

DAFTAR PUSTAKA

- [1] a. pasta and m. adensya, “rancang bangun sistem monitoring daya listrik berbasis iot menggunakan esp8266 - amik mdp repository,” *mdp.ac.id*, jul. 2019
- [2] muhamad sarifatullah, “perancangan sistem proteksi thermal pada motor induksi 3 fasa berbasis kontrol arduino menggunakan jaringan iot,” jurnal online mahasiswa (jom) bidang teknik elektro, vol. 1, no. 1, 2020, accessed: apr. 15, 2022.
- [3] sahdan ashari, i. made, and i. bagus, “rancang bangun sistem monitoring dan pengaman motor induksi tiga phasa berbasis mikrokontroler atmega 8535,” dielektrika, vol. 2, no. 2, pp. 123–130, 2015, accessed: apr. 15, 2022.
- [4] o. y. hutajulu, eka dodi suryanto, and denny haryanto sinaga, “implementasi sistem kontrol on/off pompa air sistem tadon berbasis arduino untuk penghematan konsumsi listrik pompa,” jurnal teknik informatika unika santo thomas, vol. 4, no. 2, pp. 232–239, 2019, doi: 10.17605/jti.v4i2.609.
- [5] syarif moh rofiq al- ghony, subuh isnur haryudo, and jati widyo leksono, “perancangan sistem otomatisasi control motor 3 phase menggunakan bluetooth berbasis arduino uno,” reaktom : rekayasa keteknikan dan optimasi, vol. 4, no. 2, pp. 50–55, dec. 2019, doi: 10.33752/reaktom.v4i2.1246.
- [6] h. susanto and amir hamzah, “penerapan konsep internet of things (iot) sebagai monitoring tegangan dan arus pada motor induksi 1 phase,” prosiding snast, pp. 261–270, 2018, accessed: jul. 26, 2022.
- [7] figo rezaldy j, “sistem monitoring, protection dan controlling forward reverse motor listrik tiga fasa berbasis internet of things”
- [8] f. khair, “internet of things, sejarah, teknologi dan penerapannya : review,” j.ilm. teknol. inf., vol. iv, no.3, hal. 62-66, 2015.
- [9] wakhid, a. "penerapan iot dalam rancang bangun sistem proteksi motor induksi tiga fasa dari gangguan beban lebih berbasis mikrokontroler." skripsi pendidikan teknik elektro universitas negeri semarang (2020).
- [10] qoriatul fitriyah, tri vira putr, aditya wirangga p, and m. prihadi eko w, “pemanfaatan aplikasi blynk sebagai alat bantu monitoring energi listrik pada kulkas 1 pintu,” *prosiding seminar nasional nci et*, vol. 1, no. 1, pp. 84–92, 2020.

- [11] t. suryana, "implementasi komunikasi web server nodemcu esp8266 dan web server apache mysql untuk otomatisasi dan kontrol peralatan elektronik jarak jauh via internet - repository unikom," unikom.ac.id, jul. 2021.
- [12] s. anwar, tri artono, nasrul nasrul, dasrul dasrul, and a fadli, "pengukuran energi listrik berbasis pzem-004t," prosiding seminar nasional politeknik negeri lhokseumawe, vol. 3, no. 1, p. 272, 2019, accessed: jul. 26, 2022.
- [13] ridwan, muchlis burhanuddin. "sistem monitoring tanaman hidroponik dengan sensor ph, suhu air dan pemupukan berbasis internet of thing." j. teknoains ftie uty (2019).
- [14] s. anwar, tri artono, nasrul nasrul, dasrul dasrul, and a fadli, "pengukuran energi listrik berbasis pzem-004t," prosiding seminar nasional politeknik negeri lhokseumawe, vol. 3, no. 1, p. 272, 2019, accessed: jul. 26, 2022.
- [15] muhamad sarifatullah, "perancangan sistem proteksi thermal pada motor induksi 3 fasa berbasis kontrol arduino menggunakan jaringan iot," jurnal online mahasiswa (jom) bidang teknik elektro, vol. 1, no. 1, 2020, accessed: jul. 26, 2022.

