LAMPIRAN

PZEM-004T V3.0 User Manual

Overview

This document describes the specification of the PZEM-004T AC communication module, the module is mainly used for measuring AC voltage, current, active power, frequency, power factor and active energy, the module is without display function, the data is read through the TTL interface.

PZEM-004T-10A: Measuring Range 10A (Built-in Shunt)

PZEM-004T-100A: Measuring Range 100A (external transformer)

1. Function description

1.1 Voltage

1.1.1 Measuring range:80～260V

1.1.2 Resolution: 0.1V

1.1.3 Measurement accuracy: 0.5%

1.2 Current

1.2.1 Measuring range: 0～10A（PZEM-004T-10A）; 0～100A（PZEM-004T-100A）

1.2.2 Starting measure current: 0.01A（PZEM-004T-10A）; 0.02A（PZEM-004T-100A）

1.2.3 Resolution: 0.001A

1.2.4 Measurement accuracy: 0.5%

1.3 Active power

1.3.1 Measuring range: 0～2.3kW（PZEM-004T-10A）; 0～23kW（PZEM-004T-100A）

1.3.2 Starting measure power: 0.4W

1.3.3 Resolution: 0.1W

1.3.4 Display format:

＜1000W, it display one decimal, such as: 999.9W

≥1000W, it display only integer, such as: 1000W

1.3.5 Measurement accuracy: 0.5%

INSTITUT TEKNOLOGI KALIMANTAN

About This Guide

This document provides introduction to the specifications of ESP32 hardware.
The document structure is as follows:

Chapter	Title	Subject
Chapter 1	Overview	An overview of ESP32, including featured solutions, basic and advanced features, applications and development support
Chapter 2	Pin Definitions	Introduction to the pin layout and descriptions
Chapter 3	Functional Description	Description of the major functional modules
Chapter 4	Peripheral Interface	Description of the peripheral interfaces integrated on ESP32
Chapter 5	Electrical Characteristics	The electrical characteristics and data of ESP32
Chapter 6	Package Information	The package details of ESP32
Chapter 7	Supported Resources	The related documents and community resources for ESP32
Appendix	Touch Sensor	The touch sensor design and layout guidelines

Release Notes

Date	Version	Release notes
2016.08	V1.0	First release

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.
All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein. The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.
All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.
Copyright © 2016 Espressif Inc. All rights reserved.

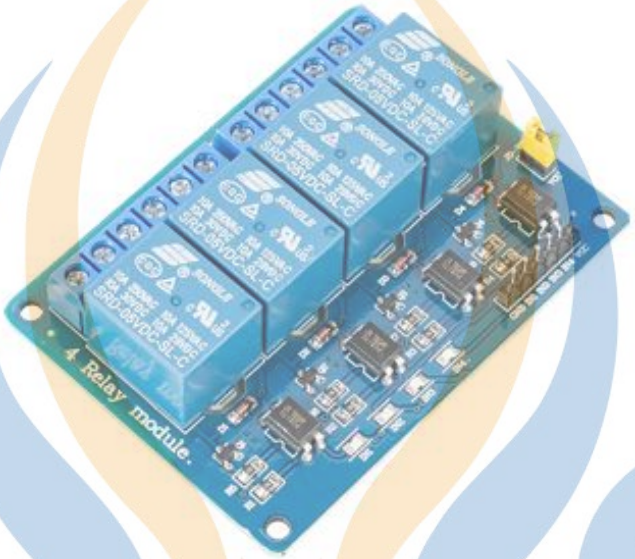
PZEM-017 300V 300A Shunt+USB Cable Car Voltmeter DC Ampere Meter RS485

Model Number:
PZEM-017
Display Type:
Digital Only
Measuring Range:
0.02-300a
Accuracy Class:
1.0grade
Supply Voltage:
0.05-300V DC
Operating Temperature:
-20'C ~ +60'C
Dimensions:
90*60.5*23mm(L*W*H)
Voltage test range:
0.05-300V
Voltage resolution:
0.01V
Current resolution:
0.01A
Power test range:
0.1-3kW
Power resolution:
0.1W
Energy test range:
0-9999kwh
Energy resolution:
1wh
Certificate:
CE FCC
Communication:
RS485 Port MODBUS-RTU Protocol

Product Type	PZEM-017
Current Range	0-300A
Accessory	300A shunt+USB cable
Voltage Range	DC 0-300V
Power Range	0-90KW
Accuracy	±1%
Energy Range	0-9999Kwh
Weight	299g

INSTITUT TEKNOLOGI KALIMANTAN

4 Channel 5V Relay Module



This is a LOW Level 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by micro-controller.

Features

- Power Supply Voltage: 5VDC, 12VDC
- Current: Greater than 100mA
- Load: 250V 10A AC or 30V 10A DC
- Size: 73mm x 54mm x 19.5mm (LxWxH)
- Equipped with mounting holes around, hole diameter 3.1mm
- Relay Type: Single Pole Double Throw (SPDT)
- Optocoupler isolation, good anti-interference capability
- When input is at low level, the relay is off, and the indicator light is on.
- Standard interface that can be directly controlled by microcontroller (8051, AVR, PIC, DSP, ARM, MSP430, TTL logic)

INSTITUT TEKNOLOGI
KALIMANTAN

PRELIMINARY

DALLAS SEMICONDUCTOR

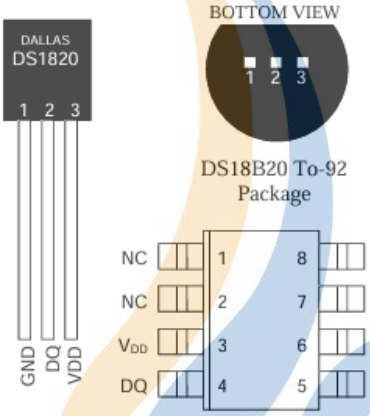
DS18B20
Programmable Resolution 1-Wire® Digital Thermometer

www.dalsemi.com

FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +257°F
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



DS18B20 To-92 Package

DS18B20Z
8-Pin SOIC (150 mil)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

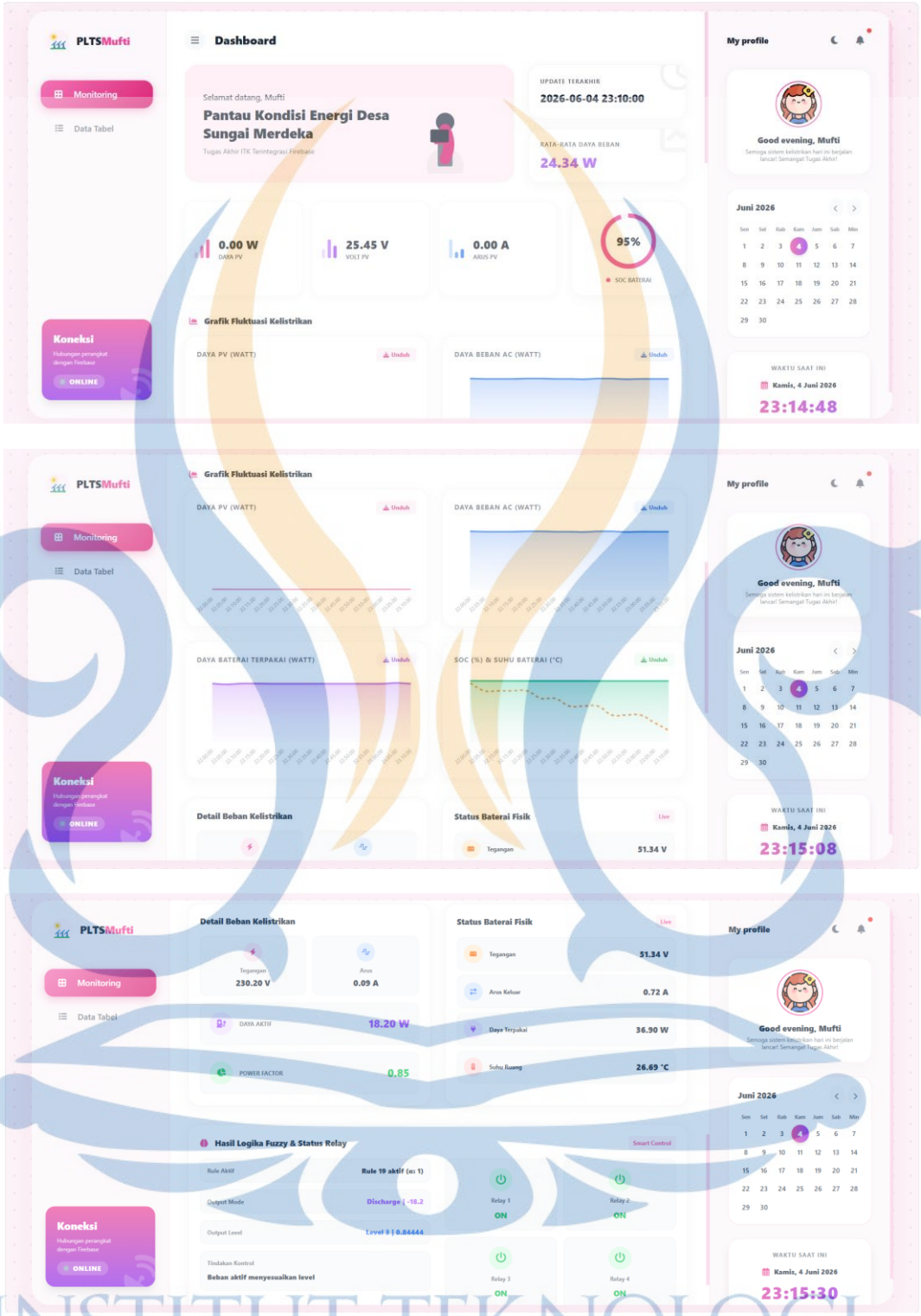
DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

