LAMPIRAN

Lampiran 1

Surat Keterangan Plagiat

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/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : PERANCANGAN SISTEM PENYIRAMAN DAN PEMELIHARAAN TANAMAN ANGGREK CATTLEYA BERBASIS INTERNET OF THINGS (IOT)

Penulis : Muchamad Hifdi
NIM : **2021230020**
Tgl pemeriksaan : 25 Juli 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **14%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 22 Juli 2025

Ka.UPT Perpustakaan Unsada

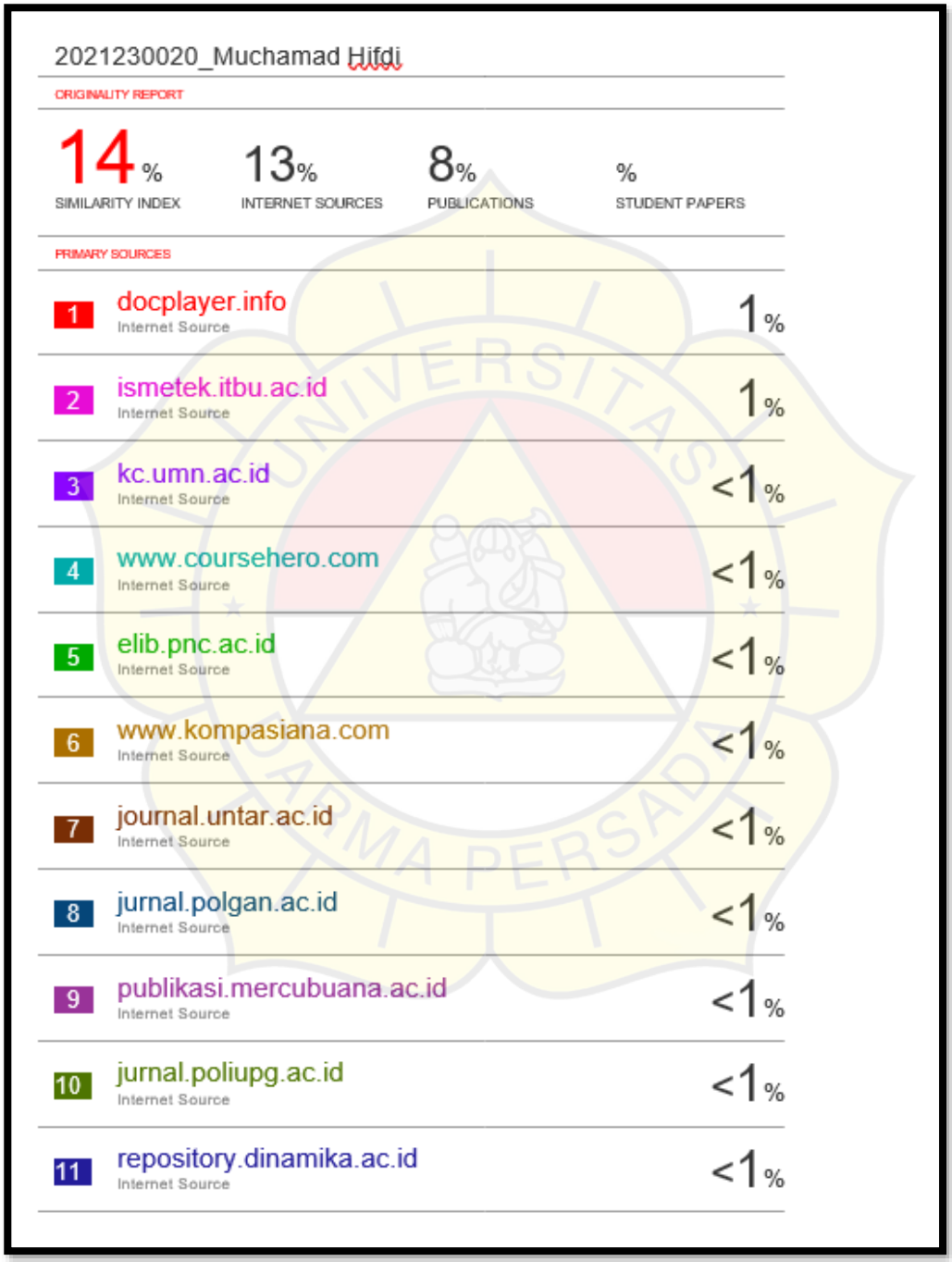
Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana

Lampiran 2

Hasil Turnitin



2021230020_Muchamad Hifdi

ORIGINALITY REPORT

14%	13%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.info Internet Source	1%
2	ismetek.itbu.ac.id Internet Source	1%
3	kc.umn.ac.id Internet Source	<1%
4	www.coursehero.com Internet Source	<1%
5	elib.pnc.ac.id Internet Source	<1%
6	www.kompasiana.com Internet Source	<1%
7	journal.untar.ac.id Internet Source	<1%
8	jurnal.polgan.ac.id Internet Source	<1%
9	publikasi.mercubuana.ac.id Internet Source	<1%
10	jurnal.poliupg.ac.id Internet Source	<1%
11	repository.dinamika.ac.id Internet Source	<1%

Lampiran 2

Source Code Arduino

```
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
#include <HTTPClient.h>

// ============ Konfigurasi Firebase ============
#define          FIREBASE_HOST          "testing-d3365-default-
  rtdb.firebaseio.com"
#define                                  FIREBASE_AUTH
"fhrWy5rvPqYR7ChZMijlc8dtki6tChCTSAAnMGEA"

// ============ WiFi ============
char ssid[] = "MH";
char pass[] = "11112222";

// ============ Firebase Objects ============
FirebaseData firebaseData;
FirebaseAuth auth;
FirebaseConfig config;

// ============ Pin dan Sensor ============
#define DHTPIN 5
#define DHTTYPE DHT11
#define RAIN_SENSOR_PIN 32
#define SOIL_PIN A6

#define RELAY_POMPA 25
#define RELAY_KIPAS 26
#define RELAY_BUZZER 27
#define SERVO_PIN 14

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo myServo;

// ============ Kalibrasi ============
const int AirValue = 2620;
```

```
const int WaterValue = 1180;

// ============ Variabel ============
int soilMoistureValue = 0;
int soilmoist = 0;
int temp = 0;
int sistem = 1;
int rainValue = 0;
bool isRaining = false;
int rainCount = 0;
const int rainDetectionCount = 3;

bool buzzerActive = false;
bool buzzerState = false;
bool kipasState = false;
String servoStatus = "Terbuka";

bool manualPump = false;
bool manualFan = false;
bool manualBuzzer = false;
bool manualServo = false;

// ============ Threshold dari Firebase ============
int thresholdSoil = 50;
int thresholdTemp = 33;
int thresholdRain = 3600;

// ============ Timer Kirim MySQL ============
unsigned long lastSendTime = 0;
const unsigned long sendInterval = 1800000;

// ============ Timer LCD ============
unsigned long lastLCDUpdate = 0;
const unsigned long lcdUpdateInterval = 2000;

void setup() {
  Serial.begin(115200);
  lcd.begin(16, 2);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("= WELCOME =");
  lcd.setCursor(0, 1);
  lcd.print(" SMART ORCHID ");
```

```
  pinMode(RAIN_SENSOR_PIN, INPUT);
  pinMode(SOIL_PIN, INPUT);
  pinMode(RELAY_POMPA, OUTPUT);
  pinMode(RELAY_KIPAS, OUTPUT);
  pinMode(RELAY_BUZZER, OUTPUT);

  digitalWrite(RELAY_POMPA, HIGH);
  digitalWrite(RELAY_KIPAS, HIGH);
  digitalWrite(RELAY_BUZZER, HIGH);

  dht.begin();
  myServo.attach(SERVO_PIN);
  myServo.write(0);

  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected");

  config.host = FIREBASE_HOST;
  config.signer.tokens.legacy_token = FIREBASE_AUTH;
  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);

  initThresholdInFirebase();
  delay(2000);
}

void loop() {
  readThresholdFromFirebase();
  read_SoilMoist();
  read_DHT11();
  read_RainSensor();
  checkFirebaseControl();
  kontrolAktuator();
  updateFirebase();

  if (millis() - lastSendTime >= sendInterval) {
    kirimKeMySQL();
    lastSendTime = millis();
  }

  if (millis() - lastLCDUpdate >= lcdUpdateInterval) {
```

```
    lcd.setCursor(0, 0);
    lcd.print("Mst:");
    lcd.print(soilmoist);
    lcd.print("%   ");
    lcd.setCursor(10, 0);
    lcd.print("T:");
    lcd.print(temp);
    lcd.print("C  ");
    lcd.setCursor(0, 1);