Lampiran

Lampiran A. Program Mikrokontroler

1. Program pada NodeMCU ESP8266 untuk beban pertama

```
#define BLYNK_TEMPLATE_ID "TMPLms99pYuy"
#define BLYNK_DEVICE_NAME "Monitoring Motor 1"

#define BLYNK_FIRMWARE_VERSION    "0.1.0"

#define BLYNK_PRINT Serial

#define APP_DEBUG

// Define board NODE MCU //
#define USE_NODE_MCU_BOARD

// include library sensor //
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>
#include "BlynkEdgent.h"

//inisialisasi alamat LCD //
LiquidCrystal_I2C lcd(0x27,20,4);

// inisialisasi pin SSR //
const int SSR = D8;

// inisialisasi pin relay //
const int relayalarm = D9;

// inisialisasi DS18B20 //
#define ONE_WIRE_BUS D7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
```

```

// ON OFF SSR //
BLYNK_WRITE(V8) {
int i = param.asInt();
if (i==1){
digitalWrite(SSR, HIGH);
}
else if (i==0){
digitalWrite(SSR, LOW);
}
}

// setting Blynk timmer //
BlynkTimer timer;

// pin PZEM004T //
PZEM004Tv30 pzem1(D5, D6, 0x01); //D5=TX D6=RX
PZEM004Tv30 pzem2(D5, D6, 0x02); //D5=TX D6=RX
PZEM004Tv30 pzem3(D5, D6, 0x03); //D5=TX D6=RX

// pin hall sensor //
const int hallSensorPin = D0; // connect the hall effect
sensor on pin 2
const unsigned long sampleTime = 1000;
int rpmMaximum = 0;

// inisialisasi pembacaan sensor //
float voltage1, voltage2, voltage3;
float current1, current2, current3;
float power1, power2, power3;
float tempC;
int rpm;

void setup()
{
// Debug console //
Serial.begin(9600);

// Set LCD //
lcd.init();

```

```

lcd.begin(20,4);
lcd.backlight();

// Panggil Sensor Suhu
sensors.begin();

// Set Input //
pinMode(hallSensorPin,INPUT);

// Set Output //
pinMode(SSR, OUTPUT);

// set output alarm //
pinMode(relayalarm, OUTPUT);

// Blynk Send Sensor //
timer.setInterval(1000L, bacasensor);
BlynkEdgent.begin();
}

void bacasensor(){
// pembacaan sensor //
sensors.requestTemperatures();
tempC = sensors.getTempCByIndex(0);
rpm = getRPM();
if (rpm > rpmMaximum) rpmMaximum = rpm;
voltage1 = pzem1.voltage();
voltage2 = pzem2.voltage();
voltage3 = pzem3.voltage();
current1 = pzem1.current();
current2 = pzem2.current();
current3 = pzem3.current();
power1 = pzem1.power();
power2 = pzem2.power();
power3 = pzem3.power();
delay(100);

// pengiriman data sensor ke aplikasi Blynk //
Blynk.virtualWrite(V0, voltage1);

```

```

Blynk.virtualWrite(V1, voltage2);
Blynk.virtualWrite(V2, voltage3);
Blynk.virtualWrite(V3, current1);
Blynk.virtualWrite(V4, current2);
Blynk.virtualWrite(V5, current3);
Blynk.virtualWrite(V6, rpm);
Blynk.virtualWrite(V7, tempC);
Blynk.virtualWrite(V9, power1);
Blynk.virtualWrite(V10, power2);
Blynk.virtualWrite(V11, power3);
delay(500);

if
(isnan(voltage1)||isnan(voltage2)||isnan(voltage3)||current1>=2||current2>=2||current3>=2||tempC>=120)
{
    // notifikasi Blynk //
    Blynk.logEvent("motortrip", "MOTOR TRIP!");
}
}

int getRPM()
{
    int count = 0;
    boolean countFlag = LOW;
    unsigned long currentTime = 0;
    unsigned long startTime = millis();
    while (currentTime <= sampleTime)
    {
        if (digitalRead(hallSensorPin) == HIGH)
        {
            countFlag = HIGH;
        }
        if (digitalRead(hallSensorPin) == LOW && countFlag == HIGH)
        {
            count++;
            countFlag=LOW;
        }
    }
}

