Design and Development of a Microcontroller-Based Automatic Chicken Feeder

Maulana Mansur
Mechanical Engineering, Faculty of Industrial Technology, National Institute of Technology
Email: maulanamansur2001@gmail.com

Received: 2025-02-26 Received in revised from 2025-09-26 Accepted: 2025-09-29

Abstract

The use of technology such as automatic feed management systems helps create convenience in feeding chickens regularly and optimally for chickens, most village chicken farmers have other activities such as work, etc. therefore farmers can be negligent or forget to feed their livestock, due to this problem, many farmers fail to raise livestock. That is why the idea was born to make an automatic chicken feed system based on the ESP8266 microcontroller that can provide feed for chickens automatically and regularly. This system uses the RTC (real time clock) conveyor as a timer input to run the system, after a predetermined time the ESP8266 will command the DC motor to rotate the spiral screw which will carry the feed from the hoper to the 3 containers where in the test results the feeding time for 3 minutes 40 seconds produces 700grams of feed where the first container contains 300grams, the second container 200grams and the third container 200grams.

Keywords: Auto; ESP8266; RTC; Screw Conveyor

1. Introduction

Advancements in technology, such as automated monitoring systems, feed management, and animal health maintenance technology, have enabled increased efficiency and productivity [1]. The use of technology, such as automatic feed management systems, helps facilitate regular and optimal feeding for chickens [2]-[3].

Feeding time is crucial for the growth and health of chickens, and consistent feeding can also reduce stress in the animals[4]. Raising chickens as livestock requires a significant amount of time and effort. Most backyard chicken farmers have other activities such as working, studying, and other responsibilities, which may lead to negligence or forgetfulness in feeding their livestock[5]. Due to this issue, many farmers fail in poultry farming.

This is why the idea of developing a microcontroller-based automatic chicken feeding system emerged. This system is designed to dispense feed regularly, operating automatically according to a pre-programmed schedule.

2. Metode

2.1. Flow Diagram

Below is an explanation of the flowchart presented in Figure 1:

a. Specification and Needs Identification The system is designed for 14 chickens with a feed capacity lasting for six days. Feeding is scheduled in two sessions: at 08:00 AM and 03:00 PM. The feeding process distributes feed into three containers.
b. Design Process. The design is divided into two parts. Structural Design – Focuses on constructing the chicken feeder and Control System Design – Develops the electrical control system based on specific needs.
    1) The development process consists of three main stages:
    2) Construction of the chicken feeder.
    3) Assembly of the structural and electrical control components.
    4) Program development.

5) System testing.
6) Automated feeding according to the scheduled times: from 08:00–08:04 AM and 03:00–03:04 PM.

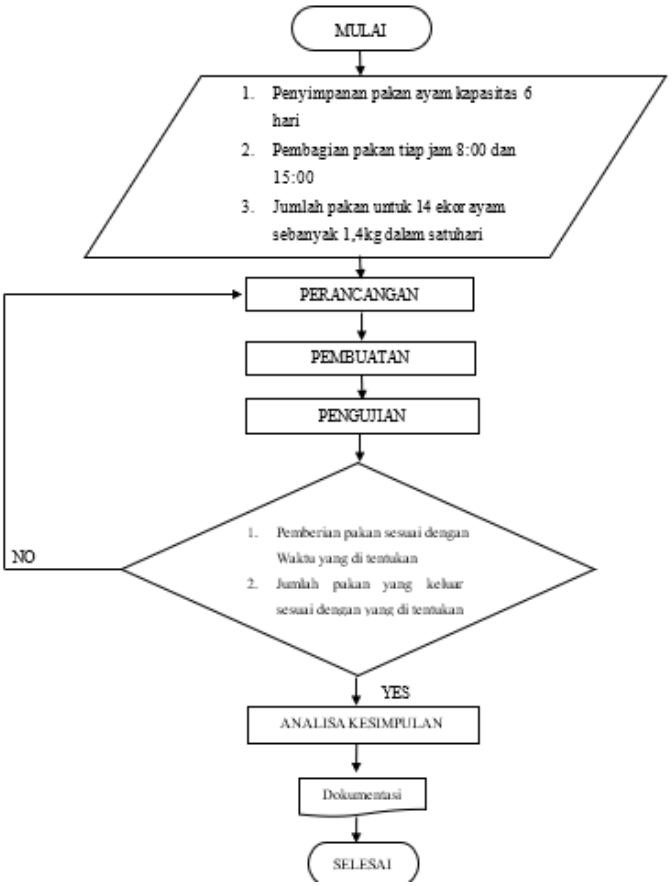


Figure 1. Flow Diagram

c. Feed Distribution. The amount of dispensed feed is predefined. In the first feeding session, 700 grams of feed is distributed into three containers First container: 300 grams, Second and third containers: 200 grams each, The second session follows the same distribution pattern. The total feed dispensed per day is 1.4 kg.
d. Analysis and Conclusion Collected data is analyzed to evaluate the performance of the automatic feeder. The results are processed to draw conclusions regarding the efficiency and effectiveness of the microcontroller-based chicken feeder system.
e. Documentation Includes report compilation, technical drawings, and specifications derived from the research.

