

## DAFTAR PUSTAKA

- [1] F. Damayanti and T. Supriyatin, “Bercocok Tanam Dengan Sistem Hidroponik Berbasis Ramah Lingkungan Melalui Pemanfaatan Sampah Botol Plastik,” *J. Pelayanan dan Pengabd. Masy.*, vol. 4, no. 1, pp. 9–19, 2020.
- [2] Biro Komunikasi dan Pelayanan Masyarakat, Kementerian Kesehatan RI. (2019, Januari 16). Kurangi Penggunaan Plastik, Limbahnya Berpotensi Bahayakan Kesehatan. Diakses pada 21 Juni 2023 melalui Kementerian Kesehatan Republik Indonesia ([kemkes.go.id](http://kemkes.go.id)).
- [3] M. Iman Mujiarto, ST., “SIFAT DAN KARAKTERISTIK MATERIAL PLASTIK DAN BAHAN ADITIF Iman Mujiarto,” *Repository.Uin-Suska.Ac.Id*, 2023.
- [4] S. Doriza and V. U. G. Putri, “Pemanfaatan Limbah Botol Plastik Melalui Pelatihan Wirausaha Produk Aksesoris Bagi Ibu Rumah Tangga,” *Sarwahita*, vol. 11, no. 2, p. 99, 2019.
- [5] W. L. District *et al.*, “Pemanfaatan Limbah Botol Plastik Sebagai Media,” *J. Abdi Mas TPB*, vol. 3, pp. 40–48, 2021.
- [6] Tondi Haqira, “Rancang Bangun Mesin Ekstruder Filamen 3D Printer,” *Skripsi Tek. Mesin, Fak. Teknol. Ind. Univ. Islam Indones.*, pp. 1–66, 2019.
- [7] Hanafi, I. Sujana, and R. A. Wicaksono, “Rancang Bangun Alat Ekstruder Dengan Pemanfaatan Limbah Plastik Polypropylene Dan Polyethylene Terephthalate Untuk Menghasilkan Filamen 3D Printing,” *J. Teknol. Rekayasa Tek. Mesin*, vol. 3, no. 1, pp. 20–26, 2022.
- [8] D. Sumardiyanto and S. Putra, “Alat pengolahan limbah filament 3D print dengan material polylactic acid (PLA),” *J. Kaji. Tek. Mesin Univ. 17 Agustus 1945 Jakarta*, vol. 6, no. 2, pp. 13–23, 2021.
- [9] M. Taufik, G. Suryani Lubis, M. Ivanto, P. Studi Teknik Mesin, U. Tanjungpura, and J. H. Hadari Nawawi, “Rancang Bangun Mesin Pultrusion Pembuat Filamen 3D Printing Berbasis Limbah Plastik Botol PET,” *Lubis & Ivanto*, vol. 4, no. 1, pp. 1–08, 2023.
- [10] K. Mikula *et al.*, “3D printing filament as a second life of waste plastics—a review,” *Environ. Sci. Pollut. Res.*, vol. 28, no. 10, pp.

