

LAMPIRAN

LAMPIRAN A HASIL PENGUKURAN

Tabel 1 Hasil Pengujian Sensor Tegangan

Hasil Pengukuran Sensor Tegangan		
Voltage Divider (V)	Multimeter (V)	%Error
3.01	3	0.3%
4.02	4	0.5%
5.01	5	0.2%
6.02	6	0.3%
7.05	7	0.7%
8.12	8	1.5%
9.14	9	1.5%
10.14	10	1.4%
11.2	11	1.8%
12.21	12	1.75%
Rata-rata		0.995%

Tabel 2 Hasil Pengujian Sensor Arus

Hasil Pengukuran Sensor Arus		
ACS712 (A)	Multimeter (A)	%Error
1.01	1	1%
2.07	2	3.5%
3.1	3	3%
4.12	4	3%
5.15	5	3%
6.18	6	3%
7.2	7	2.8%
8.21	8	2.8%
9.21	9	2.3%
10.23	10	2.3%
Rata-rata		2.67%

Tabel 3 Hasil Pengujian Sensor Suhu

Hasil Pengukuran Sensor Arus		
DS18B20 (°C)	Termometer (°C)	%Error
4	4	0%
5	5	0%
6	6	0%
7.08	7	1%
8.1	8	1.25%
49.19	49	0.38%
50.44	50	0.83%
51.18	51	0.35%
52.13	52	0.25%
55.25	55	0.45%
Rata-rata		0.456%

Tabel 4 Hasil Pencatatan Data Kondisi Cuaca Cerah

Waktu	Irradiance (W/m ²)	Tegangan Panel (V)	Arus Panel (A)	Suhu Panel (°C)	Tegangan Baterai (V)	Arus Charging (A)	SoC (%)	Daya Output Panel (Watt)
6:00:24 AM	27	12.145	0.279	24.9	12.8	0.1	69	3.39
7:00:14 AM	60	12.543	0.685	26.2	12.8	0.2	70	8.59
8:00:44 AM	175	13.217	0.847	26.5	12.8	0.2	70	11.19
9:00:04 AM	295	13.624	1.543	27.3	12.9	0.3	72	21.02
10:00:14 AM	320	15.392	3.29	34	13	0.5	75	50.64
11:00:00 AM	623.8	16.009	6.237	38.8	13.2	0.8	82	99.85
12:00:10 PM	843.6	25.062	8.149	45.5	13.5	1.2	93	204.23
1:00:05 PM	760	31.869	2.502	40.8	13.8	0.4	100	79.74
2:00:10 PM	655.1	34.662	1.823	37.5	13.8	0.2	100	63.19
3:00:10 PM	215.4	18.362	1.2	37.1	13.8	0.2	100	22.03
4:00:15 PM	101.6	14.039	1.098	30.1	13.8	0.2	100	15.41
5:00:15 PM	44.6	1.514	0.17	30	13.8	0.1	100	0.26

Tabel 5 Hasil Pencatatan Data Kondisi Cuaca Berawan

Waktu	Irradiance (W/m ²)	Tegangan Panel (V)	Arus Panel (A)	Suhu Panel (°C)	Tegangan Baterai (V)	Arus Charging (A)	SoC (%)	Daya Output Panel (Watt)
6:00:01 AM	28	12.097	0.288	26.4	12.8	0.1	69	3.48
7:00:02 AM	126.1	12.149	1.307	27.4	12.8	0.2	69	15.88
8:00:02 AM	217.2	12.933	2.418	28.8	12.9	0.4	73	31.27
9:00:02 AM	225.9	13.373	3.013	32	13.1	0.5	78	40.29
10:00:17 AM	626.1	14.087	6.399	34.3	13.2	1	83	90.14
11:00:02 AM	375	15.272	3.584	35.2	13.4	0.7	90	54.73
12:00:02 PM	551.3	25.03	2.894	43.5	13.5	0.5	95	72.44
1:00:03 PM	376.1	30.859	1.615	43	13.8	0.3	99	49.84
2:00:27 PM	375	32.659	1.203	35.6	13.8	0.2	100	39.29
3:00:07 PM	139.5	34.875	0.954	33.5	13.8	0.2	100	33.27
4:00:07 PM	161.8	23.052	0.923	31.5	13.8	0.2	100	21.28
5:00:07 PM	62.5	13.937	0.62	29.5	13.8	0.2	100	8.64

