

SISTEM MONITORING UNTUK PENGGUNAAN DAYA LISTRIK PADA SMARTHOME WITH METHOD FUZZY LOGIC BERBASIS IOT

¹Alfonsius Yohandrik Bi Dawe_1, ²Puput Dani Prasetyo Adi_2, ³Wahyu Dirgantara_3

1 Program Studi Teknik Elektro, Universitas Merdeka Malang

2 Simpang Simpang Megamendung, Kec. Sukun, Kota Malang, Jawa Timur 65146

e-mail: ronzydawe319@gmail.com | Email Penulis_1, ² Email Penulis_2

Abstract : *Smart Home is a smart home concept that has a remote control system based on the internet of things, in order to provide convenience and security for owners to control household electronic equipment such as lights, dispensers and fans, not only controlling the house but home owners can monitor power, voltage, current and electricity bills in real-time as well as the status of the electrical load used in a lit state or not. And users can also limit the use of power to use to prevent overloads that have an impact on rising bills or electricity usage costs. As well as monitoring lighting automatically on the terrace of the house and monitoring room temperature automatically using the fuzzy logic method. As well as monitoring bathroom lights by detecting the presence and absence of movement. The switch system on electrical devices can be replaced using the ESP8266 microcontroller so that it can be connected to Android with the Openhabian application installed. This control system can also be carried out anywhere and from anywhere without having area restrictions as long as the user can connect to the internet.*

Keywords : *Android, ESP8266, Raspberry Pi, Smarthome, Internet Of Things*

Abstrak: *Smart Home* adalah sebuah konsep rumah pintar yang memiliki sistem pengendali jarak jauh berbasis *internet of things*, guna memberikan kenyamanan dan keamanan bagi pemilik untuk mengontrol peralatan elektronik rumah tangga seperti lampu, dispenser dan kipas, bukan hanya mengontrol rumah saja tapi pemilik rumah bisa memonitoring daya, tegangan, arus dan tagihan listrik secara *real-time* serta status beban listrik terpakai dalam keadaan menyala atau tidak. Serta pengguna juga bisa membatasi penggunaan daya menggunakan untuk mencegah beban berlebih yang berdampak pada naiknya tagihan atau biaya penggunaan listrik. Serta memonitoring pencahayaan secara otomatis pada teras rumah dan memonitoring suhu ruangan secara otomatis menggunakan metode *fuzzy logic*. Serta memonitoring lampu kamar mandi dengan mendeteksi ada dan tidak adanya pergerakan. Sistem saklar pada perangkat listrik dapat di gantikan menggunakan mikrokontroler ESP8266 sehingga dapat terhubung ke *Android* yang telah terinstal aplikasi *Openhabian*. Sistem kendali ini juga

dapat dilakukan di mana saja dan dari mana saja tanpa memiliki batasan wilayah selama pengguna dapat terhubung dengan internet.

Kata kunci : *Android, ESP8266, Raspberry Pi, Smarthome, Internet Of Things*

I. Pendahuluan

Energi listrik telah menjadi salah satu kebutuhan primer manusia karena semua lingkungan membutuhkan energi listrik seperti di rumah, tempat kerja, pabrik, dan lain-lain. Kebutuhan listrik di disediakan oleh perusahaan Negara di mana setiap konsumen harus membayar biaya sesuai dengan seberapa banyak listrik yang digunakan. Perhitungan pemakaian daya listrik dihitung berdasarkan besar pemakaian daya dikalikan waktu dengan satuan kilo watt per jam, yang dapat dilihat pada KWH meter.

Sistem monitoring dan pengambilan keputusan merupakan sebuah bentuk monitoring dan pengambilan keputusan berkaitan dengan penggunaan daya terhadap beban listrik rumah tangga, yang semuanya mampu dikendalikan secara langsung sesuai keinginan oleh pemilik. Sistem *Smarthome* saat ini ada yang menggunakan instalasi kabel dan tanpa kabel. Sehingga pemanfaatan dan implementasi untuk instalasi secara nirkabel direalisasikan.

Fuzzy logic digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), pemengambil keputusan pemakaian daya listrik dan pengatur intensitas cahaya lampu yang diekspresikan dengan gelap, redup dan terang. Dan logika fuzzy menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah.

Internet of things pemakaian daya diupayakan untuk menghemat dan membatasi penggunaan energi listrik harus dilaksanakan dengan mengirim notifikasi ke android bilamana pemakaian daya sudah besar. Untuk itulah pada penelitian tugas akhir ini akan di buat alat system monitoring dan

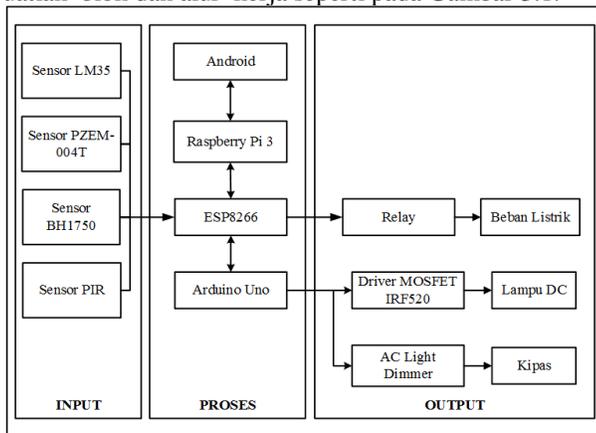
pengambil keputusan penggunaan daya listrik yang dapat memberikan informasi hasil monitoring penggunaan daya keseluruhan beban listrik dengan sistem *internet of things* yang di akses pada aplikasi *openhajian* di ponsel Android, agar nantinya dapat di amati oleh pengguna secara *realtime*, sehingga pemakaian energi listrik dapat di monitoring kapan saja dan di mana saja mengaplikasinya. daya listrik yang dapat memberikan informasi hasil monitoring penggunaan daya keseluruhan beban listrik dengan sistem *internet of things* yang di akses pada aplikasi.

II. Metode Penelitian

Metode adalah sebuah cara atau rangkaian yang digunakan untuk menyelesaikan masalah. Metode yang digunakan dalam penelitian ini adalah *Fuzzy Logic* pengambil keputusan penggunaan daya listrik sebagaimana penghematan dalam penggunaan daya listrik.

A. Blok Diagram Alat

Dibutuhkan beberapa komponen elektronika dan *device* penunjang dalam perancangan perangkat keras atau *hardware* ini agar sistem dapat bekerja dan berjalan dengan baik sesuai dengan fungsinya. Maka dibuatlah blok dan alur kerja seperti pada Gambar 3.1.