2.2. Research Location

The research was conducted in Cimenyan Village, Bandung Regency. Most poultry farmers in Cimenyan still feed their chickens manually. Figure 2 illustrates the research location.

Figure 2. Research Location

2.3. Top View Layout of the Cage

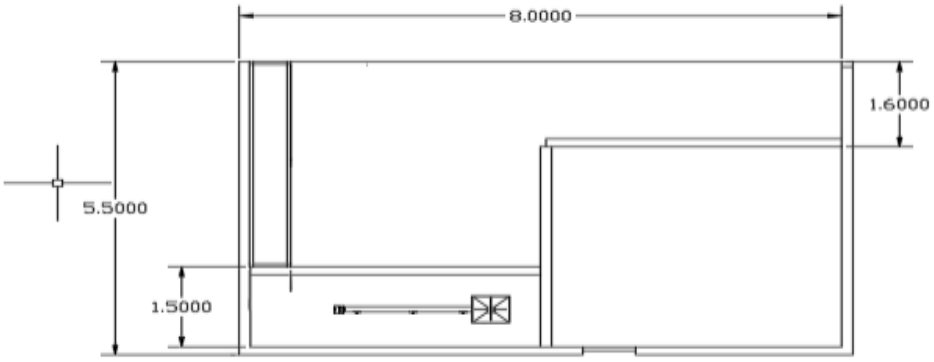


Figure 3. Top View Layout of the Cage

2.4. Tool Scheme

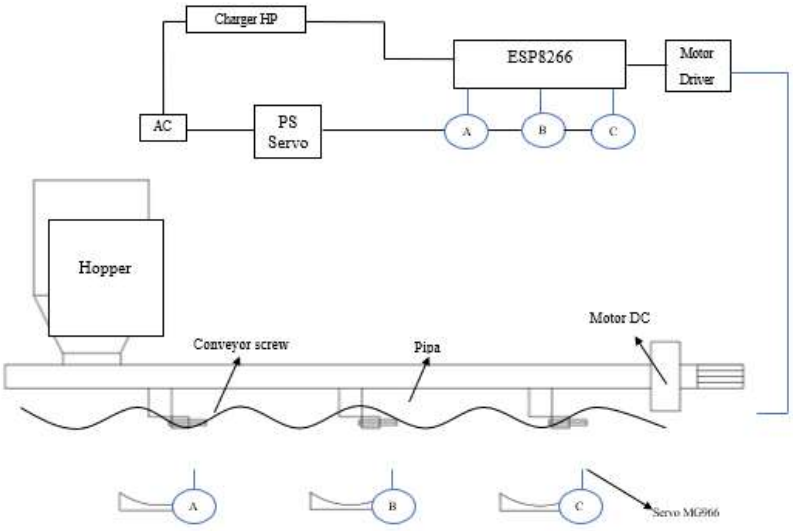


Figure 4. Tool Scheme

3. Result and Discussion

3.1. Rangkaian Node MCU ESP8266

NodeMCU ESP8266 Circuit. The ESP8266 circuit is used as the control system for this setup. The circuit board is shown in Figure 5 below, and more detailed specifications can be found in Table 1. The circuit integrates various components, including LCD for display, RTC (Real-Time Clock) for time management, Outputs such as DC motors and servo motors, Relay module to protect components from current surges, Step-down module to stabilize voltage These components work together to ensure efficient and stable operation of the system.