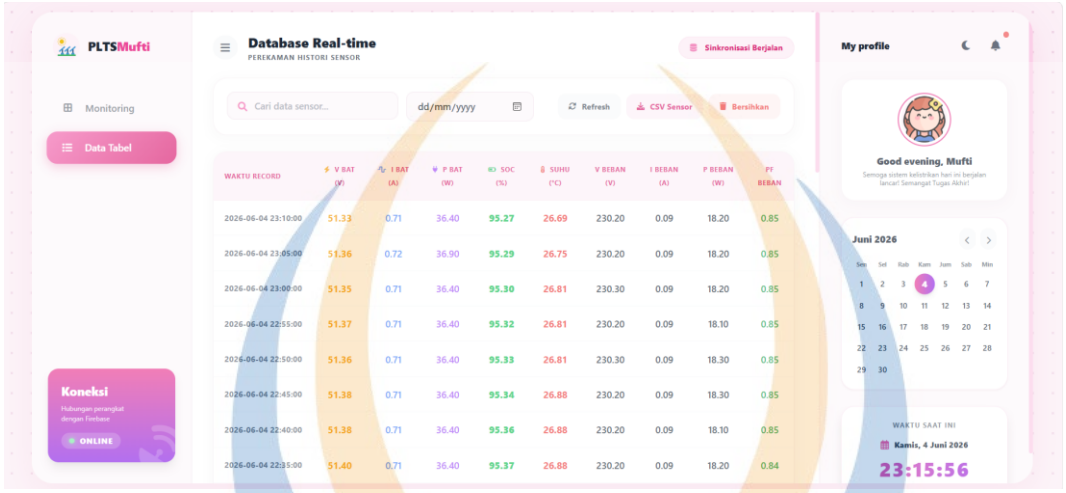
Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

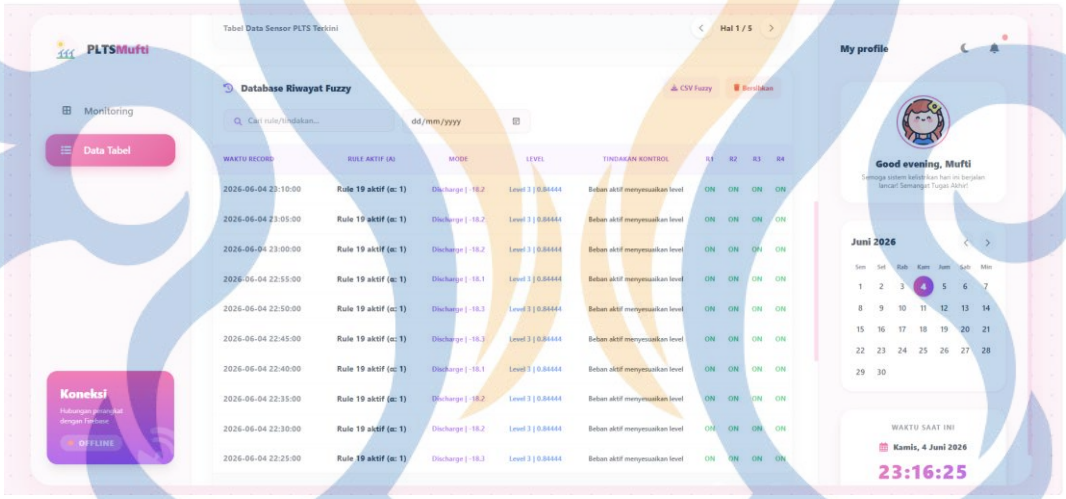
Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