Gambar 1. Blok Diagram Alat

Pada gambar 1 menunjukkan blok diagram kerja dari alat di mulai dari inputnya: Sensor PZEM-004T, sensor yang akan membaca berupa: daya listrik, energi total, arus dan tegangan. Sensor PIR, sensor yang akan membaca ada pergerakan. Sensor BH1750, sensor dengan parameter lux guna untuk membaca intensitas cahaya matahari. Sensor LM35, sensor yang akan membaca suhu ruangan.

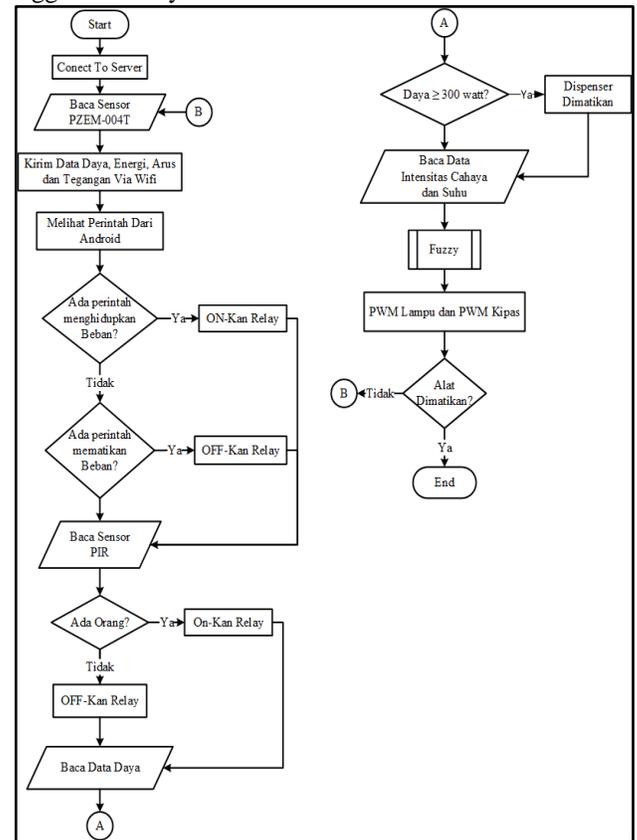
Pada bagian proses memerlukan beberapa mikrokontroler mendukung dalam pembuatan alat antara lain: Android, Sebuah handphone yang akan mengakses data dari server

Raspberry Pi 3 dengan aplikasi Openhabian. Raspberry Pi 3, server utama yang secara real-time mengambil data dari ESP8266 via wiffi sekaligus dalam pembuatan aplikasi Openhabian yang dapat di akses pada layar kaca android. ESP8266, mikrokontroler yang terhubung langsung dengan sensor PZEM-004T, sensor BH1750, sensor PIR, relay dan mosfet IRF520. Sekaligus pengirim data ke raspberry via wifi. Arduino Uno, mikrokontroler yang memogram sistem fuzzyfikasi input intensitas cahaya ke output lampu dc dan input suhu ke output kipas.

Dan di bagian outputnya: Mosfet IRF520 digunakan untuk mengatur intensitas cahaya lampu dc. Ac Light Dimmer digunakan untuk mengatur PWM kipas.

B. Flowchart System

Pada perancangan perangkat lunak berikut bertujuan untuk mengontrol Raspberry Pi untuk memproses data dari Sensor PZEM-004T, Sensor PIR, Sensor Cahaya BH1750 dan mengontrol relay melalui android.



Gambar 2. Flowchart System Keseluruhan

Pada Gambar 2 menunjukkan *Flowchart* Perangkat Lunak di mulai dari proses pembacaan sensor PZEM-004T kemudian dapat di baca berupa daya listrik, energi total, arus, tegangan . Nilai yang di dapat secara otomatis di kirim ke server dan di ekspor melalui jaringan Wifi pengguna sehingga user dapat mengaksesnya dengan menggunakan android melalui aplikasi Openhabian. User dapat memerikan perintah menghidupkan atau mematikan relay yang terhubung ke beban listrik terpakai melalui android.

Selain itu juga sensor PIR gunanya mendeteksi adanya pergerakan orang di dalam ruangan. Bilamana terdeteksi ada pergerakan orang di dalam ruangan maka relaynya akan menghubungkan aliran listrik ke lampu dan timer 60 detik dan jika tidak adanya pergerakan orang di dalam ruangan timer akan menghitung mundur 60-0 detik secara otomatis relay memutuskan aliran listrik ke lampu.

Baca data intensitas cahaya merupakan pembacaan intensitas cahaya pada sensor BH1750 dengan 3 keanggotaan input, yaitu: gelap, normal, dan terang yang berpengaruh pada output PWM lampu. Baca data intensitas cahaya tersebut difuzzyfikasi menjadi 3 keanggotaan output, yaitu: padam, redup, dan terang yang berpengaruh pada PWM lampu. Baca data suhu merupakan pembacaan suhu pada sensor LM35 yang dengan 5 keanggotaan input, yaitu: dingin, sejuk, normal, panas dan sangat panas yang berpengaruh pada PWM kipas. Baca data suhu tersebut difuzzyfikasi menjadi 5 keanggotaan output, yaitu: padam, pelan, normal, cepat dan sangat cepat yang berpengaruh pada PWM kipas. Baca data daya merupakan hasil dari pembacaan sensor PZEM-004T dengan pembacaan keseluruhan beban mencapai 400 watt. Dengan rule fuzzy jika pemakaian daya ≥ 300 watt diberikan notifikasi "Hati2" pada aplikasi Openhabian sekaligus otomatis dalam memutuskan relay beban dispenser setelah daya ≤ 300 watt notifikasi berubah "Aman".

III. Hasil Dan Analisa

Setelah perancangan dan pembuatan alat selesai, pada bab ini akan berisi tentang pengujian sistem pada alat yang di rangkai sesuai dengan perancangan pada bab sebelumnya untuk mengetahui apakah alat berjalan sesuai yang direncanakan. Pengujian ini dilakukan untuk membandingkan hasil perancangan teoritis dengan hasil percobaan. Dari hasil pengujian dapat diketahui apakah alat telah bekerja sesuai dengan yang diinginkan pada perancangan alat di bab sebelumnya.

I. Pengujian Pengambil Keputusan *Switch ON* Dan *Switch OFF* Secara Manual Pada Beban Listrik

Pada perancangan ini penulis mencoba merancang tampilan aplikasi *Openhabian* agar dapat mengendalikan beban listrik terpakai dengan jarak jauh. Langkah awal yang dilakukan membuat menu *Things* dengan keterangan *Saklar Manual* pada *Home* aplikasi *Openhabian*. Untuk melihat hasil tampilannya dapat di lihat pada Gambar 3.



