

**IMPLEMENTASI *INTERNET OF THINGS* PADA TRAINER
PEMBANGKIT LISTRIK TENAGA SURYA MULTI PANEL**

TUGAS AKHIR



Oleh :

OLAENI ERTINA

NIT. 30118018

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2021

**IMPLEMENTASI *INTERNET OF THINGS* PADA TRAINER
PEMBANGKIT LISTRIK TENAGA SURYA MULTI PANEL**

TUGAS AKHIR

Diajukan sebagai Syarat Menempuh Tugas Akhir

Pada Program Studi Diploma 3 Teknik Listrik Bandar Udara



Oleh :

OLAENI ERTINA

NIT. 30118018

**PROGRAM STUDI DIPLOMA 3 TEKNIK LISTRIK BANDARA
POLITEKNIK PENERBANGAN SURABAYA**

2021

LEMBAR PERSETUJUAN

**IMPLEMENTASI *INTERNET OF THINGS* PADA TRAINER
PEMBANGKIT LISTRIK TENAGA SURYA MULTI PANEL**

Oleh :

Olaeni Ertina

NIT. 30118018

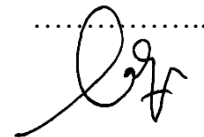
Disetujui untuk diujikan pada :

Surabaya, 03 Agustus 2021

Pembimbing I : FIQQIH FAIZAH., S.T., M.T.
NIP. 19850709 200912 2 005



Pembimbing II : LADY SILK M., S.Kom., M.T.
NIP. 19871109 200912 2 002



LEMBAR PENGESAHAN

IMPLEMENTASI *INTERNET OF THINGS* PADA TRAINER PEMBANGKIT LISTRIK TENAGA SURYA MULTI PANEL

Oleh :

Olaeni Ertina

NIT. 30118018

Telah dipertahankan dan dinyatakan lulus pada Ujian Tugas Akhir Program

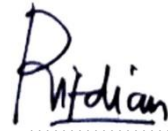
Pendidikan Diploma 3 Teknik Listrik Bandara

Politeknik Penerbangan Surabaya

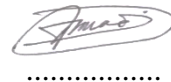
Pada tanggal : Surabaya, 03 Agustus 2021

Panitia Penguji :

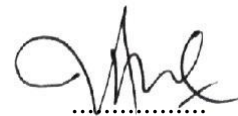
1. Ketua : RIFDIAN I. S., S.T., M.M., M.T.
NIP. 19810629 200912 1 002



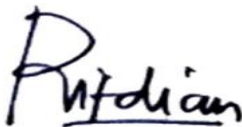
2. Sekretaris : DARMADJI., S.T.
Kode dosen : 1956060241001



3. Anggota : FIQQIH FAIZAH., S.T., M.T.
NIP. 19850709 200912 2 005



Ketua Program Studi
D3 Teknik Listrik Bandara



RIFDIAN I. S., S.T., M.M., M.T.
NIP. 19810629 200912 1 002

PERNYATAAN KEASLIAN DAN HAK CIPTA

Saya yang bertanda tangan dibawah ini :

Nama : Olaeni Ertina
NIT : 30118018
Program Studi : D-III Teknik Listrik Bandara
Judul Tugas Akhir : Implementasi Internet Of Things Pada Trainer
Pembangkit Listrik Tenaga Surya Multipanel

dengan ini menyatakan bahwa :

1. Tugas Akhir ini merupakan karya asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Politeknik Penerbangan Surabaya maupun di Perguruan Tinggi lain, serta dipublikasikan, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
2. Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan Hak Bebas Royalti Non Eksklusif (*Non-Exclusive Royalty-Free Right*) kepada Politeknik Penerbangan Surabaya beserta perangkat yang ada (jika diperlukan), dengan hak ini, Politeknik Penerbangan Surabaya berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya dengan tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya. Apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Politeknik Penerbangan Surabaya.