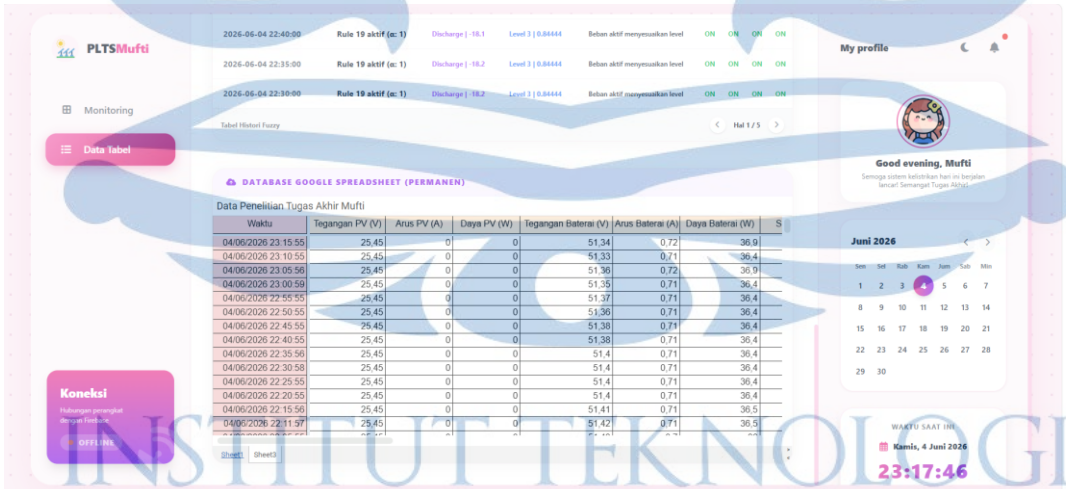
INSTITUT TEKNOLOGI KALIMANTAN



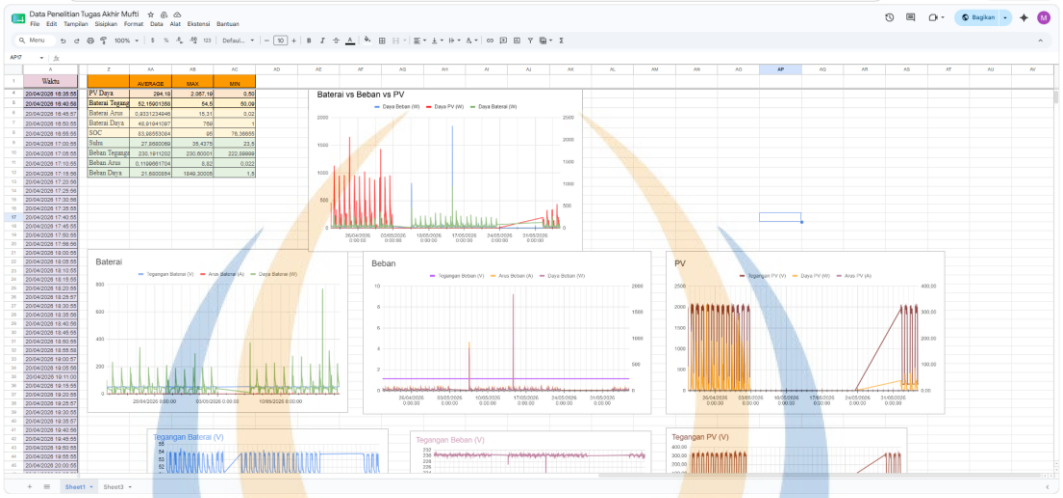
INSTITUT TEKNOLOGI KALIMANTAN

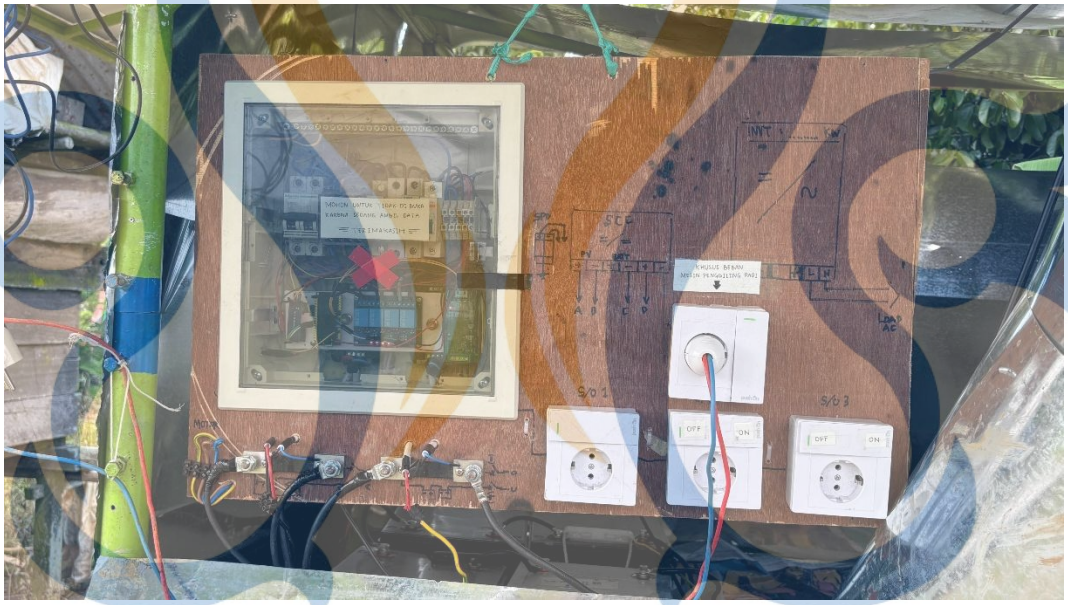


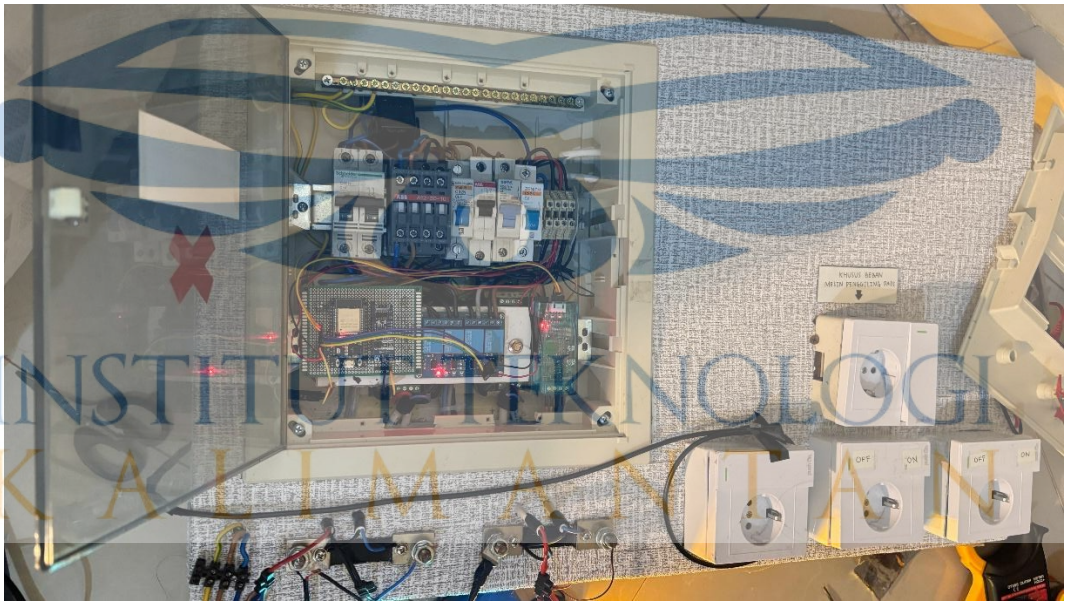




INSTITUT TEKNOLOGI KALIMANTAN







INSTITUT TEKNOLOGI KALIMANTAN

No.	Kode Program
1	`#include <ModbusMaster.h>`
2	`#include <PZEM004Tv30.h>`
3	`#include <HardwareSerial.h>`
4	`#include <OneWire.h>`
5	`#include <DallasTemperature.h>`
6	`#include <Fuzzy.h>`
7	
8	`// ==========================`
9	`// TAMBAHAN: LIBRARIES FIREBASE, WIFI, & TIME`
10	`// ==========================`
11	`#include <WiFi.h>`
12	`#include <esp_now.h>`
13	`#include <esp_arduino_version.h>`
14	`#include <Firebase_ESP_Client.h>`
15	`#include "time.h"`
16	
17	`// Menyediakan Helper untuk Token Generation & RTDB`
18	`#include "addons/TokenHelper.h"`
19	`#include "addons/RTDBHelper.h"`
20	
21	`// ==========================`
22	`// TAMBAHAN: KREDENSIAL WIFI & FIREBASE`
23	`// ==========================`
24	`#define WIFI_SSID "Agrinergy.ITK"`
25	`#define WIFI_PASSWORD "K@mpusM3rdeka!"`
26	
27	`#define API_KEY "AIzaSyD3juElH5KEWPCrK-uR8AIh7t2ZOf2Uaas"`
28	`#define DATABASE_URL "data-penelitian-tugas-akhir-default-rtdb.firebaseio.com"`
29	
30	`// ==========================`
31	`// PENGATURAN NTP`
32	`// ==========================`
33	`const char* ntpServer = "pool.ntp.org";`
34	`const long  gmtOffset_sec = 28800; // WITA (Balikpapan GMT+8)`
35	`const int   daylightOffset_sec = 0;`
36	
37	`// Objek Firebase`
38	`FirebaseData fbdo;`
39	`FirebaseAuth auth;`
40	`FirebaseConfig config;`
41	`bool signupOK = false;`
42	
43	`// ==========================`
44	`// RELAY`

INSTITUT TEKNOLOGI KALIMANTAN

```
// ========================
#define RELAY1 12
#define RELAY2 14
#define RELAY3 27
#define RELAY4 26

// ========================
// PZEM-017 via MAX485
// ========================
#define MAX485_RE_DE 4
#define RXD2 16
#define TXD2 17

HardwareSerial mySerial(2);
ModbusMaster pzemPV;
ModbusMaster pzemBat;

// ========================
// PZEM-004T
// ========================
#define PZEM004T_RX 32
#define PZEM004T_TX 33

HardwareSerial serialPZEM004T(1);
PZEM004Tv30 pzemAC(serialPZEM004T, PZEM004T_RX, PZEM004T_TX);

// ========================
// DS18B20
// ========================
#define DS18B20_PIN 13
OneWire oneWire(DS18B20_PIN);
DallasTemperature sensors(&oneWire);

// ========================
// PARAMETER BATTERY SOC & WAKTU
// ========================
const float BATTERY_CAPACITY_AH = 400.0;

// SOC tidak langsung diset 100%
// SOC awal akan ditentukan dari tegangan baterai pertama kali terbaca
float socBattery = 0.0;

// Penanda apakah SOC awal sudah ditentukan
bool socInitialized = false;

// Waktu terakhir perhitungan SOC
```

INSTITUT TEKNOLOGI

```
91  unsigned long lastSocMillis = 0;
92
93  // Tegangan baterai sebelumnya
94  float lastVBat = 0.0;
95
96  // Threshold perubahan tegangan untuk menentukan arah coulomb counting
97  const float DV_CHARGE_THRESHOLD = 0.04;      // jika tegangan naik minimal 0.04 V, dianggap charging
98  const float DV_DISCHARGE_THRESHOLD = -0.04;  // jika tegangan turun minimal 0.04 V, dianggap discharging
99
100 // Arus sangat kecil dianggap idle / tidak signifikan
101 const float I_IDLE = 0.10;
102
103 // Efisiensi charging
104 const float CHARGE_EFFICIENCY = 0.90;
105
106 unsigned long previousMillisData = 0;
107 const long intervalData = 300000;          // 5 Menit Data (300.000 ms)
108
109 unsigned long previousMillisStatus = 0;
110 const long intervalStatus = 10000;         // 10 Detik Heartbeat (10.000 ms)
111
112 // ====================================================
113 // SISTEM FUZZY LEVEL BEBAN AKTIF
114 // Input  : Daya PV, SoC, Daya Beban
115 // Output : Level Beban Aktif
116 // Mode operasi ditentukan terpisah berdasarkan selisih daya
117 // ====================================================
118 Fuzzy *fuzzy = new Fuzzy();
119
120 // Input fuzzy
121 FuzzyInput *fuzzySocInput;
122 FuzzyInput *fuzzyPvInput;
123 FuzzyInput *fuzzyLoadInput;
124
125 // Output fuzzy
126 FuzzyOutput *fuzzyLevelOut;
127
128 // Fuzzy set input
129 FuzzySet *soc_low, *soc_med, *soc_high;
130 FuzzySet *pv_low, *pv_med, *pv_high;
131 FuzzySet *load_light, *load_med, *load_heavy;
132
133 // Fuzzy set output
134 FuzzySet *level1, *level2, *level3;
```

