

BAB IV

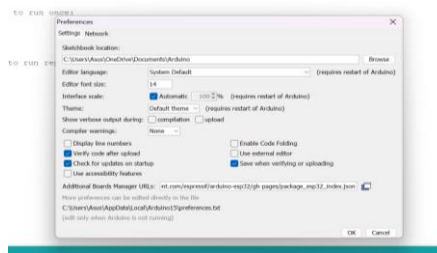
IMPLEMENTASI SISTEM

4.1 *Implementation and Unit Testing*

Implementasi yang penulis lakukan dengan melakukan pengaplikasian dari implementasi sistem, implementasi perangkat keras, dan implementasi perangkat lunak. Implementasi sistem dengan membuat program dengan *software* Arduino IDE. Implementasi perangkat keras dengan melakukan perakitan dari perancangan elektronika hingga menjadi *hardware* yang siap digunakan. Implementasi perangkat lunak dengan membuat tampilan antarmuka dengan website dengan menggunakan Bahasa pemrograman PHP pada Visual Studio Code.

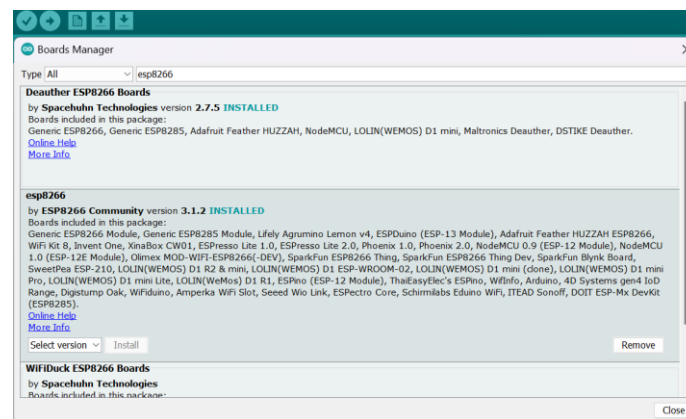
4.1.1 Implementasi Sistem

Pada implementasi sistem, penulis melakukan instalasi pada *software* Arduino IDE. Software Arduino IDE yang penulis gunakan versi 1.8.19. Langkah awal penggunaan software Arduino IDE dengan melakukan instalasi dan konfigurasi board manager dan instalasi library yang digunakan.



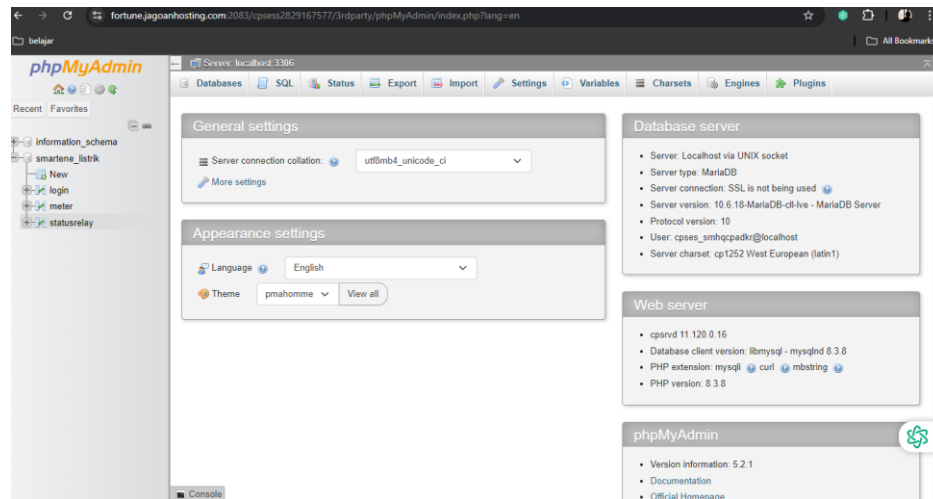
Gambar 4.1 Konfigurasi board manager

Gambar 4.1 merupakan tampilan menu *preferences* untuk konfigurasi awal pada software Arduino IDE dengan melakukan penambahan *board manager*. Fungsi penambahan *board manager* sebagai langkah awal menambahkan *board* dari NodeMCU ESP8266 pada Arduino IDE. Dalam konfigurasinya dilakukan dengan membuka menu *file – preference – tambah board manager url*.



Gambar 4.2 Instalasi Bord Manager

Gambar 4.2 merupakan tampilan dari menu *Boards Manager*. Pada Menu *Boards Manager* terdapat banyak *library* yang tersedia. Dalam menambahkan *board* NodeMCU ESP8266 dilakukan dengan membuka menu *Sketch – Include – Library – Manage Library* dan mencari dengan kata kunci “esp8266”. Fungsi dari penambahan *board* NodeMCU ESP8266 adalah sebagai *board* yang nantinya digunakan sebagai papan yang digunakan dalam pembuatan program sistem sesuai dengan kebutuhan perangkat NodeMCU yang digunakan. Pada menu *Boards Manager* ini penulis juga melakukan instalasi *library* yang dibutuhkan dalam sistem.



Gambar 4.3 Pembuatan database pada phpMyAdmin

Gambar 4.3 merupakan tahapan pembuatan database pada phpMyAdmin. Penulis membuat database dengan nama “smartene_listrik” pada server cloud *jagoan hosting*. Pembuatan database ini sebagai langkah awal sebagai wadah untuk menyimpan data pada server website.

