

DAFTAR PUSTAKA

- [1] S. Bahroni, *Monitoring Solar Cell Tracking System Monitoring Solar Cell Tracking System*. 2020.
- [2] H. Pranata, A. Soetedjo, and M. I. Ashari, "Rancang Bangun Solar Tracker Dual Axis Panel Surya Berbasis Arduino," *Pros. SENIATI*, vol. 6, no. 1, pp. 9–16, Jul. 2022, doi: 10.36040/seniati.v6i1.4833.
- [3] R. Duanaputri, I. Heryanto/Eryk, M. F. Sajidan, and Ayusta Lukita Wardani, "Sistem Monitoring Online Dan Analisis Performansi Plts Panel Surya Monocrystalline 100 Wp Berbasis Web," *Elposys J. Sist. Kelistrikan*, vol. 10, no. 1, pp. 1–6, 2023, doi: 10.33795/elposys.v10i1.715.
- [4] D. C. D. Pratiwi and H. Habibullah, "Rancang Bangun Sistem Monitoring Volume Air Pada Tandon Menggunakan Pompa Air Otomatis Bertenaga Solar Cell Berbasis Android," *JTEIN J. Tek. Elektro Indones.*, vol. 3, no. 1, pp. 41–53, 2022, doi: 10.24036/jtein.v3i1.175.
- [5] Hanifiyah Darna Fidya Amaral, A. H. Santoso, P. S. Harijanto, and K. A. Wibisono, "Monitoring Sun Tracking Solar Panel Statis Secara Real-Time Berbasis Website," *Elposys J. Sist. Kelistrikan*, vol. 10, no. 3, pp. 178–182, 2023,
- [6] T. D. Hakim and M. Sukma, "Rancang Bangun Dual-Axis Solar Tracker Menggunakan Mikrokontroler Arduino Mega 2560," *J. Elektro*, vol. 10, no. ISSN, pp. 2302–4712, 2022.
- [7] S. P. Sari, O. Candra, and J. Asmi, "Alat Pendeteksi Kebakaran Menggunakan SMS," *JTEIN J. Tek. Elektro Indones.*, vol. 1, no. 2, pp. 251–254, 2020, doi: 10.24036/jtein.v1i2.69.
- [8] A. N. Nazilah Chamim, R. Syahputra, and R. P. Putra, "Rancang Bangun Monitoring Solar Home System Berbasis Internet of Thing Pada Komplek Peternakan Kambing PE Marsudi Luhur," *Berdikari J. Inov. dan Penerapan Ipteks*, vol. 10, no. 2, pp. 185–195, 2022,
- [9] F. Felycia, "Solar Cell Tracking System Dengan Lux Meter Berbasis Arduino Uno R3," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 7, no. 2, pp. 132–140, 2020, doi: 10.30656/prosisko.v7i2.2491.
- [10] Septianisa azzahra, Chistiono and hastuti azia, "Prototype Pembelajaran Pemanfaatan Energi Baru Terbarukan Berbasis Energi Surya", Vol. 4 No. 1 2021: TERANG : Jurnal

Pengabdian Pada Masyarakat Menerangi Negeri.

