SKRIPSI

RANCANG BANGUN SISTEM PROTEKSI KABEL MOTOR 3 FASE MENGGUNAKAN MONITORING DAYA DAN SUHU BERBASIS IOT

DARWIN SAPUTRA

71240912011

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS ISLAM SUMATERA UTARA
MEDAN
2025

RANCANG BANGUN SISTEM PROTEKSI KABEL MOTOR 3 FASE MENGGUNAKAN MONITORING DAYA DAN SUHU BERBASIS IOT

SKRIPSI

Diajukan Sebagai Syarat Untuk Memperoleh Gelar Sarjana Teknik
Program Studi Teknik Elektro
Universitas Islam Sumatera Utara

Oleh:

DARWIN SAPUTRA

71240912011

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS ISLAM SUMATERA UTARA
MEDAN

2025

RANCANG BANGUN SYSTEM PROTEKSI KABEL MOTOR 3 PHASE MENGGUNAKAN MONITORING DAYA DAN SUHU BERBASIS IOT

SKRIPSI

Untuk Memperoleh Gelar Sarjana Teknik (ST)
Pada Program Studi Teknik Elektro, Fakultas Teknik
Universitas Islam Sumatera Utara

Oleh :

DARWIN SAPUTRA

NPM : 71240912011

Disetujui oleh :

Pembimbing I :	Pembimbing II :
Ir. Raja Harahap, MT	**Ir. Armansyah, MT**
NIDN : 013016503	**NIDN : 0004096302**

Ketua Program Studi :

Ir. H. Raja Harahap. MT

NIDN : 0013016503

KATA PENGHANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Puji Syukur kehadirat Allah SWT atas Rahmat-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul RANCANG BANGUN SYSTEM PROTEKSI KABEL MOTOR 3 FASA MENGGNAKAN MONITORING DAYA DAN SUHU BERBASIS IoT" dengan baik. Skripsi ini dibuat untuk memenuhi persyaratan untuk mencapai gelar Sarjana Teknik (S1) program studi Teknik Elektro sesuai dengan kurikulum Fakultas Teknik UISU untuk menambah wawasan tentang elektro dan pemanfaatannya di bidang teknologi industri. Sholawat serta salam senantiasa tercurahkan atas Nabi Muhammad SAW yang telah membawa umat manusia dari jalan kegelapan menuju jalan yang terang benderang. Semoga kita semua mendapat syafaatnya di Yaumil Akhir kelak. Selama melaksanakan perkuliahan dan dalam menyelesaikan skripsi ini penulis telah menerima banyak bimbingan, pengarahan, petunjuk, saran, doa dan dukungan dari berbagai pihak. Untuk itu menyampaikan ucapan terima kasih yang sebesar-besarnya kepada yang terhormat: Ibu Prof. Dr. Safrida, S.E., M.Si. Selaku Rektor Universitas Islam Sumatera Utara , Ibu Ir. Hj. Darlina Tanjung, M.T. selaku Dekan Fakultas Teknik Universitas Islam Sumatera Utara, Bapak Ir. H. Raja Harahap, M.T. selaku Ketua Program Studi Teknik Elektro yang mendukung dengan penuh segala kegiatan kami sebagai mahasiswa dan juga selaku pembimbing I yang memberikan contoh dan nasehat yang baik kepada mahasiswa terlebih koreksi yang baik dalam kemajuan penulisan skripsi ini, Bapak Ir. H. Armansyah, M.T selaku dosen pembimbing I1 Yang mendukung penuh perjalanan Pendidikan sarjana berupa pembelajaran hingga koreksi yang baik dalam penulisan skripsi ini. Sosok yang tegas dan keperdulian yang besar dari beliau menjadi salah satu jalan mempermudah penyelesaian skripsi ini, Bapak Ir. Sudaryanto selaku salah satu staff biro Teknik Elektro yang memiliki sikap dan perhatian yang baik dalam melayani kebutuhan mahasiswa, Dosen pengajar lainnya yang sudah berpartisipasi dalam memberikan materi pengetahuan yang banyak sehingga menjadi salah satu modal dalam penyusunan skripsi ini, Seluruh

staff biro Teknik UISU, yang bekerjasama dalam kemajuan Teknik elektro, Rekan-rekan seperjuangan diTeknik Elektro yang selalu mendukung satu sama lain, dan yang lainnya yang tidak bisa disebutkan satu persatu, Dan Keluaraga ku Kedua orang tuaku,Istri Dan Anak ku atas dukungan penuh baik teori maupun materi dan nasehat untuk senantiasa bisa bersabar dalam ujian, berjuang untuk masa depan yang lebih baik, dan mengajarkan bagaimana tutur sapa yang baik kepada semua orang, Seluruh pihak yang tidak bisa disebutkan satu per satu.

