

DOKUMEN
HAK KEKAYAAN INTELEKTUAL KARYA CIPTAAN



**PROGRAM MIKROKONTROLER UNTUK MONITORING
DAYA LISTRIK TIGA FASE BERBASIS INTERNET OF
THINGS**

PENCIPTA :

- 1. Diah Suwarti Widyastuti, S.T., M.Eng**
- 2. Arif Basuki, S.T., M.T.**
- 3. Enggar Sulisty Nugroho, S.T.**

INSTITUT TEKNOLOGI NASIONAL YOGYAKARTA

2021

Program Mikrokontroler Untuk Monitoring Daya Listrik Tiga Fase berbasis Internet of Things

Diah Suwarti Widyastuti¹, Arif Basuki², Enggar Sulisty Nugroho³

¹Prodi Teknik Elektro, Institut Teknologi Nasional Yogyakarta

²Prodi D3 Teknik Elektronika, Institut Teknologi Nasional Yogyakarta

³Alumni Prodi Teknik Elektro, Institut Teknologi Nasional Yogyakarta

Korespondensi: diahsuwarti@itny.ac.id

Abstrak

Karya Ciptaan jenis Program komputer ini adalah Program Mikrokontroler yang diinstalasi dalam Sistem Monitoring Daya Listrik Tiga Fase berbasis Internet of Things. Sistem tersebut merupakan sistem elektronik yang bertujuan untuk melakukan pengukuran, dan pengiriman besaran listrik tiga fase ke suatu server *platform Internet of Things* melalui jaringan internet. Sistem ini dapat mengukur daya listrik tiga fase, memberikan tampilan penggunaan daya listrik dan memberikan indikasi penggunaan daya listrik dengan memanfaatkan aplikasi Blynk yang sudah terpasang pada suatu *smartphone*. Proses pengukuran dan penampilan daya listrik dilakukan secara real time dengan periode setiap detik.

Kata kunci: Program Mikrokontroler, Monitoring daya listrik, Internet of Things, aplikasi Blynk

PROGRAM MIKROKONTROLER UNTUK MONITORING DAYA LISTRIK TIGA FASE BERBASIS INTERNET OF THINGS

Karya Ciptaan jenis Program komputer ini adalah Program Mikrokontroler yang diinstalasi dalam Sistem Monitoring Daya Listrik Tiga Fase berbasis Internet of Things. Sistem tersebut merupakan sistem elektronik yang bertujuan untuk melakukan pengukuran, dan pengiriman besaran listrik tiga fase ke suatu server *platform Internet of Things* melalui jaringan internet. Sistem ini dapat memberikan tampilan penggunaan daya listrik dan memberikan indikasi penggunaan daya listrik dengan memanfaatkan aplikasi Blynk yang sudah terpasang pada suatu *smartphone*.