Gambar 3. Tampilan Menu *Things* Aplikasi *Openhabian*

Pada gambar di atas adalah tampilan keterangan *Saklar Manual* pada menu *Things* aplikasi *openhabian* sebagai menu utama yang mengendalikan *switch ON* dan *switch OFF* pada channel *Things* yang akan di buat. Setelah masuk ke menu *Things* di buat juga *channel* dengan *type channel* berupa *switch ON* dan *switch OFF* yang digunakan menghubungkan dan memutuskan aliran listrik ke beban listrik terpakai. Tampilan keterangan *channel* pada menu *Things* aplikasi *Openhabian* dapat di lihat pada Gambar 4.



Gambar 4. Tampilan *Channel* Pada Menu *Things* saat *switch ON* dan Saat *switch OFF*

Pada gambar di atas terdapat *switch* untuk mengambil keputusan. *Switch ON* untuk menghubungkan aliran listrik ke beban listrik terpakai dan *switch OFF* untuk memutuskan aliran listrik ke beban terpakai. Berikut adalah melihat hasil respon *switch ON-OFF* dari keterangan tiap-tiap *channel* pada menu *Things* aplikasi *Openhabian* di layar kaca android yang sudah terhubung melalui jaringan internet.



Gambar 5. Hasil Respon *Switch ON* Dan *Switch OFF* Pada Beban Listrik

Pada gambar di atas melihat langsung bagaimana melihat respon langsung *switch ON* dan *OFF* terhadap beban listrik terpakai pada maket. Berikut tabel 1. hasil pengujian *switch on-off* beban listrik dan waktu deteksi.

Tabel 1. Hasil Pengujian *Switch ON-OFF* Beban Listrik Dan Waktu Deteksi

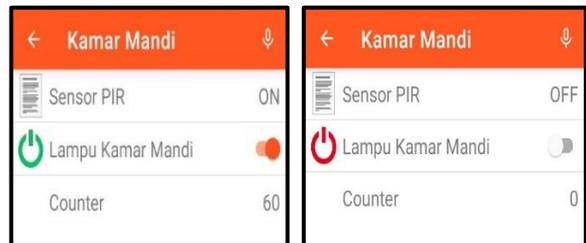
| No | Sample Beban | Hasil Pengujian <i>Switch ON</i> | | | Hasil Pengujian <i>Switch OFF</i> | | |
|-----------|----------------------|----------------------------------|--------|--------------|-----------------------------------|--------|--------------|
| | | Time (ms) | Sukses | Tidak Sukses | Time (ms) | Sukses | Tidak Sukses |
| 1 | Lampu Ruang Tamu | 0,67 | ✓ | - | 0,63 | ✓ | - |
| 2 | Lampu Ruang Keluarga | 0,64 | ✓ | - | 0,66 | ✓ | - |
| 3 | Lampu Kamar Belakang | 0,64 | ✓ | - | 0,65 | ✓ | - |
| 4 | Lampu Dapur | 0,66 | ✓ | - | 0,62 | ✓ | - |
| 5 | Lampu Kamar Depan | 0,68 | ✓ | - | 0,64 | ✓ | - |
| 6 | Dispenser | 0,64 | ✓ | - | 0,66 | ✓ | - |
| 7 | Saklar Utama | 0,62 | ✓ | - | 0,62 | ✓ | - |
| Rata-rata | | 0,65 | | | 0,64 | | |

Pada Tabel 1. dapat di lihat bagaimana hasil pengujian pengambil keputusan *switch ON* dan *switch OFF* dengan

menggunakan android pada 7 *sample* beban listrik terpakai. Didapatkan rata-rata *error* untuk *switch ON* sebesar 0,65 dan rata-rata *switch OFF* sebesar 0,64.

II. Pengujian Pengambil Keputusan Pada Sensor PIR

Pada pengujian ini melihat hasil pembacaan sensor PIR terhadap ada pergerakan dan tidak adanya pergerakan di dalam kamar mandi yang nantinya akan berpengaruh pada lampu kamar mandi tersebut. Sensor PIR di program dengan membaca setiap satu kali pergerakan, maka lampunya secara otomatis keadaannya terang dengan waktu di program sebanyak 1 menit (60 detik) dan secara *realtime* dalam waktu 60 detik tersebut akan menghitung mundur sampai 0, jika sudah mencapai 0 lampu secara otomatis akan padam. Berikut gambar Pengujian pengambil keputusan pada sensor PIR.



Gambar 6. Pengujian Pengambil Keputusan Pada Sensor PIR

Dari hasil pengujian pengambil keputusan pada sensor PIR yang ditampilkan pada aplikasi *Openhabian* dengan *Items* lampu kamar mandi. Saat sensor PIR mendeteksi ada pergerakan tangan, maka *switch* menjadi *ON* dan *counter* 60 lampu secara otomatis dalam keadaan terang. Ketika sensor PIR tidak mendeteksi ada pergerakan *switch* menjadi *OFF* dan *counter* akan menghitung mundur dari 60 sampai 0, ketika *counter* 0 lampu kamar mandi secara otomatis dalam keadaan gelap.

a. Pengujian Pembacaan Pada Sensor PZEM-004T Dengan Pengukuran Kwh Modul

Pada pengujian ini akan melihat pembacaan sensor PZEM-004T pada aplikasi *Openhabian* dengan keseluruhan *sample* beban listrik terpakai. Data yang akan di ukur berupa: daya, arus dan tegangan. Data tersebut akan dibandingkan tingkat *error* dengan menggunakan Kwh modul.

Langkah awal dalam melakukan pengujian pembacaan sensor PZEM-004T adalah membuat keterangan *Things* kWh dan membuat *Items* untuk memasukan data pembacaan sensor PZEM-004T ke menu *Things* dengan keterangan kWh. Data yang ditampilkan berupa: Daya(watt), Arus(ampere), Tegangan(volt) . Hasil tampilan pembacaan sensor PZEM-004T dan Kwh modul dapat di tunjukan pada gambar berikut ini.

| ← kWh | |
|----------------|-------|
| Arus(ampere) | 0.136 |
| Tegangan(volt) | 220 |
| Daya(watt) | 12 |

Gambar 7. Tampilan Pembacaan PZEM-004T

Pada gambar di atas adalah pembacaan sensor PZEM-004T yang di tampilkan pada aplikasi *Openhabian* yang akan membandingkan datanya dengan kwh modul.

Pengujian ini dilakukan untuk mengukur dari 9 *sample* beban terpakai antara lain: lampu teras, lampu ruang tamu, lampu kamar depan, lampu ruang keluarga, lampu kamar belakang, dispenser, kipas angin, lampu kamar mandi dan lampu dapur. Data yang di baca sensor PZEM-004T akan dibandingkan tingkat akurasi dengan hasil pengukuran kWh modul. Berikut adalah Tabel 2. hasil pengukuran daya.

