

## DAFTAR LAMPIRAN

### LAMPIRAN A

#### *Listening Program*

1. *Listening Program* Pada Arduino UNO

```
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal_I2C.h>
#include <DS3231.h>
#include <EEPROM.h>
#define ONE_WIRE_BUS 8
#include <Servo.h>

#define trigPin1 6
#define echoPin1 5
#define trigPin2 4
#define echoPin2 3

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

LiquidCrystal_I2C lcd(0x27, 20, 4);
DS3231 rtc(SDA, SCL);
Servo myservo;

int addr1 = 0;
int addr2 = 1;
int addr3 = 2;
int addr4 = 3;

Time t;
int jampagi, jamsiang, jamsore;
float TempC;
```

```
float pHValue;
int adcPH;
int TempCx;
int pHValueX;
long duration1, distance1;
long duration2, distance2;
int btset = 2;
int btup = 11;
int btdown = 10;
int btok = 12;
int btsetx;
int btupx;
int btdownx;
int btokx;
float batasph;
int relayout = A3;
int relayin = A2;
int kolam;
int tinggipakan;

bool sudahPagi = false;
bool sudahSiang = false;
bool sudahMalam = false;

void setup() {

    jumpagi = EEPROM.read(addr1);
    jamsiang = EEPROM.read(addr2);
    jamsore = EEPROM.read(addr3);
    batasph = EEPROM.read(addr4);

    pinMode(relayout, OUTPUT);
    pinMode(relayin, OUTPUT);
    digitalWrite(relayout, LOW);
    digitalWrite(relayin, LOW);

    pinMode(btset, INPUT_PULLUP);
    pinMode(btup, INPUT_PULLUP);
```

```

pinMode(bttdown, INPUT_PULLUP);
pinMode(btok, INPUT_PULLUP);

pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);

Wire.begin();
sensors.begin();
Serial.begin(9600);
lcd.init();
lcd.backlight();
lcd.clear();

rtc.begin();
//rtc.setDOW(WEDNESDAY);      // Set Day-of-Week to
SUNDAmY
  rtc.setTime(14, 59, 30);    // Set the time to 12:00:00 (24hr
format)
  //rtc.setDate(21, 05, 2023); // Set the date to January 1st,
2014

myservo.attach(9);
myservo.write(100);
}

void loop() {

  btsetx = digitalRead(btset);
  btupx = digitalRead(btup);
  bttdownx = digitalRead(bttdown);
  btokx = digitalRead(btok);

  if (btsetx == 0) {
    delay(1000);
    lcd.clear();
    setjampagi();
    setjamsiang();
  }
}

```

```

    setjamsore();

    setbatasph();
}

t = rtc.getTime();

//ketinggian air kolam
digitalWrite(trigPin1, LOW); // Added this line
delayMicroseconds(2); // Added this line
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10); // Added this line
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = (duration1 / 2) / 29.1;

//ketinggian pakan ikan
digitalWrite(trigPin2, LOW); // Added this line
delayMicroseconds(2); // Added this line
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10); // Added this line
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2 = (duration2 * 0.0343) / 2;

if (distance1 > 35) {
    distance1 = 35;
}

if (distance2 > 15) {
    distance2 = 15;
}

kolam = 35 - distance1;
tinggipakan = 15 - distance2;

if ((kolam >= 20) && (kolam <= 20)) {
    digitalWrite(relayout, LOW);
}

```

```

if (kolam > 20) {
    digitalWrite(relayout, HIGH);
}

if (kolam < 20) {
    digitalWrite(relayout, LOW);
}

adcPH = analogRead(A0); //menggunakan pin A0 untuk
membaca output sensor pH
pHValue = (adcPH - 906.84) / -37.283;

pHValuex = pHValue * 100.0;

sensors.requestTemperatures();
TempC = sensors.getTempCByIndex(0); // Celcius
TempCx = TempC * 100.0;

if((t.hour == jampagi)&&(jampagi > 0)&&(tanda == 0)){
    myservo.write(150);
    delay(5000);
    myservo.write(100);
    tanda = 1;
}

if((t.hour == jamsiang)&&(jamsiang > 0)&&(tanda == 1)){
    myservo.write(150);
    delay(5000);
    myservo.write(100);
    tanda = 2;
}

if((t.hour == jamsore)&&(jamsore > 0)&&(tanda == 2)){
    myservo.write(150);
    delay(5000);
    myservo.write(100);
    tanda = 0;
}

