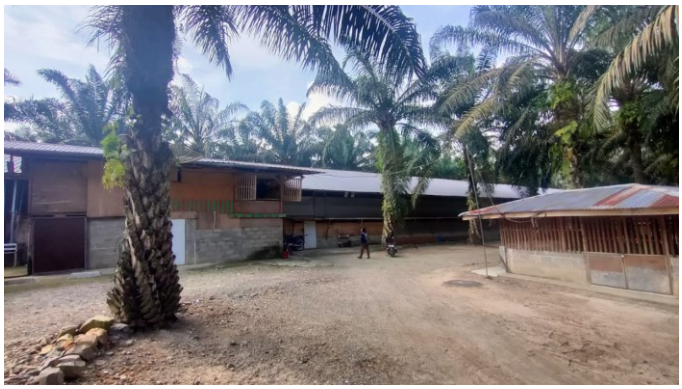
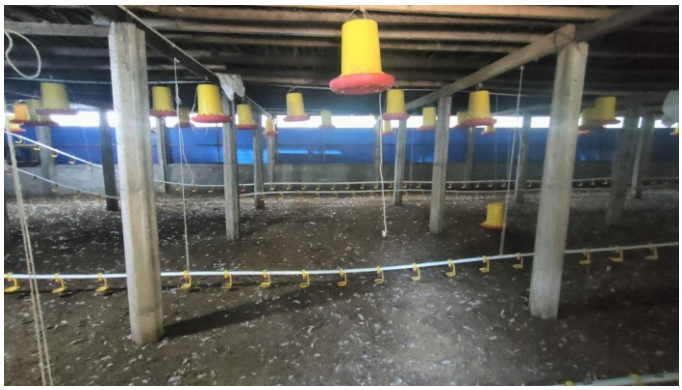
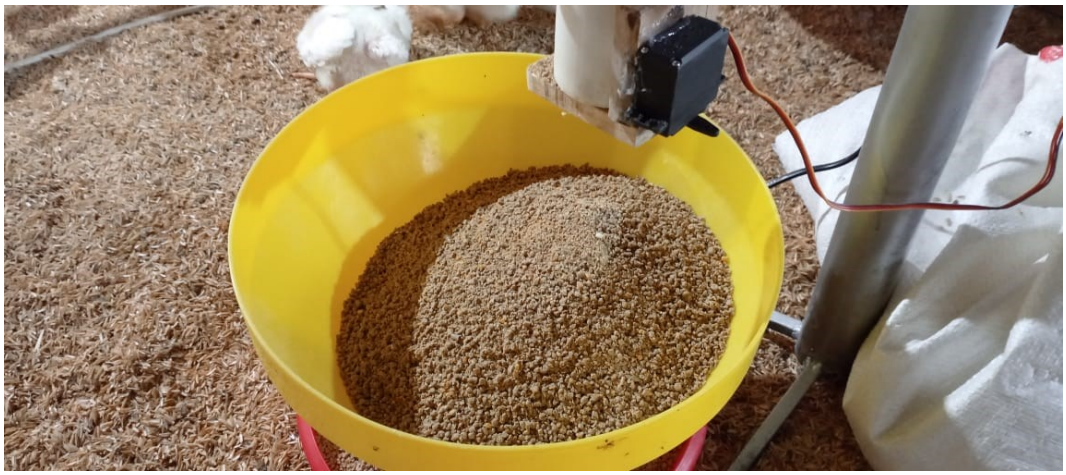
LAMPIRAN FOTO DOKUMENTASI







LAMPIRAN LIST LISTING

Node MCU ESP 32

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

const char* ssid = "TECNO POVA Neo 3";
const char* password = "Azhari27";
const char* botToken =
"7100541764:AAFiFDmgmU6QDyrL17CU09EkOn6kbrgIzWk";

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

String chatId = "5263662125";

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


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) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
  client.setInsecure(); // Skip certificate verification for simplicity
}

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;

