

**SISTEM MONITORING DAYA LISTRIK *INTERNET OF THINGS* (IoT)  
MENGGUNAKAN ALGORITMA FUZZY LOGIC SUGENO DAN  
FIREBASE BERBASIS ANDROID**

**SKRIPSI**

**Diajukan Untuk Memenuhi Salah Satu Persyaratan Memperoleh Gelar  
Sarjana Komputer (S.Kom)**



**Oleh :**

**Ahmad Muzakir**

**11180910000014**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH**

**JAKARTA**

**2023 M / 1444 H**

## **PERNYATAAN ORISINALITAS**

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan benar hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila dikemudian hari terbukti karya ini bukan hasil saya sendiri atau merupakan hasil jiplakan karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, 19 Juni 2023

Penulis



Ahmad Muzakir

11180910000014

## PENYATAAN PERSETUJUAN PUBLIKASI SKRIPSI

Sebagai civitas akademika UIN Syarif Hidayatullah Jakarta. Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Muzakir  
NIM : 11180910000014  
Program Studi : Teknik Informatika  
Fakultas : Sains dan Teknologi  
Jenis Karya : Skripsi

Demi pembuatan ilmu pengetahuan saya menyetujui untuk memberikan kepada UIN Syarif Hidayatullah Jakarta Hak Bebas Royalti Non Eksklusif (*Non Exclusive Royalty Free Right*) atas karya ilmiah yang berjudul:

### SISTEM MONITORING DAYA LISTRIK INTERNET OF THINGS (IoT) MENGGUNAKAN ALGORITMA FUZZY LOGIC SUGENO DAN FIREBASE BERBASIS ANDROID

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Ekslusif ini UIN Syarif Hidayatullah Jakarta berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data, merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Jakarta, Juli 2023

Penulis

Ahmad Muzakir  
11180910000014

<b>Nama</b>	<b>: Ahmad Muzakir</b>
<b>Program Studi</b>	<b>: Teknik Informatika</b>
<b>Judul</b>	<b>: SISTEM MONITORING DAYA LISTRIK INTERNET OF THINGS (IoT) MENGGUNAKAN ALGORITMA FUZZY LOGIC SUGENO DAN FIREBASE BERBASIS ANDROID</b>

## ABSTRAK

Penelitian ini bertujuan mengembangkan sistem monitoring daya listrik berbasis IoT dengan algoritma *Fuzzy Logic* Sugeno dan *Firebase*. Sistem ini memantau penggunaan daya listrik secara real-time untuk meningkatkan efisiensi energi. Metode penelitian meliputi studi literatur, perancangan perangkat keras menggunakan ESP8266, integrasi dengan *Firebase* sebagai database real-time, pengembangan aplikasi Android menggunakan Flutter, dan implementasi algoritma *Fuzzy Logic* Sugeno. Sistem yang dikembangkan mampu mengumpulkan data penggunaan daya listrik melalui sensor pada ESP8266 dan mengirimkannya ke *Firebase* menggunakan API Key sebagai metode komunikasi. Aplikasi Android yang dibangun memungkinkan pengguna memantau penggunaan daya listrik secara real-time dan menganalisis data melalui antarmuka yang intuitif. Hasil pengujian menunjukkan sistem berfungsi baik, memberikan data yang akurat, dan responsif terhadap perubahan penggunaan daya listrik. Sistem ini dapat membantu pengguna memantau dan mengoptimalkan penggunaan daya listrik, dengan potensi mengurangi pemborosan energi dan meningkatkan efisiensi energi secara keseluruhan.

<b>Kata Kunci</b>	<b>: Monitoring System, IoT, <i>Firebase</i>, ESP8266, Android, Power consumption</b>
-------------------	---

**Name**

**: Ahmad Muzakir**

**Study Program**

**: Informatic Engineering**

**Title**

**: INTERNET OF THINGS (IoT) ELECTRIC POWER  
MONITORING SYSTEM USING FUZZY LOGIC  
SUGENO AND FIREBASE ALGORITHMS BASED ON  
ANDROID**

## **ABSTRACT**

This study aims to develop an IoT-based power monitoring system using the *Fuzzy Logic* Sugeno algorithm and *Firebase*. The system monitors real-time power consumption to enhance energy efficiency. The research methodology includes literature review, hardware design using ESP8266, integration with *Firebase* as a real-time database, Android application development using Flutter, and implementation of the *Fuzzy Logic* Sugeno algorithm. The developed system is capable of collecting power consumption data through sensors on the ESP8266 and sending it to *Firebase* using an API Key for communication. The Android application allows users to monitor real-time power usage and analyze the data through an intuitive interface. The testing results demonstrate that the system functions effectively, provides accurate data, and is responsive to changes in power consumption. This system can assist users in monitoring and optimizing power usage, potentially reducing energy wastage and improving overall energy efficiency.

**Kata Kunci**

**: Monitoring System, IoT, *Firebase*, ESP8266, Android,  
Power consumption**

## KATA PENGANTAR

*Bismillahirrahmanirrahim*

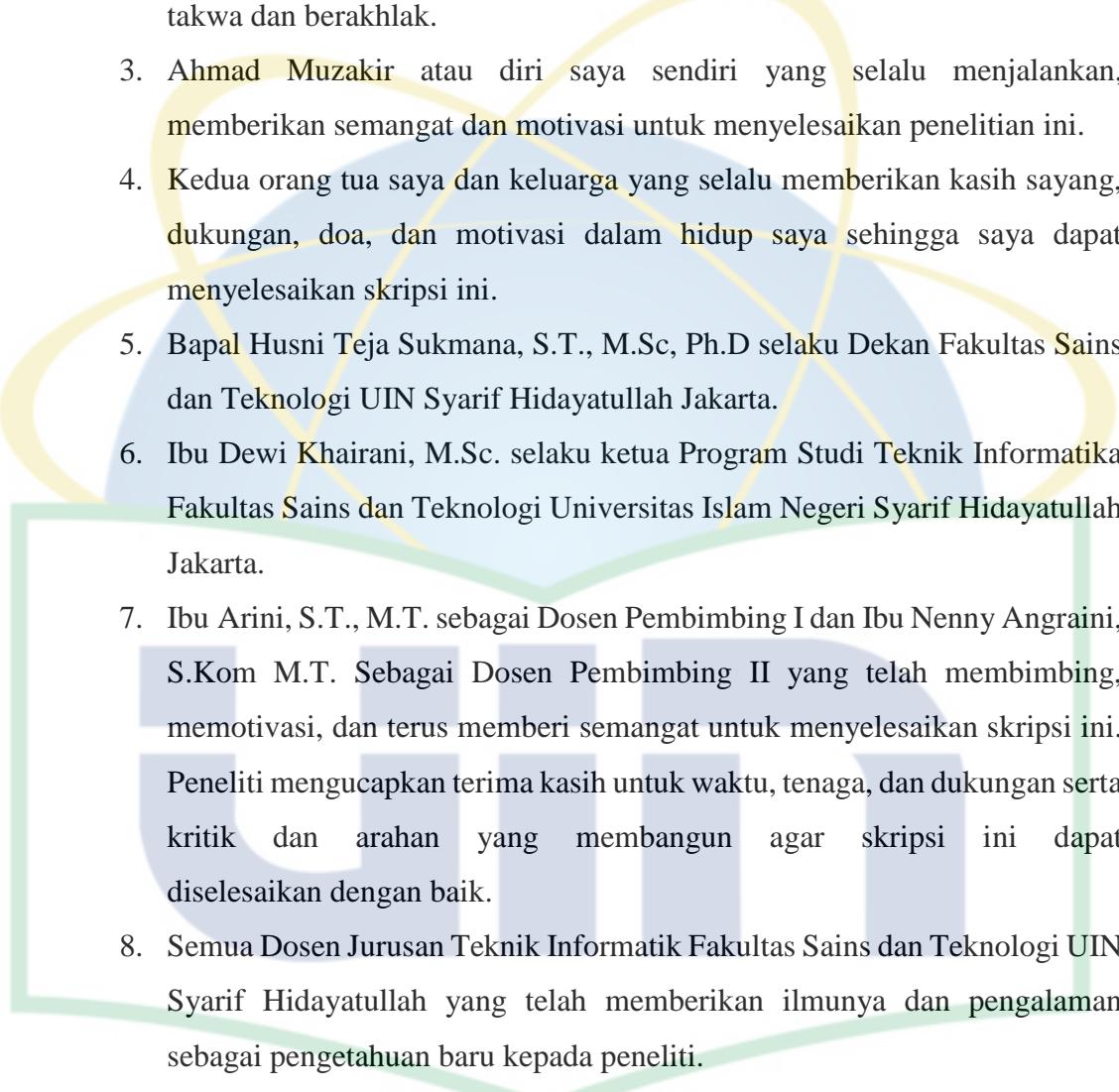
*Assalamu'alaikum Warahmatullah Wabarakatuh,*

Puji dan syukur kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga peneliti dapat menyelesaikan skripsi ini dengan baik. Shalawat serta salam tak lupa selalu tercurahkan kepada junjungan kita nabi besar, Nabi Muhammad SAW yang telah memberikan tuntunan dan petunjuk kepada umat manusia menuju kehidupan dan peradaban yang lebih baik, serta para keluarga, sahabat Nabi yang kita cintai.

Peneliti menyadari bahwa dalam pembuatan skripsi ini masih banyak kekurangan. Hal ini semata-mata karena kurangnya pengetahuan dan pengalaman yang dimiliki peneliti. Namun demikian peneliti berharap skripsi ini dapat memenuhi syarat dalam memperoleh gelar Sarjana (S1) dalam bidang Teknik Informatika dari Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Skripsi yang berjudul “**SISTEM MONITORING DAYA LISTRIK INTERNET OF THINGS (IoT) MENGGUNAKAN ALGORITMA FUZZY LOGIC SUGENO DAN FIREBASE BERBASIS ANDROID**”, akhirnya dapat dapat diselesaikan sesuai yang diharapkan. Selama penyusunan skripsi ini tentunya terdapat banyak kesulitan dan hambatan yang dialami oleh peneliti, baik dalam pengumpulan data, implementasi metode dan lain sebagainya, namun berkat bantuan dari berbagai pihak, kesulitan tersebut dapat diatasi. Hal ini merupakan suatu kebahagian bagi peneliti dapat mempersembahkan kepada kedua orang tua, keluarga besar, dan pihak-pihak yang telah ikut andil dalam penyelesaian skripsi ini.

Sebagai bentuk penghargaan izinkan peneliti untuk menuangkannya dalam bentuk ucapan terima kasih sebesar-besarnya kepada :

- 
1. Allah subhanahu wa ta'ala yang maha kuasa yang telah memberikan nikmat dan jalan agar penulis dapat menyelesaikan penulisan skripsi ini.
  2. Rasullullah Nabiyallahu Muhammad SAW yang telah menjadi sebaik-baiknya suri teladan dan panutan bagi peniliti untuk menjadi manusia yang takwa dan berakhhlak.
  3. Ahmad Muzakir atau diri saya sendiri yang selalu menjalankan, memberikan semangat dan motivasi untuk menyelesaikan penelitian ini.
  4. Kedua orang tua saya dan keluarga yang selalu memberikan kasih sayang, dukungan, doa, dan motivasi dalam hidup saya sehingga saya dapat menyelesaikan skripsi ini.
  5. Bapak Husni Teja Sukmana, S.T., M.Sc, Ph.D selaku Dekan Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
  6. Ibu Dewi Khairani, M.Sc. selaku ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.
  7. Ibu Arini, S.T., M.T. sebagai Dosen Pembimbing I dan Ibu Nenny Angraini, S.Kom M.T. Sebagai Dosen Pembimbing II yang telah membimbing, memotivasi, dan terus memberi semangat untuk menyelesaikan skripsi ini. Peneliti mengucapkan terima kasih untuk waktu, tenaga, dan dukungan serta kritik dan arahan yang membangun agar skripsi ini dapat diselesaikan dengan baik.
  8. Semua Dosen Jurusan Teknik Informatik Fakultas Sains dan Teknologi UIN Syarif Hidayatullah yang telah memberikan ilmunya dan pengalaman sebagai pengetahuan baru kepada peneliti.
  9. Terimakasih kepada Saudara saya, warga rumah dan jamaah masjid yang memberikan dukungan dan doa yang tiada hentinya kepada penulis selama proses penyelesaian skripsi.
  10. Rekan-rekan angkatan 2018 Teknik Informatika UIN Syarif Hidayatullah atas kebersamaannya selama ini.

11. Seluruh pihak yang tidak bisa penulis sebutkan satu per satu, terima kasih banyak penulis ucapkan kepada kalian semua atas doa baik dan dukungannya selama proses penulisan skripsi ini.

Atas segala bantuan dan dukungan dari semua pihak, peneliti berterima kasih dan akan selalu mendoakan kalian semoga apa yang telah diberikan dijadikan amal kebaikan dan bermanfaat, serta mendapatkan balasan yang lebih baik di akhirat kelak, dan mudah-mudahan skripsi ini bermanfaat bagi kita semua yang membacanya, Aamiin Allahumma Aamiin.

*Wassalamualaikum Wr. Wb.*

Jakarta, Juli 2023

Penulis

Ahmad Muzakir



## DAFTAR ISI

LEMBAR PERSETUJUAN .....	i
LEMBAR PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PENYATAAN PERSETUJUAN PUBLIKASI SKRIPSI .....	iv
ABSTRAK .....	v
ABSTRACT .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xvi
BAB I .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	6
1.3 Batasan Masalah .....	6
1.4 Tujuan Penelitian .....	7
1.5 Manfaat Penelitian .....	7
1.5.1 Manfaat Bagi Penulis .....	7
1.5.2 Manfaat Bagi Pembaca .....	8
1.5.3 Manfaat Bagi Universitas .....	8
1.6 Metodologi Penelitian .....	8
1.6.1 Metode Pengumpulan Data .....	8
1.6.2 Metode Pengembangan Sistem .....	9
1.7 Sistematika Penulisan .....	9
BAB II .....	12
2.1 Sistem Monitoring .....	12
2.2 <i>Internet Of Things (IoT)</i> .....	13
2.1.1 Layer IoT .....	13
2.1.2 Arsitektur IoT .....	17
2.1.3 Protokol IoT .....	20

2.1.4 Komunikasi IoT .....	29
2.3 Karakteristik Sumber Listrik .....	41
2.2.1 Tegangan listrik .....	41
2.2.2 Arus Listrik.....	41
2.2.3 Hambatan Listrik .....	42
2.2.4 Daya Listrik.....	42
2.2.5 Hubungan Antara Besaran-Besaran Listrik .....	42
2.4 Tarif listrik per KwH.....	42
2.5 Android Studio .....	44
2.6 Platform IoT .....	47
2.7 <i>Firebase</i> .....	50
2.8 <i>Fuzzy Logic</i> .....	53
2.9 Metode Sugeno.....	54
2.9.1 Fungsi Keanggotaan .....	56
2.9.2 Tahapan Metode Sugeno .....	63
2.10 ESP8266.....	81
2.10.1 Karakteristik ESP8266 .....	81
2.10.2 Komponen utama ESP8266.....	83
2.11 PZEM 004T .....	85
2.12 Prototyping .....	87
2.13 <i>Structured Analysis with Real-time (SA-RT)</i> .....	90
2.14 Data Visualization .....	90
2.15 Arduino IDE .....	91
2.16 Draw.lo .....	93
2.17 Visual studio code .....	94
2.18 Fritzing.....	97
2.19 Python .....	99
2.20 Flutter.....	102
2.21 Perangkat Android .....	105
2.22 <i>Unified Modelling Language</i> .....	106
BAB III Metode Penelitian .....	114
3.1 Metode Pengumpulan Data .....	114
3.1.1 Data primer .....	114

3.1.2 Data Sekunder.....	115
3.2 Metode Pengembangan Sistem.....	115
3.2.1 Tahap Komunikasi .....	115
3.2.2 Perencanaan cepat ( <i>Quick planning</i> ) .....	116
3.2.3 Pemodelan Cepat ( <i>Quick Modeling</i> ) .....	117
3.2.4 Konstruksi (Construction) .....	118
3.2.5 Evaluasi .....	118
3.3 Alur Penelitian.....	118
<b>BAB IV ANALISIS DESAIN, IMPLEMENTASI, DAN PENGUJIAN SISTEM</b>	
.....	120
4.1 Tahap Komunikasi .....	120
4.2 Tahap Perencanaan cepat ( <i>Quick planning</i> ) .....	121
4.2.1 Ruang Lingkup.....	121
4.2.2 Analisis Kebutuhan Hardware.....	121
4.2.3 Analisis Sistem Berjalan .....	122
4.2.4 Analisis Sistem Usulan.....	123
4.2.5 Analisa Kebutuhan <i>Software</i> dan <i>Tools</i> .....	124
4.2.6 Analisis kebutuhan Fungsional Sistem.....	125
4.3 Tahap Pemodelan cepat .....	127
4.3.1 Kemasan.....	127
4.3.2 Skematik Sistem dengan perangkat Hardware.....	129
4.3.3 Skematik dengan <i>Firebase</i> .....	132
4.3.4 Skematik dengan android dan Python .....	137
4.4 Kontruksi .....	142
4.4.1 Pengkodean Arduino IDE .....	142
4.4.2 Pembuatan <i>Firebase</i> .....	143
4.4.3 Pengkodean <i>Fuzzy Logic</i> dengan Metode Sugeno.....	146
4.4.4 Pengkodean aplikasi android Sistem monitoring Isitrik.....	155
4.5 Evaluasi dengan <i>Black Box Testing</i> .....	155
4.5.1 Pengujian Arduunio IDE .....	156
4.5.2 Pengujian <i>Firebase</i> .....	157
4.5.2 Pengujian Aplikasi Android .....	159
4.5.3 Pengujian <i>Fuzzy Logic</i> .....	160

4.5.4 Pengujian Output Terhadap Input Berdasarkan Aturan <i>Fuzzy</i> .....	164
4.5.5 Pengujian Lokasi Sistem monitoring listrik .....	165
4.5.6 Pengujian Gangguan Sistem.....	166
<b>BAB 5 HASIL DAN PEMBAHASAN.....</b>	<b>170</b>
5.1 Sistem Monitoring listrik.....	170
5.2 Internet Of Things .....	170
5.3 Hasil Arduino IDE.....	171
5.4 Hasil <i>Firebase</i> .....	172
5.5 Hasil Android.....	175
5.6 Hasil <i>Fuzzy Sugeno</i> .....	181
<b>BAB 6 KESIMPULAN DAN SARAN .....</b>	<b>182</b>
6.1 Kesimpulan.....	182
6.2 Saran .....	183
Daftar Pustaka .....	184
Lampiran .....	186

## DAFTAR GAMBAR

Gambar 2. 1 Arsitektur IoT (Sumber :Dgtl Infra ).....	17
Gambar 2. 2 Protokol IoT Stack (sumber : particle).....	25
Gambar 2. 3 Komponen Arsitektur IoT .....	40
Gambar 2. 4 Flowchart IoT peneliti.....	41
Gambar 2. 5. HUKUM OHM LISTRIK .....	42
Gambar 2. 6 Tarif listrik per KwH.....	43
Gambar 2. 7 Proses <i>firebase</i> dan ESP8266 (Sumber : Nerd).....	51
Gambar 2.8 Cara Kerja <i>Firebase</i> (Sumber :Nyebarinilmu.com).....	52
Gambar 2. 9 Keanggotaan linear (sumber Rizky).....	57
Gambar 2. 10 Keanggotaan Segitiga (Sumber : Researhgate).....	57
Gambar 2. 11 Keanggotaan Trapesium (Sumber : ResearchGate) .....	58
Gambar 2.12 Kurva gauss (Sumber : ResearchGate).....	60
Gambar 2. 13 Kurva generalized bell (Sumber : ResearchGate).....	61
Gambar 2. 14 Kurva sigmoid(Sumber : ResearchGate).....	62
Gambar 2. 15 Fungsi keanggotaan biaya .....	67
Gambar 2.16 Varibel keanggotaan output <i>Fuzzy</i> .....	73
Gambar 2. 17 ESP8266 (Sumber pitco robo) .....	81
Gambar 2. 18 PIN NodeMCU ESP8266 (Sumber : smartkits Blog) .....	82
Gambar 2. 19 PZEM-004T (Sumber : Vas Labs) .....	85
Gambar 2. 20 alur prototipe (Sumber: Metode Prototype) .....	87
Gambar 2. 21 contoh grafik Time series grarfik (Sumber : R-bloogger).....	91
Gambar 2. 22 Arduino IDE (Sumber : Mspoweruser).....	91
Gambar 2. 23 tampilan Arduino.....	92
Gambar 2. 24 Icon draw io (sumber: draw.io).....	93
Gambar 2. 25 Cara kerja Draw.Io .....	93
Gambar 2. 26 Tampilan Visual Studio Code .....	97
Gambar 2. 27 Tampilan Fritzing penelitian .....	99
Gambar 2. 28 Penerapan Python untuk IoT .....	102
Gambar 2. 29 Tampilan Flutter .....	105
Gambar 2. 30 Tampilan aplikasi di Perangkat Android.....	106
Gambar 3. 1 Mapping Studi literatur .....	111
Gambar 3. 2 Alur Penelitian.....	119
Gambar 4. 1 Analisis Sistem Berjalan .....	
Gambar 4. 2 Analisis Sistem Usulan.....	123
Gambar 4. 3 DFD Level 0.....	125
Gambar 4. 4 DFD Level 1.....	126
Gambar 4. 5 Kemasan Hardware .....	127
Gambar 4. 6 hasil kemasan .....	128
Gambar 4. 7 Skematik full Sistem dengan Sensor Daya listrik dan ESP8266 .....	

Gambar 4. 8 Flowchart/DFD Level 2 bagian hardware .....	130
Gambar 4. 9 Skema <i>Firebase</i> dan ESP8266 .....	132
Gambar 4. 10 Struktur NoSQL <i>Firebase</i> .....	134
Gambar 4. 11 DFD level 2 bagian <i>firebase</i> .....	135
Gambar 4. 12 Use Case .....	137
Gambar 4. 13 Squence diagram user.....	
Gambar 4. 14 Activity diagram.....	139
Gambar 4. 15 Data Flow Diagram Android.....	140
Gambar 4. 16 Tampilan Arduino IDE.....	142
Gambar 4. 17 Tahap pembuatan <i>Firebase</i> part 1 .....	143
Gambar 4. 18 Tahap pembuatan <i>Firebase</i> part 2 .....	143
Gambar 4. 19 Tahap pembuatan <i>Firebase</i> part 3 .....	144
Gambar 4. 20 Tahap pembuatan <i>Firebase</i> part 4 .....	144
Gambar 4. 21 Tahap pembuatan <i>Firebase</i> part 5 .....	144
Gambar 4. 22 Tahap pembuatan <i>Firebase</i> part 6 .....	145
Gambar 4. 23 Tahap pembuatan <i>Firebase</i> part 7 .....	145
Gambar 4. 24 Tahap pembuatan <i>Firebase</i> part 9 .....	146
Gambar 5. 1 Serial monitor part 1.....	171
Gambar 5. 2 Serial Monitor Part 2 .....	172
Gambar 5. 3 Hasil <i>Firebase</i> (panel view) .....	173
Gambar 5. 4 Hasil <i>Firebase</i> (Query builder) .....	174
Gambar 5. 5 Hasil <i>Firebase</i> minotor rules.....	175
Gambar 5. 6 Tampilan Login .....	176
Gambar 5. 7 Tampilan Realtime .....	177
Gambar 5. 8 Tampilan analisis memilih tanggal .....	178
Gambar 5. 9 Tampilan tabel analisis.....	179
Gambar 5. 10 Tampilan Logout .....	180
Gambar 5. 11 Tampilan Fuzzy atau analisis grafik .....	181

## **DAFTAR TABEL**

Tabel 2. 1 Layer IoT peneliti .....	15
Tabel 2. 2 Arsitektur IoT Peneliti .....	18
Tabel 2. 3 Layer Protokol IoT peneliti.....	27
Tabel 2. 4 Tipe Komunikasi Peneliti .....	31
Tabel 2.5 Model Komunikasi peneliti.....	35
Tabel 2. 6 Model Konektivitas peniliti .....	38
Tabel 2. 7 Variabel Himpunan Fuzzy .....	64
Tabel 2. 8 tabel data keanggotaan Fuzzy .....	72
Tabel 2. 9 Keanggotaan output Fuzzy.....	73
Tabel 2. 10 Rule keputusan Fuzzy .....	74
Tabel 2. 11 Karakteristik esp .....	81
Tabel 3. 1 Studi Literatur sejenis .....	107
Tabel 4. 1 Analisis kebutuhan hadrware .....	121
Tabel 4. 2 Kebutuhan software .....	124
Tabel 4. 3 Pin Hardware.....	129
Tabel 4. 4 Pengujian Ardunio IDE.....	156
Tabel 4. 5 Pengujian Blcak box testing Firebase .....	157
Tabel 4. 6 Pengujian Aplikasi Android.....	159
Tabel 4. 7 black box testing .....	161
Tabel 4. 8 Rules pengujian Fuzzy .....	163
Tabel 4. 9 Fuzzy Output pengujian .....	163
Tabel 4. 10 Output terhadapa input berdasarkan aturan Fuzzy .....	164
Tabel 4. 11 Pengujian Lokasi.....	165
Tabel 4. 12 Pengujian gangguan sistem.....	166

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Energi listrik merupakan salah satu kebutuhan pokok yang sangat penting bagi kehidupan manusia saat ini . Hampir setiap kegiatan yang dilakukan manusia berhubungan dengan energi listrik Salah satunya konsumen rumah tangga membutuhkan energi listrik untuk dapat melakukan aktivitas. Hal ini menyebabkan konsumsi akan energi listrik semakin boros.oleh karena itu listrik merupakan kebutuhan yang mutlak bagi semua orang, untuk itu perlu adanya sistem yang dapat melakukan monitoring penggunaan listrik agar pemakaian listrik menjadi lebih terkendali(Samallo et al., 2020).

Berdasarkan laporan PLN,(PLN, 2021) konsumsi lisrik rumah tangga menjadi kontributor terbesar terhadap struktur PDB tahun 2021, yaitu sebesar 52,91%. Selama tahun 2021, konsumsi listrik rumah tangga mengalami pertumbuhan sebesar 3,55%, kontras dengan pertumbuhan di tahun 2020 sebesar -3,61%. Dimana mayoritas pelanggannya masih menggunakan jenis meteran listrik manual atau analog dengan sistem pembayaran bulanan atau pascabayar. Banyak kasus terjadi seputar penggunaan listrik prabayar yang tidak terkendali, seperti token listrik yang diisi dalam jumlah kWh yang banyak, namun tenaga listrik yang dialirkan tidak bertahan lama. Dimana pelanggannya rata-rata tidak mengetahui jumlah pemakaian listriknya selama pemakaian di bulan berjalan. Dan hanya akan mengetahui pada saat tagihan listrik bulanan sudah terbit atau dikeluarkan oleh pihak PLN (Mitha Djaksana et al., 2020). Apabila sering mengalami overload listrik akan membuat perangkat elektronik rusak dan akan menambah pengeluaran biaya sehingga pengeluaran dalam rumah tangga akan semakin banyak.Dengan penggunaan listrik pada rumah tangga tidak terkontrol sehingga dapat membuat tagihan pada rumah tangga sangat tinggi atau terkendali, dibutuhkan informasi penggunaan harian dari listrik prabayar untuk mempermudah dalam melakukan

penghematan energi listrik, sehingga pengguna akan memahami terkait dasar kebutuhan arus listrik.(Murdiyat et al., 2021)

*Internet of Things* merupakan sebuah gagasan yang bertujuan untuk memperluas fungsi dari koneksi internet yang terhubung secara terus-menerus. Adapun kegunaan yang dimiliki seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata(Tukadi et al., 2019). Dalam Penelitian *IoT* (*Internet of Things*) menjadi topik menarik untuk diterapkan dalam beberapa penelitian di bidang kontrol dan monitoring. Dalam teknologi IoT sistem kontrol dan monitoring dari sistem yang dirancang dapat dilakukan secara portable dengan memanfaatkan Android dari user. Oleh karena itu IoT menjadi solusi yang penting dan menarik untuk dikembangkan. Demikian juga dengan topik penelitian monitoring pemakaian daya listrik telah memasuki era tersebut(Zaen & Rozaq, 2021)

Dalam Monitoring yang dilakukan (Ma & Hamdani, 2019) Mengukur dan mengetahui penggunaan daya listrik dirumah tangga, Implementasi internet memudahkan user dalam memonitoring pemakaian listrik dengan jarak jauh sebagai keuntungan dari sistem yang dirancang. Penilitian ini berguna untuk memberikan informasi kepada user listrik rumah tangga dalam penggunaan listrik.Pada penelitian (Nasional et al., 2021)melakukan perancangan KWh listrik berbasis internet menggunakan mikrokontroler dengan sensor arus yang bisa di akses oleh ponsel. Bedasarsrkan sumber Tokopedia (Tokpedia, 2022), Terdapat barang monitoring listrik yang digunakan tidak secara Jarak jauh tetapi mengetahui pemakaian daya listrik menggunakan Lcd.

*Firebase* adalah platform seluler yang membantu developer mengembangkan aplikasi berkualitas tinggi secara cepat dan berbasis pengguna, *Firebase* terdiri dari fitur lengkap yang bisa dipadukan sesuai dengan kebutuhan (Google) Data yang tersimpan di database dalam komputer yang terhubung internet. Dalam penyimpanan data di IoT dan mesinkronkan antara pengguna secara Realtime bisa dalam keadaan offline disebut *Firebase Realtime Database*, Pada *Firebase* akan tersimpan data daya, waktu yang meliputi tanggal hingga jam dan juga status

kondisi listrik tersambung atau terputus. Kemudian pemilik kos dapat melihat data tersebut menggunakan aplikasi android yang terhubung dengan *Firebase* yang digunakan (Mitha Djaksana et al., 2020).

Menurut (Dwisaputra et al., 2021) Android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak dalam Judul Kontrol dan Monitoring Stop Kontak Berbasis Android, Kontrol dan Monitoring Stop Kontak Berbasis Android. Penelitian yang berjudul *Android-Based Monitoring and Control of Electricity Consumption* (Samallo et al., 2020) tampilan Android untuk monitoring pemakaian listrik(KwH).

*Fuzzy Logic* dapat digunakan untuk menyampaikan informasi dari data yang bersifat *ambiguous*. Terdapat beberapa metode dalam logika *Fuzzy* yaitu; metode Tsukamoto, metode Mamdani dan metode Sugeno(Rizky et al., 2020). Metode Sugeno merupakan suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan(Suprapto & Simanjuntak, 2020). *Fuzzy logic* pada penelitian ini digunakan untuk menentukan keputusan apakah pemakaian listrik lebih hemat dalam pengukuran KwH di rumah.

Data *Fuzzy logic* mengambil dari jurnal (Suprapto & Simanjuntak, 2020) dengan judul *Fuzzy logic* untuk memprediksi pemakaian listrik , terdapat variabel *Fuzzy* pengguna listrik yaitu KwH atau daya listrik dalam waktu perbulan dengan keanggotaan yang sudah diolah menjadi 3 keanggotaan variabel KwH. Data tersebut digunakan pada penelitian kali ini untuk menjadi acuan variabel tersebut.

Berdasarkan hasil penelitian yang dilakukan oleh(Rizky et al., 2020) menyatakan bahwasannya dari ketiga metode yang dimiliki oleh *Fuzzy logic*, metode Sugeno memiliki nilai *error* terkecil dibandingkan dengan metode lain,

sehingga berdasarkan hasil yang didapat metode Sugeno yang paling baik untuk digunakan dalam menentukan keputusan.

Menurut (Lailla, 2023) ada beberapa cara Mengetahui Kapasitas Daya kWh Meter Listrik yaitu mengukur daya listrik seperti CL meteran atau alat komersial dengan kelebihan tahan terhadap gangguan listrik tetapi memiliki kelemahan fitur yang terbatas dan tidak dapat dikustomisasi sesuai dengan kebutuhan pengguna, dan Mengecek Daya Listrik Melalui Website PLN, Aplikasi pengisi listrik(Tokopedia,Ovo dan lainnya) ,Internet banking dan SMS banking yang memberikan Kemudahan yang ditawarkan PLN berupa pengecekan daya listrik dan dapat melakukan pembayaran tagihan dengan mudah dan cepat melalui internet,ponsel atau layanan perbankan tetapi memiliki kekurangan tidak menyediakan fitur monitoring daya listrik secara real-time dan tidak mempunyai analisis data listrik dalam waktu yang ditentukan oleh pengguna listrik.