INSTITUT TEKNOLOGI KALIMANTAN

```
135
136 // Variabel hasil fuzzy
137 float fuzzyModeValue = 0.0;
138 float fuzzyLevelValue = 0.0;
139 String fuzzyModeLabel = "-";
140 String fuzzyLevelLabel = "-";
141
142 // ==========================
143 // DATA RULE AKTIF UNTUK FIREBASE
144 // ==========================
145 String ruleAktifText = "-";
146 int ruleAktifDominan = 0;
147 float alphaRuleAktifDominan = 0.0;
148
149 // ==========================
150 // TEGANGAN PV DARI ESP TEMAN VIA ESP-NOW
151 // ==========================
152 // PZEM PV hanya dipakai untuk membaca arus PV.
153 // Tegangan PV diterima dari ESP teman melalui ESP-NOW.
154 // ESP teman disarankan mengirim tegangan PV setiap 1-5 detik,
    // agar saat data utama diambil tiap 5 menit, nilai tegangan yang dipakai
155 masih baru.
156 typedef struct struct_message_pv {
157   float teganganPV;
158 } struct_message_pv;
159
160 struct_message_pv dataTeganganPVMasuk;
161
162 float teganganPVKalibrasi = 0.0;
163 unsigned long lastTerimaTeganganPV = 0;
164 bool teganganPVSiap = false;
    const unsigned long BATAS_VALID_TEGANGAN_PV = 60000; // data tegangan valid
165 maksimal 60 detik
    String statusTeganganPVESPNow = "Belum menerima tegangan PV dari ESP
166 teman";
167
168 // ==========================
169 // PARAMETER PROTEKSI SUHU BATERAI
170 // ==========================
171 const float BATTERY_TEMP_HIGH = 45.0;   // batas proteksi suhu baterai (°C)
172 bool proteksiSuhuAktif = false;
173 String statusProteksiSuhu = "Normal";
174
175 // ======================================================
176 // FUNGSI MEMBERSHIP MANUAL UNTUK MENAMPILKAN RULE AKTIF
177 // ======================================================
178 float trapmf(float x, float a, float b, float c, float d) {
```

INSTITUT TEKNOLOGI KALIMANTAN

```
  if (x < a || x > d) return 0.0;

  if (x >= b && x <= c) return 1.0;

  if (x >= a && x < b) {
    if (b == a) return 1.0;
    return (x - a) / (b - a);
  }

  if (x > c && x <= d) {
    if (d == c) return 1.0;
    return (d - x) / (d - c);
  }

  return 0.0;
}

float min3(float a, float b, float c) {
  return min(a, min(b, c));
}

// ============================================================
// LABEL BERDASARKAN RULE
// ============================================================
String getModeRule(int rule) {
  if (rule == 1 || rule == 2 || rule == 3 ||
      rule == 10 || rule == 11 || rule == 12 ||
      rule == 19 || rule == 20 || rule == 21) {
    return "Discharge";
  }

  if (rule == 5 || rule == 6 || rule == 9 ||
      rule == 13 || rule == 14 || rule == 15 || rule == 18 ||
      rule == 22 || rule == 23 || rule == 24 ||
      rule == 25 || rule == 26 || rule == 27) {
    return "Supply";
  }

  if (rule == 4 || rule == 7 || rule == 8 ||
      rule == 16 || rule == 17) {
    return "Charge";
  }

  return "-";
}

```

INSTITUT TEKNOLOGI KALIMANTAN

```
225 String getLevelRule(int rule) {
226   if (rule == 1 || rule == 2 || rule == 3 ||
227       rule == 5 || rule == 6 ||
228       rule == 11 || rule == 12) {
229     return "Level 1";
230   }
231
232   if (rule == 4 || rule == 7 || rule == 8 || rule == 9 ||
233       rule == 10 ||
234       rule == 13 || rule == 14 || rule == 15 ||
235       rule == 20 || rule == 21) {
236     return "Level 2";
237   }
238
239   if (rule == 16 || rule == 17 || rule == 18 ||
240       rule == 19 ||
241       rule == 22 || rule == 23 || rule == 24 ||
242       rule == 25 || rule == 26 || rule == 27) {
243     return "Level 3";
244   }
245
246   return "-";
247 }
248
249 // ========================================================
250 // LABEL HASIL DEFUZZIFIKASI
251 // ========================================================
252 String labelLevel(float nilaiLevel) {
253   if (nilaiLevel < 0.40) return "Level 1";
254   else if (nilaiLevel < 0.70) return "Level 2";
255   else return "Level 3";
256 }
257
258 // ========================================================
259 // PENENTUAN MODE OPERASI BERDASARKAN SELISIH DAYA
260 // Pnet = Ppv - Pbeban
261 // Mode operasi tidak lagi ditentukan oleh fuzzy.
262 // Fuzzy hanya digunakan untuk menentukan level beban aktif.
263 // ========================================================
264 String tentukanModeOperasi(float pPV, float pBeban, float socVal) {
265   if (pPV < pBeban) {
266     return "Discharge";
267   }
268   else if (pPV >= pBeban && socVal < 95.0) {
269     return "Charge";
270   }
```

INSTITUT TEKNOLOGI KALIMANTAN

```
271     else {
272       return "Supply";
273     }
274   }
275
276   // ========================================================
277   // MENAMBAHKAN RULE FUZZY
278   // ========================================================
      void addFuzzyRuleCustom(int num, FuzzySet* a, FuzzySet* b, FuzzySet* c,
279   FuzzySet* oLevel) {
280     FuzzyRuleAntecedent *ant1 = new FuzzyRuleAntecedent();
281     ant1->joinWithAND(a, b);
282
283     FuzzyRuleAntecedent *ant2 = new FuzzyRuleAntecedent();
284     ant2->joinWithAND(ant1, c);
285
286     FuzzyRuleConsequent *cons = new FuzzyRuleConsequent();
287     cons->addOutput(oLevel);
288
289     fuzzy->addFuzzyRule(new FuzzyRule(num, ant2, cons));
290   }
291
292   // ========================================================
293   // ESP-NOW: MENERIMA TEGANGAN PV DARI ESP TEMAN
294   // ========================================================
295   void simpanDataTeganganPV(const uint8_t *incomingData, int len) {
296     if (len == sizeof(dataTeganganPVMasuk)) {
        memcpy(&dataTeganganPVMasuk, incomingData,
297   sizeof(dataTeganganPVMasuk));
298
299       teganganPVKalibrasi = dataTeganganPVMasuk.teganganPV;
300       lastTerimaTeganganPV = millis();
301       teganganPVSiap = true;
302       statusTeganganPVESPNow = "Tegangan PV dari ESP teman diterima";
303
304       Serial.print("ESP-NOW PV diterima | Tegangan PV: ");
305       Serial.print(teganganPVKalibrasi, 2);
306       Serial.println(" V");
307     }
308     else {
309       statusTeganganPVESPNow = "Format data ESP-NOW tidak sesuai";
        Serial.println("Data ESP-NOW diterima, tetapi ukuran data tidak
310   sesuai.");
311     }
312   }
313
314   #if ESP_ARDUINO_VERSION >= ESP_ARDUINO_VERSION_VAL(3, 0, 0)
```

INSTITUT TEKNOLOGI KALIMANTAN

```
void OnDataRecv(const esp_now_recv_info_t *recv_info, const uint8_t
*incomingData, int len) {
  simpanDataTeganganPV(incomingData, len);
}
#else
void OnDataRecv(const uint8_t *mac, const uint8_t *incomingData, int len) {
  simpanDataTeganganPV(incomingData, len);
}
#endif

// ================================================
// SETUP FUZZY
// ================================================
void setupFuzzy() {
  // ================ INPUT FUZZY ================
  fuzzySocInput = new FuzzyInput(1);
  fuzzyPvInput = new FuzzyInput(2);
  fuzzyLoadInput = new FuzzyInput(3);

  // SoC
  soc_low  = new FuzzySet(0, 0, 35, 50);
  soc_med  = new FuzzySet(40, 60, 60, 80);
  soc_high = new FuzzySet(70, 85, 100, 100);

  fuzzySocInput->addFuzzySet(soc_low);
  fuzzySocInput->addFuzzySet(soc_med);
  fuzzySocInput->addFuzzySet(soc_high);
  fuzzy->addFuzzyInput(fuzzySocInput);

  // Daya PV
  pv_low  = new FuzzySet(0, 0, 300, 800);
  pv_med  = new FuzzySet(500, 1100, 1100, 1700);
  pv_high = new FuzzySet(1400, 1800, 2200, 2200);

  fuzzyPvInput->addFuzzySet(pv_low);
  fuzzyPvInput->addFuzzySet(pv_med);
  fuzzyPvInput->addFuzzySet(pv_high);
  fuzzy->addFuzzyInput(fuzzyPvInput);

  // Daya Beban
  load_light = new FuzzySet(0, 0, 30, 150);
  load_med   = new FuzzySet(100, 600, 600, 1200);
  load_heavy = new FuzzySet(900, 1500, 2000, 2000);

  fuzzyLoadInput->addFuzzySet(load_light);
  fuzzyLoadInput->addFuzzySet(load_med);
```

