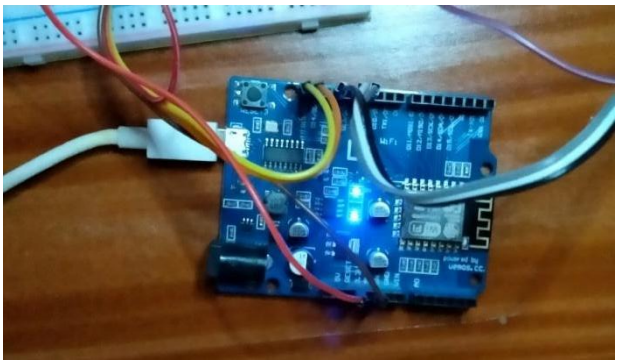
# LAMPIRAN

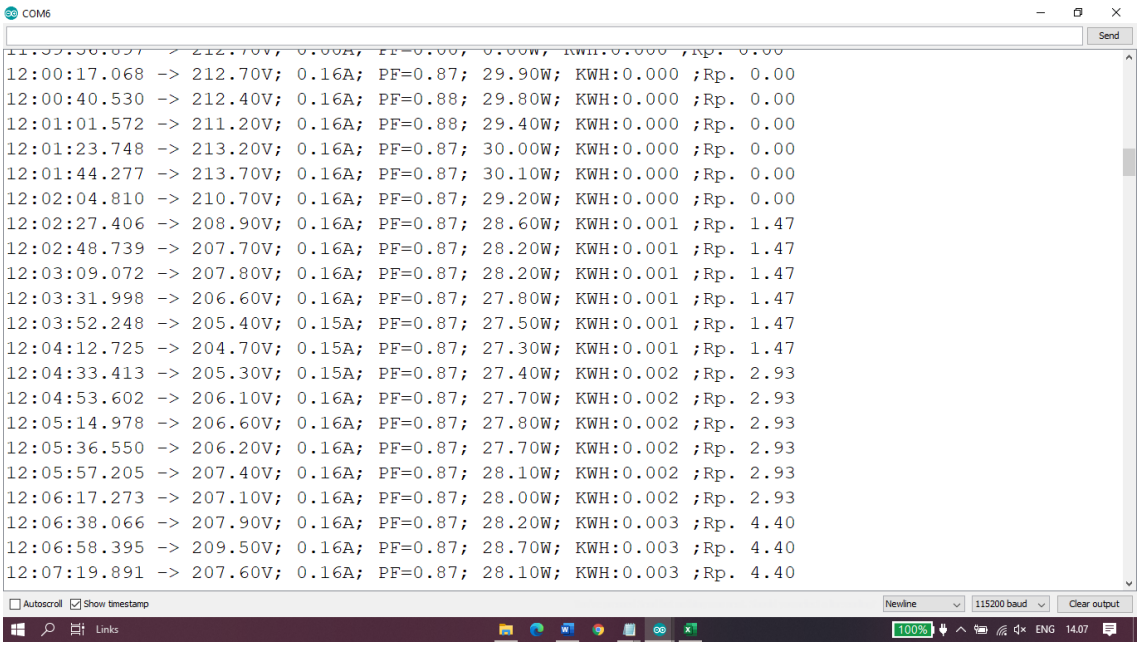
# LAMPIRAN A
# DOKUMENTASI PENELITIAN



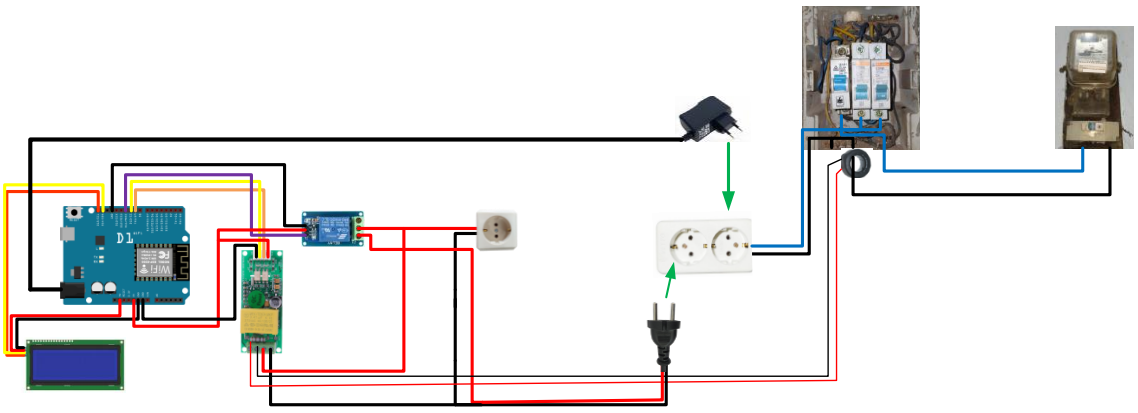
Gambar A. 1 Pengujian Wemos D1R1

Gambar A. 2 Pengujian Relay dan LCD


Gambar A. 3 Pengujian Alat Monitoring




Gambar A. 4 Tampilan Data pada Arduino IDE



Gambar A. 5 Pengujian pada Beban Listrik Rumah Tangga

Mulai
Inisialisai *IoT device*
Wemos D1R1 menghubungkan ke internet
Koneksi berhasil?
Ya
Tidak
B
Relay aktif
Sensor PZEM004T membaca tegangan, arus, daya dan energi
Wemos D1R1 mengolah data dari sensor
Kirim data ke database di Googlesheet
Menampilkan pada LCD
Selesai
Pengguna
Membuka bot telegram untuk alat monitoring kendali
Menekan tombol "Start" atau ketik /start pada bot telegram
Bot telegram menampilkan menu pesan perintah monitoring kendali
Mengirimkan pesan perintah dengan cara menekan menu atau ketik pesan perintah
Menerima pesan balasan telegram dari Wemos D1R1
Menekan url realtime database pada pesan balasan
Menampilkan halaman database pada googlesheet
Menampilkan dan menyimpan data monitoring secara realtime
Sistem Monitoring
Mengirimkan pesan perintah monitoring
Pesan diterima Wemos D1R1?
Ya
Tidak
Sensor PZEM004T membaca parameter sesuai pesan perintah
Menampilkan data sesuai pesan perintah ke telegram
Sistem Kendali
Mengirimkan pesan perintah kendali : •/relayon •/relayoff
Pesan diterima Wemos D1R1?
Ya
Tidak
Menyalakan atau mematikan relay
Memberikan pesan balasan ke telegram jika perintah telah berhasil dijalankan
Selesai