Tabel 6 Hasil Pencatatan Data Kondisi Cuaca Hujan

Waktu	Irradiance (W/m ²)	Tegangan Panel (V)	Arus Panel (A)	Suhu Panel (°C)	Tegangan Baterai (V)	Arus Charging (A)	SoC (%)	Daya Output Panel (Watt)
6:00:24 AM	0	0.835	0.157	26.3	12.8	0.1	69	0.13
7:00:24 AM	40.2	1.002	1.146	28.7	12.8	0.1	69	1.15
8:00:24 AM	225.4	12.277	1.233	31.6	12.8	0.2	70	15.14
9:00:24 AM	346	13.593	2.805	34.3	12.9	0.5	71	38.13
10:00:24 AM	280.1	13.651	4.159	33.3	13.2	0.7	74	56.77
11:00:24 AM	240	13.747	4.488	34.7	13.3	0.8	77	61.70
12:00:24 PM	280.1	24.746	3.965	35.3	13.4	0.7	83	98.12
1:00:24 PM	328.1	25.842	2.59	35.3	13.5	0.5	88	66.93
2:00:24 PM	294.6	26.269	1.267	36.1	13.8	0.2	89	33.28
3:00:24 PM	119.4	27.242	0.98	32.4	13.8	0.1	94	26.70
4:00:24 PM	102.7	31.29	0.825	30.4	13.8	0.1	98	25.81
5:00:24 PM	19	13.456	0.784	29.3	13.8	0.1	100	10.55

LAMPIRAN B *LISTING* PROGRAM

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include "FS.h"
#include "SD.h"
#include "SPI.h"
#include "RTClib.h"

// Deklarasi Thingier.io //
#include <ThingierESP32.h>
#define THINGER_SERIAL_DEBUG
#define USERNAME "arisuryak"
#define DEVICE_ID "SKRIPSI"
#define DEVICE_CREDENTIAL "42&ka41#WM$AUHNu"

#define SSID "UNTIRTAKU"
#define SSID_PASSWORD "untirtajawara"
ThingierESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

//=====//

// Pembacaan RTC //
RTC_DS1307 rtc;
int tanggal, bulan, tahun, jam, menit, detik;

//=====//

// Pembacaan Voltage Divider //
int Vpin = 25; // pin ke esp32
```

```

float R1 = 27000.0; //27k ohm
float R2 = 10000.0; //10K ohm
float Vmodul = 0.0;
float hasil = 0.0;
int value = 0;
float volt0 = 0;

// ===== //

// Pembacaan Sensor ACS712 //
int a1 = 26; // Sensor Arus
float Vmid = 2.4557; // Titik tengah ACS712
float arus0;
float arus1;
// ===== //

// Pembacaan Sensor Suhu DS18B20 //
#define SENSOR_PIN 4
OneWire oneWire(SENSOR_PIN);
DallasTemperature DS18B20(&oneWire);
float tempC;

//=====//

// Pembacaan SD CARD //
#define SD_CS 2
String dataMessage;

// ===== //

// Pembacaan LCD 16x2 //

```

```

LiquidCrystal_I2C lcd(0x27, 20, 4);
// ===== //

// Constrain SOC //
const int a = 0; //batas bawah persentase SoC
const int b = 100; //batas atas persentase SoC
// ===== //

// DEFINISI SOC //
float SOC0 = 9.5;
float SOC100 = 12.7;
float getSOC = 0.0;
float deltaV = 0.0;
int countSOC = 0;
int samples = 0;
// ===== //

// Pembacaan LCD //
float Iradiasi;
// ===== //

int halaman = 1;
int NUMBER_OF_SAMPLE = 150;

void setup() {

  Serial.begin(115200);
  WiFi.begin(SSID, SSID_PASSWORD);
  thing.add_wifi(SSID, SSID_PASSWORD);

// Memulai LCD 16x2 //
  lcd.init();

```

```

lcd.backlight();
// ===== //

DS18B20.begin();

// Pengirima data ke THINGER.IO //
thing["DATA"] >> [] (pson& out) {
  out["Tegangan"] = hasil;
  out["Arus"] = arus1;
  out["Suhu"] = tempC;
  out["Pyranometer"] = Iradiasi;

};
// ===== //

// Memulai RTC //
if (! rtc.begin())
{
  Serial.println("RTC Tidak Ditemukan");
  Serial.flush();
  abort();
}
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
//=====//

