DAFTAR PUSTAKA

Abizar Rachman, Zainal Arifin, S. M. (2020). Sistem Pengendali Suhu Ruangan Berbasis Internet of Things (IoT) Menggunakan Air Conditioner (AC) Dan NodeMCU V3 ESP82. *Prosiding Seminar Nasional Ilmu Komputer Dan Teknologi Informasi*, *5*(1), 19–23.

Artiyasa, M., Nita Rostini, A., Edwinanto, & Anggy Pradifta Junfithrana. (2021). Aplikasi Smart Home Node Mcu Iot Untuk Blynk. *Jurnal Rekayasa Teknologi Nusa Putra*, *7*(1), 1–7. https://doi.org/10.52005/rekayasa.v7i1.59

Furqon, A., Prasetijo, A. B., & Widianto, E. D. (2019). Rancang Bangun Sistem Monitoring dan Kendali Daya Listrik pada Rumah Kos Menggunakan NodeMCU dan Firebase Berbasis Android. *Techné : Jurnal Ilmiah Elektroteknika*, *18*(02), 93–104. https://doi.org/10.31358/techne.v18i02.202

Garci Reyes, L. (2019). Analisis kebutuhan listrik dan penambahan pembangkit listrik. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Huda, B., & Priyatna, B. (2019). Penggunaan Aplikasi Content Management System (CMS) Untuk Pengembangan Bisnis Berbasis E-commerce. *Systematics*, *1*(2), 81. https://doi.org/10.35706/sys.v1i2.2076

Huda, N., & Khamami, F. (2017). Modifikasi Sistem Kendali Sepeda Listrik. *Jurnal Cahaya Bagaskara*, *1*(1), 30–35.

Hudan, Ivan Safril, R. T. (2019). Rancang Bangun Sistem Monitoring Daya Listrik Pada Kamar Kos Berbasis Internet of Things (Iot). *Jurnal Teknik ELEKTRO*, *08*(01), 91–99.

Kurniawan, A., Despa, D., & Komarudin, M. (2014). Monitoring Besaran Listrik dari Jarak Jauh pada Jaringan Listrik 3 Fasa Berbasis Single Board Computer BCM2835. *Jurnal Informatika Dan Teknik Elektro Terapan*, *2*(3). https://doi.org/10.23960/jitet.v2i3.529

Kurniawan, B., & Lomi, A. (2020). *MEMONITOR ENERGI LISTRIK BERBASIS INTERNET OF THINGS (IoT)*. 1–8.

Kusumah, H., & Pradana, R. A. (2019). Penerapan Trainer Interfacing Mikrokontroler Dan Internet of Things Berbasis Esp32 Pada Mata Kuliah Interfacing. *Journal CERITA*, *5*(2), 120–134. https://doi.org/10.33050/cerita.v5i2.237

Pangestu, A. D., Ardianto, F., & Alfaresi, B. (2019). Sistem Monitoring Beban ·ik Berbasis Arduino Nodemcu Esp8266. *Jurnal Ampere*, *4*(1), 187. s://doi.org/10.31851/ampere.v4i1.2745

D., Andayati, D., & Fakhiyah, E. (2019). Pendeteksi Dini Kebocoran Pada ıng Gas Menggunakan Sensor MQ-6 Berbasis Arduino. *Manajemen Dan*

Teknik Informatika, *07*(01), 19–29.

Saepudin Nirwan, & Hafidz MS. (2020). 871-Article Text-1481-1-10-20200824. *Rancang Bangun Aplikasi Untuk Prototipe Sistem Monitoring Konsumsi Energi Listrik Pada Peralatan Elektronik Berbasis Pzem-004T*, *12*(2), 23–24.