Tabel A- 1 Library yang digunakan pada Arduino IDE

| No. | *Library* | Deskripsi |
|---|---|---|
| 1 | #include <TRIGGER_WIFI.h> | Mengimport *library* TRIGGER_WIFI, yang digunakan untuk mengirimkan data daya listrik ke bot Telegram dan *Google Sheets.* |
| 2 | #include <TRIGGER_*GOOGLE SHEETS*.h> | Mengimport *library* TRIGGER_*GOOGLE SHEETS*, yang digunakan untuk menyimpan data daya listrik di *Google Sheets*. |
| 3 | #include <ESP8266WiFi.h> | Mengimport *library* ESP8266WiFi, yang digunakan untuk koneksi WiFi. |
| 4 | #include <PZEM004Tv30.h> | Mengimport *library* PZEM004Tv30, yang digunakan untuk membaca data daya listrik dari modul PZEM004Tv30. |
| 5 | #include <LiquidCrystal_I2C.h> | Mengimport *library* LiquidCrystal_I2C, yang digunakan untuk menampilkan data daya listrik di LCD. |
| 6 | #include <Wire.h> | Mengimport *library* Wire, yang digunakan untuk komunikasi I2C. |
| 7 | #include <WiFiClientSecure.h> | Mengimport *library* WiFiClientSecure, yang digunakan untuk koneksi WiFi yang aman. |
| 8 | #include <Firebase.h> | Mengimport *library* Firebase, yang digunakan untuk menyimpan data daya listrik di Firebase. |
| 9 | #include <TimeLib.h> | Mengimport *library* TimeLib, yang digunakan untuk manajemen waktu. |
| 10 | #include <WiFiUdp.h> | Mengimport *library* WiFiUdp, yang digunakan untuk koneksi WiFi UDP. |
| 11 | #include <NTPClient.h> | Mengimport *library* NTPClient, yang digunakan untuk klien NTP. |
| 12 | #include <UniversalTelegramBot.h> | Mengimport *library* UniversalTelegramBot, yang digunakan untuk berinteraksi dengan bot Telegram. |