Sementara itu, sistem monitoring listrik berbasis IoT menggunakan *Firebase* dan algoritma *Fuzzy logic* Sugeno yang diimplementasikan pada sistem android pada penelitian ini akan memiliki beberapa kelebihan yaitu Sistem ini dapat dikustomisasi sesuai dengan kebutuhan pengguna, misalnya dapat terdapat fitur-fitur monitoring listrik ,Dapat diakses dari jarak jauh, Informasi yang lebih detail;Sistem ini dapat memberikan informasi yang lebih detail tentang penggunaan listrik secara real-time , misalnya konsumsi listrik pada waktu tertentu, pemakaian listrik per perangkat, dan mengetahui hasil pengguna dalam waktu harian dengan metode *Fuzzy logic sugeno* .

Dalam penelitian (Pela & Pramudita, 2021) ESP8266 dilengkapi dengan chip microcontroller yang mengontrol semua fungsi dalam modul, termasuk koneksi WiFi, pengiriman dan penerimaan data, dan pemrosesan data. Microcontroller ini memungkinkan ESP8266 untuk diatur dan dikontrol melalui kode pemrograman, jaringan WiFi memungkinkan ESP8266 untuk mengirim dan menerima data secara nirkabel, sehingga dapat digunakan dalam berbagai proyek IoT pada monitoring daya listrik.

Sensor PZEM004T digunakan untuk keperluan monitoring daya listrik pada sistem rumah tangga, industri, atau pada proyek-proyek IoT yang membutuhkan pengukuran daya listrik. Keunggulan dari PZEM004T adalah mudah digunakan dan dapat diintegrasikan dengan berbagai platform mikrokontroler seperti Arduino, ESP8266, PZEM004T juga dilengkapi dengan fitur komunikasi serial yang memungkinkan penggunaannya dengan berbagai aplikasi dan sistem monitoring yang memerlukan pengukuran daya listrik(Dwisaputra et al., 2021).

Secara keseluruhan, rancang bangun penelitian ini akan berhasil memadukan teknologi IoT dengan algoritma *Fuzzy Logic Sugeno* dalam sistem monitoring daya listrik yang dapat dikontrol melalui aplikasi Android. dari jurnal yang dijadikan acuan beberapa fungsi menjadi sistem yang dikembangkan. Monitrong yang peneliti kembangkan selain melihat *interface* hasil dari monitron listrik yang dilakukan alat hardware(PZEM004T dan ESP8266) melalui *smartphone* atau alat genggam dengan melihat tagihan listrik,daya listrik dan harga listrik dengan waktu yang ditentukan pengguna yaitu peneliti mengembangkan fitur penggunaan listrik dengan output pengguna rendah,sedang dan tinggi dengan metode algoritma *Fuzzy logic sugeno* sebagai pengambilan keputusan, output dari hasil tersebut berupa grafik untuk mempermudah membaca hasil akhir. Output dalam aplikasi android menampilkan rincian listrik dalam bentuk tabel , jumlah energi listrik dan biaya listrik sesuai dengan permintaan tanggal yang ingin diliat oleh pengguna listrik.

Penelitian ini berfungsi untuk Pengguna bisa mengetahui KWh yang digunakan di rumah tanpa dari PLN.Oleh karena itu, Peneliti merancang Prototipe agar pengguna listrik mengetahui KWh yang dikeluarkan dengan menggunakan microcontroller NodeMCU ESP8266 V3 yang sudah terintegrasi Wi-Fi,dan modul pembaca arus listrik PZEM 004T. penelitian ini yang berjudul **“Implementasi Internet of Thing (IOT) pada Sistem Monitoring Daya Listrik Menggunakan Firebase Berbasis Android Dengan Algoritma Fuzzy Logic”** bisa mengontrol untuk kuantitas yang dikeluarkan oleh daya listrik rumah tangga dengan terhubung intenet dengan Algoritma *Fuzzy logic* membantu proses dari keputusan sistem monitoring listrik.

Dengan memanfaatkan sistem IoT monitoring listrik, diharapkan pengguna dapat mengoptimalkan penggunaan energi listrik di rumah atau bangunan, mengurangi biaya tagihan listrik, dan meningkatkan efisiensi energi. Selain itu, sistem ini juga dapat membantu mengidentifikasi perangkat listrik yang membutuhkan perbaikan atau penggantian, serta mencegah overloading atau korsleting listrik yang dapat membahayakan keselamatan pengguna.

## 1.2 Rumusan Masalah

Dari latar belakang tersebut, maka dapat dirumuskan permasalahan dalam skripsi ini yaitu

1. Bagaimana pengguna listrik bisa mengetahui penggunaan daya listrik secara mandiri menggunakan *mobile* berbasis android?
2. Bagaimana cara Pengguna listrik memantau (*Monitoring*) daya listrik *Internet Of things* menggunakan algoritma *Fuzzy logic* sugeno dan *firebase* berbasis android?

## 1.3 Batasan Masalah

Agar pembahasan yang akan dipaparkan oleh penulis sesuai dengan latar belakang, maka penulis akan membatasi permasalahan sebagai berikut :

1. Implementasi Internet of Things (IoT) pada sistem monitoring daya listrik menggunakan *Firebase* berbasis Android dengan algoritma *Fuzzy Logic* Sugeno.
2. Sistem monitoring daya listrik ini dirancang untuk mengukur dan memantau daya listrik pada suatu tempat secara real-time, sehingga pengguna dapat mengetahui besaran daya listrik yang sedang digunakan dan memantau konsumsi daya listrik sepanjang waktu.
3. Rumah tangga menengah kebawah dengan tegangan Rumah tangga listrik (900VA,1300VA,2200VA) sebagai objek untuk penerapan sistem monitoring listrik.
4. *Firebase* digunakan sebagai penyimpanan data untuk memudahkan akses dan memproses data yang dihasilkan oleh sistem monitoring.

5. Algoritma *Fuzzy Logic* Sugeno digunakan untuk mengoptimalkan penggunaan energi dengan mengatur nilai daya listrik yang dihasilkan oleh sistem berdasarkan kondisi yang sedang terjadi.
6. Nilai yang ditetapkan dari proses Algoritma *Fuzzy Logic* Sugeno digunakan untuk pengambilan keputusan menentukan penggunaan listrik Rendah, Sedang dan Tinggi dalam waktu perhari yang dimunculkan berupa Visualisasi data yaitu grafik.
7. Sistem ini diimplementasikan pada *framework flutter* dibuat aplikasi untuk operasi sistem android versi berapapun sebagai antarmuka pengguna untuk mengetahui hasil dari data yang diolah atau data hasil dari *microcontroller*.
8. Nilai *Fuzzyifikasi* daya listrik dalam penelitian menggunakan variabel listrik yaitu daya listrik (KwH), Tegangan Rumah listrik (900VA,1300VA,2200VA) jumlah perangkat dan Harga listrik harian rumah tangga untuk diolah di python.
9. Alat hardware yang digunakan menggunakan *microcontroller* ESP8266 dan PZEM004T sebagai sensor untuk mengetahui daya listrik.

#### **1.4 Tujuan Penelitian**

Tujuan diadakannya penelitian ini yaitu untuk :

1. Pengguna dapat mengetahui daya penggunaan listrik setiap daya listrik yang keluar dengan rancangan alat yang dibuat secara mandiri menggunakan moblie berbasis android.
2. Mengetahui bagaimana cara melakukan *monitoring* daya listrik dengan menggunakan *firebase* berbasis android dengan algoritma *Fuzzy logic* sugeno.

#### **1.5 Manfaat Penelitian**

Adapun manfaat penelitian yang diharapkan dari penelitian ini sebagai berikut:

##### **1.5.1 Manfaat Bagi Penulis**

1. Menambah Pemahaman dan wawasan dalam bidang *Internet of Things*.

2. Memberikan pemahaman dan wawasan mengenai sistem monitoring listrik dengan menggunakan *firebase* dan Arduino IDE dengan metode *Fuzzy Logic Sugeno*.
3. Mendapatkan pemahaman dan wawasan tentang merancang sistem monitoring listrik dengan menggunakan *firebase* dan Arduino IDE dengan metode *Fuzzy Logic Sugeno* secara baik dan benar.

#### 1.5.2 Manfaat Bagi Pembaca

Penelitian ini diharapkan dapat menambah pengertian tentang *Internet of Things*. Monitoring listrik pada kehidupan sehari – hari dan sebagai referensi untuk mengembangkan Penelitian ini.

#### 1.5.3 Manfaat Bagi Universitas

1. Menambah kumpulan skripsi dari salah satu mahasiswa Teknik Informatika Fakultas Sains dan Teknologi
2. Dapat mengetahui bagaimana kemampuan mahasiswa dalam mengimplementasikan pembelajaran selama kuliah
3. Dapat dijadikan bahan referensi bagi penelitian yang sejenis di masa mendatang

### 1.6 Metodologi Penelitian

Metode yang digunakan penulis dalam penulisan dan penelitian dibagi menjadi dua, yaitu metode pengumpulan data dan metode pengembangan sistem. Berikut penjelasan kedua metode tersebut:

#### 1.6.1 Metode Pengumpulan Data

Dalam melakukan analisis data dan penulisan skripsi ini, penulis menggunakan 2 metode pengumpulan data, yaitu:

1. Data Primer : Studi Lapangan dengan menggunakan metode observasi dan wawancara.
2. Data sekunder : Studi pustaka yang meliputi referensi berupa buku-buku, website, dan studi literatur(berupa tabel perbandingan dan mapping literatur).

### **1.6.2 Metode Pengembangan Sistem**

Pada penelitian ini digunakan metode pernembangan sistem *Prototyping* (prototipe), terdapat 5 tahapan dalam metode ini (Vieri, 2021):

1. Komunikasi.
2. Perencanaan cepat.
3. Pemodelan cepat.
4. Konstruksi.
5. Evaluasi

### **1.7 Sistematika Penulisan**

Sistematika penulisan yang akan dilakukan dalam penelitian ini terdiri dari 6 bab, yaitu:

#### **BAB I**

#### **PENDAHULUAN**

Dalam bab ini akan dibahas mengenai latar belakang, Batasan masalah, tujuan, dan manfaat serta sistematika penulisan.

#### **BAB II**

#### **TINJAUAN PUSTAKA DAN LANDASAR TEORI**

Dalam bab ini akan dibahas mengenai berbagai teori yang mendasari analisis permasalahan dan berhubungan dengan topik yang dibahas.

#### **BAB III**

#### **METODOLOGI PENELITIAN**

Bab ini membahas mengenai metode penelitian yang akan digunakan dalam merancang dan membangun prototipe sistem.

#### **BAB IV**

#### **ANALISIS, DESAIN, IMPLEMENTASI, DAN PENGUJIAN SISTEM**

Pada bab ini membahas mengenai hasil dari analisis, perancangan, implementasi sesuai dengan metode yang dilakukan pada sistem

yang dibuat serta hasil dari pengujian.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi hasil dari pembahasan yang didapat dari penelitian.

## **BAB VI KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diusulkan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih baik.



## BAB II

### LANDASAN TEORI

#### 2.1 Sistem Monitoring

Sistem dapat diartikan sebagai kumpulan elemen yang terorganisir dan saling berinteraksi untuk mencapai tujuan tertentu. Sistem terdiri dari berbagai komponen atau unsur yang bekerja bersama secara harmonis dan terkoordinasi untuk menjalankan fungsi-fungsi tertentu (Riadi, 2020).

Menurut KBBI (KBBI, 2021) , Monitoring ialah suku awal ‘Monitor’ yang dalam artinya alat untuk memantau (seperti alat penerima yang digunakan untuk melihat gambar yang diambil oleh kamera televisi, alat untuk mengamati kondisi atau fungsi biologis, alat yang memantau kerja suatu sistem, terutama sistem komputer, dan sebagainya) dan Memonitor ialah mengawasi, mengamati, atau mengecek dengan cermat, terutama untuk tujuan khusus.

Sistem monitoring dapat dikategorikan sebagai bagian dari sistem otomasi (growth), 2021). Dengan menggunakan teknologi dan alat yang tepat, monitoring system dapat dilakukan secara otomatis tanpa memerlukan pengawasan manual yang intensif. Sistem otomasi memungkinkan pengumpulan data, analisis, dan pelaporan dilakukan secara otomatis, sehingga memudahkan pengguna untuk memantau kinerja, kondisi, atau parameter tertentu dalam suatu sistem.

Dalam penelitian ini, sistem monitoring yang dikembangkan difokuskan pada pemantauan dan pengawasan penggunaan listrik. Sistem ini bertujuan untuk memberikan informasi secara real-time tentang konsumsi listrik, pemakaian daya, dan faktor-faktor terkait lainnya kepada pengguna, Dengan adanya sistem monitoring listrik, pengguna dapat memantau dan mengontrol penggunaan listrik mereka dengan lebih efektif. Mereka dapat melihat data penggunaan listrik dalam bentuk grafik, tabel, atau visualisasi lainnya

## 2.2 Internet Of Things (IoT)

*Internet of Things* merupakan sebuah ide yang bertujuan untuk Mengembangkan fungsi dari koneksi internet yang terhubung secara terus-menerus. Adapun kegunaan yang dimiliki seperti berbagi data, remote control, dan sebagainya, termasuk juga pada benda di dunia nyata (Tukadi et al., 2019).

Penelitian yang dilakukan (Pela & Pramudita, 2021) Internet Of Things untuk kegiatan sehari-hari pada rumah tangga. Yang dilakukan peneliti menyatakan bahwa sebuah penggunaan daya listrik di rumah tangga selama ini hanya dapat dilihat melalui alat ukur kWh meter yang hanya didistribusikan oleh PLN.

### 2.1.1 Layer IoT

Dalam penjelasan dari (Jhonson, 2022) Layer IoT untuk monitoring listrik terdiri dari beberapa komponen yang terhubung untuk membentuk suatu sistem yang dapat mengukur, memproses, dan menganalisis data penggunaan listrik. Berikut adalah penjelasan mengenai layer-layer tersebut:

#### a. Perangkat Keras (*Hardware Devices*)

Perangkat keras adalah komponen utama dalam sistem monitoring listrik. Perangkat keras yang digunakan untuk monitoring listrik biasanya berupa sensor-sensor yang dapat mengukur besaran daya listrik seperti arus, tegangan, dan frekuensi. Selain sensor, perangkat keras lain seperti mikrokontroller, modul wifi/ethernet, dan relay juga diperlukan untuk menghubungkan sensor ke jaringan internet dan mengontrol penggunaan listrik.

#### b. Jaringan (Network)

Jaringan adalah komponen yang menghubungkan perangkat keras dan perangkat lunak dalam sistem monitoring listrik. Jaringan yang digunakan biasanya berupa jaringan nirkabel (wireless), seperti Wi-Fi,

Bluetooth, atau Zigbee, namun juga dapat menggunakan kabel (wired) seperti Ethernet.

#### c. Middleware

Middleware adalah komponen yang berfungsi sebagai penghubung antara perangkat keras dan perangkat lunak. Middleware mengelola data yang dikirim oleh perangkat keras dan menyimpannya ke dalam database atau platform cloud. Middleware juga dapat memproses data, seperti menghitung jumlah energi yang digunakan atau menentukan tingkat pemakaian listrik yang optimal.

#### d. Cloud Platform

Cloud platform adalah platform yang digunakan untuk menyimpan, memproses, dan menganalisis data yang dikirim oleh perangkat keras melalui middleware. Cloud platform juga dapat digunakan untuk memonitor penggunaan listrik dari jarak jauh, mengontrol penggunaan listrik, dan mengoptimalkan penggunaan energi.

#### e. Aplikasi

Aplikasi adalah antarmuka yang digunakan oleh pengguna untuk mengakses data dan mengontrol penggunaan listrik. Aplikasi dapat berupa aplikasi web atau aplikasi mobile yang terkoneksi dengan cloud platform. Aplikasi dapat menampilkan data penggunaan listrik dalam bentuk grafik atau tabel, serta memberikan notifikasi jika terdapat pemakaian listrik yang berlebihan.

Dengan menggunakan layer-layer tersebut, sistem monitoring listrik dapat diimplementasikan dengan efektif dan efisien. Data penggunaan listrik dapat diakses dan dianalisis dengan mudah, sehingga dapat membantu pengguna dalam menghemat biaya dan mengurangi

penggunaan energi, berikut tabel layer layer yang diterapkan pada penelitian yang dibuat :

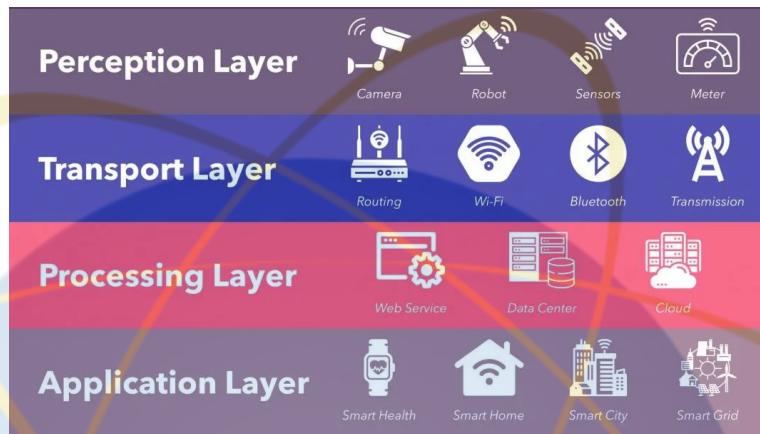
Tabel 2. 1 Layer IoT peneliti

Layer	Penjelasan
Layer Hardware	Lapisan hardware terkait dengan perangkat keras yang digunakan dalam sistem monitoring daya listrik. Pada skripsi ini, lapisan hardware dapat melibatkan perangkat seperti mikrokontroler (ESP8266), sensor daya listrik (PZEM-004T), dan komponen pendukung lainnya yang diperlukan untuk mengukur dan mengirim data daya listrik.
Layer Jaringan	Lapisan jaringan berhubungan dengan koneksi antara perangkat IoT dengan infrastruktur jaringan. Pada skripsi ini, lapisan jaringan melibatkan protokol komunikasi seperti Wi-Fi atau Ethernet untuk menghubungkan perangkat IoT ke internet. Protokol HTTP dapat digunakan untuk mentransmisikan data antara perangkat IoT dan <i>Firebase</i> . Protokol ini memungkinkan perangkat untuk mentransmisikan data ke lapisan berikutnya.
Layer Middleware	Lapisan middleware berperan sebagai penghubung antara perangkat keras (hardware) dan platform cloud (cloud platform). Pada penelitian ini, lapisan middleware dapat berfungsi sebagai perantara antara perangkat IoT dan <i>Firebase</i> , yang digunakan sebagai platform

	cloud untuk penyimpanan data dan layanan backend.
Layer Cloud Platform	Lapisan cloud platform merupakan tempat penyimpanan data dan komputasi yang dilakukan secara cloud-based. Pada penelitian ini, <i>Firebase</i> berperan sebagai cloud platform yang digunakan untuk menyimpan data monitoring daya listrik dan menyediakan layanan backend untuk sistem. <i>Firebase</i> menyediakan API dan layanan yang memungkinkan interaksi dengan data dan integrasi dengan aplikasi Android.
Layer Aplikasi IoT	Lapisan aplikasi IoT terkait dengan pengembangan aplikasi yang berinteraksi dengan perangkat IoT dan menyediakan antarmuka pengguna untuk mengakses dan mengontrol sistem. Pada skripsi ini, lapisan aplikasi melibatkan pengembangan aplikasi Android yang memanfaatkan data dari <i>Firebase</i> untuk menampilkan informasi monitoring daya listrik kepada pengguna. Aplikasi ini dapat memberikan visualisasi data, pengaturan alarm, dan fitur lainnya untuk memantau dan mengontrol penggunaan daya listrik.

Dengan menggunakan lapisan-lapisan tersebut, penelitian ini membangun sistem monitoring daya listrik yang mengintegrasikan perangkat keras, koneksi jaringan, platform cloud, dan aplikasi Android. Hal ini memungkinkan pengguna untuk memantau dan mengontrol penggunaan daya listrik secara efektif melalui antarmuka yang mudah digunakan.

### 2.1.2 Arsitektur IoT



Gambar 2. 1 Arsitektur IoT (Sumber :Dgtl Infra )

Dalam konteks *Internet of Things* (Priya, 2022), arsitektur umum sering kali terdiri dari empat lapisan utama: *perception layer*, *transport layer*, *processing layer*, dan *application layer*. Berikut adalah penjelasan singkat tentang masing-masing lapisan:

a. *Perception Layer* (Lapisan Persepsi)

Lapisan Persepsi adalah lapisan terbawah dalam arsitektur IoT. Lapisan ini berfungsi untuk mengumpulkan data dari berbagai sumber dan perangkat sensor yang terhubung ke jaringan IoT. Sensor-sensor ini dapat mencakup berbagai jenis, seperti sensor suhu, sensor gerak, sensor kelembaban, dan sebagainya. Data yang dikumpulkan oleh sensor-sensor ini kemudian dikonversi menjadi format yang dapat dipahami oleh sistem.

b. *Transport Layer* (Lapisan Transportasi)

Lapisan Transportasi bertanggung jawab atas pengiriman data dari lapisan persepsi ke lapisan pemrosesan. Lapisan ini menyediakan protokol dan mekanisme untuk mentransfer data secara aman dan

efisien antara perangkat IoT dan infrastruktur jaringan. Protokol yang umum digunakan di lapisan ini adalah protokol jaringan seperti WiFi dan Prtokol data seperti HTTP.

c. *Processing Layer* (Lapisan Pemrosesan)

Lapisan Pemrosesan merupakan lapisan di mana data dari lapisan transportasi diolah dan dianalisis. Di lapisan ini, data diklasifikasikan, diproses, dan diekstraksi informasi yang relevan. Pada tahap ini, algoritma dan teknik pemrosesan data seperti algoritma *Fuzzy logic*, machine learning, atau analisis statistik dapat diterapkan untuk mendapatkan wawasan dan informasi yang berarti dari data IoT.

d. *Application Layer* (Lapisan Aplikasi)

Lapisan Aplikasi adalah lapisan teratas dalam arsitektur IoT. Lapisan ini berfungsi sebagai antarmuka pengguna yang menghubungkan sistem IoT dengan pengguna akhir. Aplikasi yang dikembangkan di lapisan ini dapat beragam, mulai dari aplikasi mobile, aplikasi web, hingga sistem manajemen yang mengintegrasikan data dan menghasilkan laporan atau notifikasi bagi pengguna.

Dengan penjelasan diatas, peneliti memberikan arsitektur yang peneliti lakukan sebagai tabel berikut :

Tabel 2. 2 Arsitektur IoT Peneliti

Layer	Penjelasan
<i>Perception Layer</i>	Pada lapisan Persepsi, terdapat sensor-sensor yang terhubung ke sistem untuk mengumpulkan data daya listrik. Sensor-sensor ini dapat berupa sensor arus, sensor tegangan, atau perangkat meteran listrik yang mendeteksi

	dan mengukur penggunaan daya listrik. Data yang dikumpulkan oleh sensor-sensor ini menjadi input untuk sistem monitoring.
<i>Transport Layer</i>	Lapisan Transportasi bertanggung jawab atas pengiriman data dari lapisan Persepsi ke lapisan berikutnya dalam sistem. Dalam penelitian ini, lapisan ini dapat mencakup protokol komunikasi yang digunakan untuk mengirimkan data daya listrik dari sensor ke server <i>Firebase</i> . Protokol seperti HTTP dapat digunakan untuk mentransfer data secara aman dan efisien.
<i>Processing Layer</i>	Pada lapisan Pemrosesan, data daya listrik yang diterima dari lapisan Transportasi akan diproses dan dianalisis. Di sini, algoritma <i>Fuzzy Logic Sugeno</i> diterapkan untuk mengolah data dan menghasilkan keluaran berupa nilai daya listrik yang telah diuji secara <i>Fuzzy</i> dan diambil keputusan berdasarkan aturan <i>Fuzzy</i> . Pada tahap ini, pemrosesan data dan komputasi dilakukan untuk memperoleh informasi yang relevan.
<i>Application Layer</i>	Lapisan Aplikasi adalah lapisan teratas dalam skripsi ini. Pada lapisan ini, aplikasi Android dikembangkan untuk memberikan antarmuka pengguna yang memungkinkan pengguna untuk memantau dan mengelola data daya listrik secara real-time. Aplikasi ini dapat

	menampilkan informasi seperti penggunaan daya listrik saat ini, riwayat penggunaan, dan memberikan laporan kepada pengguna.
--	---

Masing-masing lapisan memiliki peran penting dalam membangun sistem monitoring daya listrik. Lapisan Persepsi mengumpulkan data daya listrik, lapisan Transportasi mengirimkan data ke server *Firebase*, lapisan Pemrosesan menerapkan algoritma *Fuzzy Logic Sugeno* untuk memproses data, dan lapisan Aplikasi menyediakan antarmuka pengguna yang memungkinkan pengguna untuk berinteraksi dengan sistem secara efektif.

### 2.1.3 Protokol IoT

Protokol Internet of Things (IoT) menurut (Particle Industries, 2023) adalah kumpulan aturan dan standar komunikasi yang digunakan untuk menghubungkan dan mengontrol perangkat IoT. Protokol ini memungkinkan perangkat IoT untuk berkomunikasi dan bertukar data secara efisien dan aman. Berikut penjelasan dari komponen-komponen yang ada pada protocol IoT

#### a. *IoT Network protocols*

*IoT network protocols* atau protokol jaringan dalam IoT berfungsi untuk menghubungkan perangkat melalui jaringan. Serta merupakan protokol komunikasi yang digunakan melalui Internet.

*IoT network protocols* menggunakan komunikasi data *end-to-end* dalam lingkup jaringan. Berikut ini adalah berbagai *IoT network protocols*:

- Wi-Fi

Wi-Fi adalah protokol nirkabel yang paling umum digunakan untuk menghubungkan perangkat IoT ke jaringan lokal. Wi-Fi menyediakan koneksi dengan kecepatan tinggi dan jangkauan yang luas,

membuatnya cocok untuk aplikasi rumah pintar dan lingkungan perkantoran.

- **LTE CAT 1**

LTE (Long-Term Evolution) CAT 1 adalah protokol jaringan seluler yang dirancang untuk menghubungkan perangkat IoT ke jaringan 4G. Protokol ini memberikan kecepatan data yang tinggi dan stabil, serta mendukung komunikasi dua arah.

- **LTE CAT M1**

LTE CAT M1, juga dikenal sebagai LTE-M, adalah protokol jaringan seluler yang dikembangkan khusus untuk aplikasi IoT yang membutuhkan konsumsi daya yang rendah dan jangkauan yang luas. Protokol ini cocok untuk penggunaan dalam industri, pertanian, dan sektor lain yang memerlukan konektivitas jangka panjang dan hemat energi.

- **NB-IoT**

NB-IoT (Narrowband Internet of Things) adalah protokol jaringan seluler yang dioptimalkan untuk konektivitas IoT dengan konsumsi daya yang sangat rendah dan jangkauan yang luas. Protokol ini cocok untuk aplikasi yang membutuhkan koneksi yang andal dan hemat energi, seperti meteran pintar dan sensor jarak jauh.

- **LoRaWan( Long Range Wide Area Network)**

LoRaWAN adalah protokol *media access control* (MAC) yang didesain sebagai jaringan publik skala besar dengan satu operator. LoRaWan menghubungkan apapun yang beroperasi melalui baterai secara nirkabel menuju Internet baik secara *private network* maupun *global network*.

- **Zigbee**

ZigBee merupakan protokol IoT (Internet of Things) yang dikhususkan kepada pelaku industri. Oleh karena itu, ZigBee digunakan untuk transfer data kecepatan rendah dalam jarak pendek.

Protokol ini sangat ideal untuk kegiatan industri karena mampu beroperasi pada frekuensi 2.4 Ghz, dimana data yang ditransfer melalui trafik kecil dianatara rumah atau bangunan.

- Bluetooth

Bluetooth adalah salah satu teknologi nirkabel jarak pendek yang paling banyak dipakai dalam IoT. Protokol ini sangat aman kegiatan komunikasi dan bagus pada transmisi jarak pendek, menghasilkan daya yang rendah, biaya rendah, dan nirkabel antara perangkat elektronik.

Protokol bluetooth memiliki versi yang baru diperkenalkan yakni BLE (Bluetooth Low Energy). BLE ini memberikan jangkauan kombinasi dari bluetooth konvesional dengan supremasi konsumsi daya yang lebih rendah. BLE ini cocok digunakan untuk transfser data yang yang kecil.

Dalam penelitian ini, Protokol jaringan yang digunakan adalah WiFi yang sebagai komunikasi anatara sensoring ke *firebase* dan *firebase* ke aplikasi android untuk memberikan data listrik dengan bantuan ESP8266 sebagai *microcontroller*.

#### b. IoT Data Protocols

IoT Data Protocols adalah protokol-protokol yang digunakan untuk pertukaran data antara perangkat IoT dan sistem atau platform yang terhubung. Berikut adalah penjelasan singkat tentang beberapa protokol data IoT yang umum digunakan:

- AMQP (Advanced Message Queuing Protocol)

AMQP adalah protokol yang dirancang untuk pertukaran pesan yang andal dan efisien di dalam lingkungan distribusi. Protokol ini mendukung pengiriman pesan yang terjamin, sinkronisasi, dan manajemen antrian yang kuat, membuatnya cocok untuk aplikasi IoT yang membutuhkan komunikasi yang handal dan tahan terhadap kehilangan data.

- **MQTT (Message Queuing Telemetry Transport)**

MQTT adalah protokol berbasis pub-sub yang dirancang untuk komunikasi ringan dan efisien antara perangkat IoT dan broker MQTT. Protokol ini memungkinkan perangkat untuk berlangganan topik tertentu dan menerima pesan secara real-time, serta mengirim pesan dengan konsumsi daya yang rendah. MQTT cocok untuk skenario IoT yang membutuhkan komunikasi dalam skala besar dan dengan sumber daya terbatas.

- **HTTP (Hypertext Transfer Protocol)**

HTTP adalah protokol yang umum digunakan dalam komunikasi web dan juga digunakan dalam konteks IoT. Protokol ini memungkinkan pertukaran data melalui permintaan (request) dan tanggapan (response) antara perangkat IoT dan server. HTTP mudah diimplementasikan dan kompatibel dengan infrastruktur internet yang ada, namun cenderung memerlukan overhead komunikasi yang lebih besar.

- **CoAP (Constrained Application Protocol)**

CoAP adalah protokol yang dikembangkan khusus untuk perangkat IoT dengan sumber daya terbatas, seperti perangkat yang terbatas pada jaringan yang memiliki bandwidth rendah dan konsumsi energi yang terbatas. Protokol ini menggunakan pendekatan RESTful (Representational State Transfer) untuk

komunikasi, dan mendukung pengiriman pesan yang ringan dan efisien.

- DDS (Data Distribution Service)

DDS adalah protokol yang digunakan untuk pertukaran data secara real-time antara sistem dan perangkat IoT yang berada dalam lingkungan distribusi. Protokol ini dirancang untuk keandalan dan latensi rendah, serta mendukung komunikasi yang terdistribusi dan heterogen.