```

```
lcd.setCursor(0, 0);  
lcd.print("PH:");  
lcd.print(pHValue);  
lcd.print(" ");  
lcd.print(batasph);  
lcd.print("");
```

```
lcd.setCursor(0, 1);  
lcd.print("Suhu:");  
lcd.print(TempC, 1);  
lcd.print("C ");
```

```
lcd.setCursor(0, 3);  
lcd.print("Kolam=");  
lcd.print(kolam);  
lcd.print("|");  
lcd.print("Pakan=");  
lcd.print(tinggipakan);  
lcd.print(" ");
```

```
lcd.setCursor(0, 2);  
lcd.print(t.hour, DEC);  
lcd.print(":");  
lcd.print(t.min, DEC);  
lcd.print(":");  
lcd.print(t.sec, DEC);  
lcd.print(" ");  
lcd.print(jampagi);  
lcd.print("/");  
lcd.print(jamsiang);  
lcd.print("/");  
lcd.print(jamsore);  
lcd.print(" ");
```

```
Serial.print("*");
```

```

Serial.print(pHValue * 100.0);
Serial.print(",");
Serial.print(TempC * 100.0);
Serial.println("#");
Serial.print(kolam * 100.0);
Serial.print(",");
Serial.print(tinggipakan * 100.0);
Serial.print("#");
delay(1000);
}

```

```

void setjampagi() {

```

```

    btsetx = digitalRead(btset);
    btupx = digitalRead(btup);
    btdownx = digitalRead(btdown);
    btokx = digitalRead(btok);

```

```

    lcd.setCursor(0, 0);
    lcd.print("Waktu Pakan Pagi");
    lcd.setCursor(0, 1);
    lcd.print("Jam: ");
    lcd.print(jampagi);
    lcd.print(" ");

```

```

    if (btupx == 0) {
        delay(200);
        jampagi++;
    }

```

```

    if (btdownx == 0) {
        delay(200);
        jampagi--;
    }

```

```

    if (jampagi > 23) {
        jampagi = 0;
    }

```

```
if (jampagi < 0) {
    jampagi = 23;
}

if (btokx == 0) {
    lcd.clear();
    delay(200);
    EEPROM.write(addr1, jampagi);
    return;
}

setjampagi();
}

void setjamsiang() {

    btsetx = digitalRead(btset);
    btupx = digitalRead(btup);
    btdownx = digitalRead(btdown);
    btokx = digitalRead(btok);

    lcd.setCursor(0, 0);
    lcd.print("Waktu Pakan Siang ");
    lcd.setCursor(0, 1);
    lcd.print("Jam: ");
    lcd.print(jamsiang);
    lcd.print(" ");

    if (btupx == 0) {
        delay(200);
        jamsiang++;
    }

    if (btdownx == 0) {
        delay(200);
        jamsiang--;
    }
}
```



```

if (jamsiang > 23) {
    jamsiang = 0;
}

if (jamsiang < 0) {
    jamsiang = 23;
}

if (btokx == 0) {
    lcd.clear();
    delay(200);
    EEPROM.write(addr2, jamsiang);
    return;
}

setjamsiang();
}

void setjamsore() {

    btsetx = digitalRead(btset);
    btupx = digitalRead(btup);
    btkeydownx = digitalRead(btkeydown);
    btokx = digitalRead(btok);

    lcd.setCursor(0, 0);
    lcd.print("Waktu Pakan Sore ");
    lcd.setCursor(0, 1);
    lcd.print("Jam: ");
    lcd.print(jamsore);
    lcd.print("");

    if (btupx == 0) {
        delay(200);
        jamsore++;
    }

    if (btkeydownx == 0) {