Surabaya, 20 Juli 2021

Yang membuat pernyataan



Olaeni Ertina

NIT.30118018

ABSTRAK

IMPLEMENTASI *INTERNET OF THINGS* PADA *TRAINER* PEMBANGKIT LISTRIK TENAGA SURYA MULTIPANEL

Oleh :

Olaeni Ertina

NIT : 30118018

Trainer ialah suatu bentuk media pembelajaran, yang bisa digunakan sebagai media praktik, media sebagai pembelajaran berfungsi sebagai alat bantu untuk menyampaikan informasi kepada praktikan, alat bantu yang dimaksud penulis adalah alat bantu pembelajaran peraktik atau alat peraga peraktik

Perancangan *trainer* pembangkit listrik tenaga surya ini bertujuan untuk mempermudah sistem pembelajaran dan pada sistem pembangkit listrik tenaga surya yang dirancang dilengkapi dengan sensor arus dan tegangan INA219 , Arduino mega dan Wemos D1 mini yang dihubungkan dengan aplikasi *android*.

Pada program arduino bisa ditampilkan keluaran arus dan tegangan panel surya 1, keluaran arus dan tegangan panel surya 2, keluaran arus dan tegangan baterai, dan keluaran arus dan tegangan beban.

Kata Kunci : *Trainer*, Panel Surya, mikrokontroler Arduino Mega, sensor arus dan tegangan.

ABSTRACT

IMPLEMENTATION OF INTERNET OF THINGS ON MULTIPANEL SOLAR POWER PLANT TRAINER

By :

Olaeni Ertina

NIT : 30118018

Trainer is a form of learning media, which can be used as a practical medium, media as learning serves as a tool to convey information to practitioners, the tools referred to by the author are practical learning aids or practical teaching aids

The design of this solar power plant trainer aims to facilitate the learning system and the solar power generation system which is designed to be equipped with current and voltage sensors INA219, Arduino mega and Wemos D1 mini which is connected to the android application. The Arduino program can display the current and voltage output of the solar panel 1, the output current and voltage of the solar panel 2, the output current and voltage of the battery, and the output current and voltage of the load.

Keywords: Trainer, Solar Panel, Arduino Mega microcontroller, current and voltage sensor.

KATAPENGANTAR

Puji syukur kehadirat ALLAH SWT atas segala rahmat dan karunia – Nya yang telah memberikan kesehatan, pengetahuan, keterampilan, pengalaman yang senantiasa diberikan kepada penulis, sehingga penulis bisa menyelesaikan Laporan Tugas Akhir yang berjudul **“IMPLEMENTASI INTRNET OF THINGS PADA TRAINER PEMBANGKIT LISTRIK TENAGA SURYA MULTI PANEL”** dengan baik dan lancar sesuai dengan waktu yang ditetapkan dan sebagai syarat untuk menyelesaikan program Diploma 3 Teknik Listrik Bandara di Politeknik Penerbangan Surabaya.

Selama proses penyusunan tugas khusus ini penulis banyak menerima bantuan, bimbingan dan pengarahan dari berbagai pihak baik material spiritual, materi serta saran. Pada kesempatan ini mengucapkan terima kasih kepada:

1. Orang tua yang selalu mendukung saya dan memberi semangat, menjadi motivasi yang membuat semangat tanpa batas dalam berusaha dan bekerja.
2. Bapak M. Andra Adityawarman., S.T., M.T. selaku Direktur Politeknik Penerbangan Surabaya.
3. Bapak Rifdian I. S., S.T., M.M., M.T. selaku Ketua Program Studi Teknik Listrik Bandara Politeknik Penerbangan Surabaya.
4. Ibu Fiqqih Faizah., S.T., M.T. selaku Dosen Pembimbing Materi Tugas Akhir.
5. Ibu Lady Silk Moonlight., S.kom., M.T. selaku Pembimbing Materi penulisan Tugas Akhir.
6. Dosen Politeknik Penerbangan Surabaya yang telah memberikan pengetahuan dan memberikan pelajaran yang berharga untuk penulis serta teman–teman Teknik Listrik Bandara angkatan XIII yang telah memberikan banyak bantuan, support dan motivasi.

Penulis menyadari bahwa penulisan ini masih jauh dari sempurna dan masih banyak kekurangan, maka dari itu penulis mengharapkan saran dan kritik yang

dapat membangun dari semua pihak agar dapat membantu untuk menjadikan penulisan Tugas Akhir selanjutnya lebih baik. Akhir kata penulis berharap semoga penulisan ini dapat bermanfaat dan selanjutnya dapat dikembangkan, berguna bagi semua pihak.