LAMPIRAN

```
#include <PZEM004Tv30.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define DISPLAY_ADDRESS   0x3C // or 0x3D
#define DISPLAY_SCL_PIN   3
#define DISPLAY_SDA_PIN   13  // OLED SDA to GPIO 13(D7)
#define DISPLAY_WIDTH     128 // OLED display width, in pixels
#define DISPLAY_HEIGHT    64  // OLED display height, in pixels
#define DISPLAY_RESET_PIN -1  // Reset pin # (or -1 if sharing with ESP8266 reset pin)
#define BLYNK_PRINT_SERIAL
Adafruit_SSD1306 display(DISPLAY_WIDTH, DISPLAY_HEIGHT, &Wire, DISPLAY_RESET_PIN);

PZEM004Tv30 pzem1(4, 5);   // GPIO4(D2) to Tx PZEM004; GPIO5(D1) to Rx PZEM004
PZEM004Tv30 pzem2(2, 0);   // GPIO2(D4) to Tx PZEM004; GPIO0(D3) to Rx PZEM004
PZEM004Tv30 pzem3(12, 14); // GPIO12(D6) to Tx PZEM004; GPIO14(D5) to Rx PZEM004

char ssid[] = "Kapal Api";
char pass[] = "#71ab21#";
      [] = "Blynk Token"; //Blynk Token


      age1, current1, power1, energy1, frequency1, pf1, va1, VAR1;
```

```
float voltage2, current2, power2, energy2, frequency2, pf2, va2, VAR2;
float voltage3, current3, power3, energy3, frequency3, pf3, va3, VAR3;
float voltage3ph, current3ph, power3ph, energy3ph, frequency3ph, pf3ph, va3ph,
VAR3ph;

void setup() {
  Serial.begin(115200);
  SetupDisplay();
  Blynk.begin(auth, ssid, pass, "iot.serangkota.go.id", 8080); //
}

void SetupDisplay() {
  Wire.begin(DISPLAY_SDA_PIN, DISPLAY_SCL_PIN);
  display.begin(SSD1306_SWITCHCAPVCC, DISPLAY_ADDRESS);

  display.clearDisplay();
  display.setCursor(12, 0);
  display.setTextSize(2);
  display.setTextColor(SSD1306_WHITE);
  display.println("PZEM-004T");
  display.setCursor(15, 20);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("3-PHASE AC METER");
  display.setCursor(12, 46);
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.println("CONNECTING BLYNK..");
  display.display();
```


```
() {
```

```
voltage1 = pzem1.voltage();
voltage1 = zeroIfNan(voltage1);
current1 = pzem1.current();
current1 = zeroIfNan(current1);
power1 = pzem1.power();
power1 = zeroIfNan(power1);
energy1 = pzem1.energy() / 1000; //kwh
energy1 = zeroIfNan(energy1);
frequency1 = pzem1.frequency();
frequency1 = zeroIfNan(frequency1);
pf1 = pzem1.pf();
pf1 = zeroIfNan(pf1);
if (pf1 == 0) {
  va1 = 0;
} else {
  va1 = power1 / pf1;
}
if (pf1 == 0) {
  VAR1 = 0;
} else {
  VAR1 = power1 / pf1 * sqrt(1-sq(pf1));
}

voltage2 = pzem2.voltage();
voltage2 = zeroIfNan(voltage2);
current2 = pzem2.current();
current2 = zeroIfNan(current2);
power2 = pzem2.power();
power2 = zeroIfNan(power2);
         = pzem2.energy() / 1000; //kwh
         = zeroIfNan(energy2);
      cy2 = pzem2.frequency();
```

```
frequency2 = zeroIfNan(frequency2);
pf2 = pzem2.pf();
pf2 = zeroIfNan(pf2);
if (pf2 == 0) {
  va2 = 0;
} else {
  va2 = power2 / pf2;
}
if (pf2 == 0) {
  VAR2 = 0;
} else {
  VAR2 = power2 / pf2 * sqrt(1-sq(pf2));
}

voltage3 = pzem3.voltage();
voltage3 = zeroIfNan(voltage3);
current3 = pzem3.current();
current3 = zeroIfNan(current3);
power3 = pzem3.power();
power3 = zeroIfNan(power3);
energy3 = pzem3.energy() / 1000; //kwh
energy3 = zeroIfNan(energy3);