INSTITUT TEKNOLOGI KALIMANTAN

```
360   fuzzyLoadInput->addFuzzySet(load_heavy);
361   fuzzy->addFuzzyInput(fuzzyLoadInput);
362
363   // ================ OUTPUT FUZZY ================
364   // Fuzzy hanya menghasilkan Level Beban Aktif.
365   // Mode operasi ditentukan terpisah berdasarkan selisih daya.
366   fuzzyLevelOut = new FuzzyOutput(1);
367
368   // Level Beban Aktif
369   level1 = new FuzzySet(0, 0, 0.20, 0.40);
370   level2 = new FuzzySet(0.30, 0.50, 0.50, 0.70);
371   level3 = new FuzzySet(0.60, 0.80, 1.00, 1.00);
372
373   fuzzyLevelOut->addFuzzySet(level1);
374   fuzzyLevelOut->addFuzzySet(level2);
375   fuzzyLevelOut->addFuzzySet(level3);
376   fuzzy->addFuzzyOutput(fuzzyLevelOut);
377
378   // ================ RULE BASE TERBARU ================
379   addFuzzyRuleCustom(1, soc_low, pv_low, load_light, level1);
380   addFuzzyRuleCustom(2, soc_low, pv_low, load_med, level1);
381   addFuzzyRuleCustom(3, soc_low, pv_low, load_heavy, level1);
382
383   addFuzzyRuleCustom(4, soc_low, pv_med, load_light, level2);
384   addFuzzyRuleCustom(5, soc_low, pv_med, load_med, level1);
385   addFuzzyRuleCustom(6, soc_low, pv_med, load_heavy, level1);
386
387   addFuzzyRuleCustom(7, soc_low, pv_high, load_light, level2);
388   addFuzzyRuleCustom(8, soc_low, pv_high, load_med, level2);
389   addFuzzyRuleCustom(9, soc_low, pv_high, load_heavy, level2);
390
391   addFuzzyRuleCustom(10, soc_med, pv_low, load_light, level2);
392   addFuzzyRuleCustom(11, soc_med, pv_low, load_med, level1);
393   addFuzzyRuleCustom(12, soc_med, pv_low, load_heavy, level1);
394
395   addFuzzyRuleCustom(13, soc_med, pv_med, load_light, level2);
396   addFuzzyRuleCustom(14, soc_med, pv_med, load_med, level2);
397   addFuzzyRuleCustom(15, soc_med, pv_med, load_heavy, level2);
398
399   addFuzzyRuleCustom(16, soc_med, pv_high, load_light, level3);
400   addFuzzyRuleCustom(17, soc_med, pv_high, load_med, level3);
401   addFuzzyRuleCustom(18, soc_med, pv_high, load_heavy, level3);
402
403   addFuzzyRuleCustom(19, soc_high, pv_low, load_light, level3);
404   addFuzzyRuleCustom(20, soc_high, pv_low, load_med, level2);
405   addFuzzyRuleCustom(21, soc_high, pv_low, load_heavy, level2);
```

```
406
407   addFuzzyRuleCustom(22, soc_high, pv_med, load_light, level3);
408   addFuzzyRuleCustom(23, soc_high, pv_med, load_med, level3);
409   addFuzzyRuleCustom(24, soc_high, pv_med, load_heavy, level3);
410
411   addFuzzyRuleCustom(25, soc_high, pv_high, load_light, level3);
412   addFuzzyRuleCustom(26, soc_high, pv_high, load_med, level3);
413   addFuzzyRuleCustom(27, soc_high, pv_high, load_heavy, level3);
414
415   Serial.println("Fuzzy siap digunakan.");
416 }
417
418 // ====================================================
419 // KONTROL RELAY BERDASARKAN LEVEL BEBAN AKTIF
420 // Relay aktif LOW:
421 // LOW  = ON
422 // HIGH = OFF
423 //
424 // Beban 1 = prioritas tinggi  -> Relay 1
425 // Beban 2 = prioritas sedang  -> Relay 2
426 // Beban 3 = prioritas rendah  -> Relay 3
427 // Relay 4 = cadangan prioritas rendah
428 // ====================================================
429 void kontrolRelayByLevel(String levelLabel) {
430   if (levelLabel == "Level 1") {
431     // Level 1: hanya beban prioritas tinggi yang aktif
432     digitalWrite(RELAY1, LOW);   // Beban 1 ON
433     digitalWrite(RELAY2, HIGH);  // Beban 2 OFF
434     digitalWrite(RELAY3, HIGH);  // Beban 3 OFF
435     digitalWrite(RELAY4, HIGH);  // Relay cadangan OFF
436
    Serial.println("Kontrol Relay: Level 1 -> Beban prioritas tinggi
437 aktif");
438   }
439   else if (levelLabel == "Level 2") {
440     // Level 2: beban prioritas tinggi dan sedang aktif
441     digitalWrite(RELAY1, LOW);   // Beban 1 ON
442     digitalWrite(RELAY2, LOW);   // Beban 2 ON
443     digitalWrite(RELAY3, HIGH);  // Beban 3 OFF
444     digitalWrite(RELAY4, HIGH);  // Relay cadangan OFF
445
    Serial.println("Kontrol Relay: Level 2 -> Beban prioritas tinggi dan
446 sedang aktif");
447   }
448   else if (levelLabel == "Level 3") {
449     // Level 3: seluruh beban aktif
450     digitalWrite(RELAY1, LOW);   // Beban 1 ON
```

INSTITUT TEKNOLOGI KALIMANTAN

```
451       digitalWrite(RELAY2, LOW);    // Beban 2 ON
452       digitalWrite(RELAY3, LOW);    // Beban 3 ON
          digitalWrite(RELAY4, LOW);    // Relay cadangan ON sebagai prioritas
453   rendah
454
455       Serial.println("Kontrol Relay: Level 3 -> Semua beban aktif");
456     }
457     else {
458       // Kondisi aman jika level tidak valid
459       digitalWrite(RELAY1, HIGH);
460       digitalWrite(RELAY2, HIGH);
461       digitalWrite(RELAY3, HIGH);
462       digitalWrite(RELAY4, HIGH);
463
          Serial.println("Kontrol Relay: Level tidak valid -> Semua beban
464   dimatikan");
465     }
466   }
467
468   // ========================================================
469   // PROTEKSI SUHU BATERAI
470   // Kondisi proteksi aktif jika suhu baterai >= 45°C.
471   // Pada kondisi ini, beban non-prioritas dilepas.
472   // Relay aktif LOW: LOW = ON, HIGH = OFF.
473   // ========================================================
474   void kontrolRelayProteksiSuhu() {
475     digitalWrite(RELAY1, LOW);    // Beban prioritas tinggi tetap aktif
476     digitalWrite(RELAY2, HIGH);   // Beban prioritas sedang OFF
477     digitalWrite(RELAY3, HIGH);   // Beban prioritas rendah OFF
478     digitalWrite(RELAY4, HIGH);   // Relay cadangan OFF
479
          Serial.println("Kontrol Relay: Proteksi suhu -> Beban non-prioritas
480   dilepas");
481   }
482
483   void prosesProteksiSuhu(float tempC) {
484     proteksiSuhuAktif = true;
485     statusProteksiSuhu = "Proteksi Suhu Aktif";
486
          // Nilai fuzzy diberi tanda khusus karena fuzzy tidak digunakan saat
487   proteksi suhu aktif
488     fuzzyModeValue = -1.0;
489     fuzzyLevelValue = -1.0;
490     fuzzyModeLabel = "Proteksi Suhu";
491     fuzzyLevelLabel = "Level 1";
492
493     Serial.println("HASIL FUZZY");
494     Serial.print("Proteksi Suhu: AKTIF | Suhu Baterai: ");
```