    lcd.print("Hujan: ");
    lcd.print(isRaining ? "Ya " : "Tidak");
    lcd.print("   ");
    lastLCDUpdate = millis();
  }
}

void initThresholdInFirebase() {
                if                (!Firebase.get(firebaseData,
"/smart_orchid/threshold/soil_moisture")) {
                                  Firebase.setInt(firebaseData
, "/smart_orchid/threshold/soil_moisture", thresholdSoil);
  }
                if                (!Firebase.get(firebaseData,
"/smart_orchid/threshold/temperature")) {
                                  Firebase.setInt(firebaseData
, "/smart_orchid/threshold/temperature", thresholdTemp);
  }
                if                (!Firebase.get(firebaseData,
"/smart_orchid/threshold/rain_value")) {
                                  Firebase.setInt(firebaseData
, "/smart_orchid/threshold/rain_value", thresholdRain);
  }
}

void read_SoilMoist() {
  soilMoistureValue = analogRead(SOIL_PIN);
  soilmoist = map(soilMoistureValue, AirValue, WaterValue, 0,
100);
  soilmoist = constrain(soilmoist, 0, 100);
  Serial.print("Soil Moisture: ");
  Serial.print(soilmoist);
  Serial.println("%");
}

void read_DHT11() {
```

```
  float t = dht.readTemperature();
  if (!isnan(t)) {
    temp = t;
    Serial.print("Temperature: ");
    Serial.println(temp);
  } else {
    Serial.println("Failed to read temperature");
  }
}

void read_RainSensor() {
  rainValue = analogRead(RAIN_SENSOR_PIN);
  Serial.print("Rain Sensor Value: ");
  Serial.println(rainValue);
  if (rainValue < thresholdRain) {
    rainCount++;
  } else {
    rainCount = 0;
  }
  isRaining = (rainCount >= rainDetectionCount);
   Serial.println("Status: " + String(isRaining ? "Hujan" :
"Tidak Hujan"));
}

void readThresholdFromFirebase() {
                if              (Firebase.getInt(firebaseData,
"/smart_orchid/threshold/soil_moisture")) {
    thresholdSoil = firebaseData.intData();
  }
                if              (Firebase.getInt(firebaseData,
"/smart_orchid/threshold/temperature")) {
    thresholdTemp = firebaseData.intData();
  }
                if              (Firebase.getInt(firebaseData,
"/smart_orchid/threshold/rain_value")) {
    thresholdRain = firebaseData.intData();
  }
}

void checkFirebaseControl() {
   if (Firebase.getString(firebaseData, "/smart_orchid/mode"))
{
    String mode = firebaseData.stringData();
    sistem = (mode == "auto") ? 1 : 0;
  }
```

```
  if (sistem == 0) {
                                  Firebase.getBool(firebaseData,
"/smart_orchid/manual_control/pump")     ?     manualPump     =
firebaseData.boolData() : false;
                                  Firebase.getBool(firebaseData
,    "/smart_orchid/manual_control/fan")    ?    manualFan    =
firebaseData.boolData() : false;
                                  Firebase.getBool(firebaseData
,   "/smart_orchid/manual_control/buzzer")   ?   manualBuzzer   =
firebaseData.boolData() : false;
                                  Firebase.getBool(firebaseData
,   "/smart_orchid/manual_control/servo")   ?   manualServo   =
firebaseData.boolData() : false;
  }
}

void kontrolAktuator() {
  static unsigned long pumpStartTime = 0;
  static unsigned long pumpCooldownStart = 0;
  static bool pumpOn = false;
  static bool pumpCooldown = false;
  static unsigned long buzzerStartTime = 0;
  unsigned long currentMillis = millis();

  if (sistem == 1) {
    if (pumpOn && (currentMillis - pumpStartTime >= 3000)) {
      digitalWrite(RELAY_POMPA, HIGH);
      pumpOn = false;
      pumpCooldown = true;
      pumpCooldownStart = currentMillis;
    }
     if (pumpCooldown && (currentMillis - pumpCooldownStart >=
60000)) {
      pumpCooldown = false;
    }