Surabaya, 30 Juli 2021

Penulis

MOTTO

VISI TANPA EKSEKUSI ADALAH HALUSINASI

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN KEASLIAN DAN HAK CIPTA.....	iii
KATAPENGANTAR	vi
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	2
1.3 Batasan masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	4
BAB 2.....	5
LANDASAN TEORI.....	5
2.1 Teori Penunjang	5
2.1.1 Definisi Trainer	5
2.1.2 Panel Surya.....	6
2.1.2.1 Cara Kerja Panel Surya.....	6
2.1.2.3 Jenis-jenis Panel Surya	8
2.1.3 Baterai.....	11
2.1.4 DC to DC Converter.....	13
2.1.4.1 Komponen DC to DC Converter	13
2.1.4.2 Rangkaian Regulator Tegangan DC	16
2.1.4.3 Pulse Width Modulation (PWM).....	17
2.1.5 Maximum Power Point Tracker (MPPT).....	18
2.1.6 Arduino.....	19
2.1.7 IDE Arduino	20
2.1.8 Sensor Arus INA219	21
2.1.9 Sensor Tegangan	22
2.1.10 Internet of Things (IoT).....	23

2.1.10.1 Teknologi Internet of Things (IoT).....	24
2.1.10.2 Prinsip Kerja Internet of Things (IoT)	24
2.1.11 WEMOS MINI D1	24
2.2 Kajian Penelitian yang Relevan.....	26
BAB 3.....	27
METODE PENELITIAN.....	27
3.1 Desain Penelitian.....	27
3.2 Perancangan Alat.....	28
3.2.2 Cara Kerja Alat	31
3.2.3 Komponen Alat.....	32
3.2.3.2 Perangkat Lunak.....	35
3.3 Teknik Pengujian.....	38
3.4 Teknik Analisis Data	41
3.5 Tempat dan Waktu Penelitian	41
BAB 4 HASIL DAN PEMBAHASAN.....	43
4.1 Hasil Penelitian.....	43
BAB 5.....	58
PENUTUP.....	58
5.1 Kesimpulan.....	58
5.2 Saran.....	58

DAFTAR GAMBAR

<i>Gambar 2. 1 Junction antara semikonduktor tipe-p (kelebihan hole) dan tipe-n (kelebihan electron)</i>	<i>20</i>
Gambar 2. 2 Aliran elektron pada pembangkitan listrik panel surya.....	21
Gambar 2. 3 Karakteristik P-V panel surya	21
Gambar 2. 4 Panel surya jenis <i>monocrystalline</i>	22
Gambar 2. 5 Panel surya jenis <i>polycrystalline</i>	23
Gambar 2. 6 Panel surya jenis <i>Gallium Arsenide</i>	23
Gambar 2. 7 Panel surya jenis <i>amorphous silicon</i>	24
Gambar 2. 8 Bentuk fisik dari baterai 12V	24
Gambar 2. 9 <i>Dc Chopper</i> dengan beban resistif	26
Gambar 2. 10 Resistor.....	26
Gambar 2. 11 Induktor	27
Gambar 2. 12 Induktor Toroid	27
Gambar 2. 13 Kapasitor	28
Gambar 2. 14 Simbol Mosfet	28
Gambar 2. 15 Bentuk fisik IC regulator 7805.....	29
Gambar 2. 16 Ilustrasi modulasi lebar pulsa.....	30
Gambar 2. 17 Spesifikasi Arduino	32
Gambar 2. 18 IDE Arduino.....	33
Gambar 2. 19 Bentuk fisik sensor ACS712	34
Gambar 2. 20 Rangkaian panel surya	35
Gambar 2. 21 WEMOS MINI D1	37
Gambar 3. 1 Diagram Blok	39
Gambar 3. 2 <i>Wiring</i> Perancangan Alat	40
Gambar 3. 3 Flowchart sistem	41
Gambar 4. 1 Gambar Alat Keseluruhan.....	43
Gambar 4. 2 Panel Surya.....	44
Gambar 4. 3 SCC	44
Gambar 4. 4 Baterai	45
Gambar 4. 5 Inverter	46