- 12321–12333, 2021.
- [11] R. Apriliya, S. B. Bahar, and M. Sayfullah, “Pengaruh Kuat Tekan Beton dengan Menggunakan Bahan Tambah Botol Plastik Kemasan Air Mineral Jenis Polyethylene Terephthalate (Pet),” *SCEJ (Shell Civ. Eng. Journal)*, vol. 6, no. 1, pp. 39–45, 2021.
- [12] J. Program and D. Tiga, “Antara Filment Pla Dengan Filament Petg Pada 3D Printer Ender 5 Pro Politeknik Harapan Bersama Tegal Tahun 2021,” 2021.
- [13] S. Purba, M. Hariri, R. J. Banjarnahor, and S. Natalia, “LED Control System Using Arduino Wemos D1 R1 Based on Web Server Communication Via Internet of Things ( IoT ) Sistem Kontrol LED Menggunakan Arduino Wemos D1 R1 Berbasis Komunikasi Web Server Via Internet of Things ( IoT ),” vol. 2, no. 6, pp. 1397–1408, 2023.
- [14] I. Novianto, M. Hudha, and ..., “Implementasi IoT pada Monitoring Suhu dan Kelembaban Media Budidaya Maggot Berbasis Wemos D1 Mini,” *ULIL ALBAB J. ...*, vol. 1, no. 9, pp. 3115–3126, 2022.
- [15] F. A. Khan, “Developing Robot assisted Plastic 3D Printing Platform DEVELOPING ROBOT ASSISTED PLASTIC 3D PRINTING PLATFORM,” 2021.
- [16] A. Fatoni, “Analysis Of Manufacturing Cnc Router Machine Using Driver TB6560 AND Driver A4988 Based On Arduino Uno Microntroller In CV Barokah Mebel,” *JSNu J. Sci. Nusant.*, vol. 2, no. 1, pp. 7–16, 2022.
- [17] R. F. Zahra, S. P. Sakti, and D. Anggraeni, “Rancang Bangun Pengontrol Jarak Menggunakan Motor Stepper Nema 17 Berbasis Mega 2560 Pro pada Ultrasonic Atomizer Spray Coating,” *J. Fis. Univ. Brawijaya*, no. August, 2021.
- [18] M. Artiyasa, Waryani, D. Muhammad Taufik, and A. De Wibowo, “Sistem Penetasan Telur Berbasis Plc,” *J. Rekayasa Teknol. Nusa Putra*, vol. 7, no. 1, pp. 45–53, 2021.
- [19] I. A. Hiskens and A. Koeman, “Journal of Electrical and Electronics Engineering,” *Power Syst. Param. Estim.*, vol. 15, no. 2, pp. 1–8, 1999.
- [20] “Projektiranje opteretne otporničke mreže i pripadajućeg aktivnog sustava hlađenja za primjenu u DC mikromreži,” 2023.
- [21] A. Staes, “Zadání bakalářské práce,” 2024.

## LAMPIRAN A

### ---PROGRAM---

```
#include "pins.hpp"  
#include "conf.hpp"  
#include "wifi.hpp"  
#include "stepper.hpp"  
#include "hotend.hpp"  
#include "server.hpp"  
#include "ota.hpp"  
  
void setup() {  
  Serial.begin(115200);  
  delay(1000);  
  Konfigurasi();  
  Wifi();  
  OTA();  
  Hotend();  
  motorStepper();  
  Server();  
}  
  
void loop() {  
  wifiTask();  
  server.handleClient();  
  hotendReadTempTask();  
  stepperRunTask();  
  ArduinoOTA.handle();  
  bacaKonfigurasiSerial();  
}
```



## ---KONFIGURASI---

```
#include <ArduinoJson.h>
#include "FS.h"

String msg;
String status;
double So;
int Vo = 0;
int Sm;
double Kp;
double Ki;
double Kd;
double Max;
String LocalIP;
String Gateway;
String Subnet;
int R1;
char ssid[64];
char password[64];

StaticJsonDocument<512> doc;

const char *confFile = "/config.json";

String printConf() {
    String confString;
    serializeJson(doc, confString);
    return confString;
}

void simpanKonfigurasi() {
    SPIFFS.remove("/config.json");
    File file = SPIFFS.open("/config.json", "w");
    if (!file) {
```

```

    msg = "Failed to create file";
    return;
}
doc["So"] = So;
doc["Vo"] = Vo;
doc["Sm"] = Sm;
doc["Kp"] = Kp;
doc["Ki"] = Ki;
doc["Kd"] = Kd;
doc["Max"] = Max;
doc["ssid"] = ssid;
doc["password"] = password;
doc["LocalIP"] = LocalIP;
doc["Subnet"] = Subnet;
doc["Gateway"] = Gateway;
doc["R1"] = R1;
if (serializeJson(doc, file) == 0) {
    msg = "Failed to write to file";
}
Serial.println(printConf());
file.close();
analogWrite(PIN_HEATER, 0);
ESP.restart();
}

```

```

void resetKonfigurasi(){
    Serial.println("reset");
    strcpy(ssid, "");
    strcpy(password, "");
    So = 220;
    Vo = 40;
    Sm = 230;
    Kp = 23.0;
    Ki = 0.043;
    Kd = 160.0;
    Max = 200;
    LocalIP = "";
    Subnet = "255.255.255.0";
    Gateway = "";
}