Semoga kalian selalu dalam lindungan Allah SWT. Jika ada kata yang lebih bermakna dari kata "terima kasih", maka kata tersebut yang akan menggambarkan betapa bersyukurnya penulis akan dukungan dari semua pihak. Akhirnya penulis berharap semoga skripsi ini bermanfaat bagi saya pribadi dan semua pihak yang membacanya, Penulis juga menyadari skripsi ini masih jauh dari kata sempurna. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan skripsi ini. Akhir kata, semoga skripsi ini dapat memenuhi persyaratan pencapaian gelar Sarjana Teknik di Universitas Islam Sumatera Utara. Semoga amal kebaikan semua pihak mendapat ganjaran yang berlipat dari Allah SWT dan kelak kita dikumpulkan kedalam golongan yang berkasih saying karena Allah SWT

Medan, Oktober 2025

Penulis,

Darwin Saputra

71240912011

\

DAFTAR ISI

Halaman

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR PUSTAKA

Akbar SA, dkk. *Online Monitoring Kualitas Air Waduk Berbasis Thingspeak.* **TRANSMISI.** 2019;21(4):1–7.

Anonym. *Datasheet DS18B20.* 2018. Tersedia pada: http://www.alldatasheet.com/datasheet-pdf/pdf/58557/DALLAS/DS18B20.html

Budijono S, Felita. *Smart Temperature Monitoring System Using ESP32 and DS18B20.* **IOP Conference Series: Earth and Environmental Science.** 2021;794(1).

Burange A, Misalkar H. *Review of Internet of Things in Development of Smart Cities with Data Management and Privacy.* **International Conference on Advances in Computer Engineering and Applications.** 2015:189–195.

Desiana Wahyuningsih. *Monitoring dan Evaluasi Untuk Tercapainya Tujuan Kinerja.* **Jurnal Ilmiah Pendidikan Teknik Elektro.** 2021;5(1):87.

Digiware Store. *DS18B20 Waterproof Temperature Probe Sensor Suhu Air*. [dikutip 10 Okt 2025]. Tersedia pada: https://digiwarestore.com/en/temperature-sensor/ds18b20-waterproof-temperature-probe-sensor-suhu-air-296473.html

Jefri Stender. *Rancang Bangun Sistem Monitoring Syslog Terpusat Menggunakan Kibana.* 2018.

Laghari AA, Wu K, Laghari RA, Ali M, Khan AA. *A review and state of art of Internet of Things (IoT).* **Archives of Computational Methods in Engineering.** 2022;29(3):1395–1413.

Muliadi, Imran A, Rasul M. *Pengembangan Tempat Sampah Pintar Menggunakan ESP32.* **Jurnal Media Elektrik.** 2020;17(2):73–79.

Nicholas LLG, Logenthiran T, Abidi K. *Application of Internet of Things (IoT) for Home Energy Management.* **IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC).** 2017.

NN Digital. *Mengenal PZEM-004T Modul Elektronik untuk Alat Pengukuran Listrik.* [dikutip 10 Okt 2025]. Tersedia pada: https://www.nn-digital.com/blog/2019/07/10/mengenal-pzem-004t-modul-elektronik-untuk-alat-pengukuran-listrik/.

Rakhman.net. *Tegangan 3 Fasa.* [dikutip 15 Okt 2025]. Tersedia pada: https://rakhman.net/electrical-id/tegangan-3-fasa/.

Reichelt Elektronik. *SBC-NODEMCU-ESP32 Datasheet V1.2*. [dikutip 15 Okt 2025]. Tersedia pada: https://cdn-reichelt.de/documents/datenblatt/A300/SBC-NODEMCU-ESP32-DATASHEET_V1.2.pdf

Win H. *Implementation of WiFi-Based Single Fasa Meter for Internet of Things (IoT)*. **IEEE International Electrical Engineering Congress.** Pattaya, Thailand; 2017.