Tabel A- 2 Variabel yang digunakan pada Arduino IDE

| Konstanta/Variabel | Deskripsi |
|---|---|
| RX dan TX | Pin yang digunakan untuk komunikasi serial dengan modul PZEM004Tv30. |
| rl | Pin yang digunakan untuk mengendalikan relay. |
| FIREBASE_HOST, FIREBASE_AUTH | Kredensial yang digunakan untuk mengakses Firebase. |
| ssid1, password1, ssid2, dan password2 | SSID dan password untuk dua jaringan WiFi yang mungkin digunakan. |
| ntpServer | Nama server NTP yang digunakan untuk sinkronisasi waktu. |
| column_name_in_sheets | Nama kolom di *Google Sheets* tempat data daya listrik akan disimpan. |
| Sheets_GAS_ID | ID spreadsheet *Google Sheets* yang digunakan untuk menyimpan data. |
| No_of_Parameters | Jumlah parameter yang akan dikirim ke *Google Sheets*. |
| BOTtoken | Token bot Telegram yang digunakan untuk berinteraksi dengan bot Telegram. |
| CHAT_ID | ID chat bot Telegram yang digunakan untuk mengirim dan menerima pesan. |

# LAMPIRAN B

# DATA PENELITIAN

Tabel B- 1 Rekapitulasi Data Pengujian selama 1 Bulan (25 Juni 2023-25 Juli 2023)

| Tanggal | Tegangan | Arus | Daya | kWh | Biaya (Rp) |
|---|---|---|---|---|---|
| 25/06/2023 | 192.80 | 1.88 | 317.20 | 0,00 | Rp0 |
| 26/06/2023 | 201.20 | 1.38 | 243.50 | 6,34 | Rp8.573 |
| 27/06/2023 | 203.60 | 1.74 | 285.30 | 12,27 | Rp16.592 |
| 28/06/2023 | 204.30 | 1.57 | 245.60 | 18,56 | Rp25.099 |
| 29/06/2023 | 204.10 | 1.68 | 283.70 | 24,41 | Rp33.008 |
| 30/06/2023 | 187.90 | 5.94 | 1079.90 | 31,80 | Rp42.990 |
| 01/07/2023 | 200.70 | 2.51 | 428.30 | 38,91 | Rp52.601 |
| 02/07/2023 | 211.80 | 1.71 | 295.50 | 46,23 | Rp62.507 |
| 03/07/2023 | 188.50 | 01.01 | 158.90 | 53,01 | Rp71.671 |
| 04/07/2023 | 201.20 | 1.79 | 315.80 | 58,78 | Rp79.467 |
| 05/07/2023 | 200.10 | 1.14 | 198.20 | 65,21 | Rp88.157 |
| 06/07/2023 | 205.40 | 2.38 | 417.90 | 72,24 | Rp97.663 |
| 07/07/2023 | 209.80 | 1.91 | 336.70 | 78,50 | Rp106.135 |
| 08/07/2023 | 197.30 | 1.91 | 299.30 | 84,96 | Rp114.861 |
| 09/07/2023 | 209.70 | 2.59 | 440.10 | 91,28 | Rp123.405 |
| 10/07/2023 | 202.40 | 2.26 | 387.30 | 97,97 | Rp132.460 |
| 11/07/2023 | 214.30 | 02.05 | 336.80 | 105,56 | Rp142.710 |
| 12/07/2023 | 204.50 | 2.16 | 357.40 | 110,96 | Rp150.019 |
| 13/07/2023 | 204.00 | 1.70 | 287.20 | 117,98 | Rp159.500 |
| 14/07/2023 | 205.80 | 2.10 | 365.30 | 125,60 | Rp169.804 |
| 15/07/2023 | 201.60 | 0.99 | 166.30 | 131,40 | Rp177.647 |
| 16/07/2023 | 202.70 | 2.32 | 365.10 | 137,56 | Rp185.982 |
| 17/07/2023 | 206.90 | 4.17 | 847.00 | 143,03 | Rp193.375 |
| 18/07/2023 | 203.10 | 1.25 | 221.50 | 149,18 | Rp201.693 |
| 19/07/2023 | 214.20 | 2.65 | 463.30 | 156,42 | Rp211.477 |
| 20/07/2023 | 202.10 | 1.73 | 276.40 | 162,70 | Rp219.976 |
| 21/07/2023 | 201.80 | 2.30 | 378.30 | 168,77 | Rp228.177 |
| 22/07/2023 | 198.40 | 1.00 | 163.00 | 175,16 | Rp236.811 |
| 23/07/2023 | 204.30 | 0.78 | 141.00 | 182,84 | Rp247.197 |
| 24/07/2023 | 205.20 | 0.86 | 176.30 | 189,18 | Rp255.769 |
| 25/07/2023 | 204.10 | 4.42 | 884.00 | 195,37 | Rp264.142 |