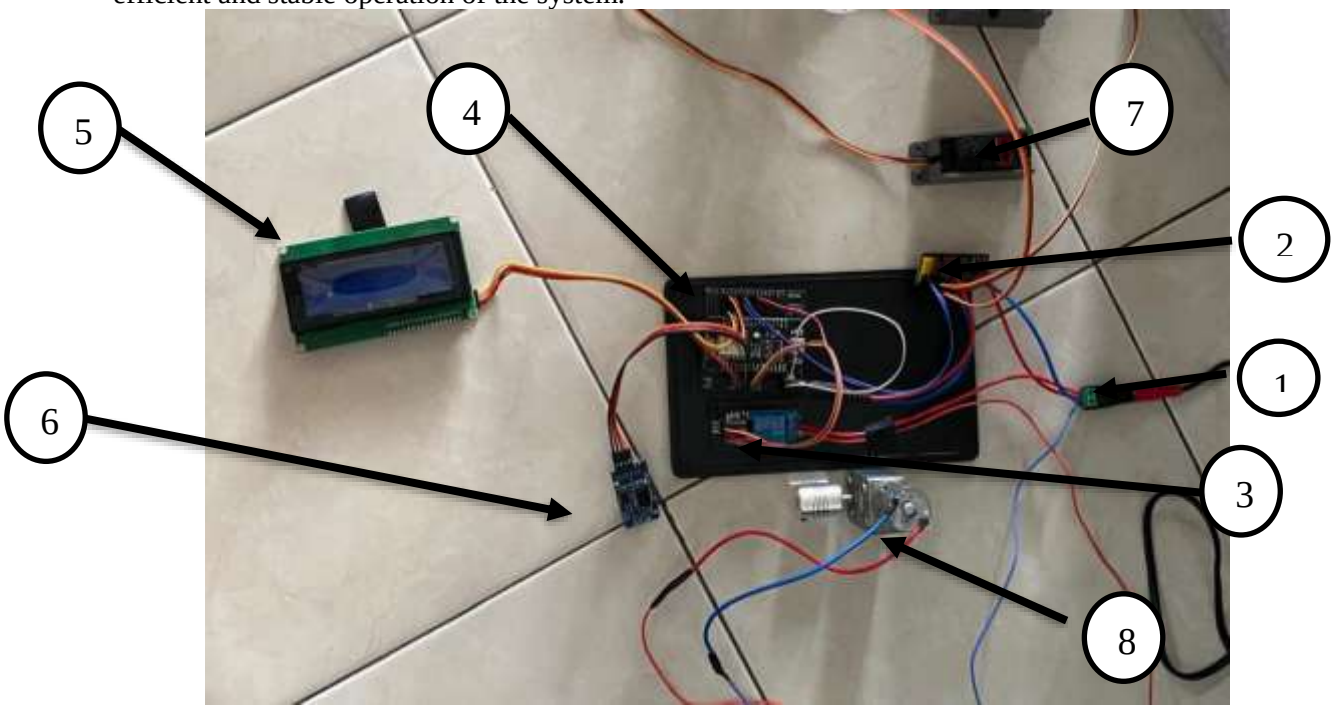


Figure 5. ESP8266 MCU Node Circuit

Information:

a. Jack DC *Male*
b. Step-down
c. Relay
d. NodeMcu ESP8266
e. LCD
f. RTC (*Real Time Clock*)
g. Motor Servo
h. Motor DC

Table 1. Circuit on ESP8266

Jack DC Male	Step down	Relay	ESP 8266	RTC	LCD	Servo 1	Servo 2	Motor DC
	IN							
+	+							
-	_							
	OUT							
	+		Power in					
	_		Power in					
	S		Pin D4					
	S		Pin D4					
+		Come						
		VCC	5V					
		IN	Pin D6					
		GND	GND					

			Pin D2	SCL				
			Pin D3	SDA				
			5V	VCC				
			GND	GND				
			Pin D1		GND			
			Pin D2		VCC			
			5V		SDA			
			Gnd		SCL			
	S					Orange	Orange	
	5V					Merah	Merah	
	G					Coklat	Coklat	
		NO						+
-								_

3.2. Device Construction

The construction of the device was carried out after determining the dimensions and materials of the automatic chicken feeder during the design phase. The assembly process was performed using bench work techniques.The final result of the automatic chicken feeder can be seen in Figure 6 below.

Figure 6. Chicken feeder

3.3. Hopper Construction

The hopper was made using plywood with a thickness of 1.5 cm. The dimensions of the hopper are as follows Width: 225 mm, Height: 360 mm, Length: 400 mm. The hopper construction process can be seen in Table 2 below.

Table 2. Hopper Construction

NO	**Tahapan pembuatan hooper**	**Proses pengerjaan**
1	First Stage Preparing Tools for Hopper Construction. The first step in constructing the hopper is to prepare the necessary tools. Below is a list of tools used: 1. Saw 2. Ruler 3. Right-angle ruler 4. Marker 5. Drill 6. Screws 7. Screwdriver	

2	The second stage involves measuring the plywood and drawing guidelines with a width of 225 mm, a height of 360 mm, and a length of 400 mm.	
3	Once the measurements are complete, the plywood is cut into five pieces.	
4	These pieces are then assembled into a box shape using screws for joining.	
5	After assembling the hopper, a hole with a diameter of 42 mm is drilled at the bottom using a drill machine. The exact drilling position follows the design specifications. Once the hole is made, the PVC pipe is inserted to ensure a proper fit. If the pipe fits correctly, the hopper construction is complete.	

3.4. PVC Pipe Construction

The PVC pipe was made using PVC material with a total length of 1.5 meters. Inside the hopper, the pipe was cut in half along its diameter. Original Pipe Diameter: 42 mm. Cut Diameter: 21 mm, Cut Length: 400 mm. The PVC pipe construction process can be seen in Table 3 below**.**

Table 3. PVC Pipe Construction

No	Stages of Pipe Manufacturing	Work Process
1	The first step is to prepare the tools needed to make the PVC pipe. The following tools will be used: 1. Saw 2. Ruler 3. Square ruler 4. Marker 5. Cutter 6. Drill	

2	The second step is to cut the pipe to the required length of 1.5 meters	
3	Before cutting the top section of the pipe, mark it with a guideline first, where the length of the mark is 400mm with a depth of 21mm.	
4	After completing the cutting process, the final result of the cut can be observed.	