    if (!pumpOn && !pumpCooldown && soilmoist < thresholdSoil)
{
      digitalWrite(RELAY_POMPA, LOW);
      pumpStartTime = currentMillis;
      pumpOn = true;
    }
    if (!pumpOn && soilmoist >= thresholdSoil) {
      digitalWrite(RELAY_POMPA, HIGH);
      pumpCooldown = false;
```

```
    }

    if ((soilmoist < thresholdSoil || temp > thresholdTemp) &&
!buzzerActive) {
      digitalWrite(RELAY_BUZZER, LOW);
      buzzerStartTime = currentMillis;
      buzzerActive = true;
      buzzerState = true;
    }
     if (buzzerActive && (currentMillis - buzzerStartTime >=
5000)) {
      digitalWrite(RELAY_BUZZER, HIGH);
      buzzerActive = false;
      buzzerState = false;
    }

     digitalWrite(RELAY_KIPAS, temp > thresholdTemp ? LOW :
HIGH);
    kipasState = (temp > thresholdTemp);

    if (isRaining) {
      myServo.write(90);
      servoStatus = "Tertutup";
    } else {
      myServo.write(0);
      servoStatus = "Terbuka";
    }

  } else {
    digitalWrite(RELAY_POMPA, manualPump ? LOW : HIGH);
    digitalWrite(RELAY_KIPAS, manualFan ? LOW : HIGH);
    digitalWrite(RELAY_BUZZER, manualBuzzer ? LOW : HIGH);
    kipasState = manualFan;
    buzzerState = manualBuzzer;

    if (manualServo) {
      myServo.write(90);
      servoStatus = "Tertutup";
    } else {
      myServo.write(0);
      servoStatus = "Terbuka";
    }
  }
}
```

```
void updateFirebase() {
  String basePath = "/smart_orchid/data/";

  Serial.println("=== Mengirim Data ke Firebase ===");
  Serial.println("Temperature: " + String(temp));
  Serial.println("Soil Moisture: " + String(soilmoist));
  Serial.println("Rain Status: " + String(isRaining ? "Hujan"
: "Tidak Hujan"));

   Firebase.setInt(firebaseData, basePath + "temperature",
temp);
   Firebase.setInt(firebaseData, basePath + "soil_moisture",
soilmoist);
   Firebase.setString(firebaseData, basePath + "rain_status",
isRaining ? "Hujan" : "Tidak Hujan");

   Firebase.setString(firebaseData, basePath + "pump_status",
digitalRead(RELAY_POMPA) == LOW ? "ON" : "OFF");
  Firebase.setString(firebaseData, basePath + "status_buzzer",
digitalRead(RELAY_BUZZER) == LOW ? "ON" : "OFF");
  Firebase.setString(firebaseData, basePath + "status_kipas",
digitalRead(RELAY_KIPAS) == LOW ? "ON" : "OFF");
  Firebase.setString(firebaseData, basePath + "status_servo",
servoStatus);

   Firebase.setString(firebaseData, basePath + "wifi_status",
WiFi.status() == WL_CONNECTED ? "Koneksi OK" : "Tidak
Terhubung");
   Firebase.setString(firebaseData, basePath + "wifi_ssid",
WiFi.SSID());
}

void kirimKeMySQL() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
                http.begin("http://testcardeki.my.id/api_smart-
orchid/insert_data.php");
      http.addHeader("Content-Type", "application/x-www-form-
urlencoded");

    String postData =
      "suhu=" + String(temp) +
      "&kelembapan=" + String(soilmoist) +
       "&hujan=" + String(isRaining ? "Hujan" : "Tidak Hujan")
+
```

```
        "&pompa_status=" + String(digitalRead(RELAY_POMPA) ==
LOW ? "ON" : "OFF") +
        "&kipas_status=" + String(digitalRead(RELAY_KIPAS) ==
LOW ? "ON" : "OFF") +
       "&buzzer_status=" + String(digitalRead(RELAY_BUZZER) ==
LOW ? "ON" : "OFF") +
       "&servo_status=" + servoStatus;

    int httpResponseCode = http.POST(postData);
    if (httpResponseCode > 0) {
      String response = http.getString();
      Serial.print("Response dari server: ");
      Serial.println(response);
    } else {
      Serial.print("Error mengirim data ke server. Kode: ");
      Serial.println(httpResponseCode);
    }
    http.end();
  } else {
    Serial.println("WiFi tidak terhubung, tidak dapat mengirim
ke MySQL");
  }
}
```