

**SISTEM MANAJEMEN ENERGI BERBASIS *INTERNET*  
*OF THINGS* MENGGUNAKAN MODULASI LORA (*LONG*  
*RANGE*)**

SKRIPSI

Diajukan sebagai salah satu syarat  
untuk memperoleh gelar  
Sarjana Teknik



Oleh:

OKI SETIAWAN

NIM. I0716027

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS SEBELAS MARET  
SURAKARTA  
2020**



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS SEBELAS MARET  
FAKULTAS TEKNIK  
**PROGRAM STUDI TEKNIK ELEKTRO**  
*Jalan. Ir. Sutami nomor 36 A Kentingan Surakarta 57126*  
*Telepon. 0271 647069 psw 438, faksimili: 0271 662118*

## **SURAT TUGAS**

**Nomor : 062/TA/TE/2020**

Kepala Program Studi Teknik Elektro Fakultas Teknik Universitas Sebelas Maret memberikan tugas kepada:

Nama Mahasiswa	: Oki Setiawan
NIM	: 10716027
Bidang peminatan	: Sistem Energi Listrik (SEL)
Pembimbing Utama	: Agus Ramelan, S.Pd., M.T. NIP. 199203152019031017
Pembimbing Pendamping	: Chico Hermanu BA, S.T., M.Eng. NIP. 198804162015041002
Mata kuliah pendukung	: 1. Manajemen Energi 2. Instrumentasi 3. Teknik Instalasi

untuk mengerjakan dan menyelesaikan Tugas Akhir dengan judul :

**Sistem Manajemen Energi Berbasis Internet of Things Menggunakan  
Modulasi LoRa (Long Range)**

Surat tugas ini dibuat untuk dilaksanakan dengan sebaik-baiknya.

Surakarta, 16 Maret 2020  
Kepala Program Studi

Feri Adriyanto, Ph.D.  
NIP. 196801161999031001

Tembusan:

1. Mahasiswa ybs.
2. Dosen Pembimbing TA
3. Koordinator TA
4. Arsip

## **SURAT PERNYATAAN INTEGRITAS PENULIS**

Saya mahasiswa Program Studi Sarjana Teknik Elektro Universitas Sebelas  
Maret yang bertanda tangan di bawah ini:

Nama : Oki Setiawan  
NIM : I0716027  
Judul tugas akhir :SISTEM MANAJEMEN ENERGI BERBASIS  
*INTERNET OF THINGS* MENGGUNAKAN  
MODULASI LORA (*LONG RANGE*)

Dengan ini menyatakan bahwa tugas akhir yang saya susun tidak mencontoh atau melakukan plagiat dari karya tulis orang lain. Jika terbukti tugas akhir yang saya susun tersebut merupakan hasil plagiat dari karya orang lain maka tugas akhir yang saya susun tersebut dinyatakan batal dan gelar sarjana yang saya peroleh dengan sendirinya dibatalkan atau dicabut.

Demikian surat pernyataan ini saya buat dengan sebenarnya dan apabila dikemudian hari terbukti melakukan kebohongan maka saya sanggup menanggung segala konsekuensinya.

Surakarta, Juli 2020

Oki Setiawan  
NIM. I0716027

**HALAMAN PENGESAHAN TIM PEMBIMBING DAN PENGUJI**

**SISTEM MANAJEMEN ENERGI BERBASIS *INTERNET OF THINGS*  
MENGUNAKAN MODULASI LORA (*LONG RANGE*)**

Disusun oleh :

**OKI SETIAWAN**  
NIM. I0716027

Dosen Pembimbing I

Dosen Pembimbing II

**Agus Ramelan, S.Pd., M.T**  
NIP. 199203152019031017

**Chico Hermanu B.A., S.T., M.Eng.**  
NIP. 198804162015041002

Telah dipertahankan di hadapan Tim Dosen Penguji pada hari Selasa tanggal  
28 Juli 2020.

1. **Agus Ramelan S.Pd., M.T.**  
NIP. 199203152019031017 .....
2. **Chico Hermanu B. A S.T., M.Eng.**  
NIP. 198804162015041002 .....
3. **Irwan Iftadi, S.T., M.Eng.**  
NIP. 197004041996031002 .....
4. **Subuh Pramono, S.T., M.T.**  
NIP. 198106092003121002 .....

Kepala Prodi Teknik Elektro

Mengetahui,  
Koordinator Tugas Akhir

**Feri Adriyanto Ph.D.**  
NIP. 196801161999031001

**Muhammad Hamka I., S.T., M.Eng.**  
NIP. 1988122920161001

# **SISTEM MANAJEMEN ENERGI BERBASIS *INTERNET OF THINGS* MENGGUNAKAN MODULASI LORA (*LONG RANGE*)**

**OKI SETIAWAN**

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Sebelas Maret

Email: [oki.setiawan@student.uns.ac.id](mailto:oki.setiawan@student.uns.ac.id)

## **ABSTRAK**

Seiring berjalannya waktu, kebutuhan akan energi listrik akan terus mengalami peningkatan. Dengan masih mayoritasnya energi listrik yang berasal dari bahan bakar fosil, maka akan terjadi krisis energi pada masa yang akan datang. Salah satu hal yang dapat dilakukan untuk mengatasi krisis energi adalah dengan cara melakukan manajemen energi. Perancangan sistem manajemen energi yang dilakukan pada penelitian ini bertujuan untuk mempermudah dalam *monitoring* dan kontrol energi. Dengan menggunakan *Internet of Things* (IoT) dengan modulasi LoRa (*Long Range*), manajemen energi akan semakin mudah dilakukan. Hasilnya didapatkan pada *end node* 1 memiliki akurasi sistem sebesar 98.35 % dan presisi sistem sebesar 96.3 % serta waktu tunda relay sebesar 2.2 detik. Sedangkan pada *end node* 2 memiliki akurasi sistem sebesar 98.55 % dan presisi sistem sebesar 98.95 % serta waktu tunda relay sebesar 2.6 detik. Pada pengujian jangkauan, jarak maksimal yang mampu dicapai LoRa adalah sebesar 250 meter pada area perkantoran atau gedung. Sedangkan pada pengujian *reliability*, didapatkan *packet loss* yang terjadi pada sistem sebesar 8.25 %.

Kata Kunci : *Internet of Things*, manajemen energi, *monitoring*, *controlling*, dan LoRa.

# **ENERGY MANAGEMENT SYSTEM BASED ON INTERNET OF THINGS USING LORA (LONG RANGE) MODULATION**

**OKI SETIAWAN**

*Department of Electrical Engineering, Engineering Faculty*

Universitas Sebelas Maret

Email: [oki.setiawan@student.uns.ac.id](mailto:oki.setiawan@student.uns.ac.id)

## **ABSTRACT**

*Over time, the need for electrical energy will continue to increase. With the majority of electricity coming from fossil fuels, there will be an energy crisis in the future. One of the things that can be done to overcome the energy crisis is to do energy management. The design of energy management systems carried out in this study aims to facilitate the monitoring and control of energy. By using the Internet of Things (IoT) with LoRa (Long Range) modulation, energy management will be easier to do. The results obtained at end node 1 have a system accuracy of 98.35% and a system precision of 96.3% and a relay time delay of 2.2 seconds. While the end node 2 has a system accuracy of 98.55% and a system precision of 98.95% and a relay delay time of 2.6 seconds. In the range testing, the maximum distance that can be achieved by LoRa is 250 meters in the office or building area. Meanwhile, in the reliability test, the packet loss that occurred in the system was obtained at 8.25%.*

*Keywords : Internet of Things, energy management, monitoring, controlling, dan LoRa.*

## KATA PENGANTAR

Segala puji bagi Allah SWT Tuhan semesta alam, atas segala karunia, hidayah, dan izin-Nya sehingga tugas akhir dengan judul “Sistem Manajemen Energi Berbasis *Internet of Things* Menggunakan Modulasi Lora (*Long Range*)” ini dapat terselesaikan dengan lancar.

Tugas akhir ini disusun untuk memenuhi salah satu syarat agar dapat menyelesaikan studi serta dalam rangka memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik Universitas Sebelas Maret Surakarta.

Dalam penyusunan tugas akhir ini tentunya penulis selalu mendapat banyak dukungan dan bimbingan baik moril dan materil dari berbagai pihak, sehingga pada akhirnya tugas akhir ini dapat terselesaikan. Oleh karena itu, dengan segala kerendahan hati dan rasa hormat, penghargaan serta ucapan terima kasih yang tak terhingga penulis ucapkan kepada:

1. Bapak Agus Ramelan, S.Pd., M.T., selaku dosen pembimbing satu, yang senantiasa membimbing, mengarahkan dan memberikan motivasi.
2. Bapak Chico Hermanu B. A., S.T., M.Eng., selaku dosen pembimbing dua, yang senantiasa memberikan saran dan masukan baik terhadap karya tulis, presentasi ataupun metode penelitian yang digunakan.
3. Bapak Irwan Iftadi, S.T., M.Eng. dan Bapak Subuh Pramono, S.T., M.T., selaku dosen penguji yang juga memberikan saran dan masukan saat seminar hasil dan sidang pendadaran guna perbaikan penelitian yang akan datang.
4. Bapak Feri Adriyanto, PhD. selaku Kepala Program Studi Teknik Elektro.
5. Bapak Muhammad Hamka Ibrahim, S.T., M.Eng., selaku dosen pembimbing akademik.
6. Kedua orangtua saya, Alm. Bapak Siman dan Ibu Suwarti serta kakak saya Anisa Setiowati, S.Pd., yang senantiasa memberikan bantuan, semangat dan doa kepada saya.
7. Segenap jajaran dewan dosen dan staff Prodi Teknik Elektro yang telah memberikan sumbangsih terhadap Teknik Elektro.

8. Kepada teman saya Kevin Sebastian Arif dan Hisbullah Ahmad Fathoni yang telah membantu penulis dalam menyelesaikan penelitian ini.
9. Kepada teman – teman Teknik Elektro UNS yang telah menemani selama masa studi dan terus memberikan semangat serta dukungan.

Dengan adanya karya tulis ini, penulis berharap semoga dapat dimanfaatkan oleh banyak pihak, khususnya pihak institusi, dan dunia industri secara umum.

Surakarta, Juli 2020

Penulis



## DAFTAR ISI

SISTEM MANAJEMEN ENERGI BERBASIS <i>INTERNET OF THINGS</i> MENGGUNAKAN MODULASI LORA ( <i>LONG RANGE</i> ).....	i
HALAMAN PENUGASAN .....	ii
SURAT PERNYATAAN INTEGRITAS PENULIS .....	iii
HALAMAN PENGESAHAN TIM PEMBIMBING DAN PENGUJI .....	iv
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiv
DAFTAR SINGKATAN.....	xv
BAB I 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 II TINJAUAN PUSTAKA.....	5
2.1. <i>Internet of Things</i> (IoT) .....	5
2.2. <i>MQTT Protocol</i> .....	6
2.3. Manajemen Energi Listrik .....	7
2.4. <i>LoRa (Long Range)</i> .....	7
2.5. <i>LoRa Gateway</i> (Dragino LG01-N).....	7
2.6. <i>Dragio LoRa Shield</i> 915 MHZ .....	8
2.7. <i>Arduino Uno</i> .....	8
2.8. <i>Relay</i> .....	9
2.9. <i>Sensor Arus ACS712</i> .....	10
2.10. <i>Sensor Tegangan ZMPT101B</i> .....	11
2.11. <i>Node</i> .....	11
2.12. <i>App Inventor2</i> .....	12
2.13. <i>Thingspeak</i> .....	12
2.14. <i>RSSI</i> .....	13
2.15. <i>Penelitian Terdahulu</i> .....	14
BAB III METODOLOGI PENELITIAN .....	17

3.1.	Lokasi dan Waktu Penelitian .....	17
3.2.	Alat dan Bahan Penelitian .....	17
3.3.	Spesifikasi Rancangan Sistem.....	18
3.4.	Prosedur Penelitian .....	19
3.5.	Tahap Persiapan.....	20
3.6.	Tahap Perancangan Hardware .....	21
3.6.1.	Konfigurasi LoRa <i>Gateway</i> .....	22
3.6.2.	Perancangan Sistem <i>Monitoring</i> .....	23
3.6.3.	Perancangan Sistem Kontrol .....	24
3.7.	Tahap Perancangan <i>Software</i> .....	25
3.8.	Uji Performa .....	27
3.8.1.	Pengujian Presisi .....	27
3.8.2.	Pengujian Akurasi .....	28
3.8.3.	Pengujian Waktu Tunda Relay.....	29
3.8.4.	Pengujian Jangkauan.....	29
3.8.5.	Pengujian Keandalan ( <i>Reliability</i> ) .....	30
BAB IV HASIL DAN PEMBAHASAN .....		31
4.1.	Hasil Perancangan <i>End Node 1</i> .....	31
4.1.1.	Hasil Uji Presisi Sensor Tegangan ZMPT101b .....	31
4.1.2.	Hasil Uji Presisi Sensor Arus ACS712 .....	33
4.1.3.	Hasil Uji Akurasi Sensor Tegangan ZMPT101b .....	36
4.1.4.	Hasil Uji Akurasi Sensor Arus ACS712 .....	37
4.1.5.	Waktu Tunda Relay.....	41
4.2.	Hasil Perancangan <i>End Node 2</i> .....	43
4.2.1.	Hasil Uji Presisi Sensor Tegangan ZMPT101b .....	44
4.2.2.	Hasil Uji Presisi Sensor Arus ACS712 .....	46
4.2.3.	Hasil Uji Akurasi Sensor Tegangan ZMPT101b .....	48
4.2.4.	Hasil Uji Akurasi Sensor Arus ACS712 .....	49
4.2.5.	Waktu Tunda Relay.....	52
4.3.	Hasil Pengujian Jangkauan.....	54
4.3.1.	Jarak 1 Meter .....	54
4.3.2.	Jarak 50 Meter .....	54
4.3.3.	Jarak 100 Meter .....	54
4.3.4.	Jarak 150 Meter .....	55
4.3.5.	Jarak 200 Meter .....	55
4.3.6.	Jarak 250 Meter .....	55

4.3.7. Jarak 300 Meter .....	55
4.4. Hasil Pengujian Keandalan (Reliability) .....	56
4.5. Hasil Perancangan Aplikasi Android .....	58
4.5.1. <i>Screen</i> Menu Home .....	58
4.5.2. <i>Screen</i> Monitoring Lantai 1 .....	59
4.5.3. <i>Screen</i> Monitoring Lantai 2 .....	59
4.5.4. <i>Screen</i> Control Lantai 1 .....	60
4.5.5. <i>Screen</i> Control Lantai 2 .....	61
BAB V PENUTUP .....	63
5.1. Kesimpulan .....	63
5.2. Saran .....	64
DAFTAR PUSTAKA .....	65
LAMPIRAN .....	68

## DAFTAR GAMBAR

Gambar 2.1 Gambaran konsep IoT .....	6
Gambar 2.2 Konsep <i>Protocol</i> MQTT untuk IoT.....	6
Gambar 2.3 LG01-N LoRa IoT <i>Gateway</i> .....	8
Gambar 2.4 Dragino LoRa <i>Shield</i> 915 MHZ .....	8
Gambar 2.5 Arduino Uno .....	9
Gambar 2.6 Relay Module 4 Channel.....	9
Gambar 2.7 Sensor Arus ACS712.....	10
Gambar 2.8 Sensor Tegangan ZMPT101B.....	11
Gambar 2. 9 Node .....	11
Gambar 3.1 Arsitektur Sistem Manajemen Energi dengan LoRa .....	19
Gambar 3.2 Prosedur Penelitian.....	20
Gambar 3.3 Blok Diagram .....	21
Gambar 3.4 <i>Setting network gateway</i> .....	22
Gambar 3.5 <i>Setting client</i> .....	22
Gambar 3.6 LoRa <i>setting</i> .....	23
Gambar 3.7 <i>Setting cloud server</i> .....	23
Gambar 3.8 Diagram alir sistem <i>monitoring</i> .....	24
Gambar 3.9 Diagram Alir Sistem Kontrol .....	25
Gambar 3.10 Block menu home.....	26
Gambar 3. 11 <i>Block menu monitoring</i> .....	26
Gambar 3.12 <i>Block menu control</i> .....	27
Gambar 4.1 <i>End Node</i> 1 .....	31
Gambar 4.2 Pembacaan ZMPT101B 50 Kali.....	33
Gambar 4.3 Pembacaan ACS712 50 Kali .....	36
Gambar 4.4 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter .....	37
Gambar 4.5 Perbandingan Pembacaan Arus pada ACS712_1 dan Multimeter ....	39
Gambar 4.6 Perbandingan Pembacaan Arus pada ACS712_2 dan Multimeter ....	40
Gambar 4.7 Perbandingan Pembacaan Arus pada ACS712_3 dan Multimeter ....	40
Gambar 4.8 Perbandingan Pembacaan Arus pada ACS712_4 dan Multimeter ....	41
Gambar 4.9 <i>End Node</i> 2 Tampak Depan – Samping Kanan.....	43
Gambar 4.10 <i>End Node</i> 2 Tampak Belakang – Samping Kiri .....	43
Gambar 4.11 Pembacaan ZMPT101B 50 kali .....	45
Gambar 4.12 Pembacaan ACS712 50 kali .....	48
Gambar 4.13 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter .....	49
Gambar 4.14 Perbandingan Pembacaan Arus pada ACS712_1 dan Multimeter ..	51
Gambar 4.15 Perbandingan Pembacaan Arus pada ACS712_2 dan Multimeter ..	51
Gambar 4.16 Perbandingan Pembacaan Arus pada ACS712_3 dan Multimeter ..	52
Gambar 4.17 Perbandingan Pembacaan Arus pada ACS712_4 dan Multimeter ..	52
Gambar 4.18 Energi Saat Pengujian Keandalan .....	56
Gambar 4.19 (a) Grafik energi outlet 1, (b) Grafik energi outlet 2, (c) Grafik energi outlet 3, (d) Grafik energi outlet 4.....	57
Gambar 4.20 Aplikasi Android Menu Home .....	58
Gambar 4.21 Aplikasi Android Menu Monitoring Energi Lantai 1 .....	59
Gambar 4. 22 Aplikasi Android Menu Monitoring Energi Lantai 2 .....	59

Gambar 4. 23 Aplikasi Android Menu Control Lantai 1 .....	60
Gambar 4.24 Aplikasi Android Menu Control Lantai 2 .....	61

## DAFTAR TABEL

Tabel 2.1 Level sinyal RSSI.....	13
Tabel 2.2 Penelitian Terdahulu .....	14
Tabel 3.1 Tabel Alat dan Bahan Penelitian.....	17
Tabel 3.2 <i>Path Loss Exponent</i> .....	30
Tabel 4.1 Hasil Pembacaan ZMPT101b 50 Kali.....	31
Tabel 4.2 Hasil Pembacaan ACS712 50 Kali .....	34
Tabel 4.3 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter .....	36
Tabel 4.4 Perbandingan Pembacaan Arus pada ACS712_1 dan Multimeter .....	37
Tabel 4.5 Perbandingan Pembacaan Arus pada ACS712_2 dan Multimeter .....	38
Tabel 4.6 Perbandingan Pembacaan Arus pada ACS712_3 dan Multimeter .....	38
Tabel 4.7 Perbandingan Pembacaan Arus pada ACS712_4 dan Multimeter .....	39
Tabel 4.8 Waktu Tunda Relay Kondisi ON-OFF .....	41
Tabel 4.9 Waktu Tunda Relay Kondisi OFF-ON .....	42
Tabel 4.10 Hasil Pembacaan ZMPT101B 50 Kali.....	44
Tabel 4.11 Hasil Pembacaan ACS712 50 Kali .....	46
Tabel 4.12 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter .....	48
Tabel 4.13 Perbandingan Pembacaan Arus pada ACS712_1 dan Multimeter .....	49
Tabel 4.14 Perbandingan Pembacaan Arus pada ACS712_2 dan Multimeter .....	50
Tabel 4.15 Perbandingan Pembacaan Arus pada ACS712_3 dan Multimeter .....	50
Tabel 4.16 Perbandingan Pembacaan Arus pada ACS712_4 dan Multimeter .....	51
Tabel 4.17 Waktu Tunda Relay Kondisi ON-OFF .....	53
Tabel 4.18 Waktu Tunda Relay Kondisi ON-OFF .....	53
Tabel 4.19 Pengujian Jangkauan.....	54

## DAFTAR SINGKATAN

IoT	: <i>Internet of Things</i>
LoRa	: <i>Long Range</i>
MQTT	: <i>Message Queue Telemetry Transport</i>
RSSI	: <i>Received Signal Strength Indicator</i>
SD	: <i>Standard Deviation</i>
RSD	: <i>Relative Standard Deviation</i>
MAPE	: <i>Mean Absolute Percentage Error</i>

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Untuk mengatasi krisis energi, salah satu hal yang dapat dilakukan adalah manajemen energi. Manajemen energi dapat diartikan sebagai upaya yang berdampak langsung pada organisasi, teknik dan aksi yang ekonomis dalam rangka meminimalisir penggunaan energi [1]. Manajemen energi pun mempunyai percabangan yaitu konservasi dan pemantauan [2]. Pada penelitian ini akan lebih di fokuskan pada sisi pemantauan.