LAMPIRAN

Program lengkap

```
#include <PZEM004Tv30.h> // Sertakan library untuk modul pemantau energi PZEM004T v3.0.

// Definisikan pin default untuk komunikasi PZEM jika belum didefinisikan.
// PZEM_RX_PIN: Pin penerima untuk modul PZEM.
// PZEM_TX_PIN: Pin pengirim untuk modul PZEM.
#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
#define PZEM_RX_PIN 16
#define PZEM_TX_PIN 17
#endif

// Definisikan port serial untuk komunikasi PZEM jika belum didefinisikan.
// Serial2 umumnya digunakan pada ESP32 untuk port serial hardware tambahan.
#if !defined(PZEM_SERIAL)
#define PZEM_SERIAL Serial2
#endif

#define NUM_PZEMS 3 // Definisikan jumlah modul PZEM yang digunakan (untuk 3 fasa: R, S, T).

// Deklarasikan array objek PZEM004Tv30, satu untuk setiap modul PZEM.
PZEM004Tv30 pzems[NUM_PZEMS];

// Array untuk menyimpan nama fasa agar lebih mudah direferensikan dalam output (misalnya, "R", "S", "T").
const char* PHASE_NAMES[] = {"R", "S", "T"};

// Array untuk menyimpan parameter kelistrikan yang dibaca dari setiap modul PZEM.
float voltage[NUM_PZEMS];    // Tegangan untuk setiap fasa
float current[NUM_PZEMS];    // Arus untuk setiap fasa
float power[NUM_PZEMS];      // Daya aktif untuk setiap fasa
float energy[NUM_PZEMS];     // Total energi yang dikonsumsi untuk setiap fasa
float frequency[NUM_PZEMS];  // Frekuensi untuk setiap fasa
float pf[NUM_PZEMS];         // Faktor daya untuk setiap fasa

// Sertakan library untuk tampilan LCD I2C.
```

```
#include <Wire.h>             // Diperlukan untuk komunikasi I2C
#include <LiquidCrystal_I2C.h> // Library untuk tampilan LCD I2C

// Inisialisasi objek LCD dengan alamat I2C-nya (0x27), dan dimensi (16 kolom, 2 baris).
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Sertakan library untuk sensor suhu DS18B20.
#include <OneWire.h>         // Diperlukan untuk komunikasi bus OneWire
#include <DallasTemperature.h> // Library untuk sensor suhu Dallas DS18B20

// Definisikan pin digital yang terhubung ke bus data OneWire (sensor DS18B20).
#define ONE_WIRE_BUS 25

// Siapkan instance OneWire untuk berkomunikasi dengan perangkat OneWire apa pun (sensor DS18B20).
OneWire oneWire(ONE_WIRE_BUS);

// Teruskan referensi oneWire kita ke library Dallas Temperature.
DallasTemperature sensors(&oneWire);

// Deklarasikan variabel untuk menyimpan alamat unik setiap sensor DS18B20.
DeviceAddress sensor1, sensor2, sensor3;
// Variabel untuk menyimpan suhu yang dibaca dari setiap sensor dalam Celsius.
float tempC1, tempC2, tempC3;

// Sertakan dan definisi terkait ThingSpeak.
#include <WiFi.h>        // Library untuk konektivitas WiFi pada ESP32
#include "ThingSpeak.h"  // Library untuk berinteraksi dengan platform IoT ThingSpeak
#include "secrets.h"     // File header kustom yang diharapkan berisi SSID dan kata sandi WiFi

// Kredensial WiFi (didefinisikan dalam secrets.h).
char ssid[] = SECRET_SSID;   // SSID jaringan Anda (nama)
char pass[] = SECRET_PASS;   // Kata sandi jaringan Anda

// ID Saluran ThingSpeak dan Kunci API untuk Parameter Kelistrikan Fasa R + Suhu 1.
// **PENTING**: Ganti dengan ID Saluran ThingSpeak dan Kunci API Tulis Anda yang sebenarnya.
unsigned long THINGSPEAK_CHANNEL_ID_R = 3010324;
```

```
const char* THINGSPEAK_API_KEY_R = "JN5E21EHM9SCGO5M";

// ID Saluran ThingSpeak dan Kunci API untuk Parameter Kelistrikan Fasa S + Suhu 2.
