

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Penelitian terdahulu yang memiliki topik sejenis dengan tugas akhir ditinjau sebagai referensi. Studi yang dikaji berkaitan dengan sistem akuisisi data konsumsi listrik berbasis *Internet of Things* (IoT). Tabel 2.1 meninjau penelitian terdahulu berdasarkan perangkat keras, perangkat lunak, dan hasil.

Tabel 2.1 Tinjauan Pustaka

Peneliti	Perangkat Keras	Perangkat Lunak	Hasil
Jokanan dkk [10]	Sensor PZEM-004T, Mikrokontroler NodeMCU ESP8266, Relay	Android studio, aplikasi monitoring daya listrik (MDL), dan Firebase	Hasil pengukuran diuji keakuratannya menggunakan <i>clamp</i> multimeter dan <i>power</i> meter saat diberi beban listrik berupa solder, <i>vacuum cleaner</i> , dan <i>heatgun</i> . Dari pengujian didapatkan rata-rata eror tegangan 0,35%, arus 6,7%, daya 6,6%, dan energi 7,8%.
Pangestu dkk [11]	ACS712, NodeMCU ESP8266, ATmega8, LCD	Blynk	Keakuratan alat diuji dengan beban induktif (2 buah lampu) dan beban resistif (setrika 350W). Tingkat akurasi alat dalam membaca besaran arus dan daya berkisar antara 96% hingga 98%.
Muralidhara dkk [14]	ACS712, Arduino Uno, NodeMCU ESP8266	<i>ThingSpeak</i>	Keakuratan sistem diuji menggunakan peralatan yang memiliki konsumsi daya tetap (lampu meja dan televisi) dan bervariasi (pemanggang roti dan setrika). Hasil pengukuran menunjukkan sensor memiliki eror bervariasi dari 0.03% sampai 0.45% setelah dibandingkan dengan spesifikasi daya yang diketahui.
L. B. Rahawarin [15]	ACS712, Arduino Uno, NodeMCU ESP8266	<i>ThingSpeak</i> dan MIT App Inventor	Hasil penelitian menunjukkan daya rata-rata yang dihitung oleh program adalah 3,04 W dengan <i>error</i> 1,3% dari perbandingan dengan spesifikasi daya lampu di kotak yang sebesar 3 W.
R. Ibrahim dan B. Yulianti [16]	PZEM-004T V3, NodeMCU ESP8266	Blynk IoT	Hasil penelitian ini menunjukkan bahwa nilai rata-rata <i>error</i> tegangan adalah 0,0038%, arus adalah 0,1116%, dan daya adalah 0,00294% berdasarkan pengujian menggunakan alat ukur multimeter. Nilai rata-rata

Peneliti	Perangkat Keras	Perangkat Lunak	Hasil
			eror ini masih dalam batas toleransi sensor yang sebesar 0,5%.
Soh dkk [17]	ACS712, Intel Edison dengan WiFi, LCD	Ubidots IoT	Sistem memantau konsumsi energi dan mengirim data konsumsi energi ke <i>Ubidots Cloud</i> melalui koneksi <i>Wi-Fi</i> . Penggunaan energi ditampilkan secara <i>real-time</i> melalui <i>dashboard online</i> dan <i>Ubidots Event Manager</i> akan memberikan peringatan melalui email dan Telegram jika konsumsi energi melebihi batas yang ditetapkan.
B. S. Mahendra dan A. S. Budi [18]	ACS712, ZMPT101B, NodeMCU ESP32	nRF Connect	Sensor tegangan memiliki nilai akurasi sebesar 98,77% dan eror 1.23%. Sementara itu, kedua sensor arus memiliki eror sebesar 22.93% dan 29.33%.
Huda dkk [19]	Sensor LDR, NodeMCU ESP8266	Arduino Sketch	Pengembangan perangkat keras dan lunak yang dapat memantau konsumsi listrik secara <i>real-time</i> melalui internet, serta mencatat data kWh yang akurat dan dapat diakses dengan mudah.
Chamdareno dkk [20]	Power Meter Toky, Modbus RS485 to Ethernet converter	Software LabVIEW	Ditemukan pemborosan karena pengoperasian di luar jam operasional. Pembatasan jam operasional dapat menghemat energi listrik sebesar 4898 kWh/tahun.
Laksana dkk [21]	Sensor Arus, NodeMCU	Kodular	Data penggunaan listrik dari <i>database</i> terlihat di aplikasi, sensor akurat dalam mengukur konsumsi daya listrik.