- [11] A. Maharani, R. Kusumanto, A. Rahman, T. Elektronika, and J. Teknik Elektro -Politeknik Negeri Sriwijaya, “Sistem *Monitoring* Suhu dan Kelembaban Pada Alat Pengering Padi Berbasis *Solar Cell*,” *Teliska*, vol. 16, no. II Juli, pp. 39–44, 2023, [Online].
- [12] B. A. Kusuma and I. A. Rozaq, “Sistem *Monitoring* Kuat Arus, Tegangan Dan Daya *Solar Cell* Pada Alat Ukur Kualitas Air Tambak Udang Vaname,” *J. Elektro Kontrol*, vol. 2, no. 1, pp. 1–16, 2022, doi: 10.24176/elkon.v2i1.7601.
- [13] L. Aditya and D. A. Santoso, “Rancang Bangun Catu Daya Portable 160Watt dengan Panel Surya Monocrystalline 100WP,” vol. 11, no. 1, pp. 47–56, 2023.
- [14] S. Mulyono, M. Qomaruddin, and M. S. Anwar, “Penggunaan Node-Red pada Sistem *Monitoring* dan Kontrol *Green House* berbasis Protokol MQTT,” vol. 3, no. 1, pp. 31–44, 2021.
- [15] A. Fajrur, “Pengenalan Node-Red,” *Wiki.Rdd-Tech.Com*, 2020, [Online].
- [16] S. Mulyono, M. Qomaruddin, and M. Anwar, “Anwar, MS,” *J. Transistor Elektro dan Inform. (TRANSISTOR EI)*, vol. 3, no. 1, pp. 31–44, 2020.
- [17] B. D. Rosalin and W. Aribowo, “*Monitoring* Arus dan Tegangan Pada Pembangkit Listrik Tenaga Ombak Berbasis Node Red dan ESP8266,” *J. Tek. Elektro*, vol. 12, no. 2, pp. 84–91, 2023, doi: 10.26740/jte.v12n2.p84-91.
- [18] D. S. Siregar, D. Setiawan, J. Halim, and I. Zulkarnain, “Sistem *Monitoring* Arah Solar Panel Untuk Pengecasan Baterai Dengan Menggunakan Metode *Fuzzy* Berbasis Iot,” *J. Sist. Komput. Triguna Dharma (JURSIK TGD)*, vol. 2, no. 1, pp. 8–16, 2023, doi: 10.53513/jursik.v2i1.6273.
- [19] M. A. Prasetyo and H. K. Wardana, “Rancang Bangun *Monitoring Solar Tracking System* Menggunakan Arduino dan Nodemcu Esp8266 Berbasis IoT,” *Resist. (Elektronika Kendali Telekomun. Tenaga List. Komputer)*, vol. 4, no. 2, p. 163, 2021, doi: 10.24853/resistor.4.2.163-168.
- [20] M. Usman, “Analisis Intensitas Cahaya Terhadap Energi Listrik Yang Dihasilkan Panel Surya,” *Power Elektron. J. Orang Elektro*, vol. 9, no. 2, pp. 52–57, 2020, doi: 10.30591/polekro.v9i2.2047.

LAMPIRAN A

Listening Program Arduino

```
#include <Arduino.h>
#include <Servo.h>
#include <Adafruit_Sensor.h>
#include "DHT.h"
#include <ArduinoJson.h>
#include <Wire.h>
#include <time.h>
#include <LiquidCrystal_I2C.h>

// Inisialisasi LCD I2C 16x2 (alamat I2C biasanya 0x27 atau 0x3F)
LiquidCrystal_I2C lcd(0x27, 16, 2); // Ubah alamat I2C sesuai
dengan modul Anda

// Inisialisasi DHT11
#define DHTPIN 4 // Pin data DHT11 terhubung
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int ldrSensorPins[] = {A6, A7, A8, A9}; // Pins untuk sensor
LDR
const int ledMerah = 24;
const int ledHijau = 22;
const int buzzer = 26;

// Variabel untuk mengatur tampilan bergantian pada LCD
int displayState = 0;

// Karakter khusus untuk lambang derajat
byte degreeSymbol[8] = {
    B00111,
    B00101,
    B00111,
    B00000,
```

