvements |
|---|---|
| Use of the DHT11 sensor, which has lower precision compared to DHT22 or other digital temperature sensors | Integration of energy sensors such as PZEM-004T to monitor power consumption of each actuator (servo, lamp) |
| Lack of notification systems (e.g., SMS or email) for critical conditions | Implementation of data logging using SD cards or Google Sheets |
| Absence of actual electrical energy measurement (current/voltage sensors) | Visualization of historical data in graphical form |
| – | Utilization of ESP32 sleep mode to improve battery efficiency in off-grid installations |

Based on the implementation and testing results, this IoT-based automatic quail coop system has successfully functioned as designed. The integration of the ESP32 microcontroller, environmental sensors, and the Adafruit IO platform demonstrates that the system can automatically maintain temperature, lighting, and air quality, while also allowing for remote control. The system's performance is assessed as stable, efficient, and user-friendly. These results indicate that the application of IoT technology in the livestock sector can enhance monitoring effectiveness, reduce manual intervention, and improve animal welfare through more precise environmental control.

## 5. Conclusion

This research successfully designed and implemented an IoT-based Automatic Quail Cage System capable of monitoring and controlling temperature, light intensity, and air quality in real-time. The findings indicate that the DHT11, LDR, and MQ135 sensors can provide the necessary data, and the system responds automatically through actuators such as lights, cooling curtains, and waste disposal servos. Additionally, integration with the Adafruit IO platform via the MQTT protocol operates smoothly, allowing for continuous data monitoring and enabling users to perform manual control remotely.

Based on these results, the system has proven to enhance cage maintenance efficiency by reducing manual intervention and improving environmental stability for the quails. The automatic control of temperature, light, and air quality contributes to creating a more ideal cage environment, while visual features like feed indicators and LCD displays facilitate direct monitoring on-site. Overall, the implementation of IoT in this cage system positively impacts the effectiveness of cage management and holds potential for increased productivity and livestock welfare.

## References


[1] A. D. N. Utomo and P. S. Berliana, “Analisa Penerapan Internet Of Things untuk Optimalisasi Efisiensi dan Produktivitas Budidaya Peternakan Telur Puyuh,” 2023.

[2] F. Komara, I. R. H. Soesanto, and S. Wahjuni, “Pengamatan Lingkungan Kandang Berbasis Internet Of Things (Iot) pada Pertumbuhan Ayam Pedaging,” *Ilmu Komput.*, vol. 11, no. 1, pp. 50–63.

[3] M. Sukafona, K. D. G. Supartha, A. A. G. B. Ariana, and M. D. P. Asana, “Sistem Monitoring Kandang Ayam Broiler Berbasis Iot untuk Optimalisasi Lingkungan Ternak di Desa Rendang, Karangasem,” *J. Soc. Work Empower.*, vol. 5, no. 1, 2025.

[4] Z. Abidin and N. A. A. Saragih, “Sistem Monitoring Kandang Burung Puyuh Berbasis Internet Of Things pada Platform Node-Red menggunakan Metode Naive Bayes,” no. 5, pp. 16–24.

[5] K. D. Goda and A. D. P. S. Neta, “Kajian Pengembangan Internet Of Things (IOT) pada Sektor Pertanian Di Kabupaten Ngada, Nusa Tenggara Timur,” *J. KRIDATAMA SAINS DAN Teknol.*, vol. 6, no. 02, pp. 478–493, 2024.

[6] R. N. Ariefin and Suryanto, “Sistem Monitoring Kualitas Udara, Suhu dan Kebersihan Kandang Ayam Otomatis Berbasis Internet Of Things,” *J. Ind. Manag. Technol.*, vol. 4, no. 2, pp. 117–123, 2023.

[7] F. Nuryadi, N. W. P. Septiani, and M. Lestari, “Implementasi Esp32 untuk Sistem Pemantauan Kesuburan Tanah Berbasis Iot,” *Semin. Nas. Ris. dan Inov. Teknol.*, pp. 79–86, 2025.

ISSN xxxx-xxxx

[8] I. F. Dzakiy, E. D. Marindani, and Syaifurrahman, "Perancangan Sistem Otomatisasi dan Monitoring Kandang Ayam Broiler berbasis Long Range (Lora) dan Internet Of Things (IOT)," *J. Innov. Creat.*, vol. 5, no. 2, pp. 2253–2264, 2025.

[9] R. A. Tambun, "Rancang Bangun Sistem Monitoring pada Kandang Burung Puyuh Petelur Secara Otomatis berbasis Internet Of Things," 2025.

[10] M. N. Ilham, D. Riandadari, A. M. Sakti, and Y. S. Widoretno, "Pengembangan Desain Kandang Burung Puyuh Otomatis berbasis IOT menggunakan Metode Quality Function Deployment (QFD) di Grosir Bibit Puyuh Pare," vol. 10, no. 0, pp. 509–516, 2025.

[11] H. Basori, F. Lazim, and A. Baijuri, "Autonomos Kandang Ayam Broiler berbasis IOT," *J. Ilm. Tek. dan Ilmu Komput.*, vol. 4, no. 3, pp. 218–224, 2025.

[12] J. D. Susatyono and Y. Fitrianto, "Sistem Monitoring Kualitas Udara dan Otomatisasi Pemberian Pakan Ayam Berbasis IOT," *J. Tek. Inform.*, vol. 9, no. 2, pp. 1–10, 2021, doi: 10.32832/kreatif.v9i2.5650.

[13] R. Muttaqin, W. S. W. Prayitno, N. E. Setyaningsih, and U. Nurbaiti, "Rancang Bangun Sistem Pemantauan Kualitas Udara berbasis IOT (Internet Of Things) dengan Sensor DHT11 dan Sensor MQ135," *J. Pengelolaan Lab. Pendidik.*, vol. 6, no. 2, pp. 102–115, 2024.

[14] A. Budiawan, R. R. Suryono, and D. Darwis, "Implementasi Sensor Gas Amonia Berbasis Internet Of Things pada Peternakan Ayam Potong dengan Sistem Monitoring dan Pengendalian Kualitas Udara Otomatis," *Mach. Learn. Comput. Sci.*, vol. 5, no. January, pp. 343–349, 2025.

[15] G. A. Rohman and A. R. Isnain, "Otomatisasi Pengendalian Suhu dan Kelembaban Berbasis Internet Of Things pada Kandang Ayam Potong," *Mach. Learn. Comput. Sci.*, vol. 5, no. April, pp. 558–565, 2025.