Menurut Badan Pengkajian dan Penerapan Teknologi (BPPT), kebutuhan listrik di Indonesia diproyeksikan meningkat lebih dari 6 kali lipat dari 226 TWh pada tahun 2017 menjadi 1.471 Twh pada tahun 2050 [3]. Pada tahun 2017 lebih dari 88 % energi listrik di Indonesia dihasilkan dari bahan bakar fosil seperti batu bara (65%), gas (18%) dan minyak (5%). Hanya 12% dari total produksi listrik di Indonesia yang berasal dari sumber energi baru dan terbarukan (EBT). Sebagai energi fosil yang tidak dapat diperbarui, bahan bakar seperti batu bara, gas dan minyak akan semakin berkurang ketersediaannya yang berdampak pada naiknya tarif dasar listrik (TDL).

Salah satu langkah yang dapat dilakukan adalah dengan cara melakukan *monitoring* energi listrik serta kontrol peralatan listrik yang digunakan. Seiring dengan perkembangan IoT, langkah-langkah penghematan energi akan semakin mudah dilakukan.

Penelitian tentang manajemen energi berbasis IoT pernah dilakukan oleh beberapa peneliti lain. Diantaranya penelitian oleh Tui-Yi Yang [4] dimana penulis merancang sebuah *smart home energy management system* (HEMS) yang mampu untuk menganalisis penggunaan listrik and riwayat seluruh peralatan rumah tangga yang digunakan dan hasilnya didapatkan skema yang dibuat mampu untuk mengendalikan biaya energi yang dipakai dengan cara monitoring cerdas pada kondisi kelistrikan rumah tangga. Mohammed Abo-Zahhad [5] melakukan perancangan sebuah *energy management system* (EMS) yang terdiri dari Arduino Uno, Xbee-PRO ZigBee module, ACS712 sensor serta NI LABVIEW *software*. Hasilnya EMS dapat mengurangi konsumsi energi pada saat beban puncak dan



dengan demikian mengurangi emisi carbon. Pada penelitian Yatharth Gupta [6] dirancang sebuah sistem manajemen energi berupa *real-time monitoring* beban dan meramalkan beban yang dibutuhkan, manajemen sumber energi serta proteksi arus lebih dan otomatisasi beban dengan IoT. Khwanchai Pawasan [7] mengembangkan perangkat IoT untuk *monitoring* konsumsi energi pada suatu bangunan. Perangkat tersebut menggunakan PZEM-004t sensor energi listrik, mikrokontroler Arduino Nano dan ESP8266 serial-to-WiFi board. Hasilnya, perangkat dapat bekerja dengan baik dan mampu untuk mengumpulkan data energi yang berguna untuk melakukan manajemen energi secara efisien. Penelitian Haithem Chaouch [8] merancang sebuah sistem *heating, ventilation dan air conditioning* (HVAC) yang diimplementasikan guna mengontrol dan memonitor konsumsi energi pada gedung menggunakan *fuzzy logic*, komunikasi *machine to machine* (M2M) dan teknologi internet. Sistem yang dirancang mampu mengurangi konsumsi energi secara otomatis. Pada penelitian Hyeonwoo Jang [9] dibuat sistem yang menggunakan teknologi *mixed reality*. Teknologi *mixed reality* merupakan teknologi yang menciptakan dunia virtual dengan cara memadukan realita dengan informasi virtual melalui interaksi. Penelitian ini diproyeksikan untuk pengembangan sistem manajemen energi terintegrasi pada suatu *smart city*.

Oleh karena itu, dalam penelitian kali ini akan dilakukan perancangan sistem manajemen energi berbasis IoT dengan modulasi LoRa yang memiliki fungsi untuk *monitoring* daya listrik yang serta kontrol peralatan listrik.

## 1.2. Rumusan Masalah

1. Bagaimana merancang sistem *monitoring* penggunaan energi listrik dan kontrol peralatan listrik berbasis IoT?
2. Bagaimana melakukan pengujian performa sistem *monitoring* penggunaan energi listrik dan kontrol peralatan listrik berbasis IoT?
3. Bagaimana merancang aplikasi android sistem *monitoring* penggunaan energi listrik dan kontrol peralatan listrik berbasis IoT?

### 1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Manajemen energi yang dimaksud dalam penelitian ini dibatasi pada proses *monitoring* dan *controlling* energi listrik, tidak termasuk audit energi dan perhitungan intensitas konsumsi energi (IKE).
2. Hardware sistem energi direpresentasikan dengan dua buah *end nodes* yang merupakan sebuah *smart power outlet* dengan masing-masing *end nodes* terdiri dari 4 buah outlet listrik AC 220 V.
3. Sistem energi dianggap tidak mengalami gangguan harmonik.
4. Seluruh beban dalam penelitian ini dianggap memiliki faktor daya sebesar 0,85.

### 1.4. Tujuan Penelitian

1. Merancang *hardware* sistem manajemen energi berbasis IoT menggunakan modulasi LoRa.
2. Merancang *software* sistem manajemen energi berbasis IoT menggunakan modulasi LoRa.
3. Melakukan pengujian performa sistem manajemen energi berbasis IoT menggunakan modulasi LoRa berupa :
  - a. Pengujian akurasi sensor
  - b. Pengujian presisi sensor
  - c. Pengujian waktu tunda relay
  - d. Pengujian jangkauan
  - e. Pengujian keandalan (*reliability*)
4. Melakukan analisis sistem manajemen energi berbasis IoT menggunakan modulasi LoRa untuk mengetahui apakah sistem *monitoring* dan kontrol bekerja dengan baik atau tidak.

### 1.5. Manfaat Penelitian

1. Diharapkan dari penelitian ini dapat menekan penggunaan energi listrik dengan diberikannya fitur manajemen energi berupa *monitoring* penggunaan energi listrik dan kontrol peralatan elektronik berbasis IoT.

## 1.6. Sistematika Penulisan

Sistematika penulisan yang digunakan penulis agar proposal skripsi ini mudah dipahami adalah sebagai berikut:

### BAB I : PENDAHULUAN

Pada bab ini memberikan penjelasan singkat mengenai latar belakang, rumusan masalah, tujuan, manfaat dan sistematika penulisan.

### BAB II: KAJIAN PUSTAKA

Pada bab ini berisi uraian sistematis mengenai dasar dasar teori dari internet of things, MQTT protocol, monitoring energi listrik, lora (*long range*), lora *gateway*, lora shield, arduino uno, relay, sensor arus acs712, sensor tegangan zmpt101b, *node*, app inventor 2, thingspeak serta kaitannya dengan penelitian sebelumnya.

### BAB III: METODOLOGI PENELITIAN

Pada bab ini menceritakan metode dan perancangan yang akan dilakukan diantaranya lokasi dan waktu penelitian, alat dan bahan penelitian, spesifikasi rancangan sistem, prosedur penelitian, tahap persiapan, tahap pembuatan alat yang terdiri dari tahap perencanaan hardware dan tahap perencanaan software dan yang terakhir uji performa.

### BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini disajikan hasil serta analisis dari rancangan sistem manajemen energi berbasis *Internet of Things* menggunakan modulasi LoRa. Data hasil pengujian disajikan dalam bentuk tabel dan grafik.

### BAB V : KESIMPULAN DAN SARAN

Pada bab ini diberikan kesimpulan dan saran hasil penelitian serta analisis sistem manajemen energi berbasis *Internet of Things* menggunakan modulasi LoRa.

### DAFTAR PUSTAKA

## BAB II

### TINJAUAN PUSTAKA

#### **2.1. *Internet of Things (IoT)***

IoT merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus yang memungkinkan kita untuk menghubungkan mesin, peralatan, dan benda fisik lainnya dengan sensor jaringan dan aktuator untuk memperoleh data dan mengelola kinerjanya sendiri, sehingga memungkinkan mesin untuk berkolaborasi dan bahkan bertindak berdasarkan informasi baru yang diperoleh secara independen [10].

IoT adalah sebuah gagasan dimana semua benda di dunia nyata dapat berkomunikasi satu dengan yang lain sebagai bagian dari satu kesatuan sistem terpadu menggunakan jaringan internet sebagai penghubung. Misalnya CCTV yang terpasang di sepanjang jalan dihubungkan dengan koneksi internet dan disatukan di ruang kontrol yang jaraknya mungkin puluhan kilometer. atau sebuah rumah cerdas yang dapat dimanage lewat *smartphone* dengan bantuan koneksi internet. pada dasarnya perangkat IoT terdiri dari sensor sebagai media pengumpul data, sambungan internet sebagai media komunikasi dan server sebagai pengumpul informasi yang diterima sensor dan untuk analisa [10].

Ide awal IoT pertama kali dikenalkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai menggeluti IoT sebut saja Intel, Microsoft, Oracle, dan banyak lainnya. Banyak yang memprediksi bahwa pengaruh IoT adalah "*the next big thing*" di dunia teknologi informasi, hal ini disebabkan IoT menawarkan banyak potensi yang bisa digali. Contoh sederhana dari manfaat dan implementasi dari IoT yaitu sebuah kulkas yang dapat memberitahukan kepada pemiliknya via SMS atau email tentang makanan dan minuman apa saja yang sudah habis dan harus distok lagi [10].

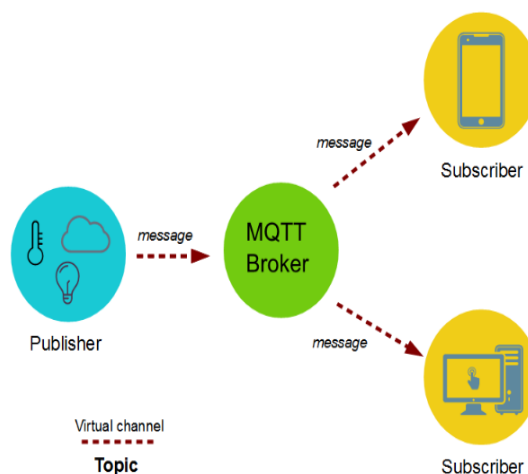


Gambar 2.1 Gambaran konsep IoT

Sumber : myspsolution.com

## 2.2. MQTT Protocol

MQTT yaitu protokol untuk komunikasi yang bersifat *machine to machine* atau M2M dan bekerja di layer ketujuh atau aplikasi dan bersisifat *lightweight message*. Meskipun koneksi dalam keadaan terputus, semua pesan yang dikirim akan terjamin oleh protokol MQTT. Metode komunikasi *publish/subscribe* merupakan metode pengiriman yang digunakan oleh protokol MQTT. Pesan pada MQTT dikirim ke *broker* dan berisi topik yang dikirimkan oleh *publisher*. Kemudian topik tadi diolah untuk diteruskan ke *subscriber* berdasarkan dari permintaan pengguna [11].



Gambar 2.2 Konsep Protocol MQTT untuk IoT

Sumber : matematika.fst.unair.ac.id

### 2.3. Manajemen Energi Listrik

Manajemen energi adalah kegiatan di suatu perusahaan yang terorganisir dengan menggunakan prinsip-prinsip manajemen, dengan tujuan agar dapat dilakukan konservasi energi, sehingga biaya energi sebagai salah satu komponen biaya produksi/operasi dapat ditekan serendah-rendahnya [12].

Pada penelitian ini manajemen energi yang dimaksud adalah *monitoring* serta *controlling* energi listrik berbasis IoT. Sehingga tidak termasuk audit energi maupun konservasi energi.

### 2.4. LoRa (*Long Range*)

LoRa (*Long Range*) merupakan suatu teknologi komunikasi secara *wireless* yang menggunakan frekuensi radio sebagai media trasmisinya. Teknologi ini mulai banyak dipergunakan sejak tahun 2012 ketika diakuisisi oleh Semtech Teknologi ini mampu menawarkan suatu komunikasi dengan jarak yang terbilang jauh, yaitu  $\pm 10$  km di wilayah urban [13].

Dengan menggunakan satu penerima di jaringan LoRa, ia dapat menangani banyak *node* di beberapa lokasi di dalam area, tidak seperti sistem berbasis Wi-Fi yang perlu memiliki banyak titik akses untuk meningkatkan area jangkauan. Kombinasi teknologi LoRa dan Wi-Fi mengurangi biaya penyebaran sistem IoT [14].

### 2.5. LoRa Gateway (Dragino LG01-N)

LoRa *gateway* merupakan jembatan penghubung antara LoRa *node* dengan *cloud server* [15]. Modul ini memiliki 4 frekuensi, yaitu 433MHz, 868MHz, 915MHz, dan 920MHz.

Berdasarkan tata letak penggunaannya, modul ini dibagi menjadi dua jenis yaitu bertipe *indoor* dan *outdoor*. Kemudian berdasarkan frekuensi yang dapat diterima, modul ini juga terbagi menjadi dua jenis yaitu bertipe *single channel* dan *dual channel*. Pada penelitian ini, akan digunakan Dragino LoRa Gateway LG01-N. Gateway ini menyediakan antarmuka WiFi, port Ethernet dan port USB. Gateway ini diklaim memiliki jangkauan hingga 5-10 km [16].



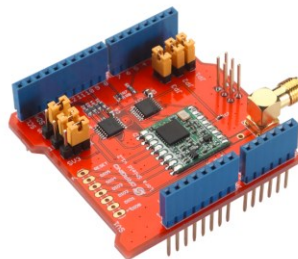
Gambar 2.3 LG01-N LoRa IoT Gateway

Sumber : dragino.com

## 2.6. Dragino LoRa Shield 915 MHZ

LoRa *shield* adalah transceiver jarak jauh yang didasarkan pada *open source library*. Lora *shield* memungkinkan pengguna untuk mengirim data dan mencapai rentang yang sangat panjang pada data yang rendah. LoRa *shield* menyediakan komunikasi spektrum penyebaran jarak jauh dan gangguan imunitas sambil meminimalkan konsumsi energi.

LoRa *shield* didasarkan pada *chip* Semtech SX1276 / SX1278, ia menargetkan aplikasi jaringan sensor nirkabel profesional seperti sistem irigasi, pengukuran cerdas, kota pintar, deteksi ponsel cerdas, otomatisasi bangunan, dan sebagainya [17].



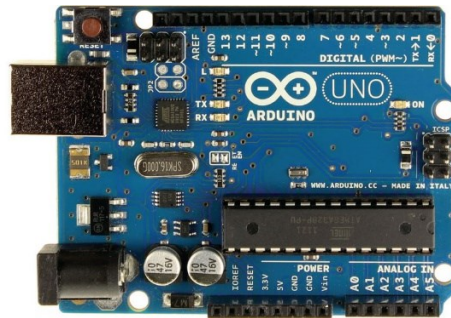
Gambar 2.4 Dragino LoRa Shield 915 MHZ

Sumber : dragino.com

## 2.7. Arduino Uno

Arduino ini merupakan sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya. ATmega328 pada Arduino Uno hadir dengan sebuah

*bootloader* yang memungkinkan kita untuk mengupload kode baru ke ATmega328 tanpa menggunakan pemrogram *hardware* eksternal [18].



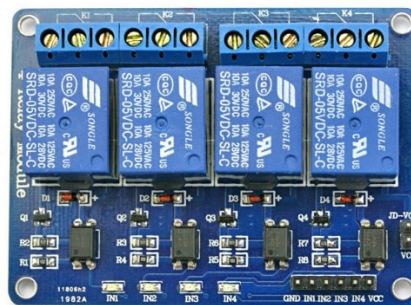
Gambar 2.5 Arduino Uno

Sumber : [electricityofdream.com](http://electricityofdream.com)

## 2.8. Relay

Relay merupakan saklar elektronik yang dapat membuka atau menutup rangkaian dengan menggunakan kontrol dari rangkaian elektronik lain. Sebuah relay tersusun atas kumparan, pegas, saklar (terhubung pada pegas) dan 2 kontak elektronik (*normally close* dan *normally open*) [19].

Berdasarkan prinsip dasar cara kerjanya, relay dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan kerja relay maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak NC ke kontak NO. jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak NC [19].



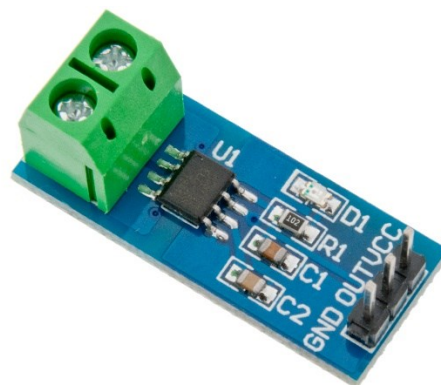
Gambar 2.6 Relay Module 4 Channel

Sumber : [electronics-lab.com](http://electronics-lab.com)



## 2.9. Sensor Arus ACS712

Sensor arus ACS-712 adalah solusi untuk pembacaan arus didalam dunia industri, otomotif, komersil dan sistem-sistem komunikasi. Sensor ini biasanya digunakan untuk mengontrol motor, deteksi beban listrik, *switchedmode power supplies* dan proteksi beban berlebih. Sensor ini memiliki pembacaan dengan ketepatan yang tinggi, karena didalamnya terdapat rangkaian *low-offset linear Hall* dengan satu lintasan yang terbuat dari tembaga. Cara kerja sensor ini adalah arus yang dibaca mengalir melalui kabel tembaga yang terdapat didalamnya yang menghasilkan medan magnet yang di tangkap oleh *integrated Hall IC* dan diubah menjadi tegangan proporsional. Ketelitian dalam pembacaan sensor dioptimalkan dengan cara pemasangan komponen yang ada didalamnya antara penghantar yang menghasilkan medan magnet dengan *hall transducer* secara berdekatan. Persisnya, tegangan proporsional yang rendah akan menstabilkan *Bi CMOS Hall IC* yang didalamnya yang telah dibuat untuk ketelitian yang tinggi oleh pabrik. Dimana titik tengah *output* sensor sebesar ( $>VCC/2$ ) saat peningkatan arus pada penghantar arus yang digunakan untuk pendeteksian. Hambatan dalam penghantar sensor sebesar  $1,5m\Omega$  dengan daya yang rendah. Ketebalan penghantar arus didalam sensor sebesar 3x kondisi *overcurrent*. Sensor ini telah dikalibrasi oleh pabrik [20].



Gambar 2.7 Sensor Arus ACS712

Sumber :electrokit.com

### 2.10. Sensor Tegangan ZMPT101B

Pengukuran tegangan AC dapat dilakukan dengan cara dirubah menjadi DC agar lebih mudah dibaca oleh mikrokontroler. Mikrokontroler yang dilengkapi dengan ADC (*Analog to Digital Converter*) tidak dapat membaca sinyal negatif, maka dari itu tegangan negatif harus dinaikkan *offset*-nya menjadi 2,5 V sehingga terdapat perbedaan antara nilai negatif dan positif. Sensor tegangan ZMPT101B telah dilengkapi *summing amplifier* sehingga dapat digunakan untuk menaikkan tegangan negatif sehingga baik untuk pengukuran tegangan dengan menggunakan mikrokontroler [21]. Berikut merupakan gambar fisik dari sensor tegangan ZMPT101B yang ditunjukkan pada Gambar 2.8.

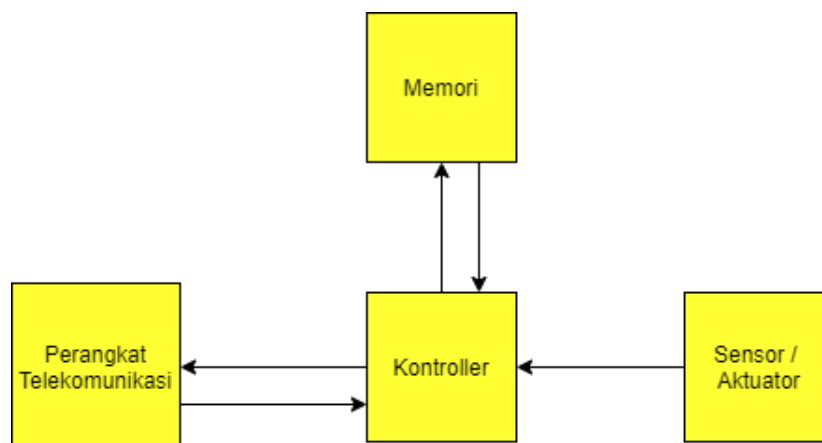


Gambar 2.8 Sensor Tegangan ZMPT101B

Sumber : [forum.fritzing.org](http://forum.fritzing.org)

### 2.11. Node

Pada intinya sebuah sensor node terdiri dari komponen pengendali (kontroler), sensor/aktuator, memori, perangkat komunikasi dan catu daya (*power supply*). Karena komponen-komponenya tersebut maka sensor node pada penelitian ini disebut juga dengan *smart/intelligent sensor*. Gambar 2.9 berikut menggambarkan sebuah node beserta komponen pendukungnya.