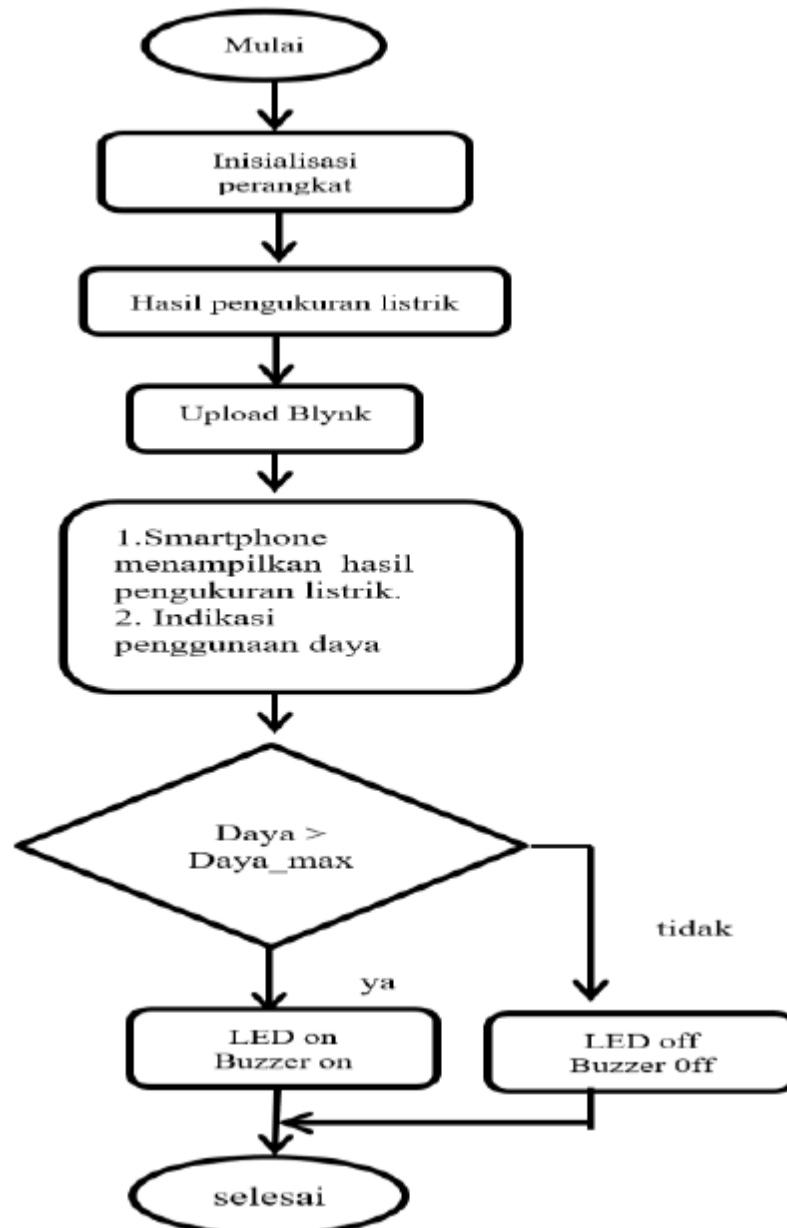
Prinsip Kerja

Sistem Monitoring Daya Listrik Tiga Fase berbasis Internet of Things memiliki prinsip kerja sebagai berikut.

1. Melakukan pengukuran beberapa besaran listrik tiga fase (tegangan, arus, daya, energi, frekuensi dan faktor daya listrik) menggunakan 3 modul sensor, dibawah kendali program yang sudah terpasang pada suatu modul mikrokontroler.
2. Mengirimkan hasil pengukuran ke server *platform Internet of Things* (server Blynk) dengan periode pengiriman setiap 1 detik untuk satu fase listrik. Hasil pengukuran di server dapat diakses dan ditampilkan oleh *smartphone* yang sudah terpasang aplikasi Blynk.
3. Membandingkan besar daya listrik terukur dengan ambang batas daya listrik yang sudah disetting untuk setiap fase listrik. Jika daya listrik hasil pengukuran lebih besar dari nilai ambang batas, alarm berupa bunyi dari *buzzer* dan nyala lampu LED diaktifkan.
4. Memantau adanya pengiriman nilai ambang batas daya listrik dari server Blynk. Jika ada pengiriman, dilakukan setting ulang nilai ambang batas daya listrik yang baru diterima.

Algoritma Pemrograman

Berdasarkan prinsip kerja, Sistem Monitoring Daya Listrik Tiga Fase berbasis Internet of Things memiliki algoritma pemrograman yang ditunjukkan dengan diagram alir pada gambar berikut.