- LwM2M (Lightweight Machine to Machine)

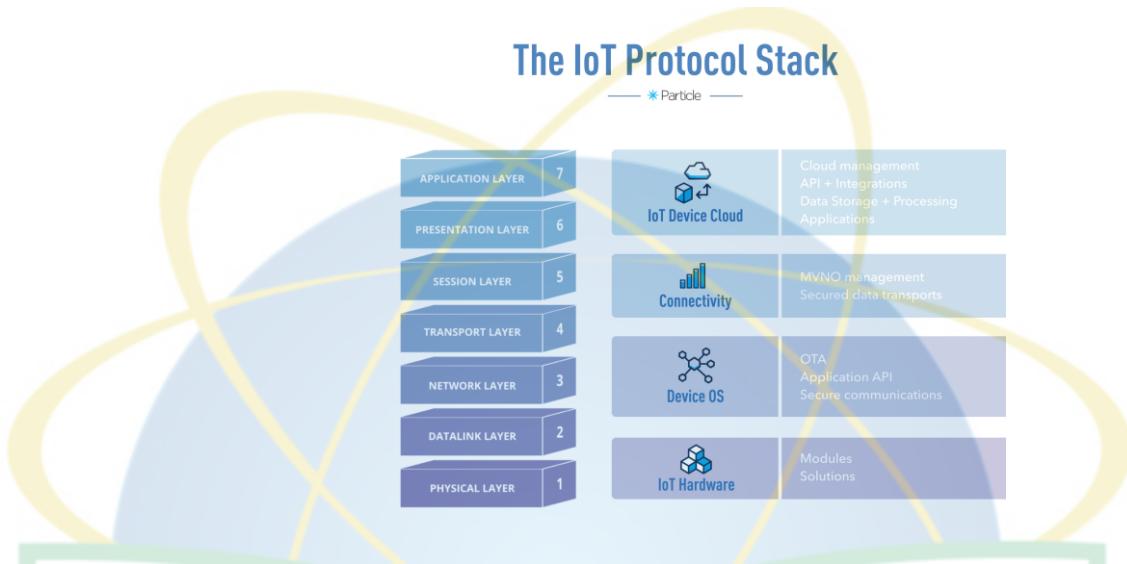
LwM2M adalah protokol yang dirancang untuk manajemen perangkat dan pertukaran data pada perangkat IoT dengan sumber daya terbatas. Protokol ini menyediakan mekanisme untuk pengaturan, pemantauan, dan pengumpulan data dari perangkat IoT, serta mendukung pembaruan perangkat lunak secara over-the-air.

Dalam penelitian ini data protokol IoT yang digunakan ialah HTTP dengan fungsi diantaranya Komunikasi antara Perangkat IoT dan *Firebase*, protokol HTTP dapat digunakan untuk mengirimkan data dari perangkat IoT (seperti ESP8266) ke *Firebase*. Data monitoring daya listrik yang dikumpulkan oleh perangkat akan dikirim melalui protokol HTTP dalam bentuk permintaan (*request*) *POST* ke *Firebase REST API*. Data ini akan disimpan di database *Firebase* untuk diolah dan dianalisis lebih lanjut.

Komunikasi antara Aplikasi Android dan *Firebase*: Protokol HTTP juga digunakan dalam komunikasi antara aplikasi Android dengan *Firebase*. Aplikasi Android akan mengirimkan permintaan HTTP (misalnya *GET* atau *POST*) ke *Firebase REST API* untuk mengakses dan memanipulasi data monitoring daya listrik. Permintaan ini dapat

mencakup operasi seperti mendapatkan data, menyimpan data, atau menghapus data dari database *Firebase*.

### c. IoT Protocol Stack



Gambar 2. 2 Protokol IoT Stack (sumber : particle)

IoT Protocol Stack adalah hierarki protokol yang digunakan untuk memfasilitasi komunikasi antara perangkat IoT dan infrastruktur jaringan. Protokol stack ini terdiri dari beberapa lapisan yang menyediakan fungsi dan layanan yang berbeda untuk memungkinkan pertukaran data yang aman dan efisien dalam ekosistem IoT. Berikut adalah penjelasan lebih lanjut tentang masing-masing lapisan dalam IoT Protocol Stack:

- **Physical Layer:** Lapisan fisik adalah lapisan terbawah dalam protokol stack dan berhubungan langsung dengan perangkat keras fisik yang digunakan dalam komunikasi IoT. Ini termasuk media transmisi seperti kabel, nirkabel, atau jaringan satelit. Lapisan fisik menentukan cara perangkat IoT terhubung dan berkomunikasi dengan jaringan, seperti protokol Wi-Fi, Bluetooth, Zigbee, dan sebagainya.

- Data Link Layer: Lapisan ini bertanggung jawab atas pengelolaan dan pengaturan aliran data antara perangkat IoT dalam jaringan. Lapisan ini memastikan pengiriman data yang handal melalui protokol seperti Ethernet, Wi-Fi (802.11), Zigbee (802.15.4), dan lainnya. Lapisan ini juga menangani deteksi dan pengendalian kesalahan dalam pengiriman data.
- Network Layer: Lapisan jaringan menangani routing data dan pengiriman paket melalui jaringan. Protokol yang digunakan pada lapisan ini termasuk IP (Internet Protocol) dan protokol jaringan lainnya seperti IPv4, IPv6, dan lainnya. Lapisan ini memastikan pengiriman paket data yang efisien antara perangkat IoT dan infrastruktur jaringan.
- Transport Layer: Lapisan transportasi bertanggung jawab atas pengaturan koneksi dan pengiriman data yang handal antara perangkat IoT. Protokol yang digunakan pada lapisan ini termasuk TCP (Transmission Control Protocol) dan UDP (User Datagram Protocol). Lapisan ini memastikan kehandalan pengiriman data dan pengendalian aliran data antara perangkat.
- Session Layer: Lapisan sesi bertanggung jawab atas pembentukan, pemeliharaan, dan penghentian sesi komunikasi antara perangkat IoT. Lapisan ini mengatur inisialisasi dan penutupan koneksi serta mengelola pertukaran data antara perangkat.
- Presentation Layer: Lapisan presentasi menangani format data dan enkripsi dalam komunikasi antara perangkat IoT. Lapisan ini mengubah format data yang dikirim dari perangkat IoT ke format yang dapat dipahami oleh penerima dan juga menyediakan enkripsi data untuk keamanan.
- Application Layer: Lapisan aplikasi adalah lapisan teratas dalam protokol stack dan berhubungan langsung dengan aplikasi dan

layanan yang dijalankan pada perangkat IoT. Protokol yang digunakan pada lapisan ini tergantung pada jenis aplikasi yang diimplementasikan, seperti MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), HTTP (Hypertext Transfer Protocol), dan banyak lainnya. Lapisan ini menyediakan antarmuka bagi pengguna untuk berinteraksi dengan perangkat IoT.

Dalam penelitian ini, protokol stack IoT yang digunakan berbeda-beda tergantung pada rancangan dan implementasi sistem tersebut. Namun, berikut adalah penjelasan umum tentang *layer-layer* protokol yang mungkin terlibat dalam implementasi tersebut:

Tabel 2. 3 Layer Protokol IoT peneliti

Layer	Penjelasan
Physical Layer	Lapisan fisik akan melibatkan perangkat keras dan media transmisi yang digunakan untuk menghubungkan perangkat IoT dengan jaringan. Penggunaan Wi-Fi sebagai media transmisi nirkabel untuk menghubungkan perangkat IoT dengan jaringan internet.
Data Link Layer	Lapisan data link akan mengatur pengelolaan dan pengaturan aliran data antara perangkat IoT dan infrastruktur jaringan. Wi-Fi dapat digunakan untuk pengiriman data yang handal antara perangkat IoT dan jaringan.
Network Layer	Lapisan jaringan akan menangani pengiriman paket data melalui jaringan.

	Internet Protokol dengan IP 192.168.1.12 digunakan untuk mengarahkan dan mengirimkan paket data antara perangkat IoT dan infrastruktur jaringan.
Transport Layer	Lapisan transportasi akan mengatur pengiriman data yang handal antara perangkat IoT dan infrastruktur jaringan. Protokol seperti TCP (Transmission Control Protocol) mungkin digunakan untuk memastikan kehandalan dan pengendalian aliran data.
Session Layer	Lapisan sesi akan mengatur pembentukan, pemeliharaan, dan penghentian sesi komunikasi antara perangkat IoT dan infrastruktur jaringan. Lapisan ini bertanggung jawab atas inisialisasi dan penutupan koneksi serta pengelolaan pertukaran data antara perangkat IoT.
Presentation Layer	Lapisan presentasi akan mengelola format data dan enkripsi dalam komunikasi antara perangkat IoT dan infrastruktur jaringan. Lapisan ini dapat mengubah format data yang dikirim dari perangkat IoT ke format yang dapat dipahami oleh penerima dan menyediakan enkripsi data untuk menjaga keamanan.
Application Layer	Lapisan aplikasi akan berhubungan langsung dengan aplikasi dan layanan yang dijalankan

	pada perangkat IoT. Protokol HTTP digunakan untuk mengirimkan data monitoring daya listrik antara perangkat IoT dan aplikasi yang berjalan di platform <i>Firebase</i> berbasis Android.
--	--

#### 2.1.4 Komunikasi IoT

*IoT (Internet of Things) Communication* (guduruuaishwarya09, 2022) adalah proses pertukaran data dan informasi antara perangkat yang terhubung dalam ekosistem IoT. Komunikasi dalam IoT memungkinkan perangkat untuk saling berinteraksi, berbagi data, dan melakukan berbagai tugas dalam jaringan IoT.

##### a. Tipe Komunikasi

- Komunikasi M2M (Machine-to-Machine)

Jenis komunikasi ini melibatkan pertukaran data antara perangkat IoT tanpa melibatkan intervensi manusia. Perangkat IoT saling berkomunikasi untuk bertukar informasi atau melakukan tindakan tertentu.

Komunikasi M2M memungkinkan perangkat IoT saling berinteraksi dan berkolaborasi secara mandiri, tanpa perlu campur tangan manusia. Hal ini memungkinkan adanya otomatisasi, pemantauan real-time, dan pengambilan keputusan yang cepat dalam lingkungan IoT.

- Komunikasi M2H (Machine-to-Human)

Jenis komunikasi ini melibatkan pertukaran data antara perangkat IoT dan manusia. Contohnya adalah perangkat kesehatan yang mengirimkan data medis ke smartphone pengguna.

Komunikasi M2H memungkinkan pengguna manusia menerima informasi penting dari perangkat IoT dan berinteraksi dengan mereka. Pengguna dapat menerima notifikasi, mengakses data secara real-time, atau memberikan instruksi kepada perangkat IoT melalui antarmuka pengguna yang disediakan, seperti aplikasi mobile atau dashboard web. Hal ini memungkinkan pengguna untuk memiliki kontrol dan interaksi yang lebih baik dengan lingkungan IoT.

- Komunikasi H2M (Human-to-Machine)

Jenis komunikasi ini melibatkan interaksi manusia dengan perangkat IoT. Pengguna dapat memberikan instruksi atau meminta informasi dari perangkat IoT melalui antarmuka yang sesuai.

Dalam H2M Communication, manusia berperan sebagai pengguna yang memberikan instruksi atau menerima informasi dari perangkat IoT. Komunikasi dapat dilakukan secara langsung antara pengguna dan perangkat, atau melalui jaringan internet jika perangkat terhubung secara online.

- Komunikasi H2H (Human-to-Human)

Jenis komunikasi ini melibatkan interaksi antara manusia melalui perangkat IoT. Contohnya adalah aplikasi obrolan (chat) yang memungkinkan pengguna berkomunikasi satu sama lain melalui perangkat IoT yang terhubung.

Komunikasi H2H melalui perangkat IoT memungkinkan manusia berkomunikasi secara efisien dan cepat dengan bantuan teknologi. Perangkat IoT berperan dalam menyediakan koneksi dan pengiriman pesan antara pengguna manusia. Ini memungkinkan kolaborasi, interaksi sosial, dan pertukaran informasi antara individu yang terhubung dalam ekosistem IoT.

Dalam penelitian yang dilakukan memiliki beberapa tipe komunikasi diantaranya Human-Machine, Machine-Human dan Machine-Machine, berikut yang peneiliti lakukan :

Tabel 2. 4 Tipe Komunikasi Peneliti

Tipe Komunikasi	Penjelasan
M2M ( <i>Machine-to-Machine</i> )	<p>Dalam penelitian ini, komunikasi M2M terjadi antara perangkat-perangkat yang terlibat dalam sistem monitoring daya listrik. Contohnya, perangkat pemantauan daya listrik seperti sensor daya akan berkomunikasi dengan server <i>Firebase</i>. Proses komunikasi M2M ini memungkinkan perangkat pemantauan untuk mengirimkan data daya listrik yang dikumpulkan ke server <i>Firebase</i> secara otomatis. Data tersebut dapat mencakup informasi seperti konsumsi daya, tegangan, jumlah perangkat dan arus.</p> <p>sistem monitoring daya listrik dapat mengumpulkan data secara real-time dan mengirimkannya ke server <i>Firebase</i> untuk diproses lebih lanjut. Informasi yang dikumpulkan ini dapat digunakan untuk melakukan analisis, pemantauan, dan pengambilan keputusan terkait penggunaan daya listrik.</p>
M2H ( <i>Machine-to-Human</i> )	<p>Dalam penelitian ini, komunikasi M2H terjadi antara perangkat pemantauan daya listrik yang terhubung dengan sistem monitoring dan pengguna manusia.</p> <p>Perangkat pemantauan daya listrik, yaitu aplikasi berbasis Android, akan memberikan informasi kepada</p>

	<p>pengguna tentang konsumsi daya listrik secara realtime, tabel data listrik dan hasil analisis yang dilakukan menggunakan algoritma <i>Fuzzy Logic Sugeno</i>. Pengguna dapat menerima informasi ini melalui antarmuka pengguna pada aplikasi yang dikembangkan.</p>
H2M ( <i>Human-to-Machine</i> )	<p>Dalam penelitian ini, komunikasi H2M terjadi ketika pengguna berinteraksi dengan aplikasi berbasis Android yang digunakan untuk memantau data listrik dan memilih data yang akan dianalisis.</p> <p>Pengguna manusia dapat menggunakan antarmuka pengguna pada aplikasi berbasis Android untuk melakukan berbagai tindakan, seperti memantau konsumsi listrik, dan mengatur tanggal untuk mendapatkan data analisis sesuai keinginan pengguna listrik.</p>

Dalam penelitian ini, interaksi manusia lebih fokus pada penggunaan aplikasi yang menyediakan tampilan informasi dan kontrol yang diperlukan, seperti melihat data daya listrik secara realtime, mengatur tanggal data, dan hasil *Fuzzy* dari jarak jauh. Oleh karena itu, komunikasi H2H tidak diperlukan dalam implementasi ini, karena fokus utama adalah pada komunikasi antara perangkat IoT dan server *Firebase* yang digunakan sebagai platform backend untuk menyimpan dan memproses data, dan Komunikasi antara manusia digantikan dengan antarmuka aplikasi Android yang memungkinkan pengguna untuk memantau dan mengontrol sistem secara langsung melalui perangkat *mobile* dengan jarak jauh

## b. Model Komunikasi

Model komunikasi dalam konteks IoT (Airtel, 2022) mengacu pada pola atau metode yang digunakan untuk memfasilitasi pertukaran data dan informasi antara perangkat IoT. Berikut adalah beberapa model komunikasi yang umum digunakan dalam IoT:

- Komunikasi *Bidirectional* (Dua Arah)

Model komunikasi ini melibatkan pertukaran data dalam dua arah antara perangkat IoT dan sistem backend. Artinya, tidak hanya perangkat yang mengirim data, tetapi juga menerima data dan meresponsnya. Hal ini memungkinkan adanya interaksi dan kolaborasi antara perangkat yang terhubung dalam sistem IoT.

Contohnya adalah perangkat smart home yang dapat menerima instruksi dari pengguna dan mengirimkan laporan status kembali ke sistem backend.

- Komunikasi *Publikasi-Subscribe*

Model ini berdasarkan pada konsep publikasi dan langganan. Perangkat IoT yang berfungsi sebagai penerbit (publisher) mengirimkan data ke topik tertentu, dan perangkat IoT yang berfungsi sebagai pelanggan (subscriber) mendaftar untuk menerima data dari topik tersebut.

Contohnya adalah sistem manajemen lalu lintas yang mengirimkan informasi lalu lintas ke perangkat seluler pengguna yang berlangganan.

- Komunikasi *Command-Response*

Komunikasi *Command-Response* adalah model komunikasi yang digunakan dalam sistem IoT di mana perangkat atau entitas mengirimkan perintah (*command*) kepada perangkat lainnya dan menerima tanggapan

(*response*) sebagai balasan atas perintah yang dikirim. Model ini mirip dengan komunikasi antara pengirim dan penerima dalam bentuk pertanyaan dan jawaban.

Contohnya adalah perangkat pengukur suhu yang menerima perintah untuk mengirimkan data suhu saat ini dan merespons dengan mengirimkan data tersebut.

- Komunikasi *Event-Triggered*

Komunikasi *Event-Triggered* adalah model komunikasi yang terjadi dalam sistem IoT ketika peristiwa atau event tertentu terjadi pada perangkat atau lingkungan, dan informasi tentang peristiwa tersebut dikirimkan kepada entitas lain untuk memberikan respons atau tindakan yang sesuai. Model ini didasarkan pada konsep pengiriman pesan berdasarkan adanya perubahan atau trigger dari suatu kejadian.

Contohnya adalah sistem keamanan rumah yang mengirimkan notifikasi ke ponsel pengguna ketika gerakan terdeteksi oleh sensor gerak.

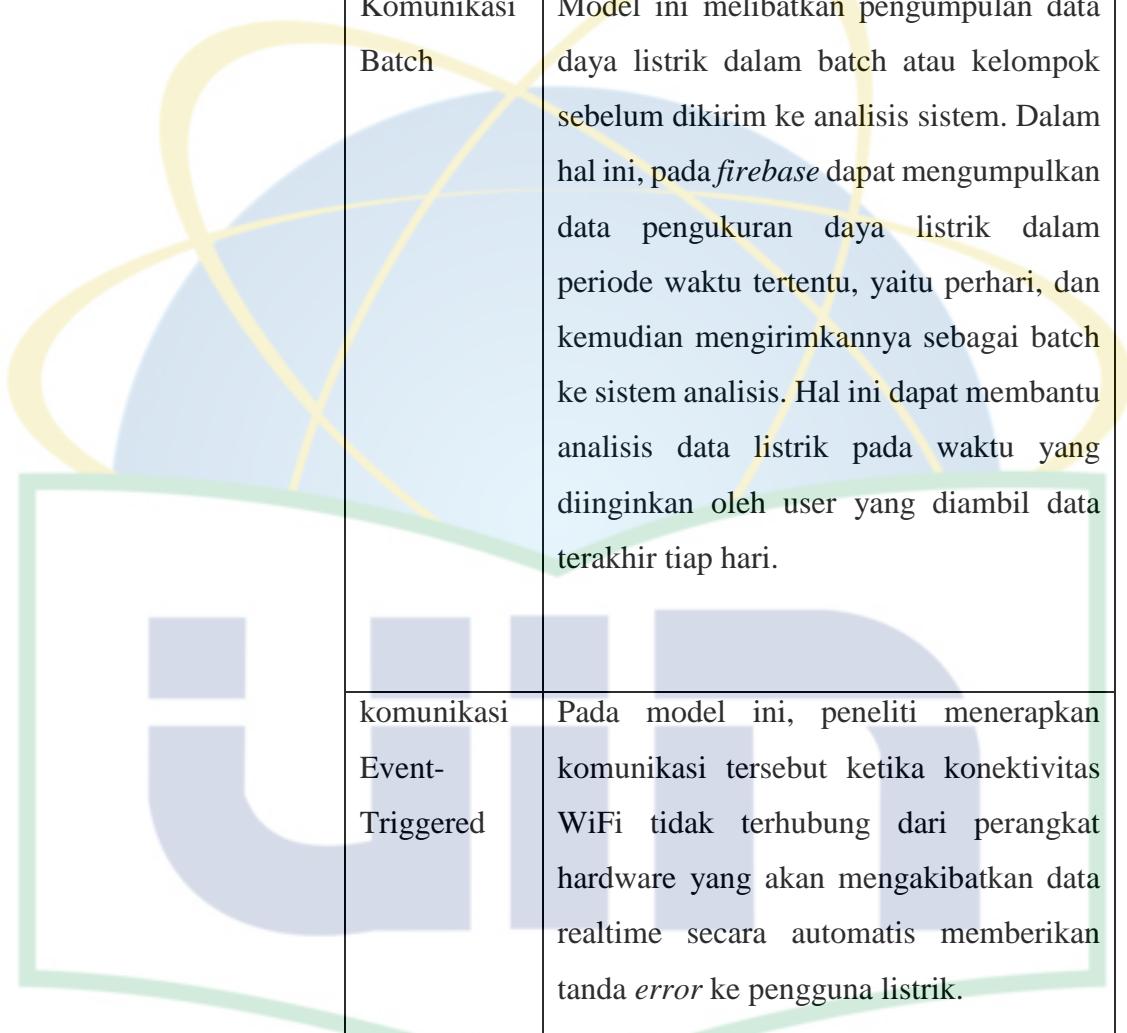
- Komunikasi *Batch*

Model ini melibatkan pengiriman data dalam jumlah besar secara periodik. Contohnya adalah perangkat pengumpul data cuaca yang mengirimkan data cuaca harian ke sistem backend setiap jam.

Dalam penelitian yang dilakukan memiliki beberapa model komunikasi sebagai berikut:

Tabel 2.5 Model Komunikasi peneliti

Model Komunikasi	Penjelasan
Komunikasi <i>Bidirectional</i> (Dua Arah)	Model ini memungkinkan pertukaran data antara server <i>Firebase</i> dan <i>mobile</i> . Pengguna listrik melakukan permintaan tanggal data yang ingin diketahui melalui <i>mobile</i> , lalu data listrik pengguna akan dikirimkan dari <i>firebase</i> melalui proses pengambilan <i>req</i> data.
Komunikasi Publikasi- <i>Subscribe</i>	Model ini melibatkan penggunaan mekanisme publikasi dan langganan. Perangkat IoT dapat mempublikasikan data daya listrik ke topik tertentu pada server <i>Firebase</i> . Pada saat yang sama, aplikasi Android yang berlangganan topik tersebut akan menerima pembaruan data secara otomatis. Pada komunikasi ini data <i>realtime</i> diproses.
Komunikasi <i>Command-Response</i> :	Model ini melibatkan pengiriman perintah dari aplikasi Android ke sistem. Aplikasi Android dapat mengirimkan perintah kontrol, seperti untuk <i>set</i> tanggal yang pengguna ingin ketahui data listrik pengaturan sistem monitor daya listrik pada <i>mobile</i> dan akan merespons dengan memberikan analisis data berupa grafik



	dan tabel sesuai tanggal yang diinginkan data tersebut berupa jumlah KwH,biaya listrik,jumlah perangkat dan hasil <i>output Fuzzy</i> dalam perhari/ <i>req</i>
Komunikasi Batch	Model ini melibatkan pengumpulan data daya listrik dalam batch atau kelompok sebelum dikirim ke analisis sistem. Dalam hal ini, pada <i>firebase</i> dapat mengumpulkan data pengukuran daya listrik dalam periode waktu tertentu, yaitu perhari, dan kemudian mengirimkannya sebagai batch ke sistem analisis. Hal ini dapat membantu analisis data listrik pada waktu yang diinginkan oleh user yang diambil data terakhir tiap hari.
komunikasi Event-Triggered	Pada model ini, peneliti menerapkan komunikasi tersebut ketika konektivitas WiFi tidak terhubung dari perangkat hardware yang akan mengakibatkan data realtime secara automatis memberikan tanda <i>error</i> ke pengguna listrik.

### c. Model Konektivitas

Model Konektivitas dalam konteks Internet of Things (IoT) mengacu pada cara perangkat IoT terhubung satu sama lain dan dengan infrastruktur jaringan untuk berbagi data dan berkomunikasi. Model Konektivitas ini menentukan cara komunikasi antara

perangkat IoT, baik secara langsung maupun melalui perantara seperti gateway atau cloud. ada 4 model konektivitas menurut (Chris, 2016) sebagai berikut :

- Device-to-Cloud: Model ini melibatkan komunikasi antara perangkat IoT dan cloud computing. Perangkat IoT mengirimkan data dan informasi ke cloud untuk penyimpanan, analisis, dan pengolahan lebih lanjut. Hal ini memungkinkan akses data dari mana saja dan memungkinkan pemantauan dan pengontrolan perangkat secara jarak jauh.
- Device-to-Gateway: Dalam model ini, perangkat IoT berkomunikasi dengan gateway atau hub yang bertindak sebagai titik pusat untuk mengumpulkan, mengolah, dan mengirimkan data ke cloud atau sistem lainnya. Gateway bertindak sebagai perantara antara perangkat IoT dan infrastruktur jaringan yang lebih besar.
- Device-to-Device: Model ini melibatkan komunikasi langsung antara perangkat IoT tanpa melibatkan cloud atau gateway. Perangkat IoT dapat berinteraksi satu sama lain, berbagi data, atau saling mengontrol berdasarkan aturan yang ditetapkan.
- Back-End Data-Sharing: Model ini melibatkan berbagi data antara berbagai perangkat IoT dan platform atau aplikasi lainnya yang terhubung ke backend. Data yang dikumpulkan oleh perangkat IoT dapat dibagikan dengan berbagai aplikasi atau layanan untuk analisis, visualisasi, atau tindakan selanjutnya.

Tabel 2. 6 Model Konektivitas peniliti

Model Konektivitas	
Device-to-Cloud	Model ini melibatkan komunikasi antara perangkat IoT yang digunakan untuk monitoring daya listrik dengan layanan cloud seperti <i>Firebase</i> . Data yang dikumpulkan oleh perangkat IoT dikirim ke cloud untuk penyimpanan, analisis, dan pengelolaan. Pengguna dapat mengakses data tersebut melalui aplikasi berbasis Android yang terhubung ke <i>Firebase</i> .
Device-to-Gateway:	Model ini melibatkan komunikasi antara perangkat IoT dan gateway, yang bertindak sebagai perantara antara perangkat IoT dan cloud. Gateway berfungsi mengumpulkan dan mengirimkan data dari perangkat IoT ke cloud. Dalam konteks penelitian ini, gateway digunakan untuk menghubungkan perangkat IoT yang digunakan untuk monitoring daya listrik ke layanan cloud <i>Firebase</i> .
Back-End Data-Sharing	Model ini melibatkan berbagi data antara sistem yang berjalan di back-end, seperti server atau database. Dalam penelitian ini, data yang dikumpulkan oleh perangkat IoT dan disimpan di cloud <i>Firebase</i> dapat dibagikan atau diintegrasikan dengan sistem lain yang terhubung ke back-end. Hal ini data daya listrik yang terkumpul dapat digunakan untuk tujuan analisis
Device-to-Device	Model ini melibatkan perangkat dengan perangkat yaitu ESP dan PZEM dapat berkomunikasi secara

	langsung antara satu sama lain dalam komunikasi Device-to-Device untuk mengirim dan menerima data pengukuran daya listrik.
--	--

Pada penjelasan yang dibahas diatas terdapat hal penting yang peneliti lakukan untuk mendapatkan sistem IoT yang baik, Internet of Things (IoT) telah menjadi landasan utama dalam pengembangan sistem monitoring daya listrik. IoT memungkinkan integrasi perangkat elektronik dengan jaringan komunikasi untuk memantau perangkat secara jarak jauh.

Layer IoT terdiri dari Perception Layer, Network Layer, Processing Layer, dan Application Layer. Perception Layer menggunakan sensor-sensor untuk mengumpulkan data daya listrik. Network Layer mengatur jaringan dan komunikasi antara perangkat IoT. Processing Layer melibatkan pemrosesan data menggunakan algoritma *Fuzzy Logic Sugeno*. Application Layer mencakup aplikasi Android yang digunakan untuk memantau dan mengontrol sistem monitoring daya listrik.

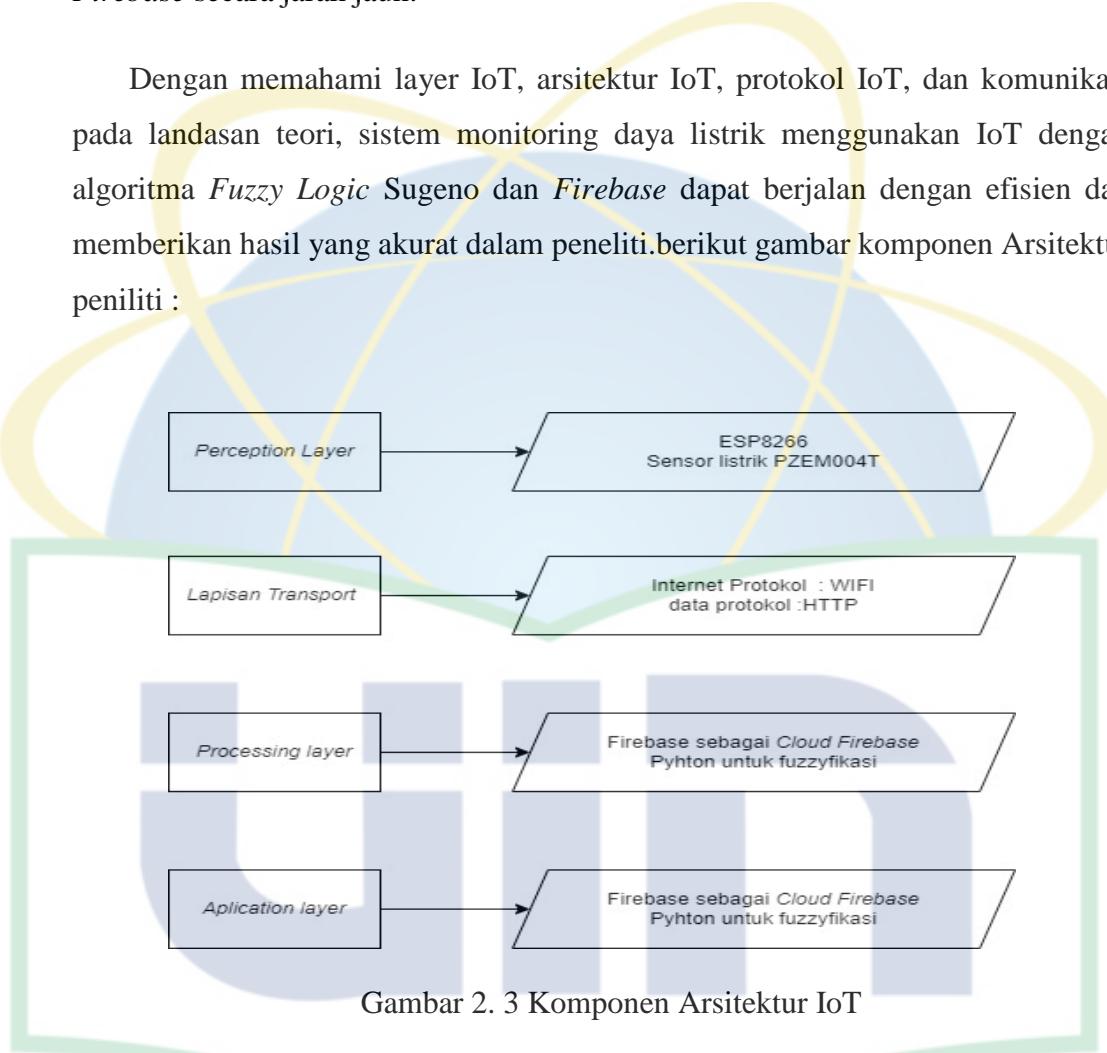
Arsitektur IoT yang digunakan dalam skripsi ini melibatkan perangkat IoT, platform *Firebase*, dan aplikasi Android. Perangkat IoT berperan sebagai sensor daya listrik yaitu PZEM 004T dan ESP8266 yang mengumpulkan data dan mengirimkannya ke platform *Firebase*. *Firebase* digunakan sebagai platform cloud untuk menyimpan data, melakukan analisis, dan menyediakan antarmuka untuk aplikasi Android. Aplikasi Android digunakan oleh pengguna untuk memantau sistem monitoring daya listrik.

Pada penelitian ini Protokol IoT yang digunakan adalah HTTP (Hypertext Transfer Protocol). HTTP digunakan untuk pertukaran data antara perangkat IoT dan *Firebase* melalui protokol HTTP. *Firebase* API juga digunakan sebagai antarmuka untuk berkomunikasi dengan platform *Firebase*.

