

LAMPIRAN

Lampiran 1 Surat Kesiediaan Pembimbing

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini

Nama : Eko Budihartono, S.T., M. Kom.
NIDN : 0605037304
NIPY : 12.013.170
Jabatan Struktural : Sub Pelatihan dan Sertifikasi
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Nur Wahida Setiawati
NIM : 22041061
Program Studi : Diploma III Teknik Komputer

Judul TA : RANCANG BANGUN ALAT PENYIRAMAN DAN
PEMBERIAN NUTRISI OTOMATIS PADA TANAMAN
MELATI BERBASIS *INTERNET OF THINGS*

Dengan ini menyatakan agar dilaksanakan sebagaimana mestinya.

Tegal, April 2025

Mengetahui
Ketua Program Studi
DIII Teknik Komputer



Ida Afriliana, S.T., M. Kom.
NIPY. 12.013.168

Dosen Pembimbing I

Eko Budihartono, S.T., M. Kom.
NIPY. 12.013.170

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini

Nama : Arfan Haqiqi Sulasmoro, M. Kom.
NIDN : 0623037704
NIPY : 02.009.054
Jabatan Struktural : -
Jabatan Fungsional : Lektor

Dengan ini menyatakan bersedia untuk menjadi pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Nur Wahida Setiawati
NIM : 22041061
Program Studi : Diploma III Teknik Komputer

Judul TA : RANCANG BANGUN ALAT PENYIRAMAN DAN
PEMBERIAN NUTRISI OTOMATIS PADA TANAMAN
MELATI BERBASIS *INTERNET OF THINGS*

Dengan ini menyatakan agar dilaksanakan sebagaimana mestinya.

Tegal, April 2025

Mengetahui
Ketua Program Studi
DIII Teknik Komputer



Ida Airliana, S.T., M. Kom.
NIPY. 12.013.168

Dosen Pembimbing II

Arfan Haqiqi Sulasmoro, M. Kom.
NIPY. 02.009.054

Lampiran 2 Surat Izin Observasi



POLITEKNIK HARAPAN BERSAMA
The Best Vocational College

D-3 Teknik Komputer

Nomor : 245.03/KOM-PHB/III/2025
Lampiran : -
Perihal : Permohonan Izin Observasi Tugas Akhir (TA)

Kepada Yth.

Pimpinan CV. Harum Melati Jaya

Di Desa Maribaya Kec. Tarub Kab. Tegal

Dengan Hormat,

Sehubungan dengan tugas mata kuliah Tugas Akhir (TA) yang akan diselenggarakan di semester V Program Studi Diploma III Teknik Komputer Politeknik Harapan Bersama Tegal, Maka dengan ini kami mengajukan izin observasi pengambilan data di CV. Harum Melati Jaya yang Bapak / Ibu Pimpinan, untuk kepentingan dalam pembuatan produk Tugas Akhir, dengan Mahasiswa sebagai berikut:

No	NIM	NAMA	NO HP
1	22041061	NUR WAHIDA SETIAWATI	081225098092
2	22041077	AUFA NAWAL KUMARA ADI	088226368581

Demikian surat permohonan ini kami sampaikan atas izin dan kerjasamanya kami sampaikan terima kasih.

Tegal, 25 April 2025
Ka. Prodi DIII Teknik Komputer
Politeknik Harapan Bersama Tegal



