

LAMPIRAN

Lampiran 1 Surat Kesiediaan Membimbing TA Pembimbing 1

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : Mohammad Humam, M.Kom
NIDN : 0618117901
NIPY : 12.002.007
Jabatan Struktural : Dosen Tetap
Jabatan Fungsional : Lektor

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

Nama : Dio Apriansyah
NIM : 22040063
Program Studi : DIII Teknik Komputer

Judul TA : RANCANG BANGUN JENDELA ATAP CERDAS BERBASIS ARDUINO

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, Maret 2025

Mengetahui,
Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing I,



Mohammad Humam, M.Kom
NIPY 12.002.007

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing 2

SURAT KESEDIAN MEMBIMBING TA

Yang bertanda tangan di bawah ini:

Nama : M. Teguh Prihandoyo, M.Kom
NIDN : 0607117001
NIPY : 02.005.012
Jabatan Struktural : -
Jabatan Fungsional : Asisten Ahli

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

Nama : Dio Apriansyah
NIM : 22040063
Program Studi : DIII Teknik Komputer

Judul TA : RANCANG BANGUN SKYLIGHT OTOMATIS
BERBASIS ARDUINO

Demikian pernyataan dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, Maret 2025

Mengetahui,
Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing II,



Ida Afrilijana, ST., M.Kom
NIPY. 12.013.168

M. Teguh Prihandoyo, M.Kom
NIPY. 02.005.012

Lampiran 3 Surat Observasi

Lampiran 4 Source Code

```
#include <AccelStepper.h>

#define LDR_PIN A0
#define RAIN_PIN A1

AccelStepper stepper(AccelStepper::HALF4WIRE, 8, 10, 9, 11);

bool isOpen = false;
bool autoMode = true;
String status = "tutup";
unsigned long lastSend = 0;

void setup() {
  Serial.begin(9600);
  stepper.setMaxSpeed(1000);
  stepper.setAcceleration(500);
  delay(1000);
}

void loop() {
  int ldrValue = analogRead(LDR_PIN);
  int rainValue = analogRead(RAIN_PIN);

  // Mode otomatis aktif
  if (autoMode) {
    if (ldrValue > 800 && rainValue > 700 && !isOpen) {
      bukaJendela();
    }
  }

```

```

} else if ((ldrValue <= 800 // rainValue <= 700) && isOpen) {
    tutupJendela();
}
}

// Kirim data setiap 5 detik
if (millis() - lastSend > 5000) {
    lastSend = millis();

    String data = "LDR:" + String(ldrValue) + ",RAIN:" + String(rainValue) +
",STATUS:" + status;

    Serial.println(data);
}

// Baca perintah dari ESP32
while (Serial.available()) {
    String perintah = Serial.readStringUntil('\n');
    perintah.trim();

    if (perintah == "OPEN") {
        autoMode = false;
        bukaJendela();
    } else if (perintah == "CLOSE") {
        autoMode = false;
        tutupJendela();
    } else if (perintah == "AUTO") {
        autoMode = true;
    }
}
}

```

```

    stepper.run();
}

void bukaJendela() {
    stepper.moveTo(2048 * 5);
    while (stepper.distanceToGo() != 0) {
        stepper.run();
    }
    isOpen = true;
    status = "buka";
}

void tutupJendela() {
    stepper.moveTo(0);
    while (stepper.distanceToGo() != 0) {
        stepper.run();
    }
    isOpen = false;
    status = "tutup";
}

Code esp 32
#include <WiFi.h>
#include <HTTPClient.h>

#define RXD2 16
#define TXD2 17

```

```

const char* SSID = "sulton";
const char* PASS = "12345678";
const String SERVER = "http://192.168.117.7/jendela/monitoring";

unsigned long lastCheck = 0;

String mode = "auto";
String statusJendela = "tutup";

void setup() {
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);

  WiFi.begin(SSID, PASS);
  Serial.print("Menghubungkan ke WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi Terhubung!");
}

void loop() {
  // Terima data dari Arduino
  if (Serial2.available()) {
    String data = Serial2.readStringUntil('\n');
    data.trim();
  }
}

```

```

if (data.length() > 0 && data.startsWith("LDR:")) {
    Serial.println("Dari Arduino: " + data);

    int ldr = 0, rain = 0;
    String status = "";

    int idx1 = data.indexOf("LDR:") + 4;
    int idx2 = data.indexOf(",RAIN:");
    int idx3 = data.indexOf(",STATUS:");

    if (idx1 >= 4 && idx2 > idx1 && idx3 > idx2) {
        ldr = data.substring(idx1, idx2).toInt();
        rain = data.substring(idx2 + 6, idx3).toInt();
        status = data.substring(idx3 + 8);

        kirimKeServer(ldr, rain, status);
    }
}

// Cek perintah dari server tiap 5 detik
if (millis() - lastCheck > 5000) {
    lastCheck = millis();
    cekMode();
    cekStatusJendela();

    if (mode == "auto") {
        Serial2.println("AUTO");
    }
}

```

```

    } else {
        if (statusJendela == "buka") Serial2.println("OPEN");
        else if (statusJendela == "tutup") Serial2.println("CLOSE");
    }
}
}

void kirimKeServer(int ldr, int rain, String status) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        String url = SERVER + "/kirim?ldr=" + ldr + "&rain=" + rain + "&status="
+ status;

        http.begin(url);
        http.setTimeout(5000);

        int code = http.GET();
        Serial.println("Kirim ke: " + url);
        Serial.println("HTTP Code: " + String(code));

        if (code > 0) {
            Serial.println("Respons: " + http.getString());
        }

        http.end();
    }
}

void cekMode() {

```

```

if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(SERVER + "/get_mode");
    http.setTimeout(5000);

    int code = http.GET();
    if (code == 200) {
        mode = http.getString();
        mode.trim();
        Serial.println("Mode: " + mode);
    }

    http.end();
}
}

void cekStatusJendela() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(SERVER + "/status_jendela");
        http.setTimeout(5000);

        int code = http.GET();
        if (code == 200) {
            statusJendela = http.getString();
            statusJendela.trim();
            Serial.println("Status Jendela: " + statusJendela);
        }
    }
}

```

```
http.end();  
}  
}
```

Lampiran 5 Foto Dokumentasi