Table	Action	Rows	Type	Collation	Size	Overhead
login	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 K	-
meter	Browse Structure Search Insert Empty Drop	4,428	InnoDB	latin1_swedish_ci	352.0 K	-
statusrelay	Browse Structure Search Insert Empty Drop	142	InnoDB	latin1_swedish_ci	16.0 K	-
3 tables	Sum	4,573	InnoDB	latin1_swedish_ci	384.0 K	0 B

Gambar 4.4 Tabel pada database

Gambar 4.4 merupakan tampilan tabel pada database yang telah dibuat. Setelah berhasil membuat database baru, langkah selanjutnya yaitu dengan membuat tiga tabel dengan masing masing tabel memiliki fungsi yang berbeda. Tabel login berfungsi untuk membuat data penggunaan yang dapat mengakses website. Tabel meter berfungsi

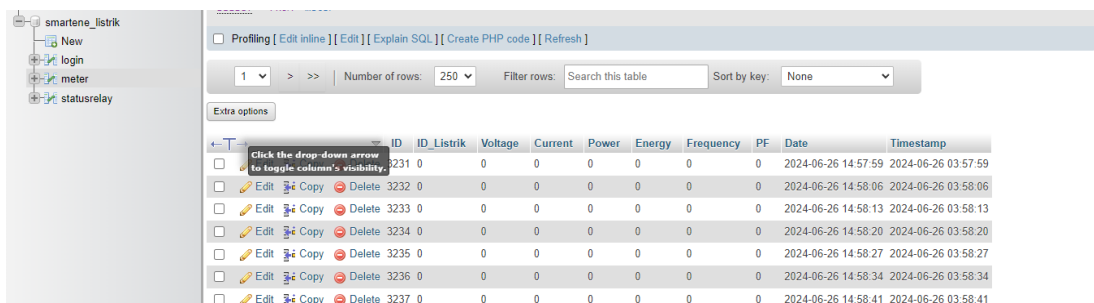
dalam menyimpan data pengujian sensor. Tabel statusrelay berfungsi dalam menyimpan hasil kondisi relay.



	id	username	password	level
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	a	a	admin
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	b	b	user
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	admin	admin	admin

Gambar 4.5 Tabel login

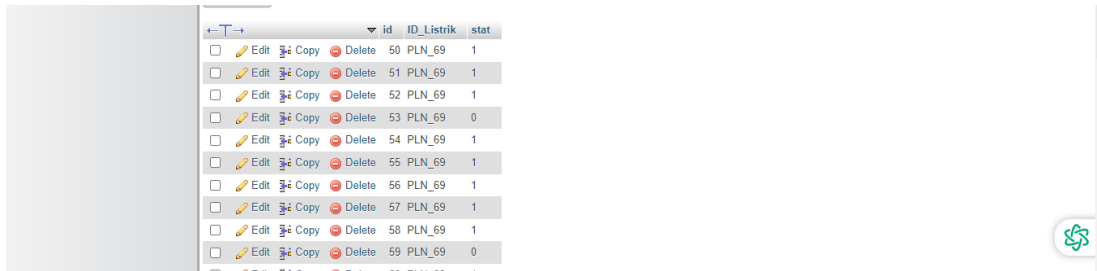
Gambar 4.5 adalah tampilan tabel login. Pada tabel login terdapat 4 kolom diantaranya id, username, password, dan level. Pada kolom level ini nantinya dapat dibedakan dengan pengguna atau admin.



	ID	ID_Listrik	Voltage	Current	Power	Energy	Frequency	PF	Date	Timestamp
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3231	0	0	0	0	0	0	0	2024-06-26 14:57:59	2024-06-26 03:57:59
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3232	0	0	0	0	0	0	0	2024-06-26 14:58:06	2024-06-26 03:58:06
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3233	0	0	0	0	0	0	0	2024-06-26 14:58:13	2024-06-26 03:58:13
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3234	0	0	0	0	0	0	0	2024-06-26 14:58:20	2024-06-26 03:58:20
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3235	0	0	0	0	0	0	0	2024-06-26 14:58:27	2024-06-26 03:58:27
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3236	0	0	0	0	0	0	0	2024-06-26 14:58:34	2024-06-26 03:58:34
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3237	0	0	0	0	0	0	0	2024-06-26 14:58:41	2024-06-26 03:58:41

Gambar 4.6 Tabel meter

Pada gambar 4.6 merupakan tampilan tabel meter. Pada tabel meter berfungsi dalam menyimpan data hasil pengujian sensor. Data yang disimpan dikirim secara berkala ke server database ini. Data yang disimpan ID_Listrik, *voltage*, *current*, *power*, *frequency*, *power factor*, *date*, dan *timestamp*.



	id	ID_Listrik	stat
<input type="checkbox"/>	50	PLN_69	1
<input type="checkbox"/>	51	PLN_69	1
<input type="checkbox"/>	52	PLN_69	1
<input type="checkbox"/>	53	PLN_69	0
<input type="checkbox"/>	54	PLN_69	1
<input type="checkbox"/>	55	PLN_69	1
<input type="checkbox"/>	56	PLN_69	1
<input type="checkbox"/>	57	PLN_69	1
<input type="checkbox"/>	58	PLN_69	1
<input type="checkbox"/>	59	PLN_69	0

Gambar 4.7 Tabel *statusrelay*

Gambar 4.7 merupakan tabel dari status relay. Pada tabel *statusrelay* sebagai penyimpan hasil kondisi relay. Terdapat kolom dengan ID_Listrik yang terintegrasi dengan ID_Listrik pada tabel meter sebelumnya. Pada kolom stat digunakan untuk menyimpan hasil pembacaan relay dengan nilai 1 untuk ketika listrik hidup dan relay 0 untuk listrik mati.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
```

Kode diatas merupakan kode untuk inisiasi *library* yang siap digunakan pada pembuatan program sistem. Kode ini berfungsi dalam membaca apakah *library* yang sudah bisa digunakan. *Library* yang dibutuhkan dalam pembuatan program sistem yaitu *library* NodeMCU ESP8266 wifi dan HTTPClient, *library* LCD 20x4 I2C, *library* Modul sensor PZEM004Tv30, dan *library* *Software Serial*.

