

DAFTAR PUSTAKA

1. Muzar, Muhammad Aidil ,Syahrizal, dan Mahdi Syukri. (2018). Analisis Pengaruh Suhu Akibat Pembebanan Terhadap Susut Umur Transformator Daya Di Gardu Induk Lambaro. Aceh:Universitas Syiah Kuala
2. PT PLN (Persero) Trans-JBTB. (2018). SOP Deklarasi Kesiapan Trafo dan Transmisi. Sidoarjo.
3. Arifin, Bagus. (2012). Rancang Bangun Sistem Simulasi Pendingin Mesin Secara Otomatis Berbasis Mikrokontroler Avr Atmega128l. Semarang: Universitas Diponegoro.
4. Riza, Novi Ainur. Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis Internet of Things (IoT). (2017). Surabaya: Institut Teknologi Sepuluh Nopember.
5. Ishyafaputro, Bismo. (2016). Rancang Bangun Sistem Peringatan Dini Gangguan Overload Pada Transformator Distribusi Berbasis Mikrokontroler Yang Dilengkapi Dengan Gis. Surabaya: Institut Teknologi Sepuluh Nopember.
6. PT PLN (Persero). (2012). Buku Pedoman Pemeliharaan Trafo. Jakarta.
7. Aprianto, Agung. (2012). Pemeliharaan Trafo Distribusi. Semarang: Universitas Diponegoro.
8. Kadir, Abdul. (2012). Transformator. Jakarta : UI Press.
9. Kadir, Abdul. (2018). Arduino dan Sensor. Yogyakarta: Penerbit Andi.
10. Nurcahyo, Sidik. (.2012). Aplikasi dan teknik Pemrograman Mikrokontroler. Penerbit Andi
11. Karim, Syaiful. (2013). Sensor dan Aktuator 1. Jakarta: Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan.

12. Nurazizah, Ellia. (2017). Rancang Bangun Termometer Digital Berbasis Sensor Ds18b20 Untuk Penyandang Tunanetra. Universitas Telkom.
13. Nofandra, Gatra. (2014). Prototype Kwh Meter Digital. Medan.: Universitas Sumatera Utara.
14. Iswanto. (2015). Mikrokontroller: Teori dan Praktik Atmega 16 dengan Bahasa C. Yogyakarta.: Universitas Muhammadiyah Yogyakarta
15. Prihono, Aru Tirta. (2019). Pemrograman Mikrokontroler dan mikroprosesor. Penerbit Andi



Unipa Surabaya

UNIVERSITAS PGRI ADI BUANA SURABAYA FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

Website : www.ft.unipasby.ac.id E-mail : ft@unipasby.ac.id

BERITA ACARA BIMBINGAN SKRIPSI

Form Skripsi-03

Nama	: Regina Yustisia Arifin	
NIM	: 193609014	
Program Studi	: Teknik Elektro	
Pembimbing	: Atmiasri, S.T., M.T.	
Periode Bimbingan	: Gasal/Genap*) Tahun 2020 / 2021	
Judul Skripsi	Perancangan Alat Monitoring Data Beban Realtime Dan Sistem Pendinginan Otomatis Pada Trafo	

KEGIATAN KONSULTASI / BIMBINGAN

No	Tanggal	Materi pembimbingan	Keterangan	Paraf
1	4 November 2020	Mengubah format proposal ke laporan tugas akhir	acc	
2	13 Desember 2020	Menambahkan foto peralatan pada bab 2	acc	
3	18 Desember 2021	Merapihkan margin penulisan	acc	
4	13 Januari 2021	Penulisan sitasi pada bab 2 dan daftar pustaka	acc	
5	18 Januari 2021	Tabel pengujian sensor	acc	
6	18 Januari 2021	Kesimpulan dan saran	acc	
7	19 Januari 2021	menambahkan surat pernyataan keaslian	acc	
8	19 Januari 2021	menambahkan lampiran	acc	
9	21 Januari 2021	siap diujikan	acc	

Dinyatakan selesai tanggal : 21 Januari 2021



Pembimbing,

Atmiasri, S.T., M.T.