// **PENTING**: Ganti dengan ID Saluran ThingSpeak dan Kunci API Tulis Anda yang sebenarnya.
unsigned long THINGSPEAK_CHANNEL_ID_S = 3010326;
const char* THINGSPEAK_API_KEY_S = "KA3BILHQC1X03MIB";

// ID Saluran ThingSpeak dan Kunci API untuk Parameter Kelistrikan Fasa T + Suhu 3.
// **PENTING**: Ganti dengan ID Saluran ThingSpeak dan Kunci API Tulis Anda yang sebenarnya.
unsigned long THINGSPEAK_CHANNEL_ID_T = 3010327;
const char* THINGSPEAK_API_KEY_T = "COOY8YT7WTWNWNQC";

WiFiClient client; // Deklarasikan objek WiFiClient untuk komunikasi ThingSpeak.

unsigned long previousMillis = 0; // Variabel untuk menyimpan waktu terakhir LCD diperbarui.
const long interval = 2000;       // Interval untuk refresh tampilan LCD (2 detik).

unsigned long lastThingSpeakUpdate = 0; // Variabel untuk menyimpan waktu terakhir ThingSpeak diperbarui.
const long thingSpeakUpdateInterval = 20000; // Interval untuk pembaruan ThingSpeak (20 detik).
                                             // ThingSpeak memiliki interval pembaruan minimum 15 detik per saluran.

int displayIndex = 0; // Indeks untuk mengontrol data mana yang saat ini ditampilkan di LCD.
float temp_rev = 60;  // Ambang batas suhu untuk alarm buzzer (60 derajat Celsius).
int buzzer = 19;      // Pin digital yang terhubung ke buzzer.

void setup() {
    // Inisialisasi LCD.
    lcd.begin();
    lcd.backlight(); // Nyalakan lampu latar LCD.
    lcd.setCursor(0,0); // Atur kursor ke kolom pertama, baris pertama.
    lcd.print("Connecting to"); // Tampilkan pesan koneksi.
    lcd.setCursor(0,1); // Atur kursor ke kolom pertama, baris kedua.
    lcd.print(ssid); // Tampilkan SSID WiFi.

    Serial.begin(115200); // Inisialisasi komunikasi serial untuk debugging pada baud rate 115200.
    pinMode(buzzer, OUTPUT); // Atur pin buzzer sebagai output.
```

```
// Pengaturan koneksi WiFi.
Serial.print("Connecting to WiFi...");
while (WiFi.status() != WL_CONNECTED) { // Ulangi sampai WiFi terhubung.
    WiFi.begin(ssid, pass); // Coba sambungkan ke WiFi.
    delay(5000); // Tunggu 5 detik sebelum mencoba lagi.
    Serial.println("."); // Cetak titik ke serial untuk menunjukkan menunggu.
}
Serial.println("\nWiFi connected"); // Cetak pesan sukses.
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); // Cetak alamat IP yang ditetapkan.

ThingSpeak.begin(client); // Inisialisasi library ThingSpeak dengan klien WiFi.

// Inisialisasi setiap modul PZEM.
// Alamat untuk modul PZEM diatur ke 0x10, 0x11, 0x12 untuk tiga modul.
for(int i = 0; i < NUM_PZEMS; i++) {
    pzems[i] = PZEM004Tv30(PZEM_SERIAL, PZEM_RX_PIN, PZEM_TX_PIN, 0x10 + i);
}

// Mulai library Dallas Temperature untuk sensor DS18B20.
sensors.begin();

// Temukan perangkat DS18B20 di bus OneWire.
Serial.println("Locating DS18B20 devices...");
int deviceCount = sensors.getDeviceCount(); // Dapatkan jumlah sensor yang terhubung.
Serial.print("Found ");
Serial.print(deviceCount, DEC); // Cetak hitungan dalam desimal.
Serial.println(" DS18B20 devices.");

// Peringatan jika ditemukan kurang dari 3 sensor.
if (deviceCount < 3) {
    Serial.println("Warning: Less than 3 DS18B20 sensors found. Please check wiring.");
}

// Dapatkan dan cetak alamat sensor yang terhubung.
```

```
    if (deviceCount >= 1) {
        sensors.getAddress(sensor1, 0); // Dapatkan alamat sensor pertama (indeks 0).
        Serial.print("Sensor 1 Address: ");
        printAddress(sensor1); // Panggil fungsi pembantu untuk mencetak alamat.
