

## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi Sistem

Implementasi sistem adalah tahapan yang dilakukan setelah melakukan analisa dan perancangan sistem. Pada tahap ini akan membahas tentang perangkat keras dan perangkat lunak yang digunakan.

Pada sistem yang sudah dibuat, implementasi sistem berupa perangkat keras kemudian dirakit, lalu saling bertukar data agar sistem dapat berjalan sesuai dengan fungsi yang diinginkan.

#### 4.1.1 Perangkat Keras

Pada pembahasan implementasi Pakan Ternak Ayam Otomatis berbasis *Internet of Things* dibutuhkan beberapa komponen yang akan berintegrasi dengan kebutuhan sistem. Berikut beberapa komponen untuk membuat Sistem Pakan ayam Otomatis berbasis *Internet of Things*:

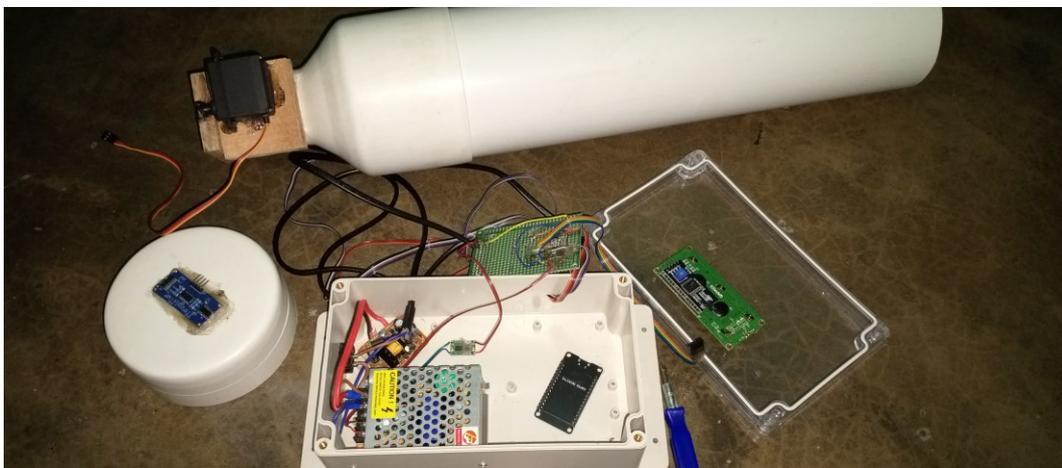
1. Module
  - a. Node MCU ESP32
2. Sensor
  - a. Sensor *Ultrasonic*
3. Komponen *prototype*
  - a. Kabel jumper
  - b. *Baseboard* ESP32
  - c. Motor Servo
4. Komponen penunjang
  - a. *Smartphone*
  - b. Power Adaptor 12V 2A
  - c. Kabel USB
  - d. Box
  - e. Pipa

#### 4.1.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam Sistem Pakan Ayam Otomatis berbasis *Internet of Things* ini adalah menggunakan Telegram yang terhubung dengan NodeMCU melalui token yang disediakan *bot* di Telegram.

#### 4.2 Rangkaian Keseluruhan Sistem

Berikut ini beberapa rangkaian keseluruhan sistem, yang akan disusun sebelum finishing dan beberapa penjelasannya.



**Gambar 4.1 Rangkaian Sebelum Disusun Didalam Box**

Berikut ini beberapa penjelasan fungsi pada masing-masing modul :

1. NodeMCU ESP32

Berfungsi sebagai mikrokontroler sekaligus sebagai penghubung antara perangkat keras dengan *Baseboard*.

2. Sensor *Ultrasonic*

Berfungsi untuk mendeteksi volume pakan di dalam tabung pipa.

3. Motor Servo

Berfungsi sebagai penggerak pembuka atau penutup pakan.

4. LCD I2C

Berfungsi untuk menampilkan data sensor.

5. Arduino Nano

Arduino merupakan sebuah platform elektronik yang bersifat open source dan sering digunakan untuk merancang dan membuat perangkat elektronik.

Platform Arduino didesain dengan tujuan utama untuk mempermudah penggunaan perangkat elektronik dalam berbagai aplikasi.

6. RTC (*Real Time Clock*)

Digunakan untuk mengatur detik, menit, hari, tanggal, bulan dan lain-lain. Dengan menggunakan Real Time Clock (RTC), kita bisa membuat program waktu yang tepat yang diperlukan dalam menyalakan lampu pada traffic light.

