

## LAMPIRAN A

### PROGRAM PENGENDALIAN DAN MONITORING KOTAK KONTAK BERBASIS IOT

#### A. Mitapp

```
#include <Wire.h>
#include <RtcDS3231.h>
#include <RBDdimmer.h>
#include <PZEM004Tv30.h>
#include <LiquidCrystal_I2C.h>
RtcDS3231<TwoWire> Rtc(Wire);

#include <WiFi.h>
#include <FirebaseESP32.h>

#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
#define WIFI_SSID "p"
#define WIFI_PASSWORD "qwertyuiop"
#define API_KEY
"AIzaSyByrGgb7iPxpZE9_16BmNJLzEzPVvU5YE"
#define DATABASE_URL "smart-stopkontak-default-
rtdb.firebaseio.com"
#define USER_EMAIL "stopkontaksmart@gmail.com"
#define USER_PASSWORD "smartstopkontak2022"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

#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
#define NUM_PZEMS 4
PZEM004Tv30 pzems[NUM_PZEMS];
#if defined(USE_SOFTWARE_SERIAL) &&
defined(ESP32)
#error "Can not use SoftwareSerial with ESP32"
#elif defined(USE_SOFTWARE_SERIAL)
#include <SoftwareSerial.h>
SoftwareSerial pzemSWSerial(PZEM_RX_PIN,
PZEM_TX_PIN);
#endif

```

```

#define outputPin1 2
#define outputPin2 4
#define zerocross 15
#define r1 26
#define r2 25
#define r3 33
#define r4 32

```

```

LiquidCrystal_I2C lcd(0x27, 20, 4);
dimmerLamp dimmer1(outputPin1, zerocross );
dimmerLamp dimmer2(outputPin2, zerocross );

```

```

int flagRun, timeShow, jeda;
int detik, menit, jam; //for rtc
int v, v1, v2, v3;
float a, p, e, f, pf ;
float a1 , p1, e1, f1 , pf1;
float a2, p2 , e2, f2, pf2;

```

```

float a3 , p3 , e3 , f3 , pf3;
String mod1, mod2, mod3, mod4;
String sp_energy1 , sp_energy2 , sp_energy3 , sp_energy4 ,
sp_timer1 , sp_timer2 , sp_timer3 , sp_timer4 ;
String sp_arus1 , sp_arus2 , sp_arus3 , sp_arus4 , manual1,
manual2, manual3 , manual4 , dimer1 , dimer2, mode;
int swmode, man1, man2, man3 , man4, dim1, dim2;
int energy1 , energy2 , energy3 , energy4 , timer1 , timer2 ,
timer3 , timer4, arus1 , arus2 , arus3, arus4 ;
int mode0, mode1, mode2, mode3;
long proveus, counttt, proveus0, counttime1, counttime2,
counttime3, counttime4;
String spjamnyala, spmenitnyala, spjammati, spmenitmati,
spjamnyala1, spmenitnyala1, spjammati1, spmenitmati1,
spjamnyala2, spmenitnyala2, spjammati2, spmenitmati2,
spjamnyala3, spmenitnyala3, spjammati3, spmenitmati3;
unsigned long sendDataPrevMillis = 0;

void setup() {
  Serial.begin(57600);
  setupRtc();
  pzemsetup();
  dimerssetup();
  lcdsetup();
  firebasessetup();
  pinMode(r1, OUTPUT);
  pinMode(r2, OUTPUT);
  pinMode(r3, OUTPUT);
  pinMode(r4, OUTPUT);
  dimmer1.setPower(100);
  dimmer2.setPower(100);
  delay(2000);
  lcd.clear();
  counttime1 = 0;

```

```

    countime2 = 0;
    countime3 = 0;
    countime4 = 0;
}

void loop() {
  loopRTC();
  readzem();
  show();
  firebaseSet();
  mainn();
  dimmer1.setPower(dim1);
  dimmer2.setPower(dim2);
  sermon();
}

```

#### B. DIMMER

```

void dimersetup() {
  dimmer1.begin(NORMAL_MODE, ON); //dimmer
  initialisation: name.begin(MODE, STATE)
  dimmer2.begin(NORMAL_MODE, ON);
}