```

```
    delay(200);
    jamsore--;
}

if (jamsore > 23) {
    jamsore = 0;
}
if (jamsore < 0) {
    jamsore = 23;
}

if (btokx == 0) {
    lcd.clear();
    delay(200);
    EEPROM.write(addr3, jamsore);
    return;
}

setjamsore();
}

void setbatasph() {

    btsetx = digitalRead(btset);
    btupx = digitalRead(btup);
    btdownx = digitalRead(btdown);
    btokx = digitalRead(btok);

    lcd.setCursor(0, 0);
    lcd.print("BATAS PH ");
    lcd.setCursor(0, 1);
    lcd.print("PH: ");
    lcd.print(batasph);
    lcd.print(" ");

    if (btupx == 0) {
        delay(200);
        batasph = batasph + 0.1;
    }
}
```

```

    }

    if (btkeydown == 0) {
        delay(200);
        batasph = batasph - 0.1;
    }

    if (batasph > 100.0) {
        batasph = 0;
    }

    if (btokx == 0) {
        lcd.clear();
        delay(200);
        EEPROM.write(addr4, batasph);
        return;
    }

    setbatasph();
}

```

## 2. *Listening* Program ESP8266

```

#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include <WiFiClient.h>

int temp;
int x = 5;
int y;
int Kolam;
int Pakan;
int relayPin = 2;

int value1;
int value2;
int value3;
int value4;

```

```

int ph;
int suhu;
long duration1, distance1;
long duration2, distance2;

float datain1;
float datain2;
int datain3;
int datain4;

String dataIn;
String dt[10];
int i;
boolean parsing = false;
unsigned int durasi, jarak;

String apiKey = "0L5ZF2EXWJGICMS6";    // Enter your
Write API key from ThingSpeak
const char* resource = "/update?api_key=";

const char* ssid = "Redmi";    // replace with your wifi ssid
and wpa2 key
const char* pass = "12341234";
const char* server = "api.thingspeak.com";

WiFiClient client;

#define trigPin1 6
#define echoPin1 5
#define trigPin2 4
#define echoPin2 3

void setup()
{
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);

  dataIn = "";

```

```

Serial.begin(9600);
delay(10);

Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

}

void loop(){
if (std::data == '1' ) {
  digitalWrite(relayPin, HIGH);
  sendDataToThingSpeak(1);
} else if (data == '0') {
  digitalWrite(relayPin, LOW);
  sendDataToThingSpeak(0);} }
{
while (Serial.available() > 0) {
  // dataIn="";
  char inChar = (char)Serial.read();
  dataIn += inChar;
  if (inChar == '\n') {
    parsing = true;

```

```

    }
}

if (parsing) {
    parsingData();

    if (client.connect(server, 80) // "184.106.153.149" or
        api.thingspeak.com
        {

        String postStr = apiKey;
        postStr += "&field1=";
        postStr += String(ph);
        postStr += "\r\n\r\n";

        client.print(String("GET ") + resource + apiKey +
            "&field1=" + datain1 + "&field2=" + datain2 + "&field3=" +
            datain3 + "&field4=" + datain4 + " HTTP/1.1\r\n" + "Host: " +
            server + "\r\n" + "Connection: close\r\n\r\n");

        client.print("Host: api.thingspeak.com\r\n");
        client.print("Connection: close\r\n");
        client.print("X-THINGSPEAKAPIKEY: " + apiKey +
            "\n");
        client.print("Content-Type:      application/x-www-form-
            urlencoded\r\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);

        Serial.println(". Send to Thingspeak.");
    }

    client.stop();

    Serial.println("Waiting...");
}