Berdasarkan beberapa penelitian terdahulu, perangkat keras yang digunakan dalam pembuatan sistem akuisisi data adalah sensor, mikrokendali, dan tambahan seperti relay dan LCD. Perangkat lunak yang digunakan sebagai antarmuka pengguna bervariasi, seperti website atau aplikasi. Metode yang digunakan pada penelitian terdahulu adalah eksperimental. Tugas akhir ini menggunakan sensor PZEM-004T, mikrokendali NodeMCU ESP8266, *relay*, *website ThingSpeak*, dan aplikasi Telegram Bot untuk pembuatan sistem akuisisi data konsumsi listrik.

## 2.2 Dasar Teori

### 2.2.1 Parameter Listrik

Parameter listrik merupakan besaran fisis yang digunakan untuk mengukur listrik. Dalam sistem akuisisi data, parameter-parameter ini berguna gambaran

umum mengenai efisiensi, keamanan, dan biaya penggunaan energi. Parameter listrik yang umumnya digunakan adalah sebagai berikut:

1. Tegangan (Volt/V)

Tegangan listrik adalah beda potensial antara dua titik dalam suatu rangkaian [22]. Beda potensial di antara dua titik menjadi pendorong utama bagi arus untuk mengalir.

2. Arus (Ampere/A)

Arus listrik merupakan aliran muatan listrik yang mengalir tiap satuan waktu [22]. Nilai arus bergantung dengan beban yang terhubung dengan rangkaian.

3. Frekuensi (Hz)

Frekuensi listrik adalah jumlah gelombang listrik yang terdapat dalam satu detik [23]. Frekuensi listrik dalam jaringan listrik di Indonesia adalah 50Hz.

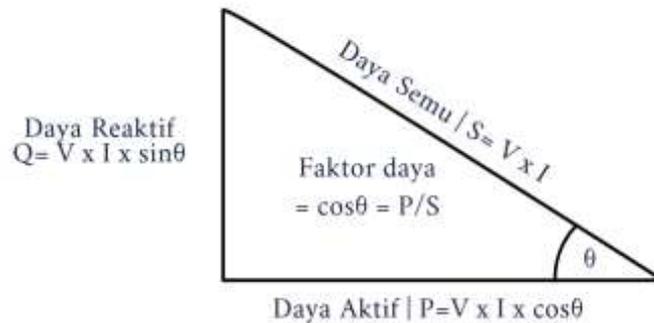
4. Faktor daya

Faktor daya adalah rasio antara daya aktif dan daya semu dalam suatu rangkaian listrik. Nilai faktor daya berkisar antara -1 sampai 1 (kisaran nilai  $\cos \theta$ ), semakin mendekati 0, maka semakin banyak energi yang terbuang. Faktor daya dapat di hitung menggunakan rumus berikut [24]:

$$Faktor\ daya = \frac{P}{S} = \frac{VI\cos\theta}{VI} = \cos\theta \quad (1)$$

5. Daya listrik.

Daya listrik adalah besarnya energi listrik yang digunakan suatu beban dalam rangkaian listrik setiap satuan waktu. Pada arus bolak balik atau *alternating current* (AC), daya listrik terdiri dari daya aktif, daya reaktif, dan daya semu.