```

#### C. DISPLAY

```

void lcdsetup() {
  lcd.init();
  lcd.backlight();
}

```

```

void show() {
  if (millis() - timeShow > 3000) {
    lcd.init();
    lcd.backlight();
    delay(50);
    lcd.clear();
  }
}

```

```
delay(500);  
lcd.setCursor(0,0);  
lcd.print("Processing...");  
flagRun++;  
timeShow = millis();  
}
```

```
switch (flagRun) {  
  case 0:  
    delay(50);  
    flagRun = 1;  
    break;  
  case 1:  
    lcd.setCursor(0, 0);  
    lcd.print("  PZEM 1  ");  
    lcd.setCursor(0, 1);  
    lcd.print(String(v) + " V");  
    lcd.setCursor(0, 2);  
    lcd.print(String(a, 1) + " A");  
    lcd.setCursor(0, 3);  
    lcd.print(String(e, 1) + " KWh");  
    lcd.setCursor(10, 1);  
    lcd.print(String(p, 1) + " W");  
    lcd.setCursor(10, 2);  
    lcd.print(String(pf, 1) + " pf");  
    lcd.setCursor(10, 3);  
    lcd.print(String(f, 1) + " Hz");  
    break;  
  
  case 2:  
    lcd.setCursor(0, 0);  
    lcd.print("  PZEM 2  ");  
    lcd.setCursor(0, 1);  
    lcd.print(String(int(v1)) + " V");
```

```
lcd.setCursor(0, 2);  
lcd.print(String(a1, 1) + " A");  
lcd.setCursor(0, 3);  
lcd.print(String(e1, 1) + " KWh");  
lcd.setCursor(10, 1);  
lcd.print(String(p1, 1) + " W");  
lcd.setCursor(10, 2);  
lcd.print(String(pf1, 1) + " pf");  
lcd.setCursor(10, 3);  
lcd.print(String(f1, 1) + " Hz");  
break;
```

case 3:

```
lcd.setCursor(0, 0);  
lcd.print(" PZEM 3 ");  
lcd.setCursor(0, 1);  
lcd.print(String(v2) + " V");  
lcd.setCursor(0, 2);  
lcd.print(String(a2, 1) + " A");  
lcd.setCursor(0, 3);  
lcd.print(String(e2, 1) + " KWh");  
lcd.setCursor(10, 1);  
lcd.print(String(p2, 1) + " W");  
lcd.setCursor(10, 2);  
lcd.print(String(pf2, 1) + " pf");  
lcd.setCursor(10, 3);  
lcd.print(String(f2, 1) + " Hz");  
break;
```

case 4:

```
lcd.setCursor(0, 0);  
lcd.print(" PZEM 4 ");  
lcd.setCursor(0, 1);  
lcd.print(String(v3) + " V");
```

```

lcd.setCursor(0, 2);
lcd.print(String(a3, 1) + " A");
lcd.setCursor(0, 3);
lcd.print(String(e3, 1) + " KWh");
lcd.setCursor(10, 1);
lcd.print(String(p3, 1) + " W");
lcd.setCursor(10, 2);
lcd.print(String(pf3, 1) + " pf");
lcd.setCursor(10, 3);
lcd.print(String(f3, 1) + " Hz");
break;

```

case 5:

```

delay(500);
flagRun = 0;
break;

```

```

}

```

```

}

```

#### D. FIREBASE

```

void firebaseSetup() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    lcd.setCursor(0, 0);
    lcd.print("connecting...");
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}

```

```

lcd.setCursor(0, 0);
lcd.print(WiFi.localIP());
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.setStringDigits(2);

// Firebase.setString(fbdo, F("/sp manual1"), "0");
// Firebase.setString(fbdo, F("/sp manual2"), "0");
// Firebase.setString(fbdo, F("/sp manual3"), "0");
// Firebase.setString(fbdo, F("/sp manual4"), "0");
//
// Firebase.setString(fbdo, F("/sp arus1"), "0");
// Firebase.setString(fbdo, F("/sp arus2"), "0");
// Firebase.setString(fbdo, F("/sp arus3"), "0");
// Firebase.setString(fbdo, F("/sp arus4"), "0");
// Firebase.setString(fbdo, F("/mode"), "0");
}

void firebaseSet() {
v = pzems[0].voltage();
a = pzems[0].current();
p = pzems[0].power();
e = pzems[0].energy();
f = pzems[0].frequency();
pf = pzems[0].pf();
}