frequency3 = pzem3.frequency();
frequency3 = zeroIfNan(frequency3);
pf3 = pzem3.pf();
pf3 = zeroIfNan(pf3);
if (pf3 == 0) {
  va3 = 0;
} else {
  power3 / pf3;

  = 0) {
```

```
  VAR3 = 0;
} else {
  VAR3 = power3 / pf3 * sqrt(1-sq(pf3));
}


voltage3ph = sqrt(sq(voltage1) + sq(voltage2) + sq(voltage3));

if ((current1 > 0) && (current2 > 0) && (current3 > 0)) {
current3ph = 1/3 * (current1+current2+current3);
} else {
    if ((current1 == 0) && (current2 > 0) && (current3 > 0)) {
    current3ph = 1/2 * (current2+current3);
   }
    if ((current1 > 0) && (current2 == 0) && (current3 > 0)) {
    current3ph = 1/2 * (current1+current3);
   }
    if ((current1 > 0) && (current2 > 0) && (current3 == 0)) {
    current3ph = 1/2 * (current1+current2);
   }
    if ((current1 > 0) && (current2 == 0) && (current3 == 0)) {
    current3ph = current1;
   }
   if ((current1 == 0) && (current2 > 0) && (current3 == 0)) {
    current3ph = current2;
   }
   if ((current1 == 0) && (current2 == 0) && (current3 > 0)) {
    current3ph = current3;
   }
        rrent1 == 0) && (current2 == 0) && (current3 == 0)) {
        ıt3ph = 0;
```

```
}

power3ph = (power1 + power2 + power3);
energy3ph = (energy1 + energy2 + energy3);
va3ph = (va1 + va2 + va3);
VAR3ph = (VAR1 + VAR2 + VAR3);

if ((frequency1 > 0) && (frequency2 > 0) && (frequency3 > 0)) {
frequency3ph = 1/3 * (frequency1 + frequency2 + frequency3);
} else {
 if((frequency1 > 0) && (frequency2 > 0) && (frequency3 == 0)) {
 frequency3ph = 1/2 * (frequency1+frequency2);
 }
 if((frequency1 > 0) && (frequency2 == 0) && (frequency3 > 0)) {
 frequency3ph = 1/2 * (frequency1+frequency3);
 }
 if((frequency1 == 0) && (frequency2 > 0) && (frequency3 > 0)) {
 frequency3ph = 1/2 * (frequency2+frequency3);
 }
 if((frequency1 > 0) && (frequency2 == 0) && (frequency3 == 0)) {
 frequency3ph = frequency1;
 }
 if((frequency1 == 0) && (frequency2 > 0) && (frequency3 == 0)) {
 frequency3ph = frequency2;
 }
 if((frequency1 == 0) && (frequency2 == 0) && (frequency3 > 0)) {
 frequency3ph = frequency3;
 }
 if((frequency1 == 0) && (frequency2 == 0) && (frequency3 == 0)) {
        ıcy3ph = 0;
```

```
if ((pf1 > 0) && (pf2 > 0) && (pf3 > 0)) {
pf3ph = 1/3 * (pf1+pf2+pf3);
} else {
  if((pf1 > 0) && (pf2 > 0) && (pf3 == 0)) {
  pf3ph = 1/2 * (pf1+pf2);
  }
  if((pf1 > 0) && (pf2 == 0) && (pf3 > 0)) {
  pf3ph = 1/2 * (pf1+pf3);
  }
  if((pf1 == 0) && (pf2 > 0) && (pf3 > 0)) {
  pf3ph = 1/2 * (pf2+pf3);
  }
  if((pf1 > 0) && (pf2 == 0) && (pf3 == 0)) {
  pf3ph = pf1;
  }
  if((pf1 == 0) && (pf2 > 0) && (pf3 == 0)) {
  pf3ph = pf2;
  }
  if((pf1 == 0) && (pf2 == 0) && (pf3 > 0)) {
  pf3ph = pf3;
  }
  if((pf1 == 0) && (pf2 == 0) && (pf3 == 0)) {
  pf3ph = 0;
  }
}

Blynk.run();
WidgetLED led10(V10); //Phase-1 (R) Led Indicator
LED led20(V20); //Phase-2 (S) Led Indicator
LED led30(V30); //Phase-3 (T) Led Indicator
```

```
if (voltage1 > 100) {
  led10.on();
} else {
  led10.off();
}
if (voltage2 > 100) {
  led20.on();
} else {
  led20.off();
}
if (voltage3 > 100) {
  led30.on();
} else {
  led30.off();
}

Blynk.virtualWrite(V1, voltage3ph);