    }
    if (deviceCount >= 2) {
        sensors.getAddress(sensor2, 1); // Dapatkan alamat sensor kedua (indeks 1).
        Serial.print("Sensor 2 Address: ");
        printAddress(sensor2);
    }
    if (deviceCount >= 3) {
        sensors.getAddress(sensor3, 2); // Dapatkan alamat sensor ketiga (indeks 2).
        Serial.print("Sensor 3 Address: ");
        printAddress(sensor3);
    }

    Serial.println(""); // Cetak baris kosong untuk pemformatan.
    delay(2000); // Penundaan singkat sebelum membersihkan LCD.
    lcd.clear(); // Bersihkan layar LCD.
}

void loop() {
    // --- Periksa koneksi WiFi ---
    // Ini memastikan perangkat mencoba menyambung kembali jika WiFi terputus.
    if (WiFi.status() != WL_CONNECTED) {
        Serial.println("Reconnecting to WiFi...");
        while (WiFi.status() != WL_CONNECTED) {
            WiFi.begin(ssid, pass);
            delay(5000);
            Serial.println(".");
        }
        Serial.println("\nWiFi reconnected");
    }

    // --- Baca Data PZEM ---
    // Loop melalui setiap modul PZEM untuk membaca parameter kelistrikannya.
```

```
for(int i = 0; i < NUM_PZEMS; i++) {
    voltage[i] = pzems[i].voltage();     // Baca tegangan
    current[i] = pzems[i].current();     // Baca arus
    power[i] = pzems[i].power();         // Baca daya
    energy[i] = pzems[i].energy();       // Baca energi
    frequency[i] = pzems[i].frequency(); // Baca frekuensi
    pf[i] = pzems[i].pf();               // Baca faktor daya

    // Periksa apakah ada nilai yang dibaca adalah NaN (Not a Number), yang menunjukkan kesalahan pembacaan.
    // Jika terjadi kesalahan, atur nilai ke 0.0 untuk mencegah masalah dengan ThingSpeak.
    if(isnan(voltage[i]) || isnan(current[i]) || isnan(power[i]) ||
       isnan(energy[i]) || isnan(frequency[i]) || isnan(pf[i])) {
        Serial.print("Error reading from PZEM module for Phase "); Serial.println(PHASE_NAMES[i]);
        voltage[i] = 0.0;
        current[i] = 0.0;
        power[i] = 0.0;
        energy[i] = 0.0;
        frequency[i] = 0.0;
        pf[i] = 0.0;
    }
}

// --- Baca Suhu DS18B20 ---
Serial.print("Requesting temperatures...");
sensors.requestTemperatures(); // Kirim perintah ke semua sensor untuk melakukan konversi suhu.
Serial.println("DONE");

// Baca suhu dari setiap sensor jika terdeteksi.
if (sensors.getDeviceCount() >= 1) {
    tempC1 = sensors.getTempC(sensor1); // Dapatkan suhu dari sensor 1.
    if(isnan(tempC1)) tempC1 = 0.0; // Tangani NaN untuk suhu.
} else { tempC1 = 0.0; } // Jika sensor tidak ditemukan, atur ke 0.0.

if (sensors.getDeviceCount() >= 2) {
    tempC2 = sensors.getTempC(sensor2); // Dapatkan suhu dari sensor 2.
    if(isnan(tempC2)) tempC2 = 0.0;
```

```
    } else { tempC2 = 0.0; }

    if (sensors.getDeviceCount() >= 3) {
        tempC3 = sensors.getTempC(sensor3); // Dapatkan suhu dari sensor 3.
        if(isnan(tempC3)) tempC3 = 0.0;
    } else { tempC3 = 0.0; }

    // --- Output Monitor Serial ---
    // Cetak semua data yang dikumpulkan ke monitor serial untuk debugging dan verifikasi.
    Serial.println("\n--- Pembacaan PZEM ---");
    for(int i = 0; i < NUM_PZEMS; i++) {
        Serial.print("Fasa "); Serial.print(PHASE_NAMES[i]); Serial.print(": ");
        Serial.print("Tegangan: "); Serial.print(voltage[i]); Serial.print("V | ");