```



```

if (counttt == 0) {
    Firebase.setString(fbdo, F("/tegangan0"), String(v));
    counttt = 1;
}
else if (counttt == 1) {
    Firebase.setString(fbdo, F("/arus0"), String(a, 2));
    counttt = 2;
}
else if (counttt == 2) {
    Firebase.setString(fbdo, F("/P1"), String(p, 2));
    counttt = 3;
}
else if (counttt == 3) {
    Firebase.setString(fbdo, F("/energi0"), String(e, 2));
    counttt = 4;
}
else if (counttt == 4) {
    Firebase.setString(fbdo, F("/fekuensi0"), String(f, 1));
    counttt = 5;
}
else if (counttt == 5) {
    Firebase.setString(fbdo, F("/pf1"), String(pf, 2));
    counttt = 6;
}

else if (counttt == 6) {
    Firebase.setString(fbdo, F("/tegangan1"), String(v1));
    counttt = 7;
}
else if (counttt == 7) {
    Firebase.setString(fbdo, F("/arus1"), String(a1, 2));
    counttt = 8;
}
else if (counttt == 8) {

```

```

    Firebase.setString(fbdo, F("/P2"), String(p1, 2));
    counttt = 9;
}
else if (counttt == 9) {
    Firebase.setString(fbdo, F("/energi1"), String(e1, 2));
    counttt = 10;
}
else if (counttt == 10) {
    Firebase.setString(fbdo, F("/fekuensi1"), String(f1, 1));
    counttt = 11;
}
else if (counttt == 11) {
    Firebase.setString(fbdo, F("/pf2"), String(pf1, 2));
    counttt = 12;
}

else if (counttt == 12) {
    Firebase.setString(fbdo, F("/tegangan2"), String(v2));
    counttt = 13;
}
else if (counttt == 13) {
    Firebase.setString(fbdo, F("/arus2"), String(a2, 2));
    counttt = 14;
}
else if (counttt == 14) {
    Firebase.setString(fbdo, F("/P3"), String(p2, 2));
    counttt = 15;
}
else if (counttt == 15) {
    Firebase.setString(fbdo, F("/energi2"), String(e2, 2));
    counttt = 16;
}
else if (counttt == 16) {
    Firebase.setString(fbdo, F("/fekuensi2"), String(f2, 1));

```

```

    counttt = 17;
}
else if (counttt == 17) {
    Firebase.setString(fbdo, F("/pf3"), String(pf2, 2));
    counttt = 18;
}

else if (counttt == 18) {
    Firebase.setString(fbdo, F("/tegangan3"), String(v3));
    counttt = 19;
}
else if (counttt == 19) {
    Firebase.setString(fbdo, F("/arus3"), String(a3, 2));
    counttt = 20;
}
else if (counttt == 20) {
    Firebase.setString(fbdo, F("/P4"), String(p3, 2));
    counttt = 21;
}
else if (counttt == 21) {
    Firebase.setString(fbdo, F("/energi3"), String(e3, 2));
    counttt = 22;
}
else if (counttt == 22) {
    Firebase.setString(fbdo, F("/fekuensi3"), String(f3, 1));
    counttt = 23;
}
else if (counttt == 23) {
    Firebase.setString(fbdo, F("/pf4"), String(pf3, 2));
    counttt = 0;
}
}
}
E. MAIN
void mainn() {

```

```

if (jeda == 0) {
//  firebasesetup();
    jeda = 1;
}

if (jeda == 1) {
    mod1 = Firebase.getString(fbdo, F("/mode0")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 2;
}
else if (jeda == 2) {
    mod2 = Firebase.getString(fbdo, F("/mode1")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 3;
}
else if (jeda == 3) {
    mod3 = Firebase.getString(fbdo, F("/mode2")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 4;
}
else if (jeda == 4) {
    mod4 = Firebase.getString(fbdo, F("/mode3")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 5;
}
else if (jeda == 5) {
    dimer1 = Firebase.getString(fbdo, F("/dimer3")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 6;
}
else if (jeda == 6) {
    dimer2 = Firebase.getString(fbdo, F("/dimer4")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    jeda = 1;
}