Gambar 2.1 Segitiga Daya [24]

- a. Daya aktif adalah daya yang benar-benar digunakan oleh beban listrik untuk melakukan kerja. Persamaan daya aktif adalah [11]:

$$P = V \times I \times \cos \theta \quad (2)$$

- b. Daya reaktif adalah daya yang tidak digunakan beban untuk kerja, melainkan untuk membangkitkan medan magnet di kumparan beban. Persamaan daya reaktif adalah [11]:

$$Q = V \times I \times \sin \theta \quad (3)$$

- c. Daya semu adalah penjumlahan dari daya aktif dan daya reaktif. Daya ini adalah total daya yang disediakan oleh sumber listrik untuk beban. Persamaan daya semu adalah [11]:

$$S = V \times I \quad (4)$$

Dengan,

P : Daya Aktif (W)

Q : Daya Reaktif (VAr)

S : Daya Semu (VA)

V : Tegangan RMS (V)

I : Arus RMS (A)

$\cos \theta$  : Faktor daya

### 2.2.2 Konsumsi Listrik

Energi listrik digunakan dalam kehidupan sehari-hari untuk menghidupkan peralatan elektronik. Listrik yang dikonsumsi oleh pengguna

harus dibayar ke penyedia listrik. Konsumsi listrik dapat dihitung dengan menggunakan Persamaan 5 di bawah ini [25]:

$$\text{Konsumsi Energi Listrik (kWh)} = P \times t \quad (5)$$

Keterangan:

P : Daya aktif (kW)

t : Waktu (jam)

Pembangkit Listrik Negara (PLN) merupakan penyalur listrik utama di Indonesia. Biaya listrik yang dibayarkan ke PLN disesuaikan dengan golongan tarif tenaga listrik seperti di Tabel 2.2

Tabel 2.2 Tarif Pelanggan Non-Subsidi April 2025 [26]

Golongan	Daya	Tarif per kWh
Rumah tangga kecil (R-1/TR)	900 VA	Rp 1.352
	<b>1.300 VA</b>	<b>Rp 1.444,70</b>
	2.200 VA	Rp 1.444,70
Rumah tangga menengah (R-2/TR)	3.500-5.500 VA	Rp 1.699,53
Rumah tangga besar (R-3/TR)	6.600 VA ke atas	Rp 1.699,53
Bisnis menengah (B-2/TR)	6.600 VA – 200 kVA	Rp 1.444,70
Pemerintah sedang (P-1/TR)	6.600 VA – 200 kVA	Rp 1.699,53

Biaya konsumsi listrik per hari didapatkan dengan mengonversi konsumsi daya menjadi kWh. Konsumsi daya yang sudah dikonversi di masukan ke Persamaan 6 untuk menemukan biaya konsumsi listrik.

$$\text{Estimasi biaya} = \text{Konsumsi listrik} \times \text{Biaya gol. tarif} \quad (6)$$

Golongan tarif disesuaikan dengan batas daya yang digunakan. Sektor rumah tangga terbagi menjadi 6 golongan tarif, yaitu rumah tangga kecil dengan daya 900 VA dikenakan biaya Rp1.352,00/kWh, rumah tangga kecil dengan daya 1300 VA dikenakan biaya Rp1.444,70/kWh, rumah tangga kecil dengan daya 2.200 VA dikenakan biaya Rp1.444,70/kWh, rumah tangga menengah dengan daya 3.500 sampai 5.500 VA dikenakan biaya Rp1.699,53/kWh, dan

rumah tangga menengah dengan daya 6.600 VA ke atas dikenakan biaya Rp1.699,53/kWh.

### 2.2.3 Mikrokendali NodeMCU ESP8266

NodeMCU ESP8266 merupakan mikrokendali yang mendukung pengembangan *Internet of Things* (IoT). NodeMCU terintegrasi dengan modul ESP8266 dari Espressif Systems yang memungkinkan perangkat terhubung ke jaringan internet secara langsung menggunakan protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP) [16]. Modul ini awalnya menggunakan bahasa pemrograman Lua yang memiliki logika dan susunan pemrograman seperti bahasa C namun memiliki *syntax* yang berbeda [27]. Namun seiring perkembangannya, NodeMCU juga dapat menggunakan bahasa pemrograman lain seperti C dan C++, khususnya ketika digunakan bersama Arduino IDE [4], [28].