7. Power Supply Switching Adaptor 5V, 3A

Digunakan untuk bermacam macam piranti elektronik yg menggunakan tegangan 5 Volt, Kualitas Power Supply terbaik, menggunakan komponen berkualitas, Menggunakan casing full aluminium dengan disain khusus untuk pembuangan panas yang optimal.

8. Wifi *SwitchBoard*

Switchboard dengan fungsinya sebagai penghubung dan pembagi biasa disebut dengan panel hubung bagi atau perlengkapan hubung bagi (PHB).

9. Baseboard ESP32

Berfungsi sebagai papa dari penyangga dari perangkat yang digunakan sebagai pendukung.

10. Kabel Jumper

Merupakan kabel elektrik yang mempunyai pin konektor di setiap ujungnya dan memungkinkan untuk menghubungkan dua komponen.

11. Kabel USB

Digunakan sebagai konfigurasi pada bagian NodMCU ESP32

12. Box

Digunakan sebagai sarang dari perangkat yang akan dibuat didalam box.

13. Pipa

Digunakan sebagai penampung pakan ayam.

14. Smartphone

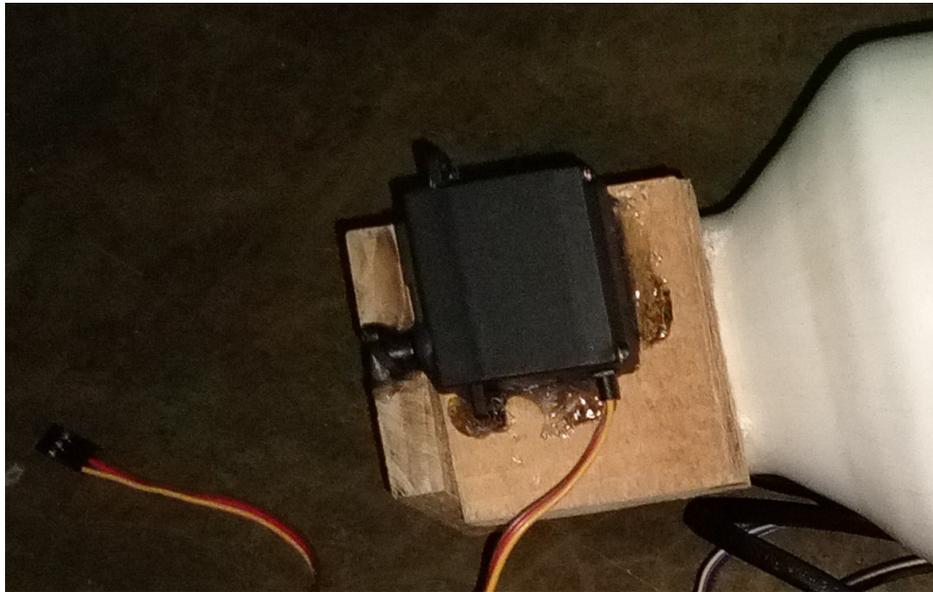
Digunakan sebagai sarana untuk mengatur waktu pada pakan ayam.

15. Power Adaptor 12

Sebagai media arus listrik pada tegangan perangkat.

#### 4.2.1 Rangkaian Motor Servo

Merupakan rangkaian yang sudah dirakit katup penutup dan pembuka pakan pada ayam.



Gambar 4.2 Rangkaian Motor Servo

#### 4.2.2 Rangkaian Sensor *Ultrasonic*

Merupakan rangkaian yang digunakan sebagai mengukur pakan ayam, dengan takaran yang sudah ditentukan.



Gambar 4.3 Rangkaian Sensor *Ultrasonic*

### 4.2.3 Rangkaian LCD

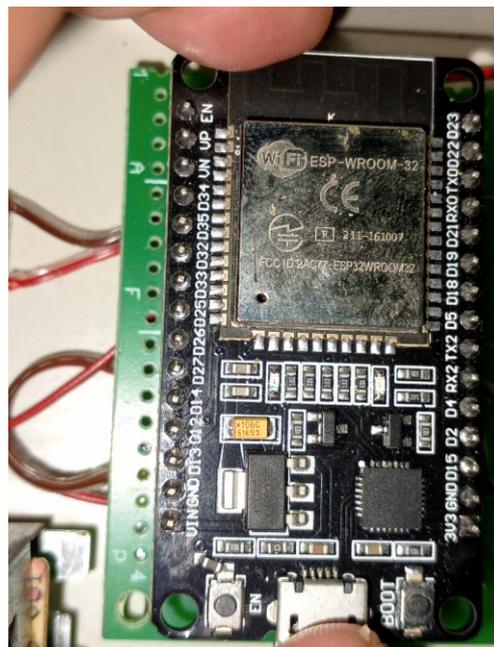
Rangkaian LCD yang terhubung dengan NodeMCU didalam Box. Seperti pada gambar berikut :



**Gambar 4.4 Rangkaian LCD**

### 4.2.4 Rangkaian NodeMCU

Merupakan gambar rangkaian NodeMCU yang berfungsi sebagai mikrokontroler yang mengendalikan semua sensor dan mengirim data ke Telegram.