```

```

    currentTime = millis() - startTime;
  }
  int countRpm = int(10000/float(sampleTime))*count;
  return countRpm;
}

void mylcd(){
  lcd.clear();
  // Tampilkan Tegangan R //
  lcd.setCursor(0,0);
  lcd.print("R:"); lcd.print(voltage1); lcd.print("V");
  // Tampilkan Tegangan S //
  lcd.setCursor(0,1);
  lcd.print("S:"); lcd.print(voltage2); lcd.print("V");
  // Tampilkan Tegangan T //
  lcd.setCursor(0,2);
  lcd.print("T:"); lcd.print(voltage3); lcd.print("V");
  // Tampilkan Arus R //
  lcd.setCursor(11,0);
  lcd.print("R:"); lcd.print(current1); lcd.print("A");
  // Tampilkan Arus S //
  lcd.setCursor(11,1);
  lcd.print("S:"); lcd.print(current2); lcd.print("A");
  // Tampilkan Arus T //
  lcd.setCursor(11,2);
  lcd.print("T:"); lcd.print(current3); lcd.print("A");
  // Tampilkan Temperatur //
  lcd.setCursor(0,3);
  lcd.print("TMP:"); lcd.print(tempC); lcd.print("*C/");
  // Tampilkan RPM //
  lcd.setCursor(12,3);
  lcd.print("RPM:"); lcd.print(rpm);
  delay (5000);
  lcd.clear();
  // tampilkan daya R //
  lcd.setCursor(0,0);
  lcd.print("P:"); lcd.print(power1); lcd.print("W");
  // tampilkan daya R //

```

```

    lcd.setCursor(0,1);
    lcd.print("P:"); lcd.print(power2); lcd.print("W");
    // tampilkan daya R //
    lcd.setCursor(0,2);
    lcd.print("P:"); lcd.print(power3); lcd.print("W");
    delay(5000);
}

void serialmonitor(){
    // sensor PZEM 1 //
    if( !isnan(voltage1) ){
        Serial.print("Voltage: ");
        Serial.print(voltage1);
        Serial.println("V");
    } else {
        Serial.println("Error reading voltage");
    }
    if( !isnan(current1) ){
        Serial.print("Current: ");
        Serial.print(current1);
        Serial.println("A");
    } else {
        Serial.println("Error reading current");
    }
    // sensor PZEM 2 //
    if( !isnan(voltage2) ){
        Serial.print("Voltage: ");
        Serial.print(voltage2);
        Serial.println("V");
    } else {
        Serial.println("Error reading voltage");
    }
    if( !isnan(current2) ){
        Serial.print("Current: ");
        Serial.print(current2);
        Serial.println("A");
    } else {
        Serial.println("Error reading current");
    }
}

```



```

// sensor PZEM 3 //
if( !isnan(voltage3) ){
Serial.print("Voltage: ");
Serial.print(voltage3);
Serial.println("V");
} else {
Serial.println("Error reading voltage");
}
if( !isnan(current3) ){
Serial.print("Current: ");
Serial.print(current3);
Serial.println("A");
} else {
Serial.println("Error reading current");
}
// Sensor suhu //
if( !isnan(tempC) ){
Serial.print("Temperature: ");
Serial.println(tempC);
} else {
Serial.println("Error reading temprature");
}
// Sensor suhu //
if( !isnan(rpm) ){
Serial.print("RPM: ");
Serial.println(rpm);
}
}

void protection(){
if
(isnan(voltage1)||isnan(voltage2)||isnan(voltage3)||current1>=10||cu
rrent2>=10||current3>=10||tempC>=120)
{
// alarm on
digitalWrite(relayalarm, LOW);
// matikan sistem
digitalWrite(SSR, LOW);
}
}

```

```

else {
    digitalWrite(relayalarm, HIGH);
}
}
void loop() {
    getRPM();
    mylcd();
    serialmonitor();
    protection();
    BlynkEdgent.run();
    timer.run();
}

```

2. Program pada NodeMCU ESP8266 untuk beban kedua

```

// Fill-in information from your Blynk Template here //
#define BLYNK_TEMPLATE_ID "TMPLuPrs4kzK"
#define BLYNK_DEVICE_NAME "Monitoring Motor 2"

#define BLYNK_FIRMWARE_VERSION    "0.1.0"

#define BLYNK_PRINT Serial

#define APP_DEBUG

// Define board NODE MCU //
#define USE_NODE_MCU_BOARD

// include liblary sensor //
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>
#include "BlynkEdgent.h"

//inisialisasi alamat LCD //
LiquidCrystal_I2C lcd(0x27,20,4);

// inisialisasi pin SSR //