Tabel 2. Hasil Pengujian Pengukuran Daya

| No | Keterangan Beban Aplikasi Openhabian | Pengukuran Daya | | Error(%) |
|-----------|--------------------------------------|------------------------|-----------------|----------|
| | | Sensor PZEM-004T(Watt) | Modul Kwh(Watt) | |
| 1 | Lampu Teras | 12 | 12 | 0% |
| 2 | Lampu Ruang Tamu | 12 | 12 | 0% |
| 3 | Lampu kamar Depan | 11 | 11 | 0% |
| 4 | Lamp Ruang Keluarga | 12 | 12 | 0% |
| 5 | Lampu Kamar Belakang | 11 | 11 | 0% |
| 6 | Kipas Angin | 14 | 14 | 0% |
| 7 | Dispenser | 361 | 361 | 0% |
| 8 | Lampu Kamar Mandi | 12 | 12 | 0% |
| 9 | Lampu Dapur | 12 | 11 | 0.09% |
| Rata-rata | | | | 0.01% |

Pada Tabel 3. merupakan hasil *error %* pengukuran daya pada sensor PZEM-004T dan modul Kwh. Untuk mencari perhitungan Error % adalah:

$$error = \frac{Data\ Alat\ prototipe - Data\ Alat\ ukur}{Data\ Alat\ ukur} \times 100\% \quad (1)$$

Hasil pengujian *error %* di dapat dari nilai asli pembacaan daya pada sensor PZEM-004T dan nilai terukur dengan menggunakan kwh modul yang di ukur untuk melihat *error* pengukuran daya. Dan nilai rata-rata di dapat dari hasil pembagian antara jumlah nilai error keseluruhan dibagi dengan banyaknya percobaan yang di ambil dan didapatkan nilai rata-rata error sebesar 0.01%. Berikut adalah Tabel 3. hasil pengukuran tegangan.

Tabel 3. Hasil Pengukuran Tegangan

| No | Keterangan Beban Aplikasi Openhabian | Pengukuran Tegangan | | Error (%) |
|-----------|--------------------------------------|------------------------|-----------------|-----------|
| | | Sensor PZEM-004T(volt) | Modul Kwh(Volt) | |
| 1 | Lampu Teras | 219 | 219 | 0% |
| 2 | Lampu Ruang Tamu | 217 | 217 | 0% |
| 3 | Lampu kamar Depan | 220 | 220 | 0% |
| 4 | Lampu Ruang Keluarga | 221 | 220 | 0.45% |
| 5 | Lampu Kamar Belakang | 217 | 217 | 0% |
| 6 | Kipas Angin | 210 | 208 | 0.96% |
| 7 | Kipas Dispenser | 219 | 219 | 0% |
| 8 | Lampu Kamar Mandi | 216 | 214 | 0.93% |
| 9 | Lampu Dapur | 219 | 217 | 0.92% |
| Rata-rata | | | | 0.36% |

Pada Tabel 3. merupakan hasil *error%* pengukuran tegangan pada sensor PZEM-004T dan modul Kwh. Untuk mencari perhitungan *error%* adalah:

$$\%error = \frac{nilai\ asli\ pembacaan - nilai\ terukur}{nilai\ terukur} \times 100 \quad (2)$$

Hasil pengujian *error* di dapat dari nilai asli pembacaan tegangan pada sensor PZEM-004T dan nilai terukur dengan menggunakan kwh modul yang di ukur untuk melihat tingkat kesalahan *error* pengukuran tegangan. Dan nilai rata-rata di dapat dari hasil pembagian antara jumlah nilai error keseluruhan dibagi dengan banyaknya percobaan yang di ambil dan didapatkan nilai rata-rata error sebesar 0.36%. Berikut adalah Tabel 4. hasil pengujian pengukuran arus.

Tabel 4. Hasil Pengukuran Arus

| No | Keterangan Beban Aplikasi Openhabian | Pengukuran Arus | | Error (%) |
|-----------|--------------------------------------|--------------------------|-------------------|-----------|
| | | Sensor PZEM-004T(Ampere) | Modul Kwh(Ampere) | |
| 1 | Lampu Teras | 0,26 | 0,25 | 4% |
| 2 | Lampu Ruang Tamu | 0,26 | 0,25 | 4% |
| 3 | Lampu kamar Depan | 0,26 | 0,25 | 4% |
| 4 | Lampu Ruang Keluarga | 0,25 | 0,25 | 0% |
| 5 | Kipas Angin | 1,02 | 1,02 | 0% |
| 6 | Dispenser | 1,111 | 1,111 | 0% |
| 7 | Lampu Kamar Mandi | 0,26 | 0,25 | 4% |
| 8 | Lampu Dapur | 0,26 | 0,25 | 4% |
| Rata-rata | | | | 2,5% |

Pada tabel hasil pengukuran arus pada sensor PZEM-004T dan modul Kwh untuk mencari selisih kesalahan perhitungan adalah:

$$\%error = \frac{nilai\ asli - nilai\ terukur}{nilai\ terukur} \times 100\% \quad (3)$$

Hasil pengujian akurasi di dapat dari nilai asli pembacaan arus pada sensor PZEM-004T dan nilai terukur dengan

menggunakan kwh modul yang di ukur untuk melihat tingkat kesalahan atau *error* pada pengukuran arus. Dan nilai rata-rata di dapat dari hasil pembagian antara jumlah nilai error keseluruhan dibagi dengan banyaknya percobaan yang di ambil dan didapatkan nilai rata-rata error sebesar 2,5%.

III. Pengujian Merubah KWH Menjadi Rupiah

Pada pengujian ini dilakukan bertujuan untuk merubah beban KWH yang digunakan menjadi rupiah. Studi kasus yang di ambil adalah daya 1300 VA dengan golongan konsumen rumah tangga kecil. Dengan biaya yang di tetapkan adalah Rp. 1.445/kWh.

Biaya per Kwh = Rp. 1.445, misalnya energi total = 1kwh. Biaya = Rp. 1 x 1.445 = Rp. 1.445. Jadi diperoleh biaya listrik dikeluarkan adalah Rp. 1.445. Hitungan biaya yang sudah tertera di program pada keterangan *Things* aplikasi *Openhabian*. Yang akan ditampilkan dalam bentuk *number* pada *Things* dengan keterangan *Items* kWh. Berikut tampilan keterangan *Item* kWh dapat ditunjukkan pada Gambar 8.



Gambar 8. Pengujian Merubah KWH menjadi Rupiah

Pada Gambar 8. pengujian konversi biaya hari pertama dengan energi total yang didapatkan 0,835 kWh dan dikonversikan ke biaya listrik, hasil yang didapatkan adalah Rp. 1.206,575.