A-2

```
B00000,  
B00000,  
B00000,  
B00000};
```

```
// Variabel untuk delay pembacaan sensor  
unsigned long delayReadIna = 0;  
unsigned long delayPrintLcd = 0;  
unsigned long delayLdr = 0;  
unsigned long delayDhtSensor = 0;  
unsigned long delaySendJson = 0;
```

```
// Inisialisasi objek servo  
Servo horizontal;  
Servo vertical;
```

```
// Variabel untuk menyimpan posisi servo saat ini  
int currentHorizontal = 0;  
int currentVertical = 0;
```

```
// Variabel untuk menyimpan nilai pembacaan LDR  
float ldr1, ldr2, ldr3, ldr4 = 0;  
float averageLdr = 0;  
// Koneksi pin LDR  
// 1 = kiri bawah a7  
// 2 = kanan atas a6  
// 3 = kiri atas a8  
// 4 = kanan bawah a9  
int ldr1t = A6; // LDR kiri atas  
int ldr1r = A8; // LDR kanan atas  
int ldr1d = A9; // LDR kiri bawah  
int ldr1b = A7; // LDR kanan bawah
```

```
// Variabel untuk menyimpan nilai voltage, current, suhu, dan kelembaban
```

```
float voltage0 = 12.4, voltage1 = 12.4, voltage2 = 12.4;
float current0 = 1.32, current1 = 1.21, current2 = 0.41;
float temperature, humidity;

void moveServoGradually(Servo &servo, int &currentPosition, int
targetPosition);
void sendSerialJson();
void readDhtSensor();
void printLcd();
void readLdr();

void setup()
{
    // Menghubungkan objek servo ke pin yang sesuai
    Serial.begin(115200);
    Serial1.begin(9600);
    dht.begin();
    pinMode(ledMerah, OUTPUT);
    pinMode(ledHijau, OUTPUT);
    pinMode(buzzer, OUTPUT);

    // Inialisasi LCD
    lcd.init();
    lcd.backlight();
    lcd.createChar(0, degreeSymbol); // Membuat karakter khusus
    untuk lambang derajat
    delay(1000);
    lcd.clear();

    digitalWrite(ledHijau, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(700);
    digitalWrite(buzzer, LOW);
    digitalWrite(ledMerah, LOW);
    digitalWrite(ledHijau, LOW);
```

A-4

```
horizontal.attach(2);
vertical.attach(3);
horizontal.write(currentHorizontal); // Mengatur posisi awal servo
horizontal ke 180 derajat
vertical.write(0); // Mengatur posisi awal servo vertikal
ke 45 derajat
delay(15000); // Menunggu 2,5 detik untuk
memastikan servo berada di posisi awal

// Menggerakkan servo horizontal ke posisi 110 derajat (arah
barat)
moveServoGradually(horizontal, currentHorizontal, 130);
}

void loop()
{
  if (Serial1.available())
  {
    // Membaca data dari Serial1
    String unixTimeString = Serial1.readStringUntil('\n');

    // Menampilkan data yang diterima di Serial Monitor
    Serial.print("Unix time diterima: ");
    Serial.println(unixTimeString);

    // Mengonversi string Unix time menjadi integer
    unsigned long unixTime = unixTimeString.toInt();

    // Mendapatkan waktu dari Unix time
    struct tm *timeinfo;
    timeinfo = gmtime((time_t *)&unixTime);
    int currentHour = (timeinfo->tm_hour + 7) % 24; // Offset 7
jam untuk WIB (UTC+7)
    int currentMinute = timeinfo->tm_min;
```

```

// Mengatur posisi servo vertikal berdasarkan waktu jam 8:00-
16:00
if (currentHour >= 8 && currentHour < 16)
{
    // Hitung total menit dari jam 8:00
    int totalMinutes = (currentHour - 8) * 60 + currentMinute;

    // Hitung target posisi servo dari 0 hingga 75 derajat
    int targetPosition = map(totalMinutes, 0, 480, 0, 75);

    // Gerakkan servo ke posisi target secara bertahap
    moveServoGradually(vertical, currentVertical,
targetPosition);
}
else
{
    // Jika waktu di luar rentang, atur servo ke posisi 0
    if (currentVertical != 0)
    {
        moveServoGradually(vertical, currentVertical, 0);
    }
}

// Debugging: Tampilkan waktu dan posisi servo
Serial.print("Current Time: ");
Serial.print(currentHour);
Serial.print(":");
Serial.print(currentMinute);
Serial.print(" | Servo Position: ");
Serial.println(currentVertical);
}
sendSerialJson();
readDhtSensor();
printLcd();
readLdr();