3.5. Screw Cutting

The process of cutting the screw is done using a hacksaw. The screw originally has a length of 2 meters and is then cut to 1.5 meters. The image of the screw after being cut to 1.5 meters can be seen in Figure 7

Figure 7. Screw Cutting

3.6. Manufacturing of Join Screw

The join screw is made using a plate material with a thickness of 2.5mm. The plate is shaped into a cylinder with a diameter of 40mm. The shaft is then welded to the cylinder. The shaft itself has a diameter of 6mm and a length of 50mm. The image of the join screw can be seen in Figure 8**.**

Figure 8. Join Screw

3.7. Program Development

The program is developed using the Arduino IDE, which includes C/C++ libraries to simplify input and output operations. The programming itself is primarily done in Java. The coding process for controlling the automatic chicken feeder is outlined in Table 4.

Table 4. Program Development.

No	Creating coding for chicken feeder control
1	First download the Arduino Ide application
2	After downloading the Arduino IDE application, open it. Before starting the coding process, go to Tools and select the NodeMCU 1.0 (ESP-12E Module) board. 
3	Once the correct board is selected, the next step is to choose the port connected to the board by clicking Tools and selecting the appropriate port. 
4	After selecting the correct board, upload the required libraries that have been downloaded. The function of libraries in Arduino is to simplify the programming process. By uploading the library, there is no need to write the code from scratch. 
5	//Memanggil librar y Blynk, servo dan LCD. #include <BlynkSimpleEsp8266.h> #include <Servo.h> #include<Wire.h> #include<LiquidCrystal_I2C.h>

```
#include <RTClib.h>          // Include RTClib library for RTC
#include "Countimer.h"
Countimer CD;
int sec = 1000;

LiquidCrystal_I2C lcd(0x27, 20, 4);

//Atur pin Buzzer dan Led.
#define buz 13 //Sambungkan ke pin D7.
#define led 0 //Sambungkan ke pin D3.
#define rly 14 //Sambungkan ke pin D5.

String ssid = "Tselhome-48F0"; // Ganti nama WiFi
String pass = "80...."; // Ganti password

//Memanggil informasi dari Blynk.
//Masukkan Template, Nama, dan Token yang telah diberikan oleh Blynk.
#define BLYNK_TEMPLATE_ID "TMPL6LoRZlWdc"
#define BLYNK_TEMPLATE_NAME "Pakan Ayam"
#define BLYNK_AUTH_TOKEN "7KKTU1yPG2uZpaFFIPupHPPiiCX7Nve0"

RTC_DS3231 rtc;              // Create an RTC object
int jam, menit;

//Memberikan inisal atau nama lain untuk servo.
Servo servoku;

//Deklarasi Pakan dan Mode sebagai integer sekaligus variabel.
int Pakan;
int Mode;
bool countdownRunning = false; // Status countdown

// SETTING WAKTU UNTUK MENYALAKAN SECARA OTOMATIS
int onHour = 8;              // Set time for ON (e.g., 8 AM)
int onMinute = 0 ;
int onHour2 = 15;
int onMinute2 = 0 ;
//SETTING WAKTU HITUNG MUNDUR NYALA SERVO
int CDhour = 0;
int CDmin = 3;
int CDsec = 40;
//Memanggil fungsi setup.
void setup()
{
//Atur baudrate Serial monitor pada 9600.
Serial.begin(9600);
//memulai LCD.
lcd.begin();
//Atur buzzer dan led sebagai OUTPUT.
pinMode(buz, OUTPUT);
pinMode(led, OUTPUT);
pinMode(rly, OUTPUT);
digitalWrite(rly, HIGH);
//Atur pin servo pada pin 2 atau D4.
//Atur posisi awal servo pada 0.
servoku.attach(2); //Sambungkan ke pin D4.
servoku.write(0);
if (!rtc.begin()) {
Serial.println("Couldn't find RTC");
while (1);
}
if (rtc.lostPower()) {
Serial.println("RTC lost power, setting the time!");
rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Set RTC to compile time
```

```
}
//Atur ssid dan password WiFi.
//Cek sambungan WiFi dan nodeMCU.
WiFi.begin(ssid, pass);
Serial.print("Connecting to WiFi ..");
//mengatur display LCD sebelum WiFi tersambung.
lcd.setCursor(0, 0);
lcd.print(" Connecting ... ");
while (WiFi.status() != WL_CONNECTED) {
Serial.print('.');
//mengatur kedipan lampu sebelum WiFi tersambung.
digitalWrite(led, LOW);
delay(300);
digitalWrite(led, HIGH);
delay(300);
}
//mengatur display LCD setelah WiFi tersambung.
lcd.setCursor(0, 0);
lcd.print(" WiFi Connected ");
//mengatur kedipan lampu setelah WiFi tersambung.
digitalWrite(led, HIGH);
delay(100);
digitalWrite(led, LOW);
delay(100);
digitalWrite(led, HIGH);
delay(100);
digitalWrite(led, LOW);
delay(100);
digitalWrite(led, HIGH);
delay(100);
digitalWrite(led, LOW);
delay(500);
digitalWrite(led, HIGH);
//Atur sambungan Blynk dengan memasukkan token Blynk, ssid, dan password WiFi.
Blynk.begin(BLYNK_AUTH_TOKEN, "Pakan", "12345678");
Serial.println("Blynk Connected");
lcd.setCursor(0, 1);
lcd.print("Blynk Connected ");
delay(3000);
//Atur tampilan awal LCD.
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("   Pakan OTOMATIS ");
if (!rtc.begin()) {
Serial.println("Couldn't find RTC");
lcd.setCursor(0, 2);
lcd.print("Couldn't find RTC");
while (1);
}
if (rtc.lostPower()) {
Serial.println("RTC lost power, setting the time!");
rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // Set RTC to compile time
}
CD.setCounter(CDhour, CDmin, CDsec, CD.COUNT_DOWN, selesai);
CD.setInterval(waktu, sec);
}
//Memanggil fungsi pengulangan.
void loop()
{
DateTime now = rtc.now();  // Get current time
jam     = now.hour(), DEC;
menit   = now.minute(), DEC;
CD.run();
```