```

```

    R1 = 10000;
    simpanKonfigurasi();
}

void bacaKonfigurasiSerial(){
    StaticJsonDocument<512> docInput;

    if (Serial.available() > 0)
    {
        // Deserialize the JSON document
        DeserializationError error = deserializeJson(docInput, Serial);
        if (error)
        {
            Serial.println(F("deserializeJson() failed: "));
            Serial.println(error.c_str());
            return;
        } else {
            //Serial.println("json ok");
            doc=docInput;
            //serializeJson(doc,Serial);
            File file = SPIFFS.open("/config.json", "w");
            if (!file) {
                msg = "Gagal membuat file";
                return;
            }
            if (serializeJson(doc, file) == 0) {
                msg = "File gagal";
            }
            file.close();
            Serial.println("konfigurasi dipebarui, memulai kembali...");
            analogWrite(PIN_HEATER, 0);
            ESP.restart();
        }
    }
}

void loadKonfigurasi(bool reset=false) {
    File file = SPIFFS.open("/config.json", "r");
    if (!file) {

```

```

    msg = "gagal membuka /config.json";
    Serial.println("gagal membuka /config.json");
    resetKonfigurasi();
    analogWrite(PIN_HEATER, 0);
    ESP.restart();
}
DeserializationError error = deserializeJson(doc, file);
if (error) {
    msg = "Gagal membaca file";
    Serial.println("Gagal membaca file");
    resetKonfigurasi();
    return;
}
file.close();

strcpy(ssid,
        doc["ssid"] | "",
        sizeof(ssid));

strcpy(password,
        doc["password"] | "",
        sizeof(password));

So = doc["So"] | 220;
Vo = doc["Vo"] | 40;
Sm = doc["Sm"] | 230;
Kp = doc["Kp"]?doc["Kp"].as<double>():23.0;
Ki = doc["Ki"]?doc["Ki"].as<double>():0.043;
Kd = doc["Kd"]?doc["Kd"].as<double>():160.0;
Max = doc["Max"]?doc["Max"].as<double>():200;
LocalIP = doc["LocalIP"] | "";
Subnet = doc["Subnet"] | "255.255.255.0";
Gateway = doc["Gateway"] | "";
R1 = doc["R1"] | 10000;

Serial.println();
Serial.println("So:Suhu");
Serial.println("Vo:Kecepatan");
Serial.println("Sm:Suhu Maksimal");

```



```

Serial.println("Kp:Kp");
Serial.println("Ki:Ki");
Serial.println("Kd:Kd");
Serial.println("R1:R1");
Serial.println("Max:Nilai Maksimal MOSFET (0-255)");
Serial.println("ssid:SSID");
Serial.println("password:SSID Password");
Serial.println("LocalIP:IP address");
Serial.println("Subnet:Subnet");
Serial.println("Gateway:Gateway");
Serial.println(printConf());

}

void Konfigurasi() {
  if (!SPIFFS.begin()) {
    msg = "Kesalahan pemasangan sistem";
    return;
  }
  loadKonfigurasi();
}

```

**---HOTEND---**

```

#include <PID_v1.h>

double T;    //arus temperature

bool F = false;
bool Fc = false;
bool Fi = false;

double Output; //output PID

PID myPID(&T, &Output, &So, Kp, Ki, Kd, DIRECT);

double tempLastSample;
double tempLastFilament;
double tempLastNoFilament;
double tempLastStart;

```

```

//thermistor
float logR2, R2;
//koefisien steinhart untuk thermistor
float c1 = 0.8438162826e-03, c2 = 2.059601750e-04, c3 =
0.8615484887e-07;

double Thermistor(float Volts) {
  R2 = R1 * (1023.0 / (float)Volts - 1.0); //kalkulasi resistansi thermistor

  logR2 = log(R2);
  T = (1.0 / (c1 + c2 * logR2 + c3 * logR2 * logR2 * logR2)); // suhu
dalam satuan Kelvin
  T = T - 273.15; //mengubah satuan Kelvin ke satuan Celcius
  return T;
}

void start(){
  if (tempLastStart==0){
    status = "on";
    V = Vo;
    tempLastStart = millis();
    if (tempLastStart==0) tempLastStart = 1;
  }
}

void stop(){
  status = "off";
  V = 0;
  tempLastStart = 0;
}

void Hotend(){
  myPID.SetTunings(Kp, Ki, Kd);
  myPID.SetOutputLimits(0,Max);
  pinMode(LED_BUILTIN , OUTPUT);
  pinMode(PIN_LIMIT , INPUT);
  if (status=="") start();
}