// Memulai SD CARD //
SD.begin(SD_CS);
if(!SD.begin(SD_CS)) {
  Serial.println("Card Mount Failed");
  return;
}
uint8_t cardType = SD.cardType();
if(cardType == CARD_NONE) {
  Serial.println("No SD card attached");
  return
}

```



```

}
Serial.println("Initializing SD card...");
if (!SD.begin(SD_CS)) {
  Serial.println("ERROR - SD card initialization failed!");
  return;    // init failed
}
File file = SD.open("/data.txt");
if(!file) {
  Serial.println("File doesn't exist");
  Serial.println("Creating file...");
  writeFile(SD, "/data.txt", "ESP32 and SD Card \r\n");
}
else {
  Serial.println("File already exists");
}
file.close();
}

void loop(){
  thing.handle();

  // RTC //
  DateTime now = rtc.now();
  tanggal = now.day(), DEC;
  bulan    = now.month(), DEC;
  tahun    = now.year(), DEC;
  jam      = now.hour(), DEC;
  menit    = now.minute(), DEC;
  detik    = now.second(), DEC;
  Serial.println(String() + tanggal + "-" + bulan + "-" + tahun);
  Serial.println(String() + jam + ":" + menit + ":" + detik);

  // voltage divider //
  value = analogRead(Vpin);
  Vmodul= (value*3.45)/4095.0;
  float hasil = Vmodul/(R2/(R1+R2));

```

```

// sensor arus //
a1 = analogRead(26);
arus0 = a1 * 3.3/4095.0;
arus1 = abs((arus1-(Vmid))/0.1);

// Sensor Suhu DS18B20//
DS18B20.requestTemperatures();
tempC = DS18B20.getTempCByIndex(0);

delay(200);

// Tampilan LCD 20x4 //
{
  lcd.setCursor(0,0);
  lcd.print(String() + tanggal + "-" + bulan + "-" + tahun);
  lcd.setCursor(12,0);
  lcd.print(String() + jam + ":" + menit + ":" + detik);
  lcd.setCursor(0,1);
  lcd.print("V:");
  lcd.print(hasil);
  lcd.print("V");
  lcd.setCursor(12,1);
  lcd.print("T:");
  lcd.print(tempC);
  lcd.print("C");
  lcd.setCursor(0,2);
  lcd.print("I:");
  lcd.print(arus1);
  lcd.print("A");
  lcd.setCursor(0,3);
  lcd.print("Rad:");
  lcd.print("W/m2")
}

```

```

}

//SOC Baterai
deltaV = (SOC100 - SOC0) / 100; // *SOC100 = 12,7volt *SOC0 = 9.5volt
countSOC = (volt0 - SOC0) / deltaV; // hasil ex. pin (+) battery
//avg
countSOC = countSOC - 0.2 + random(0, 400) * 0.001;
//avg
samples++;
Serial.println (countSOC);

// ===== //
// Penyimpanan SD CARD //
logSDCard();
delay(5000); //Wait for 5 seconds before writing the next data
}

// Pencatatan data sensor //
void logSDCard() {
  dataMessage = "Variable Suhu = " + String(tempC) + "\n";
  Serial.print("Save data: ");
  Serial.println(dataMessage);
  appendFile(SD, "/data.txt", dataMessage.c_str());
}

// Pencatatan pada SD CARD //
void writeFile(fs::FS &fs, const char * path, const char * message) {
  Serial.printf("Writing file: %s\n", path);
  File file = fs.open(path, FILE_WRITE);
  if(!file) {
    Serial.println("Failed to open file for writing");
    return;
  }
  if(file.print(message)) {
    Serial.println("File written");
  } else {

```

```

    Serial.println("Write failed");
}
file.close();
}

// Append data ke SD CARD
void appendFile(fs::FS &fs, const char * path, const char * message) {
    Serial.printf("Appending to file: %s\n", path);
    File file = fs.open(path, FILE_APPEND);
    if(!file) {
        Serial.println("Failed to open file for appending");
        return;
    }
    if(file.print(message)) {
        Serial.println("Message appended");
    } else {
        Serial.println("Append failed");
    }
    file.close();
}

void ReadVresistor()
{
    tempC = analogRead(SENSOR_PIN);
    Serial.println(tempC)
}

```