```

A-6

```
}

void moveServoGradually(Servo &servo, int &currentPosition, int
targetPosition)
{
    // Menggerakkan servo secara bertahap ke posisi target
    if (currentPosition != targetPosition)
    {
        int step = (targetPosition > currentPosition) ? 1 : -1;
        for (int pos = currentPosition; pos != targetPosition; pos +=
step)
        {
            servo.write(pos);
            delay(100); // Menyesuaikan kecepatan gerakan servo
        }
        currentPosition = targetPosition;
    }
}

void sendSerialJson()
{
    int hor = currentHorizontal - 40;
    if (millis() - delaySendJson >= 1000)
    {
        delaySendJson = millis();

        // Menggunakan StaticJsonDocument untuk JSON
        JsonDocument doc;

        // Mengisi dokumen JSON dengan data sensor
        doc["voltage1"] = voltage0;
        doc["voltage2"] = voltage1;
        doc["voltage3"] = voltage2;
        doc["current1"] = current0;
        doc["current2"] = current1;
```



```

doc["current3"] = current2;
doc["ldr1"] = ldr1;
doc["ldr2"] = ldr2;
doc["ldr3"] = ldr3;
doc["ldr4"] = ldr4;
doc["humidity"] = humidity;
doc["temperature"] = temperature;
doc["derajatHorizontal"] = hor;
doc["derajatVertical"] = currentVertical;

// Serialize JSON ke buffer
char buffer[512];
serializeJson(doc, buffer);

// Mengirim data dari buffer ke Serial 1
Serial.println(buffer);
Serial1.println(buffer);
}
}

void readDhtSensor()
{
  if (millis() - delayDhtSensor >= 2000)
  {
    delayDhtSensor = millis();

    // Membaca nilai suhu dan kelembaban
    temperature = dht.readTemperature();
    humidity = dht.readHumidity();

    if (isnan(temperature) || isnan(humidity))
    {
      Serial.println("Failed to read from DHT sensor!");
      return;
    }
  }
}

```

```
// Menampilkan hasil ke Serial Monitor
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.print(" *C ");
Serial.print("Humidity: ");
Serial.print(humidity);
Serial.println(" %");
}
}

void printLcd()
{
  if (millis() - delayPrintLcd >= 5000)
  {
    delayPrintLcd = millis();
    // Tampilkan data pada LCD secara bergantian menggunakan
    switch case
    lcd.clear();
    switch (displayState)
    {
    case 0:
      lcd.setCursor(0, 0);
      lcd.print("V0:");
      lcd.print(voltage0, 2);
      lcd.print(" I0:");
      lcd.print(current0, 2);
      lcd.setCursor(0, 1);
      lcd.print("V1:");
      lcd.print(voltage1, 2);
      lcd.print(" I1:");
      lcd.print(current1, 2);
      break;

    case 1:
```

```

    lcd.setCursor(0, 0);
    lcd.print("V2:");
    lcd.print(voltage2, 2);
    lcd.print(" I2:");
    lcd.print(current2, 2);
    lcd.setCursor(0, 1);
    lcd.print("T:");
    lcd.print(temperature, 1);
    lcd.write(0); // Tampilkan lambang derajat
    lcd.print("C H:");
    lcd.print(humidity, 1);
    lcd.print("%");
    break;
}

displayState = (displayState + 1) % 2;
}
}

void readLdr()
{

    // Memeriksa apakah cukup waktu telah berlalu sejak pembacaan
    terakhir
    if (millis() - delayLdr >= 200)
    {
        delayLdr = millis(); // Mengupdate waktu pembacaan sensor
        terakhir

        int lt = analogRead(ldrlt); // Membaca nilai dari LDR kiri atas
        int rt = analogRead(ldrrt); // Membaca nilai dari LDR kanan
        atas
        int ld = analogRead(ldrld); // Membaca nilai dari LDR kiri
        bawah