Merubah kwh ke rupiah akan di uji selama 30 hari untuk melihat hasil seberapa besar biaya yang di keluarkan dari total energi listrik terpakai. Berikut tabel pengujian mengubah Kwh menjadi Rupiah selama 30 hari.

Tabel 5. Pengujian Merubah KWH Menjadi Rupiah 30 Hari

| Hari ke | Energi(kWh) | Biaya(Rupiah) |
|---------|-------------|---------------|
| 1 | 0,835 | 1.206,575 |
| 2 | 1,821 | 2.631,345 |
| 3 | 2,294 | 3.314,830 |
| 4 | 2,926 | 4.228,070 |
| 5 | 3,474 | 5.019,930 |
| 6 | 4,164 | 6.016,980 |
| 7 | 5,12 | 7.398,40 |
| 8 | 5,542 | 8.808,190 |

| Hari ke | Energi(kWh) | Biaya(Rupiah) |
|---------|-------------|---------------|
| 9 | 6,244 | 9.022,580 |
| 10 | 6,937 | 10.023,965 |
| 11 | 7,763 | 11.217,545 |
| 12 | 8,337 | 12.046,965 |
| 13 | 8,997 | 13.000,665 |
| 14 | 9,715 | 14.038,175 |
| 15 | 10,554 | 15.250,530 |
| 16 | 11,147 | 16.107,415 |
| 17 | 11,806 | 17.059,670 |
| 18 | 12,618 | 18.233,010 |
| 19 | 13,213 | 19.092,785 |
| 20 | 13,868 | 20.039,260 |
| 21 | 14,564 | 21.044,980 |
| 22 | 15,255 | 22.043,475 |
| 23 | 15,986 | 23.099,770 |
| 24 | 16,779 | 24.245,655 |
| 25 | 17,372 | 25.102,540 |
| 26 | 18,014 | 26.030,230 |
| 27 | 18,714 | 27.041,730 |
| 28 | 19,467 | 28.129,815 |
| 29 | 20,049 | 29.035,830 |
| 30 | 20,811 | 30.071,895 |

Tabel 5. merupakan pengujian merubah kwh menjadi Rupiah yang di uji untuk memastikan berapa besar biaya yang di keluarkan berdasarkan energi total yang terpakai. Pada pemakaian hari 30 hari terbaca nilai kwh yang terpakai adalah 20,811 dan beban biaya yang di keluarkan sebesar Rp. 30.071,895.

IV. Pengujian Notifikasi Daya Melebihi Batas

Pada pengujian ini menerapkan *Rule* untuk membatasi jika terjadinya adanya pemakaian daya yang lebih. Dari hasil pengukuran daya keseluruhan beban terpakai mencapai 410 watt, maka di buat Rule Then If daya ≥ 300 watt maka di buat aksi untuk memutuskan satu beban terpakai. Langkah pertama yang dilakukan dengan program raspberry pi dengan bahasa *GUI* menggunakan *software PuTTY* agar bisa dikendalikan langsung pada alat yang diteliti.

Program notifikasi daya melebihi batas dapat menampilkan notifikasi pada aplikasi *Openhabian* dengan keterangan *Things* kWh. Bilamana pemakaian daya ≥ 300 watt memberi notifikasi “Hati” sekaligus aksi untuk memutuskan relay yang terhubung ke beban dispenser. Dan bilamana pemakaian daya ≤ 300 watt memberi notifikasi “Aman”.

Untuk melihat hasil kerja program tersebut dapat ditunjukkan pada Gambar 9.

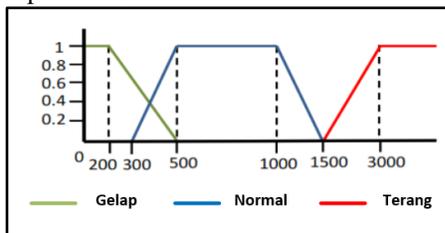
| Parameter | Value | Notification |
|----------------|-----------|--------------|
| Arus(ampere) | 0.136 | Aman |
| Tegangan(volt) | 220 | Aman |
| Daya(watt) | 94 | Aman |
| Energi(kWh) | 21.784 | Aman |
| Biaya(Rp) | 31477.880 | Aman |
| Arus(ampere) | 1.78 | Hati2 |
| Tegangan(volt) | 220 | Hati2 |
| Daya(watt) | 410 | Hati2 |
| Energi(kWh) | 21.821 | Hati2 |
| Biaya(Rp) | 31531.345 | Hati2 |
| Notifikasi | | Hati2 |

Gambar 9 Tampilan Hasil Pengujian Notifikasi Daya Melebihi Batas

Pada Gambar 9 daya listrik terbaca 94 watt dan notifikasi masih keadaan “Aman”, karena pemakaian daya masih ≤ 94 watt dan tidak adanya aksi memutuskan relay ke beban dispenser. Pemakaian daya listrik terbaca 410 watt dan notifikasi berubah menjadi “Hati” artinya pemakaian daya ≥ 300 watt maka adanya aksi dalam memutuskan relay pada dispenser.

V. Pengujian Sensor BH1750 Pada Intensitas Cahaya

Pada pengujian ini dilakukan input cahaya terhadap 5 fungsi keanggotaan yaitu: sangat gelap, gelap, remang, terang dan sangat terang. Masing-masing mempunyai domain range atau batasan nilai yang akan diaplikasikan pada sensor BH1750, mikrokontroler Arduino dan ESP8266. Tampilan *domain range* atau batasan nilai *input* cahaya dapat ditunjukkan pada Gambar 10.



Gambar 10. Keanggotaan Input Cahaya

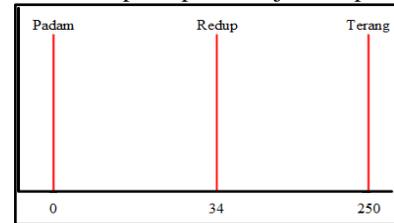
Berikut adalah tampilan *domain range* input cahaya dapat ditunjukkan pada Tabel 6.

Tabel 6. Tampilan *Domain Range* Input Cahaya

| No. | Fungsi Keanggotaan | Range (Lux) |
|-----|--------------------|--------------------|
| 1 | Gelap | [0 500] |
| 2 | Normal | [300 1500] |
| 3 | Terang | [1500 \geq 3000] |

Pada *output* lampu terdapat 3 fungsi keanggotaan yaitu padam, redup dan terang. Masing-masing memiliki *domain*

range atau batasan nilai yang mengacu pengkondisian lampu pada ruangan menggunakan logika fuzzy. Tampilan *domain range* output PWM lampu dapat ditunjukkan pada Gambar 11.