Gambar 2. 9 Node

Sebuah pengendali akan memproses semua data yang relevan dan berkemampuan untuk mengeksekusi semua kode-kode, sedangkan beberapa memori digunakan untuk menyimpan program dan data *intermediate* yang nantinya akan dikirimkan ke *controller board (getaway)*. Sensor dan aktuator merupakan *interface* terhadap parameter-parameter fisik dari lingkungan. Perangkat komunikasi digunakan sebagai peralatan jaringan dalam mengirimkan dan menerima informasi melalui kanal nirkabel. *Power supply* digunakan sebagai penyimpan energi untuk mengaktifkan komponen-komponen utama yang lainnya.

### 2.12. App Inventor2

App Inventor 2 (AI2) adalah aplikasi web sumber terbuka yang awalnya dikembangkan oleh Google, dan saat ini dikelola oleh Massachusetts Institute of Technology (MIT). App Inventor merupakan sebuah pemrograman yang menghasilkan aplikasi yang dapat digunakan di sistem Android. AI2 berbasis *cloud* yang diakses menggunakan internet browser. Keuntungan dari App Inventor terletak pada kemudahan dalam pemrograman karena pengguna tidak perlu memiliki pengetahuan dasar programmer, memahami kode, atau memiliki pengalaman dalam TI. Hal yang paling penting dalam membuat aplikasi menggunakan App Inventor adalah bagaimana programmer menggunakan logikanya seperti ketika seseorang menyusun teka-teki [22].

### 2.13. Thingspeak

ThingSpeak adalah platform IoT terbuka yang mendukung Matlab. Inti dari *platform* ini adalah IoT analytics dan visualisasi data. Menyediakan visualisasi data secara *real-time* dan dengan dukungan matlab, memungkinkan untuk menambahkan data untuk keperluan analisis dan pemrosesan. Berapa fitur utama yang dimiliki oleh ThingSpeak adalah sebagai berikut:

- Visualisasi data sensor secara *realtime*
- Agregasi data dari penyedia pihak ketiga
- *Task IoT analytics* terjadwal guna menganalisis data
- *Event Scheduling*
- Menjalankan aksi berdasarkan data yang diterima

Platform ini mendukung beberapa perangkat misalnya Arduino, Raspberry Pi, hingga ESP. Pembaca bisa mengunjungi situsnya di ThingSpeak.com untuk mendapatkan akun dan mulai membangun *project*.

#### 2.14. RSSI

RSSI merupakan parameter pengukuran yang digunakan untuk mengukur kualitas penerimaan sinyal yang diperoleh. RSSI diukur pada sisi penerima (*RECEIVER*) pada saat sedang melakukan komunikasi dengan pengirim (*transmitter*). Pengukuran kekuatan sinyal menggunakan unit pengukuran disebut decibel miliwatt atau dBm. Decibel merupakan unit sederhana yang berhubungan antara dua pengukuran daya. Berdasarkan definisi decibel miliwatt maka 0 dBm = 1 mW. Jika nilai daya lebih besar dari 1 mW maka nilai dari dBm bernilai positif, sedangkan untuk nilai dibawah 1 mW bernilai negatif [23].

Nilai RSSI dibagi menjadi beberapa level kategori seperti yang ditunjukkan pada tabel di bawah ini:

Tabel 2.1 Level sinyal RSSI

Level RSSI (dBm)	Keterangan
-30 s.d -60	Sangat kuat. Jarak antara penerima dan pemancar dekat
-60 s.d -90	Sangat baik. Cakupan dekat.
-90 s.d -105	Baik. Terdapat beberapa data yang tidak diterima.
-105 s.d -115	Buruk. Dapat menerima tetapi sering <i>drop out</i>
-115 s.d -120	Sangat lemah. Data sering hilang.

## 2.15. Penelitian Terdahulu

Tabel 2.2 Penelitian Terdahulu

No	Nama Peneliti	Judul	Keterangan
1.	Putri Mandarani [24]	Perancangan Dan Implementasi <i>User Interface</i> Berbasis Web Untuk <i>Monitoring</i> Suhu, Kelembaban Dan Asap Pada Ruang Berbeda Dengan Memanfaatkan Jaringan <i>Local Area Network</i>	Pada paper ini dirancang sebuah sistem <i>monitoring</i> yang bertujuan untuk memantau kondisi beberapa ruangan dengan memanfaatkan jaringan LAN. <i>Hardware</i> yang digunakan adalah Arduino Uno, Ethernet <i>Shield</i> , sensor DHT11 (temperatur dan kelembaban) dan sensor MQ-9 (gas <i>analog</i> ). Sedangkan <i>software</i> yang digunakan adalah Arduino versi 1.0.5 untuk pemograman mikrokontroler Arduino Uno dan XAMPP yang berisi modul Apache, PHP dan Mysql.
2.	Wahri Sunanda, Irwan Dinata [25]	Penerapan Perangkat <i>Wireless Monitoring</i> Energi Listrik Berbasis Arduino dan Internet	Pada paper ini dilakukan perancangan perangkat yang terdiri dari 4 (empat) bagian yaitu sensor, <i>processor</i> , <i>display</i> dan <i>network</i> . Bagian sensor terdiri dari <i>Current Transformer</i> dan <i>AC to AC Power Adapter</i> . <i>Processor</i> digunakan Arduino UNO yang akan mengolah hasil sensor, <i>display</i> menggunakan <i>Liquid Crystal Display</i> (LCD) untuk menampilkan data keluaran <i>real time</i> . Bagian terakhir yaitu <i>network</i> terdiri dari Ethernet <i>Shield</i> , 3G Router, 3G Modem

			untuk komunikasi ke <i>Database Server</i> sebagai tempat penyimpanan tetap dan pengolahan data lebih lanjut. Melalui perangkat ini kita dapat memantau penggunaan energi listrik setiap saat tanpa harus mendatangi lokasi titik pengukuran tersebut.
3.	Temy Nusa, Sherwin R.U.A. Sompie, ST.,MT., Dr.Eng Meita Rumbayan, ST.,MT [26]	Sistem <i>Monitoring</i> Konsumsi Energi Listrik Secara <i>Real Time</i> Berbasis Mikrokontroler	Pada paper ini dirancang alat untuk memonitor konsumsi energi listrik ini memanfaatkan transformator <i>step-down</i> untuk mengukur tegangan sumber dari PLN, sementara untuk mengukur arus beban memanfaatkan sensor arus ACS712 dan mikrokontroler ATmega 328 buatan ATMEL, difungsikan untuk mengolah semua data dari parameter – parameter yang dibutuhkan untuk mendapatkan nilai konsumsi energi listrik, serta menampilkannya pada LCD karakter 20x4 untuk memberikan informasi kepada pengguna listrik.
4.	Totok Budioko [27]	Sistem <i>Monitoring</i> Suhu Jarak Jauh Berbasis <i>Internet</i> <i>Of Things</i> Menggunakan Protokol MQTT	Pada paper ini dibahas implementasi protokol MQTT untuk sistem <i>monitoring</i> suhu jarak jauh. Implementasi sistem menggunakan sensor suhu LM35, Arduino UNO dan modul wifi Esp8266 ver 01.

5.	Sri Suryaningsih, Sahrul Hidayat, Faisal Abid [28]	Rancang Bangun Alat Pemantau Penggunaan Energi Listrik Rumah Tangga Berbasis Internet	Pada paper ini dirancang alat pemantau dan penghitung jarak jauh penggunaan daya listrik pada suatu alat elektronik maupun keseluruhan penggunaan daya listrik pada rumah tangga dengan menggunakan transmisi data media internet, dimana daya listrik yang terpakai bisa ditampilkan pada halaman web.
----	--	---	---

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di Laboratorium *Internet of Things* dan Laboratorium Konversi Energi & Sistem Tenaga Listrik, Program Studi Teknik Elektro, Fakultas Teknik, Universitas Sebelas Maret, Surakarta. Sedangkan waktu penelitian dilaksanakan pada bulan Maret – Juli 2020.

#### 3.2. Alat dan Bahan Penelitian

Tabel 3.1 Tabel Alat dan Bahan Penelitian

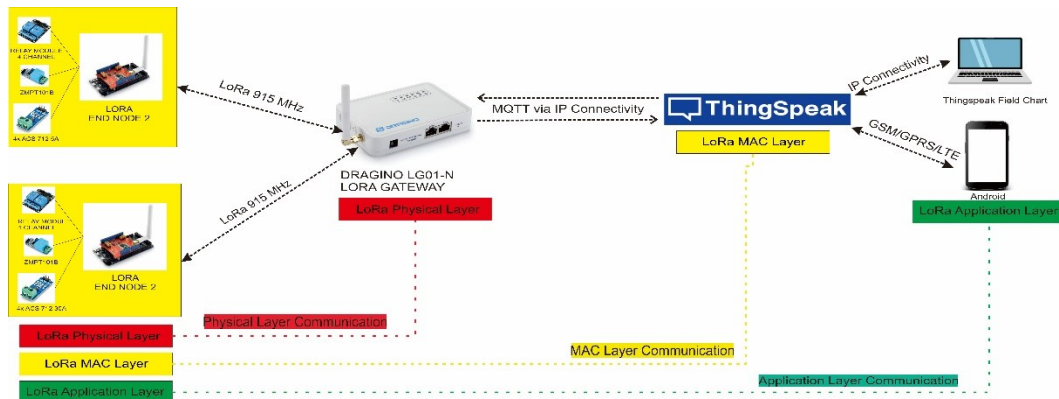
No.	Nama Alat / Spesifikasi	Kuantitas	Keterangan
1	LG01-N LoRa IoT Gateway	1	Sebagai penghubung antara LoRa <i>end node</i> dengan Internet.
2	Dragino LoRa Shield 915 MHZ	2	Sebagai penghubung arduino dengan LoRa Gateway.
3	Arduino UNO R3	2	Sebagai mikrokontroler yang mengolah nilai masukan dari sensor.
4	Sensor Arus ACS712	8	Untuk mengukur arus pada beban.
5	Sensor Tegangan ZMPT101B	2	Untuk mengukur tegangan pada beban.
6	Relay Module 4 Channel	2	Untuk mengontrol beban pada yang digunakan.
7	Kabel Jumper	Secukupnya	Untuk menghubungkan antara sensor, relay ke arduino.
8	Project Board	2	Untuk melakukan prototyping rangkaian.



9	Tenol	5m	Untuk menyatukan komponen pada PCB.
10	Solder listrik	1	Peleleh tenol.
11	Laptop	1 unit	Laptop yang sudah terinstall Arduino IDE
12	<i>Software</i> Arduino IDE	1	Untuk menuliskan program pada Arduino.
13	Website Thingspeak	1	Untuk menampilkan data dari <i>end node</i> .
14	Website MIT App Inventor	1	Untuk membuat aplikasi android.
15	Multimeter	1	Untuk melakukan pengukuran.

### 3.3. Spesifikasi Rancangan Sistem

Terdapat 2 buah *end node* yang dirancang dalam penelitian ini. Masing-masing *end node* terdapat 4 buah sensor arus ACS712 dan sebuah sensor tegangan ZMPT101B yang digunakan dalam mengukur besaran arus dan tegangan pada beban. Dalam masing-masing node juga terdapat 1 buah relay module 4 channel yang digunakan untuk mengontrol beban yang digunakan. Kedua sensor dan relay ini dihubungkan dengan mikrokontroller Arduino UNO yang terintegrasi dengan LoRa *shield* Dragino untuk melakukan komunikasi dengan LoRa *gateway* yang akan menghubungkan ke *server* Thingspeak melalui internet. Data arus dan tegangan kemudian akan diolah oleh Arduino UNO supaya didapatkan hasil energi listrik dalam satuan *Watt Hour* (Wh). Kedua data ini kemudian dikirimkan ke IoT cloud server yaitu Thingspeak. Terakhir, hasilnya akan ditampilkan berupa grafik pada *field* di Thingspeak serta di Android melalui aplikasi BEMS yang dibuat melalui MIT App Inventor. Beban pada node juga bisa dikontrol melalui aplikasi android yang telah dibuat.



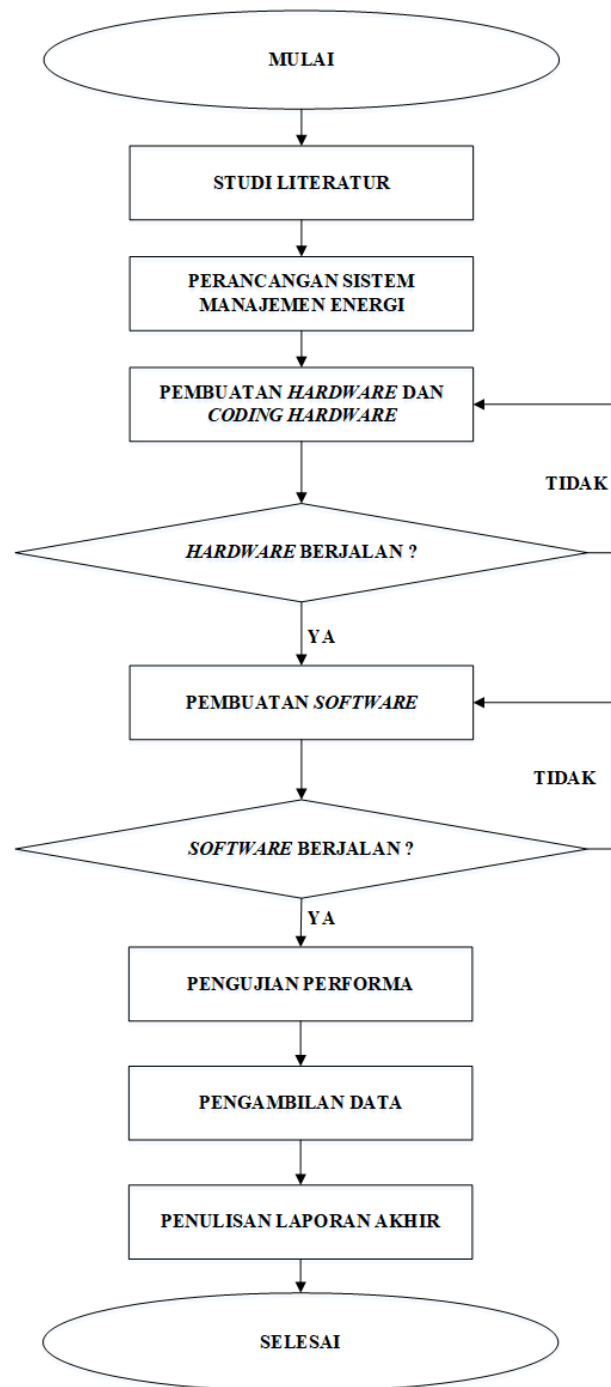
Gambar 3.1 Arsitektur Sistem Manajemen Energi dengan LoRa

### 3.4. Prosedur Penelitian

Penelitian ini dimulai dengan studi literatur untuk memahami prinsip kerja cara pembuatan sistem manajemen energi berbasis IoT menggunakan LoRa. Referensi yang dipakai berupa paper, laporan penelitian terdahulu yang sejenis, buku serta *datasheet*.

Setelah konsep dipahami maka selanjutnya adalah pembuatan *hardware* dan *software* sistem manajemen energi berbasis IoT menggunakan LoRa. Pembuatan *hardware* terdiri dari *monitoring* dan *control*. Pada sisi *monitoring*, parameter yang dimonitor adalah energi listrik dalam satuan watt hour (wh) sehingga dilakukan perancangan sistem dengan menggunakan sensor arus ACS712 dan sensor tegangan ZMPT101B. Pada sisi *control*, dilakukan perancangan dengan menggunakan relay 4 channel yang digunakan untuk menghubungkan atau memutus beban dari sumber listrik. Sedangkan pada perancangan *software*, dilakukan perancangan aplikasi android yang berfungsi untuk memonitor energi listrik dan kontrol peralatan listrik. Perancangan dilakukan dengan menggunakan MIT App Inventor.

Serangkaian pengujian dilakukan untuk melihat performa dari sistem yang dirancang. Pada pengujian sisi *monitoring* untuk memastikan bahwa nilai yang ditampilkan mendekati nilai sebenarnya maka dilakukan uji presisi dan uji akurasi. Sedangkan pada sisi *control* dilakukan pengujian lama waktu tunda (*delay*) yang dibutuhkan oleh relay untuk mengeksekusi perintah yang diberikan.



Gambar 3.2 Prosedur Penelitian

### 3.5. Tahap Persiapan

Tahap persiapan merupakan tahapan awal dalam melakukan penelitian, pada tahap ini penulis melakukan studi literatur dengan mencari berbagai acuan baik melalui buku, jurnal, tugas akhir maupun artikel dengan narasumber yang jelas dan terpercaya dengan tujuan untuk melengkapi literatur mengenai penelitian ini. Penulis juga melakukan identifikasi masalah pada penelitian ini, membaca dan

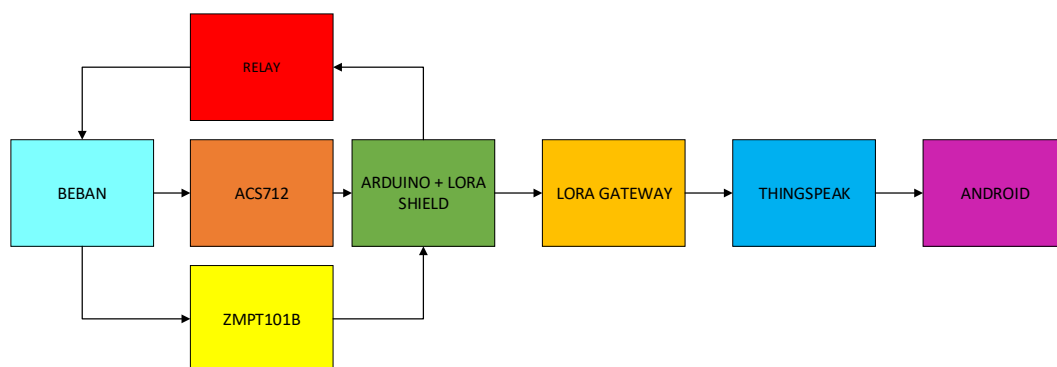
memahami kelebihan dan kekurangan dari penelitian yang sudah dilakukan sebelumnya. Dan juga penulis menyiapkan alat dan bahan yang diperlukan dalam penelitian ini untuk mempersiapkan menuju ke tahap selanjutnya.

### 3.6. Tahap Perancangan *Hardware*

Sistem alat yang akan dibuat meliputi sensor arus ACS712 yang berfungsi untuk membaca arus dan sensor tegangan ZMPT101B yang berfungsi untuk membaca tegangan pada tiap beban yang digunakan. Sistem bekerja sesuai dengan blok diagram pada gambar 3.3. Inputan yang digunakan adalah data arus yang dibaca oleh sensor ACS712 dan data tegangan yang dibaca oleh ZMPT101B.

Tahap perancangan *hardware* juga meliputi pembuatan program untuk mengeksekusi rancangan *hardware* yang telah dibuat. Arduino IDE digunakan untuk mengeksekusi dan mengupload program ke mikrokontroler Arduino Uno.

Pada sensor ACS712 dan ZMPT101B yang digunakan, secara umum menggunakan bahasa C untuk membaca arus dan tegangan pada tiap tiap beban listrik yang digunakan. Data arus dan tegangan yang telah dibaca kemudian diolah oleh Arduino Uno menjadi data energi. Kemudian data akan dikirim ke Thingspeak untuk ditampilkan. Relay juga akan diprogram dengan bahasa C untuk digunakan sebagai kontrol terhadap beban elektronik yang digunakan.