**Gambar 4.5 Rangkaian NodeMCU**

#### 4.2.5 Rangkaian Sistem Sesudah Finishing

##### a. Rangkaian Tampak Dari Atas

Merupakan tampilan sistem yang dilihat dari sudut pandang tampak dari atas.



**Gambar 4.6 Rangkaian tampak Dari Atas**

##### b. Rangkaian Tampak Dari Samping

Merupakan tampilan sistem yang dilihat dari sudut pandang dari tampak samping.



**Gambar 4.7 Rangkaian tampak Dari Samping**

#### 4.2.6 Rangkaian Motor Servo Ke *Baseboard*

Berikut adalah rangkaian kabel Motor Servo yang digunakan untuk penggerak pembuka dan penutup pakan yang terhubung ke NodeMCU.

Keterangan :

- a. Kabel merah terhubung ke *5V Baseboard*
- b. Kabel coklat terhubung *Ground Baseboard*
- c. Kabel orange terhubung ke pin D5

#### 4.2.7 Rangkaian LCD ke NodeMCU

Berikut adalah rangkaian kabel yang digunakan untuk menampilkan data sensor yang melalui NodeMCU.

Keterangan :

- a. Pin SCL ke D1 NodeMCU
- b. Pin SDA ke Pin D2 NodeMCU
- c. Pin G ke *Ground*

#### 4.2.8 Rangkaian Sensor Ultrasonic

Berikut adalah rangkaian kabel Sensor Ultrasonic yang berfungsi untuk mendeteksi kualitas udara didalam ruangan.

Keterangan :

- a. Pin Analog ke A0
- b. Pin VCC ke 5V
- c. Pin G ke *Ground*

#### 4.2.9 Rangkaian NodeMCU

Berikut adalah rangkaian kabel NodeMCU yang berfungsi untuk menjalankan mikrokontroler dan juga koneksi internet.

Keterangan :

- a. Pin A0 ke Pin Analog sensor Ultrasonic
- b. Pin D1 ke SCL LCD I2C
- c. Pin D2 ke SDA LCD I2C
- d. Pin D5 ke Motor Servo

### 4.3 Integrasi Telegram

Telegram memiliki peran penting dalam monitoring sistem agar dapat melihat kinerja sistem. Telegram dalam pembuatan Sistem Pakan Ayam Otomatis berbasis *Internet of Things* sebagai monitoring yang akan berguna untuk memonitoring suhu, kelembaban dan kualitas udara.

Kemudian pada perangkat telegram ini terdapat perintah untuk buka dan tutup pakan sesuai jadwal yang ditentukan.

#### 4.3.1 Integrasi Telegram Dengan Bot Pakan Ayam

*Bot* Pakan Ayam adalah *bot* resmi dari Telegram yang tugasnya menciptakan *bot* baru. Menambahkan *bot* Telegram menggunakan *Bot* Pakan Ayam yaitu: Pada mesin pencarian *bot* Telegram ketik Pakan Ayam dan kemudian pilih *start*. Untuk Pembahasan *bot* Telegram adalah sebuah *bot* atau robot yang diprogram dengan berbagai perintah untuk menjalankan serangkaian instruksi yang diberikan oleh pengguna.

Pada aplikasi Telegram Pengguna dapat mengirim pesan, perintah, dan inline request dan juga dapat mengontrol *bot* menggunakan https ke API Telegram.

#### 4.3.2 Pembuatan *Bot* Pakan Ayam pada Telegram

Untuk membuat *bot* di Telegram, perlu menggunakan *bot* Pakan Ayam, yang merupakan *bot* resmi dari Telegram untuk membuat dan mengelola *bot*. Dalam pembuatan *bot* Pakan Ayam pada Telegram. Cara ini menggunakan Arduino IDE yang terhubung dengan cara dicoding. dengan mengakses *Bot* *Father* dari link <https://t.me/pakanayam>. Token API adalah hal utama yang diperlukan untuk dapat mengakses *Bot*. Token tersebut biasa digunakan di kode program.