Gambar 4. 6 Sensor Arus dan Tegangan.....	46
Gambar 4. 7 Mikrokontroler Arduino	47
Gambar 4. 8 Wemos MINI D1	47
Gambar 4. 9 Tampilan pada Desktop.....	48
Gambar 4. 10 Tampilan Saat Masuk ke <i>Software</i>	48
Gambar 4. 11 Pengujian Panel Surya.....	50
Gambar 4. 12 Pengujian SCC	50
Gambar 4. 13 Pengujian Baterai	51
Gambar 4. 14 Pengujian Inverter	52
Gambar 4. 15 Tampilan <i>Coding</i> Arduino	52
Gambar 4. 16 Mikrokontroler Arduino.....	53
Gambar 4. 17 Wemos D1 MINI.....	54
Gambar 4. 18 Grafik Arus.....	56
Gambar 4. 19 Tampilan Saat diuji	56

DAFTAR TABEL

Tabel 2. 1 Proses Perubahan Daya	19
Tabel 2. 2 Spesifikasi Arduino	19
Tabel 3. 1 Cara Pengujian Panel Surya	38
Tabel 3. 2 Cara Pengujian SCC.....	38
Tabel 3. 3 Cara Pengujian Baterai.....	39
Tabel 3. 4 Cara Pengujian Inverter	39
Tabel 3. 5 Cara Pengujian Sensor	40
Tabel 3. 6 Cara Pengujian Arduino.....	40
Tabel 3. 7 Hasil Analisis	41
Tabel 3. 8 Tempat dan Waktu Penelitian	41
Tabel 4. 1 Hasil Pengujian Panel Surya	49
Tabel 4. 2 Hasil Pengujian Baterai.....	50
Tabel 4. 3 Hasil Pengujian Wemos D1 MINI	53
Tabel 4. 4 Hasil Pengujian Sensor	54
Tabel 4. 5 Hasil Pengujian Sistem	55

DAFTAR PUSTAKA

- Adani, Muhammad Robith. 2020. *Mengenal Apa Itu Internet Of Things dan Contoh Penerapannya* from <https://www.sekawanmedia.co.id/pengertian-internet-of-things/>
- Arwanda, D. 2017. *Tinjauan Pustaka Solar Cell*. From <http://eprints.polsri.ac.id/4329/3/File%20III%20%20BAB%20II.pdf>
- Fitriah, Khonif Nur. 2019. *Trainer Instalasi Pembangkit Listrik Tenaga Surya (PLTS)* from http://lib.unnes.ac.id/36704/1/5301414043_Optimized.pdf
- Jatmiko W,I. 2010. *Elektronika Daya*. Yogyakarta : Kementrian Pendidikan Nasional Universitas Negeri Yogyakarta
- Kadir, A. (2016). *Simulasi Arduino*. Jakarta: Elex Media Komputindo.
- Singh. 2008. *Elektronika Daya*. From https://industri3601.wordpress.com/elektronika_daya
- S. Wilman. 2013. *Sel Surya : Struktur dan Amp, Cara Kerja Teknologi Surya*. from <https://teknologisurya.wordpress.com/dasar-teknologi-sel-surya/prinsip-kerja-sel-surya>
- Wafi, H. A. (2019). *Implementasi Internet Of Things (IOT) Dalam Sistem Kontrol Dan Monitoring Constant Current Regulator Berbasis Arduino Menggunakan Android*. Surabaya.
- Zulkifli. (2019). *Rancang Bangun Smart Parking System dengan Konsep Internetof Thing Berbasis Mikrokontroller di Bandar Udara Internasional Adi Soemarmo Solo*. Surabaya.