```

A-10

```
int rd = analogRead(ldrrd); // Membaca nilai dari LDR kanan
bawah
ldr1 = lt;
ldr2 = rt;
ldr3 = ld;
ldr4 = rd;
int avt = (lt + rt) / 2; // Menghitung nilai rata-rata atas
int avd = (ld + rd) / 2; // Menghitung nilai rata-rata bawah
int avl = (lt + ld) / 2; // Menghitung nilai rata-rata kiri
int avr = (rt + rd) / 2; // Menghitung nilai rata-rata kanan
int dvert = avt - avd; // Memeriksa perbedaan antara atas dan
bawah
int dhoriz = avl - avr; // Memeriksa perbedaan antara kiri dan
kanan

// Mencetak nilai LDR dan perhitungannya ke Serial Monitor
Serial.print("LDR Kiri Atas: ");
Serial.print(lt);
Serial.print(" | LDR Kanan Atas: ");
Serial.print(rt);
Serial.print(" | LDR Kiri Bawah: ");
Serial.print(ld);
Serial.print(" | LDR Kanan Bawah: ");
Serial.println(rd);

Serial.print("Rata-rata Atas: ");
Serial.print(avt);
Serial.print(" | Rata-rata Bawah: ");
Serial.print(avd);
Serial.print(" | Rata-rata Kiri: ");
Serial.print(avl);
Serial.print(" | Rata-rata Kanan: ");
Serial.println(avr);

Serial.print("Perbedaan Vertikal: ");
```

```
Serial.print(dvert);  
Serial.print(" | Perbedaan Horizontal: ");  
Serial.println(dhoriz);  
}
```

LAMPIRAN B

Listening Program NodeMCU ESP8266

```
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <time.h>

const char *ssid = "Tes";
const char *password = "katasandi";
const char *mqtt_server = "broker.emqx.io";
const int mqtt_port = 1883;

const char *topic_aki = "dt/solarTracking/sensor/aki";
const char *topic_panel_surya =
"dt/solarTracking/sensor/panelSurya";
const char *topic_scc = "dt/solarTracking/sensor/scc";
const char *topic_sensorLain =
"dt/solarTracking/sensor/sensorLain";
const char *topic_control = "cmd/solarTracking/control";
const char *topic_derajat = "cmd/solarTracking/sensor/derajat";

// Zona waktu yang digunakan (WITA UTC+7)
const long gmtoffset_sec = 7 * 3600;
const int daylightOffset_sec = 0; // Tidak ada daylight saving di
Indonesia

int lastMinute = -1; // Menyimpan nilai menit terakhir

#define RELAY_PIN 26

// Inisialisasi WiFi dan MQTT client
WiFiClient espClient;
PubSubClient client(espClient);

// Variabel untuk menyimpan data yang diterima
String receivedData = "";
bool newData = false;
```


B-2

```
void parseAndSendMQTT(String data);
void reconnect();
void setup_wifi();
void callback(char *topic, byte *payload, unsigned int length);

void setup()
{
  Serial.begin(115200);
  Serial2.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH);
  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
  // Mengatur waktu NTP
  configTime(gmtOffset_sec, daylightOffset_sec, "pool.ntp.org",
"time.nist.gov");

  // Menunggu sinkronisasi waktu
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo))
  {
    Serial.println("Gagal mengambil waktu");
    return;
  }

  // Menampilkan waktu saat ini
  Serial.println(&timeinfo, "Waktu saat ini: %A, %B %d %Y
%H:%M:%S");

  // Set nilai menit awal
  lastMinute = timeinfo.tm_min;
}

void loop()
{
  if (!client.connected())
  {
    reconnect();
  }
  client.loop();
}
```



```

if (Serial2.available())
{
    String receivedData = Serial2.readStringUntil('\n');
    parseAndSendMQTT(receivedData);
    // Serial.println(receivedData);
}

struct tm timeinfo;
if (getLocalTime(&timeinfo))
{
    // Cek apakah menit sudah berubah
    if (timeinfo.tm_min != lastMinute)
    {
        // Update lastMinute dengan nilai menit yang baru
        lastMinute = timeinfo.tm_min;

        // Menampilkan waktu saat ini di Serial Monitor
        Serial.println(&timeinfo, "Waktu saat ini: %A, %B %d %Y
%H:%M:%S");
        // Menampilkan waktu saat ini di Serial Monitor
        Serial.println(&timeinfo, "Waktu saat ini: %A, %B %d %Y
%H:%M:%S");
        time_t now = mktime(&timeinfo);
        Serial2.println(now);
        Serial.print("Unix time: ");
        Serial.println(now);
    }
}

void setup_wifi()
{
    delay(10);
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED)
    {