INSTITUT TEKNOLOGI KALIMANTAN

```
495     Serial.print(tempC, 2);
496     Serial.print(" C | Batas: ");
497     Serial.print(BATTERY_TEMP_HIGH, 2);
498     Serial.println(" C");
499     Serial.println("Fuzzy tidak dijalankan karena suhu baterai melewati batas proteksi.");
500     Serial.println("Rule aktif: Proteksi suhu baterai aktif");
501
502     ruleAktifText = "Proteksi suhu baterai aktif";
503     ruleAktifDominan = 0;
504     alphaRuleAktifDominan = -1.0;
505
506     Serial.println("Output Mode: Proteksi Suhu | -");
507     Serial.println("Output Level: Level 1 | -");
508
509     kontrolRelayProteksiSuhu();
510   }
511
512   // ======================================================
513   // MENAMPILKAN RULE AKTIF SECARA RINGKAS
514   // ======================================================
515   void tampilkanRuleAktif(float socVal, float pvVal, float loadVal) {
516     float socDeg[3];
517     float pvDeg[3];
518     float loadDeg[3];
519
520     socDeg[0] = trapmf(socVal, 0, 0, 35, 50);
521     socDeg[1] = trapmf(socVal, 40, 60, 60, 80);
522     socDeg[2] = trapmf(socVal, 70, 85, 100, 100);
523
524     pvDeg[0] = trapmf(pvVal, 0, 0, 300, 800);
525     pvDeg[1] = trapmf(pvVal, 500, 1100, 1100, 1700);
526     pvDeg[2] = trapmf(pvVal, 1400, 1800, 2200, 2200);
527
528     loadDeg[0] = trapmf(loadVal, 0, 0, 30, 150);
529     loadDeg[1] = trapmf(loadVal, 100, 600, 600, 1200);
530     loadDeg[2] = trapmf(loadVal, 900, 1500, 2000, 2000);
531
532     int ruleNumber[3][3][3] = {
533       {
534         {1, 2, 3},
535         {4, 5, 6},
536         {7, 8, 9}
537       },
538       {
539         {10, 11, 12},
```

INSTITUT TEKNOLOGI KALIMANTAN

```
      {13, 14, 15},
      {16, 17, 18}
    },
    {
      {19, 20, 21},
      {22, 23, 24},
      {25, 26, 27}
    }
  };

  bool adaRuleAktif = false;

  // Reset data rule aktif setiap proses fuzzy
  ruleAktifText = "";
  ruleAktifDominan = 0;
  alphaRuleAktifDominan = 0.0;

  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
      for (int k = 0; k < 3; k++) {
        float alpha = min3(socDeg[i], pvDeg[j], loadDeg[k]);

        if (alpha > 0.001) {
          adaRuleAktif = true;
          int nomorRule = ruleNumber[i][j][k];

          Serial.print("Rule aktif: Rule ");
          Serial.print(nomorRule);
          Serial.print(" aktif | alpha = ");
          Serial.println(alpha, 3);

          // Simpan semua rule aktif untuk Firebase
          if (ruleAktifText.length() > 0) {
            ruleAktifText += " ; ";
          }

          ruleAktifText += "Rule ";
          ruleAktifText += String(nomorRule);
          ruleAktifText += " aktif | alpha = ";
          ruleAktifText += String(alpha, 3);

          // Simpan rule dominan berdasarkan alpha terbesar
          if (alpha > alphaRuleAktifDominan) {
            alphaRuleAktifDominan = alpha;
            ruleAktifDominan = nomorRule;
          }
```

INSTITUT TEKNOLOGI KALIMANTAN

```
586           }
587         }
588       }
589     }
590
591     if (!adaRuleAktif) {
592       Serial.println("Rule aktif: Tidak ada rule aktif");
593
594       ruleAktifText = "Tidak ada rule aktif";
595       ruleAktifDominan = 0;
596       alphaRuleAktifDominan = 0.0;
597     }
598   }
599
600   // ======================================================
601   // PROSES FUZZY DARI DATA SENSOR
602   // ======================================================
603   void prosesFuzzySensor(float pvVal, float socVal, float loadVal) {
604     proteksiSuhuAktif = false;
605     statusProteksiSuhu = "Normal";
606
607     // Pembatas agar nilai input tidak keluar dari domain fuzzy
608     pvVal = constrain(pvVal, 0, 2200);
609     socVal = constrain(socVal, 0, 100);
610     loadVal = constrain(loadVal, 0, 2000);
611
612     fuzzy->setInput(1, socVal);
613     fuzzy->setInput(2, pvVal);
614     fuzzy->setInput(3, loadVal);
615
616     fuzzy->fuzzify();
617
618     // Mode operasi ditentukan berdasarkan selisih daya, bukan dari fuzzy
619     fuzzyModeValue = pvVal - loadVal; // Pnet = Ppv - Pbeban
620     fuzzyModeLabel = tentukanModeOperasi(pvVal, loadVal, socVal);
621
622     // Fuzzy hanya digunakan untuk menentukan level beban aktif
623     fuzzyLevelValue = fuzzy->defuzzify(1);
624     fuzzyLevelLabel = labelLevel(fuzzyLevelValue);
625
626     Serial.println("HASIL FUZZY");
627     Serial.print("Input Fuzzy -> PV: ");
628     Serial.print(pvVal, 2);
629     Serial.print(" W | SoC: ");
630     Serial.print(socVal, 2);
631     Serial.print(" % | Beban: ");
```

```
632     Serial.print(loadVal, 2);
633     Serial.println(" W");
634
635     tampilkanRuleAktif(socVal, pvVal, loadVal);
636
637     Serial.print("Mode Operasi: ");
638     Serial.print(fuzzyModeLabel);
639     Serial.print(" | Pnet = ");
640     Serial.print(fuzzyModeValue, 2);
641     Serial.println(" W");
642
643     Serial.print("Output Level: ");
644     Serial.print(fuzzyLevelLabel);
645     Serial.print(" | ");
646     Serial.println(fuzzyLevelValue, 3);
647
648     // Kontrol relay berdasarkan hasil fuzzy Level Beban Aktif
649     kontrolRelayByLevel(fuzzyLevelLabel);
650   }
651
652
653
654   // ==========================
655   // MAX485 control
656   // ==========================
657   void preTransmission() {
658     digitalWrite(MAX485_RE_DE, HIGH);
659     delay(2);
660   }
661
662   void postTransmission() {
663     delay(2);
664     digitalWrite(MAX485_RE_DE, LOW);
665   }
666
667   // ==========================
668   // Relay functions
669   // ==========================
670   void relayAllOff() {
671     digitalWrite(RELAY1, HIGH);
672     digitalWrite(RELAY2, HIGH);
673     digitalWrite(RELAY3, HIGH);
674     digitalWrite(RELAY4, HIGH);
675   }
676
677   void relayAllOn() {
```

INSTITUT TEKNOLOGI KALIMANTAN

```
678   digitalWrite(RELAY1, LOW);
679   digitalWrite(RELAY2, LOW);
680   digitalWrite(RELAY3, LOW);
681   digitalWrite(RELAY4, LOW);
682 }
683
684 void printRelayStatus() {
685   Serial.println("Status Relay:");
      Serial.print("Relay 1: "); Serial.println(digitalRead(RELAY1) == LOW ?
686 "ON" : "OFF");
      Serial.print("Relay 2: "); Serial.println(digitalRead(RELAY2) == LOW ?
687 "ON" : "OFF");
      Serial.print("Relay 3: "); Serial.println(digitalRead(RELAY3) == LOW ?
688 "ON" : "OFF");
      Serial.print("Relay 4: "); Serial.println(digitalRead(RELAY4) == LOW ?
689 "ON" : "OFF");
690 }
691
692 // ========================
693 // Read PZEM-017
694 // ========================
    bool readPZEM(ModbusMaster &node, float &voltage, float &current, float
695 &power, float &energy) {
696   uint8_t result = node.readInputRegisters(0x0000, 6);
697
698   if (result == node.ku8MBSuccess) {
699     voltage = node.getResponseBuffer(0) / 100.0;
700     current = node.getResponseBuffer(1) / 100.0;
701     power   = node.getResponseBuffer(2) / 10.0;
702     energy  = node.getResponseBuffer(3);
703     return true;
704   }
705   return false;
706 }
707
708 // ========================
709 // Read PZEM-017 ARUS SAJA UNTUK PV
710 // ========================
711 // Fungsi ini dipakai khusus untuk PZEM PV.
    // Tegangan PV tidak diambil dari PZEM karena tegangan rangkaian panel bisa
712 lebih dari 300 V.
713 bool readPZEMCurrentOnly(ModbusMaster &node, float &current) {
714   uint8_t result = node.readInputRegisters(0x0000, 6);
715
716   if (result == node.ku8MBSuccess) {
717     current = node.getResponseBuffer(1) / 100.0;
718     return true;
719   }
```