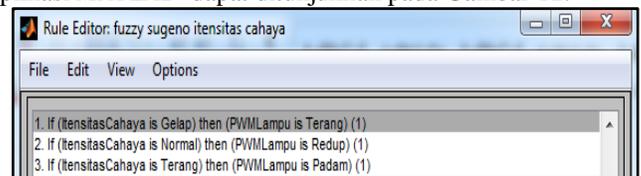
Gambar 11. Keanggotaan Output PWM Lampu

Berikut adalah tampilan *domain range output* lampu dapat ditunjukkan pada Tabel 7.

Tabel 7. Tampilan *Domain Range Output* Lampu

| No. | Fungsi Keanggotaan | Range (PWM) |
|-----|--------------------|-------------|
| 1 | Padam | 0 |
| 2 | Redup | 34 |
| 3 | Terang | 250 |

Langkah selanjutnya menerapkan aturan *Rule Editor* pada aplikasi *MATLAB* sehingga dapat dilihat keterangan perlakuan *input* intensitas cahaya (lux) dan PWM lampu dan sebuah pesan untuk *output* dari 3 fungsi Keanggotaan pada lampu. Berikut tampilan *Rule Editor* intensitas cahaya pada aplikasi *MATLAB* dapat ditunjukkan pada Gambar 12.



Gambar 12. *Rule Editor* Intensitas Cahaya Aplikasi Matlab

Keterangan :

1. Jika intensitas cahaya gelap maka PWM lampu terang.
2. Jika intensitas cahaya normal maka PWM lampu redup.
3. Jika intensitas cahaya terang maka PWM lampu Padam.

Setelah proses pada aplikasi *MATLAB* selesai dilanjutkan dengan pengujian sensor BH1750 bertujuan untuk mengetahui keselisihan dari sensor dalam mendeteksi cahaya yang masuk pada ruangan. Alat ukur yang digunakan sebagai pembanding adalah Lux Meter. Perubahan nilai input dilakukan dengan memberi cahaya dari lampu senter. Berikut merupakan hasil pengujian sensor BH1750 pada intensitas cahaya dapat ditunjukkan pada Tabel 8.

Tabel 8. Hasil Pengujian Sensor BH1750 Pada Intensitas Cahaya

| No. | Sensor BH1750(Lux) | Lux Meter(Lux) | Error % |
|-----------------|--------------------|----------------|---------|
| 1 | 0 | 0 | 0% |
| 2 | 8 | 8 | 0% |
| 3 | 19.17 | 19 | 0.89% |
| 4 | 377.5 | 377 | 0.13% |
| 5 | 430 | 430 | 0% |
| 6 | 895 | 895 | 0% |
| 7 | 1080 | 1080 | 0% |
| 8 | 1706.67 | 1706 | 0.03% |
| 9 | 1874 | 1874 | 0% |
| 10 | 2325.83 | 2385 | 0.03% |
| Nilai rata-rata | | | 0.10% |

Pada tabel 8. pengukuran intensitas cahaya pada sensor BH1750 dan Lux Meter untuk mencari selisih kesalahan perhitungan adalah:

$$Error \% = \frac{\text{nilai asli pembacaan} - \text{nilai terukur}}{\text{nilai terukur}} \times 100 \quad (4)$$

Hasil pengujian error % di dapat dari nilai asli pembacaan intensitas cahaya pada sensor BH1750 dan nilai terukur dengan menggunakan Lux Meter yang di ukur untuk melihat tingkat kesalahan dalam pengukuran intensitas cahaya. Nilai rata-rata error di dapat dari jumlah nilai error keseluruhan dibagi dengan banyaknya percobaan data yang diambil sehingga didapat nilai rata-rata error Sebesar 0.10 % .

Berikut adalah hasil perlakuan input itensitas cahaya terhadap output PWM lampu pada aplikasi *Openhabian*. Berikut tampilan aplikasi *Openhabian* pada *Things* Teras Rumah dapat ditunjukkan pada Gambar 13.



Gambar 13. Hasil Pembacaan Lux Dan PWM Lampu Aplikasi *Openhabian*

Pada Gambar 13 adalah hasil pembacaan intensitas cahaya (lux) pada sensor BH175 dan perlakuan pada PWM lampu yang sudah di tentukan antara *range input* dan outputnya. Bilamana nilai lux = 3.33 maka lampu menyala dalam keadaan terang dengan nilai PWM lampu = 250, bilamana nilai lux = 500 maka lampu menyala dalam keadaan remang dengan nilai PWM lampu = 34 dan bilamana nilai lux

= 2325.83 maka lampu akan gelap/padam dengan nilai PWM lampu = 0.

Setelah melihat hasil pembacaan intensitas cahaya (lux) pada sensor BH1750 PWM lampu pada aplikasi *Openhabian*. Dilanjutkan dengan pengujian pada aplikasi *MATLAB* bertujuan untuk mengetahui keselisihan pembacaan pada aplikasi *Openhabian* dan aplikasi *MATLAB*. Berikut merupakan hasil pengujian itensitas cahaya pada aplikasi *Openhabian* dan aplikasi *MATLAB* dapat ditunjukkan pada Tabel 9.

Tabel 9. Hasil Pengujian Itensitas Cahaya Pada Aplikasi *Openhabian* Dan Aplikasi *MATLAB*

| No | Aplikasi <i>Openhabian</i> | | Aplikasi <i>MATLAB</i> | | Error % |
|-----------------|----------------------------|-----------|------------------------|-----------|---------|
| | Lux | PWM Lampu | Lux | PWM Lampu | |
| 1 | 0 | 250 | 0 | 250 | 0% |
| 2 | 8 | 250 | 8 | 250 | 0% |
| 3 | 19.17 | 250 | 19.17 | 250 | 0% |
| 4 | 430 | 91.06 | 430 | 91.06 | 0% |
| 5 | 500 | 34 | 500 | 34 | 0% |
| 6 | 895 | 34 | 895 | 34 | 0% |
| 7 | 1080 | 32 | 1080 | 34 | 0% |
| 8 | 1706.67 | 32 | 1706.67 | 34 | 0% |
| 9 | 1874 | 0 | 1874 | 0 | 0% |
| 10 | 2325.83 | 0 | 2325.83 | 0 | 0% |
| Nilai rata-rata | | | | | 0% |

Pada tabel 9. pengujian pada aplikasi *Openhabian* dan pengukuran pada aplikasi *MATLAB* untuk mencari selisih kesalahan perhitungan adalah:

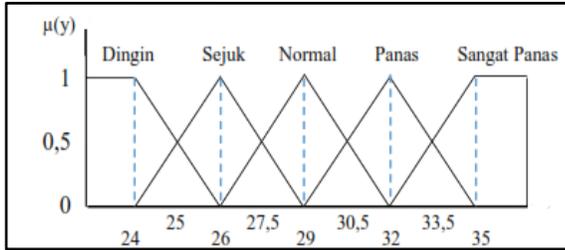
$$Error \% = \frac{\text{nilai asli pembacaan} - \text{nilai terukur}}{\text{nilai terukur}} \times 100 \quad (5)$$

Hasil pengujian *error %* di dapat dari nilai asli pembacaan intensitas cahaya (lux) dan PWM lampu pada aplikasi *Openhabian* dan nilai terukur dengan menggunakan aplikasi *MATLAB* yang di uji untuk melihat tingkat kesalahannya. Nilai rata-rata error di dapat dari jumlah nilai error keseluruhan di bagi dengan banyaknya percobaan data yang diambil sehingga didapat nilai rata-rata error Sebesar 0%.

a. Pengujian Sensor LM35 Pada Suhu Ruang

Pada pengujian ini di lakukan *input* suhu terhadap 5 fungsi keanggotaan yaitu: sangat gelap, gelap, remang, terang dan sangat terang. Masing-masing mempunyai *domain range* atau batasan nilai. Untuk pengembangan logika *fuzzy* pada

sistem pencahayaan menggunakan aplikasi *MATLAB* dan diaplikasikan pada sensor LM35, mikrokontroler Arduino dan ESP8266. Tampilan *domain range* atau batasan nilai *input* suhu dapat ditunjukkan pada Gambar 12.



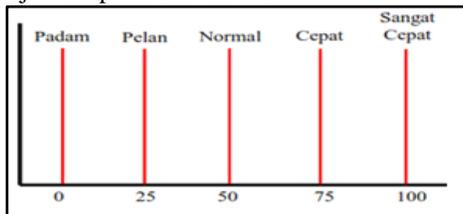
Gambar 12. Keanggotaan Input Suhu

Berikut adalah tampilan *domain range* input suhu dapat ditunjukkan pada Tabel 10.

Tabel 10. Tampilan *Domain Range* Input Suhu

| No. | Fungsi Keanggotaan | Range (Suhu) |
|-----|--------------------|--------------|
| 1 | Dingin | [0 26] |
| 2 | Sejuk | [24 29] |
| 3 | Normal | [26 32] |
| 4 | Panas | [29 35] |
| 5 | Sangat Panas | [32 ≤45] |

Pada *output* lampu terdapat 5 fungsi keanggotaan yaitu padam, pelan, normal, cepat dan sangat cepat. Masing-masing memiliki *domain range* atau batasan nilai yang mengacu pengkondisian kipas pada ruangan menggunakan logika *fuzzy internet of things*. Tampilan *domain range output* PWM Kipas dapat ditunjukkan pada Gambar 13.



Gambar 13. Keanggotaan Output PWM Kipas

Berikut adalah tampilan *domain range output* kipas dapat ditunjukkan pada Tabel 11.

Tabel 11. Tampilan *Domain Range Output* PWM Kipas

| No. | Fungsi Keanggotaan | Range (PWM) |
|-----|--------------------|-------------|
| 1 | Padam | 0 |
| 2 | Pelan | 25 |
| 3 | Normal | 50 |
| 4 | Cepat | 75 |
| 5 | Sangat Cepat | 100 |

Langkah selanjutnya menerapkan aturan *Rule Editor* pada aplikasi *MATLAB* sehingga dapat di lihat keterangan

perlakuan *input* suhu dan PWM kipas. Berikut tampilan *Rule Editor* suhu pada aplikasi *MATLAB* dapat ditunjukkan pada Gambar 14.



Gambar 14. Rule Editor Suhu Aplikasi Matlab

Keterangan :

1. Jika suhu dingin maka PWM kipas padam.
2. Jika suhu sejuk maka PWM kipas pelan.
3. Jika suhu normal maka PWM kipas normal.
4. Jika suhu panas maka PWM kipas cepat.
5. Jika suhu sangat panas maka PWM kipas sangat cepat.

Setelah proses pada aplikasi *MATLAB* selesai dilanjutkan dengan pengujian sensor LM35 bertujuan untuk mengetahui keselisihan dari sensor dalam mendeteksi cahaya yang masuk pada ruangan. Alat ukur yang digunakan sebagai pembanding adalah Termometer. Perubahan nilai input dilakukan dengan memberi suhu dari Hedrayer dan minuman dingin. Berikut merupakan hasil pengujian sensor LM35 pada suhu ruangan dapat ditunjukkan pada Tabel 12.

Tabel 12. Hasil Pengujian Sensor LM35 Pada Suhu Ruangan

| No. | Sensor LM35 (Suhu) | Termometer (Suhu) | Error % |
|-----------------|--------------------|-------------------|---------|
| 1 | 19.55 | 19 | 2.89% |
| 2 | 21.5 | 21 | 2.38% |
| 3 | 26.39 | 26.3 | 0.34% |
| 4 | 29.33 | 29 | 1.13% |
| 5 | 32.26 | 32 | 0.81% |
| 6 | 35.1 | 35.1 | 0% |
| 7 | 38.2 | 38.1 | 0.26% |
| 8 | 41.54 | 41 | 1.31% |
| 9 | 43.1 | 43.1 | 0% |
| 10 | 44 | 44 | 0% |
| Nilai rata-rata | | | 0.91% |

Pada tabel 12. pengukuran suhu ruangan pada sensor LM35 dan Termometer untuk mencari selisih kesalahan perhitungan adalah:

$$Error \% = \frac{\text{nilai asli pembacaan} - \text{nilai terukur}}{\text{nilai terukur}} \times 100 \quad (6)$$

Hasil pengujian error % di dapat dari nilai asli pembacaan intensitas cahaya pada sensor LM35 dan nilai terukur dengan menggunakan Termometer yang di ukur untuk melihat tingkat kesalahan dalam pengukuran suhu ruangan. Nilai rata-rata error di dapat dari jumlah nilai error

keseluruhan dibagi dengan banyaknya percobaan data yang diambil sehingga didapat nilai rata-rata error sebesar 0.91% .

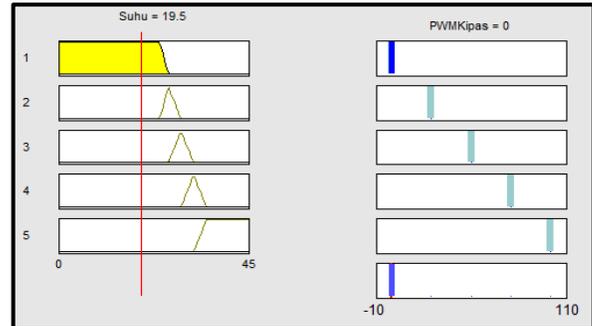
Langkah selanjutnya menerapkan aturan aplikasi *MATLAB* yang sudah di buat dan diaplikasikan pada sensor LM35, mikrokontroler Arduino dan ESP8266 sehingga dapat di lihat keterangan perlakuan *input* suhu terhadap *output* PWM kipas. Berikut tampilan pembacaan suhu dan PWM kipas pada aplikasi *Openhabian* dapat ditunjukkan pada Gambar 15.