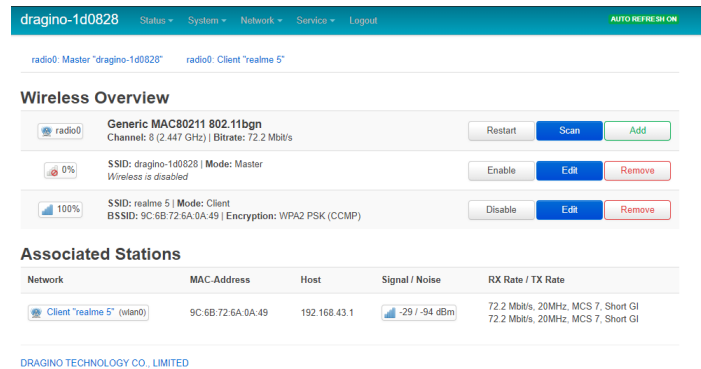
Gambar 3.3 Blok Diagram

Pada gambar diatas dapat dilihat bahwa terdapat relay yang menghubungkan antara Arduino Uno dengan beban yang berfungsi sebagai kontrol arus yang masuk ke beban. Arus & tegangan hasil pembacaan sensor ini akan dikirim oleh menuju ke *server cloud* IoT melalui *gateway*.

### 3.6.1. Konfigurasi LoRa Gateway

Pada LoRa gateway dilakukan konfigurasi supaya bisa menjadi jembatan antara komunikasi LoRa dengan Internet. Langkah – langkah yang dilakukan adalah sebagai berikut:

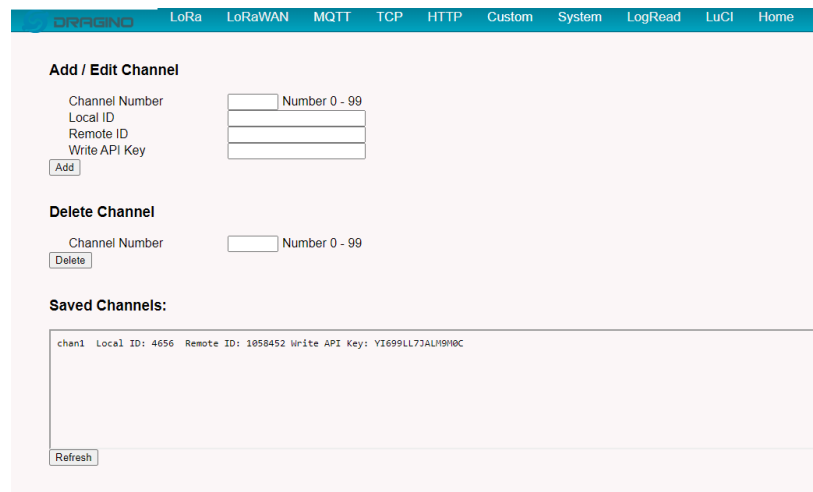
1. Menghubungkan gateway dengan koneksi WiFi. Hal ini bertujuan supaya gateway memiliki koneksi ke Internet.



Gambar 3.4 Setting network gateway

Dari gambar diatas dapat dilihat bahwa gateway sudah dapat terhubung dengan WiFi.

2. Mengatur client



Gambar 3.5 Setting client

Tujuan dari mengatur client seperti gambar diatas adalah supaya id end node dapat dikenali dan terhubung ke server Thingspeak.

### 3. Mengatur LoRa

Gambar 3.6 LoRa *setting*

Langkah diatas merupakan langkah untuk mengatur LoRa supaya bekerja sesuai spesifikasi yang diinginkan.

### 4. Mengatur *cloud server*

Gambar 3.7 *Setting cloud server*

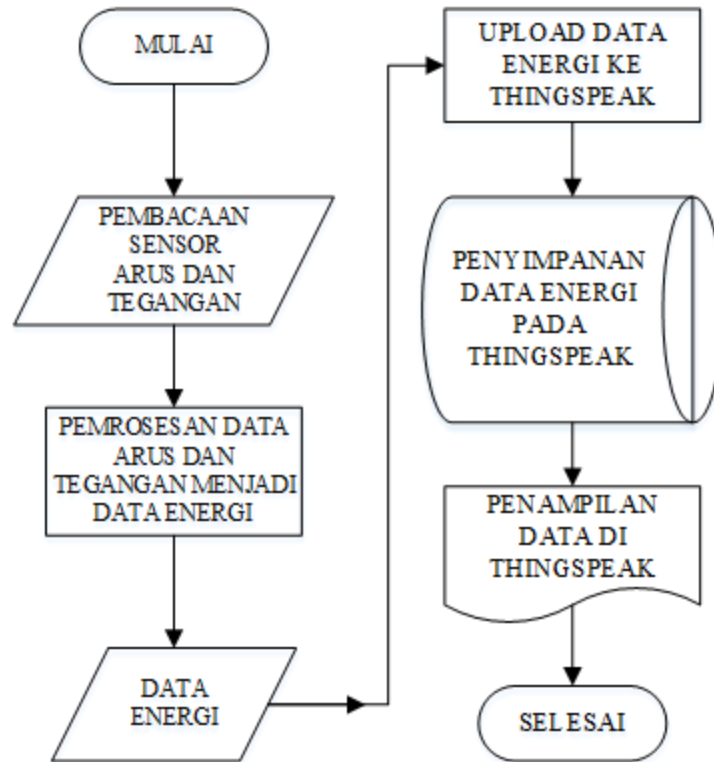
*Cloud server* disini digunakan sebagai tempat untuk *upload* data yang dikirim oleh LoRa *end node*.

#### 3.6.2. Perancangan Sistem *Monitoring*

Pada perancangan ini, algoritma dimulai saat ACS712 dan ZMPT101B membaca nilai dan tegangan. Dari data arus dan tegangan tersebut kemudian akan diolah menjadi data energi listrik dalam satuan *watt hour* (Wh) dengan persamaan:

$$\text{Energi (Wh)} = V \times I \times 0.85 \times t / 3600 \quad (3.1)$$

Dimana 0.85 merupakan nilai faktor daya standard PLN, sedangkan t adalah lama waktu penggunaan dalam satuan detik. Berikut merupakan diagram alir algoritma perancangan sistem *monitoring*.

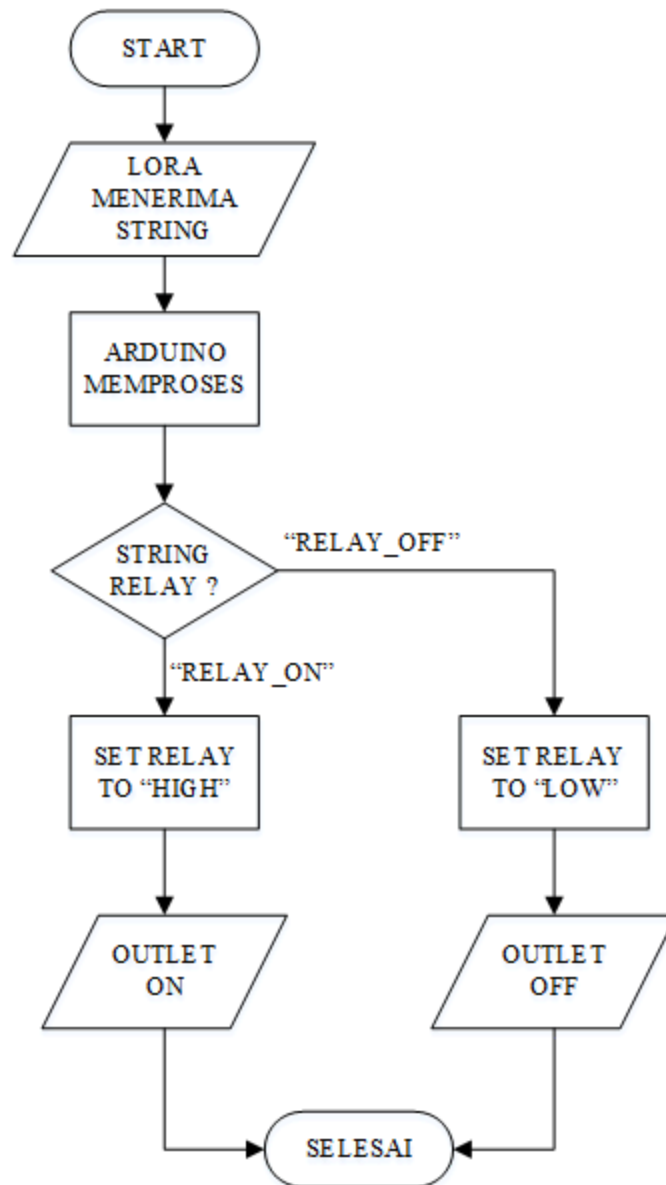


Gambar 3.8 Diagram alir sistem *monitoring*

### 3.6.3. Perancangan Sistem Kontrol

Pada perancangan sistem kontrol, *end node* akan menunggu *string* masukan yang dikirimkan dari Talkback Thingspeak. Dimana *string* yang dikirimkan berupa “RELAY[X]\_ON” dan “RELAY[X]\_OFF”. Dimana [X] disini akan berisi kode nomor relay. Relay nomor 1-4 digunakan sebagai id relay pada *end node* 1 sedangkan nomor relay 5-8 digunakan sebagai id relay pada *end node* 2.

Jika *end node* menerima *string* berupa kontrol on relay, maka akan diproses oleh arduino yang akan mengubah kondisi relay dari “LOW” ke “HIGH” yang akan menghidupkan outlet. Begitu pula sebaliknya, jika *end node* menerima *string* berupa kontrol off relay, maka akan diproses oleh arduino yang akan mengubah kondisi relay dari “HIGH” ke “LOW” yang akan mematikan outlet.



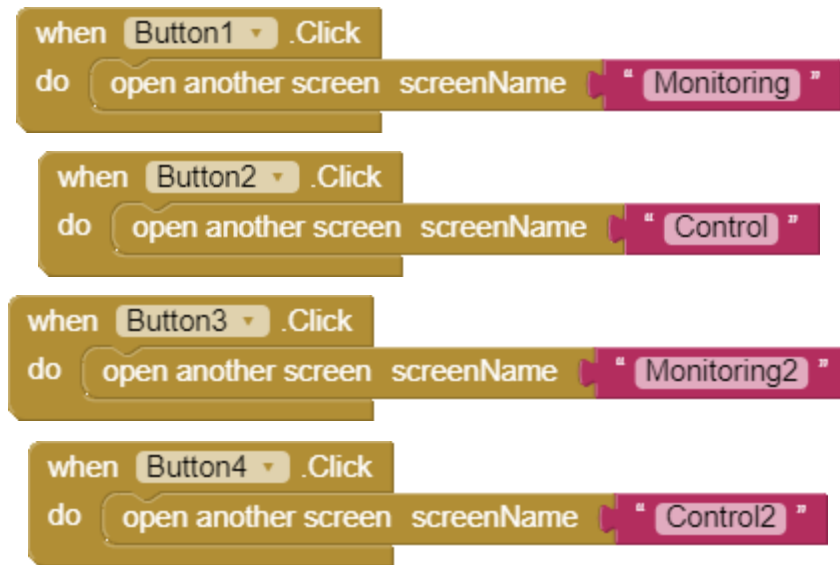
Gambar 3.9 Diagram Alir Sistem Kontrol

### 3.7. Tahap Perancangan *Software*

Perancangan aplikasi android menggunakan platform App Inventor 2 (AI2) yang dikembangkan oleh Massachusetts Institute of Technology. Penulis menggunakan platform ini dikarenakan kemudahan dalam menyusun aplikasi, dikarenakan perancangan aplikasi pada MIT App Inventor menggunakan *block – block* logika. Berikut ini merupakan *block – block* yang dibuat.

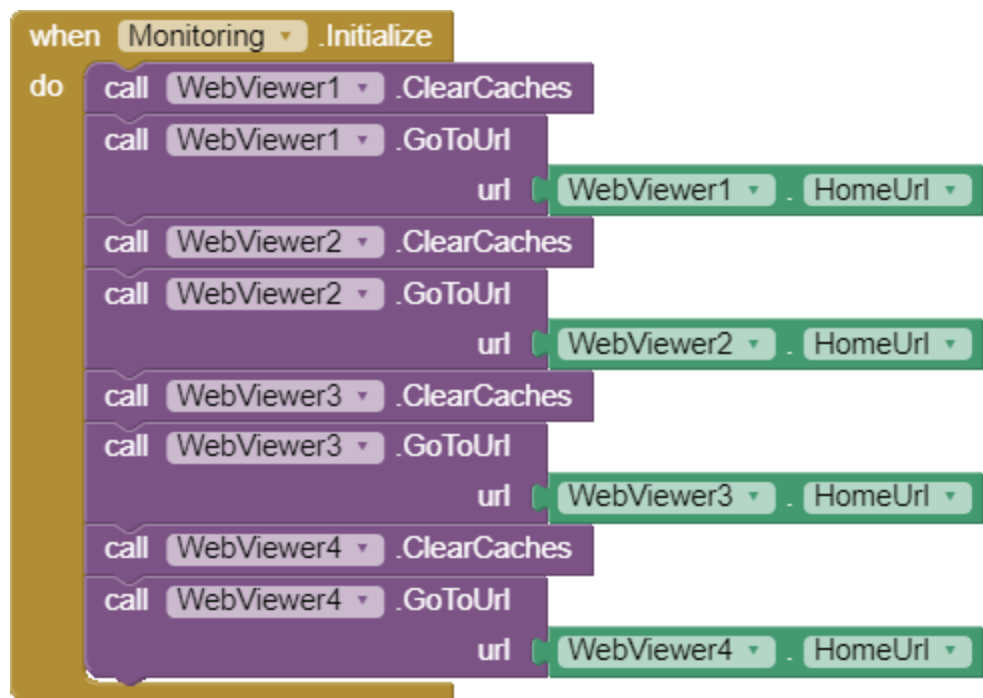


5. Pada menu *home*



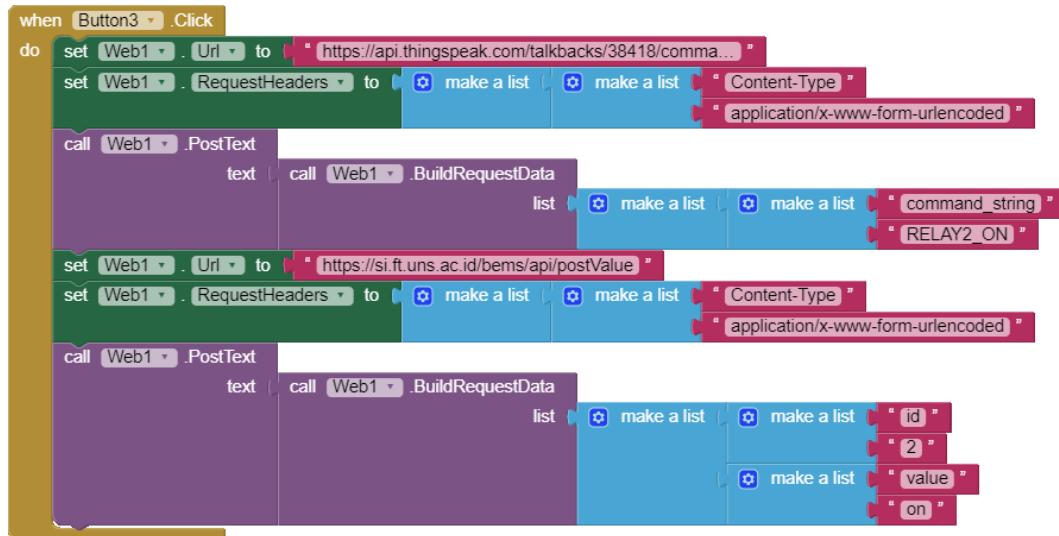
Gambar 3.10 Block menu home

6. Pada menu *monitoring*



Gambar 3.11 Block menu monitoring

## 7. Pada menu *control*



Gambar 3.12 Block menu control

### 3.8. Uji Performa

Tahap uji performa terdiri dari pengujian seluruh sistem alat yang sudah dibuat. Pengujian berupa pengambilan nilai presisi sensor, nilai akurasi sensor yang menunjukkan berapa error antara pengukuran multimeter dan hasil pembacaan sensor arus acs712 dan sensor tegangan zmpt101b. Serta dilakukan pula pengukuran waktu tunda relay terhadap masukan yang diberikan pada sisi kontrol. Selain itu, untuk mengetahui seberapa jauh jangkauan dan andal LoRa dapat bekerja, dilakukan pengujian jangkauan dan pengujian keandalan (*reliability*).

Berikut ini merupakan teknis pengujian yang dilakukan:

#### 3.8.1. Pengujian Presisi

Presisi merupakan parameter yang menunjukkan seberapa dekat hasil pengukuran ketika dilakukan pengukuran secara berulang-ulang.

Pada penelitian ini, untuk melakukan pengujian presisi dilakukan dengan cara mengambil 50 data hasil pembacaan sensor kemudian dari data tersebut dicari *standard deviation* dan *relative standard deviation* untuk menentukan nilai presisi. Dengan persamaan di bawah ini [29]:

$$SD = \sqrt{\frac{\sum Xi^2 - \frac{(\sum xi)^2}{n}}{n-1}} \quad (2.2)$$

Dimana:

- SD = Simpangan baku  
 $\Sigma X^2$  = Jumlah kuadrat pengukuran individu  
 $\Sigma Xi$  = Jumlah pengukuran individu  
 n = Jumlah sampel yang dianalisis

Setelah nilai standard deviasinya ditemukan maka selanjutnya mencari nilai RSD, dengan persamaan di bawah ini:

$$RSD = \frac{SD}{\bar{x}} \cdot 100\% \quad (2.3)$$

Dimana:

- RSD = Simpangan baku relatif  
 SD = Simpangan baku  
 $\bar{x}$  = nilai rata-rata pengukuran

Setelah itu, untuk mencari nilai presisi maka dilakukan dengan persamaan di bawah ini:

$$Presisi = 100\% - RSD \quad (2.4)$$

### 3.8.2. Pengujian Akurasi

Akurasi adalah suatu parameter yang menunjukkan seberapa dekat nilai pembacaan yang didapatkan dengan nilai yang dianggap benar (nilai sebenarnya).

Pada penelitian ini nilai yang didapatkan dari hasil pembacaan sensor dibandingkan dengan nilai yang dianggap benar yaitu nilai dari hasil pembacaan multimeter. Pada pengujian ini diambil 5 buah variasi beban untuk melihat seberapa akurasi hasil pembacaan sensor arus dan sensor tegangan. Perhitungan error dilakukan dengan mencari nilai *mean absolute percentage error* dengan persamaan sebagai berikut [29]:

$$MAPE = \frac{\frac{\sum |Xi - Fi|}{Xi}}{n} \cdot 100\% \quad (2.5)$$

$$Presisi = 100\% - MAPE \quad (2.6)$$

### 3.8.3. Pengujian Waktu Tunda Relay

Pengujian waktu tunda relay dilakukan dengan maksud untuk mengetahui seberapa cepat relay merespon masukan yang diberikan. Pada pengujian ini terdapat dua kondisi pengujian yaitu pada saat kondisi relay ON ke kondisi OFF serta pada saat relay kondisi OFF ke kondisi ON. Untuk mencari waktu tunda relay dilakukan dengan cara memberikan masukan “HIGH” atau “LOW” kepada relay, kemudian akan dihitung berapa lama waktu yang dibutuhkan relay untuk berubah kondisi dari “LOW” ke “HIGH” maupun sebaliknya.

### 3.8.4. Pengujian Jangkauan

Pengujian jangkauan dilakukan untuk mengetahui seberapa jauh LoRa dapat bekerja. Pada pengujian jangkauan ini, dilakukan dengan cara mengirimkan perintah kepada *end node* sejauh variasi jarak yang diinginkan, kemudian akan diambil sejumlah 10 nilai RSSI-nya untuk menentukan apakah sinyal yang diterima termasuk dalam kategori baik atau buruk. Variasi jarak serta lokasi pada pengujian ini ditunjukkan dibawah ini:

#### 1. Jarak 50 Meter

Pada pengujian jarak 50 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di Gedung IV FT UNS.

#### 2. Jarak 100 Meter

Pada pengujian jarak 100 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di Gedung II FT UNS.

#### 3. Jarak 150 Meter

Pada pengujian jarak 150 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di depan Lab Mekanika Tanah Gd. II FT UNS.

#### 4. Jarak 200 Meter

Pada pengujian jarak 200 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di Kantin Jawara FT UNS.

### 5. Jarak 250 Meter

Pada pengujian jarak 250 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di Jalan Pertigaan UPT Bahasa – FT- FEB.

### 6. Jarak 300 Meter

Pada pengujian jarak 300 Meter, *gateway* berada di dalam ruangan Lab IoT, Gedung III FT UNS sedangkan *end node* berada di FEB UNS.