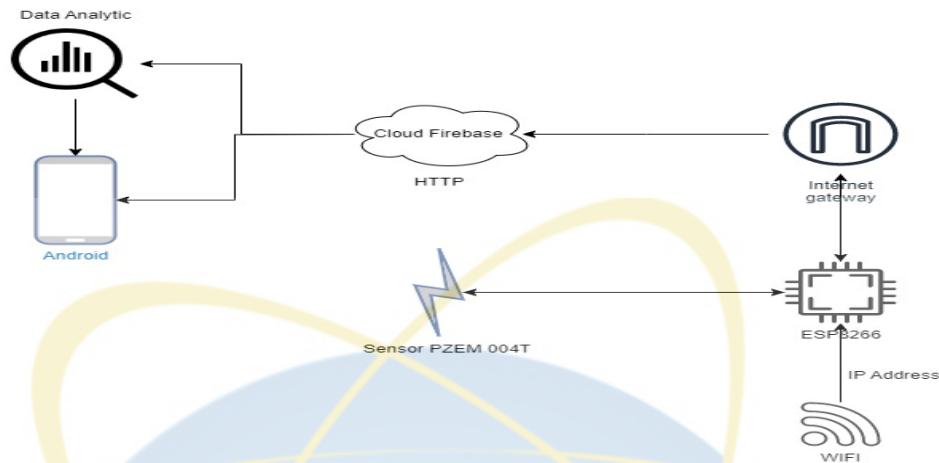
Komunikasi dalam skripsi ini melibatkan pertukaran data dan informasi antara perangkat IoT, platform *Firebase*, dan pengguna melalui aplikasi Android. Data

daya listrik yang dikumpulkan oleh sensor dikirim ke *Firebase* untuk penyimpanan dan analisis menggunakan algoritma *Fuzzy Logic Sugeno*. Pengguna dapat mengakses dan memonitor data melalui aplikasi Android yang terhubung ke *Firebase* secara jarak jauh.

Dengan memahami layer IoT, arsitektur IoT, protokol IoT, dan komunikasi pada landasan teori, sistem monitoring daya listrik menggunakan IoT dengan algoritma *Fuzzy Logic Sugeno* dan *Firebase* dapat berjalan dengan efisien dan memberikan hasil yang akurat dalam penelitian. Berikut gambar komponen Arsitektur peneliti :



Dan berikut Flowchart IoT pada penelitian ini :



Gambar 2. 4 Flowchart IoT peneliti

### 2.3 Karakteristik Sumber Listrik

Listrik adalah suatu energi, bahkan energi listrik begitu memegang peranan penting bagi kehidupan kita. Listrik adalah suatu muatan yang terdiri dari muatan positif dan muatan negatif. Arus listrik merupakan muatan listrik yang bergerak dari tempat yang berpotensial tinggi ke tempat berpotensial rendah, melewati suatu penghantar listrik (UNDIP, 2021). Terdapat besaran besaran atau komponen dalam listrik menurut (UPI, 2022) berikut komponennya :

#### 2.2.1 Tegangan listrik

Adalah potensi atau tekanan listrik dari suatu sumber listrik besar tegangan listriknya ditentukan oleh perbedaan potensi antara satu titik dengan titik lainnya. Satuan tegangan listrik adalah Volt disingkat dengan huruf V dan mempunya simbol huruf E. Alat yang digunakan untuk mengukur tegangan listrik adalah Volt Meter.

#### 2.2.2 Arus Listrik

Adalah banyaknya elektron bebas yang mengalir dalam suatu penghantar dan merupakan lingkaran tertutup persatuan waktu. Satuan arus listrik adalah

Ampere disingkat dengan huruf A dan mempunyai simbol huruf I. Alat yang digunakan untuk mengukur arus listrik adalah Ampere Meter

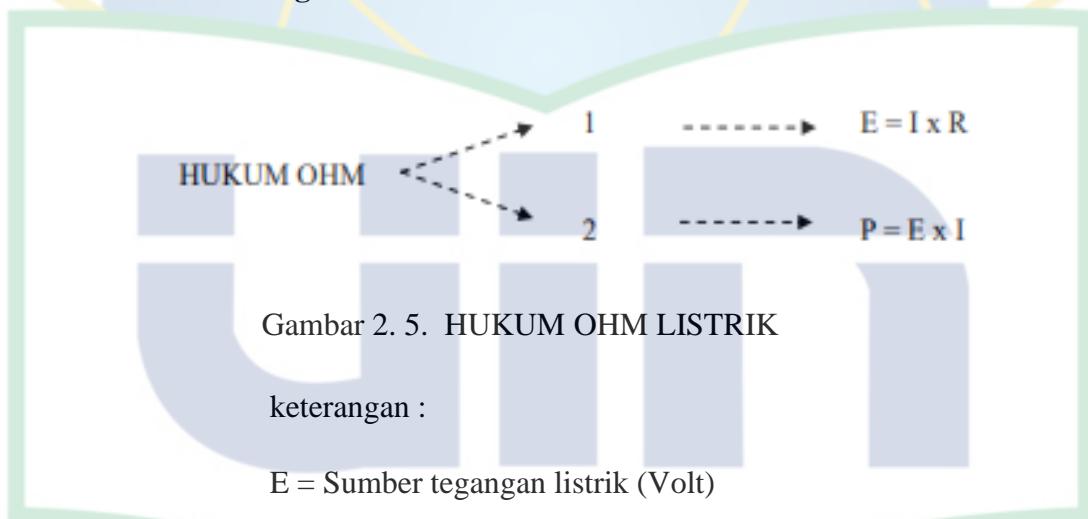
### 2.2.3 Hambatan Listrik

Adalah rintangan (Resistance) yang dihadapi oleh aliran listrik pada suatu pengantar satuan hambatan listrik adalah Ohm disingkat dengan simbol Omega ( $\Omega$ ) dan mempunyai simbol huruf R alat yang digunakan untuk mengukur hambatan listrik adalah Ohm Meter.

### 2.2.4 Daya Listrik

Adalah kekuatan yang dikandung dalam aliran arus dan tegangan listrik melalui hambatan dengan besaran tertentu satuan daya listrik adalah Watt disingkat dengan huruf W dan mempunyai simbol P.

### 2.2.5 Hubungan Antara Besaran-Besaran Listrik



### 2.4 Tarif listrik per KwH

KWh adalah akronim dari kilowatt hour atau kilowatt jam, satuan ini menunjukkan pemakaian daya sebesar 100 watt setiap jam nya. KWh juga biasa

digunakan untuk mengukur tagihan listrik rumah, semakin tinggi angka kWh di rumah maka semakin besar pula biaya yang harus dibayarkan(Arif Sandira, 2023).

Cara menghitung penggunaan kWh cukup mudah menggunakan rumus  $kWh = (\text{watt} \times \text{jam}) : 1000$ . Berikut data (Lamudi, 2023) Tarif Listrik per kWh untuk Pelanggan Rumah Tangga Tahun 2023 Tarif listrik :

Golongan	Daya Listrik	Tarif Listrik per kWh
R-1/TR	900 VA	Rp 1.352 per kWh
R-1/TR	1.300 VA	Rp 1.444,70 per kWh
R-1/ TR	2.200 VA	Rp 1.444,70 per kWh
R-2/ TR	3.500-5.500 VA	Rp 1.669,53 per kWh
R-3/ TR	6.600 VA	Rp 1.669,53 per kWh

Gambar 2. 6 Tarif listrik per KwH

Contoh perhitungan untuk mendapatkan KwH dan Harga listrik sebagai berikut (ONLINE, 2021):

1. 4 buah lampu 10 W menyala 10 jam,
2. sebuah setrika 300 W dipakai 1 jam,
3. sebuah TV 100 W hidup selama 6 jam,

Jika harga per kWh Rp1.444,70 karena menggunakan Daya listrik 1.300VA .maka daya listrik yang harus dibayar selama 1 bulan (misal bulan Maret) adalah :

- a. Menghitung energi listrik yang dihabiskan setiap hari yakni:

$$W = 4(10 \text{ W})(10 \text{ jam}) + (300 \text{ W})(1 \text{ jam}) + (100 \text{ W})(6 \text{ jam})$$

$$W = 400 \text{ Wh} + 300 \text{ Wh} + 600 \text{ Wh}$$

$$W = 1300 \text{ Wh}$$

$$W = 1,3 \text{ kWh/hari}$$

- b. Pada bulan Maret 2021 terdapat 31 hari, maka rekening listrik yang harus dibayar di bulan Maret yakni:

$$\text{Biaya} = (1,3 \text{ kWh/hari})(\text{Rp}1.444,70 / \text{kWh})(31 \text{ hari})$$

$$\text{Biaya} = \text{Rp}58.221,51$$

## 2.5 Android Studio

Android adalah suatu sistem operasi seluler yang pertama kali dikembangkan oleh perusahaan Silicon Valley dengan nama Android Inc. Android menyediakan platform untuk Developer Junior maupun senior mengembangkan aplikasi yang sendiri atau kelompok. Android terdapat fitur-fitur canggih sesuai dengan developer (bitlabs, 2022).

Menurut (STUDIO, n.d.), Android Studio adalah *Integrated Development Environment (IDE)* resmi untuk pengembangan aplikasi Android. Berbasis editor kode dan alat developer yang andal dari *IntelliJ IDEA*, Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas peneliti saat mem-build aplikasi Android, seperti:

- Sistem build berbasis Gradle yang fleksibel
- Framework dan alat pengujian yang lengkap
- Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Emulator yang cepat dan kaya fitur

Bahasa program atau platform untuk membuat suatu produk aplikasi versi Android seperti Futter,Kotlin dan lain sebagainya. Dari beberapa tersebut peneliti memilih Flutter karena memudahkan bagi peneliti atau *development* dengan fitur fitur yang *fungsional* seperti dapat langsung melihat hasil dari perubahan (Dicoding, 2020).

Teori landasan Android ini dapat digunakan untuk membangun aplikasi Android yang dapat menampilkan data monitoring listrik, baik dalam bentuk tabel atau grafik. *Layouts* dapat digunakan untuk mengatur tampilan antarmuka pengguna monitoring listrik, sementara *Activities* dapat digunakan untuk menangani interaksi pengguna monitoring listrik . *Services* dapat digunakan untuk melakukan tugas yang berjalan di latar belakang, seperti memperbarui data monitoring secara berkala, dan *SQLite Database* dapat digunakan untuk menyimpan dan mengakses data monitoring listrik.

Pada penelitian ini memiliki Proses dan mekanisme pembuatan aplikasi pada android studio versi 2022,2.1.20 sebagai berikut :

a. Analisis Kebutuhan:

- Identifikasi kebutuhan pengguna dan tujuan aplikasi.
- Analisis fitur dan fungsionalitas yang diperlukan dalam aplikasi monitoring daya listrik.
- Penentuan integrasi dengan *Firebase* untuk penyimpanan dan sinkronisasi data.

b. Desain Sistem:

- Perancangan arsitektur sistem aplikasi, termasuk antarmuka pengguna dan fitur-fitur utama.
- Desain basis data untuk menyimpan data daya listrik.
- Penentuan skema penggunaan algoritma *Fuzzy Logic Sugeno* untuk pengambilan keputusan berdasarkan data yang diterima.

c. Pengembangan Aplikasi Android:

- Pembuatan antarmuka pengguna menggunakan Android Studio.
- Implementasi logika aplikasi untuk mengambil data daya listrik dari perangkat sensor, seperti PZEM004T.
- Integrasi dengan *Firebase* untuk penyimpanan data dan sinkronisasi dengan aplikasi.

d. Implementasi Algoritma *Fuzzy Logic Sugeno*:

- Penentuan variabel masukan dan keluaran yang relevan untuk sistem monitoring daya listrik.
- Pembuatan aturan-aturan *Fuzzy* berdasarkan kondisi penggunaan daya listrik.
- Implementasi metode de*Fuzzifikasi* untuk menghasilkan output yang dapat dimengerti.

e. Pengujian dan Evaluasi:

- Pengujian aplikasi untuk memastikan keakuratan pengambilan data daya listrik dan kinerja algoritma *Fuzzy*.
- Evaluasi respons dan pengalaman pengguna terhadap antarmuka dan fungsionalitas aplikasi.
- Pembaruan dan perbaikan berdasarkan umpan balik dan hasil pengujian.

f. Implementasi dan Penyebaran:

- Menyiapkan infrastruktur yang diperlukan untuk menjalankan aplikasi, seperti server *Firebase* dan sumber daya komputasi.

- Mengelola dan menerapkan keamanan untuk melindungi data pengguna dan menghindari penyalahgunaan sistem.
- Peluncuran aplikasi kepada pengguna dan pemantauan kinerja serta pemeliharaan secara berkala.

Pada akhirnya, aplikasi "Sistem Monitoring Daya Listrik menggunakan *Firebase* berbasis Android dengan Algoritma *Fuzzy Logic Sugeno*" akan memberikan kemampuan kepada pengguna untuk memantau penggunaan daya listrik secara real-time, melakukan analisis berdasarkan data yang diterima, dan memberikan rekomendasi atau tindakan yang sesuai untuk mengoptimalkan penggunaan daya listrik.

## 2.6 Platform IoT

Platform IoT menurut (Sakovich, 2023) adalah suatu sistem atau lingkungan yang menyediakan infrastruktur dan layanan yang diperlukan untuk mengembangkan, menghubungkan, mengelola, dan menganalisis solusi Internet of Things (IoT). Platform ini memungkinkan pengembang dan pengguna untuk mengintegrasikan perangkat IoT, mengumpulkan dan menganalisis data dari perangkat tersebut, serta mengelola dan mengendalikan perangkat IoT secara efisien. Berikut beberapa platform IoT yang digunakan oleh developer IoT (Indobot, 2023):

### a. *Firebase*

*Firebase* adalah platform pengembangan aplikasi berbasis cloud yang juga dapat digunakan sebagai platform IoT. *Firebase* menyediakan berbagai fitur, termasuk penyimpanan data real-time, autentikasi pengguna, hosting, analitik, dan layanan messaging. Platform ini memungkinkan pengembang untuk dengan mudah mengintegrasikan aplikasi dengan perangkat IoT dan mengelola data yang dikirim dari perangkat tersebut.

### b. Ubidots

Ubidots adalah platform IoT yang dirancang untuk memantau dan mengelola perangkat IoT secara real-time. Ubidots menyediakan alat untuk membangun aplikasi, membuat visualisasi data, mengatur notifikasi, serta menganalisis dan menyimpan data dari perangkat IoT. Platform ini menyediakan API yang mudah digunakan untuk menghubungkan perangkat dan aplikasi dengan cepat.

c. ThingSpeak

ThingSpeak adalah platform IoT berbasis cloud yang dikembangkan oleh MathWorks. Platform ini menyediakan alat untuk mengumpulkan, menganalisis, dan memvisualisasikan data dari perangkat IoT. ThingSpeak juga mendukung integrasi dengan MATLAB, sehingga memungkinkan pengguna untuk melakukan analisis yang lebih kompleks pada data yang dikirim oleh perangkat IoT.

d. Blynk

Blynk adalah platform IoT yang dirancang untuk memudahkan pengembangan aplikasi IoT melalui ponsel pintar. Platform ini menyediakan aplikasi seluler yang memungkinkan pengguna untuk mengontrol dan memantau perangkat IoT yang terhubung. Blynk juga menyediakan library dan API untuk berbagai perangkat mikrokontroler dan protokol komunikasi.

e. Antares

Antares adalah platform IoT berbasis cloud yang menyediakan layanan untuk menghubungkan, mengelola, dan mengontrol perangkat IoT. Antares memiliki fitur untuk mengumpulkan dan menganalisis data, membangun visualisasi, serta mengatur peristiwa dan notifikasi berdasarkan data yang diterima dari perangkat IoT.

f. ThingsBoard

ThingsBoard adalah platform sumber terbuka (open-source) yang menyediakan fitur lengkap untuk membangun solusi IoT. Platform ini menyediakan alat untuk mengelola perangkat, mengumpulkan dan menganalisis data, serta membangun visualisasi yang interaktif. ThingsBoard mendukung berbagai protokol komunikasi dan memungkinkan integrasi dengan platform cloud dan sistem lainnya.

g. Thinger.io

Thinger.io adalah platform IoT yang dirancang untuk memudahkan pengembangan dan manajemen perangkat IoT. Platform ini menyediakan fitur-fitur seperti otentikasi pengguna, pengelolaan perangkat, dan visualisasi data. Thinger.io juga mendukung integrasi dengan platform cloud populer seperti AWS, Azure, dan Google Cloud.

h. Telkomsel IoT

Telkomsel IoT adalah platform IoT yang ditawarkan oleh Telkomsel, salah satu operator telekomunikasi terbesar di Indonesia. Platform ini menyediakan solusi end-to-end untuk menghubungkan, mengelola, dan mengamankan perangkat IoT. Telkomsel IoT juga menyediakan berbagai fitur seperti pemantauan, pengelolaan

i. Geeknesia

Geeknesia adalah sebuah platform yang bertujuan untuk membantu pengembangan aplikasi berbasis Internet of Things (IoT) dengan lebih mudah. Platform ini menawarkan berbagai fitur dan layanan, termasuk Internet of Things (IoT) Dashboard, Integrasi Perangkat IoT, Manajemen Perangkat, Layanan Cloud, API dan SDK.

Dalam penelitian ini, *Firebase* dipilih sebagai platform untuk peneliti untuk sistem monitoring listrik karena menyediakan fitur yang relevan seperti real-time

database, hosting web, autentikasi, keamanan, dan integrasi dengan Google Cloud Platform.

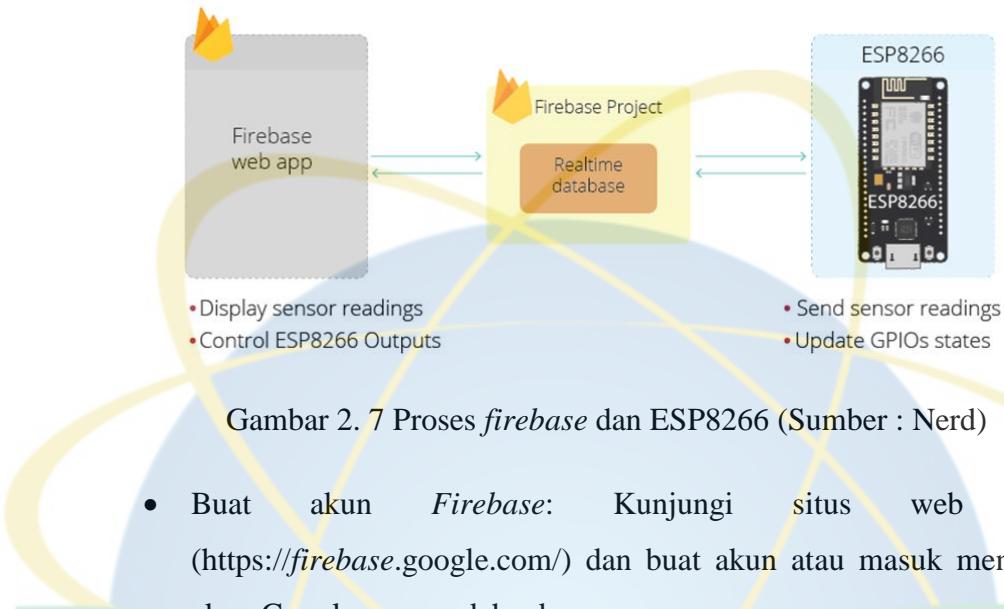
## 2.7 *Firebase*

*Firebase* adalah platform seluler yang membantu developer mengembangkan aplikasi berkualitas tinggi secara cepat, berbasis pengguna *Firebase* terdiri dari fitur pelengkap yang bisa dipadukan sesuai dengan kebutuhan (Google). Digunakannya *firebase*, karena platform yang telah dikembangkan oleh Google ini memiliki fitur cukup mumpuni untuk dimanfaatkan pada beberapa. *Database firebase* akan melakukan sinkronisasi secara otomatis terhadap aplikasi client yang terhubung kepadanya. Aplikasi multi platform yang menggunakan SDK Android, IOS dan *JavaScript* akan menerima update data terbaru secara otomatis pada saat aplikasi terhubung ke *server firebase*. Tujuan utama dari realtime database *firebase* adalah kinerja waktu akses data. Realtime database mengoptimalkan waktu akses, sehingga akses data berada dikisaran mikrodetik bahkan nanodetik, sehingga biaya akses data dapat diminimalisir (intrern, 2020).

Cloud Firestore merupakan database terbaru dari *Firebase* untuk pengembangan aplikasi seluler. Database ini melanjutkan keberhasilan Realtime Database dengan model data baru yang lebih intuitif. Seperti *Firebase* Realtime Database, Cloud Firestore membuat data tetap terhubung di aplikasi klien melalui listener realtime dan menawarkan dukungan secara offline untuk seluler dan web (Ilham Firman Maulana, 2020)

Firestore menggunakan konsep dokumentasi, di mana setiap data disimpan dalam bentuk dokumen dan diorganisir ke dalam koleksi (collection). Setiap dokumen memiliki ID yang unik dan dapat diakses dengan mudah. Firestore juga memiliki fitur seperti indexing, querying, dan event listener untuk memudahkan pengelolaan data dan memungkinkan pengguna untuk mendapatkan data secara real-time. (F, Memilih database: Cloud Firestore atau Realtime Database, 2022)

Pada proses yang didapatkan dari (NERD, 2021) berikut langkah-langkah yang dilakukan untuk menghubungkan *Firebase* dengan *ESP8266*:



Gambar 2. 7 Proses *firebase* dan *ESP8266* (Sumber : Nerd)

- Buat akun *Firebase*: Kunjungi situs web *Firebase* (<https://firebase.google.com/>) dan buat akun atau masuk menggunakan akun Google yang sudah ada.
- Buat Proyek *Firebase*: Setelah masuk ke akun *Firebase*, buat proyek baru dengan mengklik tombol "Tambahkan Proyek" dan ikuti langkah-langkah yang diberikan. Berikan nama proyek dan pilih lokasi yang sesuai.
- Konfigurasi *Firebase* Realtime Database: Setelah proyek dibuat, pergi ke menu "Database" di panel *Firebase* Console. Pilih opsi "FireStore firebase" dan aktifkan database. Anda dapat mengatur aturan akses database sesuai kebutuhan Anda.
- Dapatkan API Key: Kembali ke panel *Firebase* Console, pilih ikon "pengaturan" di bagian atas dan pergi ke tab "Umum". Di sana, akan menemukan informasi API *Firebase* pengguna, termasuk API Key. Catat API Key ini karena akan digunakan dalam kode *ESP8266*.
- Konfigurasi Koneksi WiFi pada *ESP8266*: Dalam kode *ESP8266*, pengguna perlu mengatur koneksi WiFi menggunakan SSID dan kata sandi jaringan WiFi yang digunakan.
- Pasang Library *Firebase* *ESP8266*: Dalam Arduino IDE, instal library *Firebase* *ESP8266* dengan menggunakan Library Manager. Pilih menu

"Sketch" -> "Include Library" -> "Manage Libraries" dan cari "*Firebase ESP8266*" untuk menginstalnya.

- Buat Kode ESP8266: Buat kode pada Arduino IDE untuk ESP8266 yang akan menghubungkan ke *Firebase*. Anda perlu menyertakan informasi SSID dan kata sandi WiFi, serta API Key *Firebase* yang telah dapatkan sebelumnya. Pengguna juga perlu mengatur referensi database *Firebase* yang ingin Anda akses.
- Unggah dan Jalankan Kode: Setelah selesai membuat kode, unggah ke ESP8266 dan jalankan. Anda dapat memantau output di Serial Monitor untuk memastikan bahwa ESP8266 berhasil terhubung ke *Firebase*



Gambar 2.8 Cara Kerja *Firebase* (Sumber :Nyebarilmu.com)

Pada penelitian ini untuk melakukan proses *firebase* versi 9.8.0, Langkah awal adalah melakukan registrasi dan konfigurasi proyek *Firebase* melalui konsol *Firebase*. Hal ini meliputi pembuatan proyek *Firebase*, pengaturan keamanan, dan mengintegrasikan proyek dengan aplikasi Android yang dikembangkan menggunakan Android Studio.

*Firebase* menyediakan layanan penyimpanan data secara real-time dan NoSQL. Dalam penelitian ini, data daya listrik yang dikirim oleh perangkat IoT

akan disimpan di *Firebase Database*. *Firebase Database* adalah basis data cloud yang dapat menyimpan dan menyinkronkan data secara otomatis antara perangkat dan server. Data daya listrik, seperti penggunaan daya, waktu, dan parameter lainnya, akan dikirim oleh perangkat IoT dan disimpan di *Firebase Database*.

Otentikasi Pengguna; Jika diperlukan, *Firebase* juga dapat digunakan untuk mengelola otentikasi pengguna dalam aplikasi. Misalnya, pengguna dapat login menggunakan akun *Firebase Authentication* untuk mengakses fitur-fitur tertentu dalam sistem monitoring daya listrik. Hal ini memberikan keamanan dan pengelolaan akses yang lebih baik.

*Firebase Cloud Messaging* (FCM) dapat digunakan untuk mengirimkan pesan push ke perangkat Android. Dalam konteks penelitian ini, FCM dapat digunakan untuk memberi notifikasi pengguna terkait pemantauan daya listrik, pembaruan data, atau peringatan lainnya.

Integrasi dengan Aplikasi Android, Melalui *Android Studio*, aplikasi Android akan diintegrasikan dengan *Firebase SDK* (Software Development Kit). *SDK Firebase* menyediakan berbagai fungsi dan metode yang memungkinkan pengembang untuk berkomunikasi dengan layanan *Firebase*, seperti membaca dan menulis data ke *Firebase Database*, mengelola otentikasi pengguna, atau mengirim pesan melalui FCM.

Sinkronisasi Data, Salah satu keunggulan *Firebase* adalah kemampuannya untuk menyinkronkan data secara real-time antara perangkat dan server. Dalam skripsi ini, data daya listrik yang disimpan di *Firebase Database* akan diakses dan diperbarui oleh aplikasi Android. Ketika perangkat IoT mengirimkan data baru, *Firebase* akan secara otomatis memperbarui data di *Firebase Database*, dan aplikasi Android akan menerima pembaruan tersebut secara real-time. Untuk mendapatkan keputusan berdasarkan data yang diterima dari sensor-sensor dalam sistem.

## 2.8 *Fuzzy Logic*

Logika *Fuzzy* pertama kali diperkenalkan oleh Lutfi Zadeh pada pertengahan tahun 1960 di Universitas California Barkeley. Logika ini diciptakan karena

*Boolean logic* tidak mempunyai ketelitian yang tinggi, hanya mempunyai logika 0 dan 1 saja, sehingga untuk membuat sistem yang mempunyai ketelitian yang tinggi maka kita tidak dapat menggunakan *Boolean logic*. Sedangkan logika *Fuzzy* merupakan sebuah logika yang memiliki nilai kekaburan atau kesamaran (*Fuzzyness*) antara benar dan salah. Dalam teori logika *Fuzzy* sebuah nilai bisa bernilai benar dan salah secara bersamaan namun berapa besar kebenaran dan kesalahan suatu nilai tergantung kepada bobot keanggotaan yang dimilikinya. Hal ini sejalan dengan penelitian yang di paparkan oleh (Suprapto & Simanjuntak, 2020).

Menurut (Suprapto & Simanjuntak, 2020)terdapat tiga metode yang ada di *Fuzzy logic* yaitu:

1. Tsukamoto
2. Mamdani
3. Sugeno

## 2.9 Metode Sugeno

Metode Sugeno merupakan salah satu metode dalam logika *Fuzzy*. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Sistem *Fuzzy* sugeno memperbaiki kelemahan yang dimiliki oleh sistem *Fuzzy* murni untuk menambah suatu perhitungan matematika sederhana sebagai bagian THEN. Pada perubahan ini, sistem *Fuzzy* memiliki suatu nilai rata-rata tertimbang (Weighted Average Values) di dalam bagian aturan *Fuzzy* IF-THEN. Sistem *Fuzzy* Sugeno juga memiliki kelemahan terutama pada bagian THEN, yaitu dengan adanya perhitungan matematika sehingga dapat menyediakan kerangka alami untuk mempresentasikan pengetahuan manusia dengan sebenarnya (Rizky et al., 2020).

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI, hanya saja output (konsekuensi) sistem tidak berupa himpunan *Fuzzy*, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh

Takagi - Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan Metode TSK(Tukadi et al., 2019) (Suprapto & Simanjuntak, 2020).

Metode *Fuzzy Mamdani* merupakan tipe FIS standart yang umum digunakan. Metode Mamdani memungkinkan kita untuk mendeskripsikan keahlian kita dengan cara yang lebih intuitif dan lebih manusiawi. FIS Mamdani memerlukan beban komputasi yang substansial yang mengakibatkan metode ini kurang berhasil sebab harus menghitung luas daerah di bawah kurva. Oleh karena itu, digunakan alternatif untuk menggunakan FIS tipe Metode Sugeno, yang diperkenalkan oleh Takagi-Sugeno-Kang(ICES, 2021).

Penelitian ini menggunakan *Fuzzy Sugeno* dikarenakan Metode *Fuzzy Sugeno* efektif dalam komputasi dan bekerja dengan baik dengan optimalisasi. metode *Fuzzy Sugeno* juga menjamin kontinuitas permukaan output serta lebih cocok untuk analisis secara sistematis

Menurut (widia, 2021) ada beberapa hal yang perlu diketahui dalam memahami metode *Fuzzy* yaitu.

### 1. Variabel *Fuzzy*

Variabel *Fuzzy* merupakan suatu lambang atau kata yang menunjuk kepada suatu yang tidak tertentu dalam sistem fuzz, contoh: daya listrik, biaya listrik, waktu dll.

### 2. Himpunan *Fuzzy*

Himpunan *Fuzzy* merupakan suatu kumpulan yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *Fuzzy*, Contoh :

- a. Variabel daya terbagi menjadi tiga himpunan *Fuzzy*, yaitu: RENDAH, SEDANG dan Tinggi
  - b. Variabel Biaya listrik terbagi menjadi lima himpunan *Fuzzy*, yaitu: NORMAL dan MAHAL
3. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *Fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Pada suatu kondisi tertentu nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh :

- a. Semesta pembicaraan untuk variabel daya listrik [0 4]
- b. Semesta pembicaraan untuk variabel biaya listrik [0 30000]

#### 4. Domain Domain

Domain himpunan *Fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *Fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif. Contoh domain himpunan *Fuzzy* :

- a. RENDAH = [0, 1.33]
- b. SEDANG = [0,83,2.66]
- c. TINGGI= [2.33, 4]

##### 2.9.1 Fungsi Keanggotaan

Menurut (Rizky et al., 2020) ada beberapa hal yang perlu diketahui dalam memahami metode *Fuzzy* yaitu.

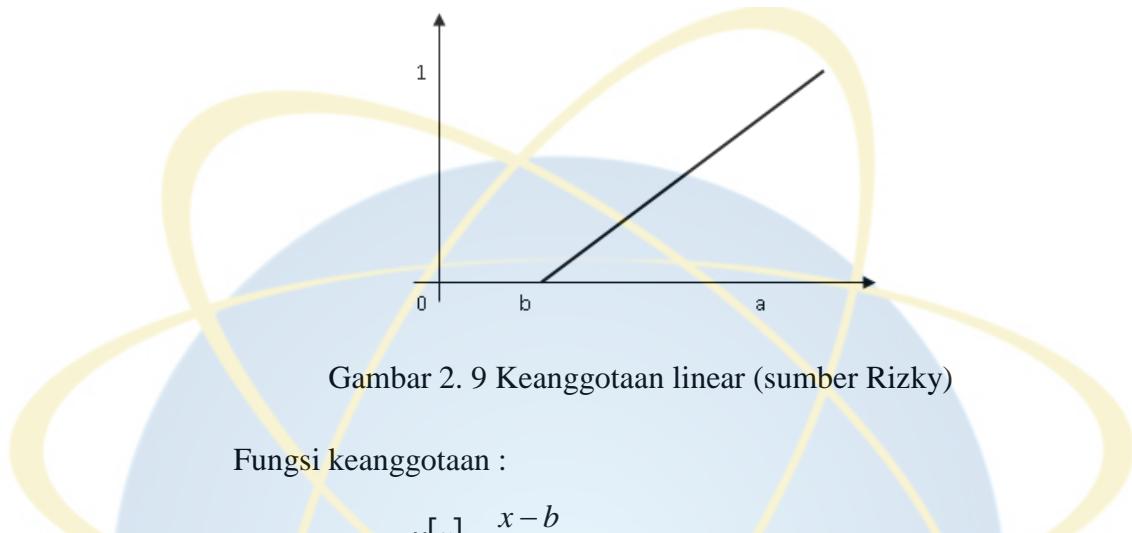
$$\tilde{A}: X \rightarrow [0,1]$$

Kebanyakan himpunan kabur berada dalam semesta bilangan *real* dengan fungsi keanggotaan yang dinyatakan dalam bentuk formula matematis antara lain sebagai berikut:

- a. Fungsi keanggotaan kurva linear

Kurva linear dapat dinyatakan dalam bentuk persamaan garis lurus, di mana variabel input berada pada sumbu x dan derajat

keanggotaan berada pada sumbu y. Pada kurva linear, setiap nilai input memiliki derajat keanggotaan yang proporsional dengan jaraknya terhadap titik awal dan titik akhir pada kurva.