# LAMPIRAN C
# LISTING PROGRAM

a. Listing program pada *AppScript Google Sheets*

```
function doGet (e) {
  Logger.log ( JSON.stringify (e) );  // view parameters
  var result = 'Ok'; // assume success
  if  (e.parameter == 'undefined') {
    result = 'No Parameters';
  }
  else {
    var sheet_id = 'kodeunikpadalinkwebsite';    // Spreadsheet ID
      var  sheet  =  SpreadsheetApp.openById  (sheet_id).getActiveSheet
();   // get Active sheet
    var newRow = sheet.getLastRow () + 1;
    var rowData = [];
    d=new Date ();
    rowData[0] = d; // Timestamp in column A
    rowData[1] = d.toLocaleTimeString (); // Timestamp in column A

    for  (var param in e.parameter) {
      Logger.log ('In for loop, param=' + param);
      var value = stripQuotes (e.parameter[param]);
      Logger.log (param + ':' + e.parameter[param]);
      switch  (param) {
        case 'value1': //Parameter 1, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[2] = value; //Value in column A
          result = 'Written on column A';
          break;
        case 'value2': //Parameter 2, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[3] = value; //Value in column B
          result += ' Written on column B';
          break;
```

```
        case 'value3': //Parameter 3, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[4] = value; //Value in column C
          result += ' Written on column C';
          break;
        case 'value4': //Parameter 1, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[5] = value; //Value in column A
          result = 'Written on column D';
          break;
        case 'value5': //Parameter 2, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[6] = value; //Value in column B
          result += ' Written on column E';
          break;
        case 'value6': //Parameter 3, It has to be updated in Column in
Sheets in the code, orderwise
          rowData[7] = value; //Value in column C
          result += ' Written on column F';
          break;
        default:
          result = "unsupported parameter";
      }
    }
    Logger.log (JSON.stringify (rowData));
    // Write new row below
    var newRange = sheet.getRange (newRow, 1, 1, rowData.length);
    newRange.setValues ([rowData]);
  }
  // Return result of operation
  return ContentService.createTextOutput (result);
}
function stripQuotes ( value ) {
  return value.replace (/^["']|['"]$/g, "");
}
```