```

```

void hotendReadTempTask() {
  if (status == "off" && myPID.GetMode() == AUTOMATIC){
    myPID.SetMode(MANUAL);
    Output = 0;
  }
  if (status == "on" && myPID.GetMode() != AUTOMATIC){
    myPID.SetMode(AUTOMATIC);
  }
  if (millis() >= tempLastSample + 100)
  {
    Thermistor(analogRead(PIN_THERMISTOR)); //Volt ke temperatur,
    update T
    if (T > Sm || isnan(T)){ // jika arus temp lebih dari suhu maksimal
      Output = 0;          // maka output PID 0
    } else {
      myPID.Compute();
    }
    if (status == "on"){ // jika status on maka akan memulai
      start();
      if (T > 150 || T > So + 20 ) { // jika arus temperatur lebih dari 150
        digitalWrite(LED_BUILTIN , LOW); // target temperatur siap dan
        led menyala
      } else { // jika tidak maka
        digitalWrite(LED_BUILTIN ,
!digitalRead(LED_BUILTIN)); //mencapai target suhu
      }
    } else { // jika sebaliknya maka led wemos akan mati
      digitalWrite(LED_BUILTIN , HIGH);
    }
  }

  analogWrite(PIN_HEATER, Output);

  Fc = digitalRead(PIN_LIMIT);

  if (Fc && !F) {
    tempLastFilament = millis();
    start();
  }
}

```

```

}

if (!Fc && F) {
    tempLastFilament = 0;
    tempLastNoFilament = millis();
}

F = Fc;

if (Fc && tempLastFilament > 0 && millis() >= tempLastFilament +
3*1000){
    Fi = true;
}

if (!Fc && Fi && tempLastNoFilament > 0 && millis() >=
tempLastNoFilament + 500) { // tidak ada filament
    stop();
    tempLastNoFilament = 0;
    Fi = false;
}

if (!Fc && !Fi && tempLastStart > 0 && millis() >= tempLastStart +
5*60*1000) { // tidak ada filament selama 5 mnt
    stop();
}

tempLastSample = millis();
}
}

```

**---ARDUINOOTA---**

```

#include <ArduinoOTA.h>

bool debugMode = false;

void OTA() {

    ArduinoOTA.onStart([]() {
        String type;

```

```

if (ArduinoOTA.getCommand() == U_FLASH) {
  type = "sketch";
} else {
  type = "filesystem";
}
Serial.println("Start updating " + type);
});
ArduinoOTA.onEnd([]) {
  Serial.println("\nEnd");
});
ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
  Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
});
ArduinoOTA.onError([](ota_error_t error) {
  Serial.printf("Error[%u]: ", error);
  if (error == OTA_AUTH_ERROR) {
    Serial.println("Auth Failed");
  } else if (error == OTA_BEGIN_ERROR) {
    Serial.println("Begin Failed");
  } else if (error == OTA_CONNECT_ERROR) {
    Serial.println("Connect Failed");
  } else if (error == OTA_RECEIVE_ERROR) {
    Serial.println("Receive Failed");
  } else if (error == OTA_END_ERROR) {
    Serial.println("End Failed");
  }
});
ArduinoOTA.begin();
}

```

**---ALAMATPIN---**

```

#define PIN_EN D1
#define PIN_STEP D2
#define PIN_DIR D3

```

```

#define PIN_THERMISTOR A0
#define PIN_HEATER D0
#define PIN_LIMIT D7

```

**---MOTOR STEPPER---**

```

#include <AccelStepper.h>

int V;

AccelStepper
stepper(AccelStepper::FULL2WIRE,PIN_STEP,PIN_DIR);

int stepsPerRevolution = 200;

bool stepperEnable = false;

void motorStepper(){
  stepper.setPinsInverted(true,false,true); //atur pin en terbalik
  stepper.setEnablePin(PIN_EN);
  stepper.disableOutputs();
  stepper.setMaxSpeed(40*stepsPerRevolution+1); // set maksimal
  kecepatan
}

void stepperRunTask(){
  if (status == "off" && stepperEnable) {
    stepper.disableOutputs();
    stepperEnable = false;
  }
  if (status == "on" && !stepperEnable) {
    stepper.enableOutputs();
    stepperEnable = true;
  }
  if (status == "on") {
    stepper.setSpeed(Vo*stepsPerRevolution); // set kecepatan
    stepper.runSpeed();
  }
}
}

