

LAMPIRAN

Lampiran 1. Turnitin

	UNIVERSITAS DARMA PERSADA UPT PERPUSTAKAAN Gedung Rektorat Lantai 3, Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450
SURAT KETERANGAN HASIL PENGECEKAN TURNITIN	
UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/ <i>similarity</i> menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:	
Judul	: SISTEM KONTROL PAKAN DAN LINGKUNGAN KANDANG BURUNG AVIARY BERBASIS INTERNET OF THINGS PLATFORM BLYNK
Penulis	: Ilham Satrio
NIM	: 2019230049
Tgl pemeriksaan	: 25 Juli 2025
Dengan hasil Tingkat Kesamaan (<i>similarity index</i>) 18%	
Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.	
Jakarta, 25 Juli 2025 Ka.UPT Perpustakaan Unsada	
 Yus Rusmiyati, SS., MM	
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"><p>Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi</p><p>Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana</p></div>	

Lampiran 2. Kode Program

```
#define BLYNK_TEMPLATE_ID "TMPL6ys8Ey3jG"

#define BLYNK_TEMPLATE_NAME "Aviary IoT"

#define BLYNK_AUTH_TOKEN "LaRTSD-oEZMsbizMXQ0PK2elcs_QHltq"

// Masukkan library yang dibutuhkan

#include <WiFi.h>

#include <BlynkSimpleEsp32.h>

#include <DHT.h>

#include <ESP32Servo.h> // Library Servo

#include <SPI.h>

#include <MFRC522.h> // Library RFID RC522

// Konfigurasi DHT22

#define DHTPIN 27 // Pin data DHT22

#define DHTTYPE DHT22 // Jenis sensor DHT22

DHT dht(DHTPIN, DHTTYPE);

// Konfigurasi Relay

#define RELAY1 15

#define RELAY2 2

#define RELAY3 4 #define

RELAY4 5
```

```

// Konfigurasi Servo

#define SERVO_PIN 33          // Pin Servo SG90

#define MG995_PIN 32         // Pin Servo MG995

#define SERVO360_PIN 14     // Pin Servo SG90 360 derajat

Servo myServo;              // Objek Servo SG90

Servo mg995Servo;          // Objek Servo MG995

Servo servo360;            // Objek Servo SG90 360 derajat

// Konfigurasi Sensor Raindrop

#define RAINDROP_SENSOR_PIN 34 // Pin analog untuk sensor raindrop

// Konfigurasi Sensor Ultrasonik HC-SR04

#define TRIG_PIN 25          // Pin TRIG HC-SR04 #define
ECHO_PIN 26                 // Pin ECHO HC-SR04

// Konfigurasi RFID RC522

#define RST_PIN 22          // Pin RST untuk RFID #define
SDA_PIN 21                 // Pin SDA untuk RFID

MFRC522 rfid(SDA_PIN, RST_PIN);

// Masukkan kredensial WiFi char ssid[] =
"SKRIPSII"; // Nama WiFi char pass[] =
"12345678"; // Password WiFi

```

```
// Fungsi untuk membaca suhu dan kelembaban, lalu mengirimkannya ke
Blynk void sendSensorData() { float suhu = dht.readTemperature();
// Membaca suhu float kelembaban = dht.readHumidity(); // Membaca
kelembaban

    if (isnan(suhu) || isnan(kelembaban))
{
    Serial.println("Gagal membaca data dari sensor DHT22!");
return;
}

// Kirim data ke Blynk
Blynk.virtualWrite(V1, suhu); // Kirim suhu ke Virtual Pin
V1
Blynk.virtualWrite(V0, kelembaban); // Kirim kelembaban ke
Virtual
Pin V0

// Tampilkan data di Serial Monitor
Serial.print("Suhu: ");
Serial.print(suhu);
Serial.print(" \u00b0C, Kelembaban: ");
Serial.print(kelembaban);
Serial.println(" %");
}
```

```

// Fungsi untuk membaca data dari sensor raindrop void
sendRaindropData() {
    int nilaiRaindrop =
    analogRead(RAINDROP_SENSOR_PIN); // Baca nilai analog sensor raindrop
    float persentaseKering = map(nilaiRaindrop, 0, 4095, 0, 100); //
    Konversi ke persentase (0% = basah, 100% = kering)

    // Kirim data ke Blynk
    Blynk.virtualWrite(V7, persentaseKering); // Kirim data ke Virtual
    Pin V7

    // Tampilkan data di Serial Monitor
    Serial.print("Sensor Raindrop: ");
    Serial.print(nilaiRaindrop);
    Serial.print(" (Kering: ");
    Serial.print(persentaseKering);
    Serial.println("%)");
}

// Fungsi untuk membaca data dari sensor
ultrasonik void sendUltrasonicData() {
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2); digitalWrite(TRIGPIN,
    HIGH); delayMicroseconds(10);

```

```

digitalWrite(TRIGPIN, LOW);

    long duration = pulseIn(ECHOPIN, HIGH); // Hitung waktu pulsa HIGH
pada pin ECHO    float distance = (duration * 0.034) / 2; // Konversi
ke jarak dalam cm

// Kirim data ke Blynk
Blynk.virtualWrite(V8, distance); // Kirim jarak ke Virtual Pin V8

// Tampilkan data di Serial Monitor
Serial.print("Jarak Ultrasonik: ");
Serial.print(distance);
Serial.println(" cm");
}

// Fungsi untuk membaca kartu RFID void readRFID() {    if
(!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) {
return; // Tidak ada kartu baru yang terdeteksi
}

// Baca UID kartu
String uid = ""; for (byte i = 0; i <
rfid.uid.size; i++) {