```

```

    delay(1000 );

}
}

void parsingData() {
    int j = 0;

    //kirim data yang telah diterima sebelumnya
    Serial.print("data masuk : ");
    Serial.print(dataIn);
    Serial.print("\n");

    //inisialisasi variabel, (reset isi variabel)
    dt[j] = "";
    //proses parsing data
    for (i = 1; i < dataIn.length(); i++) {
        //pengecekan tiap karakter dengan karakter (#) dan (, )
        if ((dataIn[i] == '#') || (dataIn[i] == ','))
        {
            //increment variabel j, digunakan untuk merubah index
            array penampung
            j++;
            dt[j] = ""; //inisialisasi variabel array dt[j]
        }
        else
        {
            //proses tampung data saat pengecekan karakter selesai.
            dt[j] = dt[j] + dataIn[i];
        }
    }

    datain1 = dt[0].toFloat();
    datain2 = dt[1].toFloat();
    datain3 = dt[2].toInt();
    datain4 = dt[3].toInt();

    //kirim data hasil parsing
    Serial.print("data 1 : ");

```

```

Serial.print(datain1);
Serial.print("\n");
Serial.print("data 2 : ");
Serial.print(datain2);
Serial.print("\n");
Serial.print("data 3 : ");
Serial.print(datain3);
Serial.print("\n");
Serial.print("data 4 : ");
Serial.print(datain4);
Serial.print("\n");

ph = datain1 / 100.0;
suhu = datain2 / 100.0;
Kolam = datain3 / 100.0;
Pakan = datain4 / 100.0;
}

```

### 3. *Listening* Program Python Raspberry 3B

```

import cv2
import telebot
import numpy as np

# Inisialisasi bot Telegram
bot_token = 'YOUR_BOT_TOKEN'
bot_chat_id = 'YOUR_CHAT_ID'
bot = telebot.TeleBot(bot_token)

# Fungsi untuk mengirim pesan ke bot Telegram
def send_message_to_telegram(message):
    bot.send_message(chat_id=bot_chat_id, text=message)

# Fungsi untuk mengirim gambar ke bot Telegram
def send_image_to_telegram(image_path):
    with open(image_path, 'rb') as image:
        bot.send_photo(chat_id=bot_chat_id, photo=image)

# Fungsi untuk mendeteksi daun menguning pada gambar

```



```

def detect_yellow_leaf(image):
    # Konversi gambar ke ruang warna HSV
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # Definisikan rentang warna kuning
    lower_yellow = np.array([20, 100, 100])
    upper_yellow = np.array([30, 255, 255])

    # Segmentasi gambar berdasarkan warna kuning
    mask = cv2.inRange(hsv, lower_yellow, upper_yellow)

    # Morphological operation (opening) untuk menghilangkan
    noise
    kernel = np.ones((5, 5), np.uint8)
    opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN,
    kernel)

    # Temukan kontur daun kuning
    contours, _ = cv2.findContours(opening,
    cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    # Gambar kotak bounding pada daun kuning yang terdeteksi dan
    beri komentar
    for contour in contours:
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
        cv2.putText(image, 'Daun tanaman menguning', (x, y - 10),
        cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

    return image, len(contours) > 0

# Fungsi untuk mengambil gambar menggunakan webcam
def capture_image():
    # Konfigurasi webcam
    cap = cv2.VideoCapture(0)

    # Ambil frame dari webcam
    ret, frame = cap.read()

```

```
if ret:
    # Deteksi daun menguning pada gambar
    image_with_leaf, is_yellow_leaf =
detect_yellow_leaf(frame)

    # Jika terdeteksi daun menguning, kirim gambar ke bot
Telegram
    if is_yellow_leaf:
        # Simpan gambar ke file
        image_path = 'captured_image.jpg'
        cv2.imwrite(image_path, image_with_leaf)

        # Mengirim gambar ke bot Telegram
        send_image_to_telegram(image_path)
        send_message_to_telegram('Daun tanaman
menguning.')

    # Tutup koneksi webcam
    cap.release()

# Loop utama untuk mengambil dan mengirim gambar hanya
ketika mendeteksi daun menguning
while True:
    capture_image()
```