Berikut adalah tata cara pembuatan Bot Telegram:

- i. Klik Start kemudian pilih */newbot*.  
Buka Telegram pada Smartphone untuk membuat *Bot* pakan ayam, jika menggunakan laptop maka perlu install telegram versi dekstop terlebih

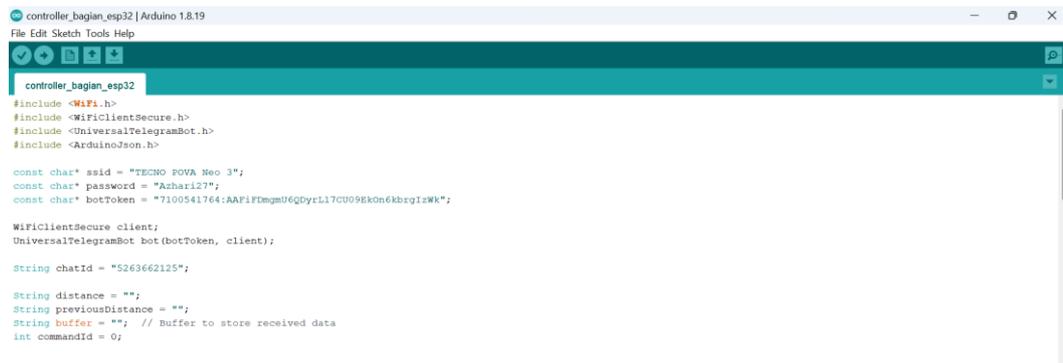
- dahulu. Apabila terdapat permintaan *launch* telegram dekstop maka pilih *allow* agar diarahkan ke telegram ke *chat Bot Pakan Ayam*.
- ii. Setelah itu proses */newbot* dan menuju untuk menulis *username* pada *Bot Pakan Ayam*.
  - iii. Membuat *nowbot* dan berhasil dibuat.
  - iv. Tuliskan judul *Bot* atau *username* yang akan dibuat, Ada aturan tertentu pembuatan *username* ini harus diakhiri dengan menggunakan “*bot*” diakhir judul seperti contoh:”Pakan Ayam Otomatis\_ *bot*”.
  - v. Tampilan *username* yang sudah dibuat pembuatan */newbot* dengan *username*.



**Gambar 4.8 Tampilan Username**

#### **4.4 Implementasi Sistem Pakan Ayam berbasis *Internet of Things***

Sistem Pakan Ayam berbasis *Internet of Things* adalah sebuah sistem yang memonitoring takaran pakan ayam, lalu kemudian data di tampilkan dalam LCD dan juga data sensor di kirimkan ke Telegram. Motor servo akan bergerak jika sudah memasuki waktu yang ditentukan dan berdasarkan tombol di Telegram. Berikut adalah *script* untuk menghubungkan dari NodeMCU menuju ke Wifi.



```

controller_bagian_esp32 | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_esp32
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "TECNO FOVA Neo 3";
const char* password = "Acha12";
const char* botToken = "7100541764:AAF1FtmgsU6QDyrL17GD9SEkOnGkbrgtzWk";

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

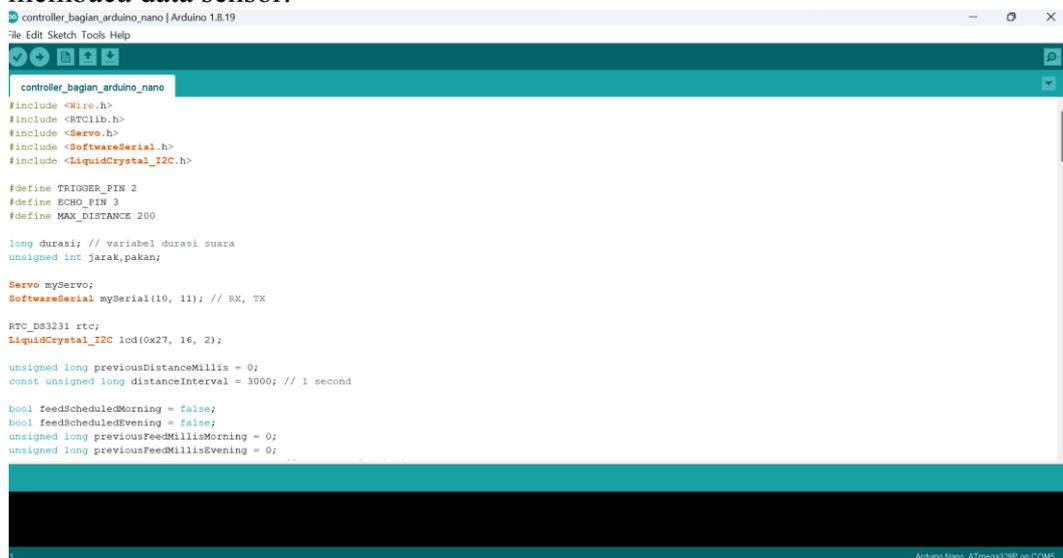
String chatId = "5263662125";

String distance = "";
String previousDistance = "";
String buffer = ""; // Buffer to store received data
int commandId = 0;