Ida Afrilliana, ST, M.Kom
NIP. 12.013.168

Lampiran 3 Dokumentasi Observasi





Lampiran 4 Script Code Arduino IDE

```
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Sensor.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <DHT.h>
#include <EEPROM.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiUdp.h>
#include <NTPClient.h>

// Inisialisasi
LiquidCrystal_I2C lcd(0x27, 16, 2);

DHT dht(D4, DHT22);
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 25200, 60000); //
GMT+7 (Asia/Jakarta)

// Pin
#define RELAY_SIRAM D6
#define RELAY_NUTRISI D7
#define SOIL_MOISTURE_PIN A0
#define LED_MERAH D0
#define LED_KUNING D3
#define LED_HIJAU D8
#define LDR_PIN D5

// WiFi
const char* ssid = "vivo";
const char* password = "1234567890";
// API
const char* urlApiSuhu =
"http://ta2025.my.id/api/dashboard/insert/";
const char* urlApiPenyiramanUpdate =
"http://ta2025.my.id/api/penyiraman/update";
const char* urlApiPenyiramanStatus =
"http://ta2025.my.id/api/penyiraman/status";
const char* urlApiPenutrisianStatus =
"http://ta2025.my.id/api/penutrisian/status";
const char* urlApiPenutrisianJadwal =
"http://ta2025.my.id/api/penutrisian/jadwal";
const char* api_key = "abcd4321";
const char* api_key_siram = "abcd0000";
const char* api_key_nutrisi = "abcd1111";
unsigned long lastSendsiram = 0;
const unsigned long intervalsiram = 60000;
unsigned long previousMillisJadwal = 0;
const unsigned long intervalCekJadwal = 60000;
bool pompaNutrisiMenyala = false;
```

```

unsigned long waktuPompaNutrisiMenyala = 0;
const unsigned long durasiPompaNutrisi = 10000;      String
currentModeSiram = "manual";
String currentStatusSiram = "OFF";
String currentModeNutrisi = "manual";
String currentStatusNutrisi = "OFF";

void connectToWiFi() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Menghubungkan");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        lcd.setCursor(0, 1);
        lcd.print(".");
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Terhubung");
    lcd.setCursor(0, 1);
    lcd.print(WiFi.localIP());
    delay(2000);
}

void kirimDataSuhuKelembapan(float suhu, int intensitasLDR, int
kelembapanUdara, int kelembapanTanah) {
    if (WiFi.status() != WL_CONNECTED) return;
    WiFiClient client;
    HTTPClient http;
    http.begin(client, urlApiSuhu);
    http.addHeader("Content-Type", "application/json");
    StaticJsonDocument<256> doc;
    doc["api_key"] = api_key;
    doc["suhu"] = suhu;
    doc["ldr"] = intensitasLDR;
    doc["udara"] = kelembapanUdara; // ☒ nama disesuaikan
    doc["ktanah"] = kelembapanTanah;
    String payload;
    serializeJson(doc, payload);
    Serial.println("📡 Payload Dikirim:");
    Serial.println(payload); // ☒ Ini sekarang tidak error
    http.POST(payload);
    http.end();
}

void kirimDatasiram(String mode, String status) {
    if (WiFi.status() != WL_CONNECTED) return;
    WiFiClient client;
    HTTPClient http;
    http.begin(client, urlApiPenyiramanUpdate);
    http.addHeader("Content-Type", "application/json");
    StaticJsonDocument<200> doc;
    doc["api_key"] = api_key_siram;
    doc["mode"] = mode;
    doc["status"] = status;
    String payload;

```

```

        serializeJson(doc, payload);
        http.POST(payload);
        http.end();
    }
    void cekStatusPenyiraman() {
        if (WiFi.status() != WL_CONNECTED) return;
        WiFiClient client;
        HTTPClient http;
        http.begin(client, urlApiPenyiramanStatus);
        int httpCode = http.GET();
        if (httpCode == HTTP_CODE_OK) {
            String response = http.getString();
            DynamicJsonDocument doc(512);
            if (!deserializeJson(doc, response)) {
                currentModeSiram = doc["data"]["mode"] | "manual";
                currentStatusSiram = doc["data"]["status"] | "OFF";
            }
        }
        http.end();
    }
    void kirimStatusPenutrisian(String mode, String status) {
        if (WiFi.status() != WL_CONNECTED) return;
        WiFiClient client;
        HTTPClient http;
        http.begin(client, urlApiPenutrisianStatus);
        http.addHeader("Content-Type", "application/json");
        StaticJsonDocument<200> doc;
        doc["api_key"] = api_key_nutrisi;
        doc["mode"] = mode;
        doc["status"] = status;
        String payload;
        serializeJson(doc, payload);
        http.POST(payload);
        http.end();
    }
    WiFiClient client;
    // Ambil status & jadwal penutrisian
    void cekStatusPenutrisian() {
        if (WiFi.status() == WL_CONNECTED) {
            HTTPClient http;
            http.begin(client,
                "http://ta2025.my.id/api/penutrisian/status");
            int httpResponseCode = http.GET();
            Serial.print("Kode Respons Nutrisi: ");
            Serial.println(httpResponseCode);
            if (httpResponseCode == 200) {
                String response = http.getString();
                Serial.print("RESPON STATUS NUTRISI: ");
                Serial.println(response);
                StaticJsonDocument<200> doc;
                DeserializationError error = deserializeJson(doc, response);
                if (error) {
                    Serial.print("Gagal parsing JSON: ");
                    Serial.println(error.c_str());
                } else {
                    // Ubah ini jika struktur JSON-nya tanpa objek "data"