b. Listing program pada Arduino

```
//Library
#include <TRIGGER_WIFI.h> // Mengimpor library TRIGGER_WIFI yang diperlukan
#include <TRIGGER_GOOGLESHEETS.h> // Mengimpor library TRIGGER_GOOGLESHEETS yang diperlukan

#include <ESP8266WiFi.h> // Mengimpor library ESP8266WiFi untuk koneksi WiFi

#include <PZEM004Tv30.h> // Mengimpor library PZEM004Tv30 untuk sensor arus dan tegangan
#include <LiquidCrystal_I2C.h> // Mengimpor library LiquidCrystal_I2C untuk tampilan LCD
#include <Wire.h>  // Mengimpor library Wire untuk komunikasi I2C

#include <WiFiClientSecure.h>
#include <Firebase.h>

#include <TimeLib.h>  // Mengimpor library TimeLib untuk manajemen waktu
#include <WiFiUdp.h>  // Mengimpor library WiFiUdp untuk koneksi WiFi UDP
#include <NTPClient.h> // Mengimpor library NTPClient untuk klien NTP

#include <UniversalTelegramBot.h> // Mengimpor library UniversalTelegramBot untuk berinteraksi dengan bot Telegram

#define RX 12
#define TX 13
#define rl 14

#define FIREBASE_HOST "internetorwifi-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "RsPOnJqa2AEAFpR6ciV74oGfg4EyMwBnC4GCAbJZ"
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
PZEM004Tv30 pzem(12, 13); // (RX,TX) connect to TX,RX of PZEM
float voltage, current, power, pf, energy, cost;
float harga = 1352;

int currentDay = 0;
int currentMonth = 0;
bool isEndOfMonth = false; // Flag untuk pengecekan akhir bulan
bool isStartOfMonth = false; // Flag untuk pengecekan awal bulan

const char *ssid1 = "namawifi1";
const char *password1 = "passwordwifi1";

const char *ssid2 = " namawifi ";
const char *password2 = " passwordwifi2";

const char* ntpServer = "pool.ntp.org";

char  column_name_in_sheets[][20]  =  {"value1",  "value2",  "value3",
"value4", "value5", "value6"};
const char* Sheets_GAS_ID = "kodeGASIDnya";
int No_of_Parameters = 6;

const char* BOTtoken = "Bot Token Telegram";
const int CHAT_ID = Nomor _ID_Akun_Telegram;
const String SENDER_ID = " Nomor ID Akun Telegram ";

UniversalTelegramBot bot(BOTtoken, client);

unsigned long lastTimeBotRan = 0;
const unsigned long botRequestDelay = 1000;

bool sendData = false;

WiFiUDP ntpUDP;
```

```
NTPClient timeClient(ntpUDP, ntpServer);

void Google_Sheets_ClearData() {
  // Implementasi untuk menghapus data pada spreadsheet
  // Kirim pesan ke bot Telegram
  String message = "Data pada spreadsheet telah direset.";
  bot.sendMessage("Nomor ID Akun Telegram", message);
  Serial.println("Reset Terkirim");
}

void handleNewMessages(int numNewMessages, String chat_id, String
message_sender_id) {
  for (int n = 0; n < numNewMessages; n++) {
    //String chat_id = String(bot.messages[n].chat_id);
    String text = bot.messages[n].text;
    String from_name = bot.messages[n].from_name;

    if (chat_id == String(CHAT_ID))
    {
      if (from_name == "")from_name = "Guest";
    }
    else
    {
      String errorMessage = "Anda tidak diizinkan mengakses bot ini.";
      bot.sendMessage(String(chat_id).c_str(), errorMessage);
    }

    if (message_sender_id == String(SENDER_ID)){
      //Cek Pembacaan PZEM
    if (text == "/tegangan")
    {
      int v = pzem.voltage();
      if (v < 0.0) v = 0.0;

       String tegangan = "Tegangan : ";
       tegangan += int(v);
```

```
   tegangan += "V";

  bot.sendMessage(chat_id,tegangan, "");
}


if (text == "/arus")
{
  float i = pzem.current();
  if (i < 0.000) i = 0.000;

   String arus = "Arus : ";
   arus += float(i);
   arus += "A";

  bot.sendMessage(chat_id,arus, "");
}

//Cek Pembacaan Sensor DHT11
if (text == "/daya")
{
  int p = pzem.power();
  if (p < 0.0) p = 0.0;
   String daya = "Daya : ";
   daya += int(p);
   daya += "W";