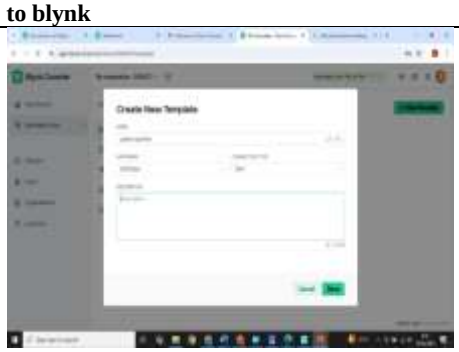
```
lcd.setCursor(0, 2);
lcd.print("Jam : ");
lcd.print(jam);
lcd.print("  ");
lcd.print("Menit : ");
lcd.print(menit);
lcd.print("  ");
lcd.setCursor(0, 3);
lcd.print("Count: ");
lcd.print(CD.getCurrentTime());
//Perintah untuk menjalankan Blynk.
Blynk.run();
//Menampilkan status Pakan pada Serial monitor.
Serial.println("Status Pakan :" + String(Pakan));
Serial.println(" Mode :" + String(Mode));
if( Mode == 0){
if (countdownRunning) {
lcd.setCursor(0, 1);
lcd.print("Servo Aktif    ");
digitalWrite(rly, LOW); // Relay tetap menyala
digitalWrite(buz, HIGH); // Buzzer tetap menyala
servoku.write(180); //ubah posisi servo menjadi 180.
delay(35000);
servoku.write(0); //ubah posisi servo menjadi 0.
delay(20000);
} else {
lcd.setCursor(0, 1);
lcd.print("Servo non-Aktif ");
digitalWrite(rly, HIGH); // Relay mati
digitalWrite(buz, LOW);  // Buzzer mati
servoku.write(0);        // Servo kembali ke posisi awal
}
// Cek apakah waktu saat ini berada dalam rentang waktu aktif
if ((jam == onHour && menit == onMinute) || (jam == onHour2 && menit == onMinute2)) {
CD.start();
countdownRunning = true; // Mulai countdown
lcd.setCursor(0, 3);
lcd.print("Count: ");
lcd.print(CD.getCurrentTime());
}
}
if (Mode == 1){
//Perintah saat kondisi servo dan motor dc nyala.
if (Pakan == 1 ) {
//tampilan LCD saat servo aktif.
lcd.setCursor(0, 1);
lcd.print("Servo Aktif    ");
digitalWrite(buz, HIGH); //perintah untuk mengaktifkan buzzer.
digitalWrite(rly, LOW); // Menyalakan relay
servoku.write(180); //ubah posisi servo menjadi 180.
delay(500);
servoku.write(0); //ubah posisi servo menjadi 0.
delay(500);
}
//Perintah saat kondisi servo dan motor dc mati.
if (Pakan == 0 ) {
//tampilan LCD saat servo aktif.
lcd.setCursor(0, 1);
lcd.print("Servo non-Aktif ");
digitalWrite(buz, LOW); //perintah untuk mengaktifkan buzzer.
digitalWrite(rly, HIGH); // Mematikan relay
servoku.write(0); //ubah posisi servo menjadi 0.
}
}
```

