LAMPIRAN A

Program

```
#include <PZEM004Tv30.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
#define WIFI_SSID "inuy"
#define WIFI_PASSWORD "00000000"
#define API_KEY
"AIzaSyCBPQ8BaAm4lKyKJQPKDNxhZgnPPTos1k8"
#define DATABASE_URL "https://inuyuni1105-default-
rtdb.firebaseio.com"
#define USER_EMAIL "inuyunierawati23@gmail.com"
#define USER_PASSWORD "politeknikcilacap23"
#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
#define PZEM_RX_PIN 16
#define PZEM_TX_PIN 17
#endif
#if !defined(PZEM_SERIAL
#define PZEM_SERIAL Serial2
#endif
#if defined(ESP32)
PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN,
PZEM_TX_PIN);
#elif defined(ESP8266)
#else
PZEM004Tv30 pzem(PZEM_SERIAL);
#endif

#define cap10 26
#define cap5 27
#define cap3 13
unsigned long timeShow = 0;
unsigned long prefMilis;
byte flagRun;
double s , q_sebelum , q_setelah , kebutuhan , capasitor , micro_farad ;
double xc;
double qc;
double tg_sudut;
float v, i , w , e, f, pf;
float v0, i0 , w0 , e0, f0, pf0;
float v1, i1 , w1 , e1, f1, pf1;
float ss, setelah, hasil;
float c_hasil;
float pf_setelah;
float pf_sebelum;
int counter;
unsigned long backTimer;
bool signupOK = false;
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
  Serial.begin(9600);
  lcd.begin();
  firebaseInitial();
  pinMode(cap10, OUTPUT);
  pinMode(cap5, OUTPUT);
  pinMode(cap3, OUTPUT);
  digitalWrite(cap10, LOW);
  digitalWrite(cap5, LOW);
  digitalWrite(cap3, LOW);
  //  delay(1000);
  //  digitalWrite(cap1, HIGH);
  //  delay(1000);
  //  digitalWrite(cap1, LOW);
}
void loop() {
  pzemRead();
  showLcd();
  //  logicAll();
  logicNew();
  rumusJJL();
  if ((millis() - prefMilis) > 1000) {
    Serial.println("voll= " + String(v) + " ");
    Serial.println("current= " + String(i) + " ");
    Serial.println("watt= " + String(w) + " ");
    Serial.println("energy= " + String(e) + " ");
    Serial.println("frek= " + String(f) + " ");
    Serial.println("pf= " + String(pf) + " ");
    Serial.println("__________________________________");
    prefMilis = millis();
  }
  firebaseSendData();
}
void pzemRead() {
  v = pzem.voltage(), 2;
  i = pzem.current(), 2;
  w = pzem.power(), 2 ;
  e = pzem.energy(), 2;
  f = pzem.frequency(), 2;
  pf = pzem.pf();
}
void rumus() {
  s = v * i;
  q_sebelum = w * (tan(acos (pf)));
  q_setelah = w * (tan(acos (0.98)));
  kebutuhan = q_sebelum - q_setelah;
  capasitor = (kebutuhan * 1000) / (((3.14 * 3.14) * f) * (v * v));
  micro_farad = (capasitor * 1000000) / 1000;
  xc = 1 / (2 * 3.14 * f * micro_farad);
  qc = (v * v) / xc;
  tg_sudut = qc / w;
  setelah = atan(tg_sudut);
  ss = sqrt((kebutuhan * kebutuhan) + (w * w));
  hasil = (w / ss);
}
void rumusJJL() {
  //Rumus S
  s = v * i;
  float radarcos1 = (pf * 3.14) / 180;
  float rad1 = acos((radarcos1) * 180 / 3.14);
  float Arccos1 = rad1 * 57.3;
  // Rumus Q1
  float radsin = (Arccos1 * 3.14) / 180;
  float Sin = sin(radsin);
  float Q1 = v * i * Sin;
  //Rumus Arccos 2
  float phi2 = 0.98;
  float radarcos2 = (phi2 * 3.14) / 180;
  float rad2 = acos((radarcos2) * 180 / 3.14);
  float Arccos2 = rad2 * 57.3;

  // Rumus Q
  float radtan = (Arccos2 * 3.14) / 180;
  float Tan = tan(radtan);
  float Q = w * Tan;
  // Rumus Q2
  float Q2 = Q1 - Q;
  //Rumus Z
  float Z = pow(v, 2) / Q2;
  float C = (1 / (2 * 3.14 * 50 * Z)) * 1000000;
  if (C < 0) {
    C = C * -1;
  }
  else {
    C = C;
  }
  c_hasil = C;
}
void showLcd() {
```

```
if (millis() - timeShow > 2000) {
  lcd.clear();
  flagRun++;
  timeShow = millis();
}
switch (flagRun) {
  case 0:
    break;
  case 1:
    lcd.setCursor(10, 0);
    lcd.print("READING ");
    lcd.setCursor(10, 1);
    lcd.print("AKTUAL  ");
    lcd.setCursor(0, 0);
    lcd.print("V=       |");
    lcd.setCursor(2, 0);
    lcd.print(v);
    lcd.setCursor(0, 1);
    lcd.print("I=       |");
    lcd.setCursor(2, 1);
    lcd.print(i);
lcd.setCursor(0, 2);
lcd.print("F=       |");
lcd.setCursor(2, 2);
lcd.print(f);

lcd.setCursor(0, 3);
lcd.print("Pf=      |");
lcd.setCursor(3, 3);
lcd.print(pf);

lcd.setCursor(10, 2);
lcd.print("E=        ");
lcd.setCursor(12, 2);
lcd.print(e);

lcd.setCursor(10, 3);
lcd.print("P=        ");
lcd.setCursor(12, 3);
lcd.print(w);

break;
```

```
case 2:
  lcd.setCursor(0, 0);
  lcd.print(" KEBUTUHAN CAPASITOR");
  lcd.setCursor(2, 2);
  lcd.print(String(c_hasil) + "uF  ");
  break;


case 4:
  lcd.setCursor(0, 0);
  lcd.print("    pf_sebelum    ");
  lcd.setCursor(2, 2);
  lcd.print(String(pf_sebelum) );
  break;
case 5:
  lcd.setCursor(0, 0);
  lcd.print("    pf_setelah    ");
  lcd.setCursor(2, 2);
  lcd.print(String(pf_setelah) );
  break;
case 6:
  lcd.setCursor(0, 0);
```

```
    lcd.print(" ---Wait a moment---");
    lcd.setCursor(0, 1);
    lcd.print("    SENDING DATA    ");
    lcd.setCursor(0, 2);
    lcd.print("     TO FIREBASE    ");


    break;
   case 7:
    flagRun = 1;
    break;
  }
}
void logicNew() {
 //  if (c_hasil <= 0 ) {
 //    digitalWrite(cap10, LOW);
 //    digitalWrite(cap5, LOW);
 //    digitalWrite(cap3, LOW);
 //    pf_sebelum = pf;
 //  }
 if (c_hasil >= 1.0 and c_hasil <= 3.0) {
   if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, HIGH);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
}
else if (c_hasil > 3.0 and c_hasil <= 5.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, HIGH);
    digitalWrite(cap3, LOW);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
}
else if (c_hasil > 5.0 and c_hasil <= 8.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, HIGH);
    digitalWrite(cap3, HIGH);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
```