NodeMCU ESP8266 seperti yang ditunjukkan pada Gambar 2.1 memiliki 30 pin, yang terdiri dari 17 pin GPIO, 1 pin analog, 2 pin reserved, 2 pin kontrol, dan 8 pin daya. Modul ini dilengkapi dengan beberapa antarmuka komunikasi seperti UART, I2C, dan SPI yang memungkinkan koneksi dengan berbagai jenis sensor dan aktuator, serta antarmuka I2S untuk pengolahan sinyal suara. Selain itu, terdapat ADC 10-bit untuk membaca sinyal analog, dan PWM untuk mengatur besarnya daya yang di berikan untuk suatu beban seperti LED atau motor [29].



Gambar 2.2 NodeMCU ESP8266 (Dokumen Penulis)

Papan sirkuit NodeMCU ESP8266 berukuran 49 mm x 26 mm dengan jarak pin sebesar 22,6 mm. NodeMCU ESP8266 memiliki *port micro* USB yang digunakan sebagai tempat masuknya sumber listrik [30]. Mikrokontroler ini dihidupkan dengan menggunakan kabel micro USB yang menghubungkan *port micro* USB ke komputer atau sumber tegangan.

#### 2.2.4 Sensor PZEM-004T

PZEM-004T adalah modul elektronik yang diproduksi oleh Peacefair dan dirancang khusus untuk memantau parameter konsumsi energi listrik. Sensor ini dapat mengukur tegangan (V), arus listrik (A), daya aktif (W), energi (Wh), frekuensi (Hz), serta faktor daya (*power factor*) dalam suatu rangkaian AC. Berdasarkan *datasheet* spesifikasi pengukuran PZEM-004T secara detail terdapat dalam Tabel 2.3 [12].

Tabel 2.3 Spesifikasi PZEM-004T [12]

Fungsi	Rentang Pengukuran		Mulai Pengukuran		Resolusi	Akurasi	Tampilan
	10A	100A	10A	100A			
Tegangan	80~260V				0,1V	0,5%	
Arus	0~10A	0~100A	0,01A	0,02A	0,001A	0,5%	
Daya	0~2.3 kW	0~23 kW	0,4W		0,1W	0,5%	<1000W, nilai daya memiliki 1 desimal; ≥1000W nilai daya adalah integer
Frekuensi	45~65Hz				0,1Hz	0,5%	
Energi	0~9999,99kWh				1WH	0,5%	<10kWh, satuan Wh; ≥10kWh, Satuan kWh

Fungsi	Rentang Pengukuran		Mulai Pengukuran		Resolusi	Akurasi	Tampilan
	10A	100A	10A	100A			
Faktor Daya	0,00~1,00				0,01	1%	

Modul PZEM-004T yang terdapat pada Gambar 2.3 dapat beroperasi dalam rentang suhu yang cukup luas, yaitu antara  $-20\text{ }^{\circ}\text{C}$  hingga  $+60\text{ }^{\circ}\text{C}$ , sehingga memungkinkan modul ini digunakan dalam berbagai kondisi lingkungan, baik di dalam maupun di luar ruangan. Modul ini memiliki dimensi fisik sebesar  $73,7\text{ mm} \times 30\text{ mm} \times 14,3\text{ mm}$ . PZEM-004T memiliki dua varian arus, yaitu 10 Ampere dan 100 Ampere. Khusus untuk varian PZEM-004T 100A, modul ini dilengkapi dengan kumparan transformator arus yang memiliki diameter dalam sebesar 33 mm. Transformator arus (CT) dapat mendeteksi aliran listrik tanpa perlu memutus jalur kabel utama.