```

```

        currentModeNutrisi = doc["mode"].as<String>();
        currentStatusNutrisi = doc["status"].as<String>();
        Serial.print("Mode Nutrisi: ");
        Serial.println(currentModeNutrisi);
        Serial.print("Status Nutrisi: ");
        Serial.println(currentStatusNutrisi);
    }
} else {
    Serial.print("Gagal HTTP GET, kode: ");
    Serial.println(httpResponseCode);
}
http.end();
} else {
    Serial.println("WiFi tidak terhubung");
}
}
// kelembapan tanah
int bacaKelembapanTanah() {
    int nilai = analogRead(SOIL_MOISTURE_PIN);
    int persen = map(nilai, 0, 1023, 100, 0);
    return constrain(persen, 0, 100);
}
int bacaIntensitasCahaya() {
    int nilaiLDR = analogRead(LDR_PIN);
    return nilaiLDR;
}
void kirimDatanutrisi() {
    static unsigned long previousMillis = 0;
    const unsigned long interval = 1000;
    if (millis() - previousMillis >= interval) {
        previousMillis = millis();
        timeClient.update();
        if (currentModeNutrisi == "otomatis" && WiFi.status() ==
WL_CONNECTED && !pompaNutrisiMenyala) {
            HTTPClient http;
            http.begin(client,
"http://ta2025.my.id/api/penutrisian/jadwal_terdekat");
            int httpCode = http.GET();
            if (httpCode == HTTP_CODE_OK) {
                String response = http.getString();
                Serial.println("📡 Response jadwal: " + response);
                StaticJsonDocument<512> doc;
                DeserializationError error = deserializeJson(doc,
response);
                if (!error && doc["success"] == true && doc["status"] == "ON") {
                    String waktuJadwal = doc["jadwal_waktu"];
                    String tanggalJadwal = doc["jadwal_tanggal"];
                    time_t rawTime = (time_t)timeClient.getEpochTime();
                    struct tm* timeinfo = gmtime(&rawTime);
                    char tanggalSekarang[11];
                    sprintf(tanggalSekarang, "%04d-%02d-%02d",
                        timeinfo->tm_year + 1900,
                        timeinfo->tm_mon + 1,
                        timeinfo->tm_mday);
                    Serial.print("Tanggal sekarang: ");
                    Serial.println(tanggalSekarang);

```

```

        Serial.print("Tanggal jadwal: ");
        Serial.println(tanggalJadwal);
        if
        (String(tanggalJadwal).equals(String(tanggalSekarang))) {
            int jam, menit, detik;
            sscanf(waktuJadwal.c_str(), "%d:%d:%d", &jam, &menit,
&detik);
            int totalDetikJadwal = jam * 3600 + menit * 60 + detik;
            int jamSekarang = timeinfo->tm_hour;
            int menitSekarang = timeinfo->tm_min;
            int detikSekarang = timeinfo->tm_sec;
            int totalDetikSekarang = jamSekarang * 3600 +
menitSekarang * 60 + detikSekarang;
            if (abs(totalDetikSekarang - totalDetikJadwal) <= 10)
{ // toleransi 10 detik
                Serial.println("☑ Jadwal cocok! Menyalakan pompa
nutrisi.");
                digitalWrite(RELAY_NUTRISI, LOW);
                digitalWrite(LED_KUNING, HIGH);
                lcd.clear();
                lcd.setCursor(0, 0);
                lcd.print("Nutrisi Otomatis");
                lcd.setCursor(0, 1);
                lcd.print("Status: ON");
                pompaNutrisiMenyala = true;
                waktuPompaNutrisiMenyala = millis();
                kirimStatusPenutrisian("otomatis", "ON");
            } else {
                Serial.println("⌚ Jadwal belum cocok.");
            }
        } else {
            Serial.println("✗ Tanggal jadwal tidak sama dengan
tanggal sekarang.");
        }
        } else {
            Serial.println("✗ Tidak ada jadwal aktif atau format
salah.");
        }
        } else {
            Serial.println("✗ Gagal ambil jadwal dari server.");
        }
        http.end();
    }
}

if (pompaNutrisiMenyala && millis() - waktuPompaNutrisiMenyala >=
durasiPompaNutrisi) {
    digitalWrite(RELAY_NUTRISI, HIGH);
    digitalWrite(LED_KUNING, LOW);
    pompaNutrisiMenyala = false;
    Serial.println("☑ Pompa nutrisi dimatikan.");
    kirimStatusPenutrisian("otomatis", "OFF");
}
}

void setup() {