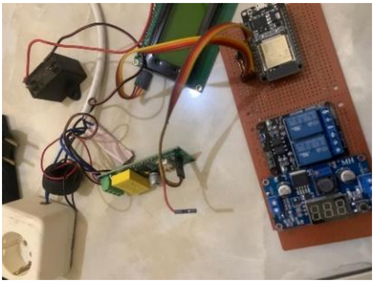
```
}
else if (c_hasil > 8.0 and c_hasil <= 10.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, HIGH);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }

}
else if (c_hasil > 10.0 and c_hasil <= 13.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, HIGH);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, HIGH);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
}
else if (c_hasil > 13.0 and c_hasil <= 15.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, HIGH);
    digitalWrite(cap5, HIGH);
    digitalWrite(cap3, LOW);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }

}
else if (c_hasil > 15.0 and c_hasil <= 18.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, HIGH);
    digitalWrite(cap5, HIGH);
    digitalWrite(cap3, HIGH);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
```

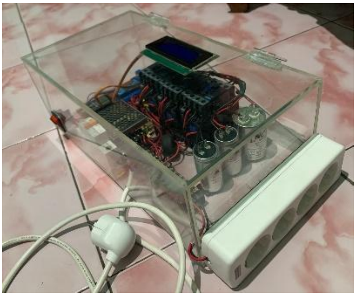
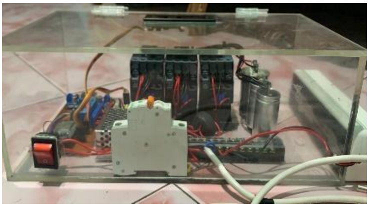
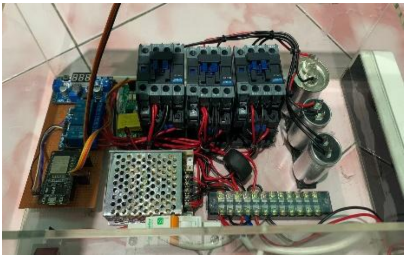
```
}
else if ( c_hasil > 18.0) {
  if (i <= 0) {
    digitalWrite(cap10, LOW);
    digitalWrite(cap5, LOW);
    digitalWrite(cap3, LOW);
    pf_sebelum = 0;
    pf_setelah = 0;
  }
  else {
    digitalWrite(cap10, HIGH);
    digitalWrite(cap5, HIGH);
    digitalWrite(cap3, HIGH);
    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
  }
}
else {
      digitalWrite(cap10, LOW);
      digitalWrite(cap5, LOW);
      digitalWrite(cap3, LOW);
      pf_sebelum = pf;
  }
}

//void logicAll() {
//
//  if (pf_sebelum >= 0.01 and  pf_sebelum <= 0.12) {
//    if (i <= 0) {
//      digitalWrite(cap20, LOW);
//      digitalWrite(cap10, LOW);
//      digitalWrite(cap5, LOW);
//      pf_sebelum = 0;
//      pf_setelah = 0;
//    }
//    else {
//      digitalWrite(cap20, HIGH);
//      digitalWrite(cap10, HIGH);
//      digitalWrite(cap5, HIGH);
//      pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//      Serial.println("35uf");
//    }
//  }
//  else if (pf_sebelum >= 0.13 and  pf_sebelum <= 0.25) {
//    digitalWrite(cap20, HIGH);
//    digitalWrite(cap10, HIGH);
//    digitalWrite(cap5, LOW);
//    pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//    Serial.println("30uf");
//  }
//  else if (pf_sebelum >= 0.26 and  pf_sebelum <= 0.38) {
//    if (i <= 0) {
//      digitalWrite(cap20, LOW);
//      digitalWrite(cap10, LOW);
//      digitalWrite(cap5, LOW);
//      pf_sebelum = 0;
//      pf_setelah = 0;
//    }
//    else {
//      digitalWrite(cap20, HIGH);
//      digitalWrite(cap10, LOW);
//      digitalWrite(cap5, HIGH);
//      pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//      Serial.println("25uf");
//    }
//  }
//  else if (pf_sebelum >= 0.39 and  pf_sebelum <= 0.51) {
//    if (i <= 0) {
//      digitalWrite(cap20, LOW);
//      digitalWrite(cap10, LOW);
//      digitalWrite(cap5, LOW);
//      pf_sebelum = 0;
//      pf_setelah = 0;
//    }
//    else {
//      digitalWrite(cap20, HIGH);
//      digitalWrite(cap10, LOW);
//      digitalWrite(cap5, LOW);
//      pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//      Serial.println("20uf");