```
#if defined(ESP32)
  #error "Software Serial is not supported on the ESP32"
#endif

/* Use software serial for the PZEM
 * Pin 12 Rx (Connects to the Tx pin on the PZEM)
```

```

    * Pin 13 Tx (Connects to the Rx pin on the PZEM)
    */
    #if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)
    #define PZEM_RX_PIN 12
    #define PZEM_TX_PIN 13
    #endif

    SoftwareSerial pzemSWSerial(PZEM_RX_PIN, PZEM_TX_PIN);
    PZEM004Tv30 pzem(pzemSWSerial);
    LiquidCrystal_I2C lcd(0x27, 20, 4);

```

Kode diatas merupakan bagian program untuk inisiasi awal pada NodeMCU ESP8266. Terdapat program untuk membaca apakah sistem menggunakan NodeMCU ESP32 atau NodeMCU ESP8266. Kemudian dalam membuat inisiasi pin pada NodeMCU ESP8266 digunakan pin 12 sebagai pin RX dan pin 13 sebagai pi TX. Selanjutnya, membuat deklarasi variabel pada pzemSWSerial untuk membaca pin sensor pzem yang digunakan, pzem untuk variabel dari pengambilan data pzemSWSerial, lcd untuk variabel pada pembacaan LCD.

```

// Replace with your network credentials
const char* ssid = "iotsaya";
const char* password = "farid123";

// Your server URL
const char* serverNamePost =
"http://smartenergylistrik.my.id/store.php";
const char* serverNameGet =
"http://smartenergylistrik.my.id/get_relay.php?ID_Listrik=PLN_69";

// set price per kwh
const float hargaPerKWh = 1.352;

// define pin relay
const int relayPin = 2;
WiFiClient client;

```

Kode diatas merupakan program dalam inisiasi *wifi*, server, pembuatan variabel harga, dan pin yang digunakan. *SSID* dan *password* digunakan untuk menghubungkan NodeMCU ESP8266 dengan koneksi *wifi*. Server Name Post dan server Name Get sebagai server untuk pengiriman data ke website. Pembuatan harga per kwh sebagai nilai awal dalam perhitungan biaya penggunaan listrik. Pin relay yang digunakan pada NodeMCU ESP8266 adalah pin 2.

```
void setup() {
  Serial.begin(115200);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW);

  WiFi.begin(ssid, password);

  Serial.print("Connecting to ");
  Serial.print(ssid);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println();
  Serial.println("Connected to WiFi");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  lcd.init();
  lcd.clear();
  lcd.backlight();
}
```

Selanjutnya, program *void setup* digunakan untuk memberikan kondisi awal sistem sebelum melakukan proses. Pengaturan relay dengan posisi *output* dan *low*, pengaturan *wifi* untuk mendeteksi sistem sudah terhubung dengan koneksi *wifi*, dan pengaturan awal pada *lcd*.

```
void getRelay() {
  if (WiFi.status() == WL_CONNECTED) { // Check Wi-Fi connection
    status
    HTTPClient http; // Declare an object of class HTTPClient
```

```

    http.begin(client, serverNameGet); // Specify request
    destination using the new API
    int httpCode = http.GET(); // Send the request

    if (httpCode > 0) { // Check the returning code
        String payload = http.getString(); // Get the request response
        payload
        Serial.println(payload);

        if (payload == "0") {
            digitalWrite(relayPin, HIGH); // Turn LED on
        } else if (payload == "1") {
            digitalWrite(relayPin, LOW); // Turn LED off
        } else {
            Serial.println("Unexpected response");
        }
    } else {
        Serial.println("Error on HTTP request");
    }

    http.end(); // Close connection
}
}

```

Kode diatas merupakan program untuk mendapatkan nilai dari pembacaan relay. Dalam penerapannya pada sistem, terdapat dua kondisi relay yang dibuat, yaitu HIGH dan LOW. Ketika relay dalam kondisi terbaca “0” maka relay akan hidup yang dimana relay memutus arus listrik dan jika relay dalam kondisi terbaca “1” maka relay akan mati. Data nilai tersebut kemudian dikirimkan pada server dengan menggunakan metode HTTP get pada server Name Get.

```

void sendData(String url) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        // Use the new API method with WiFiClient
        http.begin(client, url);

        // Send HTTP GET request
        int httpResponseCode = http.GET();
    }
}

```



```

    if (httpResponseCode > 0) {
        if (httpResponseCode == HTTP_CODE_MOVED_PERMANENTLY ||
            httpResponseCode == HTTP_CODE_FOUND) {
            // Follow the redirect
            String newUrl = http.getLocation();
            Serial.print("Redirected to: ");
            Serial.println(newUrl);
            http.end();
            http.begin(client, newUrl);
            httpResponseCode = http.GET();
        }

        String response = http.getString();
        Serial.println(httpResponseCode); // Print return code
        Serial.println(response);         // Print request answer
    } else {
        Serial.print("Error on sending GET: ");
        Serial.println(httpResponseCode);
    }

    http.end();
} else {
    Serial.println("Error in WiFi connection");
}
}

```

Bagian kode *void sendData (String url)* berfungsi dalam melakukan pengiriman data ke url website. Menggunakan http dengan metode get dengan melihat kondisi sistem yang terkoneksi dengan jaringan *wifi*.