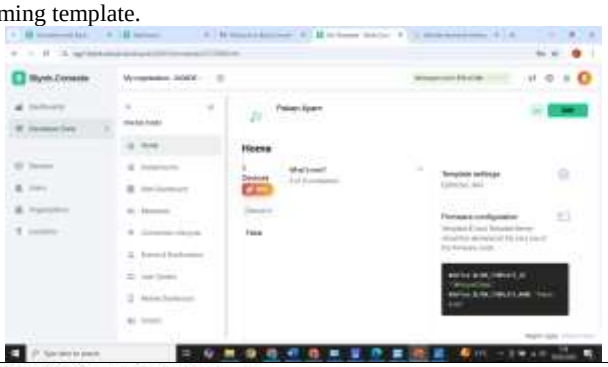
```
}
//Memanggil fungsi V0 pada Blynk.
BLYNK_WRITE(V0) {
Pakan = param.asInt(); //Memanggil variabel Pakan sebagai sebuah parameter.
}
//Memanggil fungsi V1 pada Blynk.
BLYNK_WRITE(V1) {
Mode = param.asInt(); //Memanggil variabel Pakan sebagai sebuah parameter.
}
void waktu() {
Serial.print("Countdown: ");
Serial.println(CD.getCurrentTime());
lcd.setCursor(0, 3);
lcd.print("Count: ");
lcd.print(CD.getCurrentTime());
}
void selesai() {
CD.stop();
countdownRunning = false; // Countdown selesai
lcd.setCursor(0, 0);
lcd.print("Countdown Selesai");
digitalWrite(rly, HIGH); // Mematikan relay
digitalWrite(buz, LOW);  // Mematikan buzzer
//Atur tampilan awal LCD.
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("   Pakan OTOMATIS ");
}
```

3.8. Connecting ESP8266 to Blynk

The first step in connecting ESP8266 to Blynk is to download the Blynk application. After downloading, create a Blynk account and log in. Once logged in, follow the steps to connect ESP8266 to Blynk, as outlined in Table 5.

Table 5. Connecting ESP8266 to Blynk**.**

No	How to connect ESP8266 to blynk
1	 The first step is to create a template according to your needs. Start by naming the template, then select ESP8266 as the hardware and choose WiFi as the connection type. Add a description if necessary.
2	

	Next, go to Datastreams, then click Virtual Pin to monitor or activate the system. Assign a name as desired, select Pin V1, and set the data type to Integer (whole numbers). Since the system only needs to be turned on or off, the integer value is set between 0 and 1.  For the servo, follow the same datastream settings, but use Pin VO.
3	 After configuring the datastreams, proceed to the Web Dashboard and select Switch to enable or disable the system. Then, go to Switch Settings, change the data stream mode to V1, and set Servo to V0 before saving.
4	Once the datastreams and switch setup are complete, configure a new device, select the template, and assign it to the device. 
5	After this, a Template ID will appear (as shown in the image below). Copy this Template ID and insert it into the programming template. 
6	 Below is the Blynk template for programming.

3.9. Connecting CCTV to the V380 Pro App

The first step to connect CCTV to your phone is to download the V380 Pro app. After downloading, create an account in the app. Once the account is created, log in to access the application.

To connect the CCTV to your phone, follow the steps outlined in table 6.

Table 6. Connecting CCTV to a Phone via V380 Pro.

NO	Connecting CCTV to HP
1	After downloading the V380 Pro application, click the **"+"** sign in the top right corner, then select **"Add Camera"** to scan the barcode on the CCTV. 
2	After scanning the barcode, connect the CCTV to the internet by entering the Wi-Fi ID and password. Once entered, proceed by scanning the barcode on the CCTV again. After completing all the steps, wait until the network configuration is successfully completed. 
3	Once the configuration stage is complete, the CCTV is ready to use. 

3.10. Testing

Testing was conducted to ensure that the device functions as designed. Therefore, the testing process included controller testing and measuring the feed output per minute. The completed device can be seen in Figure 9.

Figure 9. Overall image of the tool

3.11. Controller Testing

The testing was conducted before integrating the controller with the automatic chicken feeder. This test aimed to observe the system's activation and deactivation times. The testing was performed twice a day, from 08:00 to 08:05 and 15:00 to 15:05, over a 10-day period.

The observed variables included the activation time of the DC motor and servo motor, ensuring that the system operated precisely according to the predetermined schedule.

Table 7. Controller Testing presents the detailed results of these tests.

No	Date	Test Scenario	Time	Result
1	05-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
2	06-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
3	07-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
4	08-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
5	09-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light Up
6	10-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
7	11-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
8	12-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
9	13-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up
10	14-12-2024	The system is on for 5 menit	Hours 08:00-08:05 Hours 15:00-15-05	Light up Light up

3.12. Blynk to ESP8266 Testing

This test was conducted to evaluate the response of Blynk in manual mode during the chicken feeding process. The testing was carried out over five days at random times to assess the functionality of the manual mode in Blynk. The test results are presented in Table 8.