```

### ---WiFi---

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ESP8266mDNS.h>

bool apmode = false;
IPAddress local_ip;

double tempLastWifiTask;
double tempStartWifiTask;
bool wifiSiap = false;
String IpAddress2String(const IPAddress& ipAddress)
{
    return String(ipAddress[0]) + String(".") +\
        String(ipAddress[1]) + String(".") +\
        String(ipAddress[2]) + String(".") +\
        String(ipAddress[3]);
}

void AP(){
    apmode = true;
    WiFi.disconnect(true);
    IPAddress local_IP(192,168,4,1);
    IPAddress gateway(192,168,4,1);
    IPAddress subnet(255,255,255,0);
    WiFi.softAPConfig(local_IP, gateway, subnet);

    unsigned char mac[6];
    char APNAME[40];
    WiFi.macAddress(mac);

    sprintf(APNAME, "TA NO INTERNET", mac[3], mac[4], mac[5]);
    if (WiFi.softAP(APNAME)) {
        Serial.println("TA Siap!");
        apmode = true;
    }else{
        Serial.println("TA Failed!");
    }
}
```

```

}

void wifiTask(){
  if (!wifiSiap){
    if (millis() >= tempLastWifiTask + 500){
      if (WiFi.status() == WL_CONNECTED) {
        MDNS.begin("TUGAS AKHIR");
        Serial.println();
        Serial.println(WiFi.localIP());
        wifiSiap=true;
        return;
      }
      if (WiFi.status() == WL_CONNECT_FAILED) {
        AP();
        wifiSiap=true;
        return;
      }
    }
    if (millis() >= tempStartWifiTask + 10000){
      AP();
      wifiSiap=true;
      return;
    }
    Serial.print(".");
    tempLastWifiTask = millis();
  }
}
}

```

```

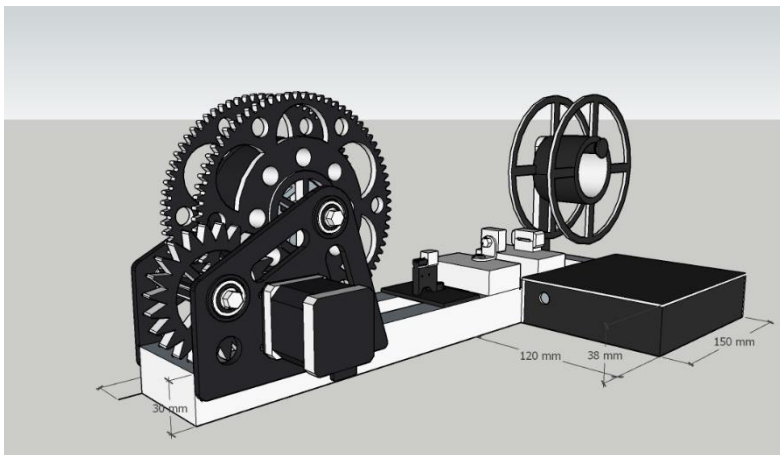
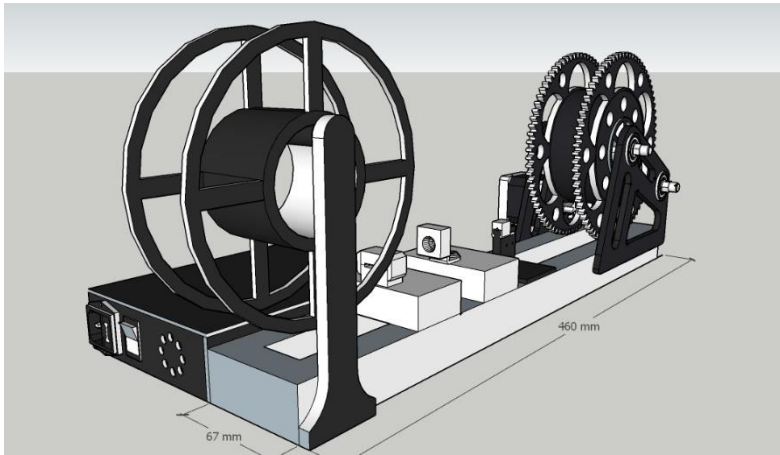
void Wifi()
{
  if (!ssid){
    AP();
    wifiSiap=true;
    return;
  } else {
    IPAddress localip;
    localip.fromString(LocalIP.c_str());
    IPAddress subnet;
    subnet.fromString(Subnet.c_str());
  }
}