```

**Gambar 4.9 Script Terhubung ke Wifi**

Setelah menghubungkan ke wifi kemudian NodeMCU ESP 32 dan Arduino Nano membaca data sensor.



```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano
#include <Wire.h>
#include <RTClib.h>
#include <Servo.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>

#define TRIGGER_PIN 2
#define ECHO_PIN 3
#define MAX_DISTANCE 200

long durasi; // variabel durasi suara
unsigned int jarak,pakan;

Servo myServo;
SoftwareSerial mySerial(10, 11); // RX, TX

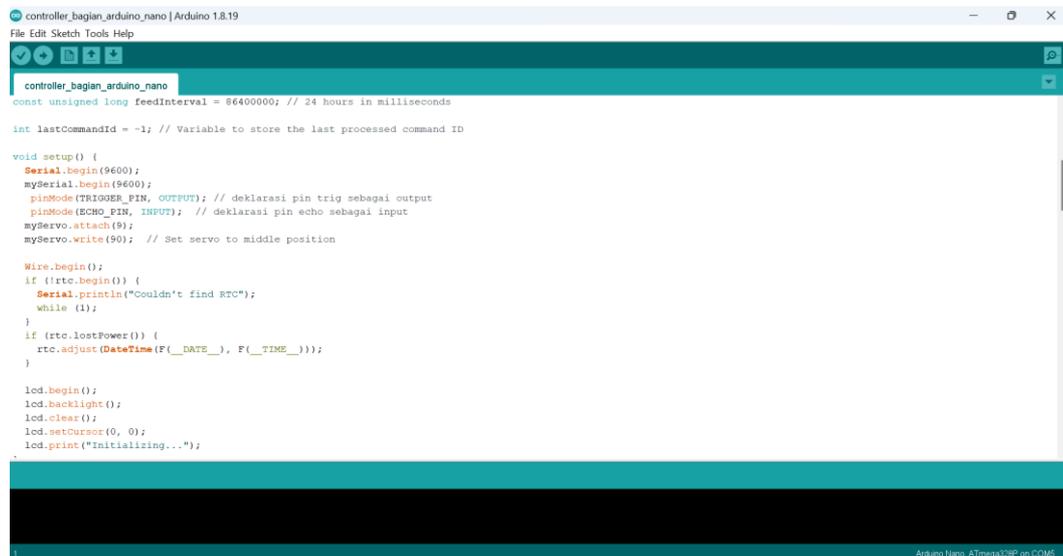
RTC_DS3231 rtc;
LiquidCrystal_I2C lcd(0x27, 16, 2);

unsigned long previousDistanceMillis = 0;
const unsigned long distanceInterval = 3000; // 1 second

bool feedscheduledMorning = false;
bool feedscheduledEvening = false;
unsigned long previousFeedMillisMorning = 0;
unsigned long previousFeedMillisEvening = 0;