Perhitungan RSSI dapat dilakukan dengan persamaan [30]:

$$RSSI = A - 10n \cdot \log d \quad (2.7)$$

Dimana:

A : RSSI dalam jarak 1 meter

d : Jarak (m)

n : *Path loss exponent*

Nilai n disesuaikan dengan kondisi pada tabel di bawah ini, yaitu:

Tabel 3.2 *Path Loss Exponent*

Kondisi Lingkungan	<i>Path Loss Exponent (n)</i>
<i>Free space</i>	2
<i>Urban area cellular radio</i>	2.7 s.d 3.5
<i>Shadowed urban cellular radio</i>	3 s.d 5
<i>In building line of sight</i>	1.6 s.d 1.8
<i>Obstructed in buildings</i>	4 s.d 6
<i>Obstructed in factories</i>	2 s.d 3

### 3.8.5. Pengujian Keandalan (*Reliability*)

Pengujian ini dilakukan dengan cara menghidupkan *hardware* secara terus menerus selama 24 jam. Kemudian selama 24 jam tersebut akan dilihat berapa banyak data (*entries*) yang masuk. Kemudian akan dicari nilai *packet loss* dengan persamaan sebagai berikut:

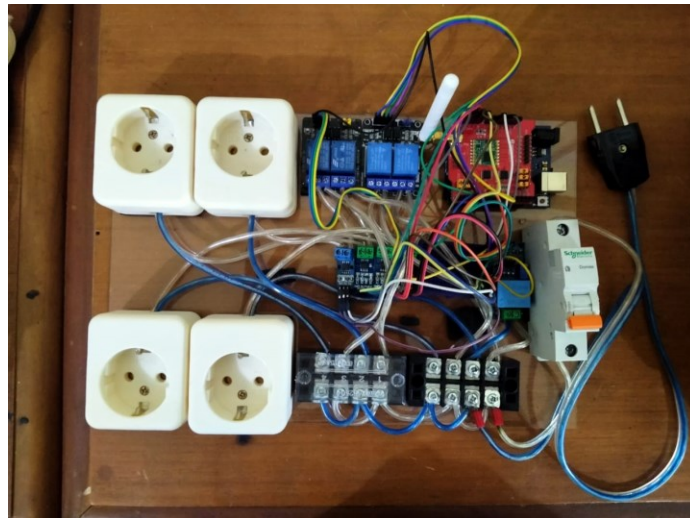
$$Packet\ loss = \frac{Expected\ Packet\ Received - Packet\ Received}{Expected\ Packet\ Received} \cdot 100\% \quad (2.8)$$

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Hasil Perancangan *End Node 1*

Perancangan *hardware end node 1* memiliki spesifikasi sebuah arduino uno sebagai mikrokontroller, LoRa *shield* dragino untuk komunikasi LoRa, 1 buah sensor tegangan ZMPT101B, 4 buah sensor arus ACS712 5 Ampere, relay module 4 channel yang terhubung ke 4 buah *output* berupa outlet serta dilengkapi dengan proteksi berupa MCB 4 Ampere. Node 1 pada penelitian ini masih berupa *prototype*. *End node 1* merepresentasikan pada lantai 1 Gedung III Fakultas Teknik UNS.



Gambar 4.1 *End Node 1*

##### 4.1.1. Hasil Uji Presisi Sensor Tegangan ZMPT101b

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor tegangan ZMPT101B. Data ini diambil sebanyak 50 sampel pada saat node 1 tidak memiliki beban.

Tabel 4.1 Hasil Pembacaan ZMPT101b 50 Kali

Pengukuran ke-	Tegangan Terbaca pada sensor ZMPT101B (V)
1	222.67
2	221.16
3	225.05
4	223.04
5	219.53

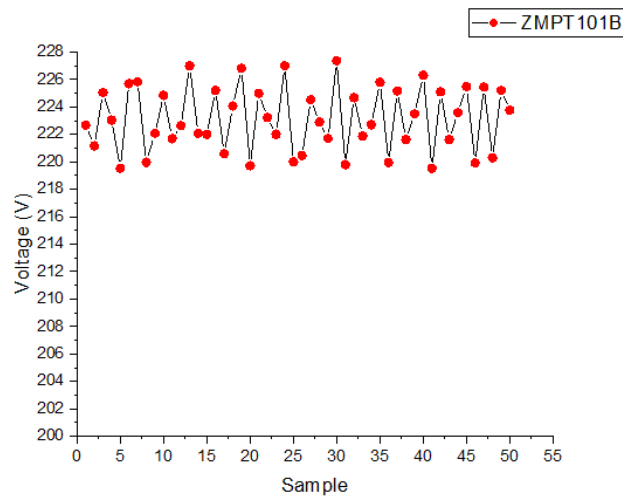
Tabel 4.1 Hasil Pembacaan ZMPT101b 50 Kali

<b>Pengukuran ke-</b>	<b>Tegangan Terbaca pada sensor ZMPT101B (V)</b>
6	225.70
7	225.84
8	219.96
9	222.08
10	224.85
11	221.69
12	222.64
13	227.00
14	222.07
15	222.00
16	225.20
17	220.59
18	224.06
19	226.82
20	219.71
21	224.99
22	223.23
23	222.00
24	227.00
25	220.00
26	220.46
27	224.53
28	222.90
29	221.70
30	227.37
31	219.79
32	224.67
33	221.88
34	222.71
35	225.80
36	219.95
37	225.16
38	221.62
39	223.50
40	226.31
41	219.52
42	225.11
43	221.62

Tabel 4.1 Hasil Pembacaan ZMPT101b 50 Kali

Pengukuran ke-	Tegangan Terbaca pada sensor ZMPT101B (V)
44	223.58
45	225.48
46	219.93
47	225.46
48	220.29
49	225.21
50	223.77
<b>SD</b>	<b>2.33</b>
<b>RSD</b>	<b>1.04</b>
<b>Presisi</b>	<b>98.95</b>

Dari 50 kali pengambilan data, dapat dilihat bahwa nilai standar deviasi sensor tegangan ZMPT101b memiliki nilai 2.33287. Adapun *nilai relative standard deviation* (RSD) memiliki nilai 1.04 %. Sehingga dapat disimpulkan bahwa sensor tegangan ZMPT101b pada *end node* 1 memiliki nilai presisi sebesar 98.95 %.



Gambar 4.2 Pembacaan ZMPT101B 50 Kali

#### 4.1.2. Hasil Uji Presisi Sensor Arus ACS712

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan 4 buah sensor Arus ACS712. Data ini diambil sebanyak 50 sampel pada beban yang berbeda. Sensor arus ACS712 pertama diberikan beban berupa kipas angin, sensor arus ACS712 kedua diberikan beban berupa panci elektrik pada level 1, sensor arus



ACS712 ketiga diberikan beban berupa panci elektrik pada level 2 dan sensor arus ACS712 keempat diberikan beban berupa setrika.

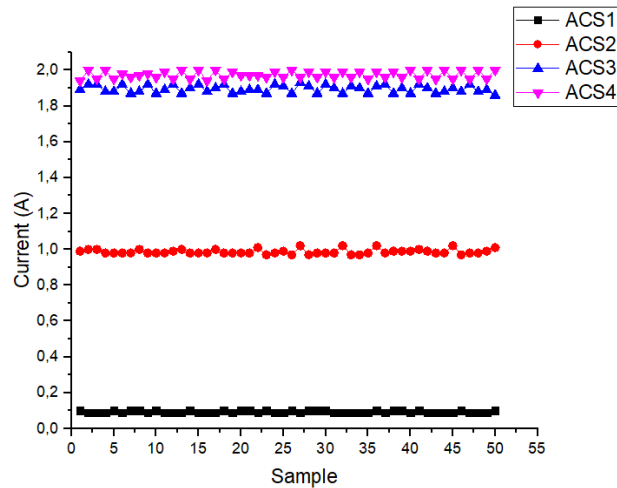
Tabel 4.2 Hasil Pembacaan ACS712 50 Kali

Pengukuran ke-	ACS1 (A)	ACS2 (A)	ACS3 (A)	ACS4 (A)
1	0.10	0.99	1.89	1.94
2	0.09	1.00	1.92	2.00
3	0.09	1.00	1.92	1.95
4	0.09	0.98	1.88	2.00
5	0.10	0.98	1.88	1.95
6	0.09	0.98	1.92	1.98
7	0.10	0.98	1.87	1.96
8	0.10	1.00	1.88	1.97
9	0.09	0.98	1.92	1.98
10	0.10	0.98	1.87	1.96
11	0.09	0.98	1.89	1.99
12	0.09	0.99	1.92	1.95
13	0.09	1.00	1.87	2.00
14	0.10	0.98	1.9	1.95
15	0.09	0.98	1.92	2.00
16	0.09	0.98	1.88	1.94
17	0.09	1.00	1.9	2.00
18	0.10	0.98	1.92	1.95
19	0.09	0.98	1.87	1.99
20	0.10	0.98	1.88	1.97
21	0.10	0.98	1.89	1.97
22	0.09	1.01	1.89	1.97
23	0.10	0.97	1.87	1.96
24	0.09	0.98	1.92	1.99
25	0.09	0.99	1.91	1.96
26	0.10	0.97	1.87	2.00
27	0.09	1.02	1.93	1.96
28	0.10	0.97	1.91	1.99
29	0.10	0.98	1.87	1.96
30	0.10	0.98	1.92	1.99
31	0.09	0.98	1.9	1.96
32	0.09	1.02	1.87	1.99
33	0.09	0.97	1.91	1.96
34	0.09	0.97	1.9	1.99

Tabel 4.2 Hasil Pembacaan ACS712 50 Kali

Pengukuran ke-	ACS1 (A)	ACS2 (A)	ACS3 (A)	ACS4 (A)
35	0.09	0.98	1.87	1.95
36	0.10	1.02	1.91	1.99
37	0.09	0.98	1.92	1.96
38	0.10	0.99	1.87	1.99
39	0.10	0.99	1.9	1.96
40	0.09	0.99	1.87	2.00
41	0.10	1.00	1.92	1.95
42	0.09	0.99	1.9	2.00
43	0.09	0.98	1.87	1.95
44	0.09	0.98	1.88	2.00
45	0.09	1.02	1.9	1.95
46	0.10	0.97	1.88	2.00
47	0.09	0.98	1.92	1.95
48	0.09	0.98	1.88	2.00
49	0.09	0.99	1.89	1.95
50	0.10	1.01	1.86	2.00
<b>SD</b>	<b>0.00495</b>	<b>0.01386</b>	<b>0.0204</b>	<b>0.02078</b>
<b>RSD</b>	<b>5.26459</b>	<b>1.40361</b>	<b>1.0773</b>	<b>1.05273</b>
<b>Presisi</b>	<b>94.7354</b>	<b>98.5964</b>	<b>98.9227</b>	<b>98.9473</b>
<b>Presisi ACS712</b>	<b>97.80</b>			

Dari 50 kali pengambilan data, dapat dilihat bahwa nilai standar deviasi sensor arus ACS712 pertama, ACS712 kedua, ACS712 ketiga dan ACS712 keempat memiliki nilai berturut-turut sebesar 0.00495, 0.01386, 0.0204 dan 0.02078. Adapun nilai *relative standard deviation* (RSD) masing-masing sensor memiliki nilai sebesar 5.26459 %, 1.40361 %, 1.0773 % dan 1.05273 %. Sehingga didapatkan nilai presisi sensor arus ACS712 pertama, ACS712 kedua, ACS712 ketiga dan ACS712 keempat memiliki nilai berturut-turut sebesar 94.7354 %, 98.5964 %, 98.9227 % dan 98.9473 %. Sehingga dapat disimpulkan bahwa sensor arus ACS712 pada *end node* 1 memiliki nilai presisi sebesar 97.80 %.



Gambar 4.3 Pembacaan ACS712 50 Kali

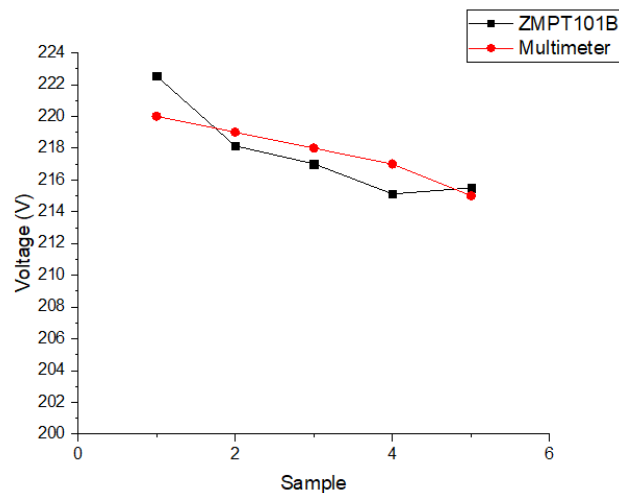
#### 4.1.3. Hasil Uji Akurasi Sensor Tegangan ZMPT101b

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor tegangan ZMPT101B. Data ini diambil sebanyak 5 sampel pada saat diberikan beban berbeda. Hasil pembacaan arus oleh sensor akan dibandingkan dengan hasil pembacaan arus oleh multimeter untuk mencari nilai persentase error.

Tabel 4.3 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter

Pengukuran ke-	ZMPT101B (V)	Multimeter (V)	Error (%)
1	222.55	220	1.16
2	218.14	219	0.40
3	217.01	218	0.45
4	215.12	217	0.86
5	215.5	215	0.23
<b>Rata-rata Error (%)</b>			<b>0.62</b>
<b>Akurasi (%)</b>			<b>99.38</b>

Dari data diatas dapat dilihat bahwa sensor tegangan ZMPT101B memiliki nilai rata-rata error sebesar 0.62, sehingga dapat disimpulkan bahwa sensor tegangan ZMPT101B memiliki nilai akurasi sebesar 99.38 %.



Gambar 4.4 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter

#### 4.1.4. Hasil Uji Akurasi Sensor Arus ACS712

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor arus ACS712. Data ini diambil sebanyak 5 sampel pada saat diberikan beban berbeda. Dimana pada beban 1 menggunakan panci elektrik level 1, beban kedua menggunakan panci elektrik level 2, beban ketiga menggunakan setrika, beban keempat menggunakan setrika + panci elektrik level 1 dan beban kelima menggunakan setrika + panci elektrik level 1. Hasil pembacaan arus oleh sensor akan dibandingkan dengan hasil pembacaan arus oleh multimeter untuk mencari nilai persentase error.

Tabel 4.4 Perbandingan Pembacaan Arus pada ACS712\_1 dan Multimeter

Pengukuran ke-	ACS1 (A)	Multimeter (A)	Error (%)
1	1.05	0.9	16.67
2	1.95	1.8	8.33
3	2.05	1.9	7.89
4	3.01	2.8	7.50
5	3.97	3.6	10.27
<b>Rata-rata error (%)</b>			10.13
<b>Akurasi (%)</b>			89.87

Dari data diatas dapat dilihat bahwa sensor arus ACS712\_1 memiliki nilai rata-rata error sebesar 10.13 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_1 memiliki nilai akurasi sebesar 89.87%.

Tabel 4.5 Perbandingan Pembacaan Arus pada ACS712\_2 dan Multimeter

Pengukuran ke-	ACS2 (A)	Multimeter (A)	Error (%)
1	1.01	1	1.00
2	1.92	1.8	6.67
3	2.01	1.9	5.78
4	3.01	2.8	7.50
5	3.81	3.6	5.83
<b>Rata-rata error (%)</b>			5.36
<b>Akurasi (%)</b>			94.64

Dari data diatas dapat dilihat bahwa sensor arus ACS712\_2 memiliki nilai rata-rata error sebesar 5.36 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_2 memiliki nilai akurasi sebesar 94.64 %.

Tabel 4.6 Perbandingan Pembacaan Arus pada ACS712\_3 dan Multimeter

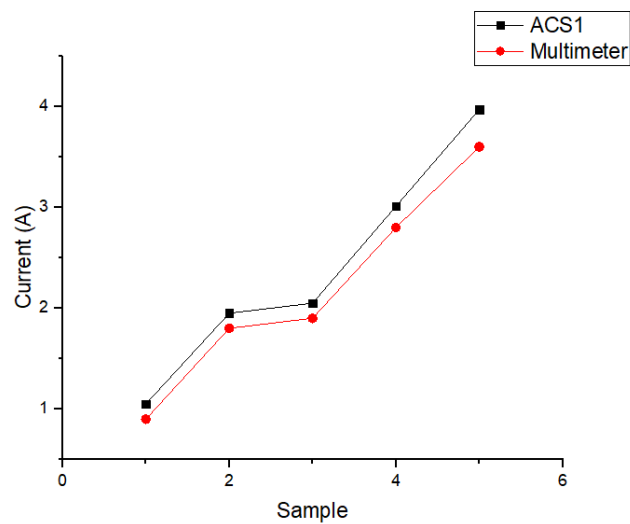
Pengukuran ke-	ACS3 (A)	Multimeter (A)	Error (%)
1	1.06	1	6.00
2	1.9	1.8	5.56
3	1.99	1.9	4.74
4	2.98	2.8	6.43
5	3.92	3.7	5.94
<b>Rata-rata error (%)</b>			5.73
<b>Akurasi (%)</b>			94.27

Dari data diatas dapat dilihat bahwa sensor arus ACS712\_3 memiliki nilai rata-rata error sebesar 5.73 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_3 memiliki nilai akurasi sebesar 94.27 %.

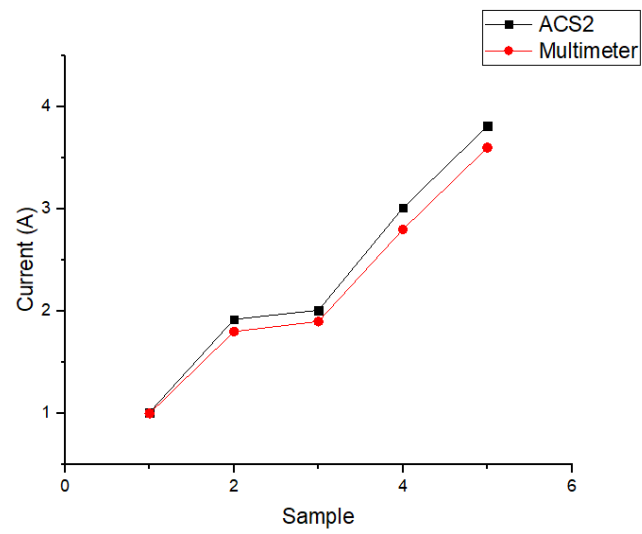
Tabel 4.7 Perbandingan Pembacaan Arus pada ACS712\_4 dan Multimeter

Pengukuran ke-	ACS4 (A)	Multimeter (A)	Error (%)
1	1.03	1	3.00
2	1.94	1.8	7.78
3	2.06	1.9	8.42
4	2.96	2.8	5.71
5	3.81	3.7	2.97
Rata-rata error (%)			5.58
Akurasi (%)			94.42

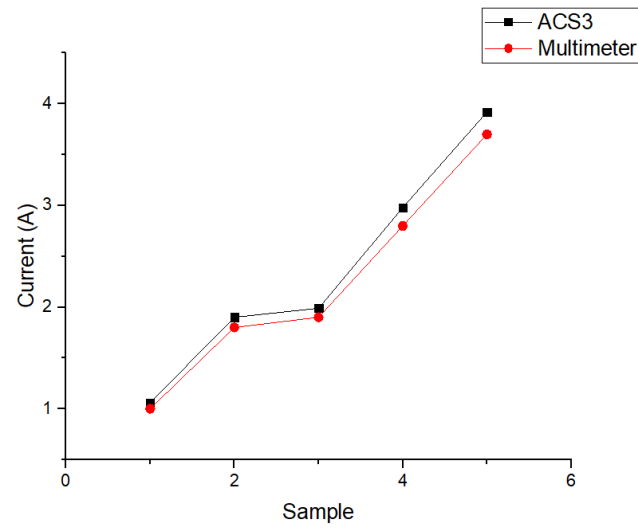
Dari data diatas dapat dilihat bahwa sensor arus ACS712\_4 memiliki nilai rata-rata error sebesar 5.58 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_4 memiliki nilai akurasi sebesar 94.42 %.