        Serial.print("Arus: "); Serial.print(current[i]); Serial.print("A | ");
        Serial.print("Daya: "); Serial.print(power[i]); Serial.print("W | ");
        Serial.print("Energi: "); Serial.print(energy[i],3); Serial.print("kWh | "); // Energi dengan 3 angka
desimal
        Serial.print("Frekuensi: "); Serial.print(frequency[i], 1); Serial.print("Hz | "); // Frekuensi dengan 1
angka desimal
        Serial.print("PF: "); Serial.println(pf[i]);
    }

    Serial.println("--- Suhu DS18B20 ---");
    Serial.print("Sensor Suhu 1 (C): "); Serial.println(tempC1);
    Serial.print("Sensor Suhu 2 (C): "); Serial.println(tempC2);
    Serial.print("Sensor Suhu 3 (C): "); Serial.println(tempC3);
    Serial.println("---------------------------------\n");

    // --- Pembaruan ThingSpeak ---
    unsigned long currentMillis = millis(); // Dapatkan waktu saat ini sejak ESP32 dimulai.

    // Periksa apakah waktu yang cukup telah berlalu untuk memperbarui ThingSpeak.
    if (currentMillis - lastThingSpeakUpdate >= thingSpeakUpdateInterval) {
        lastThingSpeakUpdate = currentMillis; // Atur ulang timer.

        Serial.println("Mengirim data ke ThingSpeak...");
```

```
// Saluran R: Parameter Kelistrikan (Fasa R) + Suhu 1
// Hanya coba menulis jika ID Saluran dan Kunci API telah diatur.
if (THINGSPEAK_CHANNEL_ID_R != 0 && strlen(THINGSPEAK_API_KEY_R) > 0) {
    ThingSpeak.setField(1, voltage[0]);   // Field 1: Tegangan Fasa R
    ThingSpeak.setField(2, current[0]);   // Field 2: Arus Fasa R
    ThingSpeak.setField(3, power[0]);     // Field 3: Daya Fasa R
    ThingSpeak.setField(4, energy[0]);    // Field 4: Energi Fasa R
    ThingSpeak.setField(5, frequency[0]); // Field 5: Frekuensi Fasa R
    ThingSpeak.setField(6, pf[0]);        // Field 6: Faktor Daya Fasa R
    ThingSpeak.setField(7, tempC1);       // Field 7: Suhu Sensor 1

    // Tulis field ke Saluran ThingSpeak R.
    int httpCodeR = ThingSpeak.writeFields(THINGSPEAK_CHANNEL_ID_R, THINGSPEAK_API_KEY_R);
    if (httpCodeR == 200) {
        Serial.println("Pembaruan Saluran R berhasil.");
    } else {
        Serial.print("Pembaruan Saluran R gagal. Kode kesalahan HTTP: ");
        Serial.println(httpCodeR);
    }
} else {
    Serial.println("ID Saluran R ThingSpeak atau Kunci API tidak diatur.");
}

delay(500); // Penundaan singkat antara pembaruan saluran untuk menghindari kelebihan beban API ThingSpeak.

// Saluran S: Parameter Kelistrikan (Fasa S) + Suhu 2
if (THINGSPEAK_CHANNEL_ID_S != 0 && strlen(THINGSPEAK_API_KEY_S) > 0) {
    ThingSpeak.setField(1, voltage[1]);   // Field 1: Tegangan Fasa S
    ThingSpeak.setField(2, current[1]);   // Field 2: Arus Fasa S
    ThingSpeak.setField(3, power[1]);     // Field 3: Daya Fasa S
    ThingSpeak.setField(4, energy[1]);    // Field 4: Energi Fasa S
    ThingSpeak.setField(5, frequency[1]); // Field 5: Frekuensi Fasa S
    ThingSpeak.setField(6, pf[1]);        // Field 6: Faktor Daya Fasa S
    ThingSpeak.setField(7, tempC2);       // Field 7: Suhu Sensor 2
```

```
        int httpCodeS = ThingSpeak.writeFields(THINGSPEAK_CHANNEL_ID_S, THINGSPEAK_API_KEY_S);
        if (httpCodeS == 200) {
            Serial.println("Pembaruan Saluran S berhasil.");
        } else {
            Serial.print("Pembaruan Saluran S gagal. Kode kesalahan HTTP: ");
            Serial.println(httpCodeS);
        }
    } else {
        Serial.println("ID Saluran S ThingSpeak atau Kunci API tidak diatur.");