```

**Gambar 4.10 Script Membaca Sensor**



```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano
const unsigned long feedInterval = 86400000; // 24 hours in milliseconds

int lastCommandId = -1; // Variable to store the last processed command ID

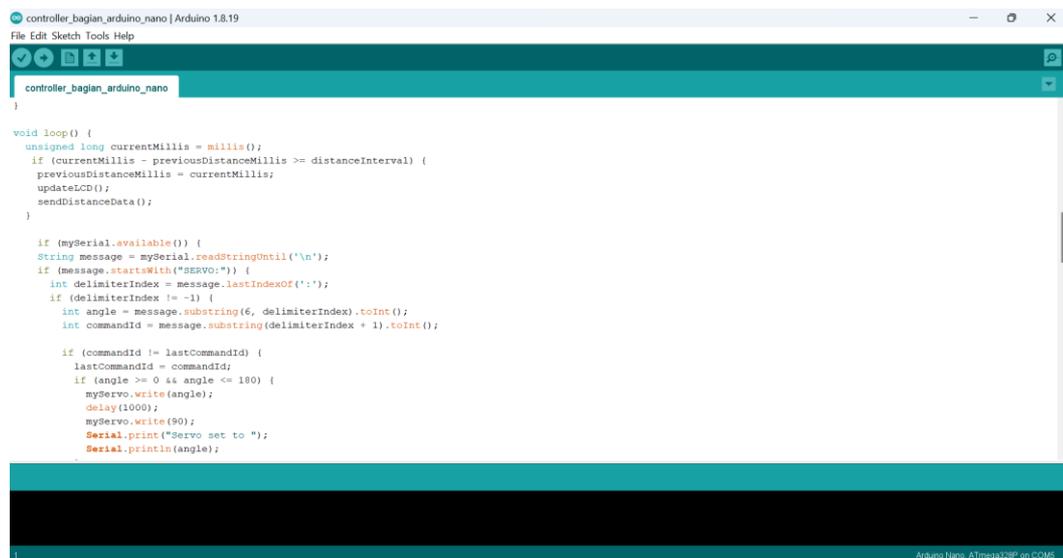
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
  pinMode(TRIGGER_PIN, OUTPUT); // deklarasi pin trig sebagai output
  pinMode(ECHO_PIN, INPUT); // deklarasi pin echo sebagai input
  myServo.attach(9);
  myServo.write(90); // Set servo to middle position

  Wire.begin();
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (rtc.lostPower()) {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initializing...");
}

```

**Gambar 4.11 Script Membaca Sensor (Lanjutan)**



```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousDistanceMillis >= distanceInterval) {
    previousDistanceMillis = currentMillis;
    updateLCD();
    sendDistanceData();
  }

  if (mySerial.available()) {
    String message = mySerial.readStringUntil('\n');
    if (message.startsWith("SERVO:")) {
      int delimiterIndex = message.lastIndexOf(':');
      if (delimiterIndex != -1) {
        int angle = message.substring(6, delimiterIndex).toInt();
        int commandId = message.substring(delimiterIndex + 1).toInt();

        if (commandId != lastCommandId) {
          lastCommandId = commandId;
          if (angle >= 0 && angle <= 180) {
            myServo.write(angle);
            delay(1000);
            myServo.write(90);
            Serial.print("Servo set to ");
            Serial.println(angle);
          }
        }
      }
    }
  }
}

```

**Gambar 4.12 Script Membaca Sensor (Lanjutan)**

```

void sendDistanceData() {
  digitalWrite(TRIGGER_PIN, LOW); // Trig tidak aktif
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH); // Trig aktif
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  // Membaca sinyal masuk pada echo
  durasi = pulseIn(ECHO_PIN, HIGH);
  // Menghitung Jarak
  jarak = durasi * 0.034 / 2; // Rumus menghitung jarak ultrasonik
  pakan = 44 - jarak;
  mySerial.print("DIST:");
  mySerial.println(pakan); // Send distance to ESP32
  Serial.println(pakan);
}

```

**Gambar 4.13 Script Membaca Sensor (Lanjutan)**

Setelah NodeMCU ESP32 berhasil membaca nilai sensor tersebut kemudian NodeMCU ESP32 mengirim data tersebut ke Telegram. Berikut adalah script untuk mengirim ke Telegram.

```
void setup() {
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, 16, 17); // RX=16, TX=17

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
```

**Gambar 4.14 Script Mengirim Data ke Telegram**



```
controller_bagian_esp32 | Arduino 1.8.19
File Edit Sketch Tools Help
controller_bagian_esp32
  delay(1000);
  Serial.println("Connecting to WiFi...");
}
Serial.println("Connected to WiFi");
client.setInsecure(); // Skip certificate verification for simplicity
}
```

**Gambar 4.15 Script Mengirim Data ke Telegram (Lanjutan)**

```
void loop() {
  // Read from Serial2 and store in buffer
  while (Serial2.available()) {
    char c = Serial2.read();
    Serial.println(c);
    if (c == '\n') {
      processMessage(buffer); // Process the complete message
      buffer = ""; // Clear the buffer
    } else {
      buffer += c;
    }
  }

  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

  while (numNewMessages) {
    for (int i = 0; i < numNewMessages; i++) {
      String chat_id = String(bot.messages[i].chat_id);
      String text = bot.messages[i].text;
    }
  }
}
```