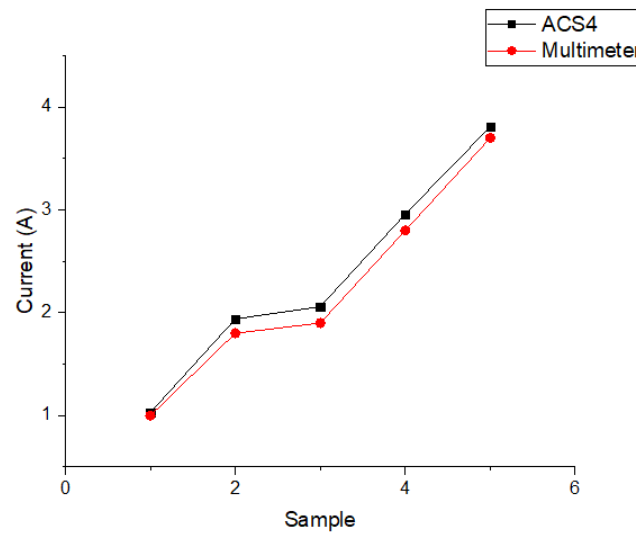
Gambar 4.5 Perbandingan Pembacaan Arus pada ACS712\_1 dan Multimeter



Gambar 4.6 Perbandingan Pembacaan Arus pada ACS712\_2 dan Multimeter



Gambar 4.7 Perbandingan Pembacaan Arus pada ACS712\_3 dan Multimeter



Gambar 4.8 Perbandingan Pembacaan Arus pada ACS712\_4 dan Multimeter

#### 4.1.5. Waktu Tunda Relay

Pengujian selanjutnya adalah pengujian *delay* atau waktu tunda dari relay yang digunakan. Pengujian dilakukan dengan cara melakukan kontrol pada seluruh relay dan menghitung berapa lama waktu yang dibutuhkan agar relay bekerja sesuai perintah yang telah diberikan. Dalam pengujian ini terdapat dua kali pengujian yaitu pada saat kondisi relay dari ON ke OFF dan pada saat kondisi relay dari ON ke OFF.

Tabel 4.8 Waktu Tunda Relay Kondisi ON-OFF

No	Relay	Delay (s)
1	Relay 1	2.40
2	Relay 2	2.11
3	Relay 3	2.75
4	Relay 4	2.30
Rata-rata		2.4

Dari data diatas dapat dilihat bahwa pada saat kondisi ON-OFF relay memiliki rata-rata waktu *delay* sebesar 2.4 detik.



Tabel 4.9 Waktu Tunda Relay Kondisi OFF-ON

No	Relay	Delay (s)
1	Relay 1	2.68
2	Relay 2	1.60
3	Relay 3	1.33
4	Relay 4	2.45
<b>Rata-rata</b>		<b>2</b>

Dari data diatas dapat dilihat bahwa pada saat kondisi ON-OFF relay memiliki rata-rata waktu *delay* sebesar 2 detik.

#### 4.2. Hasil Perancangan *End Node 2*

Perancangan hardware *end node 1* memiliki spesifikasi sebuah arduino uno sebagai mikrokontroler, LoRa shield dragino untuk komunikasi LoRa, 1 buah sensor tegangan ZMPT101B, 4 buah sensor arus ACS712 30 Ampere, relay module 4 channel yang terhubung ke 4 buah *output* berupa outlet, *master control end node*, *mode select manual - IoT* serta dilengkapi dengan proteksi berupa MCB 6 Ampere dan fuse 10 Ampere. *End node 1* merepresentasikan pada lantai 2 Gedung III Fakultas Teknik UNS.



Gambar 4.9 *End Node 2* Tampak Depan – Samping Kanan



Gambar 4.10 *End Node 2* Tampak Belakang – Samping Kiri

#### 4.2.1. Hasil Uji Presisi Sensor Tegangan ZMPT101b

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor regangan ZMPT101B. Data ini diambil sebanyak 50 sampel pada saat *end node* 2 tidak memiliki beban.

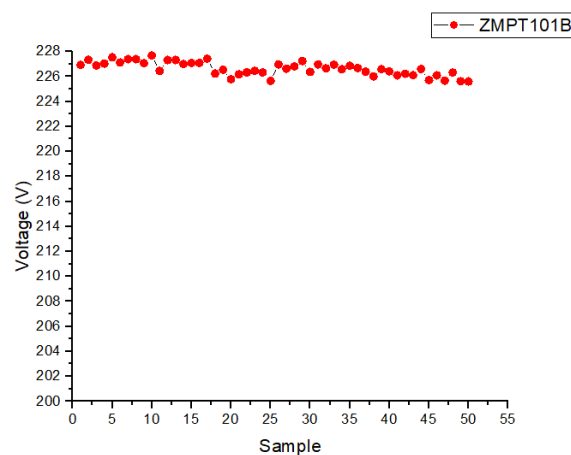
Tabel 4.10 Hasil Pembacaan ZMPT101B 50 Kali

Pengukuran ke-	Tegangan Terbaca pada Sensor ZMPT101B (V)
1	226.93
2	227.33
3	226.88
4	227.03
5	227.53
6	227.12
7	227.39
8	227.38
9	227.07
10	227.68
11	226.45
12	227.31
13	227.32
14	227.00
15	227.08
16	227.09
17	227.42
18	226.23
19	226.54
20	225.78
21	226.17
22	226.33
23	226.45
24	226.32
25	225.65
26	226.97
27	226.62
28	226.8
29	227.24
30	226.37
31	226.97
32	226.65
33	226.95

Tabel 4.10 Hasil Pembacaan ZMPT101B 50 Kali

Pengukuran ke-	Tegangan Terbaca pada Sensor ZMPT101B (V)
34	226.57
35	226.86
36	226.68
37	226.38
38	226.01
39	226.58
40	226.41
41	226.10
42	226.22
43	226.11
44	226.60
45	225.70
46	226.09
47	225.66
48	226.32
49	225.62
50	225.60
<b>SD</b>	<b>0.55458</b>
<b>RSD</b>	<b>0.24 %</b>
<b>Presisi</b>	<b>99.76 %</b>

Dari 50 kali pengambilan data, dapat dilihat bahwa nilai standar deviasi sensor tegangan ZMPT101b memiliki nilai 0.55458. Adapun *nilai relative standard deviation* (RSD) memiliki nilai 0.24 %. Sehingga dapat disimpulkan bahwa sensor tegangan ZMPT101b pada node 2 memiliki nilai presisi sebesar 99.76 %.



Gambar 4.11 Pembacaan ZMPT101B 50 kali

#### 4.2.2. Hasil Uji Presisi Sensor Arus ACS712

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan 4 buah sensor Arus ACS712. Data ini diambil sebanyak 50 sampel pada beban yang berbeda. Sensor arus ACS712 pertama diberikan beban berupa kipas angin, sensor arus ACS712 kedua diberikan beban berupa panci elektrik pada level 1, sensor arus ACS712 ketiga diberikan beban berupa panci elektrik pada level 2 dan sensor arus ACS712 keempat diberikan beban berupa setrika.

Tabel 4.11 Hasil Pembacaan ACS712 50 Kali

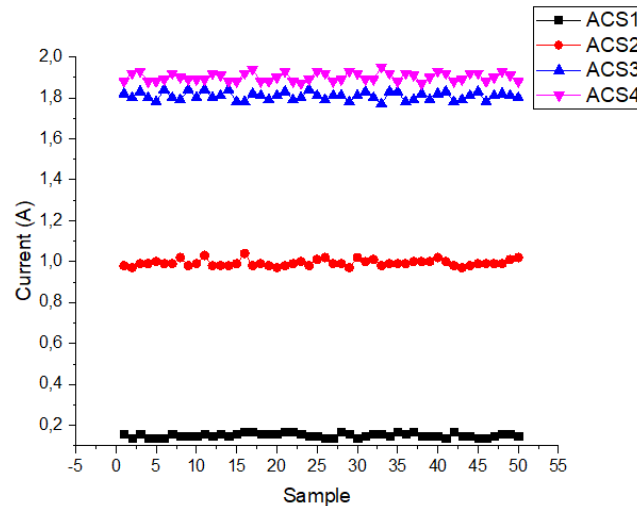
Pengukuran ke-	ACS1 (A)	ACS2 (A)	ACS3 (A)	ACS4 (A)
1	0.16	0.98	1.82	1.88
2	0.14	0.97	1.80	1.92
3	0.16	0.99	1.83	1.93
4	0.14	0.99	1.80	1.88
5	0.14	1.00	1.78	1.88
6	0.14	0.99	1.84	1.89
7	0.16	0.99	1.80	1.92
8	0.15	1.02	1.79	1.90
9	0.15	0.98	1.84	1.89
10	0.15	0.99	1.80	1.89
11	0.16	1.03	1.84	1.89
12	0.15	0.98	1.80	1.92
13	0.16	0.98	1.81	1.91
14	0.15	0.98	1.84	1.88
15	0.16	0.99	1.78	1.88
16	0.17	1.04	1.78	1.92
17	0.17	0.98	1.82	1.94
18	0.16	0.99	1.81	1.88
19	0.16	0.98	1.79	1.88
20	0.16	0.97	1.81	1.90
21	0.17	0.98	1.83	1.93
22	0.17	0.99	1.79	1.88
23	0.16	1.00	1.80	1.87
24	0.15	0.98	1.84	1.89
25	0.15	1.01	1.81	1.93
26	0.14	1.02	1.79	1.92
27	0.14	0.99	1.81	1.88
28	0.17	0.99	1.81	1.89
29	0.16	0.97	1.78	1.93

Tabel 4.11 Hasil Pembacaan ACS712 50 Kali

Pengukuran ke-	ACS1 (A)	ACS2 (A)	ACS3 (A)	ACS4 (A)
30	0.14	1.02	1.81	1.92
31	0.15	1.00	1.83	1.89
32	0.16	1.01	1.80	1.89
33	0.16	0.98	1.77	1.95
34	0.15	0.99	1.83	1.92
35	0.17	0.99	1.83	1.88
36	0.16	0.99	1.78	1.92
37	0.17	1.00	1.79	1.91
38	0.15	1.00	1.82	1.87
39	0.15	1.00	1.79	1.90
40	0.15	1.02	1.82	1.93
41	0.14	1.00	1.83	1.92
42	0.17	0.98	1.78	1.88
43	0.15	0.97	1.79	1.89
44	0.15	0.98	1.81	1.92
45	0.14	0.99	1.83	1.92
46	0.14	0.99	1.78	1.88
47	0.15	0.99	1.81	1.90
48	0.16	0.99	1.82	1.93
49	0.16	1.01	1.81	1.91
50	0.15	1.02	1.80	1.88
<b>SD</b>	<b>0.00993</b>	<b>0.01611</b>	<b>0.01967</b>	<b>0.02112</b>
<b>RSD (%)</b>	<b>6.43</b>	<b>1.62</b>	<b>1.08</b>	<b>1.11</b>
<b>Presisi (%)</b>	<b>93.56</b>	<b>98.38</b>	<b>98.92</b>	<b>98.89</b>
<b>Presisi ACS712 (%)</b>	<b>97.43</b>			

Dari 50 kali pengambilan data, dapat dilihat bahwa nilai standar deviasi sensor arus ACS712 pertama, ACS712 kedua, ACS712 ketiga dan ACS712 keempat memiliki nilai berturut-turut sebesar 0.00993, 0.01611, 0.01967 dan 0.02112. Adapun nilai *relative standard deviation* (RSD) masing-masing sensor memiliki nilai sebesar 6.43159 %, 1.62202 %, 1.62202 % dan 1.62202 %. Sehingga didapatkan nilai presisi sensor arus ACS712 pertama, ACS712 kedua, ACS712 ketiga dan ACS712 keempat memiliki nilai berturut-turut sebesar 93.5684 %,

98.378 %, 98.9116 % dan 98.8897 %. Sehingga dapat disimpulkan bahwa sensor arus ACS712 pada node 2 memiliki nilai presisi sebesar 97.43 %.



Gambar 4.12 Pembacaan ACS712 50 kali

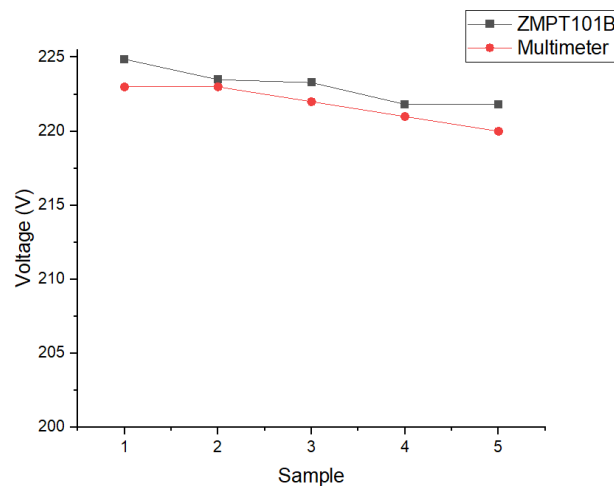
#### 4.2.3. Hasil Uji Akurasi Sensor Tegangan ZMPT101b

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor tegangan ZMPT101B. Data ini diambil sebanyak 5 sampel pada saat diberikan beban berbeda. Hasil pembacaan arus oleh sensor akan dibandingkan dengan hasil pembacaan arus oleh multimeter untuk mencari nilai persentase error.

Tabel 4.12 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter

Pengukuran ke-	ZMPT101B (V)	Multimeter (V)	Error (%)
1	224.87	223	0.83
2	223.49	223	0.22
3	223.29	222	0.58
4	221.82	221	0.37
5	221.81	220	0.82
<b>Rata-rata error (%)</b>			0.56
<b>Akurasi (%)</b>			99.44

Dari data diatas dapat dilihat bahwa sensor tegangan ZMPT101B memiliki nilai rata-rata error sebesar 0.56, sehingga dapat disimpulkan bahwa sensor tegangan ZMPT101B memiliki nilai akurasi sebesar 99.44 %.



Gambar 4.13 Perbandingan Pembacaan Tegangan pada ZMPT101B dan Multimeter

#### 4.2.4. Hasil Uji Akurasi Sensor Arus ACS712

Pada tabel di bawah ini merupakan hasil pengambilan data dari pembacaan sensor arus ACS712. Data ini diambil sebanyak 5 sampel pada saat diberikan beban berbeda. Dimana pada beban 1 menggunakan panci elektrik level 1, beban kedua menggunakan panci elektrik level 2, beban ketiga menggunakan setrika, beban keempat menggunakan setrika + panci elektrik level 1 dan beban kelima menggunakan setrika + panci elektrik level 1. Hasil pembacaan arus oleh sensor akan dibandingkan dengan hasil pembacaan arus oleh multimeter untuk mencari nilai persentase error.

Tabel 4.13 Perbandingan Pembacaan Arus pada ACS712\_1 dan Multimeter

Pengukuran ke-	ACS1 (A)	Multimeter (A)	Error (%)
1	1.01	1.0	1.00
2	1.81	1.8	0.56
3	1.91	1.9	0.52
4	2.78	2.8	0.71
5	3.59	3.7	2.97
<b>Rata-rata error (%)</b>			<b>1.15</b>
<b>Akurasi (%)</b>			<b>98.85</b>



Dari data diatas dapat dilihat bahwa sensor arus ACS712\_1 memiliki nilai rata-rata error sebesar 1.15 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_1 memiliki nilai akurasi sebesar 98.85 %.

Tabel 4.14 Perbandingan Pembacaan Arus pada ACS712\_2 dan Multimeter

Pengukuran ke-	ACS2 (A)	Multimeter (A)	Error (%)
1	0.98	1	2.00
2	1.84	1.8	2.22
3	1.92	1.9	1.05
4	2.82	2.8	0.71
5	3.65	3.7	1.35
<b>Rata-rata error (%)</b>			<b>1.47</b>
<b>Akurasi (%)</b>			<b>98.53</b>

Dari data diatas dapat dilihat bahwa sensor arus ACS712\_2 memiliki nilai rata-rata error sebesar 1.46 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_2 memiliki nilai akurasi sebesar 98.54 %.

Tabel 4.15 Perbandingan Pembacaan Arus pada ACS712\_3 dan Multimeter

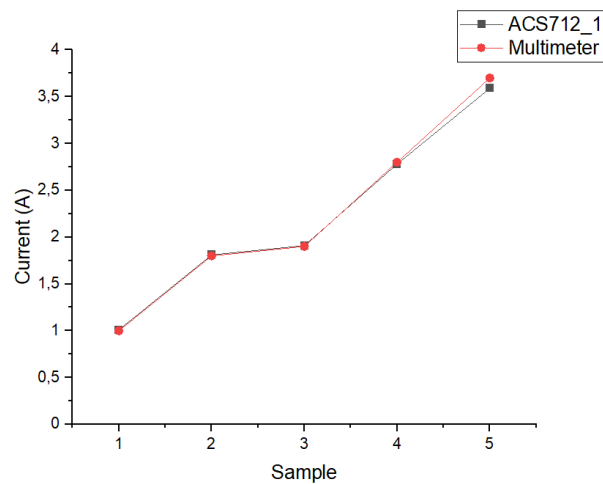
Pengukuran ke-	ACS3 (A)	Multimeter (A)	Error (%)
1	0.99	1.0	1.00
2	1.85	1.8	2.78
3	1.95	1.9	2.63
4	2.86	2.8	2.14
5	3.67	3.7	0.81
<b>Rata-rata Error (%)</b>			<b>1.87</b>
<b>Akurasi (%)</b>			<b>98.13</b>

Dari data diatas dapat dilihat bahwa sensor arus ACS712\_3 memiliki nilai rata-rata error sebesar 1.8726%, sehingga dapat disimpulkan bahwa sensor arus ACS712\_3 memiliki nilai akurasi sebesar 98.1274%.

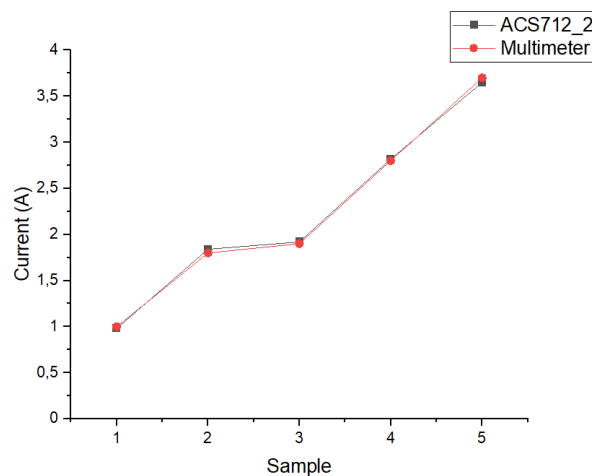
Tabel 4.16 Perbandingan Pembacaan Arus pada ACS712\_4 dan Multimeter

Pengukuran ke-	ACS4 (A)	Multimeter (A)	Error (%)
1	0.97	0.9	7.77
2	1.83	1.8	1.67
3	1.88	1.9	1.05
4	2.77	2.8	1.07
5	3.67	3.6	1.94
Rata-rata error (%)			2.71
Akurasi (%)			97.29

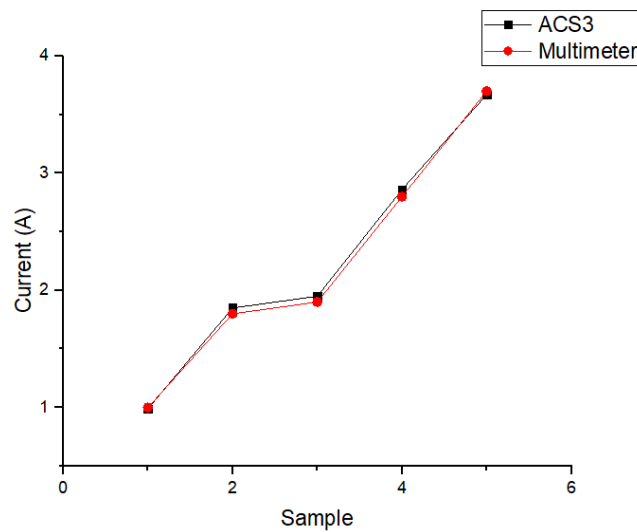
Dari data diatas dapat dilihat bahwa sensor arus ACS712\_4 memiliki nilai rata-rata error sebesar 2.71 %, sehingga dapat disimpulkan bahwa sensor arus ACS712\_4 memiliki nilai akurasi sebesar 97.29 %.