Gambar 1. Diagram alir algoritma pemrograman

LISTING PROGRAM

```
/* Program untuk Monitoring Daya Listrik Tiga Fase berbasis Internet of Things
 * Sistem ini mampu melakukan pengukuran, dan pengiriman melalui jaringan internet
 * ke perangkat handpone yang sudah terpasang aplikasi Blynk, beberapa besaran listrik tiga fase.
 * Beberapa besaran listrik yang dapat diukur untuk setiap fase adalah:
 * 1. Tegangan (volt)
 * 2. Arus (ampere)
 * 3. Daya listrik (watt)
 * 4. Enegr listrik (KWh)
 * 5. Frekuensi (hertz)
 * 6. Fartor daya (tanpa satuan)
 * Periode untuk satu siklus pengukuran, dan pengiriman beberapa besaran listrik untuk setiap fase adalah 1
 detik
 * Sistem ini juga dilengkapi dengan alarm menggunakan buzzer yang akan berbunyi jika daya listrik pada
 fase tertentu
 * melebihi daya tertentu yang telah disetting.
 */

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include "Pzem.h"

char auth[] = "__-o_4PGNmMj76-Zyn17QB2-GHgyIpH0";
char ssid[] = "Redmi"; // "ITNY - Lab Industri";
char pass[] = "22111971"; // "itnyyptn2020";
int roling = 0;
int rMax,sMax,tMax,bzr;
SoftwareSerial pzemSerial(D1, D2); // (RX,TX)
Pzem R(3, pzemSerial);
Pzem S(2, pzemSerial);
Pzem T(1, pzemSerial);
BlynkTimer timer;
BlynkTimer timer2;

void myTimerEvent(){
  if (roling==0){
    R.baca();
    Blynk.virtualWrite(V1 , R.getTegangan());
    Blynk.virtualWrite(V2 , R.getArus());
    Blynk.virtualWrite(V3 , R.getWatt());
    Blynk.virtualWrite(V4 , R.getKwh());
    Blynk.virtualWrite(V5 , R.getFrekuensi());
    Blynk.virtualWrite(V6 , R.getPowerFaktor());
    Blynk.virtualWrite(V7 , R.getAlarm());
    // Please don't send more that 10 values per second.
    Serial.println("Pengukuran besaran-besaran untuk phase R: ");
    Serial.print( "Tegangan: "); Serial.print( R.getTegangan());Serial.println(" volt");
    Serial.print( "Arus: "); Serial.print( R.getArus());Serial.println(" ampere");
    Serial.print( "Daya: "); Serial.print( R.getWatt());Serial.println(" watt");
    Serial.print( "KWh: "); Serial.print( R.getKwh());Serial.println(" KWh");
    Serial.print( "Frekuensi: "); Serial.print( R.getFrekuensi());Serial.println(" Hz");
    Serial.print( "Faktor Daya: "); Serial.println( R.getPowerFaktor());Serial.println("");
  }
}
```

```

if(roling==1){
  S.baca();
  Blynk.virtualWrite(V8 , S.getTegangan());
  Blynk.virtualWrite(V9 , S.getArus());
  Blynk.virtualWrite(V10 , S.getWatt());
  Blynk.virtualWrite(V11 , S.getKwh());
  Blynk.virtualWrite(V12 , S.getFrekuensi());
  Blynk.virtualWrite(V13 , S.getPowerFaktor());
  Blynk.virtualWrite(V14 , S.getAlarm());
  // Please don't send more that 10 values per second.
  Serial.println("Pengukuran besaran-besaran untuk phase S: ");
  Serial.print( "Tegangan: "); Serial.print( S.getTegangan());Serial.println(" volt");
  Serial.print( "Arus: "); Serial.print( S.getArus());Serial.println(" ampere");
  Serial.print( "Daya: "); Serial.print( S.getWatt());Serial.println(" watt");
  Serial.print( "KWh: "); Serial.print( S.getKwh());Serial.println(" KWh");
  Serial.print( "Frekuensi: "); Serial.print( S.getFrekuensi());Serial.println(" Hz");
  Serial.print( "Faktor Daya: "); Serial.println( S.getPowerFaktor());Serial.println("");
}
if(roling==2){
  T.baca();
  Blynk.virtualWrite(V15 , T.getTegangan());
  Blynk.virtualWrite(V16 , T.getArus());
  Blynk.virtualWrite(V17 , T.getWatt());
  Blynk.virtualWrite(V18 , T.getKwh());
  Blynk.virtualWrite(V19 , T.getFrekuensi());
  Blynk.virtualWrite(V20 , T.getPowerFaktor());
  Blynk.virtualWrite(V21 , T.getAlarm());
  // Please don't send more that 10 values per second.
  Serial.println("Pengukuran besaran-besaran untuk phase T: ");
  Serial.print( "Tegangan: "); Serial.print( T.getTegangan());Serial.println(" volt");
  Serial.print( "Arus: "); Serial.print( T.getArus());Serial.println(" ampere");
  Serial.print( "Daya: "); Serial.print( T.getWatt());Serial.println(" watt");
  Serial.print( "KWh: "); Serial.print( T.getKwh());Serial.println(" KWh");
  Serial.print( "Frekuensi: "); Serial.print( T.getFrekuensi());Serial.println(" Hz");
  Serial.print( "Faktor Daya: "); Serial.println( T.getPowerFaktor());Serial.println("");
}
if(++roling==3)roling=0;
bzc=0;
if(R.getWatt())>rMax){
  digitalWrite(D5,HIGH);
  bzc++;
}
else{
  digitalWrite(D5,LOW);
}
if(S.getWatt())>rMax){
  digitalWrite(D6,HIGH);
  bzc++;
}
else{
  digitalWrite(D6,LOW);
}
if(T.getWatt())>rMax){
  digitalWrite(D7,HIGH);
  bzc++;
}
}

```

```

else{
  digitalWrite(D7,LOW);
}
if(bzr>0){
  digitalWrite(D8,HIGH);
}
else{
  digitalWrite(D8,LOW);
}
}

BLYNK_WRITE(V22){
  rMax = param.asInt(); // assigning incoming value from pin V22 to a variable
}
BLYNK_WRITE(V23){
  sMax= param.asInt(); // assigning incoming value from pin V23 to a variable
}
BLYNK_WRITE(V24){
  tMax = param.asInt(); // assigning incoming value from pin V24 to a variable
}

void setup() {
  Serial.begin(115200);
  Serial.println("");
  Serial.println("Coba koneksi ke blynk");
  Blynk.begin(auth, ssid, pass);
  pzemSerial.begin(9600);
  timer.setInterval(1000L, myTimerEvent);
  pinMode(D5,OUTPUT);
  pinMode(D6,OUTPUT);
  pinMode(D7,OUTPUT);
  pinMode(D8,OUTPUT);
}

void loop() {
  Blynk.run();
  timer.run(); // Initiates BlynkTimer
}

```