Surabaya, 21 Januari 2021
Mahasiswa,

Regina Yustisia Arifin



UNIVERSITAS PGRI ADI BUANA SURABAYA

FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

Website : www.ft.unipasby.ac.id E-mail : ft@unipasby.ac.id

FORM REVISI SKRIPSI

Nama Mahasiswa : Regina Yustisia Arifin
NIM : 193609014
Fakultas / Progdil : Fakultas Teknik / Teknik Elektro
Judul Skripsi : Perancangan Alat Monitoring data beban realtime dan sistem pendinginan otomatis pada trafo

Ujian Tanggal :

No Bab.	Tanggal	Materi Konsultasi	Keterangan Catatan	Tanda Tangan Penguji
I	5-2-2021	menambahkan sitasi pada latar belakang	acc	
II	5-2-2021	menambahkan metode kontrol pada Bab 2	acc	
III	8-2-2021	mengusulkan rumusan masalah dan tujuan	acc	
IV	8-2-2021	pengumpulan data diperbanyak	acc	
V	10-2-2021	koreksi grafik hasil pengujian	acc	

Disetujui Dosen Penguji
Pada Tanggal, 11-2-2021

Penguji I,

()

Penguji II,

(Parana Ditya S.ST.M.T)

- a. Penyelesaian Revisi paling lambat 2 minggu dari pelaksanaan Ujian Skripsi.
b. Pengetikan, penjilidan, penandatanganan Skripsi dan mengumpulkan Skripsi paling lambat 2 minggu dari revisi.
2. Apabila sampai batas waktu tersebut (point 1,a dan b) mahasiswa belum menyelesaikan revisi dan tanda tangan, maka **Ujian dinyatakan Gugur**.
3. a. Foto copy Form Revisi diserahkan ke Program Studi.
b. Skripsi yang sudah direvisi diserahkan ke Fakultas tiga eksemplar untuk dijilid.

LAMPIRAN

Lampiran 1. Pemrograman

```
#include <Wire.h> // This library is already built in to the Arduino IDE

#include <LiquidCrystal_I2C.h> //This library you can add via Include Library >
Manage Library >

LiquidCrystal_I2C lcd(0x27, 20, 4); //alamat i2c LCD

#define BLYNK_PRINT Serial // Comment this out to disable prints and save
space

#include <SPI.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <DHT.h>

#include <Wire.h>

#include <TimeLib.h>

#include <WidgetRTC.h>

BlynkTimer timer;

WidgetRTC rtc;

#include <DallasTemperature.h>

#include <OneWire.h>

String currentTime;

String currentDate;

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "pI80CFebqIk4UWHoMVCW5BT9d1atO9KG";//Enter the Auth
code which was send by Blink
```

```

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Regina YA";
char pass[] = "bismillah";

// -----inisialisasi sensor arus

const int sensorIn = A0;

int mVperAmp = 185; // use 100 for 20A Module and 66 for 30A Module

double Voltage = 0;

double VRMS = 0;

double AmpsRMS = 0;

double Amps = 0;

//-----inisialisasi DS18B20

#define ONE_WIRE_BUS D4 //D4 pin of nodemcu

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire); // Pass the oneWire reference to
Dallas Temperature.

//-----inisialisasi DHT22

#define DHTPIN D7 // D3

// Uncomment whatever type you're using!

// #define DHTTYPE DHT11 // DHT 11

#define DHTTYPE DHT22 // DHT 22, AM2302, AM2321

```

```

#define DHTTYPE DHT21 // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);

int alarm = 4;

int BuzzerLED = D6;

int fanCooler = D5; //pin D5 untuk output pendingin

void notifyOfHghTemp()
{
    sensors.requestTemperatures(); // Send the command to get
    temperatures

    float h = dht.readHumidity();

    float tDHT = dht.readTemperature(); // or dht.readTemperature(true) for
    Fahrenheit

    float tDS = sensors.getTempCByIndex(0);

    //cek apakah bisa baca DHT

    if (isnan(h) || isnan(tDHT)) {

        Serial.println("Failed to read from DHT sensor!");

        return;

    }

    Voltage = getVPP();

    VRMS = (Voltage/2.0) *0.707; //root 2 is 0.707

    AmpsRMS = (VRMS * 1000)/mVperAmp;

    float daya;