```

void loop() {

    // Read the data from the sensor
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float energy = pzem.energy();
    float frequency = pzem.frequency();
    float pf = pzem.pf();
    float harga = (energy, 3 * hargaPerKWh;

    // Form the complete URL with parameters
    String url = String(serverNamePost) +
        "?ID_Listrik=PLN_69&Voltage=" + String(voltage) + "&Current=" +

```

```
String(current) + "&Power=" + String(power) + "&Energy=" +
String(energy) + "&Frequency=" + String(frequency) + "&PF=" +
String(pf);
```

```
// Print the URL to debug
Serial.print("Requesting URL: ");
Serial.println(url);
```

```
// Send data to the server
sendData(url);
```

```
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("MONITORING LISTRIK");
lcd.setCursor(0, 1);
lcd.print("Ratman Mebel");
lcd.setCursor(1, 2);
lcd.print("Listrik 900Va");
lcd.setCursor(1, 3);
lcd.print("Rp.");
lcd.setCursor(4, 3);
lcd.print(harga);
```

```
delay(3000);
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Voltase :");
lcd.setCursor(9, 0);
lcd.print(voltage);
lcd.setCursor(0, 1);
lcd.print("Ampere :");
lcd.setCursor(9, 1);
lcd.print(current);
lcd.setCursor(0, 2);
lcd.print("Watt :");
lcd.setCursor(9, 2);
lcd.print(power);
lcd.setCursor(0, 3);
lcd.print("kWh :");
lcd.setCursor(9, 3);
lcd.print(energy);
```

```
delay(3000);
```

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Frequency :");
lcd.setCursor(11, 0);
lcd.print(frequency);
```

```

    lcd.setCursor(0,1);
    lcd.print("PF      :");
    lcd.setCursor(11, 1);
    lcd.print(pf);

    delay(3000);

    getRelay();

    delay(3000);
}

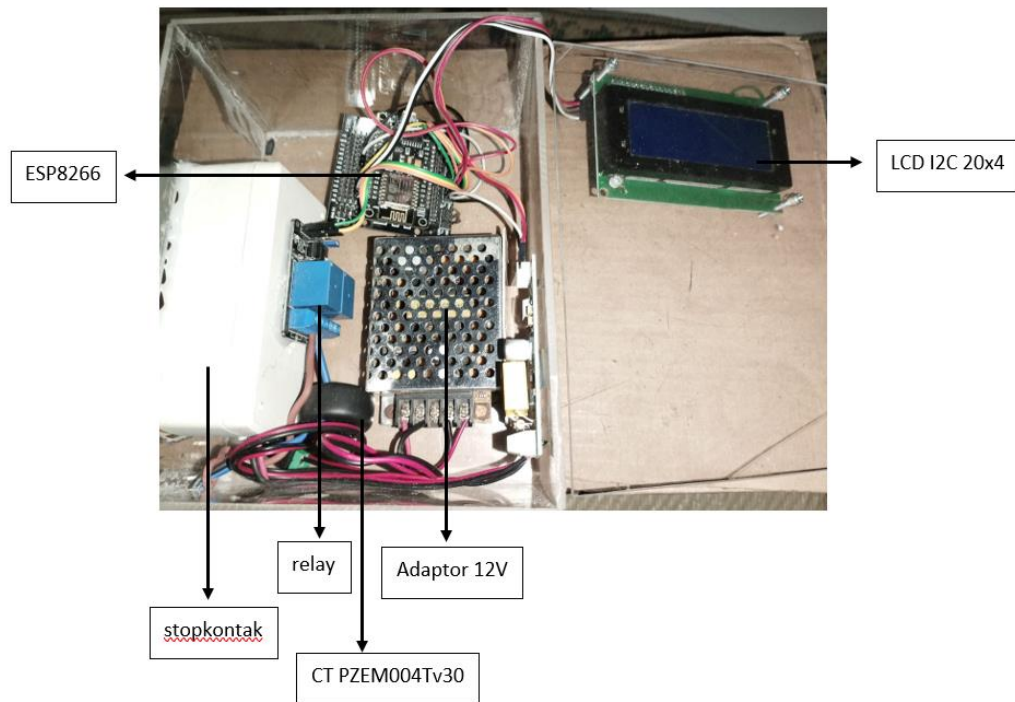
```

Pada kode *void loop* merupakan program inti yang dibuat untuk mendeteksi nilai *voltage, current, power, energy, frequency*, pf, dan pemberian harga penggunaan listrik. Pada bagian pengiriman kode dengan *serverNamePost* ini data nilai yang terbaca akan dikirimkan ke server database. Penulis melakukan pengaturan lcd sebagai interface untuk menampilkan hasil monitoring pada *hardware* sistem.

4.1.2 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan hasil perakitan mekanik dari desain perancangan yang telah dibuat. Perakitan dimulai dengan sensor PZEM004Tv30, relay, dan LCD 20x4 I2C dihubungkan dengan NodeMCU ESP8266. Penggunaan steker dihubungkan dengan stopkontak dan dicabangkan dengan kabel untuk dihubungkan dengan *Power supply* 12v. Dalam menghubungkan sensor PZEM004Tv30 ke stop kontak dihubungkan dengan dua kabel dari output sensor PZEM004Tv30 dan dua kabel output lain dihubungkan ke CT. Penggunaan step down DC to DC untuk menurunkan tegangan DC dari power supply sebelum dihubungkan ke NodeMCU ESP8266 menggunakan ESP8266 shield. Pada implementasi perangkat keras ini semua perangkat menjadi satu wadah. Wadah yang digunakan adalah akrilik yang dibuat

menjadi kotak. Tampilan hasil implementasi perangkat keras dapat dilihat pada gambar 4.8.