LAMPIRAN A CODINGAN ARDUINO MEGA

```
#include <Wire.h>
#include <Adafruit_INA219.h>
// #include <AltSoftSerial.h>
// #include <SoftwareSerial.h>

Adafruit_INA219 ina4;
Adafruit_INA219 ina3(0x41);
Adafruit_INA219 ina2(0x44);
Adafruit_INA219 ina1(0x45);

// AltSoftSerial softSerial; // RX = 9, TX = 8
// SoftwareSerial softSerial(9, 8);

void setup() {
  Serial.begin(9600);

  // Initialize the INA219.
  // By default the initialization will use the largest range
  // (32V, 2A).
  if (!ina1.begin()) {
    Serial.println("Failed to find INA219 chip at 0x40");
    while (1) { delay(10); }
  }
  // To use a slightly lower 32V, 1A range (higher precision
  // on amps):
  ina1.setCalibration_32V_1A();
  // Or to use a lower 16V, 400mA range (higher precision on
  // volts and amps):
  // ina1.setCalibration_16V_400mA();
```

```
if (!ina2.begin()) {
  Serial.println("Failed to find INA219 chip at 0x41");
  while (1) { delay(10); }
}
ina2.setCalibration_32V_1A();
// ina2.setCalibration_16V_400mA();

if (!ina3.begin()) {
  Serial.println("Failed to find INA219 chip at 0x44");
  while (1) { delay(10); }
}
ina3.setCalibration_32V_1A();
// ina3.setCalibration_16V_400mA();

if (!ina4.begin()) {
  Serial.println("Failed to find INA219 chip at 0x45");
  while (1) { delay(10); }
}
ina4.setCalibration_32V_1A();
// ina4.setCalibration_16V_400mA();

Serial1.begin(4800);
Serial.println("Initialization completed!");
}

unsigned long latestUpdateTime = 0;
#define UPDATE_TIME 2000

#define VOLTAGE_THRESHOLD 5.0
#define CURRENT_THRESHOLD 1.0
```

```
float latestVoltage1 = 0;
float latestCurrent1 = 0;
float latestVoltage2 = 0;
float latestCurrent2 = 0;
float latestVoltage3 = 0;
float latestCurrent3 = 0;
float latestVoltage4 = 0;
float latestCurrent4 = 0;

String incomingSerialData = "";

void loop() {
  // Monitor for incomingSerialData
  while (Serial1.available()) {
    char c = Serial1.read();
    Serial.write(c);

    incomingSerialData += c;
    if (c == '\n') {
      Serial.print("> incomingSerialData: ");
      Serial.println(incomingSerialData);
      incomingSerialData = "";
    }
  }

  if (millis() - latestUpdateTime > UPDATE_TIME) {
    latestUpdateTime = millis();

    float shuntvoltage = 0;
    float busvoltage = 0;
```

```

float current_mA = 0;
float power_mW = 0;
float loadvoltage = 0;

shuntvoltage = ina1.getShuntVoltage_mV();
busvoltage = ina1.getBusVoltage_V();
if (busvoltage < VOLTAGE_THRESHOLD) {
    busvoltage = 0;
}
latestVoltage1 = busvoltage * 0.6 + latestVoltage1 * 0.4
;
current_mA = ina1.getCurrent_mA();
if (current_mA < CURRENT_THRESHOLD) {
    current_mA = 0;
    latestCurrent1 = 0;
    latestVoltage1 = 0;
}
latestCurrent1 = current_mA * 0.6 + latestCurrent1 * 0.4
;
if (current_mA <= 0) {
    current_mA = latestCurrent1;
}
Serial.print("#1 Bus Voltage:   "); Serial.print(latestV
oltage1); Serial.println(" V");
Serial.print("#1 Current:      "); Serial.print(latestC
urrent1); Serial.println(" mA");
Serial.println("");

shuntvoltage = ina2.getShuntVoltage_mV();
busvoltage = ina2.getBusVoltage_V();
if (busvoltage < VOLTAGE_THRESHOLD) {

```

```

        busvoltage = 0;
    }
    latestVoltage2 = busvoltage * 0.6 + latestVoltage2 * 0.4
;
    current_mA = ina2.getCurrent_mA();
    if (current_mA < CURRENT_THRESHOLD) {
        current_mA = 0;
        latestCurrent2 = 0;
        latestVoltage2 = 0;
    }
    latestCurrent2 = current_mA * 0.6 + latestCurrent2 * 0.4
;
    if (current_mA <= 0) {
        current_mA = latestCurrent2;
    }
    Serial.print("#2 Bus Voltage:   "); Serial.print(latestV
oltage2); Serial.println(" V");
    Serial.print("#2 Current:       "); Serial.print(latestC
urrent2); Serial.println(" mA");
    Serial.println("");

    shuntvoltage = ina3.getShuntVoltage_mV();
    busvoltage = ina3.getBusVoltage_V();
    if (busvoltage < VOLTAGE_THRESHOLD) {
        busvoltage = 0;
    }
    latestVoltage3 = busvoltage * 0.6 + latestVoltage3 * 0.4
;
    current_mA = ina3.getCurrent_mA();
    if (current_mA < CURRENT_THRESHOLD) {
        // Serial.println("Below CURRENT_THRESHOLD!");