```

```

const int SSR = D8;

// inialisasi pin relay //
const int relayalarm = D9;

// inialisasi DS18B20 //
#define ONE_WIRE_BUS D7
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// ON OFF SSR //
BLYNK_WRITE(V8) {
int i = param.asInt();
if (i==1){
digitalWrite(SSR, HIGH);
}
else if (i==0){
digitalWrite(SSR, LOW);
}
}

// setting Blynk timmer //
BlynkTimer timer;

// pin PZEM004T //
PZEM004Tv30 pzem1(D5, D6, 0x01); //D5=TX D6=RX
PZEM004Tv30 pzem2(D5, D6, 0x02); //D5=TX D6=RX
PZEM004Tv30 pzem3(D5, D6, 0x03); //D5=TX D6=RX

// pin hall sensor //
const int hallSensorPin = D0; // connect the hall effect
sensor on pin 2
const unsigned long sampleTime = 1000;
int rpmMaximum = 0;

// inialisasi pembacaan sensor //
float voltage1, voltage2, voltage3;
float current1, current2, current3;
float power1, power2, power3;

```

```

float tempC;
int rpm;

void setup()
{
  // Debug console //
  Serial.begin(9600);

  // Set LCD //
  lcd.init();
  lcd.begin(20,4);
  lcd.backlight();

  // Panggil Sensor Suhu
  sensors.begin();

  // Set Input //
  pinMode(hallSensorPin,INPUT);

  // Set Output //
  pinMode(SSR, OUTPUT);

  // set output alarm //
  pinMode(relayalarm, OUTPUT);

  // Blynk Send Sensor //
  timer.setInterval(1000L, bacasensor);
  BlynkEdgent.begin();
}

void bacasensor(){
  // pembacaan sensor //
  sensors.requestTemperatures();
  tempC = sensors.getTempCByIndex(0);
  rpm = getRPM();
  if (rpm > rpmMaximum) rpmMaximum = rpm;
  voltage1 = pzem1.voltage();
  voltage2 = pzem2.voltage();
  voltage3 = pzem3.voltage();
}

```

```

current1 = pzem1.current();
current2 = pzem2.current();
current3 = pzem3.current();
power1 = pzem1.power();
power2 = pzem2.power();
power3 = pzem3.power();
delay(100);

// pengiriman data sensor ke aplikasi Blynk //
Blynk.virtualWrite(V0, voltage1);
Blynk.virtualWrite(V1, voltage2);
Blynk.virtualWrite(V2, voltage3);
Blynk.virtualWrite(V3, current1);
Blynk.virtualWrite(V4, current2);
Blynk.virtualWrite(V5, current3);
Blynk.virtualWrite(V6, rpm);
Blynk.virtualWrite(V7, tempC);
Blynk.virtualWrite(V9, power1);
Blynk.virtualWrite(V10, power2);
Blynk.virtualWrite(V11, power3);
delay(500);

if
(isnan(voltage1)||isnan(voltage2)||isnan(voltage3)||current1>=2||cur
rent2>=2||current3>=2||tempC>=120)
{
    // notifikasi Blynk //
    Blynk.logEvent("motortrip", "MOTOR TRIP!");
}
}

int getRPM()
{
    int count = 0;
    boolean countFlag = LOW;
    unsigned long currentTime = 0;
    unsigned long startTime = millis();
    while (currentTime <= sampleTime)
    {

```

```

    if (digitalRead(hallSensorPin) == HIGH)
    {
        countFlag = HIGH;
    }
    if (digitalRead(hallSensorPin) == LOW && countFlag ==
HIGH)
    {
        count++;
        countFlag=LOW;
    }
    currentTime = millis() - startTime;
}
int countRpm = int(10000/float(sampleTime))*count;
return countRpm;

}

```

```

void mylcd(){
    lcd.clear();
    // Tampilkan Tegangan R //
    lcd.setCursor(0,0);
    lcd.print("R:"); lcd.print(voltage1); lcd.print("V");
    // Tampilkan Tegangan S //
    lcd.setCursor(0,1);
    lcd.print("S:"); lcd.print(voltage2); lcd.print("V");
    // Tampilkan Tegangan T //
    lcd.setCursor(0,2);
    lcd.print("T:"); lcd.print(voltage3); lcd.print("V");
    // Tampilkan Arus R //
    lcd.setCursor(11,0);
    lcd.print("R:"); lcd.print(current1); lcd.print("A");
    // Tampilkan Arus S //
    lcd.setCursor(11,1);
    lcd.print("S:"); lcd.print(current2); lcd.print("A");
    // Tampilkan Arus T //
    lcd.setCursor(11,2);
    lcd.print("T:"); lcd.print(current3); lcd.print("A");
    // Tampilkan Temperatur //
    lcd.setCursor(0,3);
}