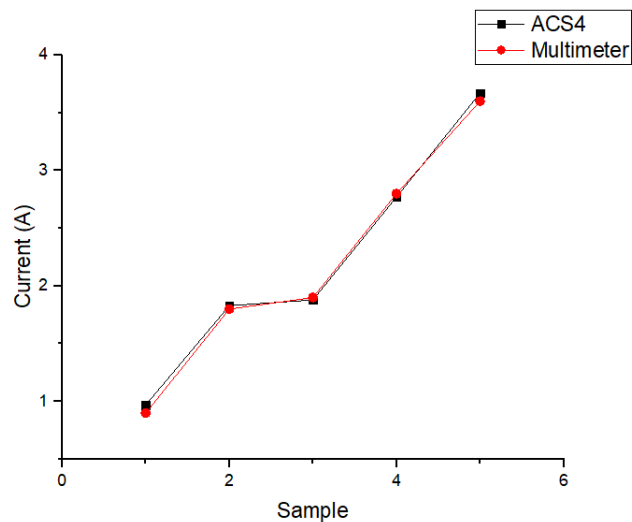
Gambar 4.14 Perbandingan Pembacaan Arus pada ACS712\_1 dan Multimeter



Gambar 4.15 Perbandingan Pembacaan Arus pada ACS712\_2 dan Multimeter



Gambar 4.16 Perbandingan Pembacaan Arus pada ACS712\_3 dan Multimeter



Gambar 4.17 Perbandingan Pembacaan Arus pada ACS712\_4 dan Multimeter

#### 4.2.5. Waktu Tunda Relay

Pengujian selanjutnya adalah pengujian *delay* atau waktu tunda dari relay yang digunakan. Pengujian dilakukan dengan cara melakukan kontrol pada seluruh relay dan menghitung berapa lama waktu yang dibutuhkan agar relay bekerja sesuai perintah yang telah diberikan. Dalam pengujian ini terdapat dua kali pengujian yaitu pada saat kondisi relay dari ON ke OFF dan pada saat kondisi relay dari ON ke OFF.

Tabel 4.17 Waktu Tunda Relay Kondisi ON-OFF

No	Relay	Delay (s)
1	Relay 1	3.24
2	Relay 2	2.83
3	Relay 3	1.56
4	Relay 4	1.72
<b>Rata-rata</b>		<b>2.3</b>

Dari data diatas dapat dilihat bahwa pada saat kondisi ON-OFF relay memiliki rata-rata waktu *delay* sebesar 2.3 detik.

Tabel 4.18 Waktu Tunda Relay Kondisi ON-OFF

No	Relay	Delay (s)
1	Relay 1	3.87
2	Relay 2	3.26
3	Relay 3	1.91
4	Relay 4	2.59
<b>Rata-rata</b>		<b>2.9</b>

Dari data diatas dapat dilihat bahwa pada saat kondisi OFF-ON relay memiliki rata-rata waktu *delay* sebesar 2.9 detik.

Pada pengujian waktu tunda relay ini diketahui bahwa pada sisi kontrol sistem terdapat kekurangan yaitu apabila mengirimkan *string* pada saat *hardware* dalam kondisi mati, maka *string* tersebut akan tetap disimpan di dalam Talkback Thingspeak. Maka, ketika *hardware* dihidupkan akan menyebabkan *outlet* pada *hardware* mengalami kondisi ON-OFF yang berulang sebanyak *string* masukan yang diberikan ketika *hardware* dalam kondisi mati.

### 4.3. Hasil Pengujian Jangkauan

Pada pengujian jangkauan ini dilakukan dengan tujuan untuk mengetahui seberapa jauh LoRa dapat bekerja. Data diambil dari jarak kelipatan 50m hingga 300m. Pada setiap jarak tersebut diambil 10 data RSSI untuk menentukan apakah kualitas sinyal yang diterima oleh *receiver (end node)* dari *gateway (transmitter)* termasuk dalam kategori baik atau buruk.

Tabel 4.19 Pengujian Jangkauan

RSSI ke-	Jarak (m)						
	1	50	100	150	200	250	300
RSSI-1 (dBm)	-62	-103	-105	-106	-109	-109	DC
RSSI-2 (dBm)	-58	-93	-105	-107	-110	-108	DC
RSSI-3 (dBm)	-47	-94	-110	-108	-111	-111	DC
RSSI-4 (dBm)	-49	-93	-105	-107	-112	-110	DC
RSSI-5 (dBm)	-58	-94	-106	-109	-112	-109	DC
RSSI-6 (dBm)	-46	-94	-111	-106	-111	-108	DC
RSSI-7 (dBm)	-58	-94	-107	-111	-111	-109	DC
RSSI-8 (dBm)	-48	-95	-109	-110	-107	-109	DC
RSSI-9 (dBm)	-51	-93	-105	-105	-106	-110	DC
RSSI-10 (dBm)	-54	-94	-104	-101	-112	-110	DC
Rata-rata (dBm)	-53.1	-94.7	-106.7	-107	-110.1	-109.3	DC

#### 4.3.1. Jarak 1 Meter

Pada pengujian jangkauan dengan jarak 1 meter. Nilai RSSI yang didapatkan bernilai -53.1 dBm sehingga termasuk dalam kategori sangat baik. Karena antara *receiver* dan *transmitter* dalam jarak yang berdekatan.

#### 4.3.2. Jarak 50 Meter

Pada pengujian jangkauan dengan jarak 50 meter. Nilai RSSI yang didapatkan bernilai -94.7 dBm sehingga termasuk dalam kategori baik.

#### 4.3.3. Jarak 100 Meter

Pada pengujian jangkauan dengan jarak 100 meter. Nilai RSSI yang didapatkan bernilai -106.7 dBm sehingga termasuk dalam kategori buruk. Namun masih bisa menerima sinyal dengan baik meskipun terdapat beberapa paket yang tidak diterima.

#### 4.3.4. Jarak 150 Meter

Pada pengujian jangkauan dengan jarak 150 meter. Nilai RSSI yang didapatkan bernilai -107 dBm sehingga termasuk dalam kategori buruk. Namun masih bisa menerima sinyal dengan baik meskipun terdapat beberapa paket yang tidak diterima.

#### 4.3.5. Jarak 200 Meter

Pada pengujian jangkauan dengan jarak 200 meter. Nilai RSSI yang didapatkan bernilai -110.1 dBm sehingga termasuk dalam kategori buruk. Namun masih bisa menerima sinyal dengan baik meskipun terdapat beberapa paket yang tidak diterima.

#### 4.3.6. Jarak 250 Meter

Pada pengujian jangkauan dengan jarak 250 meter. Nilai RSSI yang didapatkan bernilai -109.3 dBm sehingga termasuk dalam kategori buruk. Namun masih bisa menerima sinyal dengan baik meskipun terdapat beberapa paket yang tidak diterima.

#### 4.3.7. Jarak 300 Meter

Pada pengujian jangkauan dengan jarak 300 meter. *Receiver* sudah tidak mampu lagi terkoneksi dengan *transmitter*. Pada pengujian jarak 300 meter ini, tidak ada paket yang diterima oleh *receiver*. Sehingga dapat dikatakan pada jarak 300 meter, *receiver* dan *transmitter* dalam kondisi *disconnected*.

Dari hasil pengujian jarak yang telah dilakukan didapatkan jarak maksimal untuk LoRa bekerja adalah pada jarak 250 meter, hal ini disebabkan beberapa faktor antara lain terdapat penghalang berupa gedung serta penempatan *gateway* yang kurang tepat.

#### 4.4. Hasil Pengujian Keandalan (Reliability)

Pada pengujian *reliability* ini dilakukan dengan tujuan untuk mengetahui seberapa handal sistem yang dibuat. Data diambil selama 24 jam *nonstop* untuk dicari berapa nilai *packet loss* yang terjadi pada sistem.

Pada pengujian ini diasumsikan pengiriman data dilakukan setiap 18 detik sekali. 18 detik ini merupakan rata-rata waktu untuk *update* pada grafik Thingspeak dimana untuk setiap data yang masuk pada Thingspeak berkisar antara 15-20 detik. Sehingga jika diasumsikan ada data masuk setiap 18 detik sekali maka paket atau data yang disumsikan masuk selama 24 jam adalah:

$$\text{Expected Packet Received} = \frac{3600 \times 24}{18} \quad (3.1)$$

$$\text{Expected Packet Received} = 4800 \text{ Packet}$$

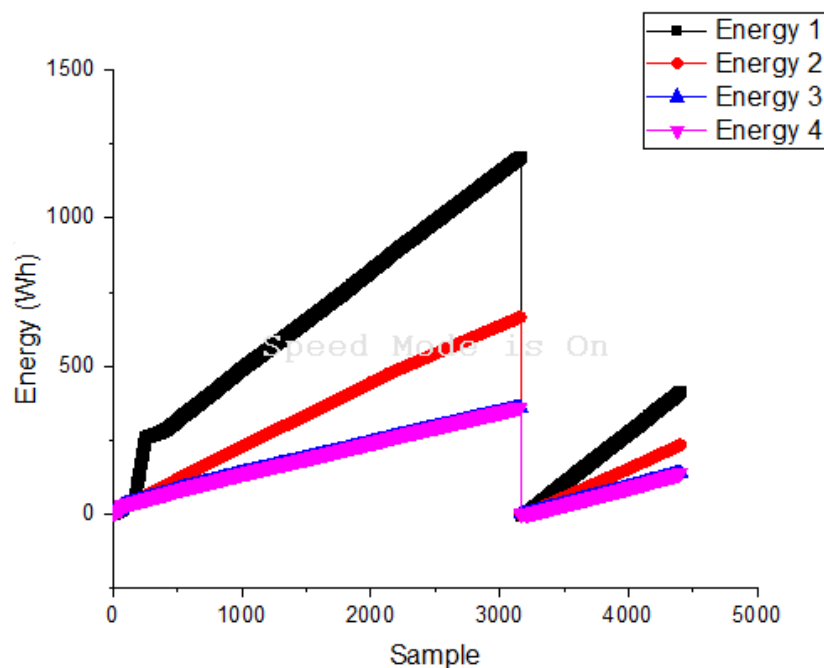
Sedangkan dari *data logger* yang telah didapatkan, *packet* yang diterima sebesar 4404 *packet*. Sehingga *packet loss* yang terjadi pada sistem yang telah dibuat adalah:

$$\text{Packet loss} = \frac{4800 - 4404}{4800} \cdot 100 \% \quad (3.2)$$

$$\text{Packet loss} = \frac{396}{4800} \cdot 100 \% \quad (3.3)$$

$$\text{Packet loss} = 8.25 \% \quad (3.4)$$

Berikut ini merupakan grafik dari data yang diambil:



Gambar 4.18 Energi Saat Pengujian Keandalan

Pada grafik dapat dilihat bahwa nilai energi 1 dan 2 mengalami kenaikan yang cukup signifikan karena terhubung dengan beban. Sedangkan energi 3 dan 4 bernilai relatif sama dikarenakan sama-sama tidak terhubung dengan beban.

Grafik energi 3 dan 4 mengalami kenaikan yang cukup tinggi dikarenakan terdapat error pada pembacaan arus, sehingga ketika diolah menjadi nilai energi dengan cara mengkalikan dengan nilai tegangan sebesar  $\pm 220$  V, maka masih memiliki nilai yang cukup tinggi dan nilai ini bertambah seiring berjalannya waktu.

Pada gambar 4.18 diatas juga bisa dilihat grafik sempat mengalami penurunan nilai dikarenakan pada saat pengujian terjadi pemadaman listrik di tempat penulis melakukan pengujian.

Pada pengujian ini penulis hanya menggunakan 2 buah beban yaitu magic com yang terhubung dengan *outlet* 1 serta kipas angin yang terhubung dengan *outlet* 2. Untuk *outlet* 3 dan 4 tidak terhubung dengan beban (tanpa beban). Berikut gambar grafik energi pada masing-masing *outlet* di *web server* Thingspeak:



Gambar 4.19 (a) Grafik energi outlet 1, (b) Grafik energi outlet 2, (c) Grafik energi outlet 3, (d) Grafik energi outlet 4

Dikarenakan penulis terdaftar di Thingspeak sebagai *free user*, maka untuk *update* nilai energi pada *web server* Thingspeak mengalami *delay* sebesar 15-20 detik.



#### 4.5. Hasil Perancangan Aplikasi Android

Perancangan aplikasi android bertujuan untuk memudahkan pengguna dalam melakukan *monitoring* dan kontrol pada beban yang terhubung dengan *end node*. Perancangan aplikasi android menggunakan MIT App Inventor dikarenakan perancangan yang mudah yang terdiri dari blok-blok logika. Pada sisi *monitoring* akan ditampilkan grafik energi dalam satuan watt hour (wh) yang diambil dari Thingspeak. Pada sisi kontrol akan ditampilkan berupa tombol-tombol kontrol untuk mematikan atau menghidupkan beban yang terhubung ke *end node*. Pada perancangan *software* ini, kata “Lantai 1” merepresentasikan *end node 1* dan kata “Lantai 2” merepresentasikan *end node 2*.

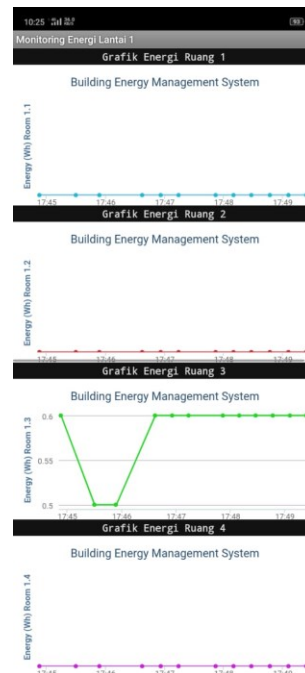
##### 4.5.1. Screen Menu Home



Gambar 4.20 Aplikasi Android Menu Home

Pada menu *home* terdapat 4 buah tombol yaitu MONITORING LANTAI 1, CONTROL LANTAI 1, MONITORING LANTAI 2 dan CONTROL LANTAI 2. Ketika tombol MONITORING LANTAI 1 ditekan maka akan menuju ke *screen* Monitoring Lantai 1, ketika tombol MONITORING LANTAI 2 ditekan maka akan menuju ke *screen* Monitoring Lantai 2, ketika tombol CONTROL LANTAI 1 ditekan maka akan menuju ke *screen* Control Lantai 1 dan ketika tombol CONTROL LANTAI 2 ditekan maka akan menuju ke *screen* Control Lantai 2.

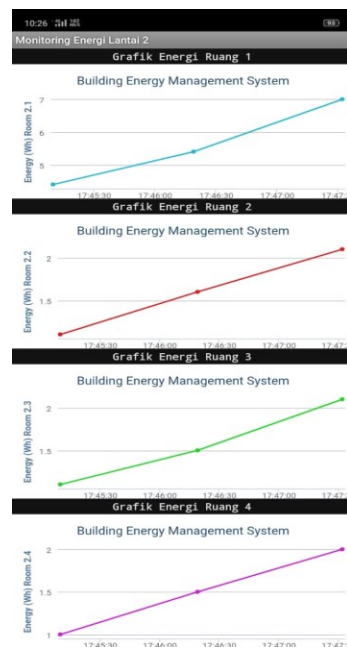
### 4.5.2. Screen Monitoring Lantai 1



Gambar 4.21 Aplikasi Android Menu Monitoring Energi Lantai 1

*Screen* Monitoring Lantai 1 akan menampilkan 4 buah grafik Thingspeak dari node 1.

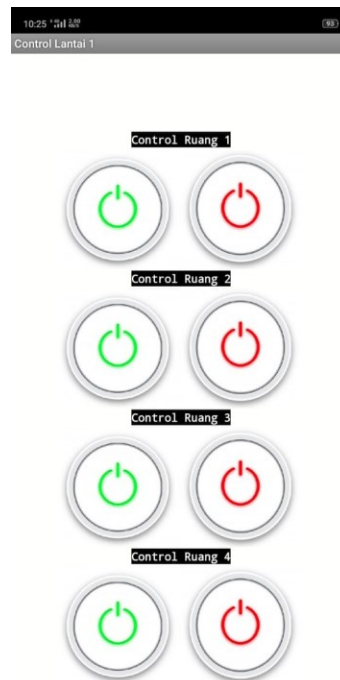
### 4.5.3. Screen Monitoring Lantai 2



Gambar 4. 22 Aplikasi Android Menu Monitoring Energi Lantai 2

*Screen* Monitoring Lantai 2 akan menampilkan 4 buah grafik Thingspeak dari node 2.

#### 4.5.4.Screen Control Lantai 1

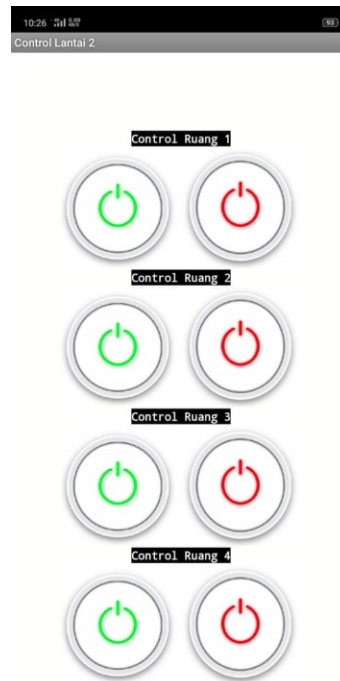


Gambar 4. 23 Aplikasi Android Menu Control Lantai 1

Pada *screen* Control Lantai 1 terdapat 4 pasang tombol ON-OFF. Pada Control Ruang 1 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY1\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 1 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY1\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 1 ke kondisi “LOW”. Pada Control Ruang 2 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY2\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 2 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY2\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 2 ke kondisi “LOW”. Pada Control Ruang 3 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY3\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 3 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY3\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 3 ke kondisi “LOW”.

Pada Control Ruang 4 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY4\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 4 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY4\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 1 untuk menset Relay 4 ke kondisi “LOW”.

#### 4.5.5.Screen Control Lantai 2



Gambar 4.24 Aplikasi Android Menu Control Lantai 2

Pada *screen* Control Lantai 2 terdapat 4 pasang tombol ON-OFF. Pada Control Ruang 1 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY5\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 1 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY5\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 1 ke kondisi “LOW”. Pada Control Ruang 2 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY6\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 2 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY6\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 2 ke kondisi “LOW”.

Pada Control Ruang 3 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY7\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 3 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY7\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 3 ke kondisi “LOW”. Pada Control Ruang 4 ketika tombol on ditekan maka akan mengirimkan *string* “RELAY8\_ON” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 4 ke kondisi “HIGH” dan ketika tombol off ditekan maka akan mengirimkan *string* “RELAY8\_OFF” ke Talkback Thingspeak yang kemudian akan diteruskan ke *end node* 2 untuk menset Relay 4 ke kondisi “LOW”.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Dalam penelitian ini dapat ditarik kesimpulan berupa:

1. Rancangan *hardware* yang telah dibuat berfungsi dengan baik, dimana *hardware* mampu untuk mengirimkan nilai energi ke *server* serta mampu menerima perintah untuk kontrol relay.
2. Aplikasi android telah mampu menampilkan data energi yang dikirimkan oleh *hardware* dan mampu untuk mengirimkan perintah untuk kontrol relay.
3. Pada pengujian performa didapatkan hasil sebagai berikut:
  - a. Pada pengujian presisi sensor, pada *end node* 1 didapatkan nilai presisi sensor tegangan ZMPT101B sebesar 98.90 % dan nilai presisi sensor arus ACS712 5A sebesar 97.80 %. Sedangkan pada *end node* 2 didapatkan nilai presisi sensor tegangan ZMPT101B sebesar 99.70 % dan nilai presisi sensor arus ACS712 30A sebesar 97.40 %.
  - b. Pada pengujian akurasi sensor, pada *end node* 1 didapatkan nilai akurasi sensor tegangan ZMPT101B sebesar 99.30 % dan nilai akurasi sensor arus ACS712 5A sebesar 93.20 %. Sedangkan pada *end node* 2 didapatkan nilai akurasi sensor tegangan ZMPT101B sebesar 99.40 % dan nilai akurasi sensor arus ACS712 30A sebesar 98.50 %.
  - c. Pada pengujian waktu tunda relay, pada *end node* 1 memiliki waktu tunda sebesar 2.4 detik pada saat kondisi OFF-ON dan waktu tunda sebesar 2 detik pada saat kondisi ON-OFF. Sedangkan, pada *end node* 2 memiliki waktu tunda sebesar 2.3 detik pada saat kondisi OFF-ON dan waktu tunda sebesar 2.9 detik pada saat kondisi ON-OFF.
  - d. Pada pengujian jangkauan, jarak maksimal untuk LoRa mampu bekerja adalah hanya pada jarak 250 meter.
  - e. Pada pengujian keandalan (*reliability*) didapatkan *packet loss* yang terjadi pada sistem adalah sebesar 8.25%.
4. Dari hasil pengujian tersebut maka dapat disimpulkan bahwa sistem memiliki performa dan dapat bekerja dengan baik.

## 5.2. Saran

Berdasarkan penelitian ini dapat diberikan beberapa saran untuk pengembangan kedepannya:

1. Salah satu kekurangan pada sistem yang dibuat adalah adanya *delay* sebesar 15-20 detik untuk *update* data pada Thingspeak sehingga tidak bisa *real time*. Untuk pengembangan penelitian selanjutnya mungkin bisa menggunakan *server* lain.
2. Terdapat antrian (*queue*) *string* perintah pada Talkback Thingspeak yang menyebabkan ketika diberikan *string* pada saat *hardware* mati, maka *string* tersebut akan dieksekusi pada saat *hardware* hidup. Pada pengembangan penelitian berikutnya bisa menggunakan metode pengontrolan lainnya.
3. Perlunya melakukan pengujian proteksi agar menjamin bahwa alat yang dibuat benar-benar aman.