```

```

        current_mA = 0;
        latestCurrent3 = 0;
    }
    latestCurrent3 = current_mA * 0.6 + latestCurrent3 * 0.4
;
    if (current_mA <= 0) {
        current_mA = latestCurrent3;
    }
    Serial.print("#3 Bus Voltage:   "); Serial.print(busvoltage);
    Serial.println(" V");
    Serial.print("#3 Current:      "); Serial.print(current_mA);
    Serial.println(" mA");
    Serial.println("");

    shuntvoltage = ina4.getShuntVoltage_mV();
    busvoltage = ina4.getBusVoltage_V();
    if (busvoltage < VOLTAGE_THRESHOLD) {
        busvoltage = 0;
    }
    latestVoltage4 = busvoltage * 0.6 + latestVoltage4 * 0.4
;
    current_mA = ina4.getCurrent_mA();
    if (current_mA < CURRENT_THRESHOLD) {
        current_mA = 0;
        latestCurrent4 = 0;
    }
    latestCurrent4 = current_mA * 0.6 + latestCurrent4 * 0.4
;
    if (current_mA <= 0) {
        current_mA = latestCurrent4;
    }

```

```
        Serial.print("#4 Bus Voltage:  "); Serial.print(latestV
oltage4); Serial.println(" V");
        Serial.print("#4 Current:      "); Serial.print(latestC
urrent4); Serial.println(" mA");
        Serial.println("");

        String message = String(latestVoltage1) + ","
            + String(latestCurrent1) + ","
            + String(latestVoltage2) + ","
            + String(latestCurrent2) + ","
            + String(latestVoltage3) + ","
            + String(latestCurrent3) + ","
            + String(latestVoltage4) + ","
            + String(latestCurrent4);

        Serial.println(message);
        Serial1.println(message);
    }
}
```

LAMPIRAN B CODINGAN WEMOS

```
#include <ArduinoWebsockets.h>
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>

SoftwareSerial softSerial( D1, D2); // RX, TX

const char* ssid = "OPPO A5 2020";
const char* password = "12345678";
// const char* ssid = "osdev";
// const char* password = "Indonesia";

const char* websockets_server_host = "dual-panel-
surya.tapoltekbangsby.com";
const uint16_t websockets_server_port = 7825;

using namespace websockets;

WebsocketsClient client;

void onMessageCallback(WebsocketsMessage message) {
    // Serial.print("--> Got Message: ");
    // Serial.println(message.data());

    String command = message.data();
    if (command.startsWith("")) {
        Serial.print("--> Got Message: ");
        Serial.println(command);
    } else {
        Serial.print("--> Forwarding: ");
```



```
        Serial.println(command);
        softSerial.println(command);
    }
}

void onEventsCallback(WebsocketsEvent event, String data) {
    if(event == WebsocketsEvent::ConnectionOpened) {
        Serial.println(" ! Connection Opened");
    } else if(event == WebsocketsEvent::ConnectionClosed) {
        Serial.println(" ! Connection Closed");
    } else if(event == WebsocketsEvent::GotPing) {
        Serial.println(" ! Got a Ping!");
    } else if(event == WebsocketsEvent::GotPong) {
        Serial.println(" ! Got a Pong!");
    }
}

void setup() {
    pinMode(14, OUTPUT);
    digitalWrite(14, LOW);

    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, LOW);

    Serial.begin(9600);

    // Connect to wifi
    WiFi.begin(ssid, password);

    // Wait some time to connect to wifi
```

```

    for(int i = 0; i < 15 && WiFi.status() != WL_CONNECTED; i+
+) {
    Serial.print(".");
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
    delay(1000);
}
while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println();
    Serial.println("Fail connecting!");
    scanNetworks();
    blink(); blink(); blink(); blink(); blink();
    ESP.restart();
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localI
P());
Serial.print("Signal Strength (RSSI): ");
Serial.print(dBmtoPercentage(WiFi.RSSI())); Serial.println
(" %");

Serial.println();
softSerial.begin(4800);

// Connecting to server...
digitalWrite(LED_BUILTIN, HIGH);
Serial.println("--> Connecting...");
bool connected = client.connect(websockets_server_host, we
bsockets_server_port, "/");
if(connected) {