```

```

lcd.print("TMP:"); lcd.print(tempC); lcd.print("*C/");
// Tampilkan RPM //
lcd.setCursor(12,3);
lcd.print("RPM:"); lcd.print(rpm);
delay (5000);
lcd.clear();
// tampilkan daya R //
lcd.setCursor(0,0);
lcd.print("P:"); lcd.print(power1); lcd.print("W");
// tampilkan daya R //
lcd.setCursor(0,1);
lcd.print("P:"); lcd.print(power2); lcd.print("W");
// tampilkan daya R //
lcd.setCursor(0,2);
lcd.print("P:"); lcd.print(power3); lcd.print("W");
delay(5000);
}

```

```

void serialmonitor(){
// sensor PZEM 1 //
if( !isnan(voltage1) ){
Serial.print("Voltage: ");
Serial.print(voltage1);
Serial.println("V");
} else {
Serial.println("Error reading voltage");
}
if( !isnan(current1) ){
Serial.print("Current: ");
Serial.print(current1);
Serial.println("A");
} else {
Serial.println("Error reading current");
}
// sensor PZEM 2 //
if( !isnan(voltage2) ){
Serial.print("Voltage: ");
Serial.print(voltage2);
Serial.println("V");
}

```

```

    } else {
    Serial.println("Error reading voltage");
    }
    if( !isnan(current2) ){
    Serial.print("Current: ");
    Serial.print(current2);
    Serial.println("A");
    } else {
    Serial.println("Error reading current");
    }
    // sensor PZEM 3 //
    if( !isnan(voltage3) ){
    Serial.print("Voltage: ");
    Serial.print(voltage3);
    Serial.println("V");
    } else {
    Serial.println("Error reading voltage");
    }
    if( !isnan(current3) ){
    Serial.print("Current: ");
    Serial.print(current3);
    Serial.println("A");
    } else {
    Serial.println("Error reading current");
    }
    // Sensor suhu //
    if( !isnan(tempC) ){
    Serial.print("Temperature: ");
    Serial.println(tempC);
    } else {
    Serial.println("Error reading temperature");
    }
    // Sensor suhu //
    if( !isnan(rpm) ){
    Serial.print("RPM: ");
    Serial.println(rpm);
    }
}

```

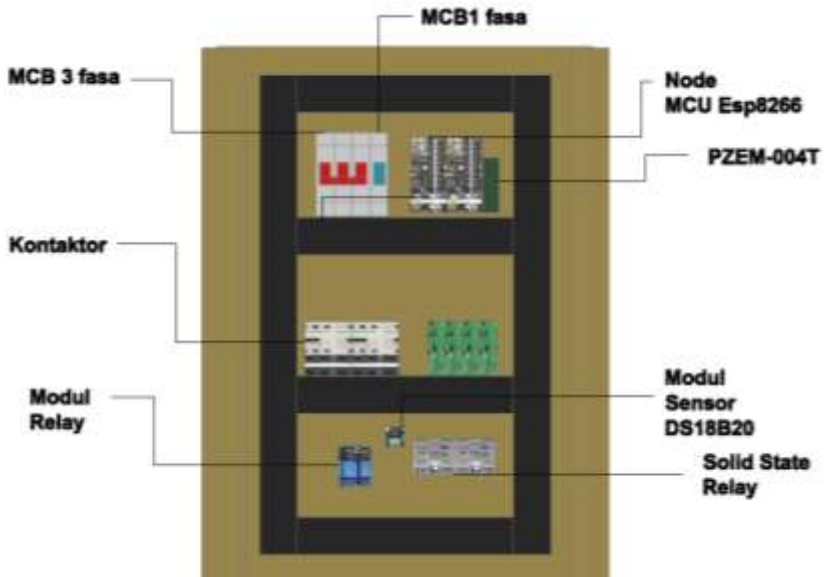


```

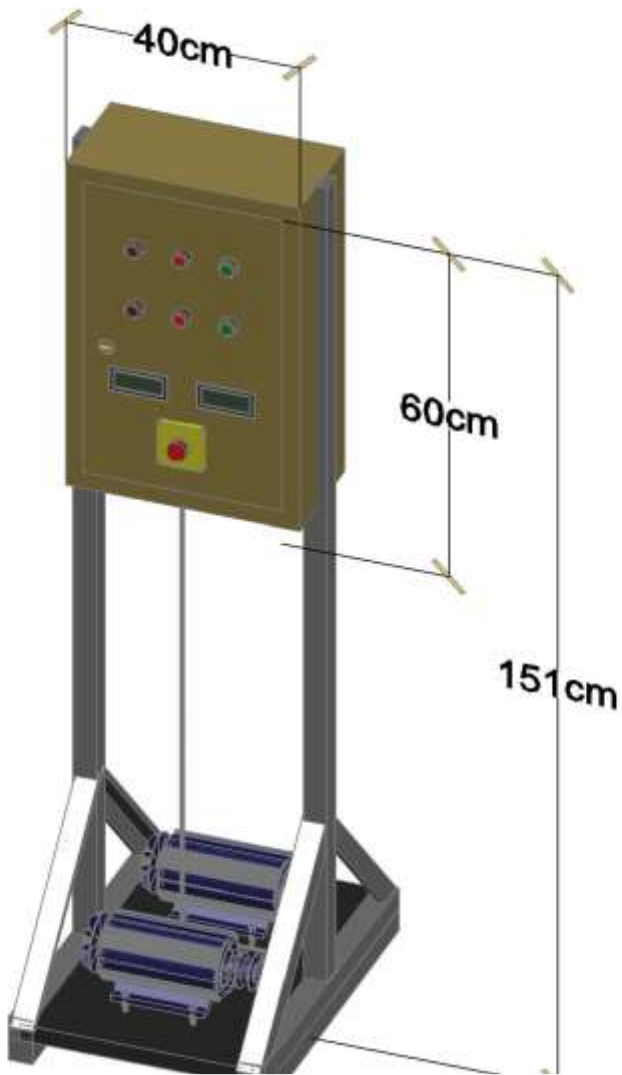
void protection(){
  if
(isnan(voltage1)||isnan(voltage2)||isnan(voltage3)||current1>=2||cur
rent2>=2||current3>=2||tempC>=120)
  {
    // alarm on
    digitalWrite(relayalarm, HIGH);
    // matikan sistem
    digitalWrite(SSR, LOW);
  }
  else {
    digitalWrite(relayalarm, LOW);
  }
}
void loop() {
  getRPM();
  mylcd();
  serialmonitor();
  protection();
  BlynkEdgent.run();
  timer.run();
}