INSTITUT TEKNOLOGI KALIMANTAN

```
  return false;
}

// ========================
// ESTIMASI SOC AWAL DARI TEGANGAN
// ========================
// Fungsi ini dipakai saat alat pertama kali hidup.
// Jadi SOC awal tidak langsung 100%, tetapi mengikuti tegangan baterai pertama kali terbaca.
//
// Mapping ini disesuaikan dari pola data baterai kamu:
// tegangan lapangan sekitar 50.8 V sampai 54.5 V.
float estimateSocFromVoltage(float vBat) {
  if (vBat >= 51.60) return 100.0;
  else if (vBat >= 51.0) return 90.0;
  else if (vBat >= 50.40) return 80.0;
  else if (vBat >= 49.80) return 70.0;
  else if (vBat >= 49.20) return 60.0;
  else if (vBat >= 48.60) return 50.0;
  else if (vBat >= 48.00) return 40.0;
  else if (vBat >= 47.40) return 30.0;
  else return 20.0;
}


// ========================
// UPDATE SOC BATTERY
// ========================
// Prinsip:
// 1. SOC awal ditentukan dari tegangan pertama kali terbaca.
// 2. Setelah itu SOC dihitung dengan coulomb counting.
// 3. Karena PZEM-017 tidak membaca arah arus (+/-),
//    maka arah charge/discharge ditentukan dari perubahan tegangan baterai.
// 4. Tegangan hanya menjadi koreksi pelan, bukan langsung memaksa SOC menjadi 100%.
void updateBatterySOC(float vBat, float iBat) {
  unsigned long now = millis();

  // ========================
  // 1. INISIALISASI SOC AWAL
  // ========================
  if (!socInitialized) {
    socBattery = estimateSocFromVoltage(vBat);
    lastVBat = vBat;
    lastSocMillis = now;
    socInitialized = true;
```

INSTITUT TEKNOLOGI KALIMANTAN

```
764
765       Serial.print("SOC awal berdasarkan tegangan baterai: ");
766       Serial.print(socBattery);
767       Serial.println(" %");
768
769       return;
770     }
771
772     // ==========================
773     // 2. HITUNG SELISIH WAKTU
774     // ==========================
775     float dtHours = (now - lastSocMillis) / 3600000.0;
776     lastSocMillis = now;
777
778     // ==========================
779     // 3. HITUNG PERUBAHAN TEGANGAN
780     // ==========================
781     float dV = vBat - lastVBat;
782     lastVBat = vBat;
783
784     // ==========================
785     // 4. COULOMB COUNTING
786     // ==========================
787     // iBat dari PZEM bernilai positif, sehingga ini hanya besar perubahan
    SOC.
788     // Tanda + atau - ditentukan oleh perubahan tegangan dV.
789     float deltaSoc = (iBat * dtHours / BATTERY_CAPACITY_AH) * 100.0;
790
791     // Proteksi kalau ada pembacaan arus/waktu yang aneh
792     if (deltaSoc > 1.0) deltaSoc = 1.0;
793
794     // Estimasi SOC berdasarkan tegangan untuk koreksi pelan
795     float socVoltage = estimateSocFromVoltage(vBat);
796
797     // ==========================
798     // 5. LOGIKA CHARGE / DISCHARGE
799     // ==========================
800     if (iBat < I_IDLE) {
801       // Arus sangat kecil, SOC dikoreksi sangat pelan berdasarkan tegangan
802       socBattery = (0.995 * socBattery) + (0.005 * socVoltage);
803     }
804     else if (dV >= DV_CHARGE_THRESHOLD) {
805       // Tegangan naik cukup jelas, dianggap charging
806       socBattery += deltaSoc * CHARGE_EFFICIENCY;
807
808       // Koreksi pelan jika tegangan menunjukkan SOC lebih tinggi
```

INSTITUT TEKNOLOGI KALIMANTAN

```
809       if (socBattery < socVoltage) {
810         socBattery = (0.98 * socBattery) + (0.02 * socVoltage);
811       }
812     }
813     else if (dV <= DV_DISCHARGE_THRESHOLD) {
814       // Tegangan turun cukup jelas, dianggap discharging
815       socBattery -= deltaSoc;
816
817       // Koreksi pelan jika tegangan menunjukkan SOC lebih rendah
818       if (socBattery > socVoltage) {
819         socBattery = (0.98 * socBattery) + (0.02 * socVoltage);
820       }
821     }
822     else {
823       // Tegangan stabil, SOC tidak banyak berubah
824       socBattery = (0.998 * socBattery) + (0.002 * socVoltage);
825     }
826
827     // ========================
828     // 6. BATAS SOC
829     // ========================
830     if (socBattery > 100.0) socBattery = 100.0;
831     if (socBattery < 0.0) socBattery = 0.0;
832   }
833
834   void setup() {
835     Serial.begin(115200);
836     delay(1000);
837
838     setupFuzzy();
839
840     Serial.print("Menghubungkan ke WiFi: ");
841     Serial.println(WIFI_SSID);
842     WiFi.mode(WIFI_STA);
843     WiFi.setSleep(false);
844     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
845     while (WiFi.status() != WL_CONNECTED) {
846       Serial.print(".");
847       delay(500);
848     }
849     Serial.println("\nWiFi Terhubung.");
850     Serial.print("IP Address: ");
851     Serial.println(WiFi.localIP());
852
853     // ========================
854     // ESP-NOW RECEIVER
```

INSTITUT TEKNOLOGI KALIMANTAN

```
855   // ========================
856   if (esp_now_init() == ESP_OK) {
857     Serial.println("ESP-NOW siap menerima tegangan PV dari ESP teman.");
858     esp_now_register_recv_cb(OnDataRecv);
859   } else {
860     Serial.println("Gagal inisialisasi ESP-NOW.");
861   }
862
863   configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
864   Serial.print("Sinkronisasi waktu WITA (Balikpapan)");
865
866   struct tm timeinfo;
867   while (!getLocalTime(&timeinfo)) {
868     Serial.print(".");
869     delay(500);
870   }
871   Serial.println("\nWaktu berhasil disinkronisasi!");
872
873   config.api_key = API_KEY;
874   config.database_url = DATABASE_URL;
875
876   if (Firebase.signUp(&config, &auth, "", "")) {
877     Serial.println("Firebase Auth Berhasil");
878     signupOK = true;
879   } else {
880     Serial.printf("%s\n", config.signer.signupError.message.c_str());
881   }
882
883   config.token_status_callback = tokenStatusCallback;
884   Firebase.begin(&config, &auth);
885   Firebase.reconnectWiFi(true);
886
887   pinMode(RELAY1, OUTPUT);
888   pinMode(RELAY2, OUTPUT);
889   pinMode(RELAY3, OUTPUT);
890   pinMode(RELAY4, OUTPUT);
891
892   // ========================
893   // DELAY AMAN SEBELUM RELAY AKTIF
894   // ========================
895   Serial.println("Menunggu 5 detik sebelum mengaktifkan Relay...");
896   delay(5000);
897
898   relayAllOn();
899   Serial.println("Relay diaktifkan.");
900
```

INSTITUT TEKNOLOGI KALIMANTAN

```
  pinMode(MAX485_RE_DE, OUTPUT);
  digitalWrite(MAX485_RE_DE, LOW);

  mySerial.begin(9600, SERIAL_8N2, RXD2, TXD2);

  pzemPV.begin(1, mySerial);
  pzemBat.begin(2, mySerial);

  pzemPV.preTransmission(preTransmission);
  pzemPV.postTransmission(postTransmission);

  pzemBat.preTransmission(preTransmission);
  pzemBat.postTransmission(postTransmission);

  serialPZEM004T.begin(9600, SERIAL_8N1, PZEM004T_RX, PZEM004T_TX);

  sensors.begin();

  Serial.println("=== 2 PZEM-017 + PZEM-004T + RELAY + DS18B20 + SOC + FUZZY + KONTROL RELAY ===");

  // Agar tidak harus menunggu 5 menit sejak awal hidup untuk update data pertama
  previousMillisData = millis() - intervalData;
}

void loop() {
  unsigned long currentMillis = millis();

  // ==========================================
  // BLOK 1: LAPORAN STATUS (HEARTBEAT) SETIAP 10 DETIK
  // ==========================================
  // Heartbeat tetap pakai millis() karena tidak perlu sinkron dengan jam dinding
  if (currentMillis - previousMillisStatus >= intervalStatus) {
    previousMillisStatus = currentMillis;

    // Selalu kirim detak jantung (24 jam)
    if (Firebase.ready() && signupOK) {
      Firebase.RTDB.setInt(&fbdo, "/status/last_beat", millis());
    }
  }

  // ==========================================
  // BLOK 2: PENGAMBILAN DATA SENSOR (SYNCHRONIZED 5 MENIT - 24 JAM)
  // ==========================================
  // HAPUS pengecekan millis() di sini, langsung pantau jam NTP setiap saat!
```