Gambar 15. Hasil Pembacaan Suhu Dan PWM Kipas Aplikasi *Openhabian*

Pada Gambar 13. adalah hasil pembacaan suhu pada sensor LM35 dan *output* berupa PWM kipas dari 5 fungsi keanggotaan lampu yang sudah di tentukan antara *range input* dan *output*nya. Di mana nilai suhu =19.55 maka, kipas dalam keadaan padam atau tidak berputar dengan nilai PWM kipas = 0, nilai suhu = 26.39 maka kipas akan berputar dalam keadaan pelan dengan nilai PWM kipas = 28.27, nilai suhu = 52.71 maka kipas akan berputar dalam keadaan normal dengan nilai PWM kipas = 29.33, nilai suhu = 32.26 maka kipas akan berputar dalam keadaan cepat dengan nilai pwm kipas = 77.15 dan jika nilai suhu = 41.54 maka kipas akan berputar dalam keadaan sangat cepat dengan nilai PWM kipas = 100.

Setelah melihat hasil pembacaan *input* suhu pada sensor LM35 dan *output* PWM kipas dari 5 fungsi keanggotaan kipas yang sudah di tentukan antara *range input* dan *output* pada aplikasi *Openhabian*. Dilanjutkan dengan pengujian pada aplikasi *MATLAB* bertujuan untuk mengetahui keselisihan pembacaan pada aplikasi *Openhabian* dan aplikasi *MATLAB*. Hasil tampilan pada aplikasi *MATLAB* dapat di tunjukan pada Gambar 16.



Gambar 16. Hasil Pembacaan Suhu Dan PWM Kipas Aplikasi *MATLAB*

Berikut merupakan hasil pengujian suhu pada aplikasi *Openhabian* dan aplikasi *MATLAB* dapat ditunjukkan pada Tabel 13.

Tabel 13. Hasil Pengujian Suhu Pada Aplikasi *Openhabian* Dan Aplikasi *MATLAB*

| No | Aplikasi <i>Openhabian</i> | | Aplikasi <i>MATLAB</i> | | Error % |
|-----------------|----------------------------|-----------|------------------------|-----------|---------|
| | Suhu | PWM Kipas | Suhu | PWM Kipas | |
| 1 | 19.55 | 0 | 19.55 | 0 | 0% |
| 2 | 20 | 0 | 20 | 0 | 0% |
| 3 | 24.93 | 11.60 | 24.93 | 11.60 | 0% |
| 4 | 26.39 | 28.27 | 26.39 | 28.2 | 0.24% |
| 5 | 27.37 | 36.40 | 27.37 | 36.4 | 0% |
| 6 | 28.35 | 44.67 | 28.35 | 44.6 | 0.15% |
| 7 | 29.33 | 52.71 | 29.33 | 52.7 | 0.01% |
| 8 | 30.3 | 60.86 | 30.3 | 60.8 | 0.09% |
| 9 | 32.26 | 77.15 | 32.26 | 77.1 | 0.06% |
| 10 | 41.54 | 100 | 41.54 | 100 | 0% |
| Nilai rata-rata | | | | | 0.05% |

Pada tabel 13 pengujian pada aplikasi *Openhabian* dan pengukuran pada *rule MATLAB* untuk mencari selisih kesalahan perhitungan adalah:

$$Error \% = \frac{\text{nilai asli pembacaan} - \text{nilai terukur}}{\text{nilai terukur}} \times 100 \quad (7)$$

Hasil pengujian *error %* di dapat dari nilai asli pembacaan intensitas cahaya (lux) dan PWM lampu pada aplikasi *Openhabian* dan nilai terukur dengan menggunakan *rule MATLAB* yang di uji untuk melihat tingkat kesalahannya. Nilai rata-rata *error* di dapat dari jumlah nilai *error* keseluruhan di bagi dengan banyaknya percobaan data yang diambil sehingga didapat nilai rata-rata error sebesar 0.05%.

VI. Kesimpulan

Berdasarkan pengujian dan analisis maka di tarik beberapa kesimpulan antara lain:

1. Komunikasi antara *user* dengan mikrokontroler dilakukan secara *wireless* dengan menggunakan *wireless wan* yang menghubungkan antara mikrokontroler dengan jaringan *internet*.
2. Tingkat kesalahan yang di dapat pada pengujian itensitas cahaya dengan aplikasi *MATLAB* yaitu 0% untuk jumlah data sebanyak 10. Pengujian suhu ruangan dengan aplikasi *MATLAB* didapatkan tingkat kesalahannya ialah 0.05% dengan jumlah data sebanyak 10.

VII. Daftar Pustaka

- [1] Hidayat Nur Isnianto, Muhammad Arrofiq, Rijeqi Rahmawati, dan Bagus Mulyo Tyoso pada tahun 2019 (April) "Sistem Telemonitoring KWH Meter Menggunakan Modul WI-FI ESP8266 Berbasis Arduino Uno" Jurnal Rekayasa Elektriika Vol. 15, No. 1, hal. 25-33.
- [2] Anggher Dea Pangestu, Feby Ardianto, Bengawan Alfaresi pada tahun 2019 (Juni)"Sistem Monitoring Beban Listrik Berbasis Arduino NODEMCU ESP826" Program Studi Elektro, Fakultas Teknik, Universitas Muhammadiyah Palembang Jurnal Ampere Vol 4 No 1.
- [3] Hartono Budi Santoso, Spto Prajogo, Sri Paryanto Mursid pada tahun 2018 (September) "Pengembangan Sistem Pemantauan Konsumsi Energi Rumah Tangga Berbasis Internet of Things (IoT) " Vol. 6 No. 3 Halaman 357 – 366.
- [4] Asep Muhamad Alipudin, Didik Notosudjono, Dimas Bangun Fiddiansyah pada tahun 2018 dengan judul "Rancang Bangun Alat Monitoring Biaya Listrik Terpakai Berbasis Internet Of Things (IOT)" Vol 1, No 1.
- [5] Arief, K., Purwanto, D., & Kusuma, H. (2019). Algoritma Menghadang Bola Dengan Metode Fuzzy Logic Untuk Robot Penjaga Gawang Sepak Bola Beroda. Jurnal Teknik ITS, Volume 7, Nomor 2.