```

```

    uid += String(rfid.uid.uidByte[i], HEX);
}

// Kirim data UID ke Blynk (Virtual Pin V9)
Blynk.virtualWrite(V9, uid);

// Tampilkan UID di Serial Monitor
Serial.print("UID Kartu RFID: ");
Serial.println(uid);

// Matikan pembacaan kartu
rfid.PICC_HaltA();
}

// Fungsi untuk mengontrol relay BLYNK_WRITE(V2) {
digitalWrite(RELAY1, param.asInt()); // Kontrol Relay 1 melalui
Virtual Pin V2
}
BLYNK_WRITE(V3) { digitalWrite(RELAY2, param.asInt()); // Kontrol
Relay 2 melalui
Virtual Pin V3
}
BLYNK_WRITE(V4) {

```

```

        digitalWrite(RELAY3, param.asInt()); // Kontrol Relay 3 melalui
Virtual Pin V4
    }
    BLYNK_WRITE(V5) {    digitalWrite(RELAY4, param.asInt()); // Kontrol
Relay 4 melalui
Virtual Pin V5
    }

// Fungsi untuk mengontrol Servo SG90 dengan tombol switch di V6
BLYNK_WRITE(V6) {    int switchState = param.asInt(); // Membaca
status tombol (0: OFF,
1: ON)    if (switchState == 1) {        myServo.write(90);
// Geser servo ke posisi 90 derajat
        Serial.println("Servo SG90 digerakkan ke 90 derajat (ON)");
    } else {        myServo.write(0); // Geser servo ke posisi 0
derajat saat OFF
        Serial.println("Servo SG90 kembali ke 0 derajat (OFF)");
    }
}

BLYNK_WRITE(V10) {    int switchState = param.asInt(); // Membaca
status tombol (0: OFF,
1: ON) if (switchState
== 1) {

```

```

        Serial.println("Servo MG995 berputar searah jarum jam 2 kali
(ON)");    for (int i = 0; i < 2; i++) {        mg995Servo.write(180);
// Sinyal untuk rotasi searah jarum jam        delay(2000);        //
Putar        selama        2        detik        (atur        sesuai        kebutuhan)
mg995Servo.write(90); // Berhenti (sinyal netral)        delay(500);
    }
} else {
    Serial.println("Servo MG995 berputar berlawanan arah jarum jam 2
kali (OFF)");    for (int i = 0; i < 2; i++) {
mg995Servo.write(0); // Sinyal untuk rotasi berlawanan arah jarum
jam        delay(2000);        // Putar selama 2 detik (atur sesuai
kebutuhan)        mg995Servo.write(90); // Berhenti (sinyal netral)
delay(500);
    }
}
}

// Fungsi untuk mengontrol Servo SG90 360 derajat dengan tombol
switch di V11
BLYNK_WRITE(V11) {
    int switchState = param.asInt(); // Membaca status tombol (0: OFF,
1: ON)    if (switchState == 1) {        servo360.attach(SERVO360PIN);
// Sambungkan servo        servo360.write(90);        // Sinyal

```

```

untuk memutar servo      Serial.println("Servo SG90 360 derajat
bergerak selama 5 detik");      delay(5000);          //
Gerakkan servo selama 5 detik      servo360.write(180);          //
Berhentikan servo      delay(5000);          // Tunggu untuk
berhenti sempurna      servo360.detach();          // Lepas sinyal
servo
        Serial.println("Servo SG90 360 derajat berhenti");
    }
} void
setup() {

    // Inisialisasi Serial Monitor
    Serial.begin(115200);

    // Inisialisasi DHT22
    dht.begin();

    // Inisialisasi Relay sebagai output
    pinMode(RELAY1, OUTPUT); pinMode(RELAY2,
OUTPUT); pinMode(RELAY3, OUTPUT);
    pinMode(RELAY4, OUTPUT);

```

```

// Pastikan semua relay dalam keadaan OFF di
awal  digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY2, HIGH);  digitalWrite(RELAY3,
HIGH);  digitalWrite(RELAY4, HIGH);

// Inisialisasi Servo  myServo.attach(SERVOPIN); // Hubungkan
servo SG90 ke pin yang ditentukan  myServo.write(90); //
Posisi awal servo SG90 pada 90 derajat  mg995Servo.attach(MG995PIN);
// Hubungkan servo MG995 ke pin yang ditentukan  delay(1000);
// Tunggu 1 detik untuk memastikan servo siap  mg995Servo.write(90);
// Posisi awal servo MG995 pada 90 derajat (netral)  delay(2000);
// Tunggu servo mencapai posisi  mg995Servo.detach(); //
Lepas  sinyal  PWM  agar  servo  tidak  bergerak
mg995Servo.attach(MG995PIN); // Sambungkan kembali saat diperlukan

// Inisialisasi Ultrasonik

```

```
pinMode(TRIGPIN, OUTPUT);
pinMode(ECHOPIN, INPUT);

// Inisialisasi RFID
SPI.begin();           // Inisialisasi SPI
rfid.PCD_Init();      // Inisialisasi modul RFID

// Hubungkan ke WiFi dan Blynk
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
} void
loop() {
  Blynk.run();         // Jalankan fungsi Blynk
  sendSensorData();    // Kirim data suhu dan kelembaban ke Blynk
  sendRaindropData(); // Kirim data raindrop ke Blynk
  sendUltrasonicData(); // Kirim data ultrasonik ke Blynk
  readRFID();          // Baca data kartu RFID delay(2000);
// Tunggu 2 detik sebelum membaca data lagi }
```