INSTITUT TEKNOLOGI KALIMANTAN

```
945
946   struct tm timeinfo;
947   if(getLocalTime(&timeinfo)){
948
949     int jamSekarang = timeinfo.tm_hour;
950     int menitSekarang = timeinfo.tm_min;
951     int detikSekarang = timeinfo.tm_sec;
952
953     // LOGIKA SINKRONISASI 5 MENIT TEPAT WAKTU!
954     // Cek apakah menit saat ini adalah kelipatan 5 (0, 5, 10, 15... 55)
955     // DAN apakah detik saat ini adalah 0 (tepat di awal menit)
956     static int lastDataMinute = -1;
957
958     if ((menitSekarang % 5 == 0) && (detikSekarang == 0) && (menitSekarang != lastDataMinute)) {
959
960       lastDataMinute = menitSekarang; // Kunci agar tidak mengambil data ganda di detik yang sama
961
962       Serial.printf("\n=======================================\n");
963       Serial.printf("Pukul %02d:%02d:%02d WITA - Waktunya Mengambil Data!\n", jamSekarang, menitSekarang, detikSekarang);
964
965       float vPV = 0, iPV = 0, pPV = 0, ePV = 0;
966       float vBat = 0, iBat = 0, pBat = 0, eBat = 0;
967
968       // PZEM PV hanya membaca arus.
969       // Tegangan PV diambil dari ESP teman melalui ESP-NOW.
970       bool okPV  = readPZEMCurrentOnly(pzemPV, iPV);
971       bool okTeganganPVESPNow = teganganPVSiap && ((millis() - lastTerimaTeganganPV) <= BATAS_VALID_TEGANGAN_PV);
972
973       if (okTeganganPVESPNow) {
974         vPV = teganganPVKalibrasi;
975         statusTeganganPVESPNow = "Tegangan PV valid dari ESP-NOW";
976       } else {
977         vPV = 0;
978         statusTeganganPVESPNow = "Tegangan PV belum diterima / data sudah terlalu lama";
979       }
980
981     if (okPV && okTeganganPVESPNow) {
982       pPV = vPV * iPV;
983     } else {
984       pPV = 0;
985     }
986
987       delay(300);
```

INSTITUT TEKNOLOGI KALIMANTAN

```
988         bool okBat = readPZEM(pzemBat, vBat, iBat, pBat, eBat);
989
990         digitalWrite(MAX485_RE_DE, LOW);
991         delay(100);
992
993         float acV  = pzemAC.voltage();
994         float acI  = pzemAC.current();
995         float acP  = pzemAC.power();
996         float acE  = pzemAC.energy();
997         float acF  = pzemAC.frequency();
998         float acPF = pzemAC.pf();
999
1000        sensors.requestTemperatures();
1001        delay(50);
1002        float tempC = sensors.getTempCByIndex(0);
1003
1004        // ==========================
1005        // UPDATE SOC BARU
1006        // ==========================
1007        // Sekarang SOC memakai vBat dan iBat.
1008        // vBat dipakai untuk menentukan arah charging/discharging.
1009        // iBat dipakai untuk besar perubahan coulomb counting.
1010        if (okBat) {
1011          updateBatterySOC(vBat, iBat);
1012        }
1013
1014        // ================= MENCETAK KE SERIAL MONITOR =================
1015        if (okPV && okTeganganPVESPNow) {
1016          Serial.println("PV");
1017          Serial.println("Voltage: dari ESP teman via ESP-NOW");
1018          Serial.printf("Voltage: %.2f V | Current: %.2f A | Power: %.2f
W\n", vPV, iPV, pPV);
1019        } else if (okPV && !okTeganganPVESPNow) {
1020          Serial.println("PV");
1021          Serial.printf("Current PZEM PV: %.2f A\n", iPV);
1022          Serial.println("Tegangan PV dari ESP teman belum valid, daya PV
belum dihitung.");
1023        } else {
1024          Serial.println("Gagal baca arus PZEM PV");
1025        }
1026
1027        if (okBat) {
1028          Serial.println("PZEM BATERAI");
1029          Serial.printf("Voltage: %.2f V | Current: %.2f A | Power: %.2f W |
SOC: %.2f %%\n", vBat, iBat, pBat, socBattery);
1030        } else {
1031          Serial.println("Gagal baca PZEM BATERAI");
```

```
1032        }
1033
1034        Serial.println("PZEM-004T (AC BEBAN)");
1035        if (isnan(acV)) {
1036          Serial.println("Tidak terbaca");
1037        } else {
1038          Serial.printf("Voltage: %.2f V | Current: %.2f A | Power: %.2f W | PF: %.2f\n", acV, acI, acP, acPF);
1039        }
1040
1041        Serial.println("DS18B20");
1042        if (tempC == DEVICE_DISCONNECTED_C || tempC == -127.0) {
1043          Serial.println("Sensor tidak terbaca");
1044        } else {
1045          Serial.printf("Suhu : %.2f C\n", tempC);
1046        }
1047
1048        // ================== HASIL FUZZY / PROTEKSI SUHU DARI DATA SENSOR ==================
1049        bool sensorSuhuOK = !(tempC == DEVICE_DISCONNECTED_C || tempC == -127.0);
1050
1051        if (sensorSuhuOK && tempC >= BATTERY_TEMP_HIGH) {
1052          prosesProteksiSuhu(tempC);
1053        }
1054        else if (okPV && okTeganganPVESPNow && okBat && !isnan(acP)) {
1055          prosesFuzzySensor(pPV, socBattery, acP);
1056        }
1057        else {
1058          Serial.println("HASIL FUZZY");
1059          Serial.println("Fuzzy tidak dijalankan karena data sensor utama belum lengkap.");
1060
1061          fuzzyModeValue = 0.0;
1062          fuzzyModeLabel = "Data Tidak Lengkap";
1063          fuzzyLevelValue = -1.0;
1064          fuzzyLevelLabel = "Data Tidak Lengkap";
1065
1066          ruleAktifText = "Fuzzy tidak dijalankan karena data sensor utama belum lengkap";
1067          ruleAktifDominan = 0;
1068          alphaRuleAktifDominan = 0.0;
1069        }
1070        // ================== AKHIR HASIL FUZZY / PROTEKSI SUHU ==================
1071
1072        printRelayStatus();
1073        Serial.println("========================================");
```

INSTITUT TEKNOLOGI KATIMANTAN

```
      // ================== AKHIR MENCETAK KE SERIAL MONITOR
==================

      if (Firebase.ready() && signupOK) {
        FirebaseJson json;

        json.set("pv_tegangan", okTeganganPVESPNow ? vPV : 0);
        json.set("pv_arus", okPV ? iPV : 0);
        json.set("pv_daya", (okPV && okTeganganPVESPNow) ? pPV : 0);
        json.set("pv_sumber_tegangan", "ESP teman via ESP-NOW");
        json.set("pv_status_tegangan", statusTeganganPVESPNow);

        json.set("bat_tegangan", okBat ? vBat : 0);
        json.set("bat_arus", okBat ? iBat : 0);
        json.set("bat_daya", okBat ? pBat : 0);
        json.set("bat_soc", socBattery);

        json.set("suhu_baterai", (tempC == DEVICE_DISCONNECTED_C || tempC
== -127.0) ? 0 : tempC);
        json.set("proteksi_suhu_aktif", proteksiSuhuAktif);
        json.set("status_proteksi_suhu", statusProteksiSuhu);

        json.set("beban_tegangan", isnan(acV) ? 0 : acV);
        json.set("beban_arus", isnan(acI) ? 0 : acI);
        json.set("beban_daya", isnan(acP) ? 0 : acP);
        json.set("beban_pf", isnan(acPF) ? 0 : acPF);

        // Tambahan data mode operasi dan fuzzy ke Firebase
        // fuzzy_mode_value berisi Pnet / selisih daya
        json.set("fuzzy_mode_value", fuzzyModeValue);
        json.set("fuzzy_mode_label", fuzzyModeLabel);
        json.set("fuzzy_level_value", fuzzyLevelValue);
        json.set("fuzzy_level_label", fuzzyLevelLabel);

        // Tambahan data rule aktif ke Firebase
        json.set("rule_aktif", ruleAktifText);
        json.set("rule_aktif_dominan", ruleAktifDominan);
        json.set("alpha_rule_dominan", alphaRuleAktifDominan);

        // Tambahan status relay ke Firebase
        json.set("relay1_status", digitalRead(RELAY1) == LOW ? "ON" :
"OFF");
        json.set("relay2_status", digitalRead(RELAY2) == LOW ? "ON" :
"OFF");
        json.set("relay3_status", digitalRead(RELAY3) == LOW ? "ON" :
"OFF");
        json.set("relay4_status", digitalRead(RELAY4) == LOW ? "ON" :
"OFF");
```

INSTITUT TEKNOLOGI KALIMANTAN

```
1116
1117          if (Firebase.RTDB.setJSON(&fbdo, "/monitoring", &json)) {
1118            Serial.println(">> Sukses Upload Data ke Firebase!");
1119          } else {
1120             Serial.println(">> Gagal Upload ke Firebase!");
1121          }
1122        }
1123      }
1124    }
1125
1126    delay(10);
1127  }
```

INSTITUT TEKNOLOGI
KALIMANTAN