```

```

Serial.println(" ! Connected!");
// run callback when messages are received
client.onMessage(onMessageCallback);
// run callback when events are occurring
client.onEvent(onEventsCallback);
digitalWrite(LED_BUILTIN, LOW);
} else {
  Serial.println(" ! Not Connected!");
}
}

unsigned long latestUpdateTime = 0;
#define UPDATE_TIME 1000

String incomingSerialData = "";
String availableSerialData = "";

void loop() {
  // Monitor for incomingSerialData
  while (softSerial.available()) {
    char c = softSerial.read();
    Serial.write(c);

    incomingSerialData += c;
    if (c == '\n') {
      Serial.print("> incomingSerialData: "); Serial.println
(incomingSerialData);
      availableSerialData = incomingSerialData;
      incomingSerialData = "";
    }
  }
}

```

```

// Monitor and forwarding availableSerialData
if (millis() - latestUpdateTime > UPDATE_TIME) {
    latestUpdateTime = millis();

    if (availableSerialData.length() > 0 && client.available
()) {
        Serial.println("> Sending availableSerialData...");
        Serial.println(availableSerialData);
        client.send(availableSerialData);
        availableSerialData = "";

        // dummy!
        // availableSerialData = "0" + String(dummyState == 1
? ",219.8,0.12,1" : ",0,0,0");
        // client.send(availableSerialData);
    } else {
        // Serial.println(" no available serial data.");
    }
}

// Monitor client and start accepting connection if possible
if(client.available()) {
    client.poll();
} else {
    // Connecting to server...
    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("--> Connecting...");
    bool connected = client.connect(websockets_server_host,
websockets_server_port, "/");

```

```
if(connected) {
  Serial.println(" ! Connected!");
  // run callback when messages are received
  client.onMessage(onMessageCallback);
  // run callback when events are occurring
  client.onEvent(onEventsCallback);
  digitalWrite(LED_BUILTIN, LOW);
} else {
  Serial.println(" ! Not Connected!");
}
}
}
```

LAMPIRAN C : RENCANA ANGGARAN BIAYA

No	Komponen/Alat	Jumlah	Harga Satuan	Jumlah Harga
1	Solar Panel 10 wp	2	150.000	300.000
2	Power inverter DC to AC 40 W	1	85.000	85.000
3	Solar Charge Controller 10 A	1	110.000	110.000
4	Arduino MEGA	1	180.000	180.000
5	Aki 12 V	1	210.000	210.000
6	Wemos	1	65.000	65.000
7	Sensor Arus dan Tegangan DC INA219	4	59.000	236.000
8	Kabel,Resistor,Konektor	3	50.000	150.000
9	Broadbord shield untuk arduino	1	40.000	40.000
10	Adaptor 9 V	1	45.000	45.000
11	Dan lain-lain		500.000	500.000
	JUMLAH			1.921.000

LAMPIRAN E. SOP alat Trainer PLTS

IMPLEMENTASI INTERNET *OF THINGS* PADA *TRAINER* SISTEM PEMBANGKIT LISTRIK TENAGA SURYA *MULTIPANEL*

Oleh:

OLAENI ERTINA

NIT: 30118018

Berikut ini merupakan langkah-langkah untuk mengoperasikan *trainer* pembangkit listrik tenaga surya *multipanel*:



1. Urutan Memasang
 - a) Hubungkan AKI/BATERRY ke controller
 - b) Hubungkan SOLAR PANEL
 - c) Hubungkan LOAD
2. Urutan Melepas
 - a) Lepas solar panel terlebih dahulu
 - b) Lepas baterai/aki dari controller

NOTE:

1. Controller harus dihubungkan dengan aki dulu, jika tidak ,BISA RUSAK!!!
2. Solar panel tidak boleh terhubung tanpa aki, jika tidak BISA RUSAK
3. Pasang kabel plus (+) dan minus (-) jangan sampai terbalik