  bot.sendMessage(chat_id,daya, "");
}


if (text == "/kwh")
{
  float e = pzem.energy();
   String kwh = "kWh : ";
   kwh += float(e);
```

```
  bot.sendMessage(chat_id,kwh, "");
}

if (text == "/biaya")
{
  float cost = (pzem.energy()*1352);
   String biaya = "Rp. ";
   biaya += float(cost);
   biaya += "";


  bot.sendMessage(chat_id,biaya, "");
}

//Kontrol relay
if (text == "/reset") {
  digitalWrite(rl, HIGH);
  pzem.resetEnergy();
  bot.sendMessage(chat_id, "Reset", "");
}
if (text == "/resetdata") {
  digitalWrite(rl, HIGH);
  Google_Sheets_ClearData();
  bot.sendMessage(chat_id, "Reset OK", "");
}

if (text == "/relayoff") {
  digitalWrite(rl, HIGH);
  bot.sendMessage(chat_id, "Relay OFF", "");
}
 if (text == "/relayon") {
  digitalWrite(rl, LOW);
  bot.sendMessage(chat_id, "Relay ON", "");
}
```

```
  //Cek Command untuk kondisi semua
  if (text == "/kondisi") {
    int v = pzem.voltage();
    float i = pzem.current();
    float p = pzem.power();
    float e = pzem.energy();
    int cost = (e*1352);
    String kondisi;
    kondisi+= (String)"Pemantauan Monitoring Listrik\n\n" +
              (String)"Tegangan : " + v + (String)"V\n\n"+
              (String)"Arus : " + i + (String)"A\n\n"+
              (String)"Daya : " + p + (String)"W\n\n"+
              (String)"KWH : " + e + "\n\n"+
              (String)"Biaya : Rp. " + cost + (String)"\n\n"+
              "KLIK DISINI: link Google Sheetsnya ";
    bot.sendMessage(chat_id, kondisi);
  }

  //Cek Command untuk setiap aksi
  if (text == "/start") {
    String welcome = "Welcome  " + from_name + ".\n";
    welcome += "/tegangan : Status tegangan\n";
    welcome += "/arus : Status arus\n";
    welcome += "/daya : Status daya\n";
    welcome += "/kwh : Status energi\n";
    welcome += "/biaya : Status biaya\n";
    welcome += "/kondisi : Data saat ini\n";
    welcome += "/relayon : Nyalakan relay\n";
    welcome += "/relayoff : Matikan relay\n";
    welcome += "/reset : Reset data\n";
    bot.sendMessage(chat_id, welcome, "Markdown");
  }

 }
}
```

```
}

void setup() {
  pinMode(rl, OUTPUT);
  digitalWrite(rl, LOW);
  Serial.begin(115200);
  lcd.begin();
  lcd.backlight();

  while (WiFi.status() == WL_DISCONNECTED) {
    myNetwork();
  }

  Serial.println("Successfully Connected ");

  Google_Sheets_Init(column_name_in_sheets,            Sheets_GAS_ID,
No_of_Parameters);

  timeClient.begin();
  timeClient.setTimeOffset(7 * 3600);

  currentDay = day();
  currentMonth = month();
}


  void myNetwork() {
  int flag = 1;
  Serial.println("Searching Wifi......");
  lcd.print("Connecting...");
  int network = WiFi.scanNetworks();          //5
  for (int i = 0 ; i < network; i++) {
    switch (flag) {
      case 1:
        flag =2;
        if (WiFi.SSID(i) == ssid1) {
```

```
          WiFi.begin(ssid1, password1);
          Serial.println("/n Wifi Found");
          delay(2000);
          Serial.println("Connecting with Wifi Please Wait ");
          delay(8000);
          Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
          break;
        }
        case 2:
           flag =1;
           if (WiFi.SSID(i) == ssid2) {
          WiFi.begin(ssid2, password2);
          Serial.println("/n Wifi Found");
          delay(2000);
          Serial.println("Connecting with Hostpot Please Wait ");
          delay(8000);
          Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
          break;
        }
    }
  }

  client.setInsecure();
  Serial.println("");
  Serial.println("Successfully Connected");
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("WiFi Connected");
  delay(1000);
  lcd.clear();
  }

void loop() {