Blynk.virtualWrite(V2, current3ph);
Blynk.virtualWrite(V3, power3ph);
Blynk.virtualWrite(V4, energy3ph);
Blynk.virtualWrite(V5, frequency3ph);
Blynk.virtualWrite(V6, pf3ph);
Blynk.virtualWrite(V7, va3ph);
Blynk.virtualWrite(V8, VAR3ph);

Blynk.virtualWrite(V11, voltage1);
Blynk.virtualWrite(V12, current1);
Blynk.virtualWrite(V13, power1);
Blynk.virtualWrite(V14, energy1);
        irtualWrite(V15, frequency1);
        irtualWrite(V16, pf1);
        irtualWrite(V17, va1);
```

```
Blynk.virtualWrite(V18, VAR1);

Blynk.virtualWrite(V21, voltage2);
Blynk.virtualWrite(V22, current2);
Blynk.virtualWrite(V23, power2);
Blynk.virtualWrite(V24, energy2);
Blynk.virtualWrite(V25, frequency2);
Blynk.virtualWrite(V26, pf2);
Blynk.virtualWrite(V27, va2);
Blynk.virtualWrite(V28, VAR2);

Blynk.virtualWrite(V31, voltage3);
Blynk.virtualWrite(V32, current3);
Blynk.virtualWrite(V33, power3);
Blynk.virtualWrite(V34, energy3);
Blynk.virtualWrite(V35, frequency3);
Blynk.virtualWrite(V36, pf3);
Blynk.virtualWrite(V37, va3);
Blynk.virtualWrite(V38, VAR3);

Serial.println("");
Serial.printf("3Ph Voltage        : %.2f\ V\n", voltage3ph);
Serial.printf("3Ph Current        : %.2f\ A\n", current3ph);
Serial.printf("3Ph Power RMS      : %.2f\ W\n", power3ph);
Serial.printf("3Ph Frequency      : %.2f\ Hz\n", frequency3ph);
Serial.printf("3Ph Cos Phi        : %.2f\ PF\n", pf3ph);
Serial.printf("3Ph Energy         : %.2f\ kWh\n", energy3ph);
Serial.printf("3Ph Apparent Power : %.2f\ VA\n", va3ph);
Serial.printf("3Ph Reactive Power : %.2f\ VAR\n", VAR3ph);
rintf("---------- END ----------");
rintln("");
```

```
Serial.println("");
Serial.printf("Ph1 (R) Voltage       : %.2f\ V\n", voltage1);
Serial.printf("Ph1 (R) Current       : %.2f\ A\n", current1);
Serial.printf("Ph1 (R) Power RMS     : %.2f\ W\n", power1);
Serial.printf("Ph1 (R) Frequency     : %.2f\ Hz\n", frequency1);
Serial.printf("Ph1 (R) Cos Phi       : %.2f\ PF\n", pf1);
Serial.printf("Ph1 (R) Energy        : %.2f\ kWh\n", energy1);
Serial.printf("Ph1 (R) Apparent Power : %.2f\ VA\n", va1);
Serial.printf("Ph1 (R) Reactive Power : %.2f\ VAR\n", VAR1);
Serial.printf("---------- END ----------");
Serial.println("");

Serial.println("");
Serial.printf("Ph2 (S) Voltage       : %.2f\ V\n", voltage2);
Serial.printf("Ph2 (S) Current       : %.2f\ A\n", current2);
Serial.printf("Ph2 (S) Power RMS     : %.2f\ W\n", power2);
Serial.printf("Ph2 (S) Frequency     : %.2f\ Hz\n", frequency2);
Serial.printf("Ph2 (S) Cos Phi       : %.2f\ PF\n", pf2);
Serial.printf("Ph2 (S) Energy        : %.2f\ kWh\n", energy2);
Serial.printf("Ph2 (S) Apparent Power : %.2f\ VA\n", va2);
Serial.printf("Ph2 (S) Reactive Power : %.2f\ VAR\n", VAR2);
Serial.printf("---------- END ----------");
Serial.println("");

Serial.println("");
Serial.printf("Ph3 (T) Voltage       : %.2f\ V\n", voltage3);
Serial.printf("Ph3 (T) Current       : %.2f\ A\n", current3);
Serial.printf("Ph3 (T) Power RMS     : %.2f\ W\n", power3);
Serial.printf("Ph3 (T) Frequency     : %.2f\ Hz\n", frequency3);
rintf("Ph3 (T) Cos Phi       : %.2f\ PF\n", pf3);