```

```

Serial.begin(115200);

pinMode(RELAY_SIRAM, OUTPUT);
pinMode(RELAY_NUTRISI, OUTPUT);
pinMode(LED_MERAH, OUTPUT);
pinMode(LED_KUNING, OUTPUT);
pinMode(LED_HIJAU, OUTPUT);
digitalWrite(RELAY_SIRAM, HIGH);
digitalWrite(RELAY_NUTRISI, HIGH);
digitalWrite(LED_MERAH, LOW);
digitalWrite(LED_KUNING, LOW);
digitalWrite(LED_HIJAU, LOW);
lcd.init();
lcd.backlight();
connectToWiFi();
dht.begin();
}
void loop() {
cekStatusPenyiraman();
delay(2000);
cekStatusPenutrisian();
float suhu = dht.readTemperature();
float kelembapan = dht.readHumidity();
int kelembapanTanah = bacaKelembapanTanah();
int intensitasLDR = bacaIntensitasCahaya();
Serial.println("===== MONITORING =====");
Serial.print("Kelembapan Tanah: ");
Serial.println(" %");
Serial.print("Suhu Udara: ");
Serial.print(suhu);
Serial.println(" °C");
Serial.print("Kelembapan Udara: ");
Serial.print(kelembapan);
Serial.println(" %");
Serial.print("Intensitas Cahaya (LDR): ");
Serial.println(intensitasLDR);
Serial.print("Mode Penyiraman: ");
Serial.println(currentModeSiram);
Serial.print("Status Penyiraman: ");
Serial.println(currentStatusSiram);
Serial.print("Mode Penutrisian: ");
Serial.println(currentModeNutrisi);
Serial.print("Status Penutrisian: ");
Serial.println(currentStatusNutrisi);
Serial.println("=====");
if (!isnan(suhu) && !isnan(kelembapan)) {
    kirimDataSuhuKelembapan(suhu, intensitasLDR, kelembapan, kelembapanTanah);
}
if (currentModeSiram == "otomatis") {
    if (kelembapanTanah < 50) {
        digitalWrite(RELAY_SIRAM, LOW);
        digitalWrite(LED_HIJAU, HIGH);
        if (currentStatusSiram != "ON") {
            currentStatusSiram = "ON";
            kirimDataSiram(currentModeSiram, "ON");
        }
    }
}
}

```

```

    }
  } else {
    digitalWrite(RELAY_SIRAM, HIGH);
    digitalWrite(LED_HIJAU, LOW);
    if (currentStatusSiram != "OFF") {
      currentStatusSiram = "OFF";
      kirimDatasiram(currentModeSiram, "OFF");
    }
  }
} else {
  if (currentStatusSiram == "ON") {
    digitalWrite(RELAY_SIRAM, LOW);
    digitalWrite(LED_HIJAU, HIGH);
  } else {
    digitalWrite(RELAY_SIRAM, HIGH);
    digitalWrite(LED_HIJAU, LOW);
  }
}
if (currentModeNutrisi == "manual") {
  if (currentStatusNutrisi == "ON") {
    digitalWrite(RELAY_NUTRISI, LOW); // Pompa nutrisi ON
    digitalWrite(LED_KUNING, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Nutrisi: ON");
    lcd.setCursor(0, 1);
    lcd.print("Mode Manual");
  } else {
    digitalWrite(RELAY_NUTRISI, HIGH);
    digitalWrite(LED_KUNING, LOW);
  }
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Suhu: ");
lcd.print(suhu);
lcd.print("C");
lcd.setCursor(0, 1);
lcd.print("Tanah: ");
lcd.print(kelembapanTanah);
lcd.print("%");
kirimDatanutrisi();
delay(5000);
}

```