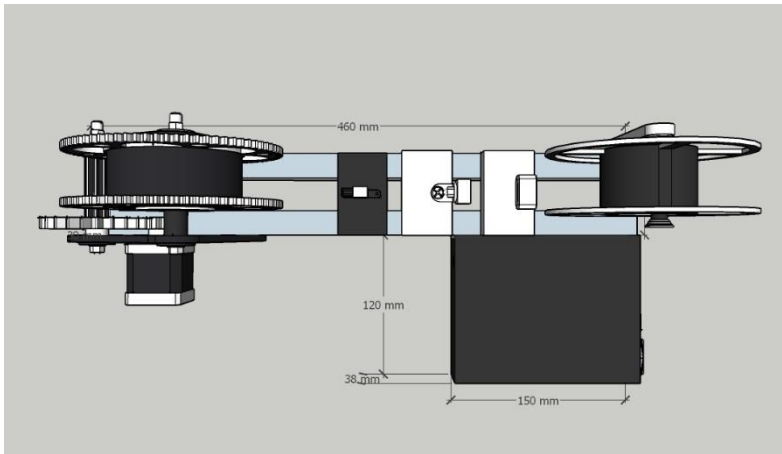
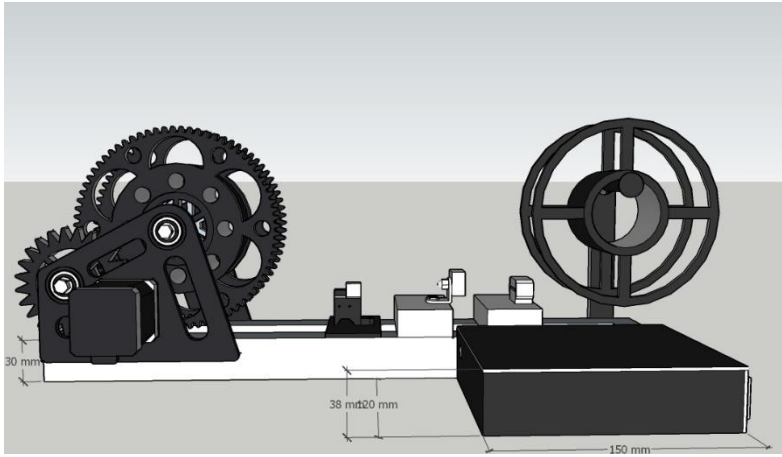
```



```
IPAddress gatewayip;
gatewayip.fromString(Gateway.c_str());
Serial.print("Connecting to ");
Serial.print(ssid);
WiFi.begin(ssid, password); //Koneksi jaringan
if (!WiFi.config(localip, gatewayip, subnet, IPAddress(8, 8, 8, 8))) {
  Serial.println("Konfigurasi wifi ips gagal");
}
tempStartWifiTask = millis();
}
}
```

# LAMPIRAN B





## BIODATA PENULIS



Nama : Alfian Riziq Al-Habsyi  
Tempat/Tanggal Lahir : Tegal, 24 April 2002  
Alamat : Desa Kertasari RT/RW : 002/003,  
Kecamatan Suradadi, Kabupaten Tegal,  
Provinsi Jawa Tengah.  
Telepon/Hp : (+62) 89502311441  
Hobi : Bersepeda dan Mancing  
Moto : Jangan pernah menunda tugasmu  
selsesainkan! kalo bisa besok, kenapa harus  
sekarang.  
Riwayat Pendidikan :  
• SD Negeri 01 Kertasari Tahun 2008-2014  
• SMP VIP AL-HUDA Kebumen Tahun 2014-2017  
• SMK Ma'arif NU 01 Kebumen Tahun 2017-2020  
• Politeknik Negeri Cilacap Tahun 2020-2023

Penulis telah mengikuti sidang pada tanggal 15 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md.)