```


B-4

```
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void reconnect()
{
    while (!client.connected())
    {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("solarTracking-zxcv12345"))
        {
            Serial.println("connected");
            client.subscribe(topic_control);
        }
        else
        {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

void parseAndSendMQTT(String data)
{
    Serial.println(data);
    JsonDocument doc;
    DeserializationError error = deserializeJson(doc, data);

    if (error)
    {
        Serial.print(F("deserializeJson() failed: "));
        Serial.println(error.f_str());
        return;
    }
}
```



```

}

// Extract and publish values
float voltage1 = doc["voltage1"];
float voltage2 = doc["voltage2"];
float voltage3 = doc["voltage3"];
float current1 = doc["current1"];
float current2 = doc["current2"];
float current3 = doc["current3"];
float ldr1 = doc["ldr1"];
float ldr2 = doc["ldr2"];
float ldr3 = doc["ldr3"];
float ldr4 = doc["ldr4"];
float humidity = doc["humidity"];
float temperature = doc["temperature"];
float derajatHorizontal = doc["derajatHorizontal"];
float derajatVertical = doc["derajatVertical"];

char payload[256];
// voltage3 dan current3 untuk topik aki
snprintf(payload, sizeof(payload), "{\"teganganAki\": %.2f,
\"arusAki\": %.2f}\", voltage3, current3);
client.publish(topic_aki, payload);
Serial.println(payload);

// voltage1 dan current1 untuk topik panel surya
snprintf(payload, sizeof(payload), "{\"teganganPanelSurya\":
%.2f, \"arusPanelSurya\": %.2f}\", voltage1, current1);
client.publish(topic_panel_surya, payload);
Serial.println(payload);

// voltage2 dan current2 untuk topik SCC
snprintf(payload, sizeof(payload), "{\"teganganSCC\": %.2f,
\"arusSCC\": %.2f}\", voltage2, current2);
client.publish(topic_scc, payload);
Serial.println(payload);

// humidity, temperature, ldr1, ldr2, ldr3, ldr4 untuk topik
intensitas

```


B-6

```
    snprintf(payload, sizeof(payload), "{\"humidity\": %.2f,
    \"temperature\": %.2f, \"ldr1\": %.2f, \"ldr2\": %.2f, \"ldr3\": %.2f,
    \"ldr4\": %.2f}\", humidity, temperature, ldr1, ldr2, ldr3, ldr4);
    client.publish(topic_sensorLain, payload);
    Serial.println(payload);

    // humidity, temperature, ldr1, ldr2, ldr3, ldr4 untuk topik
intensitas
    snprintf(payload, sizeof(payload), "{\"derajatHorizontal\": %.2f,
    \"derajatVertical\": %.2f}\", derajatHorizontal, derajatVertical);
    client.publish(topic_derajat, payload);
    Serial.println(payload);
}

void callback(char *topic, byte *payload, unsigned int length)
{
    String message;
    for (unsigned int i = 0; i < length; i++)
    {
        message += (char)payload[i];
    }

    JsonDocument doc;
    DeserializationError error = deserializeJson(doc, message);

    if (error)
    {
        Serial.print(F("deserializeJson() failed: "));
        Serial.println(error.f_str());
        return;
    }

    bool switchState = doc["switch"];
    Serial.print("switch :");
    Serial.println(switchState);
    if (switchState)
    {
        digitalWrite(RELAY_PIN, LOW); // Relay on (aktif low)
    }
    else
    {
```



```
        digitalWrite(RELAY_PIN, HIGH); // Relay off (aktif low)
    }
}
```

BIODATA PENULIS



Nama : Khayani
Tempat/Tanggal Lahir : Cirebon, 28 September 2002
Alamat : Desa Jagapura Kidul Kecamatan
Gegesik Kab.Cirebon.
Telepon/Hp : 083102066214
Hobi : Bulu Tangkis
Motto : Jika orang lain bisa, maka aku juga
harus bisa melakukannya

Riwayat Pendidikan :

- SDN 2 Jagapura Kidul Tahun 2009-2015
- SMP N2 Gegegesik Tahun 2015-2018
- SMK PUI Gegegesik Tahun 2018-2021
- Politeknik Negeri Cilacap Tahun 2021-2024

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 08 agustus 2024 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).