```

```
}

mode0 = mod1.toInt();
mode1 = mod2.toInt();
mode2 = mod3.toInt();
mode3 = mod4.toInt();

dim1 = dimer1.toInt();
dim2 = dimer2.toInt();

energy1 = sp_energy1.toInt();
energy2 = sp_energy2.toInt();
energy3 = sp_energy3.toInt();
energy4 = sp_energy4.toInt();

timer1 = sp_timer1.toInt();
timer2 = sp_timer2.toInt();
timer3 = sp_timer3.toInt();
timer4 = sp_timer4.toInt();

arus1 = sp_arus1.toInt();
arus2 = sp_arus2.toInt();
arus3 = sp_arus3.toInt();
arus4 = sp_arus4.toInt();

if (dim1 < 40) {
    dim1 = 40;
}
else if (dim1 > 100) {
    dim1 = 100;
}

if (dim2 < 40) {
    dim2 = 40;
}
```

```

}
else if (dim2 > 100) {
    dim2 = 100;
}

//=====
=====

if (mode0 == 0) {
    manual1 = Firebase.getString(fbdo, F("/manual0")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    man1 = manual1.toInt();

    if (man1 == 1) {
        Firebase.setString(fbdo, F("/status0"), "1");
        digitalWrite(r1, HIGH);
    }
    else if (man1 == 0) {
        Firebase.setString(fbdo, F("/status0"), "0");
        digitalWrite(r1, LOW);
    }
}

else if (mode0 == 1) {
    spjamnyala = Firebase.getString(fbdo, F("/spjamnyala"))
? fbdo.to<const char *>() : fbdo.errorReason().c_str();
    spmenitnyala = Firebase.getString(fbdo,
F("/spmenitnyala")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spjammati = Firebase.getString(fbdo, F("/spjammati")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    spmenitmati = Firebase.getString(fbdo,
F("/spmenitmati")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();

```

```

    if (jam == spjamnyala.toInt() && menit ==
spmenitnyala.toInt()) {
        Firebase.setString(fbdo, F("/status0"), "1");
        digitalWrite(r1, HIGH);
    }
    else if (jam == spjammati.toInt() && menit ==
spmenitmati.toInt()) {
        Firebase.setString(fbdo, F("/status0"), "0");
        digitalWrite(r1, LOW);
    }
}

else if (mode0 == 2) {
    sp_arus1 = Firebase.getString(fbdo, F("/setampere1")) ?
fbdo.to<const char *>(): fbdo.errorReason().c_str();

    if (arus1 != 0) {
        if (a > arus1) {
            Firebase.setString(fbdo, F("/status0"), "0");
            Firebase.setString(fbdo, F("/sp arus1"), "0");
            digitalWrite(r1, LOW);
        }
        else {
            Firebase.setString(fbdo, F("/status0"), "1");
            digitalWrite(r1, HIGH);
        }
    }
    else {
        Firebase.setString(fbdo, F("/status0"), "0");
        digitalWrite(r1, LOW);
    }
}
}

```

```

else if (mode0 == 3) {
    sp_energy1 = Firebase.getString(fbdo, F("/setKWh1")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

```

```

if (energy1 != 0) {
    if (e > energy1) {
        Firebase.setString(fbdo, F("/status0"), "0");
        Firebase.setString(fbdo, F("/sp energy1"), "0");
        digitalWrite(r1, LOW);
    }
    else {
        Firebase.setString(fbdo, F("/status0"), "1");
        digitalWrite(r1, HIGH);
    }
}
else {
    Firebase.setString(fbdo, F("/status0"), "0");
    digitalWrite(r1, LOW);
}
}

```

```

//=====
=====

```

```

if (mode1 == 0) {
    manual2 = Firebase.getString(fbdo, F("/manual1")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    man2 = manual2.toInt();

```

```

if (man2 == 1) {
    Firebase.setString(fbdo, F("/status1"), "1");
    digitalWrite(r2, HIGH);
}