```

```

float i = AmpsRMS;

float v = 110;

daya = i * v;

Blynk.virtualWrite(V2, AmpsRMS);

Blynk.virtualWrite(V5, h);

Blynk.virtualWrite(V6, tDS); //V6 is for Temperature DS18B20

Blynk.virtualWrite(V7, tDHT); //V7 is for Temperature

Blynk.virtualWrite(V3, daya);

if(tDS > 38){
    digitalWrite(BuzzerLED, LOW);

    Blynk.notify("Bahaya, Suhu diatas 38° C");

    digitalWrite(fanCooler, LOW);

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Pendingin ON");

    lcd.setCursor(0, 1);

    lcd.print("Buzzer ON");
}

else{
    digitalWrite(fanCooler, HIGH);

    digitalWrite(BuzzerLED, HIGH);
}

//memanggil nilai sensor

```



```

lcd.setCursor(0, 0);

int readValue = analogRead(sensorIn);

//lcd.print("ADC   :");lcd.print(readValue);lcd.println(" ");

lcd.print("Arus   :");lcd.print(AmpsRMS);lcd.println(" A");

delay(100);

lcd.setCursor(0, 1);

lcd.print("Suhu DS : "); lcd.print(tDS); lcd.print(" "); lcd.print((char)223);
lcd.print("C");

delay(100);

lcd.setCursor(0, 2);

lcd.print("Suhu DHT: "); lcd.print(tDHT); lcd.print(" "); lcd.print((char)223);
lcd.print("C");

delay(100);

lcd.setCursor(0, 3);

lcd.print("Daya   :"); lcd.print(daya); lcd.print(" Watt");

delay(100);

}

//-----//

//-----settingan waktu-----//

void clockDisplay()

{

// You can call hour(), minute(), ... at any time

// Please see Time library examples for details

currentTime = String(hour()) + ":" + minute() + ":" + second();

```

```

currentDate = String(day()) + " " + month() + " " + year();
Serial.print("Current time: ");
Serial.print(currentTime);
Serial.print(" ");
Serial.print(currentDate);
Serial.println();

// Send time to the App
Blynk.virtualWrite(V0, currentTime);

// Send date to the App
Blynk.virtualWrite(V1, currentDate);
}

//-----

void setup() {
  Serial.begin(9600);// initialize serial monitor
  Wire.begin(D2,D1); // inialisasi SDA SCL LCD
  lcd.init(); // initializing the LCD
  lcd.backlight(); // Enable or Turn On the backlight
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  sensors.begin();
  pinMode(fanCooler, OUTPUT);
  digitalWrite(fanCooler, HIGH);
  pinMode(BuzzerLED, OUTPUT);
  digitalWrite(BuzzerLED, HIGH);
}

```

```

Serial.println("Start Sensor Arus");

rtc.begin();

timer.setInterval(1000L,notifyOfHghTemp);
// Setup a function to be called every second
timer.setInterval(1000L, clockDisplay);
}

void loop() {

  Blynk.run(); // Initiates Blynk
  timer.run(); // Initiates SimpleTimer
  notifyOfHghTemp();

}

float getVPP()
{
  float result;

  int readValue;      //value read from the sensor
  int maxValue = 0;   // store max value here
  int minValue = 1024; // store min value here

  uint32_t start_time = millis();
  while((millis()-start_time) < 1000) //sample for 3 Sec
  {
    readValue = analogRead(sensorIn);

```

```
// see if you have a new maxValue
if (readValue > maxValue)
{
    /*record the maximum sensor value*/
    maxValue = readValue;
}
if (readValue < minValue)
{
    /*record the minimum sensor value*/
    minValue = readValue;
}
}

// Subtract min from max
result = ((maxValue - minValue) * 5.0)/1024.0;
return result;
}
```