LAMPIRAN F. Gambar Pengujian Alat

1. Panel surya



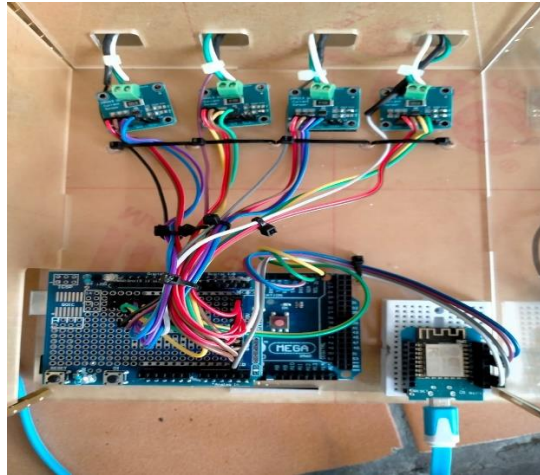
2. Solar charge controller



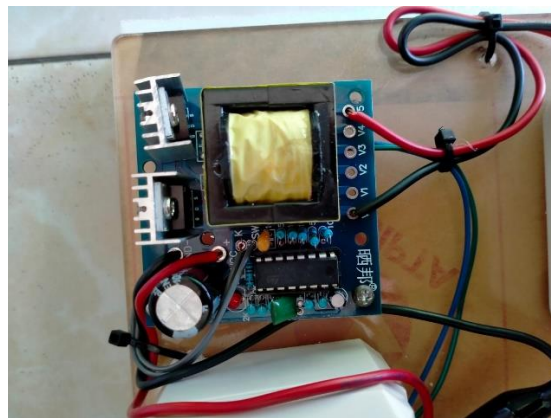
3. Baterai



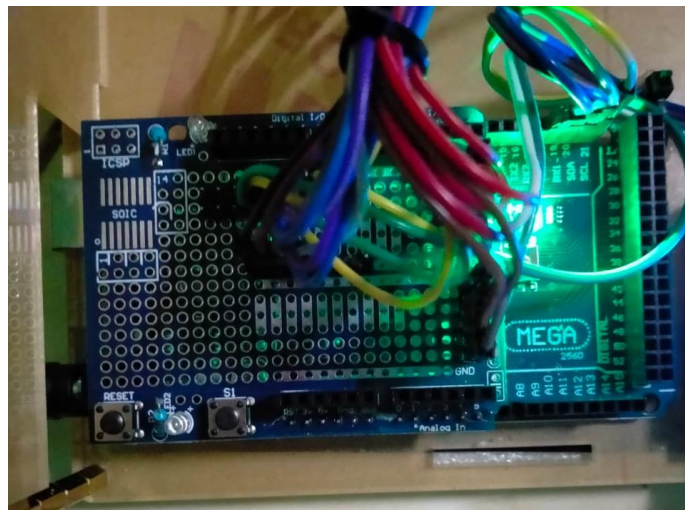
4. Sensor Arus dan Tegangan



5. Inverter



6. Arduino mega



DAFTAR RIWAYAT HIDUP



OLAENI ERTINA, lahir di Tanah Abang pada tanggal 25 Agustus 2000 putri pertama dari pasangan Bapak Sutrisman dan Ibu Sumiati. Mempunyai satu saudara kandung Olivia Nur Syafitri . Beragama Islam. Bertempat tinggal di Ds. Tanah Abang (SPE) Kec: Pamenang Kab: Merangin , JAMBI. Dengan menempuh pendidikan formal sebagai berikut :

1. SD Negeri 196 VI TANAH ABANG I Lulus tahun 2012
2. SMP Negeri 7 MERANGIN Lulus tahun 2015
3. SMA Negeri 12 MERANGIN Lulus tahun 2018

Pada bulan September 2018 diterima sebagai Taruna di Politeknik Penerbangan Surabaya Program Studi Diploma III Teknik Listrik Bandar Udara angkatan ke-XIII. Melaksanakan On the Job Training pertama dan kedua di Bandar Udara Mutiara SIS Al-Jufri Palu dari 27 juli 2020 sampai dengan 28 febuari 2021 .