```



```

else if (man2 == 0) {
    Firebase.setString(fbdo, F("/status1"), "0");
    digitalWrite(r2, LOW);
}
}

else if (mode1 == 1) {
    spjamnyala1 = Firebase.getString(fbdo,
F("/spjamnyala1")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spmenitnyala1 = Firebase.getString(fbdo,
F("/spmenitnyala1")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spjammati1 = Firebase.getString(fbdo, F("/spjammati1"))
? fbdo.to<const char *>() : fbdo.errorReason().c_str();
    spmenitmati1 = Firebase.getString(fbdo,
F("/spmenitmati1")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();

    if (jam == spjamnyala1.toInt() && menit ==
spmenitnyala1.toInt()) {
        Firebase.setString(fbdo, F("/status1"), "1");
        digitalWrite(r2, HIGH);
    }
    else if (jam == spjammati1.toInt() && menit ==
spmenitmati1.toInt()) {
        Firebase.setString(fbdo, F("/status1"), "0");
        digitalWrite(r2, LOW);
    }
}

else if (mode1 == 2) {
    sp_arus2 = Firebase.getString(fbdo, F("/setampere2")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

```

```

if (arus2 != 0) {
  if (a1 > arus2) {
    Firebase.setString(fbdo, F("/status1"), "0");
    Firebase.setString(fbdo, F("/sp arus2"), "0");
    digitalWrite(r2, LOW);
  }
  else {
    Firebase.setString(fbdo, F("/status1"), "1");
    digitalWrite(r2, HIGH);
  }
}
else {
  Firebase.setString(fbdo, F("/status1"), "1");
  digitalWrite(r2, LOW);
}
}

else if (mode1 == 3) {
  sp_energy2 = Firebase.getString(fbdo, F("/setKWh2")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

if (energy2 != 0) {
  if (e1 > energy2) {
    Firebase.setString(fbdo, F("/status1"), "0");
    Firebase.setString(fbdo, F("/sp energy2"), "0");
    digitalWrite(r2, LOW);
  }
  else {
    Firebase.setString(fbdo, F("/status1"), "1");
    digitalWrite(r2, HIGH);
  }
}
else {

```

```

        Firebase.setString(fbdo, F("/status1"), "0");
        digitalWrite(r2, LOW);
    }
}

//=====
=====

if (mode2 == 0) {
    manual3 = Firebase.getString(fbdo, F("/manual2")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    man3 = manual3.toInt();

    if (man3 == 1) {
        Firebase.setString(fbdo, F("/status2"), "1");
        digitalWrite(r3, HIGH);
    }
    else if (man3 == 0) {
        Firebase.setString(fbdo, F("/status2"), "0");
        digitalWrite(r3, LOW);
    }
}

else if (mode2 == 1) {
    spjamnyala2 = Firebase.getString(fbdo,
F("/spjamnyala2")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spmenitnyala2 = Firebase.getString(fbdo,
F("/spmenitnyala2")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spjammati2 = Firebase.getString(fbdo, F("/spjammati2"))
? fbdo.to<const char *>() : fbdo.errorReason().c_str();
    spmenitmati2 = Firebase.getString(fbdo,

```

```

F("/spmenitmati2")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();

    if (jam == spjamnyala2.toInt() && menit ==
spmenitnyala2.toInt()) {
        Firebase.setString(fbdo, F("/status2"), "1");
        digitalWrite(r3, HIGH);
    }
    else if (jam == spjammati2.toInt() && menit ==
spmenitmati2.toInt()) {
        Firebase.setString(fbdo, F("/status2"), "0");
        digitalWrite(r3, LOW);
    }
}

else if (mode2 == 2) {
    sp_arus3 = Firebase.getString(fbdo, F("/setampere3")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

    if (arus3 != 0) {
        if (a2 > arus3) {
            Firebase.setString(fbdo, F("/status2"), "0");
            Firebase.setString(fbdo, F("/sp arus3"), "0");
            digitalWrite(r3, LOW);
        }
        else {
            Firebase.setString(fbdo, F("/status2"), "1");
            digitalWrite(r3, HIGH);
        }
    }
    else {
        Firebase.setString(fbdo, F("/status2"), "0");
        digitalWrite(r3, LOW);
    }
}