Table 8. Blynk to ESP8266 Testing.

No	Date	Test Scenario	Time	Result
1	12-02-2025	The system is turned on for 2 minutes	Hours 09:57 - 09:59 Hours 13:05 – 13:02 Hours 17:48 – 17:50	light up light up light up
2	13-02-2025	The system is turned on for 2 minutes	Hours 07:04 – 07:06 Hours 12:31 – 12:33 Hours 16:00 – 16:02	light up light up light up
3	14-02-2025	The system is turned on for 1 minutes	Hours 07:30 – 07: 31 Hours 10:28 – 10:29 Hours 13:27 – 13:28	light up light up light up
4	15-02-2025	The system is turned on for 3 minutes	Hours 09:00 – 09:03	light up

			Hours 14:13 – 14:16 Hours 18:00 – 19 :03	light up light up
5	16-02-2025	The system is turned on for 4 minutes	Hours 08:15 – 08-19 Hours 12:30 – 12:34 Hours 15: 25 – 15:29	light up light up light up

3.13. Feed Output Testing

This test was conducted by weighing the feed dispensed into feed containers 1, 2, and 3 using a scale. The automated chicken feeder was tested 10 times, with varying time settings in the program, to determine the optimal feed output. The results indicate that the most optimal feeding duration is 3 minutes and 40 seconds, with an average feed output of 712 grams. Refer to Figure 10 and Table 9: Feed Output Testing Process for detailed results.

Figure 10. Feed Output Testing

Description:

a. First Container
b. Second Container
c. Third Container
d. Feed Flow Direction

The feed output testing was conducted by adjusting the time settings and the opening of valves one and two in the program. The results of this testing can be seen in Table 9: Feed Output Per Minute.

Testing Parameters:

a. DC Motor Speed: Constant at 7.5 RPM
b. Device Activation Time: Varied
c. Valve One & Two: Opened and closed at varying intervals

Table 9. Testing of Feed Coming Out Per Minute

NO	time	Container 1	Container 2	Container 3	Valve closed	Valve opened
P1	1 min	200 gram	-	-	-	-
P2	2 min 30 sec	210 gram	150 gram	80 gram	15 sec	20sec
P3	2 min 30 sec	200 gram	150 gram	100 gram	20 sec	15 sec
P4	3 min	220 gram	190 gram	120 gram	25 sec	10 sec
P5	3 min	180 gram	100 gram	200 gram	30 sec	10 sec
P6	4 min	350 gram	200 gram	200 gram	30 sec	15 sec
P7	3 min 40 sec	320 gram	200 gram	200 Gram	35 sec	20 sec
P8	3 min 40 sec	320 gram	200 gram	200 Gram	35 sec	20 sec
P9	3 min 40 sec	310 gram	210 gram	200 Gram	35 sec	20 sec
P10	3 min 40 sec	300 gram	200 gram	200 Gram	35 sec	20 sec

4. Conclusion

After conducting research through the design, development, and testing of the microcontroller-based automatic chicken feeder, the following conclusions can be drawn:

a. Design Specifications. The obtained dimensions and capacity of the hopper are as follows: Length: 400 mm, Width: 42 mm, Height: 301 mm. With these dimensions, the hopper can hold 8.4 kg of feed. For the conveyor system, a shaftless screw conveyor was selected.
The screw has a capacity of 16 kg/hour. To dispense 700 grams of feed per session, the screw requires 3 minutes and 40 seconds to distribute the feed into each container.
The specifications of the screw conveyor are detailed in Table 10.

Table 10. Screw Conveyor Specifications

Conveyor screw tanpa poros	
Specifications	Description
Length	1,5m
Diameter	36mm
Thickness	4mm
Pitch	50mm
Hourly Capacity	16,1kg/jam
Rotation Capacity	35,73gram/putaran

In designing the controller, the component specifications and number of components used are as follows:

Table 11. Tool Specifications

No	**Component**	**Specification**	**Amount**
1	NodeMcu ESP8266	Nodemcu 49mm x 26mm Board 102mm x 51mm	1
2	RTC DS3231	P =38mm L = 22mm T = 14mm	1
3	Relay 5V 1 chanel	53mm x 28mm	1
4	Step down Sy8205	42mm x 20mm x 14mm	1
5	Motor Servo MG966R	40,7mm x 19,7mm x 42,9mm Torsi 9,4 kgfcm (4,8V)	2
6	Motor DC	12V Speed: 10RPM	1
7	LCD DSP-0012	60mm x 90mm	1
8	Adaptor 12 V	12V 3A	1
9	Kabel male to female	Diameter kabel 0,6mm-1mm	1
10	Kabel Awg 24	Diameter kabel 0,551mm	2
11	CCTV IPC-V380-E27-30	360° camera rotation, iOS/Android app support	1

b. From 10 test trials with varying times to achieve 700 grams of feed per feeding session, the results from tests P7-P10 showed that the average amount of dispensed feed was 710 grams. The required time for this was 3 minutes and 40 seconds, which was selected as the optimal duration since the feed output closely matched the target of 700 grams.