Gambar 2.3 PZEM-004T 100A (dokumen penulis)

Sensor ini memiliki 8 pin, di mana 2 pin sebagai sumber daya untuk mengoperasikan sensor, 2 pin sebagai tempat komunikasi sensor dengan mikrokendali, dan 4 pin sebagai penghubung antara sensor dengan listrik yang diukur. Modul PZEM-004T ini menggunakan sensor CT non-invasif dan chip

SoC pengukuran energi SD3004. Tegangan diukur secara langsung dari terminal input, sedangkan arus diukur menggunakan sensor transformator arus (CT) yang melingkari salah satu kabel fasa. Transformator arus bekerja berdasarkan hukum induksi elektro magnetik (hukum Faraday) yang mengubah arus tinggi menjadi arus rendah. Kabel utama dimasukkan ke dalam transformator arus yang memiliki banyak jumlah lilitan. Arus yang mengalir di kabel utama menciptakan medan magnet yang menginduksi arus di kumparan transformator arus, sehingga arus di CT berbanding lurus dengan arus utama sesuai dengan rasio lilitan [31].

### 2.2.5 Relay

*Relay* seperti pada Gambar 2.4 merupakan komponen elektronik yang digunakan sebagai sakelar elektronik untuk mengontrol besarnya beban pada tegangan yang masuk ke komponen atau peralatan elektronik [32]. Prinsip kerja *relay* didasarkan pada elektromagnetisme, di mana arus listrik yang mengalir melalui kumparan menghasilkan medan magnet yang menarik tuas logam, sehingga tuas listrik dapat berpindah dari *normally closed* ke *normally open*. *Relay* umumnya digunakan untuk mengendalikan beban bertegangan tinggi dengan sinyal bertegangan rendah.



Gambar 2.4 2-Channel Relay (dokumen penulis)

*Relay* memiliki beberapa pin yang masing-masing memiliki fungsi sebagaimana dipaparkan pada Tabel 2.4 [15]:

Tabel 2.4 Fungsi Relay

Pin	Fungsi
VCC	Input tegangan positif (biasanya 5V) untuk mengaktifkan modul <i>relay</i>
GND	Kutub negatif dari sumber daya
IN	Mengaktifkan atau menonaktifkan masing-masing <i>relay</i> melalui sinyal dari mikrokendali. Ketika diberikan logika rendah (LOW), <i>relay</i> akan aktif dan menghubungkan sirkuit; sebaliknya, jika diberikan logika tinggi (HIGH), <i>relay</i> akan nonaktif dan memutuskan sirkuit.
Common (COM)	Menghubungkan <i>relay</i> dengan peralatan elektronik
Normally Closed (NC)	Membuka sirkuit agar aliran listrik berhenti
Normally Open (NO)	Menutup sirkuit agar aliran listrik mengalir

### 2.2.6 ThingSpeak

*ThingSpeak* adalah platform *Internet of Things* (IoT) bersifat *open source* yang berbasis *Application Programming Interface* (API) atau Antarmuka Pemrograman Aplikasi [33]. API berfungsi menghubungkan dua komponen perangkat lunak untuk saling berkomunikasi menggunakan serangkaian protokol [34]. Dalam operasionalnya, *ThingSpeak* menggunakan protokol HTTP untuk menerima dan mengirimkan data melalui jaringan internet ataupun *Local Area Network* (LAN) [33].

Platform *ThingSpeak* menyediakan layanan *cloud* untuk menyimpan data dari perangkat IoT secara *real-time*. Selain itu, platform ini menyediakan *dashboard* gratis yang dapat digunakan untuk memvisualisasikan data dalam bentuk grafik, sehingga memudahkan pengguna memantau konsumsi listrik. *ThingSpeak* juga memungkinkan pengguna untuk melakukan analisis data lebih lanjut menggunakan MATLAB yang terintegrasi langsung dalam platform ini [35]. *ThingSpeak* menyediakan layanan gratis dan berbayar, untuk layanan gratis jumlah *channel* terbatas pada 4 *channel* dan jumlah pesan masuk tiap harinya terbatas 8.200 pesan. Sedangkan, layanan berbayar dapat membuat hingga 10 *channel* dan setiap harinya dapat menerima hingga 90.000 pesan [36].