```

```

}

else if (mode2 == 3) {
    sp_energy3 = Firebase.getString(fbdo, F("/setKWh3")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

    if (energy3 != 0) {
        if (e2 > energy3) {
            Firebase.setString(fbdo, F("/status2"), "0");
            Firebase.setString(fbdo, F("/sp energy3"), "0");
            digitalWrite(r3, LOW);
        }
        else {
            Firebase.setString(fbdo, F("/status2"), "1");
            digitalWrite(r3, HIGH);
        }
    }
    else {
        Firebase.setString(fbdo, F("/status2"), "0");
        digitalWrite(r3, LOW);
    }
}

//=====
==

if (mode3 == 0) {
    manual4 = Firebase.getString(fbdo, F("/manual3")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();
    man4 = manual4.toInt();

    if (man4 == 1) {
        Firebase.setString(fbdo, F("/status3"), "1");
    }
}

```

```

        digitalWrite(r4, HIGH);
    }
    else if (man4 == 0) {
        Firebase.setString(fbdo, F("/status3"), "0");
        digitalWrite(r4, LOW);
    }
}

else if (mode3 == 1) {
    spjamnyala3 = Firebase.getString(fbdo,
F("/spjamnyala3")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spmenitnyala3 = Firebase.getString(fbdo,
F("/spmenitnyala3")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();
    spjammati3 = Firebase.getString(fbdo, F("/spjammati3"))
? fbdo.to<const char *>() : fbdo.errorReason().c_str();
    spmenitmati3 = Firebase.getString(fbdo,
F("/spmenitmati3")) ? fbdo.to<const char *>() :
fbdo.errorReason().c_str();

    if (jam == spjamnyala3.toInt() && menit ==
spmenitnyala3.toInt()) {
        Firebase.setString(fbdo, F("/status3"), "1");
        digitalWrite(r1, HIGH);
    }
    else if (jam == spjammati3.toInt() && menit ==
spmenitmati3.toInt()) {
        Firebase.setString(fbdo, F("/status3"), "0");
        digitalWrite(r1, LOW);
    }
}

else if (mode3 == 2) {

```

```

    sp_arus4 = Firebase.getString(fbdo, F("/setampere4")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

    if (arus4 != 0) {
        if (a3 > arus4) {
            Firebase.setString(fbdo, F("/status3"), "0");
            Firebase.setString(fbdo, F("/sp arus4"), "0");
            digitalWrite(r4, LOW);
        }
        else {
            Firebase.setString(fbdo, F("/status3"), "1");
            digitalWrite(r4, HIGH);
        }
    }
    else {
        Firebase.setString(fbdo, F("/status3"), "0");
        digitalWrite(r4, LOW);
    }
}

else if (mode3 == 3) {
    sp_energy4 = Firebase.getString(fbdo, F("/setKWh4")) ?
fbdo.to<const char *>() : fbdo.errorReason().c_str();

    if (energy4 != 0) {
        if (e3 > energy4) {
            Firebase.setString(fbdo, F("/status3"), "0");
            Firebase.setString(fbdo, F("/sp energy4"), "0");
            digitalWrite(r4, LOW);
        }
        else {
            Firebase.setString(fbdo, F("/status3"), "1");
            digitalWrite(r4, HIGH);
        }
    }
}

```

```

    }
    else {
        Firebase.setString(fbdo, F("/status3"), "0");
        digitalWrite(r4, LOW);
    }
}
}
}
F. PZEM
void pzemsetup() {
    for (int i = 0; i < NUM_PZEMS; i++)
    {

#ifdef USE_SOFTWARE_SERIAL
        pzems[i] = PZEM004Tv30(pzemSWSerial, 0x56 + i);
#elif defined(ESP32)
        pzems[i] = PZEM004Tv30(PZEM_SERIAL,
        PZEM_RX_PIN, PZEM_TX_PIN, 0x56 + i);
#else
        pzems[i] = PZEM004Tv30(PZEM_SERIAL, 0x56 + i);
#endif
    }
}

void readzem() {
    v = pzems[0].voltage();
    a = pzems[0].current();
    p = pzems[0].power();
    e = pzems[0].energy();
    f = pzems[0].frequency();
    pf = pzems[0].pf();

    v1 = pzems[1].voltage();
    a1 = pzems[1].current();
    p1 = pzems[1].power();