References

[1]Hutauruk, F. Y. (2017). Analisa Laju Korosi pada Pipa Baja Karbon dan Pipa Galvanis dengan Metode Elektrokimia. *Skripsi, Institut Teknologi Sepuluh November, Surabaya*.

[2]Sugara, I. W., Ansori, A. S. R., & Saputra, R. E. (2021). Perancangan Pakan Ayam Otomatis. *eProceedings of Engineering*, *8*(5).

[3]Kompiang, I. P., Supriyati, T. M., & Jarmani, S. N. (2001). Kinerja ayam kampung dengan sistim pemberian pakan secara memilih dengan bebas. *Jurnal Ilmu Ternak dan Feteriner*, *6*(2), 94-101.

[4]Jantan. *Jurnal Peternakan*, *17*(2), 114-124.

[5]Nurhapsa, N., Yusriadi, Y., & Nurhaedah, N. (2017). Campuran Pakan Herbal Untuk Ternak Ayam Kampung (Mixture Herbal Woof For Livestock Chicken Local). *Equilibrium: Jurnal Ilmiah Ekonomi, Manajemen dan Akuntansi*, *6*(1).

[6]Siregar, B., & Azis, A. (2016). Pengaruh pengaturan waktu pemberian pakan selama periode pertumbuhan ayam broiler terhadap rasio efisiensi penggunaan protein. *Jurnal Ilmu-Ilmu Peternakan*, *19*(2), 71-76.

[7]Zakariya, F. A. (2014). Analisa reaksi gaya screw conveyor pada rancang bangun mesin penggiling beras skala rumah tangga. Institut Teknologi Sepuluh Nopember *Surabaya, Surabaya*.

[8]Anzary, R. Z., Kurnia, D. A., & Nurdiawan, O. (2024). RANCANG BANGUN ALAT PAKAN IKAN OTOMATIS MENGGUNAKAN MIKROKONTROLER ESP8266 DENGAN BERBASIS INTERNET OF THINGS. *JTT (Jurnal Teknologi Terapan)*, *10*(1), 53-60.

[9]Mugiyono, S. (2001). Pengaruh serasah terhadap penampilan produksi dan kualitas ayam broiler. *Laporan Penelitian Fakultas Peternakan. Universitas Jendral Soedirman. Purwokerto*

[10]Rantawi, A. B. (2013). Perancangan Unit Transfer (Screw Conveyor) pada Mesin Pengisi Polibag untuk Meningkatkan Efektvitas Kinerja di Bidang Pembibitan. *Jurnal Citra Widya Edukasi*, *5*(1), 60-67.

[11]Nova, T. D., Heryandi, Y., & Ilham, P. (2020). Manajemen Pengaturan Persentase Pemberian Pakan pada Jadwal Waktu Pemberian Makan terhadap Tingkah Laku Makan Ayam Petelur

[12]Arduino Indonesia. (2023). Diakses pada 15 Oktober 2024 dari https://www.arduinoindonesia.id/2023/09/esp32-servo-motor-web-server-dengan-arduino-ide.html

[13]Bagye, W., & Zulkarnaen, M. F. (2024). Closed Circuit Television Cerdas Berbasis NodeMCU ESP-32. *Jambura Journal of Electrical and Electronics Engineering*, *6*(1), 13-17.

[14]*Conveyor Equipment Manufacturer Association* (CEMA). 2008. Belt Conveyor for Bulk Material: USA

[15]*Handson Technology (2021). Bts7960 High Current 43A H-Bridge Motor Driver*. Diakses pada 18 Oktober 2024 dari https://www.handsontec.com/dataspecs/module/BTS7960%20Motor%20Driver.pdf

[16]MesinTetasPaserKalimantanTimur.(2022) Diakses pada 10 Oktober 2024 dari http://www.youtube.com/@mesintetaspaserkalimantant6395

[17]Octavia, R., & Mu'min, N. (2023). Analisa Nutrisi Pakan Ayam Kampung Berbahan Baku Jagung dan Dedak. *Journal of Sustainable Research In Management of Agroindustry (SURIMI)*, *3*(1).

[18]Rucika. (2024). Diakses pada 15 Oktober 2024 dari https://www.rucika.co.id/produk/rucika-standard/

[19]Setiawan, R. (2020). Rancang Bangun Alat Pemberi Pakan Ikan Menggunakan Mikrokontroler. *Journal ICTEE*, *1*(1).