      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:")) {
    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;
     }
  }
}
```

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

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

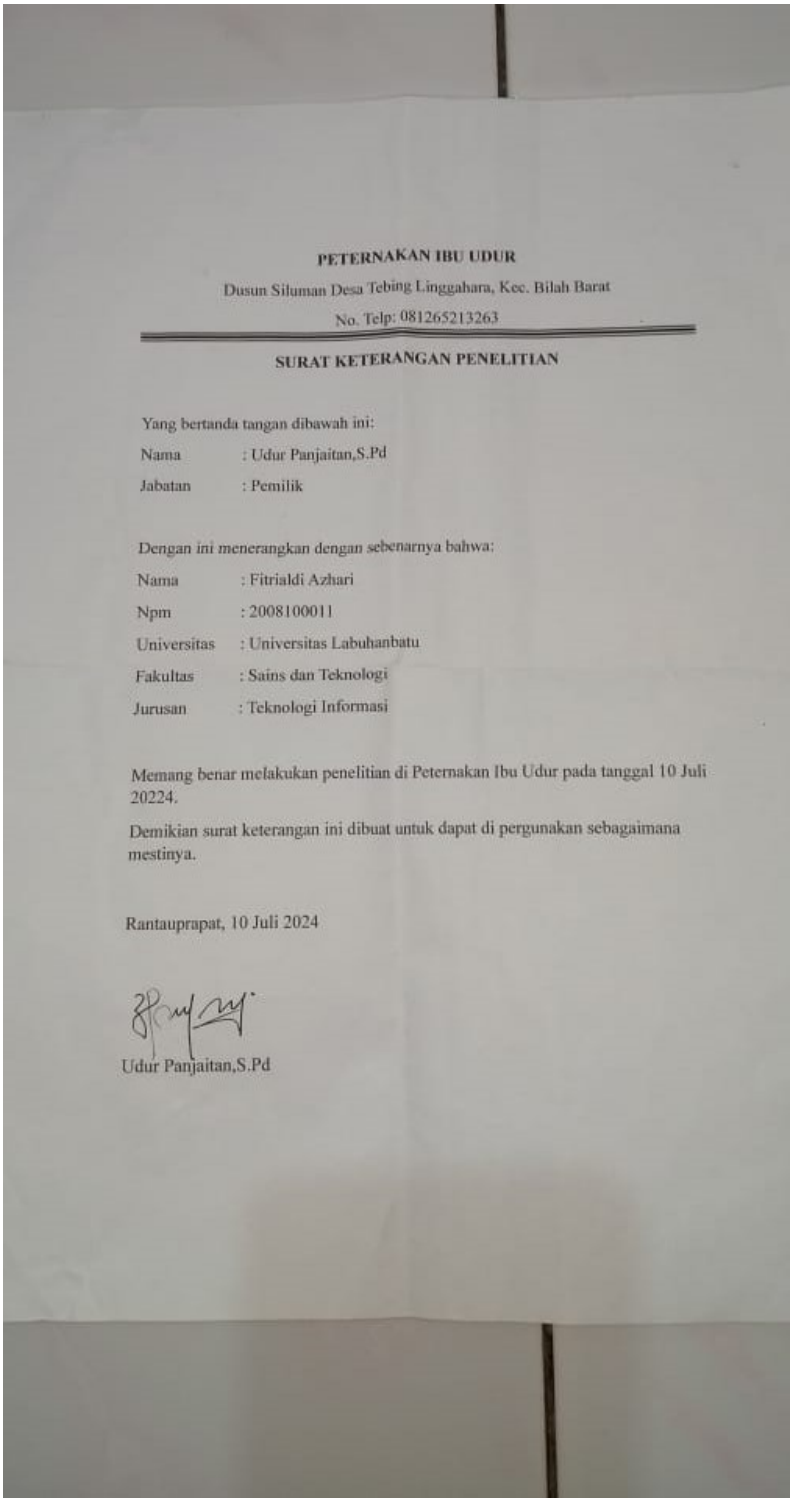
```
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
   pinMode(TRIGGER_PIN, OUTPUT); // deklarasi pin trig sebagai output
   pinMode(ECHO_PIN, INPUT);  // deklarasi pin echo sebagai input
  myServo.attach(9);
  myServo.write(90);  // Set servo to middle position

  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...");
}

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);
        }
      }
    }
  }
 }

 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;
 }
}

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);
}

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);
// 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");
}
```

LAMPIRAN SURAT IZIN PENELITIAN



PETERNAKAN IBU UDUR

Dusun Siluman Desa Tebing Linggahara, Kec. Bilah Barat

No. Telp: 081265213263

SURAT KETERANGAN PENELITIAN

Yang bertanda tangan dibawah ini:

Nama : Udur Panjaitan,S.Pd

Jabatan : Pemilik

Dengan ini menerangkan dengan sebenarnya bahwa:

Nama : Fitrialdi Azhari

Npm : 2008100011

Universitas : Universitas Labuhanbatu

Fakultas : Sains dan Teknologi

Jurusan : Teknologi Informasi

Memang benar melakukan penelitian di Peternakan Ibu Udur pada tanggal 10 Juli 20224.

Demikian surat keterangan ini dibuat untuk dapat di pergunakan sebagaimana mestinya.

Rantauprapat, 10 Juli 2024

Udur Panjaitan,S.Pd