```



```
e1 = pzems[1].energy();
f1 = pzems[1].frequency();
pf1 = pzems[1].pf();
```

```
v2 = pzems[2].voltage();
a2 = pzems[2].current();
p2 = pzems[2].power();
e2 = pzems[2].energy();
f2 = pzems[2].frequency();
pf2 = pzems[2].pf();
```

```
v3 = pzems[3].voltage();
a3 = pzems[3].current();
p3 = pzems[3].power();
e3 = pzems[3].energy();
f3 = pzems[3].frequency();
pf3 = pzems[3].pf();
```

```
if (isnan(a)) {
    v = 0;
    a = 0;
    p = 0;
    e = 0;
    f = 0;
    pf = 0;
}
```

```
if (isnan(a1)) {
    v1 = 0;
    a1 = 0;
    p1 = 0;
    e1 = 0;
    f1 = 0;
    pf1 = 0;
```

```
}  
  
if (isnan(a2)) {  
    v2 = 0;  
    a2 = 0;  
    p2 = 0;  
    e2 = 0;  
    f2 = 0;  
    pf2 = 0;  
}
```

```
if (isnan(a3)) {  
    v3 = 0;  
    a3 = 0;  
    p3 = 0;  
    e3 = 0;  
    f3 = 0;  
    pf3 = 0;  
}  
}
```

#### G. RTC

```
void setupRtc() {  
    Rtc.Begin();  
  
    RtcDateTime compiled = RtcDateTime(__DATE__,  
    __TIME__);  
    printDateTime(compiled);  
    Serial.println();  
    if (!Rtc.IsDateTimeValid())  
    {  
        if (Rtc.LastError() != 0) {  
            Serial.print("RTC communications error = ");  
            Serial.println(Rtc.LastError());  
        }  
    }  
}
```

```

else {
    Serial.println("RTC lost confidence in the DateTime!");
    Rtc.SetDateTime(compiled);
}
}

if (!Rtc.GetIsRunning()) {
    Serial.println("RTC was not actively running, starting
now");
    Rtc.SetIsRunning(true);
}
RtcDateTime now = Rtc.GetDateTime();
if (now < compiled) {
    Serial.println("RTC is older than compile time!
(Updating DateTime)");
    Rtc.SetDateTime(compiled);
}
else if (now > compiled) {
    Serial.println("RTC is newer than compile time. (this is
expected)");
}
else if (now == compiled) {
    Serial.println("RTC is the same as compile time! (not
expected but all is fine)");
}
Rtc.Enable32kHzPin(false);

Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);

}

void loopRTC() {
    RtcDateTime now = Rtc.GetDateTime();

```

```

printDateTime(now);
jam = now.Hour();//jam saat ini
menit = now.Minute();//menit saat ini
detik = now.Second();//detik saat ini
}

```

```

#define countof(a) (sizeof(a) / sizeof(a[0]))

```

```

void printDateTime(const RtcDateTime& dt)
{
char datestring[20];

snprintf_P(datestring,
            countof(datestring),
            PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
            dt.Month(),
            dt.Day(),
            dt.Year(),
            dt.Hour(),
            dt.Minute(),
            dt.Second() );
Serial.println(datestring);
}

```

#### H. SERIAL MONITOR

```

void sermon() {
Serial.println();
Serial.print(" mode0: " + String(mode0));
Serial.print(" mode1: " + String(mode1));
Serial.print(" mode2: " + String(mode2));
Serial.println(" mode3: " + String(mode3));
Serial.print(" man1: " + String(man1));
Serial.print(" man2: " + String(man2));
}