**Gambar 4.16 Script Mengirim Data ke Telegram (Lanjutan)**

```

controller_bagian_esp32 | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_esp32
if (text == "/start") {
  bot.sendMessage(chat_id, "Welcome! Use /status to get distance or /servo to control servo.", "");
}

if (text == "/status") {
  bot.sendMessage(chat_id, "Pakan Tinggal: " + distance + " cm", "");
}

if (text.startsWith("/buka")) {
  int angle = text.substring(7).toInt();
  if (angle >= 0 && angle <= 180) {
    commandId++;
    Serial2.println("SERVO:" + String(angle) + ":" + String(commandId)); // Send the angle to Arduino Nano via Serial
    Serial.println(angle);
    bot.sendMessage(chat_id, "Setting servo to " + String(angle) + " degrees", "");
  } else {
    bot.sendMessage(chat_id, "Invalid angle. Please use a value between 0 and 180.", "");
  }
}

numNewMessages = bot.getUpdates(bot.Last_message_received + 1);
}

void processMessage(String message) {
  if (message.startsWith("FEED:")) {

```

**Gambar 4.17 Script Mengirim Data ke Telegram (Lanjutan)**

```

controller_bagian_esp32 | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_esp32
}
}
numNewMessages = bot.getUpdates(bot.Last_message_received + 1);
}

void processMessage(String message) {
  if (message.startsWith("FEED:")) {
    String timestamp = message.substring(5); // Extract timestamp
    bot.sendMessage(chatId, "Pakan telah diberikan pada jam: " + timestamp, "");
  } else if (message.startsWith("DIST:")) {
    distance = message.substring(5); // Extract distance
    Serial.print("Distance: ");
    Serial.println(distance);

    if (distance.toInt() < 4) {
      bot.sendMessage(chatId, "Pakan Habis, Mohon di isi kembali", "");
    } else {
    }

    if (distance != previousDistance) {
      bot.sendMessage(chatId, "Current distance: " + distance + " cm", "");
      previousDistance = distance;
    }
  }
}
}

```

**Gambar 4.18 Gambar Script Mengirim Data ke Telegram (Lanjutan)**

Kemudian Atur waktu NodeMCU ESP32 menggunakan RTC agar pakan terbuka secara otomatis pada waktu ditentukan

```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano

}
}
}
}

DateTime now = rtc.now();

// Check if it's time to feed the chickens in the morning (9:00 AM)
if (now.hour() == 8 && now.minute() == 59 && now.second() == 0 && !feedScheduledMorning) {
  feedChickens(now);
  feedScheduledMorning = true;
  previousFeedMillisMorning = currentMillis;
}

// Check if it's time to feed the chickens in the evening (6:00 PM)
if (now.hour() == 17 && now.minute() == 59 && now.second() == 0 && !feedScheduledEvening) {
  feedChickens(now);
  feedScheduledEvening = true;
  previousFeedMillisEvening = currentMillis;
}

// Reset feedScheduled flags after 24 hours
if (currentMillis - previousFeedMillisMorning >= feedInterval) {
  feedScheduledMorning = false;
}
if (currentMillis - previousFeedMillisEvening >= feedInterval) {
  feedScheduledEvening = false;
}

Arduino Nano, ATmega328P on COM5

```

Gambar 4.19 Script Konfigurasi Waktu

```

/

void feedChickens(DateTime now) {
  myServo.write(65);
  delay(1000); // Wait for 5 seconds
  myServo.write(90);

  String timestamp = String(now.year()) + "-" +
                    String(now.month()) + "-" +
                    String(now.day()) + " " +
                    String(now.hour()) + ":" +
                    String(now.minute()) + ":" +
                    String(now.second());

  mySerial.println("FEED:" + timestamp); // Send notification to ESP32 with timestamp
}

void updateLCD() {
  DateTime now = rtc.now();
  digitalWrite(TRIGGER_PIN, LOW); // Trig tidak aktif
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH); // Trig aktif
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  // Membaca sinyal masuk pada echo
  durasi = pulseIn(ECHO_PIN, HIGH);
}

Arduino Nano, ATmega328P on COM5