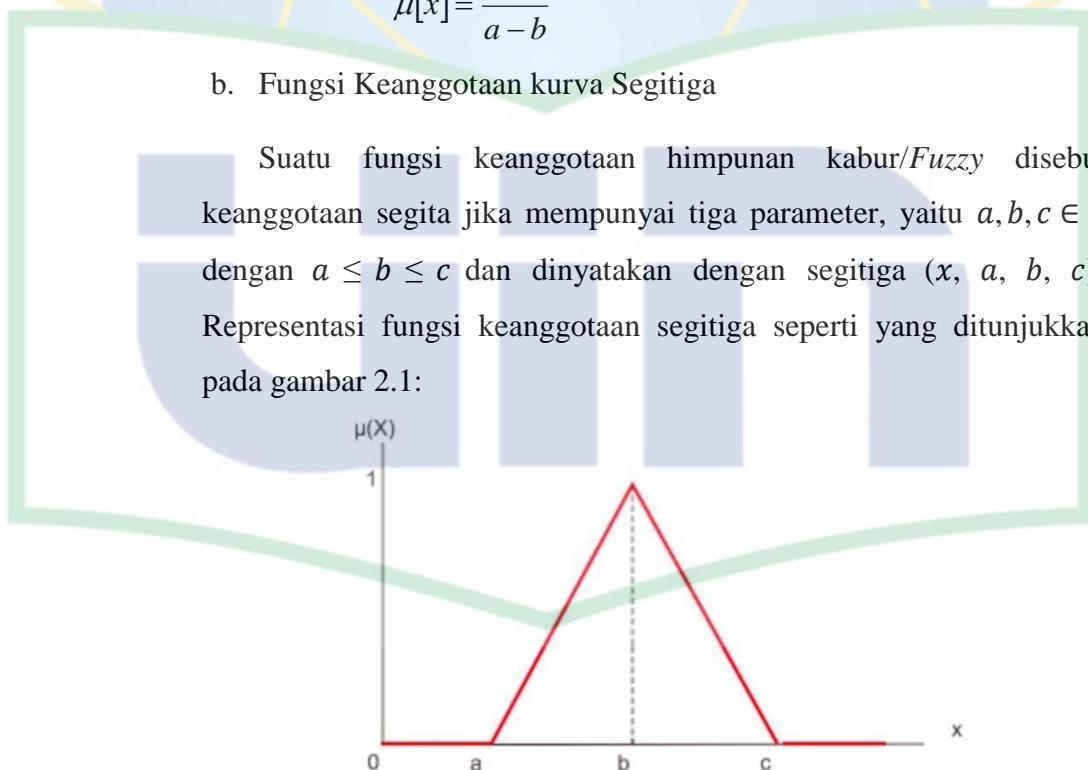


Fungsi keanggotaan :

$$\mu[x] = \frac{x-b}{a-b}$$

#### b. Fungsi Keanggotaan kurva Segitiga

Suatu fungsi keanggotaan himpunan kabur/Fuzzy disebut keanggotaan segita jika mempunyai tiga parameter, yaitu  $a, b, c \in R$  dengan  $a \leq b \leq c$  dan dinyatakan dengan segitiga  $(x, a, b, c)$ . Representasi fungsi keanggotaan segitiga seperti yang ditunjukkan pada gambar 2.1:



Gambar 2. 10 Keanggotaan Segitiga (Sumber : Researhgate)

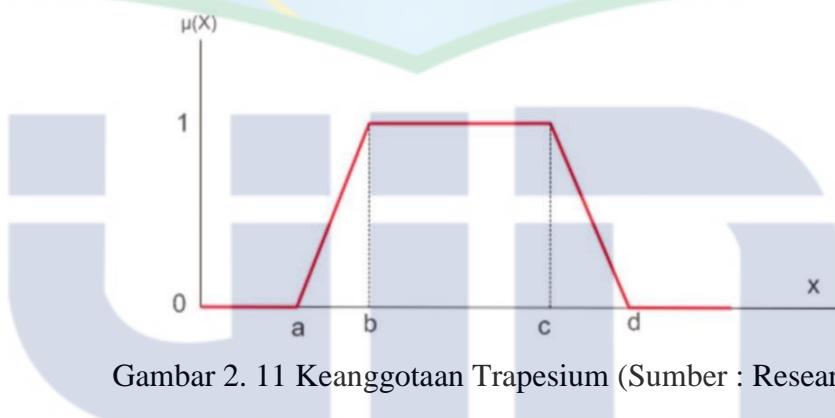
Fungsi keanggotaan segitiga dapat juga dinyatakan dengan

formula sebagai berikut

$$\mu(x) = \begin{cases} 0; & \text{Jika } x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}; & \text{Jika } a \leq x \leq b \\ \frac{b-x}{c-b}; & \text{Jika } b \leq x \leq c \\ 0; & \text{Jika } x \geq c \end{cases}$$

### c. Fungsi Keanggotaan kurva Trapesium

Suatu fungsi keanggotaan himpunan kabur disebut fungsi keanggotaan trapesium jika mempunyai empat parameter, yaitu  $a, b, c, d \in R$  dengan  $a \leq b \leq c \leq d$  dan dinyatakan dengan trapesium. Representasi fungsi keanggotaan trapesium seperti yang ditunjukkan pada gambar 2.2:



Gambar 2. 11 Keanggotaan Trapesium (Sumber : ResearchGate)

Fungsi keanggotaan trapesium dapat juga dinyatakan dengan formula sebagai berikut:

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)} & \text{jika } a \leq x \leq b \\ 1 & \text{jika } b \leq x \leq c \\ \frac{(d-x)}{(d-c)} & \text{jika } c \leq x \leq d \end{cases}$$

Keterangan:

$a$  = Nilai domain yang mempunyai derajat keanggotaan

$b$  = Nilai domain yang mempunyai derajat keanggotaan satu

$c$  = Nilai domain yang mempunyai derajat keanggotaan nol

$d$  = Nilai domain yang mempunyai derajat keanggotaan satu

$x$  = Nilai input yang akan diubah ke dalam bilangan kabur

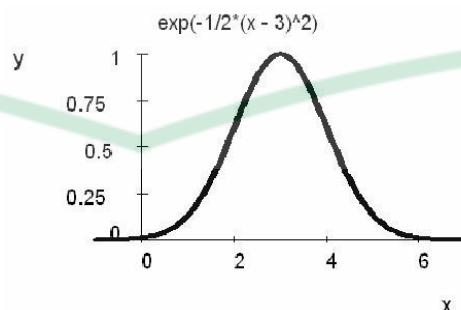
#### d. Fungsi keanggotaan Kurva Gauss

Fungsi keanggotaan *Fuzzy* dengan kurva Gauss memiliki dua parameter utama, yaitu mean (rerata) dan standard deviation (simpangan baku). Parameter ini mengontrol letak dan bentuk kurva Gauss. Mean menentukan pusat kurva, sedangkan standard deviation mengendalikan sebaran data di sekitar mean. Bentuk umum fungsi keanggotaan *Fuzzy* Gauss adalah:

$$\mu(x) = \exp(-(x - \text{mean})^2 / (2 * \text{standard deviation}^2))$$

Di mana:

- $\mu(x)$  adalah derajat keanggotaan pada suatu nilai  $x$ .
- mean adalah rerata atau pusat kurva.
- standard deviation adalah simpangan baku atau sebaran data.



Gambar 2.12 Kurva gauss (Sumber : ResearchGate)

### Fungsi keanggotaan

$$\mu[x] = e^{-\frac{(x-c)^2}{2s^2}}$$

#### e. Fungsi keanggotaan kurva *generalized bell*

Fungsi keanggotaan *Fuzzy* dengan kurva Generalized Bell, juga dikenal sebagai fungsi keanggotaan *Fuzzy Bell*, adalah salah satu jenis fungsi keanggotaan yang digunakan dalam logika *Fuzzy*. Fungsi ini memiliki bentuk kurva mirip lonceng yang dapat disesuaikan dengan parameter tertentu.

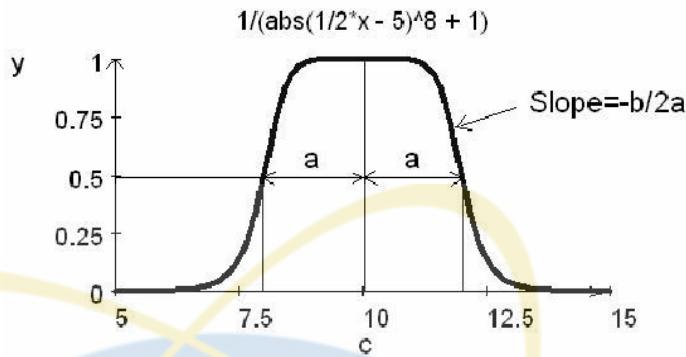
Fungsi keanggotaan *Fuzzy* dengan kurva Generalized Bell memiliki tiga parameter utama, yaitu  $a$  (nilai tengah),  $b$  (lebar kurva), dan  $c$  (kecerunan kurva). Parameter ini mengendalikan bentuk dan karakteristik kurva.

Bentuk umum fungsi keanggotaan *Fuzzy Generalized Bell* adalah:

$$\mu(x) = 1 / (1 + |(x - c) / a|^{2b})$$

Di mana:

- $\mu(x)$  adalah derajat keanggotaan pada suatu nilai  $x$ .
- $a$  adalah parameter yang mengendalikan letak nilai tengah atau puncak kurva.
- $b$  adalah parameter yang mengendalikan lebar atau kelebaran kurva.
- $c$  adalah parameter yang mengendalikan kecerunan atau bentuk kurva.



Gambar 2. 13 Kurva generalized bell (Sumber : ResearchGate)

$$\text{Fungsi keanggotaan : } \mu[x] = \frac{1}{\left| \frac{x}{a} - \frac{c}{a} \right|^{2b} + 1}$$

#### f. Fungsi keanggotaan kurva sigmoid

Fungsi keanggotaan kurva sigmoid adalah salah satu jenis fungsi keanggotaan yang digunakan dalam logika *Fuzzy*. Kurva sigmoid memiliki bentuk yang menyerupai tangen hiperbolik atau fungsi logistik, dengan pertumbuhan yang lambat di awal dan akhir namun tumbuh secara eksponensial di tengah rentang nilai.

Fungsi keanggotaan kurva sigmoid umumnya digunakan untuk menggambarkan peningkatan atau penurunan yang lebih lambat di daerah nilai ekstrem dan peningkatan yang lebih cepat di sekitar nilai tengah. Hal ini bermanfaat dalam situasi di mana variabel input memiliki tingkat keanggotaan yang rendah atau tinggi di daerah nilai ekstrem, tetapi meningkat dengan cepat di sekitar nilai tengah.

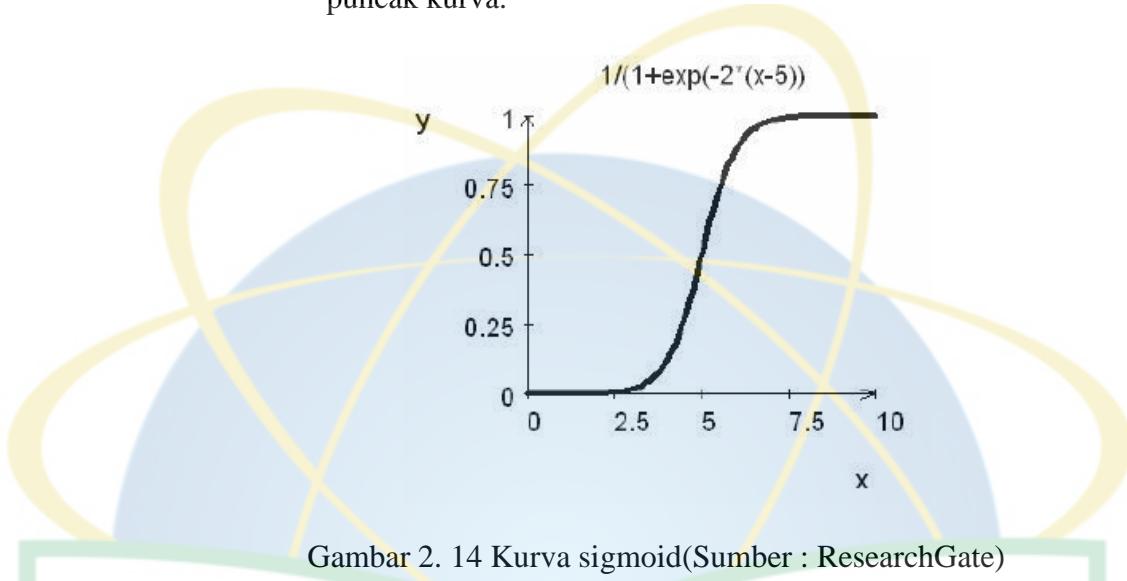
Bentuk umum dari fungsi keanggotaan kurva sigmoid adalah:

$$\mu(x) = 1 / (1 + e^{-a(x - c)})$$

Di mana:

- $\mu(x)$  adalah derajat keanggotaan pada suatu nilai  $x$ .

- $a$  adalah parameter yang mengontrol kecuraman atau kemiringan kurva.
- $c$  adalah parameter yang mengontrol posisi nilai tengah atau puncak kurva.



Fungsi keanggotaan :

$$\mu[x] = \frac{1}{e^{-a(x-c)} + 1}$$

Dalam penelitian ini terdapat kurva himpunan *Fuzzy* untuk menggambarkan data keanggotaan yang didapatkan. Kurva yang dihasil dari *Fuzzy* ada dua menurut(Rizky et al., 2020) yaitu kurva segitiga linear seperti pembahasan 2.8.1. dan penelitian ini mengambil kurva segitiga dan trapesium dimana dengan kondisi yang berbeda. Variabel yang digambarkan kurva trapesium adalah variabel KwH, Jumlah perangkat,biaya listrik dan waktu pengguna dikarena memiliki nilai keanggotaan yang *absolute* atau tetap, untuk variabel yang digambarkan kurva segitiga hanyalah varibel daya listrik atau tegangan listrik karena memiliki nilai keanggotaan yang bersifat *Fuzzy* atau ambiguitas.

Ketiga jenis fungsi keanggotaan tersebut relatif sederhana dalam hal representasi matematis dan interpretasi intuitif. Kurva linear memiliki

pertumbuhan yang konstan, sedangkan segitiga memiliki bentuk segitiga sederhana, dan trapesium memiliki bentuk trapesium dengan dua sisi paralel.

### 2.9.2 Tahapan Metode Sugeno

(Rizky et al., 2020) menjelaskan terdapat beberapa tahapan untuk mendapatkan suatu output menggunakan metode Sugeno yaitu sebagai berikut:

#### a. Pembentukan Himpunan Fuzzy

Pada tahapan ini, variabel input maupun variabel output menjadi satu atau lebih himpunan *Fuzzy*. Terdapat variabel variabel input yang didapatkan dari jurnal (Suprapto & Simanjuntak, 2020) yaitu data variabel KwH yang sudah diteliti dengan keanggotaanya yang diambil dalam waktu 30 hari sebulan dan dikonversikan menjadi waktu 1 hari, berikut data KwH yang diambil oleh jurnal tersebut :

		Batas minimal		Batas optimal		batas maksimal	
Variabel	Anggota	30	hari	30	1	30	1
KWH	RENDAH(0)	0	0	19,8	0,66	30	1
	SEDANG(1)	24,99	0,833	49,8	1,66	75	2,5
	TINGGI(2)	69,9	2,33	108,8	3,66	120	4

Tabel 2.8 1Tabel himpunan *Fuzzy*

Dari tabel diatas yang diambil data yaitu 1 hari karena penelitian ini menggunakan output Penggunaan listrik dalam waktu satu hari untuk mengetahui hasil *Fuzzy* dengan input variabel KwH , Jumlah perangkat, Waktu , biaya listrik dan Daya listrik yang ada di pembahasan selanjutnya dengan data yang didapatkan.

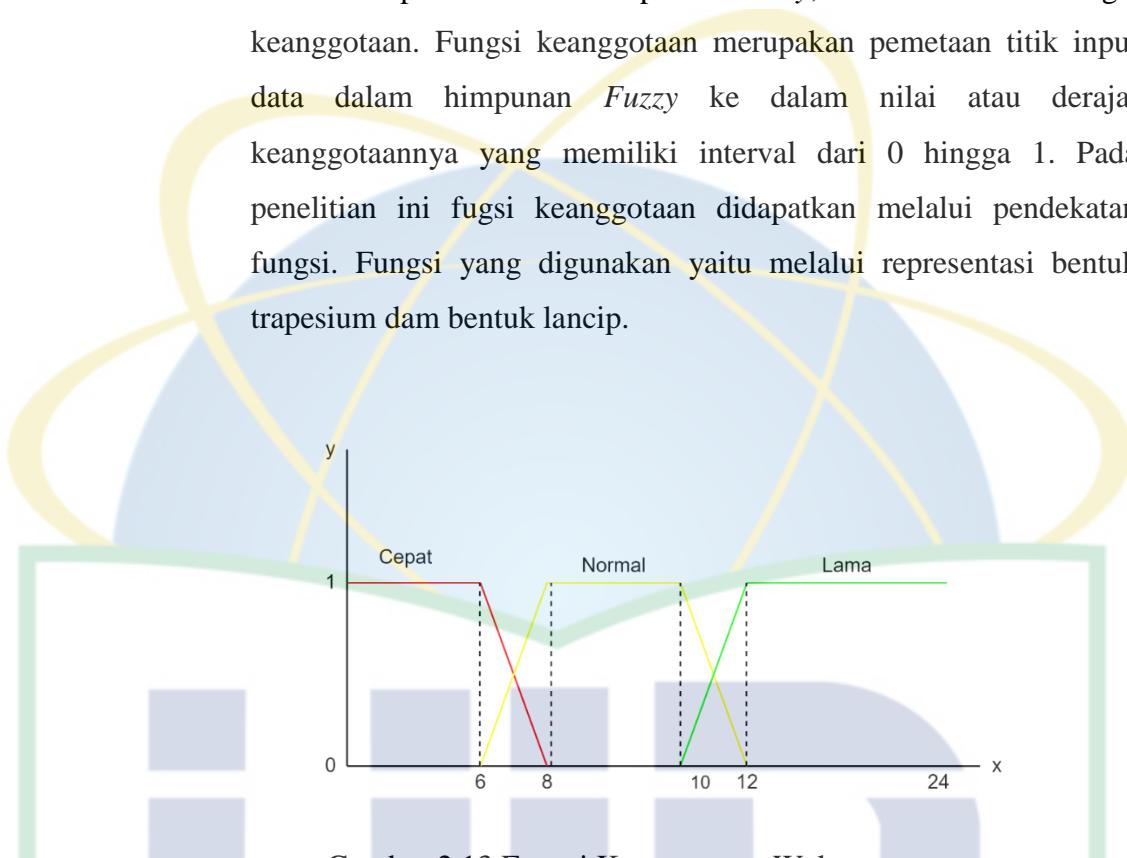
Terdapat empat variabel masukan pada penelitian ini, yaitu variabel energi listrik, daya listrik, waktu penggunaan, biaya listrik dan jumlah perangka. Setiap variabel input selanjutnya dibagi menjadi tiga himpunan *Fuzzy*. Himpunan kabur digunakan untuk mewakili kondisi dalam suatu variabel *Fuzzy*. Dari setiap himpunan kabur yang terbentuk masing-masing mempunyai domain yang nilainya terdapat dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan kabur. Berikut tabel semesta pembicaraan himpunan *Fuzzy*.

Tabel 2. 7 Variabel Himpunan *Fuzzy*

Fungsi	Nama Variabel	Nama Himpunan <i>Fuzzy</i>	Semesta Pembicaraan	Domain
Input	Waktu pengguna	Cepat	[0 , 24]	[0 , 8]
		Normal		[6 , 12]
		Lama		[10 , 24]
	Energi listrik	Rendah	[0 , $\infty$ ]	[0 , 0.833]
		Normal		[0.66 , 2.5]
		Tinggi		[1.66 , $\infty$ ]
	Daya listrik	Rendah	[0 , 2200]	[0 , 1300]
		Sedang		[900, 2200]
		Tinggi		[ 1300,2200]
	Jumlah Perangkat	Sedikit	[0 , 10]	[0 , 4]
		Sedang		[3 , 7 ]
		Banyak		[6 , 10]
	Biaya listrik	Normal	[0,30000]	[0, 13000]
		Mahal		[7600, 30000]
Output	Pengguna	Rendah	[0 , 2]	[0 , 0.66]

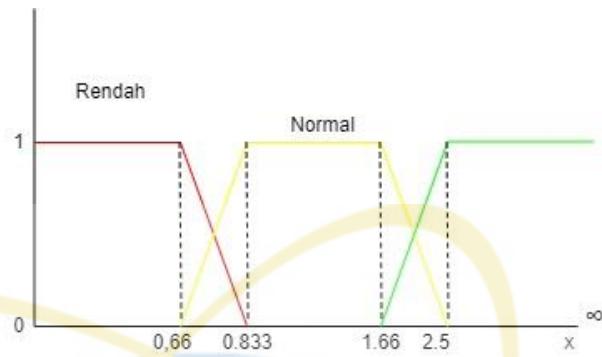
	Listrik	Sedang		[0.66 , 1.33]
		Tinggi		[1.33 , 2]

Setelah pembentukan himpunan *Fuzzy*, maka dibuatlah fungsi keanggotaan. Fungsi keanggotaan merupakan pemetaan titik input data dalam himpunan *Fuzzy* ke dalam nilai atau derajat keanggotaannya yang memiliki interval dari 0 hingga 1. Pada penelitian ini fungsi keanggotaan didapatkan melalui pendekatan fungsi. Fungsi yang digunakan yaitu melalui representasi bentuk trapesium dan bentuk lancip.



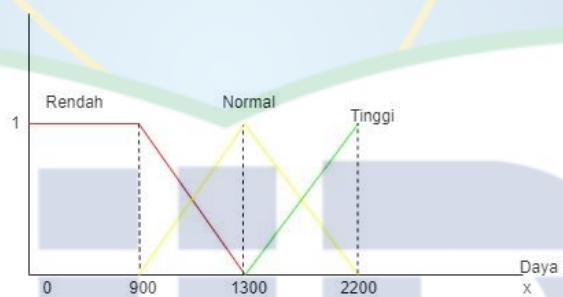
Gambar 2.13 Fungsi Keanggotaan Waktu pengguna

Pada variabel waktu memiliki range  $[0 \ 24]$  dan 3 himpunan *Fuzzy* yaitu cepat, normal, dan lama . Himpunan *Fuzzy* harian memiliki domain  $[0 \ 6 \ 8]$ . Himpunan *Fuzzy* normal memiliki domain  $[6 \ 8 \ 10 \ 12]$ . Himpunan *Fuzzy* lama memiliki domain  $[10 \ 12 \ 24]$



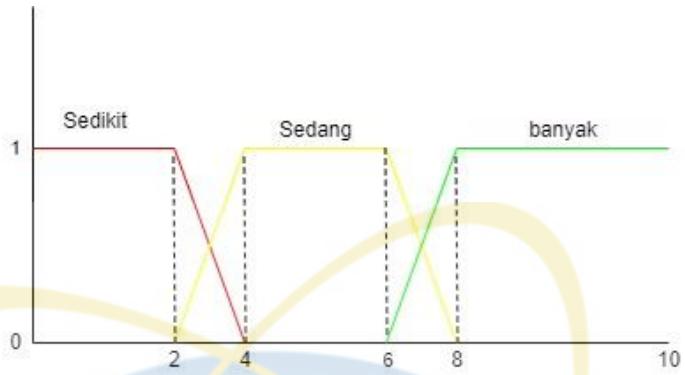
Gambar 2. 14 Fungsi Keanggotaan Energi pengguna

Pada variabel total kwh memiliki range [0 equals] dan 3 himpunan *Fuzzy* yaitu rendah, sedang, dan tinggi. Himpunan *Fuzzy* rendah memiliki domain [0 0.66 0.83]. Himpunan *Fuzzy* sedang memiliki domain [0.66 0.83 1 1.66]. Himpunan *Fuzzy* tinggi memiliki domain [1.66 2.5]



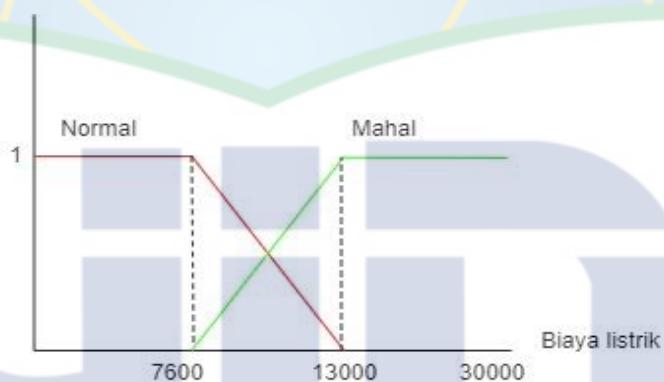
Gambar 2. 15 Fungsi Keanggotaan Daaya Listrik

Pada variabel daya memiliki range [0,2200] dan 3 himpunan *Fuzzy* yaitu rendah, sedang, dan tinggi. Himpunan *Fuzzy* rendah memiliki domain [0 0 900 1300]. Himpunan *Fuzzy* sedang memiliki domain [ 900 1300 2200. Himpunan *Fuzzy* tinggi memiliki domain [ 1300 2200 x] Bentuk keanggotaan variabel daya



Gambar 2. 16 Fungsi Keanggotaan Jumlah perangkat

Pada variabel jumlah perangkat memiliki range [0,10] dan 3 himpunan *Fuzzy* yaitu sedikit, sedang, dan banyak. Himpunan *Fuzzy* sedikit memiliki domain [0 2 4] Himpunan *Fuzzy* sedang memiliki domain [2 4 6 8] Himpunan *Fuzzy* tinggi memiliki domain [6 8 10] Bentuk keanggotaan variabel jumlah perangkat.



Gambar 2. 15 Fungsi keanggotaan biaya

Pada variabel biaya listrik memiliki range [0,30000] dan 2 himpunan *Fuzzy* yaitu normal, tinggi. Himpunan *Fuzzy* normal memiliki domain [0 7600 13000] Himpunan *Fuzzy* mahal memiliki domain [7600 13000 30000] data biaya listrik diambil dari referensi (Kompas.com, 2021),

Berdasarkan tabel 2.3, pada semua variabel input yaitu daya listrik, energi, jumlah perangkat dan waktu pengguna memiliki masing-masing tiga himpunan, Untuk daya listrik memiliki himpunan yaitu

rendah, sedang dan tinggi dengan domain yang berbeda, energi memiliki himpunan yaitu rendah, sedang dan tinggi dengan domain yang berbeda, jumlah perangkat memiliki himpunan yaitu sedikit, sedang dan banyak dengan domain, dan untuk waktu pengguna memiliki himpunan cepat, sedang dan lama. Berikut hitungan sesuai dengan kurva atau variable keanggotaan

a. Hitungan Variabel keanggotaan waktu pengguna Fuzzy

$$\mu_{\text{Waktu Cepat}}[x] = \begin{cases} 1, & x \leq 6 \\ \frac{8-x}{8-6}, & 6 \leq x \leq 8 \\ 0, & x \geq 8 \end{cases}$$

$$\mu_{\text{Waktu Sedang}}[c] = \begin{cases} 0, & x \leq 6 \text{ atau } x \geq 12 \\ \frac{x-6}{8-6}, & 6 \leq x \leq 8 \\ \frac{12-x}{12-6}, & 10 \leq x \leq 12 \\ 1, & 8 \leq x \leq 10 \end{cases}$$

$$\mu_{\text{Waktu Lama}}[c] = \begin{cases} 0, & x \leq 10 \\ \frac{12-x}{12-10}, & 10 \leq x \leq 12 \\ 1, & x \geq 12 \end{cases}$$

b. Hitungan Variabel keanggotaan KwH Fuzzy

$$\mu_{\text{KwH Rendah}}[x] = \begin{cases} 1, & x \leq 0,66 \\ \frac{0,83-x}{0,83-0,66}, & 0,66 \leq x \leq 0,83 \\ 0, & x \geq 0,83 \end{cases}$$

$$\mu_{\text{KwH Sedang}}[c] = \begin{cases} 0, & x \leq 0,66 \text{ atau } x \geq 2,5 \\ \frac{x-0,66}{0,833-0,66}, & 0,66 \leq x \leq 0,833 \\ \frac{2,5-x}{2,5-1,66}, & 1,66 \leq x \leq 2,5 \\ 1, & 0,833 \leq x \leq 1,66 \end{cases}$$

$$\mu_{\text{KwH Tinggi}}[c] = \begin{cases} 0, & x \leq 1,66 \\ \frac{x - 1,66}{2,5 - 1,66}, & 1,66 \leq x \leq 2,5 \\ 1, & x \geq 2,5 \end{cases}$$

c. Hitungan Variabel keanggotaan Daya lisytik Fuzzy

$$\mu_{\text{daya listrik Rendah}}[x] = \begin{cases} 1, & x \leq 900 \\ \frac{1300 - x}{1300 - 900}, & 900 \leq x \leq 1300 \\ 0, & x \geq 1300 \end{cases}$$

$$\mu_{\text{daya listrik Sedang}}[c] = \begin{cases} 0, & x \leq 900 \text{ atau } x \geq 2200 \\ \frac{x - 900}{1300 - 900}, & 900 \leq x \leq 1300 \\ \frac{2200 - x}{2200 - 1300}, & 1300 \leq x \leq 2200 \end{cases}$$

$\mu_{\text{daya listrik Tinggi}}[c]$

$$= \begin{cases} 0, & x \leq 1300 \\ \frac{x - 1300}{2200 - 1300}, & 2200 \leq x \leq 1300 \\ 1, & x \geq 2200 \end{cases}$$

d. Hitungan Variabel keanggotaan Jumlah perangkat

$$\mu_{\text{Jumlah perangkat sedikit}}[x] = \begin{cases} 1, & x \leq 2 \\ \frac{4 - x}{4 - 2}, & 4 \leq x \leq 2 \\ 0, & x \geq 4 \end{cases}$$

$$\mu_{\text{Jumlah perangkat Sedang}}[x] = \begin{cases} 0, & x \leq 2 \text{ atau } x \geq 8 \\ \frac{x - 2}{4 - 2}, & 2 \leq x \leq 4 \\ \frac{8 - x}{8 - 6}, & 6 \leq x \leq 8 \\ 1, & 4 \leq x \leq 6 \end{cases}$$

$$\mu_{\text{jumlah perankat banyak}}[x] = \begin{cases} 0, & x \leq 6 \\ \frac{x-6}{4-6}, & 6 \leq x \leq 8 \\ 1, & x \geq 8 \end{cases}$$

e. Hitungan Variabel keanggotaan biaya listrik

$$\mu_{\text{Jumlah biaya Normal}}[x] = \begin{cases} 1, & x \leq 7600 \\ \frac{13000 - x}{13000 - 7600}, & 13000 \leq x \leq 7600 \\ 0, & x \geq 13000 \end{cases}$$

$$\mu_{\text{Jumlah biaya Mahal}}[x] = \begin{cases} 0, & x \leq 7600 \\ \frac{x - 7600}{13000 - 7600}, & 13000 \leq x \leq 7600 \\ 1, & x \geq 13000 \end{cases}$$

b. Aplikasi Fungsi Implikasi/Agregasi

Desain program kerja ini diakhiri dengan rekomendasi untuk pengambilan keputusan. Desain rekomendasi keputusan dilakukan dengan terlebih dahulu menentukan fire strength. Nilai keanggotaan sebagai dari 2 himpunan *Fuzzy* dikenal dengan nama Fire Strength atau  $\alpha$ -predikat. Sangat mungkin digunakan operator dasar dalam proses query berupa operator berikut (Operator Dasar Zadeh untuk Operasi Himpunan *Fuzzy*) berikut beberapa operator keputusan atau aturan dari (Indah et al., 2023) sebagai berikut

- a) Interseksi, operator ini berhubungan dengan operasi interseksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan dengan persamaan berikut:

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

- b) Union, operator ini berhubungan dengan operasi union pada

himpunan. A- predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan- himpunan yang bersangkutan dengan persamaan berikut:

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

- c) Komplemen, operator ini berhubungan dengan operasi komplemen pada himpunan. A-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1 dengan persamaan berikut ini:

$$\mu = 1 - \mu_A(x)$$

Pada tahap ini akan dibuat aturan *Fuzzy* dengan operator interseksi dengan fungsi Min, yaitu dengan cara mencari nilai *minimum* berdasarkan aturan ke-*i* dan dapat dinyatakan dengan:

$$a_i = \mu_{Ai}(x) \cap \mu_{Bi}(x) = \min (\mu_{Ai}(x), \mu_{Bi}(x))$$

Keterangan:

$a$  = nilai *minimum* dari himpunan *Fuzzy* *A* dan *B* pada aturan ke-*i*

$\mu_{Ai}(x)$  = derajat keanggotaan  $x$  dari himpunan *Fuzzy* *A* pada aturan ke-*i*

$\mu_{Bi}(x)$  = derajat keanggotaan  $x$  dari himpunan *Fuzzy* *B* pada aturan ke-*i*

Pada sistem *Fuzzy* Sugeno, fungsi implikasi pada umumnya menggunakan fungsi MIN sebagai operator dalam menghitung derajat keanggotaan dari setiap aturan *Fuzzy* yang terpenuhi. Fungsi MIN digunakan untuk mencari nilai minimum antara derajat keanggotaan pada himpunan *Fuzzy* dari setiap variabel input yang

terlibat dalam suatu aturan *Fuzzy*. Dalam hal ini, nilai derajat keanggotaan terkecil dari setiap variabel input akan digunakan sebagai nilai derajat keanggotaan pada aturan *Fuzzy* tersebut. Berikut tabel keanggotaan variable input *Fuzzy sugeno*:

Tabel 2. 8 tabel data keanggotaan *Fuzzy*

Variabel	Anggota	Batas minimal	Batas optimal (tengah)	batas maksimal
DAYA	Rendah (0)	0	900	1300
	Normal(1)	900	1300	2200
	Tinggi(2)	1300	2200	2200
KWH	RENDAH(0)	0	0,66	1
	SEDANG(1)	0,833	1,66	2,5
	TINGGI(2)	2,33	3,66	4
JUMLAH PERANGKAT	sedikit (0)	0	0-1	1,5
	Sedang (1)	1	1,5-2,5	3
	Banyak(2)	2,5	3-x	x
WAKTU(24 JAM)	Cepat(0)	0	0-6	8
	Normal(1)	6	8-12	14
	Lama(2)	12	14-24	24
Biaya	Normal(0)	0	4000	7000
	Tinggi(1)	4000	7000	30000

Misalnya, jika terdapat aturan *Fuzzy* "IF daya listrik rendah AND KWH tinggi AND waktu cepat AND jumlah perangkat banyak THEN tinggi", maka nilai derajat keanggotaan dari himpunan *Fuzzy* "daya listrik rendah", "KWH tinggi", "waktu cepat", dan "jumlah perangkat banyak" akan dihitung terlebih dahulu menggunakan fungsi keanggotaan masing-masing himpunan. Selanjutnya, nilai derajat keanggotaan yang terkecil dari kedua himpunan tersebut akan diambil menggunakan fungsi MIN, dan digunakan sebagai

nilai derajat keanggotaan pada aturan *Fuzzy* tersebut.