Fitur utama dari *ThingSpeak* adalah *channel* atau saluran yang menyimpan data dari perangkat IoT. Saluran ini terdiri dari beberapa elemen penting yang berfungsi untuk penyimpanan dan visualisasi data, berikut elemen-elemen di saluran *ThingSpeak* [35]:

- 1) Delapan (8) *field* : Berfungsi menyimpan data dari sensor atau dari perangkat IoT.
- 2) Tiga (3) saluran lokasi : Berfungsi menyimpan garis lintang, garis bujur, dan ketinggian dari suatu lokasi. Ini dapat digunakan untuk melacak lokasi perangkat yang bergerak.
- 3) Satu (1) saluran status : Pesan singkat yang menjelaskan data yang tersimpan di saluran.



Gambar 2.5 *ThingSpeak* [28]

### 2.2.7 Telegram Bot

Telegram Bot API merupakan antarmuka pemrograman aplikasi yang disediakan oleh Telegram untuk memungkinkan pengembang membuat dan mengintegrasikan bot ke dalam sistem tertentu. Telegram Bot sendiri adalah akun khusus yang dapat menjalankan perintah secara otomatis dan tidak memerlukan nomor telepon tambahan untuk proses pembuatan akun. Bot ini berfungsi sebagai perantara komunikasi antara pengguna dan sistem yang berjalan di sisi server [37]. Telegram Bot mampu menerima dan merespons berbagai jenis input dari pengguna, seperti pesan teks, file, lokasi, stiker, pesan suara, dan bahkan objek interaktif seperti animasi dadu. Selain fungsi dasar

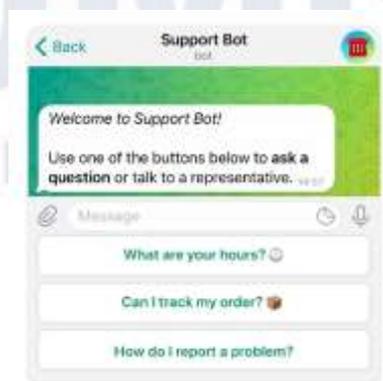
tersebut, Telegram menyediakan berbagai fitur interaktif yang memudahkan pengembangan antarmuka pengguna yang fleksibel dan responsif, di antaranya [37]:

- 1) Perintah (*Commands*): Perintah merupakan instruksi dalam bentuk teks yang digunakan untuk mengarahkan bot agar menjalankan fungsi tertentu. Perintah ini diawali dengan simbol garis miring ("/") dan berisi maksimal 32 karakter yang mencakup huruf latin, angka, dan garis bawah



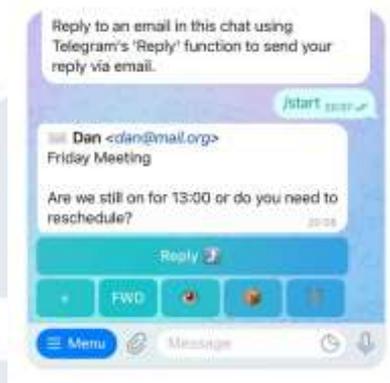
Gambar 2.6 Perintah Telegram Bot [37]

- 2) Keyboard Kustom (Custom Keyboard): Fitur ini menggantikan keyboard default pengguna dengan pilihan-pilihan jawaban yang telah ditentukan sebelumnya. Ketika pengguna memilih salah satu opsi, nilai tersebut akan dikirim ke bot dalam bentuk teks untuk kemudian diproses. Pendekatan ini menyederhanakan interaksi pengguna dan meningkatkan efisiensi komunikasi.