Gambar 4.8 Hasil Implementasi Perangkat Keras

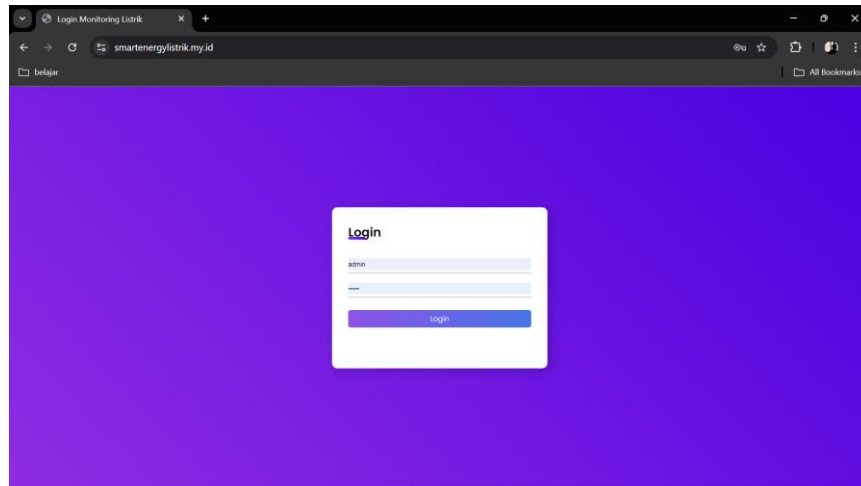
4.1.3 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak, penulis melakukan pembuatan website dengan tampilan awal halaman login, dashboard, dan tampilan grafik. Bahasa pemrograman yang digunakan menggunakan Bahasa *php* dengan pembuatan tampilan menggunakan *css*. Website dapat diakses pada :

Link : <https://smartenergylistrik.my.id/>

Username : admin

Password : admin



Gambar 4.9 Halaman login website

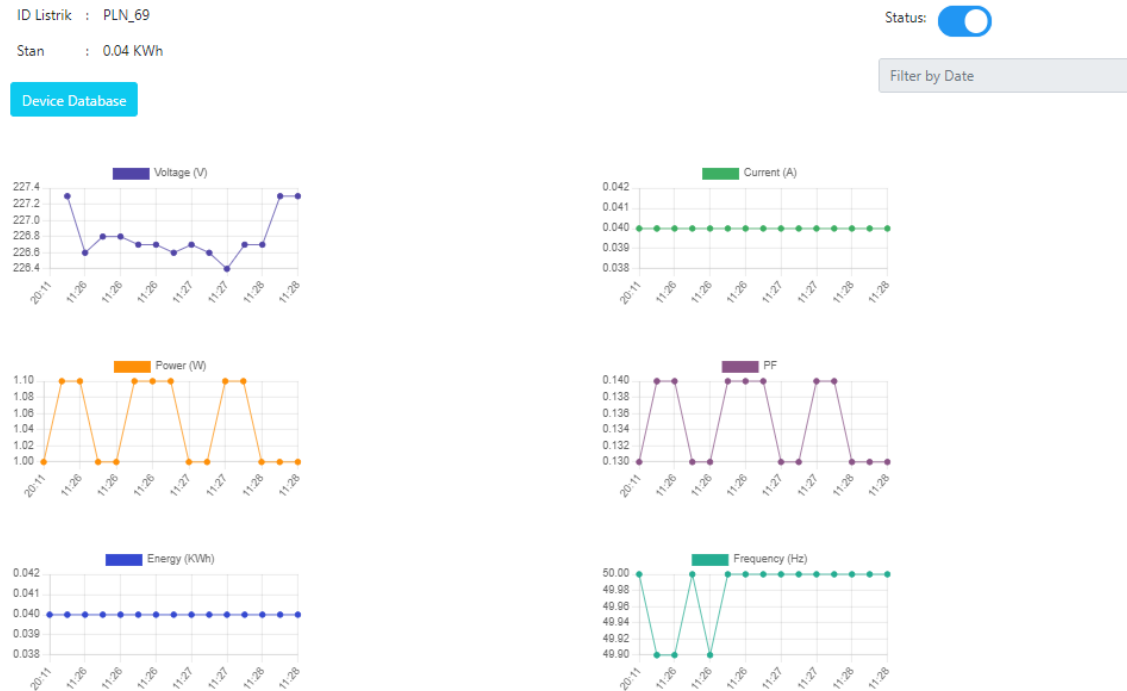
Pada gambar 4.9 merupakan hasil dari halaman login yang telah dibuat. Halaman login berfungsi dalam memberikan keterbatasan akses untuk beberapa user yang sudah terdaftar pada database.

ID Listrik	Voltage (V)	Current (A)	Power (W)	Energy (KWh)	Frequency (Hz)	PF	Timestamp
PLN_69	222.80	0.25	25.90	0.01	50.00	0.46	2024-07-10 21:06:10
PLN_999	220	10	2200	220	50	1	2024-07-10 21:01:34
PLN_911	221.60	0.16	31.50	14.52	49.90	0.88	2024-06-27 15:19:38
PLN_123	19.36	1	220	220	50	1	2024-06-26 15:33:45
PLN_002	220	1	220	220	50	1	2024-06-26 14:58:47
0	0	0	0	0	0	0	2024-06-26 14:57:59

Gambar 4.10 tampilan dashboard tabel monitoring

Gambar 4.10 merupakan tampilan dashboard pada website. Tampilan dashboard ini berisi data *ID_Listrik*, *voltage*, *current*, *power*, *energy*, *frequency*, *pf*, dan *timestamp*

yang diperoleh dari data hasil pengujian sensor PZEM004Tv30. Penggunaan ID pada tabel berbeda pada setiap ID yang digunakan. Pada percobaan penulis menggunakan ID_Listrik PLN_69.