Dalam menghitung nilai output pada sistem *Fuzzy Sugeno*, nilai derajat keanggotaan dari setiap aturan *Fuzzy* yang terpenuhi akan dikalikan dengan nilai konsekuensi atau output pada aturan *Fuzzy* tersebut. Hasil perkalian tersebut kemudian akan dijumlahkan dan dibagi dengan jumlah derajat keanggotaan pada semua aturan *Fuzzy* yang terpenuhi, untuk menghasilkan nilai output akhir dari sistem *Fuzzy Sugeno*.

Dalam output atau hasil dari variebel keanggatoaan terdapat 3 Output Penggunaan listrik yaitu RENDAH, SEDANG DAN TINGGI dengan masing masing mempunyai nilai sebagai berikut :



Gambar 2.16 Varibel keanggotaan output *Fuzzy*

Tabel 2. 9 Keanggotaan output *Fuzzy*

Penggunaan	nilai	Derajat keanggotaan
RENDAH	0	0 sampai 0,67 atau 0-2/3
SEDANG	1	2/3 (0,67) sampai 4/3(1,33)
TINGGI	2	4/3 (1,33)sampai 2

Nilai didapatkan sesuai dengan penjelasan diatas karena memiliki output dengan nilai 0,1 dan 2 dan derajat keanggotaan Rendah (0 sampai 0,67 atau 0-2/3), Sedang (2/3 samapi 4/3(1,33)) dan Tinggi 4/3 sampai 2 dengan semua derajat dibagikan dengan

tiga.

c. Komposisi Aturan/ *rules evaluation*

Pembuatan aturan *Fuzzy* dalam menentukan output berdasarkan variabel waktu pengguna, daya listrik, energi listrik dan jumlah perangkat menggunakan metode Sugeno. Aturan ini dibuat untuk menyatakan relasi antara input dan output, sehingga dapat dibentuk menjadi 162 kombinasi aturan. Pembentukan aturan dihasilkan dari kombinasi tiap kondisi tersebut yang dikenal dengan aturan keputusan. Dengan memperhatikan fungsi MIN, yaitu mengambil nilai minimal dari kedua inputan. Setiap aturan terdiri dari 3 antecedent dengan operator yang digunakan untuk menghubungkan adalah operator DAN sedangkan yang memetakan antara input dan output adalah JIKA MAKA. Komposisi aturan ini dibuat dua buah untuk masing-masing output, dimana komposisi ini memiliki aturan yang sama untuk output 3 yaitu pengguna rendah, pengguna sedang dan pengguna tinggi. berikut adalah tabel komposisi aturan dengan jumlah rules 162 :

Tabel 2. 10 Rule keputusan *Fuzzy*

Rules	Waktu	Daya	KWH	Biaya	Jumlah Perangkat	Jenis Output
1	Cepat	Jika Rendah	Jika RENDAH	Jika Normal	Jika Sedikit	Maka RENDAH
2	Cepat	Jika Normal	Jika RENDAH	Jika Normal	Jika Sedikit	Maka RENDAH
3	Cepat	Jika Tinggi	Jika RENDAH	Jika Normal	Jika Sedikit	Maka RENDAH
4	Cepat	Jika Rendah	Jika SEDANG	Jika Normal	Jika Sedikit	Maka RENDAH
5	Cepat	Jika Normal	Jika SEDANG	Jika Normal	Jika Sedikit	Maka RENDAH
6	Cepat	Jika Tinggi	Jika SEDANG	Jika Normal	Jika Sedikit	Maka RENDAH
7	Cepat	Jika Rendah	Jika TINGGI	Jika Normal	Jika Sedikit	Maka RENDAH
8	Cepat	Jika Normal	Jika TINGGI	Jika Normal	Jika Sedikit	Maka RENDAH
9	Cepat	Jika Tinggi	Jika TINGGI	Jika Normal	Jika Sedikit	Maka RENDAH

10	Normal	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
11	Normal	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
12	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
13	Normal	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
14	Normal	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
15	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
16	Normal	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	SEDANG
17	Normal	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	SEDANG
18	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	SEDANG
19	Lama	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
20	Lama	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
21	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Sedikit	Maka	RENDAH
22	Lama	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
23	Lama	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
24	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Sedikit	Maka	SEDANG
25	Lama	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	TINGGI
26	Lama	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	TINGGI
27	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Sedikit	Maka	TINGGI
28	Cepat	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
29	Cepat	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
30	Cepat	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
31	Cepat	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	RENDAH
32	Cepat	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	RENDAH
33	Cepat	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	RENDAH
34	Cepat	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	RENDAH
35	Cepat	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	RENDAH
36	Cepat	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	RENDAH
37	Normal	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
38	Normal	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
39	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
40	Normal	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG

41	Normal	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG
42	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG
43	Normal	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	SEDANG
44	Normal	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	SEDANG
45	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	SEDANG
46	Lama	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
47	Lama	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
48	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Sedang	Maka	RENDAH
49	Lama	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG
50	Lama	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG
51	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Sedang	Maka	SEDANG
52	Lama	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	TINGGI
53	Lama	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	TINGGI
54	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Sedang	Maka	TINGGI
55	Cepat	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
56	Cepat	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
57	Cepat	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
58	Cepat	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	RENDAH
59	Cepat	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	RENDAH
60	Cepat	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	RENDAH
61	Cepat	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	RENDAH
62	Cepat	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	RENDAH
63	Cepat	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	RENDAH
64	Normal	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
65	Normal	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
66	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
67	Normal	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
68	Normal	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
69	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
70	Normal	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	SEDANG
71	Normal	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	SEDANG

72	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	SEDANG
73	Lama	Jika	Rendah	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
74	Lama	Jika	Normal	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
75	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Normal	Jika	Banyak	Maka	RENDAH
76	Lama	Jika	Rendah	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
77	Lama	Jika	Normal	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
78	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Normal	Jika	Banyak	Maka	SEDANG
79	Lama	Jika	Rendah	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	TINGGI
80	Lama	Jika	Normal	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	TINGGI
81	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Normal	Jika	Banyak	Maka	TINGGI
82	Cepat	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
83	Cepat	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
84	Cepat	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
85	Cepat	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
86	Cepat	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
87	Cepat	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
88	Cepat	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
89	Cepat	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
90	Cepat	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
91	Normal	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
92	Normal	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
93	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
94	Normal	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
95	Normal	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
96	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
97	Normal	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
98	Normal	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
99	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
100	Lama	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
101	Lama	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH
102	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedikit	Maka	RENDAH

103	Lama	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
104	Lama	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
105	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedikit	Maka	SEDANG
106	Lama	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	TINGGI
107	Lama	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	TINGGI
108	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedikit	Maka	TINGGI
109	Cepat	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
110	Cepat	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
111	Cepat	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
112	Cepat	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
113	Cepat	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
114	Cepat	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
115	Cepat	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
116	Cepat	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
117	Cepat	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
118	Normal	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
119	Normal	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
120	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
121	Normal	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
122	Normal	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
123	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
124	Normal	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
125	Normal	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
126	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
127	Lama	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
128	Lama	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
129	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Sedang	Maka	RENDAH
130	Lama	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
131	Lama	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
132	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Sedang	Maka	SEDANG
133	Lama	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	TINGGI

134	Lama	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	TINGGI
135	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Sedang	Maka	TINGGI
136	Cepat	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
137	Cepat	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
138	Cepat	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
139	Cepat	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
140	Cepat	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
141	Cepat	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
142	Cepat	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
143	Cepat	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
144	Cepat	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
145	Normal	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
146	Normal	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
147	Normal	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
148	Normal	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
149	Normal	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
150	Normal	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
151	Normal	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
152	Normal	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
153	Normal	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
154	Lama	Jika	Rendah	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
155	Lama	Jika	Normal	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
156	Lama	Jika	Tinggi	Jika	RENDAH	Jika	Tinggi	Jika	Banyak	Maka	RENDAH
157	Lama	Jika	Rendah	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
158	Lama	Jika	Normal	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
159	Lama	Jika	Tinggi	Jika	SEDANG	Jika	Tinggi	Jika	Banyak	Maka	SEDANG
160	Lama	Jika	Rendah	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	TINGGI
161	Lama	Jika	Normal	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	TINGGI
162	Lama	Jika	Tinggi	Jika	TINGGI	Jika	Tinggi	Jika	Banyak	Maka	TINGGI

#### d. Defuzzifikasi

Input dari proses defuzzifikasi adalah suatu bilangan kabur yang

diperoleh dari komposisi aturan-aturan *Fuzzy*, sedangkan output yang dihasilkan merupakan bilangan pada domain himpunan kabur tersebut. Sehingga jika diberikan suatu himpunan kabur dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai output. Dalam metode Sugeno, defuzzifikasi dilakukan dengan cara mencari nilai rata-rata terbobot (*weight average*).

Metode yang digunakan untuk deFuzzyifikasi yaitu weighted average, yang menghitung rata-rata dari semua inputan. Weighted average adalah salah satu metode deFuzzyifikasi yang dapat digunakan dalam sistem kontrol *Fuzzy*. Dalam metode ini, setiap nilai keanggotaan (membership value) dipertimbangkan dengan menggunakan bobot tertentu dan dijumlahkan untuk menghasilkan nilai tegas (crisp value) yang merepresentasikan output sistem.

Pada tahap deFuzzyifikasi, output *Fuzzy* dari sistem (dalam bentuk himpunan *Fuzzy*) dikonversi menjadi nilai tegas yang dapat digunakan untuk analisis penggunaan listrik. Salah satu cara untuk melakukan hal ini adalah dengan menggunakan metode weighted average.

Pada landasan dari yang sudah dijelaskan diatas, data yang sudah dijabarkan dan didapatkan dari beberapa jurnal akan diolah dengan pengkodean yang ada pada bab 4 yaitu pembahasan pengkodean *Fuzzy Sugeno* dengan data atau variebel pada landasan teori 2.7.2. data tersebut untuk mendapatkan keputusan hasil akhir berupa grafik atau skala yang dihasilkan dari proses atau tahapan tahapan *Fuzzy Sugeno*.

Dari landasan teori tahapan *Fuzzy* sudah dijabarkan dibagian 2.7.2 peneliti mengambil data dari beberapa jurnal untuk mendapatkan variabel dan himpunan *Fuzzy* yang sudah diterbitkan oleh(Suprapto & Simanjuntak, 2020). Dari hasil

tersebut kami dapat data dan kami olah Kembali sesuai dengan fungsi atau kegunaan yang kami butuhkan.

## 2.10 ESP8266

Dalam jurnal (Pela & Pramudita, 2021)ESP8266 merupakan modul mikrokontroler yang berfungsi untuk konektivitas jaringan Wifi antara mikrokontroler itu sendiri dengan jaringan Wifi. NodeMCU berbasis bahasa



### 2.10.1 Karakteristik ESP8266

Gambar 2. 17 ESP8266 (Sumber pitco robo)

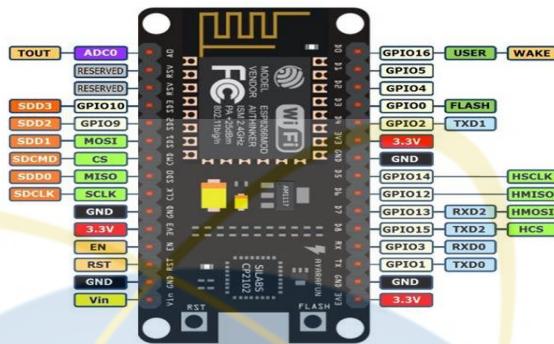
Adapun karakteristik dari NodeMCU dapat dilihat dari tabel berikut

Spesifikasi	NodeMCU
Type	ESP-12E
USB port	Micro Usb
GPIO Pin	13
ADC	1 pin (10 bit)
Usb to serial Converter	CH340G
Power Input	5 Vdc
Ukuran Module	57 x 30 mm

(Yoga, 2020):

Tabel 2. 11 Karakteristik esp

Penempatan pin untuk NodeMCU adalah sebagai berikut :



Gambar 2. 18 PIN NodeMCU ESP8266 (Sumber : smartkits Blog)

Keterangan:

- a. Micro-USB : Pasti semuanya sudah tau bagian ini ya. Fungsinya sebagai power yang dapat terhubung dengan USB port. Selain itu, biasanya juga digunakan untuk melakukan pengiriman sketch atau memantau data serial dengan serial monitor di aplikasi Arduino IDE.
- b. 3.3V : Digunakan sebagai tegangan untuk device lainnya. ada 3 tempat untuk 3.3v. Biasanya juga dituliskan hanya 3V (Sebenarnya tetap 3,3V)
- c. GND : Ground. Sebagai tegangan 0 atau nilai negatif untuk mengalirkan arus.
- d. Vin : Sebagai External Power yang akan mempengaruhi Output dari seluruh pin. Cara menggunakannya yaitu dengan menghubungkannya dengan tegangan 7 hingga 12volt.
- e. EN, RST : Pin yang digunakan untuk reset program di mikrokontroler.
- f. A0 : Analog pin, digunakan untuk membaca input secara analog.

- g. GPIO 1 – GPIO 16 : Pin yang dapat digunakan sebagai input dan output. Pin ini dapat melakukan pembacaan dan pengiriman data secara analog juga.
- h. SD1,CMD, SD0,CLK : SPI Pin untuk komunikasi SPI (Serial Peripheral Interface) dimana kita akan menggunakan clock untuk sinkronisasi deteksi bit pada receiver.
- i. TXD0, RXD0,TXD2,RXD2 : Sebagai interface UART, Pasangannya adalah TXD0 dengan RXD0 dan TXD2 dengan RXD2. TXD1 digunakan untuk upload firmware/program.
- j. SDA, SCL (I2C Pins) : Digunakan untuk device yang membutuhkan I2C.

### **2.10.2 Komponen utama ESP8266**

Dalam ESP8266 (Indobot, 2023) memiliki komponen utama yang harus dipahami untuk membuat suatu sistem Monitoring listrik, Berikut komponen utama dari ESP8266 :

- a. Microcontroller

ESP8266 dilengkapi dengan chip microcontroller yang mengontrol semua fungsi dalam modul, termasuk koneksi WiFi, pengiriman dan penerimaan data, dan pemrosesan data. Microcontroller ini memungkinkan ESP8266 untuk diatur dan dikontrol melalui kode pemrograman.

- b. WiFi

ESP8266 dilengkapi dengan antena WiFi dan modul untuk terhubung ke jaringan WiFi. WiFi memungkinkan ESP8266 untuk mengirim dan menerima data secara nirkabel, sehingga dapat

digunakan dalam berbagai proyek IoT seperti kontrol pintu dan lampu.

#### c. Arduino IDE

ESP8266 dapat diprogram menggunakan Arduino IDE, yang merupakan integrated development environment (IDE) yang memungkinkan pengembang untuk membuat dan mengunggah kode program ke modul. Arduino IDE adalah alat yang populer digunakan dalam pengembangan proyek IoT karena mudah digunakan dan memiliki banyak komunitas pengguna yang aktif.

#### d. AT Command

ESP8266 juga dapat dikontrol menggunakan AT command, yang adalah serangkaian perintah yang dikirim ke modul melalui koneksi serial untuk mengkonfigurasi modul dan memungkinkan pengiriman dan penerimaan data.

#### e. Flash Memory

ESP8266 memiliki memori flash internal yang memungkinkan pengembang untuk menyimpan kode program dan data. Memori flash dapat diakses menggunakan Arduino IDE atau AT command untuk memprogram modul dan menyimpan data.

Pada penelitian ini menggunakan ESP8266 karena sering digunakan dalam pengembangan proyek *Internet of Things* (IoT) karena kemampuannya untuk terhubung ke jaringan WiFi dan mengirim dan menerima data nirkabel dan dikontrol melalui kode pemrograman atau *AT command*, *AT command* adalah serangkaian perintah yang dikirim ke modul melalui koneksi serial untuk mengkonfigurasi modul dan memungkinkan pengiriman dan penerimaan data.

## 2.11 PZEM 004T



Gambar 2. 19 PZEM-004T (Sumber : Vas Labs)

Modul PZEM-004T adalah sebuah modul sensor multifungsi yang berfungsi untuk mengukur daya, tegangan, arus dan energi yang terdapat pada sebuah aliran listrik. Modul ini sudah dilengkapi sensor tegangan dan sensor arus yang sudah terintegrasi. Dalam penggunaannya, alat ini khusus untuk penggunaan dalam ruangan (*indoor*) dan beban yang terpasang tidak diperbolehkan melebihi daya yang sudah ditetapkan (Dwisyaputra et al., 2021)

Menurut (Widiasari, 2020)PZEM memiliki fitur atau komponen untuk mendapatkan hasil sensor listrik, dengan sebagai berikut :

### 1. Metode Pengukuran

PZEM004T menggunakan metode pengukuran non-intrusif, artinya tidak perlu memotong atau membuka sirkuit listrik untuk mengukur arus dan tegangan. Metode ini menggunakan perangkat magnetik untuk mengukur arus dan divider tegangan untuk mengukur tegangan, sehingga aman dan mudah digunakan.

### 2. Range Pengukuran

PZEM004T memiliki rentang pengukuran arus AC hingga 100A dan tegangan AC hingga 220V. Rentang pengukuran ini memungkinkan pengukuran daya listrik pada berbagai jenis peralatan elektronik yang berbeda.

### 3. Output Data

PZEM004T memiliki output data yang dapat diakses melalui komunikasi serial TTL. Data yang diberikan antara lain nilai arus, tegangan, daya aktif, daya reaktif, dan faktor daya. Output data ini dapat digunakan dalam pengembangan proyek-proyek monitoring daya listrik.

### 4. Kalibrasi

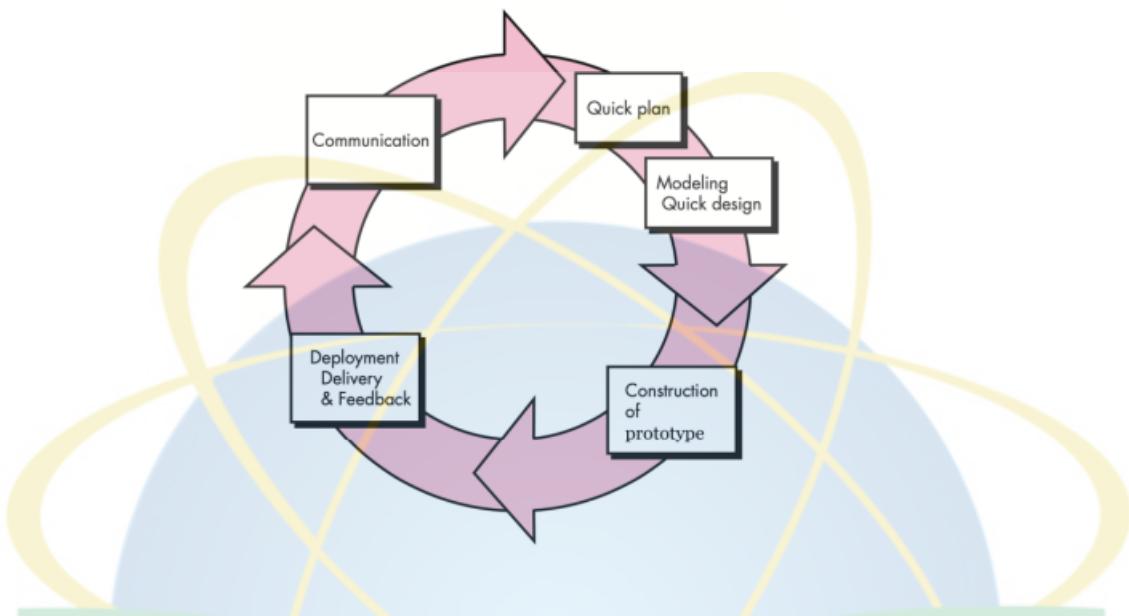
PZEM004T memungkinkan kalibrasi untuk meningkatkan akurasi pengukuran. Ada dua jenis kalibrasi yang dapat dilakukan, yaitu kalibrasi tegangan dan kalibrasi arus. Kalibrasi dilakukan dengan menggunakan potensiometer pada modul PZEM004T.

### 5. Kompatibilitas

PZEM004T dapat digunakan dengan berbagai jenis mikrokontroler seperti Arduino, ESP8266, dan Raspberry Pi. PZEM004T juga kompatibel dengan berbagai macam library atau perpustakaan pemrograman, sehingga memudahkan pengembang dalam mengembangkan proyek-proyek monitoring daya listrik.

Sebagai sensor listrik PZEM004T dapat membantu Peneliti mengembangkan proyek monitoring daya listrik yang efektif dan efisien. PZEM004T memungkinkan pengukuran daya listrik menggunakan metode non-intrusif, memiliki rentang pengukuran yang luas, output data yang mudah diakses, dan dapat dikalibrasi untuk meningkatkan akurasi pengukuran. Selain itu, PZEM004T juga kompatibel dengan berbagai jenis mikrokontroler dan library pemrograman, sehingga memudahkan dalam pengembangan proyek-proyek monitoring daya listrik.

## 2.12 Prototyping



Gambar 2. 20 alur prototipe (Sumber: Metode Prototype)

Menurut (Salamadian, 2021) Prototyping merupakan Teknik pengembangan sistem yang menggunakan prototype untuk menggambarkan sistem, sehingga pengguna atau pemilik sistem mempunyai gambaran pengembangan sistem yang akan dilakukannya. Teknik ini sering digunakan apabila pemilik sistem tidak terlalu menguasai sistem yang akan dikembangkannya, sehingga dia memerlukan gambaran dari sistem yang akan dikembangkannya tersebut. Hal ini sejalan dengan pendapat dari (Pressman, 2010) Yang mengatakan, seringkali pelanggan mendefinisikan serangkaian tujuan umum untuk perangkat lunak, tetapi tidak mendefinisikan persyaratan terperinci untuk fungsi dan fitur. Dengan prototype pihak klien dapat berdiskusi langsung dengan tim pengembang (*developer*) untuk menyamakan persepsi atau pemahaman terhadap sistem yang akan dibuat. Sehingga nantinya tidak terjadi kesalahpahaman dalam proses pembuatan sistem/aplikasi/perangkat lunak. Beberapa tahapan dari (Pressman, 2010) yaitu :

- Tahap Komunikasi

Tahap ini melibatkan komunikasi yang intens antara tim pengembang perangkat lunak dan pemangku kepentingan (stakeholders). Tujuan dari tahap ini adalah untuk memahami kebutuhan dan harapan pemangku kepentingan terkait dengan produk yang akan dikembangkan. Tim pengembang perlu menggali informasi yang cukup tentang tujuan, fungsionalitas, dan karakteristik sistem yang diinginkan.

b. Tahap perencanaan cepat(*Quick planning*)

Tahap ini dilakukan untuk merencanakan kegiatan pengembangan perangkat lunak dengan pendekatan yang cepat dan efisien. Perencanaan cepat melibatkan penentuan tujuan, jadwal, dan sumber daya yang dibutuhkan dalam pengembangan prototipe. Tim pengembang juga perlu mengidentifikasi prioritas-fitur dan risiko yang mungkin muncul selama pengembangan.

c. Tahap pemodelan cepat (*Quick Modeling*)

Tahap ini melibatkan pembuatan model awal (prototype) yang mewakili fungsionalitas dan tampilan antarmuka pengguna sistem yang akan dikembangkan. Model ini dapat berupa sketsa, diagram, atau prototipe perangkat lunak yang sederhana. Tujuan dari tahap ini adalah untuk memvalidasi konsep, mengumpulkan umpan balik, dan memastikan pemahaman yang sama antara tim pengembang dan pemangku kepentingan.

d. Tahap kontruksi (*Construction*)

Tahap ini melibatkan pembangunan prototipe perangkat lunak berdasarkan model yang telah dirancang. Tim pengembang melakukan implementasi fungsionalitas dan fitur-fitur yang telah ditetapkan. Prototipe dibangun dengan menggunakan bahasa pemrograman, framework, atau alat pengembangan perangkat lunak yang sesuai. Selama tahap ini, prototipe diperbaiki dan ditingkatkan secara bertahap berdasarkan umpan balik dari pemangku kepentingan.

#### e. Tahap Evaluasi

Tahap ini melibatkan evaluasi terhadap prototipe yang telah dibangun. Pemangku kepentingan diajak untuk menguji prototipe, memberikan umpan balik, dan mengevaluasi kelayakan dan keefektifan solusi yang diusulkan. Evaluasi dilakukan dengan membandingkan hasil prototipe dengan kebutuhan dan tujuan yang telah ditetapkan sebelumnya. Hasil evaluasi digunakan untuk memperbaiki prototipe, mengidentifikasi kekurangan, dan memutuskan langkah selanjutnya dalam pengembangan perangkat lunak.

Menurut (dewaweb, 2022)terdapat kelebihan dan kekurangan pada Teknik pengembangan prototyping yaitu:

1. Menghemat waktu pengembangan
2. Menghemat biaya pengembangan
3. Pengguna atau pemilik sistem ikut terlibat dalam pengembangan, sehingga kemungkinan-kemungkinan terjadinya kesalah pahaman dalam sistem bisa diminimalisir
4. Implementasi akan menjadi mudah, karena pengguna atau pemilik sistem sudah mempunyai gambaran tentang sistem
5. Kualitas sistem yang dihasilkan baik
6. Memungkinkan tim pengembang sistem memprediksi dan memperkirakan pengembangan-pengembangan sistem selanjutnya.

Sedangkan kelemahannya adalah, Pengguna atau pemilik sistem bisa terus menerus menambah kompleksitas sistem hingga sistem menjadi sangat kompleks, hal ini bisa menyebabkan pengembang meninggalkan pekerjaanya sehingga sistem yang dikerjakan tidak akan pernah terselesaikan.

### 2.13 Structured Analysis with Real-time (SA-RT)

Analisis Terstruktur untuk sistem, atau SA-RT, adalah notasi desain grafis yang berfokus pada analisis perilaku fungsional dan aliran informasi melalui suatu sistem. SA-RT, yang pada gilirannya merupakan penyempurnaan dari metode analisis struktural yang awalnya diperkenalkan oleh Douglass Ross dan dipopulerkan oleh Tom DeMarco pada tahun tujuh puluhan, pertama kali diperkenalkan oleh Ward dan Mellor pada tahun 1985 dan setelah itu telah disempurnakan dan dimodifikasi oleh peneliti lain, satu contoh terkenal adalah proposal Hatley dan Pirbhai (Vicky, 2020)

SART adalah salah satu metode desain yang paling sering digunakan dalam aplikasi berorientasi teknis dan real-time yang diadopsi oleh berbagai Case-Tools. Dan terbagi dalam 4 modul:

1. Context Diagram (CD).
2. Data Flows Diagram (DFD).
3. Control Flows Diagram (CFD).
4. State Transition Diagram (STD).

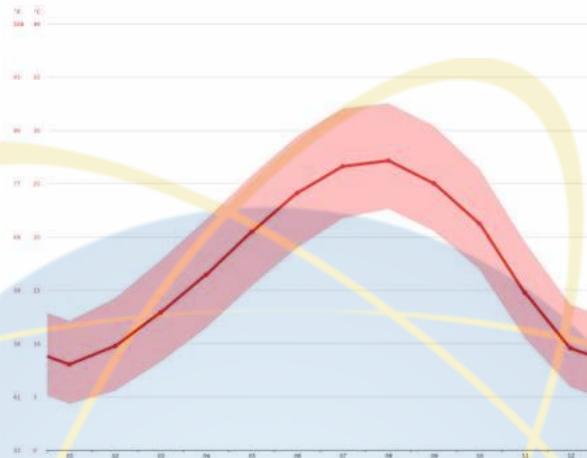
Dari 4 modul diatas peniliti menggunakan Data flow diagram untuk digunakan menganalisis sistem mekanisme dari penelitian secara detail.

### 2.14 Data Visualization

Menurut (Perdana, 2021) data *visualization* atau visualisasi data adalah proses mengubah data jadi lebih sederhana untuk ditampilkan. Bentuk data biasanya dalam grafis atau visual yang menarik. bidang interdisipliner yang berhubungan dengan representasi grafis dari data dan informasi . Ini adalah cara komunikasi yang sangat efisien ketika data atau informasi sangat banyak, misalnya rangkaian waktu (Wikipedia, n.d.)

Salah satu jenis Data Visualization adalah *time-series* yang masuk dalam Data temporal yang mempelajari deret waktu, baik dari segi teori yang menaunginya

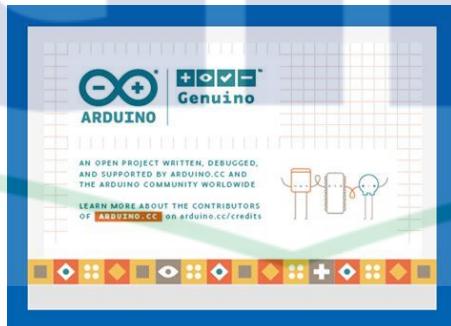
maupun untuk membuat peramalan (prediksi) yang nilai di waktu mendatang berdasarkan peristiwa yang telah terjadi.(Tanto & Riswanto, 2022)



Gambar 2. 21 contoh grafik Time series grarfik (Sumber : R-bloogger)

Dalam konteks sistem monitoring listrik berbasis Android, teori landasan Data Visualization ini dapat digunakan untuk membangun aplikasi Android yang dapat memvisualisasikan data monitoring listrik dalam bentuk yang mudah dipahami oleh pengguna. Grafik dapat digunakan untuk menampilkan perubahan daya listrik dari waktu ke waktu dengan *output Fuzzy Logic* yang sudah dirancang didalam prosesnya.