Gambar 2.7 Keyboard Custom Telegram Bot [37]

- 3) Tombol Interaktif (Inline Button): Berbeda dengan keyboard kustom, tombol ini tidak mengirim pesan secara langsung ke ruang obrolan. Sebaliknya, tombol ini dapat menjalankan perintah tertentu di latar belakang (callback button) atau mengarahkan pengguna ke URL eksternal. Tombol interaktif ini biasanya ditempatkan di bawah pesan yang berkaitan.



Gambar 2.8 Tombol Interaktif Telegram Bot [37]

- 4) Tombol Menu (*Menu Button*): Terletak di sebelah kolom input pesan, tombol ini membuka daftar menu yang menampilkan perintah-perintah bot beserta deskripsi singkatnya. Fitur ini memudahkan pengguna untuk mengenali dan menggunakan berbagai fungsi yang tersedia pada bot.

Dengan fitur-fitur tersebut, Telegram Bot API menjadi salah satu solusi komunikasi otomatis yang banyak digunakan dalam pengembangan sistem informasi, layanan pelanggan, dan integrasi sistem berbasis *Internet of Things* (IoT).

### 2.2.8 Akurasi Pengukuran

Pengukuran konsumsi listrik menggunakan sistem akuisisi data memerlukan pengujian untuk mengetahui sejauh mana hasil pengukuran mendekati nilai sebenarnya. Salah satu parameter yang umum digunakan untuk mengevaluasi tingkat keakuratan hasil pengukuran adalah persentase akurasi. Persentase akurasi menggambarkan tingkat kedekatan nilai hasil pengukuran

terhadap nilai referensi atau nilai ideal yang dianggap benar. Nilai akurasi yang tinggi menunjukkan bahwa sistem pengukuran bekerja dengan baik dan dapat dipercaya.

Parameter ini digunakan untuk menilai performa sistem, terutama dalam konteks sistem akuisisi data berbasis sensor. Dalam tugas akhir ini, persentase akurasi digunakan untuk mengevaluasi ketepatan pembacaan sensor PZEM-004T dalam mengukur tegangan, arus, dan faktor daya dibandingkan dengan alat ukur standar. Persentase akurasi dapat dihitung dengan rumus sebagai berikut [38]:

$$\%akurasi = (1 - Relatif\ Error) \times 100\% \quad (7)$$

$$\%akurasi = \left( 1 - \frac{|Nilai\ terukur - Nilai\ referensi|}{Nilai\ referensi} \right) \times 100\% \quad (8)$$

### 2.2.9 Standar Deviasi dan *Root Mean Squared Error*

Standar deviasi adalah nilai statistik yang digunakan untuk mengukur seberapa besar penyebaran atau variasi dari suatu kumpulan data terhadap nilai rata-ratanya. Semakin kecil nilai standar deviasi, maka data semakin dekat dengan nilai rata-rata, sedangkan nilai standar deviasi yang besar menunjukkan data tersebar jauh dari nilai rata-ratanya. Rumus standar deviasi adalah sebagai berikut [39]:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (9)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (10)$$

Keterangan:

- $\sigma$  = Standar deviasi
- $x_i$  = Nilai data ke - i
- $\bar{x}$  = Rata-rata data sampel

- $n$  = Jumlah data

*Root Mean Squared Error* (RMSE) adalah ukuran yang sering digunakan untuk mengevaluasi tingkat kesalahan dalam model prediksi atau perbandingan data hasil pengukuran dengan data acuan. RMSE adalah rata-rata dari kuadrat selisih nilai pengukuran dan referensi yang kemudian diakarkan. RMSE menunjukkan seberapa jauh data hasil pengukuran menyimpang dari nilai referensi dalam satuan yang sama seperti data aslinya. Rumus RMSE adalah sebagai berikut [40]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{n}} \quad (11)$$

Keterangan:

- $x_i$  = Hasil pengukuran
- $y_i$  = Nilai referensi
- $n$  = Jumlah data