Gambar 4.11 Tampilan grafik pada website

Pada gambar 4.11 merupakan tampilan grafik pada website. Pada halaman ini berisi grafik dari nilai *voltage*, *current*, *power*, *energy*, *frequency*, dan *pf* dari setiap ID_Listrik yang digunakan. Pada halaman ini terdapat tombol relay untuk memutus arus secara manual melalui website. Fitur device database memungkinkan user untuk mengunduh hasil data monitoring menjadi bentuk .csv.

4.2 Integration and System Testing

Pada tahap selanjutnya penulis melakukan pengujian dengan sampel yang telah ditentukan untuk memperoleh data pengujian dari sistem yang telah dibuat. Tujuan dari tahapan ini untuk mengetahui sejauh mana kemampuan sistem dalam membaca data sehingga dapat diketahui resiko dan perbaikan pada sistem. Pada tahap ini penulis melakukan pengujian pada pembacaan nilai pada sensor PZEM004Tv30, pengujian relay, dan pengujian pengiriman data pada website dan database.

4.2.1 Pengujian Sensor PZEM004Tv30

1. Pengujian Voltase

Pengujian pertama dengan mencari nilai *voltase* pada sistem yang telah selesai dibuat. Pengujian ini dilakukan dengan mengambil 5 data sampel yang diambil dari peralatan pada Ratman Mebel. Hasil pengujian nilai *voltase* ini kemudian dibandingkan dengan nilai *voltase* yang diukur menggunakan multimeter. Hasil dari pengujian *voltase* dapat dilihat pada tabel 4.1.

Tabel 4.1 pengujian voltage pada sistem

Alat	Voltase (sensor)	Voltase	Selisih error (%)
Bor listrik	219.70	220	0.13%
Gerinda	213.60	216	1.11%
Serut	217.10	219	0.86%
Mesin lis	211.90	214	0.98%
Mesin Las	210.70	212	0.61%

Rata – rata error pada pengujian	0.73%
---	--------------

Dari percobaan yang sudah dilakukan, pengujian voltase pada kelima data diperoleh persentase error pada sistem sebesar 0.73%.

2. Pengujian Keseluruhan Hardware

Pada pengujian keseluruhan sistem dilakukan dengan menguji sejauh mana sistem dapat membaca data dengan parameter yang sudah ditentukan. Parameter pengujian yang digunakan yaitu nilai *voltage*, *current*, *power*, *energy*, *frequency*, *pf*, dan biaya penggunaan yang terbaca. Untuk lebih memahami istilah dalam data, daftar istilah dan rumus perolehan data secara teori dapat dilihat pada tabel 4.2.

Tabel 4.2 Daftar istilah dan rumus dasar

Parameter	Fungsi	Satuan/ Rumus
<i>Voltase</i>	Potensial listrik	V (<i>Voltage</i>)
<i>Current</i>	Kuat arus atau muatan listrik	I (<i>Ampere</i>)
<i>Power</i>	Daya listrik diukur sebagai laju penggunaan listrik	P (<i>Watt</i>), $P = V \times I$
<i>Energy</i>	Energi yang dihasilkan aliran listrik dalam suatu waktu.	KWh (Kilo Watt Hours). $E = P \times t$ $KWh = (V \times I \times t)/1000.$

<i>frequency</i>	Frekuensi ini berhubungan dengan gelombang listrik.	H (Hertz), $f = n/t$
<i>Power Factor</i>	Factor daya listrik	Q

Sedangkan dalam perhitungan biaya penggunaan listrik, penulis membuat persamaan seperti berikut:

$$\text{Biaya} = \text{energy} \times 1.352 \dots \dots \dots (7)$$

Dimana :

Energy = nilai data sensor (KWh)

Nilai 1.352 merupakan tetapan harga untuk pelanggan subsidi PLN pada penggunaan meteran listrik 900vA.

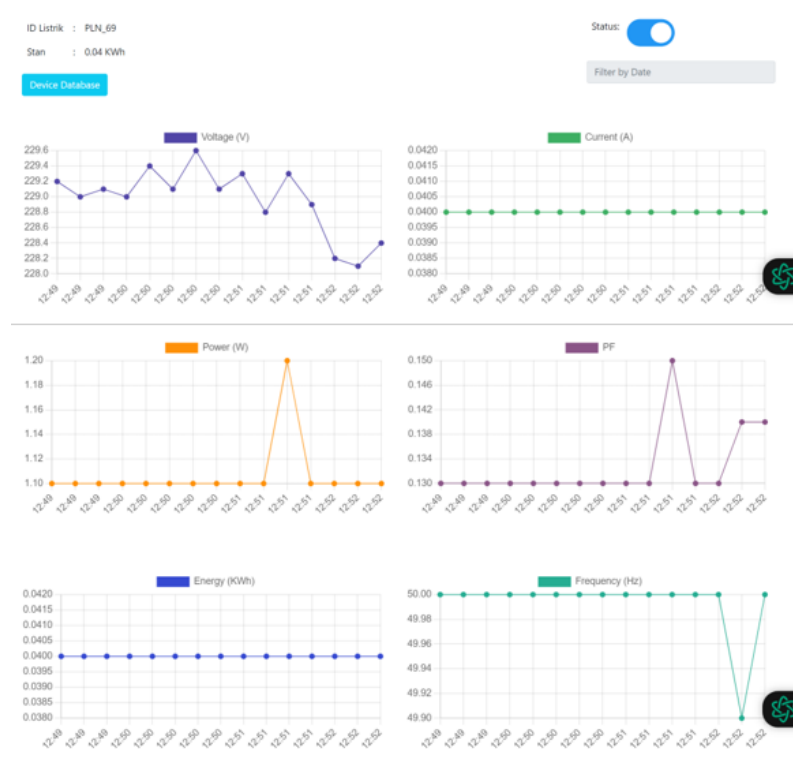
Hasil pengujian sistem dapat dilihat pada tabel 4.3.