## 2.15 Arduino IDE



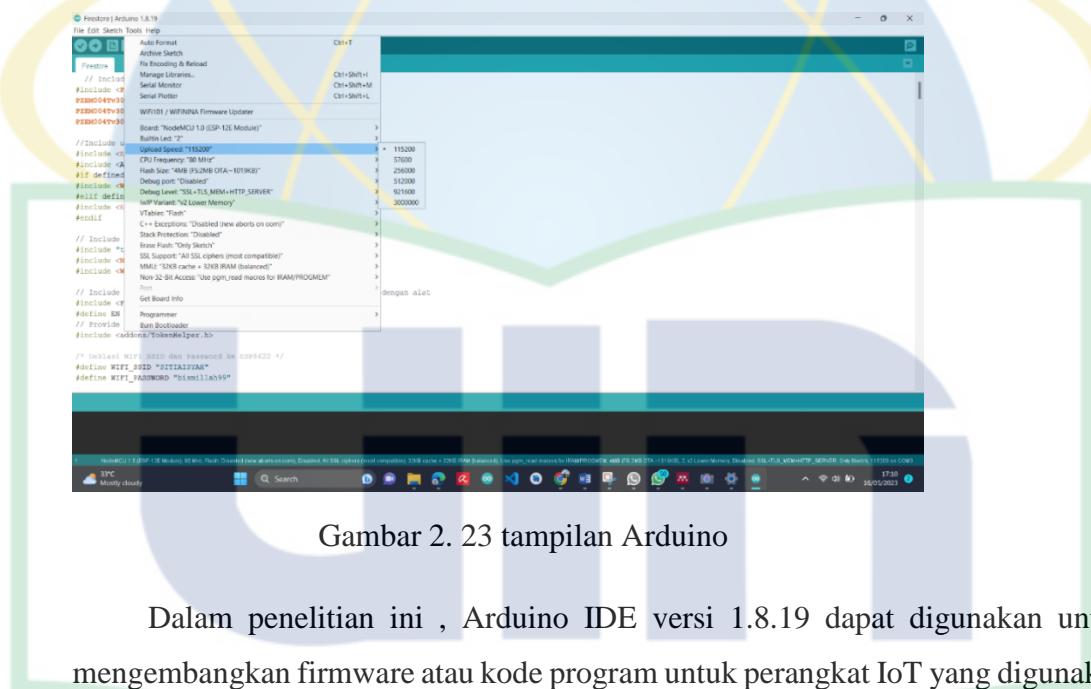
Gambar 2. 22 Arduino IDE (Sumber : Mspoweruser)

Arduino IDE merupakan perangkat lunak multiplatform yang digunakan untuk membuat dan upload kode program ke dalam board arduino. Arduino IDE memiliki banyak fungsi dan perpustakaan yang dapat digunakan untuk

mengembangkan berbagai proyek elektronik, termasuk proyek monitoring listrik berbasis Android, Data yang dihasilkan oleh sensor ini kemudian dapat dikirim ke papan mikrokontroler Arduino yang terhubung ke jaringan Wi-Fi.(Dwisisputra et al., 2021).

Pada penelitian ini Arduino IDE dapat digunakan untuk mengembangkan kode program untuk papan mikrokontroler ESP8266. Kode program harus mampu membaca data dari sensor arus dan tegangan, memproses data menggunakan algoritma *Fuzzy Logic Sugeno*, dan mengirimkan data ke *Firebase*.

Berikut tampilan dari Arduino IDE :



Gambar 2. 23 tampilan Arduino

Dalam penelitian ini , Arduino IDE versi 1.8.19 dapat digunakan untuk mengembangkan firmware atau kode program untuk perangkat IoT yang digunakan dalam sistem monitoring daya listrik. Yang memiliki fungsi mengatur dan mengontrol sensor, memproses data, dan berkomunikasi dengan perangkat lain.

Arduino IDE juga mengatur komunikasi antara perangkat IoT yang mengumpulkan data daya listrik dengan *Firebase*. Yaitu mengirim data dari perangkat IoT ke *Firebase* Realtime Database atau Firestore untuk penyimpanan dan analisis lebih lanjut. Pemrograman komunikasi nirkabel, Perangkat komunikasi yang diterapkan di Arduino IDE yaitu memprogram komunikasi nirkabel yaitu

WiFi dengan tujuan untuk mengatur protokol komunikasi yang diperlukan berkomunikasi dengan aplikasi Android dan *Firebase*.

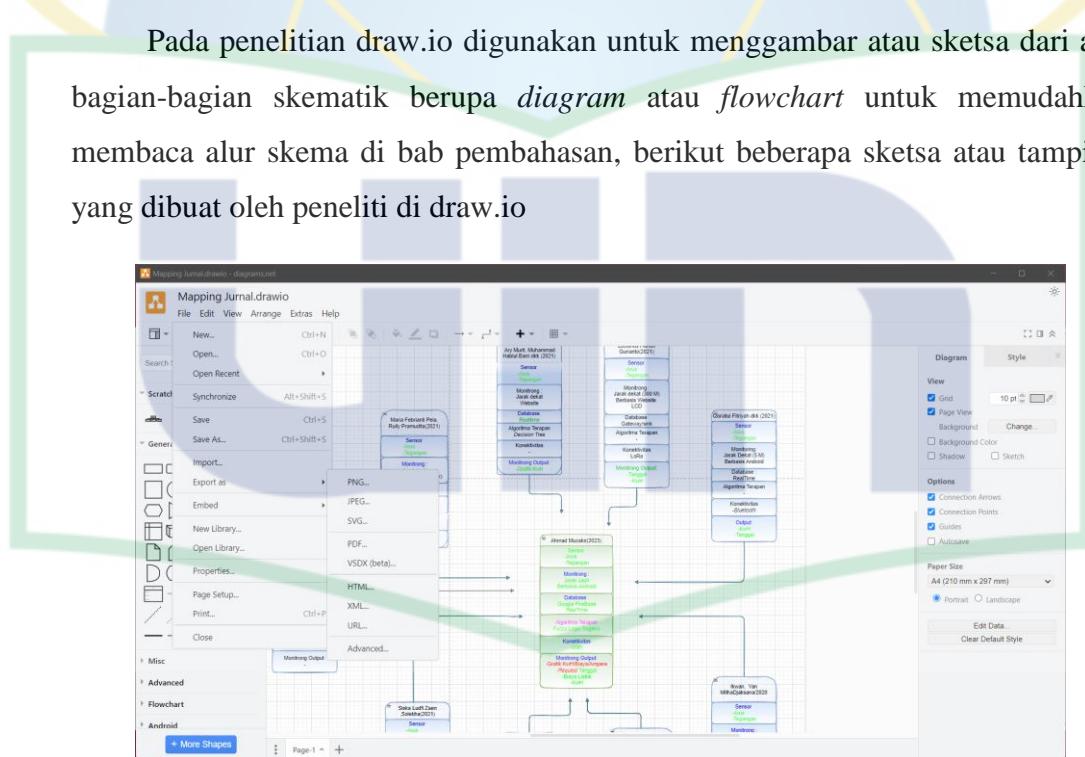
## 2.16 Draw.Io



Gambar 2. 24 Icon draw io (sumber: draw.io)

Draw.io adalah sebuah website yang didesain khusus untuk menggambarkan diagram secara online. Semua fitur yang ada pada situs ini bisa kalian nikmati hanya dengan bermodalkan browser yang mendukung HTML 5 (HIMASIS, 2022).

Pada penelitian draw.io digunakan untuk menggambar atau sketsa dari alur bagian-bagian skematik berupa *diagram* atau *flowchart* untuk memudahkan membaca alur skema di bab pembahasan, berikut beberapa sketsa atau tampilan yang dibuat oleh peneliti di draw.io



Gambar 2. 25 Cara kerja Draw.Io

Dalam penelitian, Contoh pada gambar diatas salah satu hasil dari draw.io yang peneliti buat untuk mapping studi literature sejenis dan Draw.io memiliki fitur seperti Diagram Arsitektur Sistem yang dapat menggunakan untuk membuat diagram arsitektur sistem yang menggambarkan komponen-komponen yang terlibat dalam sistem monitoring daya listrik. Misalnya, dapat membuat diagram yang menunjukkan hubungan antara perangkat IoT, *Firebase*, aplikasi Android, dan komponen lainnya. Diagram Alur Draw.io dapat digunakan untuk membuat diagram alur kerja yang menggambarkan langkah-langkah atau proses yang terjadi dalam sistem monitoring daya listrik. Misalnya, dapat membuat diagram yang menunjukkan langkah-langkah pengumpulan data, pemrosesan data menggunakan algoritma *Fuzzy Logic Sugeno*, dan pengiriman data ke *Firebase*. Diagram Komunikasi yang dapat menggunakan Draw.io untuk membuat diagram komunikasi yang menggambarkan bagaimana komponen-komponen dalam sistem berkomunikasi satu sama lain. Misalnya, dapat membuat diagram yang menunjukkan komunikasi antara perangkat IoT dengan aplikasi Android, komunikasi antara perangkat IoT dengan *Firebase*, atau komunikasi antar komponen lainnya.

Draw.io menyediakan berbagai bentuk, ikon, dan alat untuk membantu membuat diagram dengan mudah. memiliki menambahkan elemen-elemen grafis seperti kotak, panah, teks, dan simbol lainnya. Setelah selesai membuat diagram, Anda dapat menyimpannya dalam berbagai format file, seperti PNG, JPEG, PDF, atau menyimpannya langsung di cloud storage.

## 2.17 Visual studio code

*Visual Studio Code* adalah sebuah code editor yang bisa Anda gunakan di semua desktop. Software ini juga bisa Anda gunakan di semua operating system, seperti Windows, Linux, Mac. Code editor ini dibuat dan dikembangkan oleh Microsoft. Jika dilihat dari segi fungsi, maka *text editor* yang satu ini merupakan sebuah software yang cukup *powerfull* namun tetap ringan ketika digunakan. Software ini juga cukup penting untuk dipakai untuk membuat dan mengedit source

code dari berbagai bahasa pemrograman, seperti python untuk *Fuzzy logic* (Makinrajin, 2022).

Proses dan mekanisme (Arifudin, 2022)Visual Studio Code " melibatkan beberapa tahapan berikut:

1. Instalasi Visual Studio Code:

- Mengunduh dan menginstal Visual Studio Code dari situs resminya (<https://code.visualstudio.com/>).
- Memilih versi yang sesuai dengan sistem operasi yang digunakan.

2. Konfigurasi Lingkungan Pengembangan:

- Menginstal ekstensi dan plugin yang dibutuhkan untuk pengembangan aplikasi Android, *Firebase*, dan bahasa pemrograman yang digunakan (misalnya, Python).
- Mengkonfigurasi preferensi dan pengaturan sesuai dengan kebutuhan pengembangan.

3. Pembuatan Proyek:

- Membuat folder proyek baru di dalam Visual Studio Code.
- Membuat atau mengimpor file-file yang diperlukan untuk implementasi aplikasi sistem monitoring daya listrik menggunakan *Firebase*, Android, dan algoritma *Fuzzy Logic Sugeno*.

4. Pemrograman dan Debugging:

- Menulis kode program menggunakan bahasa pemrograman yang relevan (misalnya, Python untuk implementasi algoritma *Fuzzy Logic Sugeno*).

- Menggunakan fitur-fitur Visual Studio Code seperti syntax highlighting, code completion, dan code formatting untuk membantu dalam penulisan kode yang lebih efisien dan akurat.
- Menggunakan fitur debugging Visual Studio Code untuk melacak dan memperbaiki kesalahan dalam kode.

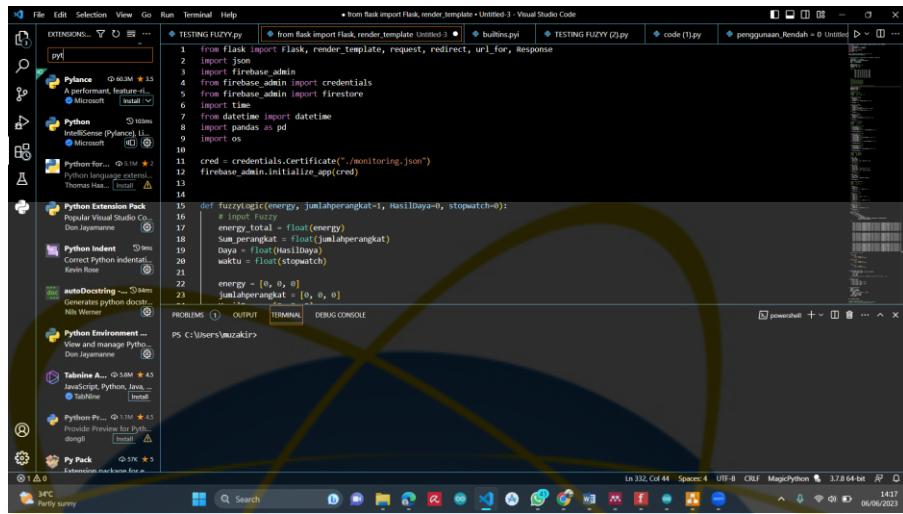
#### 5. Integrasi dengan Platform Lain:

- Menggunakan ekstensi dan plugin yang sesuai untuk mengintegrasikan Visual Studio Code dengan platform *Firebase* dan Android.
- Melakukan konfigurasi yang diperlukan untuk menghubungkan aplikasi dengan *Firebase* Realtime Database dan API Android.

#### 6. Peluncuran Aplikasi dan Uji Coba:

- Menggunakan fitur peluncuran dan emulasi Visual Studio Code untuk menjalankan aplikasi Android secara langsung di emulator atau perangkat fisik.
- Melakukan pengujian aplikasi untuk memastikan bahwa sistem monitoring daya listrik berfungsi sesuai yang diharapkan dan berintegrasi dengan *Firebase* dengan baik.

Peneliti menggunakan *visual code version 1.78.2* untuk meimplementasikan proses *Fuzzy logic* dengan menggunakan bahasa python yang sudah terinstalls di VSC (*Visual studio code*) dan juga memberikan data olahan yang sudah ada di *firebase* lalu dikirimkan ke tampilan android dengan bentuk tabel dan garfik. *Visual Studio Code* dapat digunakan untuk mengatur proyek, menulis kode Python, melakukan debugging, dan mengintegrasikan dengan platform *Firebase* dan Android untuk mengembangkan sistem monitoring daya listrik yang efektif.



Gambar 2. 26 Tampilan Visual Studio Code

## 2.18 Fritzing

Menurut (isaac, 2020)Fritzing adalah perangkat lunak sumber terbuka dirancang khusus untuk yang perlu membuat proyek elektronik, terutama perangkat keras gratis, dan yang tidak memiliki akses ke materi yang diperlukan. Itu juga dapat digunakan untuk membuat desain, mengambil contoh untuk tutorial, dll. Selain itu, alat ini memiliki komunitas hebat di belakangnya yang selalu memperbaruiinya atau bersedia membantu jika memiliki masalah. Bahkan dapat menjadi alat yang hebat. untuk pengguna yang ingin berbagi dan mendokumentasikan prototipe, dan bahkan untuk para profesional.

Fritzing tersedia di macOS, Linux dan Windows. Inisiatif ini dikembangkan oleh Potsdam University of Applied Sciences, dan dirilis di bawah lisensi GPL 3.0 atau lebih tinggi, sedangkan gambar komponen yang dapat digunakan dilisensikan di bawah lisensi Creative Commons CC BY-SA 3.0.

Proses dan mekanisme Frizting dalam membuat desain prototype sistem monitoring listrik meliputi beberapa tahapan berikut:

### a. Instalasi dan Konfigurasi:

- Mengunduh dan menginstal aplikasi Frizting dari situs resminya (<https://fritzing.org/>).

- Mengkonfigurasi preferensi dan pengaturan sesuai dengan kebutuhan pengembangan.

b. Desain Skema Perangkat:

- Membuka aplikasi Fritzing dan memulai dengan membuat proyek baru.
- Menggunakan antarmuka grafis Fritzing untuk merancang skema perangkat yang akan digunakan dalam sistem monitoring daya listrik.
- Menambahkan komponen seperti Arduino, sensor daya, dan komponen lain yang diperlukan untuk pengukuran daya listrik.

c. Koneksi dan Pengaturan:

- Menghubungkan komponen perangkat secara visual dengan menggambarkan koneksi antara mereka menggunakan kabel virtual.
- Mengkonfigurasi setiap komponen dengan memasukkan parameter dan pengaturan yang diperlukan untuk sistem monitoring daya listrik.

d. Validasi Skema:

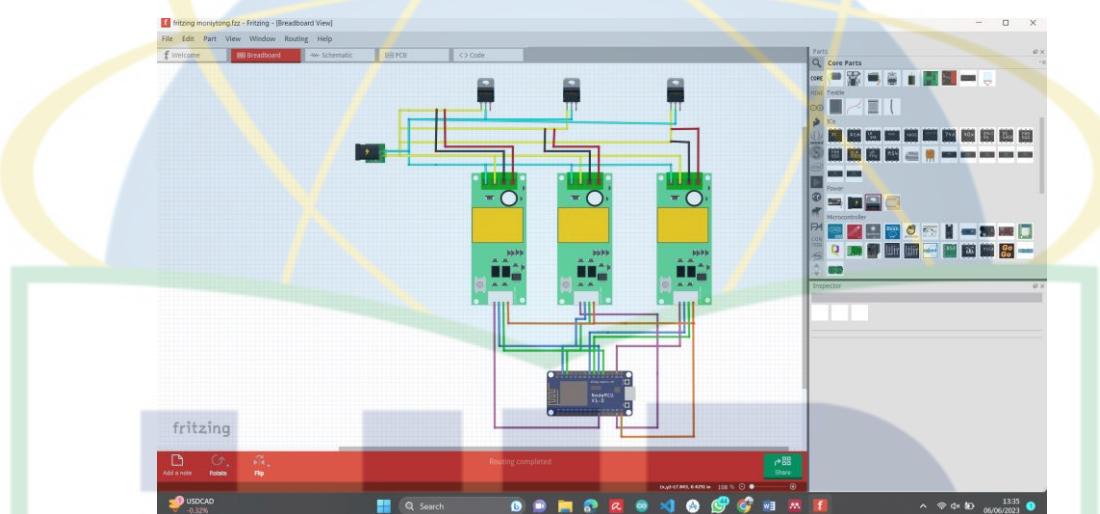
- Melakukan validasi skema untuk memastikan bahwa koneksi antara komponen perangkat telah dibuat dengan benar.
- Memeriksa kesesuaian skema dengan spesifikasi sistem monitoring daya listrik yang diinginkan.

e. Pembuatan Dokumentasi:

- Menggunakan fitur Fritzing untuk menghasilkan gambar dan dokumentasi skema perangkat.

- Menyimpan dokumentasi dan gambar dalam format yang sesuai untuk digunakan dalam skripsi.

Fritzing adalah alat yang berguna untuk merancang dan memvisualisasikan skema perangkat keras. Dalam penelitian ini, Fritzing dapat digunakan untuk merancang skema perangkat yang akan digunakan dalam sistem monitoring daya listrik. Skema ini dapat digunakan sebagai panduan untuk merakit perangkat keras yang sesuai dengan spesifikasi yang diinginkan, berikut tampilan dari pembuatan skema penelitian dari Fritzing versi 0.9.10 :



Gambar 2. 27 Tampilan Fritzing penelitian

## 2.19 Python

Menurut (hosting, 2021) Bahasa pemrograman python adalah salah satu jenis yang banyak digunakan saat kamu berkuatik dengan data. Namun tidak hanya pada pengolahan data saja, python juga bisa digunakan untuk mengatur infrastruktur *internet of things*.

Sebagai salah satu bahasa pemrograman termudah untuk dipelajari, Python menggunakan sintaksis sederhana dengan banyak kata kunci dalam bahasa Inggris. Python dirancang menjadi bahasa yang ringkas dan mudah dibaca, sehingga lebih user-friendly dibanding bahasa lain seperti C++ dan Haskell. Selain kesederhanaan dan konsistensinya, faktor lain yang berkontribusi pada kemudahan penggunaan

Python adalah sifatnya sebagai bahasa pemrograman yang diterjemahkan (interpreted), tidak seperti bahasa compiled seperti C atau C++. (Faradilla, 2023).

Python dalam skripsi Sistem Monitoring Daya Listrik menggunakan *Firebase* berbasis Android dengan Algoritma *Fuzzy Logic* Sugeno melibatkan beberapa konsep dan fitur dasar dari bahasa pemrograman Python, seperti:

a. Sintaks dan Struktur Program:

- Python memiliki sintaks yang sederhana dan mudah dibaca, membuatnya ideal untuk pemula maupun pengembang berpengalaman.
- Program Python terdiri dari serangkaian pernyataan yang diatur secara hierarkis menggunakan indentasi, bukan tanda kurung atau tanda kurawal.

b. Tipe Data dan Variabel:

- Python mendukung berbagai tipe data seperti angka, string, daftar, tuple, set, dan kamus.
- Variabel digunakan untuk menyimpan data dan dapat diubah nilainya selama program berjalan.

c. Kontrol Alur:

- Python menyediakan struktur kontrol alur seperti pernyataan if-else, perulangan for dan while, dan pernyataan break dan continue.
- Dengan struktur kontrol alur ini, penulis dapat mengatur alur eksekusi program berdasarkan kondisi tertentu.

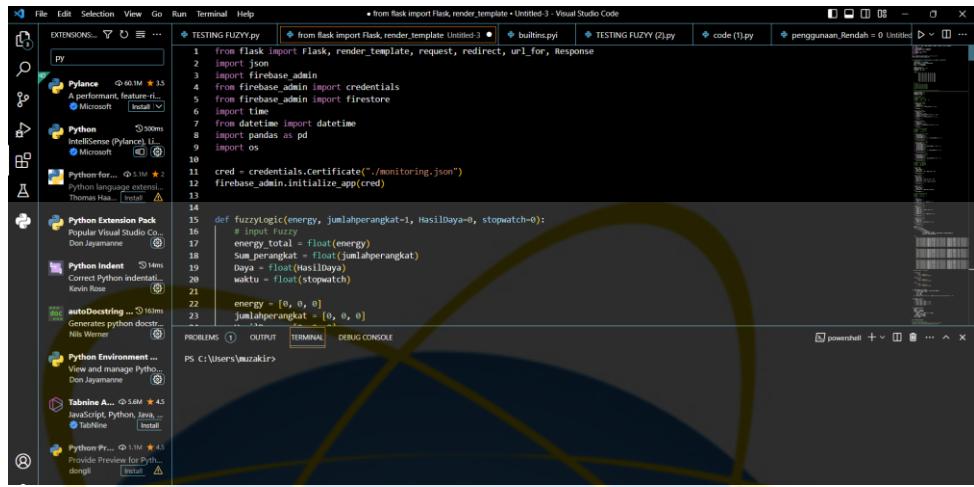
d. Fungsi dan Modul:

- Python memungkinkan definisi dan pemanggilan fungsi, yang memungkinkan pemisahan kode ke dalam unit-unit yang dapat digunakan kembali.
- Modul adalah file Python yang berisi kumpulan fungsi, kelas, dan konstanta terkait yang dapat diimpor dan digunakan dalam program lain.

e. Pustaka dan Modul Eksternal:

- Python memiliki ekosistem yang kaya dengan berbagai pustaka dan modul eksternal yang dapat digunakan untuk memperluas fungsionalitas aplikasi.
- Dalam skripsi ini, penulis mungkin menggunakan pustaka seperti pyrebase, numpy, skFuzzy, dan lainnya untuk mengintegrasikan dengan *Firebase*, melakukan analisis data, dan mengimplementasikan algoritma *Fuzzy Logic Sugeno*.

Dengan menggunakan landasan teori Python, penulis skripsi dapat mengimplementasikan fungsi-fungsi logika, menghubungkan dengan *Firebase*, dan mengelola data daya listrik dengan cara yang efektif dan efisien. Python sebagai bahasa pemrograman yang serbaguna dan mudah dipahami memberikan fleksibilitas dan kekuatan dalam pengembangan aplikasi ini. Berikut tampilan bahasa Python dengan framework dari *Visual Studio Code*:



The screenshot shows the Visual Studio Code interface with several tabs open. The main editor tab contains Python code for a Fuzzy Logic system. The code imports Flask, json, and firebase\_admin, and uses pandas and datetime libraries. It defines a function fuzzy\_logic that takes energy, jumlahperangkat, HasilDaya, and stopwatch as parameters. It calculates total energy, sum perangkat, and daya (HasilDaya). It then initializes a database connection and performs a query. The terminal tab shows a command prompt in PowerShell with the path PS C:\Users\muzahir>.

```

1  from flask import Flask, render_template, request, redirect, url_for, Response
2  import json
3  from firebase_admin import credentials
4  from firebase_admin import firestore
5  import time
6  from datetime import datetime
7  import pandas as pd
8  import os
9
10 cred = credentials.Certificate("./monitoring.json")
11 firebase_admin.initialize_app(cred)
12
13
14
15 def fuzzy_logic(energy, jumlahperangkat, HasilDaya=0, stopwatch=0):
16     # Input Fuzzy
17     energy_total = float(energy)
18     sum_perangkat = float(jumlahperangkat)
19     Daya = float(HasilDaya)
20     waktu = float(stopwatch)
21
22     energy = [0, 0, 0]
23     jumlahperangkat = [0, 0, 0]

```

Gambar 2. 28 Penerapan Python untuk IoT

## 2.20 Flutter

Menurut (Aprilia, 2021)Flutter adalah platform yang digunakan para developer untuk membuat aplikasi multiplatform hanya dengan satu basis coding (*codebase*). Artinya, aplikasi yang dihasilkan dapat dipakai di berbagai platform, baik mobile Android, iOS, web, maupun desktop.

Flutter memiliki dua komponen penting, yaitu, Software Development Kit (SDK) dan juga framework user interface (Ismi, 2021).

- Software Development Kit (SDK) merupakan sekumpulan tools yang berfungsi untuk membuat aplikasi supaya bisa dijalankan di berbagai platform.
- Framework UI merupakan komponen UI, seperti teks, tombol, navigasi, dan lainnya, yang dapat Anda kustomisasi sesuai kebutuhan.

Untuk membuat Flutter dalam Sistem Monitoring Daya Listrik menggunakan *Firebase* berbasis Android dengan Algoritma *Fuzzy Logic* Sugeno melibatkan beberapa aspek berikut:

- a. Flutter Framework: Flutter adalah kerangka pengembangan aplikasi open-source yang dikembangkan oleh Google. Ia menggunakan bahasa

pemrograman Dart untuk membuat antarmuka pengguna yang kaya dan responsif di berbagai platform seperti Android, iOS, dan web.

- b. Widget: Flutter menggunakan konsep "Everything is a widget" di mana setiap elemen dalam antarmuka pengguna, baik itu tombol, teks, gambar, atau tata letak, diwakili oleh widget. Widget dapat dikombinasikan dan disusun dalam pohon widget untuk membangun tampilan yang kompleks.
- c. Integrasi Flutter dengan *Firebase*: Flutter menyediakan paket *Firebase* SDK yang memungkinkan pengembang untuk mengintegrasikan aplikasi Flutter dengan layanan *Firebase*. Ini memudahkan akses dan manipulasi data *Firebase*, serta penggunaan fitur otentikasi dan penyimpanan.
- d. Pengembangan Aplikasi Android: Flutter memungkinkan pengembangan aplikasi Android dengan menggunakan bahasa Dart. Dart adalah bahasa pemrograman modern yang efisien dan mudah dipahami. Pengembang dapat menggunakan Dart untuk mengimplementasikan logika bisnis aplikasi, komunikasi dengan *Firebase*, dan algoritma *Fuzzy Logic Sugeno*.
- e. Tampilan Responsif: Flutter memiliki fitur "Hot Reload" yang memungkinkan pengembang melihat perubahan secara real-time saat mereka mengedit kode. Hal ini memudahkan pengembangan antarmuka pengguna yang responsif dan cepat, sehingga mempercepat siklus pengembangan.

Pada penelitian ini versi yang digunakan ialah Flutter 2.5.1. Mekanisme Flutter pada penelitian melibatkan penggunaan Flutter sebagai framework pengembangan aplikasi mobile. Flutter adalah sebuah framework open-source yang dikembangkan oleh Google untuk membangun antarmuka pengguna (UI) yang kaya dan responsif secara *cross-platform*.

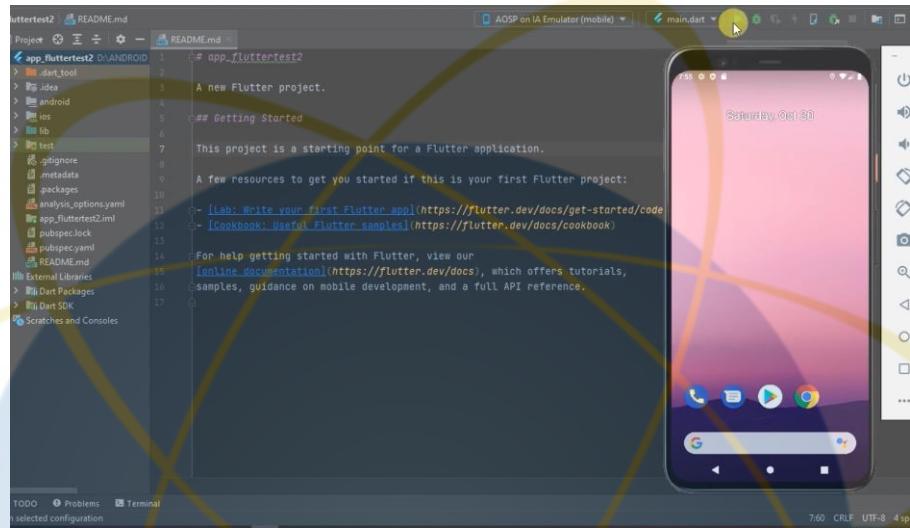
Dalam penelitian ini, Flutter digunakan untuk membangun aplikasi mobile berbasis Android yang bertujuan untuk memonitoring daya listrik. Berikut adalah beberapa mekanisme penggunaan Flutter dalam penelitian tersebut:

1. Pembangunan Antarmuka Pengguna: Flutter memungkinkan untuk membuat antarmuka pengguna yang menarik dan responsif. Peniliti dapat menggunakan berbagai widget dan elemen UI yang telah disediakan oleh Flutter, serta mengatur tata letak (layout) sesuai dengan kebutuhan aplikasi peneliti. Misalnya, peniliti dapat membuat tampilan untuk menampilkan data monitoring daya listrik secara realtime, menampilkan hasil analisis berupa atau grafik visualisasi data dan data listrik sesuai dengan permintaan pengguna aplikasi.
2. Penggunaan *Firebase*: Flutter memiliki dukungan yang baik untuk integrasi dengan *Firebase*, yang merupakan platform cloud yang digunakan untuk penyimpanan data, autentikasi pengguna, dan berbagai layanan lainnya. Anda dapat menggunakan Flutter *Firebase* SDK untuk menghubungkan aplikasi Flutter dengan *Firebase*, sehingga peniliti dapat menyimpan dan mengambil data daya listrik, mengelola pengguna, dan melakukan komunikasi dengan backend.
3. Integrasi dengan Algoritma *Fuzzy Logic* Sugeno: Jika algoritma *Fuzzy Logic* Sugeno digunakan dalam bagian backend atau pemrosesan data, peniliti dapat mengintegrasikannya dengan Flutter menggunakan bahasa pemrograman Dart yang digunakan oleh Flutter. Peniliti dapat menulis kode Dart yang memanggil fungsi atau metode yang mengimplementasikan algoritma tersebut, dan menggunakan hasilnya untuk menampilkan informasi yang relevan pada antarmuka aplikasi.

Flutter menyediakan banyak fitur dan kemampuan untuk pengembangan aplikasi mobile yang kompleks dan interaktif. Dalam penelitian ini, peniliti dapat memanfaatkan kelebihan Flutter dalam hal pengembangan UI yang menarik,

interaksi dengan *Firebase*, dan integrasi dengan algoritma *Fuzzy Logic Sugeno* untuk menciptakan aplikasi monitoring daya listrik yang efektif dan *user-friendly*.