```

```
Serial.print(" man3: " + String(man3));
Serial.println(" man4: " + String(man4));
Serial.print(" energy1: " + String(energy1));
Serial.print(" energy2: " + String(energy2));
Serial.print(" energy3: " + String(energy3));
Serial.println(" energy4: " + String(energy4));
Serial.print(" timer1: " + String(timer1));
Serial.print(" timer2: " + String(timer2));
Serial.print(" timer3: " + String(timer3));
Serial.println(" timer4: " + String(timer4));
Serial.print(" arus1: " + String(arus1));
Serial.print(" arus2: " + String(arus2));
Serial.print(" arus3: " + String(arus3));
Serial.println(" arus4: " + String(arus4));
Serial.print(" countime1: " + String(countime1));
Serial.print(" countime2: " + String(countime2));
Serial.print(" countime3: " + String(countime3));
Serial.println(" countime4: " + String(countime4));
Serial.print(" sp_arus1: " + String(sp_arus1));
Serial.print(" sp_arus2: " + String(sp_arus2));
Serial.print(" sp_arus3: " + String(sp_arus3));
Serial.println(" sp_arus4: " + String(sp_arus4));
Serial.print(" sp_energy1: " + String(sp_energy1));
Serial.print(" sp_energy2: " + String(sp_energy2));
Serial.print(" sp_energy3: " + String(sp_energy3));
Serial.println(" sp_energy4: " + String(sp_energy4));
Serial.print(" sp_jamnyala1: " + String(spjamnyala));
Serial.print(" sp_menitnyala1: " + String(spmenitnyala));
Serial.print(" sp_jammati1: " + String(spjammati));
Serial.println(" sp_menitmati2: " + String(spmenitmati));
Serial.print(" sp_jamnyala2: " + String(spjamnyala1));
Serial.print(" sp_menitnyala2: " + String(spmenitnyala1));
Serial.print(" sp_jammati2: " + String(spjammati1));
Serial.println(" sp_menitmati2: " + String(spmenitmati1));
```

```
Serial.print(" sp_jamnyala3: " + String(spjamnyala2));
Serial.print(" sp_menitnyala3: " + String(spmenitnyala2));
Serial.print(" sp_jammati3: " + String(spjammati2));
Serial.println(" sp_menitmati3: " + String(spmenitmati2));
Serial.print(" sp_jamnyala4: " + String(spjamnyala3));
Serial.print(" sp_menitnyala4: " + String(spmenitnyala3));
Serial.print(" sp_jammati4: " + String(spjammati3));
Serial.println(" sp_menitmati4: " + String(spmenitmati3));
Serial.print(" dim1: " + String(dim1));
Serial.println(" dim2: " + String(dim2));
}
```

# LAMPIRAN B

## Desain Sistem

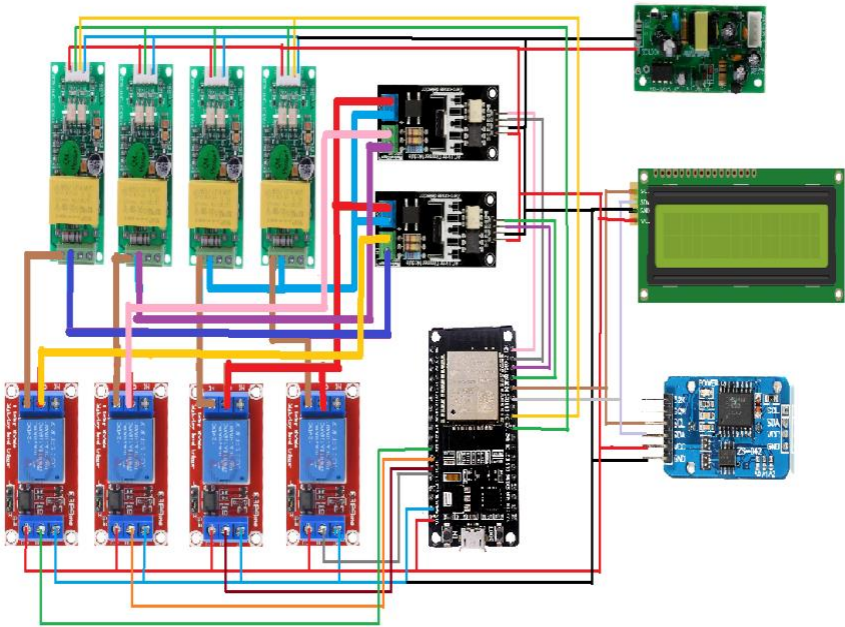






# LAMPIRAN C

## Rangkaian Wiring





## LAMPIRAN D

### Gambar Alat