Tabel 4.3. Pengujian Keseluruhan sistem

Alat	Voltase (V)	Arus (A)	Daya (W)	KWh	PF	F	Harga
Bor listrik	219.70	0,04	31.00	0.03	0.13	50.00	Rp4.06
Gerinda	213.60	1,41	298.30	0.04	0.99	50.00	Rp4.07
Serut	217.10	1,20	249.50	0.04	0.99	50.00	Rp4.07
Mesin lis	211.90	0,23	44.80	0.04	0.94	50.00	Rp4.07
Mesin las	210.70	0.75	145.20	0.04	0.91	50.00	Rp4.07

Tabel 4.3 merupakan hasil pengujian sistem dengan menggunakan 5 kali percobaan. Diperoleh parameter nilai *voltage*, *current*, *power*, *energy*, *frequency*, pf, dan biaya penggunaan. Hal ini menunjukkan sistem sudah berjalan dengan cukup baik.

Dari data yang telah diujikan, sistem dapat menampilkan hasil pengujian berupa nilai dan data grafik. Tampilan grafik pada pengujian sistem, dapat dilihat pada gambar 4.12.

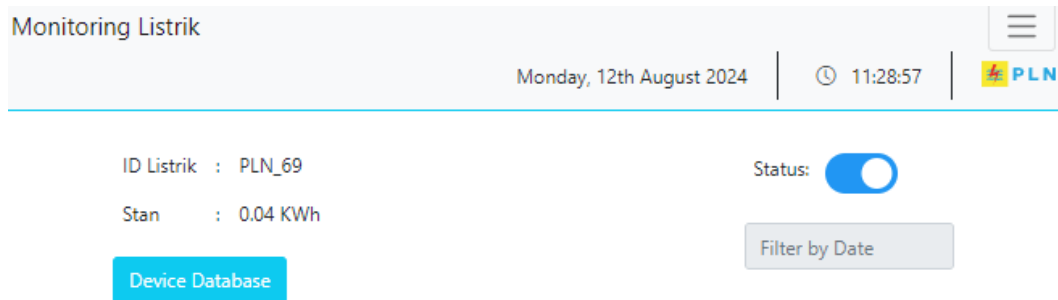


Gambar 4.12 Tampilan Grafik Pengujian

4.2.2 Pengujian Relay

Pengujian selanjutnya dilakukan pengujian pada relay. Pengujian ini bertujuan untuk menyelaraskan fungsi relay dengan tombol power pada website. Pengujian relay

ini dilakukan secara manual dengan menekan tombol pada website. Tampilan tombol penggunaan relay pada sistem menggunakan “status”. Tampilan penggunaan status untuk tombol relay pada gambar 4.13.



Gambar 4.13 Tampilan Tombol untuk status Relay

Pengujian relay dilakukan dengan menekan tombol pada website pada kondisi awal dengan listrik terhubung. Pada percobaan untuk mematikan tombol (OFF) kondisi relay ON memutus arus, *delay* dari sistem sebesar 4.86 detik. Pada kondisi tombol hidup (ON) relay OFF, diperoleh *delay* sebesar 8.89 detik. Hasil dari percobaan relay ini dapat dilihat pada tabel 4.4.

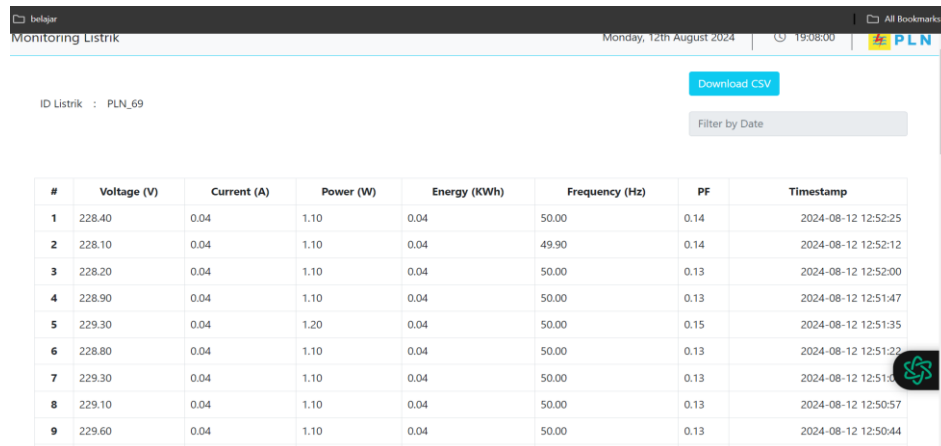
Tabel 4.4 Pengujian Relay

Kondisi relay	Tombol website	Delay
ON	OFF	4.86 detik
OFF	ON	8.89 detik

4.2.3 Pengujian Database pada Website

Pada pengujian website dilakukan dengan membaca nilai sensor yang terbaca dan dapat terkirim ke database. Pengujian yang digunakan yaitu nilai *voltage*, *current*,