//    }
//  }
//  else if (pf_sebelum >= 0.52 and  pf_sebelum <= 0.64) {
//    if (i <= 0) {
//      digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, LOW);
//     pf_sebelum = 0;
//     pf_setelah = 0;
//   }
//   else {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, HIGH);
//     digitalWrite(cap5, HIGH);
//     pf_setelah = pzem.pf() + (0.98  - pzem.pf());
//
//     Serial.println("15uf" );
//   }
// }
// else if (pf_sebelum >= 0.65 and  pf_sebelum <= 0.77) {
//   if (i <= 0) {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, LOW);
//     pf_sebelum = 0;
//     pf_setelah = 0;
//   }
//   else {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, HIGH);
//     digitalWrite(cap5, LOW);
//     pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//     Serial.println("10uf");
//   }
// }
// else if (pf_sebelum >= 0.78 and  pf_sebelum <= 0.90) {
//   if (i <= 0) {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, LOW);
//     pf_sebelum = 0;
//     pf_setelah = 0;
//   }
//   else {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, HIGH);
//     pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//     Serial.println("5uf");
//   }
// }
// else if (pf_sebelum >= 0.91 ) {
//   if (i <= 0) {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, LOW);
//     pf_sebelum = 0;
//     pf_setelah = 0;
//
//   }
//   else {
//     digitalWrite(cap20, LOW);
//     digitalWrite(cap10, LOW);
//     digitalWrite(cap5, LOW);
//     pf_setelah = pzem.pf() + (0.98 - pzem.pf());
//     Serial.println("nan uf");
//   }
//  }
//  else {
//    if (pf > 0.01 and pf < 0.98) {
//      pf_sebelum = pf;
//    }
//  }
//}
void firebaseSendData() {
  if ((millis() - backTimer) > 7000) {
    Firebase.setFloat(fbdo, F("/tegangan"), v) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/arus"), i) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/daya"), w) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/energi"), e) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/frekuensi"), f) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/pf_before"), pf_sebelum) ? "ok" : fbdo.errorReason().c_str();
    Firebase.setFloat(fbdo, F("/pf_after"), pf_setelah) ? "ok" : fbdo.errorReason().c_str();
    backTimer = millis();
```

```
 }
}
void firebaseInitial() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    lcd.setCursor(0, 0);
    lcd.print(WIFI_SSID);
    lcd.setCursor(0, 1);
    lcd.print(WIFI_PASSWORD);
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.printf("Firebase Client v%s\n\n", FIREBASE_CLIENT_VERSION);
  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;

  auth.user.password = USER_PASSWORD;

  config.database_url = DATABASE_URL;

  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

  Firebase.begin(&config, &auth);

  Firebase.reconnectWiFi(true);

  Firebase.setDoubleDigits(5);

}
```

LAMPIRAN B

Gambar Dokumentasi

	Perancangan Perangkat Keras
	Pengukuran dengan alat ukur cosphi analog
	Tampak Keseluruhan Alat

	Tampak Samping Pada Alat
	Tampak Depan Pada Alat
	Tampak Dalam Pada Alat