  timeClient.update();
  setTime(timeClient.getEpochTime());
```

```
voltage = pzem.voltage();
current = pzem.current();
power = pzem.power();
pf = pzem.pf();
energy = pzem.energy();
if (isnan(energy)) energy = 0.0;
int harga = 1352;
float cost = energy * harga;
if (isnan(cost)) {
  cost = 0.0;
}

lcd.clear();

Serial.print(voltage); Serial.print("V; ");
lcd.setCursor(0, 0);
lcd.print("V= ");
lcd.setCursor(2, 0);
lcd.print(voltage);

Serial.print(current); Serial.print("A; ");
lcd.setCursor(11, 0);
lcd.print("I= ");
lcd.setCursor(13, 0);
lcd.print(current);

Serial.print(power); Serial.print("W; ");
lcd.setCursor(11, 1);
lcd.print("P= ");
lcd.setCursor(13, 1);
lcd.print(power);

Serial.print(pf); Serial.print("");

Serial.print(energy); Serial.print("kWh;  ");
lcd.setCursor(0, 2);
```

```
  lcd.print("E= ");
  lcd.setCursor(2, 2);
  lcd.print(energy);

  Serial.print("Rp; "); Serial.println(cost);
  int roundedCost = round(cost);
  lcd.setCursor(0, 3);
  lcd.print("Rp.");
  lcd.setCursor(3, 3);
  lcd.print(roundedCost);

  int currentDay = day();
  int currentMonth = month();
  int currentYear = year();

  // Dapatkan waktu saat ini
  int currentHour = hour();
  int currentMinute = minute();
  int currentSecond = second();

  Data_to_Sheets(No_of_Parameters,   voltage,   current,  power,  pf,
energy, cost);

  Serial.println();
  delay(30000);


      // Pengecekan waktu untuk pengiriman otomatis
  if (currentHour == 17 && currentMinute == 50 && !sendData) {
    // Kirim data ke Telegram
    String sekarang = "Data hari ini\n";
    sekarang += "Tegangan: " + String(voltage) + "V\n\n";
    sekarang += "Arus: " + String(current) + "A\n\n";
    sekarang += "Daya: " + String(power) + "W\n\n";
    sekarang += "Faktor Daya: " + String(pf) + "\n\n";
    sekarang += "Energi: " + String(energy) + "kWh\n\n";
```

```
    sekarang += "Biaya: Rp " + String(cost)+ "\n\n";
    sekarang += "KLIK DISINI: link Google Sheetsnya ";

    bot.sendMessage(String(CHAT_ID), sekarang);
    sendData = true; // Setel flag untuk mencegah pengiriman berulang
kali pada hari yang sama
  Serial.println("Sended");
  }
  delay(1000);

  // Pengecekan ulang flag setiap hari baru
  if (currentHour == 18 && currentMinute == 1) {
    sendData = false; // Setel flag kembali ke false pada hari yang
baru
  }
  Serial.println("OK");
  delay(1000);



  // Pengecekan akhir bulan
  if (currentDay == 25 && currentHour == 17 && currentMinute == 55 &&
!isEndOfMonth) {
    isEndOfMonth = true;
    String message1 = "Akhir catat:\n\n";
    message1 += "Energi: " + String(energy) + "\n\n";
    message1 += "Biaya: Rp " + String(cost);
    bot.sendMessage(String(CHAT_ID), message1);
        // Reset energi PZEM004Tv30 di akhir bulan
    pzem.resetEnergy();
    Serial.println("Direset");
  }

  // Pengecekan awal bulan
  if (currentDay == 25 && currentHour == 18 && currentMinute == 0 &&
!isStartOfMonth) {
```

```
    isStartOfMonth = true;
    String message2 = "Awal catat:\n\n";
    message2 += "Energi: " + String(energy) + "kWh\n\n";
    message2 += "Biaya: Rp " + String(cost);
    bot.sendMessage(String(CHAT_ID), message2);
    Serial.println();

    }


  if (millis() > lastTimeBotRan + botRequestDelay)
  {
    int numNewMessages = bot.getUpdates(bot.last_message_received +
1);

    while (numNewMessages) {
    String chat_id = String(bot.messages[0].chat_id);
    String message_sender_id = String(bot.messages[0].from_id);
    Serial.println("Got response");
    handleNewMessages(numNewMessages, chat_id, message_sender_id);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  }

    lastTimeBotRan = millis();
    delay(1000);
  }
}
```