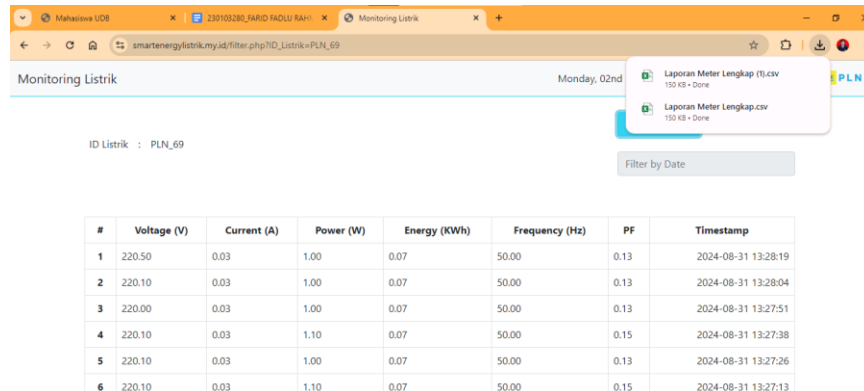
power, *energy*, *frequency*, dan *pf*. Pengiriman data ke database website dengan delay pengiriman setiap 1 detik. Tampilan data yang berhasil terkirim ke database dijelaskan pada gambar 4.14.



#	Voltage (V)	Current (A)	Power (W)	Energy (KWh)	Frequency (Hz)	PF	Timestamp
1	228.40	0.04	1.10	0.04	50.00	0.14	2024-08-12 12:52:25
2	228.10	0.04	1.10	0.04	49.90	0.14	2024-08-12 12:52:12
3	228.20	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:52:00
4	228.90	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:51:47
5	229.30	0.04	1.20	0.04	50.00	0.15	2024-08-12 12:51:35
6	228.80	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:51:23
7	229.30	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:51:11
8	229.10	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:50:57
9	229.60	0.04	1.10	0.04	50.00	0.13	2024-08-12 12:50:44

Gambar 4.14 Realtime Database pengujian sensor

Pada gambar 4.14 diperoleh hasil pengujian pada setiap nilai parameter. Delay untuk mengirimkan *realtime database* setiap 1 detik. Terdapat tabel menu *timestamp* untuk melihat waktu kapan data tersebut dikirimkan. Pada fitur database ini juga dibuat fitur “download csv” yang berfungsi untuk mengunduh database menjadi file dengan format .csv yang dapat dilihat pada gambar 4.15.



Monitoring Listrik

Monday, 02nd

ID Listrik : PLN_69

Filter by Date

#	Voltage (V)	Current (A)	Power (W)	Energy (KWh)	Frequency (Hz)	PF	Timestamp
1	220.50	0.03	1.00	0.07	50.00	0.13	2024-08-31 13:28:19
2	220.10	0.03	1.00	0.07	50.00	0.13	2024-08-31 13:28:04
3	220.00	0.03	1.00	0.07	50.00	0.13	2024-08-31 13:27:51
4	220.10	0.03	1.10	0.07	50.00	0.15	2024-08-31 13:27:38
5	220.10	0.03	1.00	0.07	50.00	0.13	2024-08-31 13:27:26
6	220.10	0.03	1.10	0.07	50.00	0.15	2024-08-31 13:27:13

Gambar 4.15 Hasil download ke format csv

4.2.4 Pengujian Internet Of Things Keseluruhan Sistem

Dari implementasi perangkat lunak dan perangkat keras, penulis melakukan pengujian dari keseluruhan fitur hardware dan website untuk memastikan sistem dan website berjalan dengan baik, data terkumpul secara akurat, dan pengendalian alat listrik berjalan sesuai dengan yang diharapkan.

Tabel 4.5 Fitur pada website

Menu	Fungsi	Keterangan
Login	Pemberian hak akses untuk user	Berhasil
Klik “ID_Listrik”	Untuk menampilkan hasil monitoring data	Berhasil
Detail.php	Menampilkan grafik hasil monitoring data	Berhasil
Device database	Menyimpan nilai data monitoring sensor pada website	Berhasil

Tombol switch	Sebagai tombol untuk status ON/OFF pada relay	Berhasil
Download csv	Mengunduh data hasil monitoring menjadi format csv	Berhasil

Tabel 4.5 merupakan hasil dari kesimpulan website yang terdiri dari halaman login, halaman dashboard dengan “ID_Listrik”, dan grafik monitoring. Fitur lain devie database, tombol switch, dan download csv berhasil diimplementasikan.

Tabel 4.6 Hasil Pengujian Hardware

Pengujian	Keterangan	Delay
PZEM004Tv30 membaca nilai sensor	Berhasil	1 detik
ESP8266 mengirim data ke LCD	Berhasil	3 detik
ESP8266 mengirim data ke Website	Berhasil	1 detik
Kondisi ON Relay	Listrik mati	4, 86 detik
Kondisi OFF Relay	Listrik hidup	8,89 detik

Pada tabel 4.6 merupakan kesimpulan dari pengujian hardware sistem dengan melakukan beberapa inti pengujian dari pengujian PZEM004Tv30, pengujian ESP8266, dan pengujian Relay. Sistem berhasil menjalankan pengiriman data ke LCD dan website.

4.3 Operation and Maintenance

Dari pengujian yang dilakukan, sistem dapat membaca nilai sensor dengan baik. Dalam pengujiannya sensor masih dalam tahapan prototype. Batas dari pengujian sistem terbatas pada nilai maksimal tegangan 220 V. Pengiriman data pengujian ke sistem juga dipengaruhi oleh jaringan internet. Pengiriman data dari pengujian nilai *voltase, current, power, energy, power factor*, dan *frekuensi* ke website harus melalui *refresh page* untuk menampilkan data ke database website.