```

Gambar 4.20 Script Konfigurasi Waktu (Lanjutan)

```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano

}

void feedChickens(DateTime now) {
  myServo.write(65);
  delay(1000); // Wait for 5 seconds
  myServo.write(90);

  String timestamp = String(now.year()) + "-" +
                    String(now.month()) + "-" +
                    String(now.day()) + " " +
                    String(now.hour()) + ":" +
                    String(now.minute()) + ":" +
                    String(now.second());

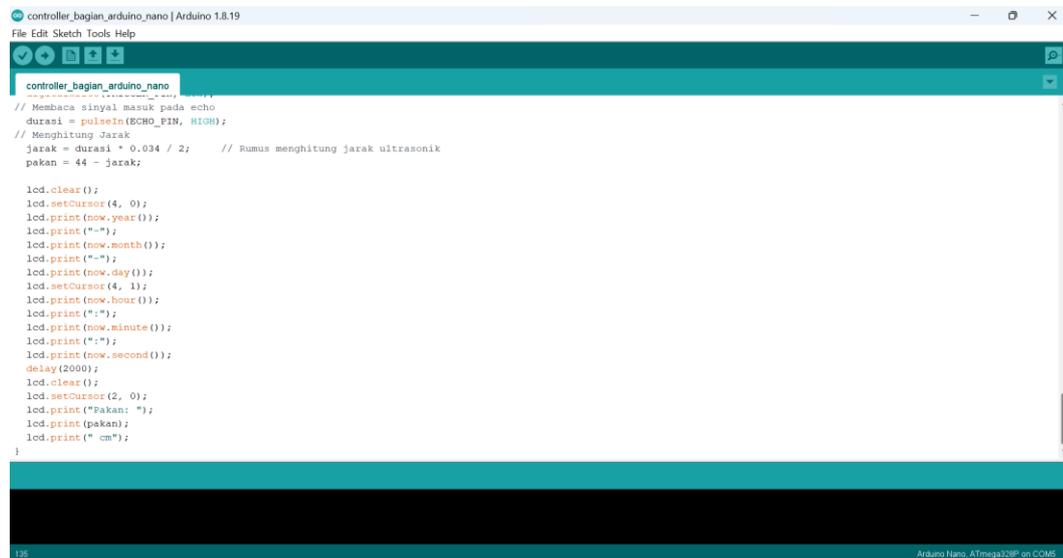
  mySerial.println("FEED:" + timestamp); // Send notification to ESP32 with timestamp
}

void updateLCD() {
  DateTime now = rtc.now();
  digitalWrite(TRIGGER_PIN, LOW); // Trig tidak aktif
  delayMicroseconds(2);
  digitalWrite(TRIGGER_PIN, HIGH); // Trig aktif
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);
  // Membaca sinyal masuk pada echo
  durasi = pulseIn(ECHO_PIN, HIGH);
}

Arduino Nano, ATmega328P on COM5

```

Gambar 4.21 Script Konfigurasi Waktu (Lanjutan)



```

controller_bagian_arduino_nano | Arduino 1.8.19
File Edit Sketch Tools Help

controller_bagian_arduino_nano
// Membaca sinyal masuk pada echo
durasi = pulseIn(ECHO_PIN, HIGH);
// Menghitung Jarak
jarak = durasi * 0.034 / 2; // Rumus menghitung jarak ultrasonik
pakan = 44 - jarak;

lcd.clear();
lcd.setCursor(4, 0);
lcd.print(now.year());
lcd.print("-");
lcd.print(now.month());
lcd.print("-");
lcd.print(now.day());
lcd.setCursor(4, 1);
lcd.print(now.hour());
lcd.print(":");
lcd.print(now.minute());
lcd.print(":");
lcd.print(now.second());
delay(2000);
lcd.clear();
lcd.setCursor(2, 0);
lcd.print("Pakan: ");
lcd.print(pakan);
lcd.print(" cm");
}

```

**Gambar 4.22 Script Konfigurasi Waktu (Lanjutan)**

#### 4.4.1 Pengujian Sistem

Setelah semua sudah diatur, dilakukan pengujian sistem apakah sistem ini berjalan sesuai dengan rencana.

##### 1. LCD

Pengujian ini bertujuan untuk melihat data secara aktual di LCD dan disamakan dengan data di Telegram.



**Gambar 4.23 Menunjukkan Data**

## 2. Motor Servo

Pengujian ini bertujuan untuk melihat apakah motor servo bisa bergerak sesuai diharapkan.

**Tabel 4.1 Motor Servo**

Keterangan	Aksi
/buka 90 (perintah di Telegram)	Motor Servo Bergerak
Memasuki waktu pemberian pakan	Motor Servo Bergerak

## 3. Telegram

Pengujian Telegram ini bertujuan untuk melihat apakah fungsi dalam telegram berfungsi dengan semestinya. Telegram digunakan untuk memonitoring, suhu, kelembaban dan kualitas udara, jika di klik “/status” maka telegram akan menampilkan data-data dari sensor. Kemudian dilengkapi tombol untuk buka atau tutup pakan.



**Gambar 4.24 Tampil Awal Bot**



**Gambar 4.25 Menu Pada Bot**