## DAFTAR PUSTAKA

- [1] Kartika, S.A., 2018. Analisis Konsumsi Energi dan Program Konservasi Energi (Studi Kasus: Gedung Perkantoran dan Kompleks Perumahan Ti). *Sebatik*, 22(2), pp.41-50.
- [2] Nataliani, R., Soelami, F.N. and Budi, E.M., 2015. Pembuatan Perangkat Keras dan Analisis Sub-Metering Konsumsi Energi Listrik. *Jurnal Otomasi, Kontrol, dan Instrumentasi*, 6(1), p.25.
- [3] Badan Pengkajian dan Penerapan Teknologi (BPPT). 2019. *OUTLOOK ENERGI INDONESIA 2019 : Dampak Peningkatan Pemanfaatan Energi Baru Terbarukan Terhadap Perekonomian Nasional*. Tangerang Selatan: Pusat Pengkajian Industri Proses dan Energi (PPIPE).
- [4] Yang, T.Y., Yang, C.S. and Sung, T.W., 2015, November. An intelligent energy management scheme with monitoring and scheduling approach for IoT applications in smart home. In *2015 Third International Conference on Robot, Vision and Signal Processing (RVSP)* (pp. 216-219). IEEE.
- [5] Abo-Zahhad, M., Ahmed, S.M., Farrag, M., Ahmed, M.F. and Ali, A., 2015, December. Design and implementation of building energy monitoring and management system based on wireless sensor networks. In *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)* (pp. 230-233). IEEE.
- [6] Gupta, Y., Bajoria, S.P., Singh, R.K. and Swathika, O.G., 2017, October. IOT based energy management system with load sharing and source management features. In *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)* (pp. 169-177). IEEE.
- [7] Wasoontarajaroen, S., Pawasan, K. and Chamnanphrai, V., 2017, October. Development of an IoT device for monitoring electrical energy consumption. In *2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 1-4). IEEE.
- [8] Chaouch, H., Bayraktar, A.S. and Çeken, C., 2019, April. Energy Management in Smart Buildings by Using M2M Communication. In *2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)* (pp. 31-35). IEEE.
- [9] Jang, H., Choi, M.I., Lee, S., Lee, J. and Park, S., 2019, May. Building Energy Management System Based on Mixed Reality for Intuitive Interface. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)* (pp. 483-486). IEEE.
- [10] Efendi, Y., 2018. Internet Of Things (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer Fakultas Ilmu Komputer Universitas Al Asyariah Mandar*, 4(2), pp.21-27.
- [11] Abilovani, Z.B., Yahya, W. and Bakhtiar, F.A., 2018. Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. E-ISSN, pp.7521-7527.
- [12] Kartika, S.A., 2018. Analisis Konsumsi Energi dan Program Konservasi Energi (Studi Kasus: Gedung Perkantoran dan Kompleks Perumahan Ti). *Sebatik*, 22(2), pp.41-50.



- [13] Augustin, A., Yi, J., Clausen, T. and Townsley, W.M., 2016. A study of LoRa: Long range & low power networks for the internet of things. *Sensors*, 16(9), p.1466.
- [14] Zourmand, A., Hing, A.L.K., Hung, C.W. and AbdulRehman, M., 2019, June. Internet of Things (IoT) using LoRa technology. In 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) (pp. 324-330). IEEE.
- [15] Zhou, Q., Zheng, K., Hou, L., Xing, J. and Xu, R., 2019. Design and Implementation of Open LoRa for IoT. *IEEE Access*, 7, pp.100649-100657.
- [16] Single Channel LoRa Gateway Datasheet.
- [17] Long Range Wireless Transceiver for Arduino.
- [18] Ichwan, M., Husada, M.G. and Rasyid, M.I.A., 2013. Pembangunan prototipe sistem pengendalian peralatan listrik pada platform android. *Jurnal Informatika*, 4(1), pp.13-25.
- [19] Isfarizky, Z., Fardian, F. and Mufti, A., 2017. Rancang Bangun Sistem Kontrol Pemakaian Listrik Secara Multi Channel Berbasis Arduino (Studi Kasus Kantor LBH Banda Aceh). *Jurnal Karya Ilmiah Teknik Elektro*, 2(2).
- [20] Budiawan H, M.S., 2017. Sistem Pengendali Beban Arus Listrik Berbasis Arduino (Doctoral dissertation, Universitas Islam Negeri Alauddin Makassar).
- [21] Anugrah I, 2017. Pengukur Daya Listrik Menggunakan Sensor Arus Acs712-05a Dan Sensor Tegangan Zmpt101b (Universitas Negeri Yogyakarta).
- [22] Risma, 2019. Pengembangan Android Mobile Learning Menggunakan Mit App Inventor Sebagai Media Pembelajaran Matematika Pada Materi Dasar-Dasar Logika (Universitas Islam Negeri Raden Intan Lampung).
- [23] Alhasan, 2019. Implementasi *Wireless Sensor Network* Sebagai Pendeteksi Kebakaran Berbasis LoRa (Universitas Jember).
- [24] Putri Mandarani. 2014. Perancangan Dan Implementasi User Interface Berbasis Web Untuk Monitoring Suhu, Kelembaban Dan Asap Pada Ruangan Berbeda Dengan Memanfaatkan Jaringan Local Area Network. *Jurnal TEKNOIF*, 2(2), pp.37-42.
- [25] Wahri Sunanda, Irwan Dinata. 2014. Penerapan Perangkat Wireless Monitoring Energi Listrik Berbasis Arduino dan Internet. *Jurnal Amplifier*, 4(2), pp.21-23.
- [26] Temy Nusa, Sherwin R.U.A. Sompie, ST., MT., Dr.Eng Meita Rumbayan, ST., MT. 2015. Sistem Monitoring Konsumsi Energi Listrik Secara Real Time Berbasis Mikrokontroler. *E-Journal Teknik Elektro dan Komputer*, 4(5), pp.19-26.
- [27] Totok Budioko. 2016. Sistem Monitoring Suhu Jarak Jauh Berbasis Internet Of Things Menggunakan Protokol MQTT. *Seminar Riset Teknologi Informasi (SRITI)*, pp.353-358.
- [28] Sri Suryaningsih, Sahrul Hidayat, Faisal Abid. 2016. Rancang Bangun Alat Pemantau Penggunaan Energi Listrik Rumah Tangga Berbasis Internet. *Prosiding Seminar Nasional Fisika*, 5, pp.87-90.
- [29] Cahyono, B.E., Utami, I.D., Lestari, N.P. and Oktaviany, N.S., 2019. Karakterisasi Sensor LDR dan Aplikasinya pada Alat Ukur Tingkat Kekeruhan Air Berbasis Arduino UNO. *Jurnal Teori dan Aplikasi Fisika*, 7(2), pp.179-186.

- [30] Garnis, A., Suroso, S. and Soim, S., 2017. Pengkajian Kualitas Sinyal Dan Posisi Wifi Access Point Dengan Metode Rssi Di Gedung Kpa Politeknik Negeri Sriwijaya. Prosiding SNATIF, pp.429-434.

## LAMPIRAN

### Lampiran 1 Coding Program

```

1.  /*
2.   Program Building Energy Management System
3.   dikembangkan oleh Oki Setiawan (I0716027)
4.   Mahasiswa Teknik Elektro Universitas Sebelas Maret
5.   */
6.
7.  //inisiasi library
8.  // SPI & LoRa untuk komunikasi LoRa
9.  //Filters untuk pembacaan sensor arus dan tegangan
10.
11. #include <SPI.h>
12. #include <LoRa.h>
13. #include <Filters.h>
14.
15. //mendefinisikan setiap pin dari sensor tegangan dan arus
16. #define ZMPT_Pin A0
17. #define ACS_Pin1 A1
18. #define ACS_Pin2 A2
19. #define ACS_Pin3 A3
20. #define ACS_Pin4 A4
21.
22. //mendefinisikan setiap pin relay
23. const int Relay1 = 7;
24. const int Relay2 = 6;
25. const int Relay3 = 5;
26. const int Relay4 = 4;
27.
28. //mendefinisikan variabel nilai sensor arus dan tegangan
29. float ZMPT_Value;
30. float ACS1_Value;
31. float ACS2_Value;
32. float ACS3_Value;
33. float ACS4_Value;
34.
35. //F 50hz, windows length adalah berapa lama untuk rata-
    rata sinyal, untuk statistik
36. float testFrequency = 50;
37. float windowLength = 40.0/testFrequency;
38.
39. // slope error ketika tanpa supply
40. // intercept adalah hasil dari pembagian inputstats:acs/zmpt value
41. // slope dan intercept didapatkan dari perhitungan regresi linier
42. float intercept = -5.42453914296627; // V
43. float slope = 2.85755012892355; // V
44.
45. float intercept1 = -0.0482848690479361; // ACS1
46. float slope1 = 0.0297892830078982; // ACS1
47. float intercept2 = -0.0392823056473212; // ACS2
48. float slope2 = 0.0286588962955441; // ACS2
49. float intercept3 = -0.038036688607847; // ACS3
50. float slope3 = 0.0282317439991083; // ACS3
51. float intercept4 = -0.0401437701894681; // ACS4
52. float slope4 = 0.0283121605847432; // ACS4
53.
54. //define

```

```

55.
56. float arus1, arus2, arus3, arus4, tegangan, daya1, daya2, daya3, daya4,
    bill1, bill2, bill3, bill4;
57. //inisiasi awal nilai wh
58. float Wh1=0, Wh2=0, Wh3=0, Wh4=0;
59. // 8 panjang array, \0 untuk mengisi karakter kosong jika array memiliki
    lebar lebih dari yang di inisiasi
60. char bill1_1[8]={"\0"}, Wh1_1[8]={"\0"}, bill2_1[8]={"\0"}, Wh2_1[8]={"\0"},
    bill3_1[8]={"\0"}, Wh3_1[8]={"\0"}, bill4_1[8]={"\0"}, Wh4_1[8]={"\0"};
61.
62. //node id ke gateway
63. char *node_id = "<4656>";
64. //unsigned integer 8 byte
65. uint8_t datasend[100];
66.
67. unsigned long printPeriod = 1000; // tulis ke serial monitor setiap 1 detik
68. unsigned long previousMillis = 0;
69. unsigned long new_time,old_time=0;
70. unsigned long last_time =0;
71. unsigned long current_time =0;
72.
73.
74. String message;
75. String command;
76.
77. void setup() {
78.   pinMode(Relay1, OUTPUT);
79.   pinMode(Relay2, OUTPUT);
80.   pinMode(Relay3, OUTPUT);
81.   pinMode(Relay4, OUTPUT);
82.   Serial.println(F("Start ThingSpeak"));
83.   Serial.begin(9600);
84.   while (!Serial);
85.   if (!LoRa.begin(915000000)) //frekuensi 915hz
86.   {
87.     Serial.println("Starting LoRa failed!");
88.     while (1);
89.   }
90.   LoRa.setSpreadingFactor(7); //
91.   LoRa.setSignalBandwidth(125000); //
92.   LoRa.setCodingRate4(5); //
93.   LoRa.setSyncWord(0x34); //hexa 52
94.   Serial.println("LoRa init succeeded.");
95.   delay(100);
96.   pinMode(ACS_Pin1,INPUT);
97.   pinMode(ACS_Pin2,INPUT);
98.   pinMode(ACS_Pin3,INPUT);
99.   pinMode(ACS_Pin4,INPUT);
100.   LoRa.onReceive(onReceive);
101.   LoRa.receive();
102. }
103.
104. void sensorWrite()
105. {
106.   char data[100] = "\0";
107.   for(int i = 0; i < 100; i++)
108.   {
109.     data[i] = node_id[i];
110.   }
111.
112.   dtostrf(Wh1,0,1,Wh1_1);

```

```

113.         dtostrf(Wh2,0,1,Wh2_1);
114.         dtostrf(Wh3,0,1,Wh3_1);
115.         dtostrf(Wh4,0,1,Wh4_1);
116.
117.         strcat(data,"field1=");
118.         strcat(data,Wh1_1);
119.         strcat(data,"&field2=");
120.         strcat(data,Wh2_1);
121.         strcat(data,"&field3=");
122.         strcat(data,Wh3_1);
123.         strcat(data,"&field4=");
124.         strcat(data,Wh4_1);
125.         strcpy((char *)datasend,data);
126.     }
127.
128.     void SendData()
129.     {
130.         LoRa.beginPacket();
131.         LoRa.print((char *)datasend);
132.         LoRa.endPacket();
133.         Serial.println("Packet Sent");
134.         Serial.print("#####");
135.         Serial.print( "\n" );
136.
137.     }
138.     void loop() {
139.         RunningStatistics inputStats;
140.         RunningStatistics inputStats1;
141.         RunningStatistics inputStats2;
142.         RunningStatistics inputStats3;
143.         RunningStatistics inputStats4;
144.         inputStats.setWindowSecs(windowLength);
145.         inputStats1.setWindowSecs(windowLength);
146.         inputStats2.setWindowSecs(windowLength);
147.         inputStats3.setWindowSecs(windowLength);
148.         inputStats4.setWindowSecs(windowLength);
149.         while(true) {
150.             ZMPT_Value = analogRead(ZMPT_Pin); // read the analog in val
ue:
151.             ACS1_Value = analogRead(ACS_Pin1); // read the analog in val
ue:
152.             ACS2_Value = analogRead(ACS_Pin2); // read the analog in val
ue:
153.             ACS3_Value = analogRead(ACS_Pin3); // read the analog in val
ue:
154.             ACS4_Value = analogRead(ACS_Pin4); // read the analog in val
ue:
155.             inputStats.input(ZMPT_Value); // log to Stats function
156.             inputStats1.input(ACS1_Value); // log to Stats function
157.             inputStats2.input(ACS2_Value); // log to Stats function
158.             inputStats3.input(ACS3_Value); // log to Stats function
159.             inputStats4.input(ACS4_Value); // log to Stats function
160.
161.             if((unsigned long)(millis() - previousMillis) >= printPeriod)
{
162.                 previousMillis = millis(); // update time every second
163.
164.                 //arus
165.                 arus1 = intercept1 + slope1 * inputStats1.sigma();
166.                 arus2 = intercept2 + slope2 * inputStats2.sigma();
167.                 arus3 = intercept3 + slope3 * inputStats3.sigma();
168.                 arus4 = intercept4 + slope4 * inputStats4.sigma();
169.

```

```

170.         //tegangan
171.         tegangan = intercept + slope * inputStats.sigma();
172.
173.         //daya
174.         daya1 = arus1 * tegangan * 0.85; //0.85 faktor daya
175.         daya2 = arus2 * tegangan * 0.85;
176.         daya3 = arus3 * tegangan * 0.85;
177.         daya4 = arus4 * tegangan * 0.85;
178.
179.         //wh
180.         last_time = current_time;
181.         current_time = millis();
182.         Wh1 = Wh1+ daya1 *(( current_time -
last_time) /3600000.0);
183.         Wh2 = Wh2+ daya2 *(( current_time -
last_time) /3600000.0);
184.         Wh3 = Wh3+ daya3 *(( current_time -
last_time) /3600000.0);
185.         Wh4 = Wh4+ daya4 *(( current_time -
last_time) /3600000.0);
186.
187.         Serial.print("Tegangan: ");
188.         Serial.print(tegangan);
189.         Serial.print( "\n" );
190.         Serial.print("Arus 1: ");
191.         Serial.print( arus1);
192.         Serial.print( "\n" );
193.         Serial.print("Arus 2: ");
194.         Serial.print( arus2);
195.         Serial.print( "\n" );
196.         Serial.print("Arus 3: ");
197.         Serial.print( arus3);
198.         Serial.print( "\n" );
199.         Serial.print("Arus 4: ");
200.         Serial.print( arus4);
201.         Serial.print( "\n" );
202.         sensorWrite();
203.         SendData();
204.         LoRa.receive();
205.     }
206. }
207. }
208.
209. void onReceive(int packetSize) {
210.     message = "";
211.     for (int j = 0; j < packetSize; j++)
212.     {
213.         message += (char)LoRa.read();
214.     }
215.     command = String(message);
216.     //RELAY CONTROL
217.     if (command == "RELAY1_ON"){
218.         digitalWrite(Relay1, HIGH);
219.         delay(100);
220.     }
221.
222.     if (command == "RELAY1_OFF"){
223.         digitalWrite(Relay1, LOW);
224.         delay(100);
225.     }
226.
227.     if (command == "RELAY2_ON"){
228.         digitalWrite(Relay2, HIGH);

```

```

229.         delay(100);
230.     }
231.
232.     if (command == "RELAY2_OFF"){
233.         digitalWrite(Relay2, LOW);
234.         delay(100);
235.     }
236.     if (command == "RELAY3_ON"){
237.         digitalWrite(Relay3, HIGH);
238.         delay(100);
239.     }
240.     if (command == "RELAY3_OFF"){
241.         digitalWrite(Relay3, LOW);
242.         delay(100);
243.     }
244.     if (command == "RELAY4_ON"){
245.         digitalWrite(Relay4, HIGH);
246.         delay(100);
247.     }
248.     if (command == "RELAY4_OFF"){
249.         digitalWrite(Relay4, LOW);
250.         delay(100);
251.     }
252.     Serial.print("\n");
253.     Serial.print("Perintah: ");
254.     Serial.print(command);
255.     Serial.print("\n\r");
256. }

```

## Lampiran 2 Nilai regresi kalibrasi sensor *end node 1*

- **Sensor Tegangan ZMPT101B**

multi	sensor	beban
224	80.52	0 beban
222	79.94	panci elektrik lv.1
221	78.64	panci elektrik lv.2
0	1.91	OFF

Slope = 2.85755012892355

Intercept = -5.42453914296627

- **Sensor ACS712\_1**

multi	sensor	beban
0	1.6	tanpa beban
0.1	4.65	kipas angin
1	35.86	panci elektrik lv 1
1.9	65.61	panci elektrik lv 2
2	68.23	setrika

Slope = 0.0297892830078982  
 Intercept = -0.0482848690479361

- **Sensor ACS712\_2**

multi	sensor	beban
0	1.72	tanpa beban
0.1	4.67	kipas angin
1	35.73	panci elektrik lv 1
1.8	65.98	panci elektrik lv 2
2	69.73	setrika

Slope = 0.0286588962955441  
 Intercept = -0.0392823056473212

- **Sensor ACS712\_3**

multi	sensor	beban
0	1.66	tanpa beban
0.1	4.75	kipas angin
1	36.21	panci elektrik lv 1
1.8	66.82	panci elektrik lv 2
2	70.86	setrika

Slope = 0.0282317439991083  
 Intercept = -0.038036688607847

- **Sensor ACS712\_4**

multi	sensor	beban
0	1.56	tanpa beban
0.1	4.97	kipas angin
1	36.22	panci elektrik lv 1
1.8	66.61	panci elektrik lv 2
2	70.8	setrika

Slope = 0.0283121605847432  
 Intercept = -0.0401437701894681



### Lampiran 3 Nilai regresi kalibrasi sensor *end node 2*

- **Sensor Tegangan ZMPT101B**

multi	sensor	beban
224	139.2	tanpa beban
220	137.6	setrika
1.9	1.85	OFF
0.4	1.3	OFF

Slope = 1.60396745313011

Intercept = 0

- **Sensor ACS712\_1**

multi	sensor	beban
0	1.52	tanpa beban
0.1	2.37	kipas angin
0.9	14.27	panci elektrik lv 1
1.8	26.36	panci elektrik lv 2
1.9	27.73	setrika

Slope = 0.0675764119614219

Intercept = 0

- **Sensor ACS712\_2**

multi	sensor	beban
0	1.35	tanpa beban
0.1	2.33	kipas angin
1	15.05	panci elektrik lv 1
1.8	27.49	panci elektrik lv 2
1.9	29.02	setrika

Slope = 0.0654630144860184

Intercept = 0

- **Sensor ACS712\_3**

multi	sensor	beban
0	1.36	tanpa beban
0.1	2.38	kipas angin
1	14.81	panci elektrik lv 1

1.8	27.17	panci elektrik lv 2
1.9	28.53	setrika

Slope = 0.0664191558509267

Intercept = 0

- **Sensor ACS712\_4**

multi	sensor	beban
0	1.26	tanpa beban
0.1	2.45	kipas angin
1	15.05	panci elektrik lv 1
1.8	27.32	panci elektrik lv 2
1.9	29.01	setrika

Slope = 0.0656346781170339

Intercept = 0