    }

    delay(500); // Penundaan singkat antara pembaruan saluran.

    // Saluran T: Parameter Kelistrikan (Fasa T) + Suhu 3
    if (THINGSPEAK_CHANNEL_ID_T != 0 && strlen(THINGSPEAK_API_KEY_T) > 0) {
        ThingSpeak.setField(1, voltage[2]);   // Field 1: Tegangan Fasa T
        ThingSpeak.setField(2, current[2]);   // Field 2: Arus Fasa T
        ThingSpeak.setField(3, power[2]);     // Field 3: Daya Fasa T
        ThingSpeak.setField(4, energy[2]);    // Field 4: Energi Fasa T
        ThingSpeak.setField(5, frequency[2]); // Field 5: Frekuensi Fasa T
        ThingSpeak.setField(6, pf[2]);        // Field 6: Faktor Daya Fasa T
        ThingSpeak.setField(7, tempC3);       // Field 7: Suhu Sensor 3

        int httpCodeT = ThingSpeak.writeFields(THINGSPEAK_CHANNEL_ID_T, THINGSPEAK_API_KEY_T);
        if (httpCodeT == 200) {
            Serial.println("Pembaruan Saluran T berhasil.");
        } else {
            Serial.print("Pembaruan Saluran T gagal. Kode kesalahan HTTP: ");
            Serial.println(httpCodeT);
        }
    } else {
        Serial.println("ID Saluran T ThingSpeak atau Kunci API tidak diatur.");
    }
}
```

```
// --- Manajemen Tampilan LCD ---
currentMillis = millis(); // Baca ulang currentMillis untuk timer LCD.

// Periksa apakah waktu yang cukup telah berlalu untuk memperbarui tampilan LCD.
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // Atur ulang timer LCD.

    displayIndex++; // Tingkatkan indeks tampilan untuk menampilkan set parameter berikutnya.
    if (displayIndex > 7) { // Ada 8 status tampilan (0-7).
        displayIndex = 0; // Atur ulang ke parameter pertama jika semua status telah ditampilkan.
    }

    lcd.clear(); // Bersihkan LCD sebelum mencetak data baru.

    // Gunakan pernyataan switch-case untuk menampilkan set data yang berbeda berdasarkan displayIndex.
    switch(displayIndex) {
        case 0: // Tampilkan Tegangan R & S, Arus R & S
            lcd.setCursor(0, 0); lcd.print("VR:"); lcd.print(voltage[0], 1); lcd.print("V");
            lcd.setCursor(9, 0); lcd.print("IR:"); lcd.print(current[0], 2); lcd.print("A");
            lcd.setCursor(0, 1); lcd.print("VS:"); lcd.print(voltage[1], 1); lcd.print("V");
            lcd.setCursor(9, 1); lcd.print("IS:"); lcd.print(current[1], 2); lcd.print("A");
            break;

        case 1: // Tampilkan Tegangan T & Arus T, Daya R
            lcd.setCursor(0, 0); lcd.print("VT:"); lcd.print(voltage[2], 1); lcd.print("V");
            lcd.setCursor(9, 0); lcd.print("IT:"); lcd.print(current[2], 2); lcd.print("A");
            lcd.setCursor(0, 1); lcd.print("PR:"); lcd.print(power[0], 0); lcd.print("W");
            break;

        case 2: // Tampilkan Daya R & S
            lcd.setCursor(0, 0); lcd.print("PR:"); lcd.print(power[0], 0); lcd.print("W");
            lcd.setCursor(0, 1); lcd.print("PS:"); lcd.print(power[1], 0); lcd.print("W");
            break;

        case 3: // Tampilkan Daya S & T
            lcd.setCursor(0, 0); lcd.print("PS:"); lcd.print(power[1], 0); lcd.print("W");
```

```
                lcd.setCursor(0, 1); lcd.print("PT:"); lcd.print(power[2], 0); lcd.print("W");
                break;

            case 4: // Tampilkan Energi R & S
                lcd.setCursor(0, 0); lcd.print("En R:"); lcd.print(energy[0], 2); lcd.print("kWh");