rintf("Ph3 (T) Energy        : %.2f\ kWh\n", energy3);
rintf("Ph3 (T) Apparent Power : %.2f\ VA\n", va3);
```

```
Serial.printf("Ph3 (T) Reactive Power : %.2f\ VAR\n", VAR3);
Serial.printf("---------- END ----------");
Serial.println("");

display.clearDisplay();
display.setCursor(32, 0);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println("PHASE-1 (R)");
display.setCursor(0, 9);
display.setTextSize(1);
display.printf("VOLTAGE  : %.1f\ V\n", voltage1);
display.printf("CURRENT  : %.1f\ A\n", current1);
display.printf("POWER    : %.1f\ W\n", power1);
display.printf("COS-PHI  : %.2f\ PF\n", pf1);
display.printf("APPN PWR : %.1f\ VA\n", va1);
display.printf("REAC PWR : %.1f\ VAR\n", VAR1);
display.printf("FREQ.    : %.1f\ Hz\n", frequency1);
//display.printf("ENR Ph1: %.1f\ kWh\n", energy1);
display.display();
delay(5000);

display.clearDisplay();
display.setCursor(32, 0);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println("PHASE-2 (S)");
display.setCursor(0, 9);
display.setTextSize(1);
        printf("VOLTAGE  : %.1f\ V\n", voltage2);
        printf("CURRENT  : %.1f\ A\n", current2);
        printf("POWER    : %.1f\ W\n", power2);
```

```
display.printf("COS-PHI  : %.2f\ PF\n", pf2);
display.printf("APPN PWR : %.1f\ VA\n", va2);
display.printf("REAC PWR : %.1f\ VAR\n", VAR2);
display.printf("FREQ.    : %.1f\ Hz\n", frequency2);
//display.printf("ENR Ph2: %.1f\ kWh\n", energy2);
display.display();
delay(5000);

display.clearDisplay();
display.setCursor(32, 0);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.println("PHASE-3 (T)");
display.setCursor(0, 9);
display.setTextSize(1);
display.printf("VOLTAGE  : %.1f\ V\n", voltage3);
display.printf("CURRENT  : %.1f\ A\n", current3);
display.printf("POWER    : %.1f\ W\n", power3);
display.printf("COS-PHI  : %.2f\ PF\n", pf3);
display.printf("APPN PWR : %.1f\ VA\n", va3);
display.printf("REAC PWR : %.1f\ VAR\n", VAR3);
display.printf("FREQ.    : %.1f\ Hz\n", frequency3);
//display.printf("ENR Ph3: %.1f\ kWh\n", energy3);
display.display();
delay(5000);

display.clearDisplay();
display.setCursor(2, 0);
display.setTextSize(1);
        setTextColor(SSD1306_WHITE);
        println("OVERALL 3-PHASE (WYE)");
        setCursor(0, 16);
```

```
  display.setTextSize(1);
  display.printf("VOLTAGE  : %.1f\ V\n", voltage3ph);
  display.setCursor(0, 25);
  display.printf("CURRENT  : %.1f\ A\n", current3ph);
  display.setCursor(0, 34);
  display.printf("POWER    : %.1f\ W\n", power3ph);
  display.setCursor(0, 43);
  display.printf("COS-PHI  : %.2f\ PF\n", pf3ph);
  //display.printf("APPN PWR : %.1f\ VA\n", va3ph);
  //display.printf("REAC PWR : %.1f\ VAR\n", VAR3ph);
  //display.printf("FREQ.    : %.1f\ Hz\n", frequency3ph);
  display.setCursor(0, 52);
  display.printf("ENERGY   : %.1f\ kWh\n", energy3ph);
  display.display();
  delay(2000);

}

void printValue(String label, float value) {
  if (value != NAN) {
    Serial.print(label); Serial.println(value);
  } else {
    Serial.println("Error Reading");
  }
}

float zeroIfNan(float v) {
  if (isnan(v)) {
    v = 0;
```


```
;
```