```

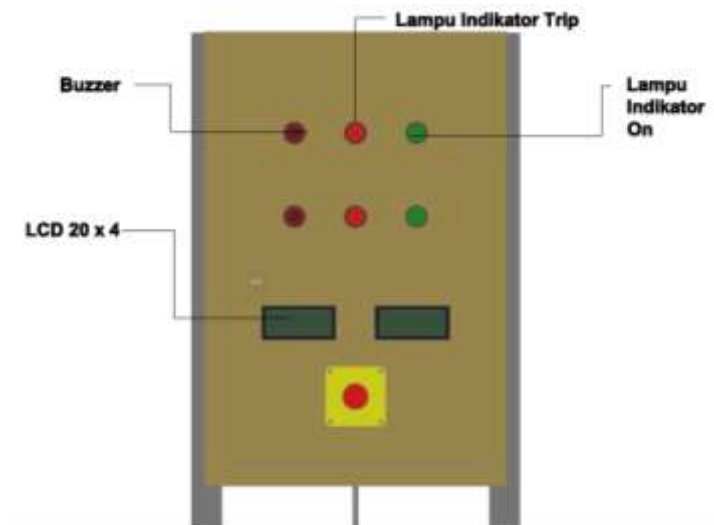

Lampiran B. Desaint mekanik



Gambar Tampak Dalam



Gambar Tampak Luar



Gambar Tampak Depan

Lampiran C. Tampilan Aplikasi Blynk



BIODATA PENULIS



Nama : Mohammad Hafiz Akbar
Tempat / tanggal lahir : Cilacap / 12 juli 2001
Alamat : Jalan Delima Rt/Rw 003/002 Tambakreja,
Cilacap Selatan, Cilacap, Jawa Tengah
Telepon/Hp : 089538240833
Hobi : Bersepeda
Motto : Jadilah hidup seperti larry

Riwayat Pendidikan:

- SD Negeri Kampus Petungan Cilacap Tahun 2007-2013
- SMP Negeri 1 Cilacap Tahun 2013-2016
- SMK Negeri 2 Cilacap Tahun 2016-2019
- Politeknik Negeri Cilacap Tahun 2019-2022

Penulis telah mengikuti sidang Tugas Akhir pada tanggal 3 Agustus 2022 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md)