                lcd.setCursor(0, 1); lcd.print("En S:"); lcd.print(energy[1], 2); lcd.print("kWh");
                break;

            case 5: // Tampilkan Energi S & T
                lcd.setCursor(0, 0); lcd.print("En S:"); lcd.print(energy[1], 2); lcd.print("kWh");
                lcd.setCursor(0, 1); lcd.print("En T:"); lcd.print(energy[2], 2); lcd.print("kWh");
                break;

            case 6: // Tampilkan Frekuensi & Faktor Daya (menggunakan Fasa R sebagai contoh, diasumsikan serupa
untuk semua)
                lcd.setCursor(0, 0); lcd.print("Freq:"); lcd.print(frequency[0], 1); lcd.print("Hz");
                lcd.setCursor(0, 1); lcd.print("PF:"); lcd.print(pf[0], 2);
                break;

            case 7: // Tampilkan Suhu dari ketiga sensor
                lcd.setCursor(0, 0); lcd.print("T1:");
                if(!isnan(tempC1)) lcd.print(tempC1, 1); else lcd.print("Err"); // Cetak "Err" jika suhu adalah NaN
                lcd.print("C");

                lcd.setCursor(8, 0); lcd.print("T2:");
                if(!isnan(tempC2)) lcd.print(tempC2, 1); else lcd.print("Err");
                lcd.print("C");

                lcd.setCursor(0, 1); lcd.print("T3:");
                if(!isnan(tempC3)) lcd.print(tempC3, 1); else lcd.print("Err");
                lcd.print("C");
                break;
        }
    }

    // Logika alarm buzzer: jika ada suhu yang melebihi temp_rev, aktifkan buzzer.
```

```
    if (tempC1 >= temp_rev || tempC2 >= temp_rev || tempC3 >= temp_rev){
        digitalWrite(buzzer,HIGH); // Nyalakan buzzer.
        delay(100);                // Biarkan menyala selama 100ms.
        digitalWrite(buzzer,LOW);  // Matikan buzzer.
        delay(100);                // Biarkan mati selama 100ms (menciptakan suara bip).
    }
    // Tidak ada penundaan eksplisit di akhir loop, karena interval pembaruan ThingSpeak dan interval LCD
    // sudah memperkenalkan penundaan, membuat loop cukup efisien.
}

// Fungsi pembantu untuk mencetak alamat perangkat DS18B20 ke monitor serial.
void printAddress(DeviceAddress deviceAddress) {
    for (uint8_t i = 0; i < 8; i++) { // Ulangi setiap byte dari alamat 8-byte.
        if (deviceAddress[i] < 0x10) Serial.print("0"); // Tambahkan nol di depan untuk nilai heksadesimal satu
digit.
        Serial.print(deviceAddress[i], HEX); // Cetak byte dalam format heksadesimal.
    }
    Serial.println(); // Baris baru setelah mencetak alamat lengkap.
}
```

```
        digitalWrite(buzzer,HIGH); // Nyalakan buzzer.
        delay(100);                // Biarkan menyala selama 100ms.
        digitalWrite(buzzer,LOW);  // Matikan buzzer.
        delay(100);                // Biarkan mati selama 100ms
(menciptakan suara bip).
    }
    // Tidak ada penundaan eksplisit di akhir loop, karena interval
pembaruan ThingSpeak dan interval LCD
    // sudah memperkenalkan penundaan, membuat loop cukup efisien.
}

// Fungsi pembantu untuk mencetak alamat perangkat DS18B20 ke monitor
serial.
void printAddress(DeviceAddress deviceAddress) {
    for (uint8_t i = 0; i < 8; i++) { // Ulangi setiap byte dari
alamat 8-byte.
        if (deviceAddress[i] < 0x10) Serial.print("0"); // Tambahkan
nol di depan untuk nilai heksadesimal satu digit.
        Serial.print(deviceAddress[i], HEX); // Cetak byte dalam
format heksadesimal.
    }
    Serial.println(); // Baris baru setelah mencetak alamat lengkap.
}
```