Berikut tampilan FrameWork Flutter di Android Studio IDE



Gambar 2. 29 Tampilan Flutter

## 2.21 Perangkat Android

Perangkat Android (Putra, 2019) adalah sistem operasi yang dikembangkan oleh Google untuk digunakan pada berbagai jenis perangkat seluler, seperti smartphone, tablet, smartwatch, dan perangkat lainnya. Perangkat Android menjadi populer karena fleksibilitasnya, kemampuan yang luas, dan adanya berbagai aplikasi yang tersedia di Google Play Store.

Perangkat Android menawarkan lingkungan yang kaya dan beragam untuk pengembangan aplikasi. Pengembang dapat menggunakan Android Studio, Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android, yang menyediakan berbagai fitur dan alat untuk membuat, menguji, dan menerapkan aplikasi Android (Setiyaningsih, 2023). Selain itu, Android juga mendukung berbagai bahasa pemrograman seperti Java dan Kotlin, yang memungkinkan pengembang untuk memilih bahasa yang sesuai dengan preferensi dan kebutuhan mereka.

Dalam penelitian ini ,perangkat Android digunakan sebagai platform untuk mengembangkan aplikasi yang memungkinkan pemantauan daya listrik melalui Internet of Things (IoT). Aplikasi ini dapat diinstal dan dijalankan pada perangkat Android untuk mengakses data daya listrik yang dikirim melalui jaringan IoT, menerapkan algoritma *Fuzzy Logic Sugeno* untuk analisis data, dan berkomunikasi dengan layanan *Firebase* untuk menyimpan dan mengelola data secara jarak jauh. Berikut tampilan aplikasi peneliti yang sudah di install di perangkat Android :



Gambar 2. 30 Tampilan aplikasi di Perangkat Android

## 2.22 Unified Modelling Language

UML (*Unified Modelling Language*) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek (Sutiono.M.Kom, 2022). UML disebut bahasa pemodelan yang digunakan untuk mendokumentasikan, merancang, dan memahami sistem perangkat lunak. UML menyediakan seperangkat notasi grafis yang digunakan untuk menggambarkan berbagai aspek sistem perangkat lunak, termasuk struktur, fungsi, interaksi, dan perilaku sistem.

Pada penelitian ini, peneliti melakukan UML untuk rancangan sistem aplikasi android dengan memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat. Jenis yang diambil untuk penelitian ini yaitu *Use Case* diagram, *Sequence Diagram* dan *Activity* diagram dari rancangan sistem aplikasi andr

## 2.23 Literatur Sejenis

Tabel 3. 1 Studi Literatur sejenis

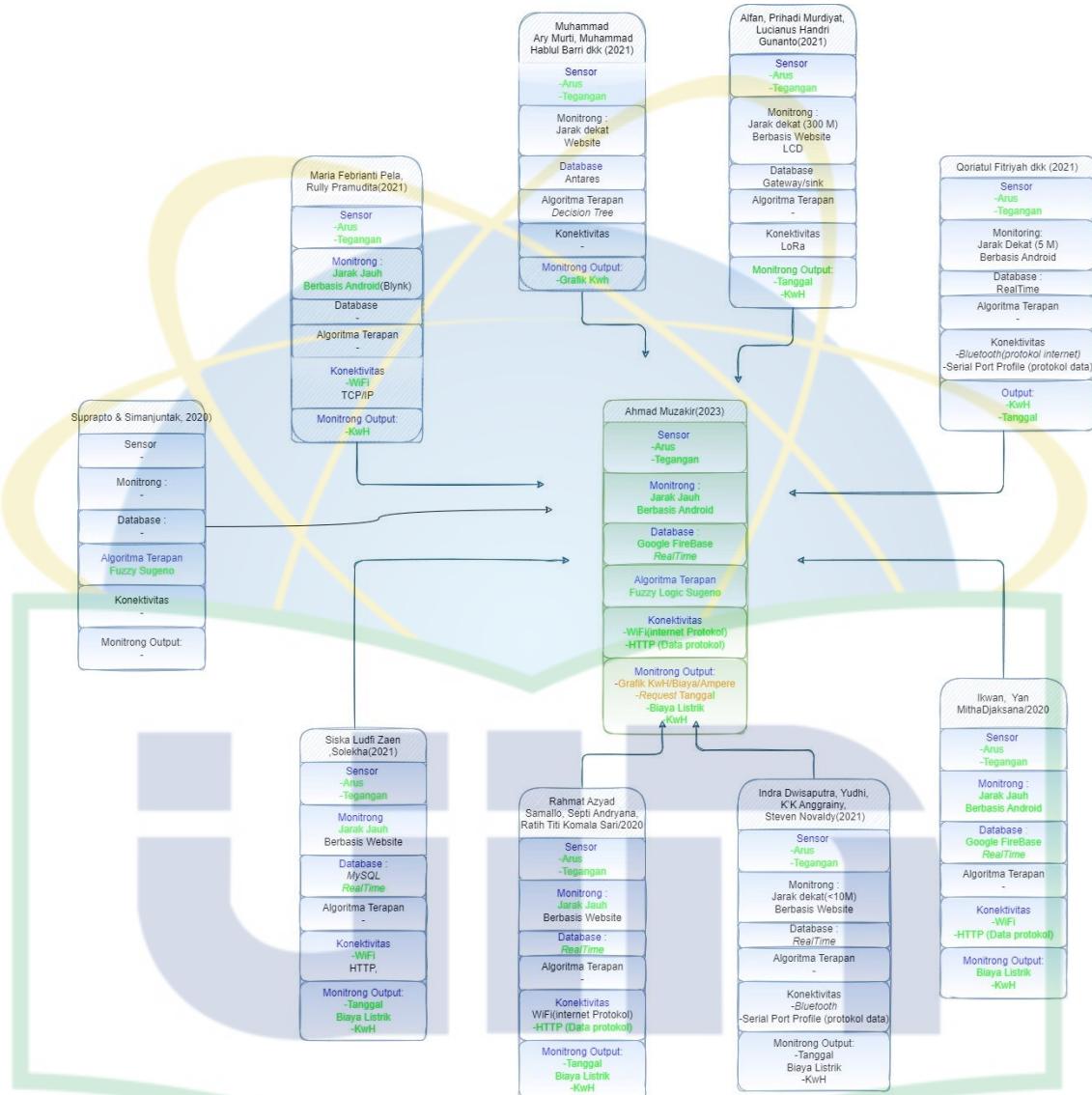
No	Nama Penulis /Tahun	Judul Penitian	Sensor		Monitoring Menggunakan Smartphone	<i>connect</i> internet	<i>Firebase</i>	RTC	Menggunakan Algoritma	Output		
			Microtroller	Sensoor						Total Kwh (req date)	Biaya Listrik req date)	Grafik Penentu penggunaan Kwh(Fuzzy Sugeno)
1	Alfan, Prihadi Murdiyat, Lucianus Handri Gunanto/2021	Rancang Bangun Sensor Node Untuk Sistem Monitoring Energi Listrik Nirkabel Pada Gedung Dalam Kampus Politeknik Negeri Samarinda	Arduinno Nano	PZEM004T	✗	√	√	✓	✗	✗	✗	✗

2	Qoriatul Fitriyah , Tri Vira Putri, Aditya Wirangga , Muhammad Prihadi Eko Wahyudi/2021	Rancang Bangun Alat Monitoring Energi Listrik Pada Kulkas Berbasis Smartphone	Arduino	Arus dan tegangan	√	✗	✗	√	✗	✗	✗	✗	✗
3	Muhammad Ary Murti, Muhammad Hablul Barri, Andi Shridivia Nuran,Selvy Zelia Sari, Difa Surya Putra, dan Syafri Akhsanul Iqbal/2021	Smart Meteringuntuk Pengidentifikasi Jenis Beban ListrikRumah Tangga Berbasis IoT Menggunakan PZEM-004T	Raspeberry PI 3B+	PZEM004T	✗	✗	✗	✗	Decision Tree	✗	✗	✗	✗

4	Maria Febrianti Pela, Rully Pramudita/2021	Sistem Monitoring Penggunaan Daya Listrik Berbasis IoT Pada rumah dengan Menggunakan Aplikasi Blynk	ESP8266	PZEM004T	√	√	✗	✗	✗	✗	✗	✗	✗
5	Siska Ludfi Zaen,Solekha, Imam Abdul Rozaq/2021	Sistem Monitoring Pemakaian Energi Listrik Rumah Tangga Berbasis Web	ESP8266	PZEM004T	✗	√	√	√	✗	✗	✗	✗	✗
6	Rahmat Azyad Samallo, Septi Andryana, Ratih Titi Komala Sari/2020	Android-Based Monitoring and Control of Electricity Consumption	ESP8266	PZEM004T	√	√	✗	✓	✗	✗	✗	✗	✗

7	Indra Dwisaputra, Yudhi, K'K Anggrainy, Steven Novaldy	Kontrol dan Monitoring Stop Kontak Berbasis Android	Arduino UNO R3	PZEM004T	√	✗	✗	√	✗	✗	✗	✗	✗
8	Ikwan, Yan MithaDjaksana/2020	Perencangan Sistem Monitoring dan Kontroling Penggunaan Daya Listrik Berbasis Android	ESP8266	PZEM004T	√	√	√	√	✗	✓	✗	✗	✗
9	Hendra Suprapto, Pastima Simanjuntak/2020	FUZZY LOGIC UNTUK MEMPREDIKSI PEMAKAIAN LISTRIK MENGGUNAKAN METODE MAMDANI	✗	✗	✗	✗	✗	✗	Fuzzy Mamdani	✗	✗	✗	✗

#### d. Mapping Studi literatur



Gambar 3. 1 Mapping Studi literatur

Berdasarkan hasil studi literatur dan mapping pada gambar di atas, terdapat beberapa penelitian yang terkait sebelumnya. maka dari itu, dapat diketahui perbedaan dan kelebihan dari penelitian yang dilakukan oleh penulis. Rancang bangun "Sistem Monitoring daya listrik Internet of Things menggunakan algoritma fuzzy logic sugeno dan firebase berbasis android" merupakan sebuah sistem yang dirancang untuk memonitor daya listrik menggunakan teknologi IoT dan mengaplikasikan algoritma *Fuzzy Logic Sugeno* pada sistem tersebut.

- 1) Dari semua jurnal sejenis yang dijadikan acuan tidak ada yang menampilkan grafik hasil dari *Fuzzy sugeno* , tidak ada yang menggunakan *by req date* untuk menampilkan total energy listrik dan biaya listrik sesuai *req* tanggal dan tidak ada rincian listrik yang diterapkan di aplikasi dalam bentuk tabel. Dalam pengujian sistem, peniliti menunjukkan bahwa sistem berhasil memonitor daya listrik dengan akurasi yang baik. pada aplikasi Android yang dibuat peniliti juga mampu menampilkan data secara real-time dengan tampilan yang cukup informatif dan mudah dipahami oleh pengguna seperti menampilkan data jumlah harga listrik, jumlah KwH yang digunakan dalam perhari ,memiliki fitur permintaan (*request*) tanggal untuk pengguna ingin menampilkan data sesuai keinginan yang ditampilkan di aplikasi dan mensajikan sebuah grafik hasil dari proses *Fuzzy sugeno*.
- 2) Sistem ini menggunakan sensor yang digunakan dari 8 jurnal yaitu PZEM-004T untuk mendapatkan data daya listrik dan kemudian dikirimkan ke board ESP8266 melalui komunikasi serial. ESP8266 menurut jurnal (Pela & Pramudita, 2021) NodeMCU telah menggabungkan ESP8266 ke dalam sebuah board yang kompak dengan berbagai fungsi layaknya mikrokontroler ditambah juga dengan kemampuan akses terhadap Wifi juga chip komunikasi USB to Serial sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data mikro USB. Pada board ESP8266 (Fitriyah et al., 2021) memudahkan pengguna listrik untuk mengakses data dari jarak yang sangat jauh dikarenakan komunikasi menggunakan WiFi kemudian mengirimkan data tersebut ke *Firebase* Realtime Database menggunakan protokol HTTP.
- 3) Untuk penyimpanan data dari hasil sensor disimpan di *Firebase*, penyimpanan disebut sangat layak digunakan karena data tersebut dikirimkan secara *realtime* atau terus-menerus. Salah satu jurnal sejenis yang menggunakan *firebase* yaitu jurnal (Mitha Djaksana et al., 2020) yang menjelaskan bahwa *firebase* bisa melakukan update data baru secara otomatis dan bisa digunakan sebagai platform sistem operasi handphone seperti android dan iOS
- 4) Dari beberapa jurnal diatas membuat sistem untuk melihat data secara realtime dibuat *template* android yang ada di *playstore* yaitu aplikasi Blynk (Mitha Djaksana et al., 2020), dan juga melihat data menggunakan *website* (Zaen & Rozaq, 2021) dengan keterbatasan akses untuk pengguna *smartphone*. Dalam penelitian ini menggunakan aplikasi android yang dibuat oleh peneliti untuk

membuat pengguna dapat melihat data daya listrik secara real-time melalui grafik yang disajikan menggunakan library MPAndroidChart. Selain itu, pengguna juga dapat melakukan kontrol terhadap perangkat listrik yang terhubung pada sistem melalui aplikasi tersebut.

- 5) Dari semua jurnal yang menjadi literatur peneliti menagambil data fungsi *Fuzzy logic* sugeno yang dimiliki oleh jurnal (Suprapto & Simanjuntak, 2020).Untuk melakukan pengambilan keputusan dalam penggunaan listrik,data listrik yang diteliti oleh dengan judul “Aplikasi Penentuan Tarif Listrik Menggunakan Metode *Fuzzy Sugeno*” diambil untuk acuan sistem menggunakan algoritma *Fuzzy Logic Sugeno* dengan ramu dengan tujuan dari penulisan. Algoritma ini menggunakan aturan-aturan yang dibuat berdasarkan kondisi-kondisi tertentu dan beroperasi dengan menggunakan fungsi implikasi, agregasi, dan defuzzifikasi dengan hasil berupa grafik di aplikasi andorid.

## **BAB III**

### **Metodologi Penelitian**

Pembahasan pada bab 3 ini penulis melakukan proses pengumpulan data dengan menggunakan metode pengumpulan data primer dan sekunder. Setelah mendapatkan data yang telah diperlukan untuk proses penelitian, selanjutnya penulis memulai penelitian dengan menggunakan metode pengembangan sistem yaitu prototipe. Dalam penelitian ini, penulis memulai dengan menentukan tujuan dari keseluruhan sistem yang akan dibuat berdasarkan latar belakang yang ada, dan mengidentifikasi apa saja yang akan dibutuhkan. Selanjutnya melakukan pemodelan berupa fungsi-fungsi apa saja yang nantinya akan diwujudkan dalam sebuah sistem berupa prototipe. Langkah setelah itu penulis akan melakukan testing terhadap prototipe yang hasilnya akan digunakan untuk evaluasi.

#### **3.1 Metode Pengumpulan Data**

Metode pengumpulan data digunakan untuk mencari dan mengumpulkan data yang terkait dengan penelitian seperti dasar teori, metodologi penulisan, metodologi proses, dan acuan penelitian sejenis. Dalam penelitian ini, metode pengumpulan data yang dilakukan adalah studi pustaka, dan studi literatur.

##### **3.1.1 Data primer**

Untuk pengumpulan data primer penulis menggunakan studi lapangan dengan metode observasi dan wawancara

###### a. Observasi

Penulis Melakukan Observasi ke Rumah Tangga disekitar Wilayah Jakarta yang bertemepat di Jalan Raya Tengah Kecamatan Kramat Jati Jakarta Timur pada tanggal 18 Agustus 2022.

Dan melihat banyak Pengguna Rumah Tangga masih melihat hasil pengguna KWH dari MCB maupun PLN.

###### b. Wawancara

Wawancara dilakukan kepada seorang responden yang berprofesi sebagai Pengguna Listrik, yang dilakukan pada tanggal 18 Agustus 2022

. Bernama Bapak Muhammad Solih dengan Berusia 29 Tahun yang

sudah memiliki rumah pribadi Bersama keluarga berdomisili di Jakarta dengan alamat di Jalan Raya tengah No 14 RT4/RW9 Kelurahan Tengah. ,hasil wawancara bisa dilihat pada lampiran dan digunakan untuk sebagai acuan dari tahapan pembangunan sistem montirong yaitu komunikasi.

### **3.1.2 Data Sekunder**

Pada tahapan pengumpulan data dengan cara studi Pustaka, dicari referensi-referensi yang relevan dengan objek yang akan diteliti. Pencarian referensi dilakukan di perpustakaan, toko buku, maupun secara online melalui internet. setelah mendapatkan mendapatkan referensi-referensi yang relevan tersebut, kemudian dipilih berbagai informasi yang dibutuhkan dalam penelitian ini. Adapun informasi yang dapat digunakan dalam penyusunan landasan teori, metodelogi penelitian serta pengembangan sistem secara langsung. Referensi yang dijadikan acuan dapat dilihat di daftar pustaka yang terdiri dari 9 referensi. Studi literatur sejenis merupakan kegiatan mencari literatur yang mempunyai persamaan atau keterkaitan dengan penelitian yang sedang dilakukan. Literatur sejenis yang didapatkan berupa penulisan skripsi dan juga produk yang kemudian ditelaah dan dibuat perbandingan sehingga penelitian ini dapat menjadi pelengkap atau penyempurnaan dari penelitian-penelitian yang sudah dilakukan sebelumnya.

## **3.2 Metode Pengembangan Sistem**

Dalam pengembangan sistem ini, digunakan metodelogi pengembangan sistem dengan metode Prototyping. Ada 5 tahapan prototyping yang digunakan dalam metode prototyping (Prototype, 2022)yaitu:

- 1) Tahap komunikasi
- 2) Tahap perencanaan cepat(*Quick planning*)
- 3) Tahap pemodelan cepat (*Quick Modeling*)
- 4) Tahap kontruksi (*Construction*)
- 5) Tahap evaluasi

### **3.2.1 Tahap Komunikasi**

Paradigma prototyping dimulai dengan adanya komunikasi antar aktor yang akan menggunakan sistem tersebut untuk menentukan sasaran hasil keseluruhan dari perangkat lunak/sistem, mengidentifikasi kebutuhan dan lingkungan dimana sistem tersebut akan digunakan. Pada tahapan ini komunikasi yang dilakukan adalah dengan mencari informasi terkait,

melalui referensi buku dan jurnal tentang masalah dalam sistem monitoring pengguna listrik rumah tangga dengan kemudian mendiskusikannya dengan pakar atau ahli dibidang penelitian penulis (diskusi dilakukan dengan dosen pembimbing skripsi) untuk menganalisis masalah lebih lanjut dan menyimpulkan sebuah solusi yang telah teruji serta manfaat penelitian kedepannya.

### 3.2.2 Perencanaan cepat (*Quick planning*)

Pada tahap perencanaan, penulis melakukan analisis kebutuhan apa saja yang didapatkan dari tahap sebelumnya. Perencanaan meliputi analisis terhadap metode, algoritma, perangkat keras dan perangkat lunak yang akan digunakan. Analisis yang dilakukan pada tahap ini adalah analisis konstruksi pada pengguna listrik rumah tangga, analisis sistem berjalan, analisis sistem usulan, analisis fungsional sistem, analisis kebutuhan perangkat keras, analisis kebutuhan perangkat lunak.

Pada tahap ini bahwa peneliti menggunakan PZEM-004T sebagai sensor daya listrik dikarenakan disemua literatur untuk mengetahui daya listrik menggunakan bahan tersebut, Sensor tersebut dapat mengetahui Arus,Tegangan dan daya listrik.

Sedangkan untuk mendapatkan hasil tersebut digunakan ESP8266 yang lebih efesien dikarena terhubung internet(*WiFi*), dari beberapa jurnal ada yang menggunakan *Bluetooth* untuk mengkoneksikan alat tersebut terapi kelamahannya tidak jauh untuk mengakses data yang sudah didapatkan.

Ketika sudah mendapatkan data hasil dari sensor akan disimpan ke *Firebase* sebagai database yang akan dioutput ke dalam Android untuk pengguna bisa membaca hasil dari monitoring daya listrik. android juga berfungsi sebagai implementasi dari *Fuzzy logic* yang mengambil varibel dari daya listrik dan luas rumah yang diinputnya.

*Fuzzy logic* dipilih sebagai algoritma pengambil keputusan karena *Fuzzy logic* memiliki cara kerja yang mirip seperti pemikiran manusia untuk mengambil keputusan apakah daya listrik masuk dalam kategori pengguna rendah,sedang atau tinggi, Metode sugeno dipilih karena berdasarkan studi literatur, metode sugeno memiliki tingkat akurasi yang lebih tinggi dari metode mamdani atau Tsukamoto.

Prototyping dimulai dengan pengumpulan persyaratan perancangan seperti bertemu pengguna pengguna listrik untuk menentukan tujuan

keseluruhan dari sistem dan alat, mengidentifikasi persyaratan apapun yang diketahui serta menentukan area garis besar yang mana definisi lanjut itu diharuskan. Pada tahapan pengumpulan kebutuhan ini, dilakukan pengumpulan data dengan cara observasi dan wawancara terhadap pihak terkait, studi pustaka dan literatur. Dari hasil pengumpulan data tersebut, penulis mendapatkan mengenai proses sistem monitoring listrik, komponen, tools, teori yang akan digunakan, data mengenai desain dan proses pembuatan alat dan sistem. Tahap pengumpulan kebutuhan ini akan terus berjalan selama masih membangun prototipe sampai tahap pengujian alatnya.

### 3.2.3 Pemodelan Cepat (*Quick Modeling*)

Dalam tahap Pemodelan Cepat (*Quick Modeling*), penulis membuat model skematik sistem dengan menggabungkan semua sensor dan aktuator yang akan digunakan dalam sistem monitoring daya listrik. Tahapan ini melibatkan beberapa langkah berikut:

a. Perancangan Kemasan:

- Penulis merencanakan desain kemasan atau wadah yang akan digunakan untuk menempatkan semua komponen sistem, termasuk sensor daya listrik, modul ESP8266, dan komponen lainnya.
- Desain kemasan harus mempertimbangkan ukuran, keamanan, dan kemudahan aksesibilitas komponen untuk perakitan dan pemeliharaan.

b. Pembuatan Skematik dengan Sensor Daya Listrik:

- Penulis membuat skematik elektronik yang menggambarkan hubungan antara ESP8266 dan sensor daya listrik (misalnya, PZEM-004T).
- Skematik ini mencakup pengaturan koneksi fisik antara sensor dan modul, seperti koneksi kabel, pin input/output yang digunakan, dan pengaturan daya.

c. Pembuatan Skematik dengan *Firebase*:

- Penulis juga membuat skematik elektronik yang menggambarkan hubungan antara modul ESP8266 dan *Firebase*.
- Skematik ini mencakup koneksi jaringan antara ESP8266 dan *Firebase*, seperti penggunaan protokol komunikasi (HTTP), pengaturan autentikasi, dan pengiriman data ke database *Firebase*.

d. Pembuatan Skematik dengan Android:

- Penulis membuat skematik yang menggambarkan interaksi antara aplikasi Android yang dibuat dengan sistem monitoring daya listrik.
- Skematik ini mencakup penggunaan API *Firebase* untuk mengakses dan memperbarui data, pengaturan antarmuka pengguna, dan proses pengolahan data yang ditampilkan kepada pengguna.

Pemodelan cepat ini membantu penulis dalam merencanakan tata letak dan hubungan antara komponen sistem secara visual. Skematik yang dibuat memberikan panduan yang jelas dalam proses perakitan dan integrasi komponen, serta memastikan bahwa semua komponen terhubung dengan benar dan berfungsi sebagaimana mestinya..

#### **3.2.4 Konstruksi (Construction)**

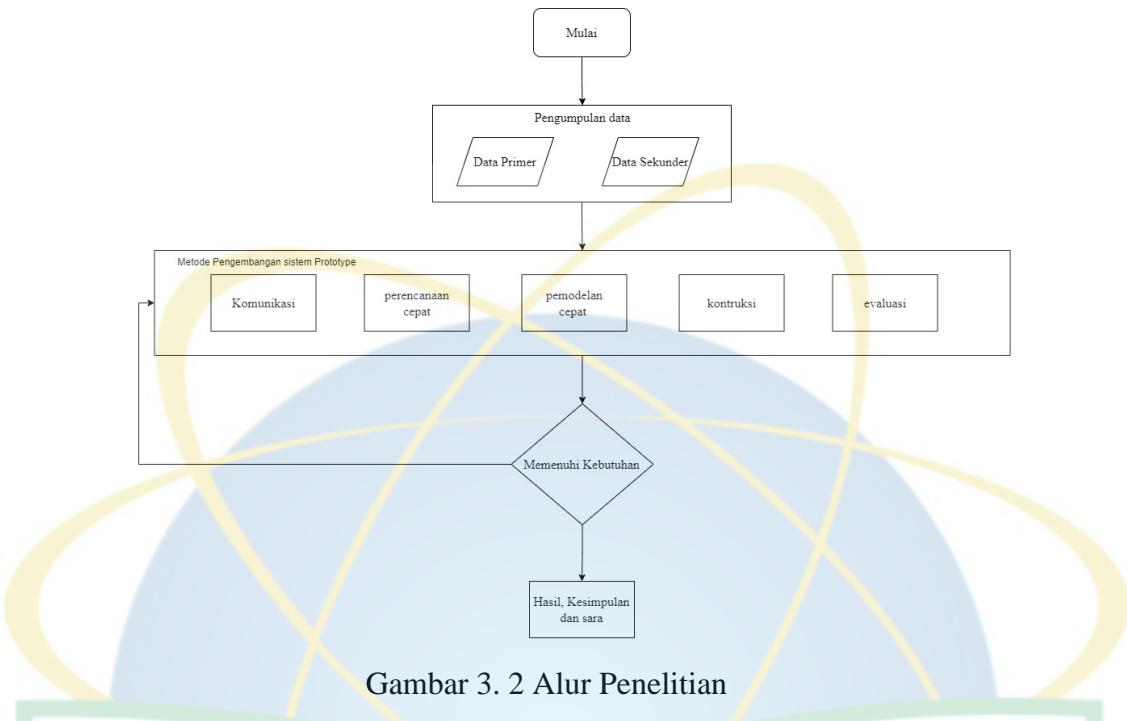
Pada tahap ini akan dilakukan konstruksi pada ESP8266 yang akan disambungkan dengan sensor daya listrik dan akan dihubungkan dengan android untuk menghasilkan output yang diolahnya yang didapatkan penyimpanan dari *firebase*. Pada tahap ini juga dibuat *Fuzzy logic* dengan kedua parameter sensor untuk diterapkan kedalam sistem. Hasil grafik dari *Fuzzy logic* juga akan dilihat untuk mengetahui detail nilai output *Fuzzy*.

#### **3.2.5 Evaluasi**

Pada tahap ini, pengujian dilakukan menggunakan black box testing, yang bertujuan untuk mengetahui kesesuaian fungsi Arduino, *firebase*, aplikasi android, logika *Fuzzy* dan fungsional sistem. Black box testing yang akan digunakan adalah pengujian Arduino, *firebase*, aplikasi android logika *Fuzzy* dan pengujian output android

### **3.3 Alur Penelitian**

Permbuatan Monitoring daya listrik rumah disusun melalui beberapa tahapan yang harus dilakukan dengan tujuan memudahkan dalam melakukan penelitian. Adapun alur atau kerangka penelitian yang dilakukan, dapat dilihat pada gambar dibawah ini:



Gambar 3. 2 Alur Penelitian

. Pada penelitian ini, alur penelitian dimulai dengan tahap pengumpulan data. Data dikumpulkan melalui dua sumber, yaitu data primer dan data sekunder. Data primer diperoleh melalui observasi langsung dan wawancara dengan responden terkait. Data sekunder diperoleh melalui studi pustaka, yaitu mengumpulkan informasi dari berbagai sumber literatur terkait dengan topik penelitian.

Setelah pengumpulan data, langkah selanjutnya adalah proses pengembangan sistem. Tahap ini melibatkan komunikasi antara peneliti dan pihak terkait untuk menentukan tujuan sistem dan persyaratan yang harus dipenuhi. Kemudian dilakukan perencanaan cepat untuk merancang skema atau desain sistem secara kasar. Pemodelan cepat dilakukan dengan membuat model skematis sistem, yang menggambarkan komponen dan interaksi antara mereka. Setelah itu, konstruksi dilakukan untuk membangun sistem sesuai dengan desain yang telah dibuat. Evaluasi dilakukan untuk memastikan kinerja dan akurasi sistem.

Setelah tahap pengembangan selesai dan sistem telah berfungsi dengan baik, dilakukan analisis data dan penyusunan hasil penelitian. Hasil penelitian ini kemudian dituangkan dalam penulisan akhir, yaitu dalam bentuk skripsi atau laporan penelitian yang mencakup latar belakang, tujuan, metodologi, hasil, dan kesimpulan penelitian.

## BAB IV

### ANALISIS DESAIN, IMPLEMENTASI, DAN PENGUJIAN SISTEM

Bab ini akan membahas secara detail dan terperinci mengenai analisis dan perancangan sistem, implementasi dan juga pengujian sistem monitoring daya listrik dengan penerapan *internet of things* menggunakan algoritma *Fuzzy logic* sugeno dan *firebase* berbasis android.

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode prototipe yang sudah dibahas pada bab sebelumnya. Isi dari bab ini yaitu menguraikan tahap pengembangan alat prototipe di antaranya adalah:

1. Tahap Komunikasi
2. Tahap Pengumpulan Kebutuhan
3. Tahap Membangun Sistem
4. Tahap Mengkodekan Sistem
5. Tahap Menguji Sistem

Berikut penjelasan detail tahap pengembangan pada penelitian ini

#### 4.1 Tahap Komunikasi

Pada tahap komunikasi ini, penulis melakukan pengumpulan data, pengumpulan data yang dimaksud adalah proses mencari dan mengumpulkan data yang dibutuhkan terkait dengan penelitian. Pada tahapan ini penulis juga melakukan studi kepustakaan dan pencarian jurnal-jurnal yang terkait, peneliti melakuakan observasi dan wawancara terhadap pengguna listrik rumah tangga yang sudah dilakukan pada tanggal 18 Agustus 2022. Observasi yang dilakukan secara langsung ke lokasi mendapatkan bahwasanya pengguna listrik masih menggunakan meteran listrik konevnsional yang dimana pengguna mengetahui tagihan listrik melalui PLN pusat dan melakukan wawancara dengan mendapatkan informasi secara detail ,agar peniliti mendapatkan informasi-informasi terkini tentang permasalahan yang ada pada proses monitoring daya listrik berbasis android.

Pada tahapan ini. Dilakukan diskusi langsung dengan dosen

pembimbing serta pengguna daya listrik untuk mengetahui gambaran sistem yang diperlukan untuk selanjutnya dijadikan bahan pertimbangan dalam membangun sistem yang dibutuhkan.

#### **4.2 Tahap Perencanaan cepat (*Quick planning*)**

Pada tahap ini, dijelaskan apa saja yang menjadi kebutuhan sistem, yang meliputi mendefinisikan ruang lingkup, analisis kebutuhan software maupun hardware, dan juga analisis sistem berjalan dan usulan.

##### **4.2.1 Ruang Lingkup**

Ruang lingkup pada penelitian ini yaitu pengguna daya listrik rumah. Sistem monitoring yang diusulkan adalah daya listrik bisa dipantau atau dimonitorng dari hasil Kwh (daya listrik) oleh pengguna listrik rumah yang berbasis Android, Monitoring bisa dipantau melalui jarak yang jauh, Pengguna bisa melihat grafik dari data Kwh sebelumnya atau seterusnya dan bisa memfilter Tanggal grafik yang diinginkan oleh User. Pengembangan dilakukan berdasarkan hasil wawancara dengan melibatkan narasumber.

##### **4.2.2 Analisis Kebutuhan Hardware**

Dalam pembuatan sistem monitoring daya listrik, dibutuhkan beberapa perangkat keras atau hardware baik berupa mikrokomputer dan komponen elektronik lainnya. Pemilihan spesifikasi hardware menjadi sangat penting agar sistem dapat berjalan dengan baik sesuai dengan kebutuhan pengguna. Tabel 4.1 menerangkan daftar komponen yang dibutuhkan:

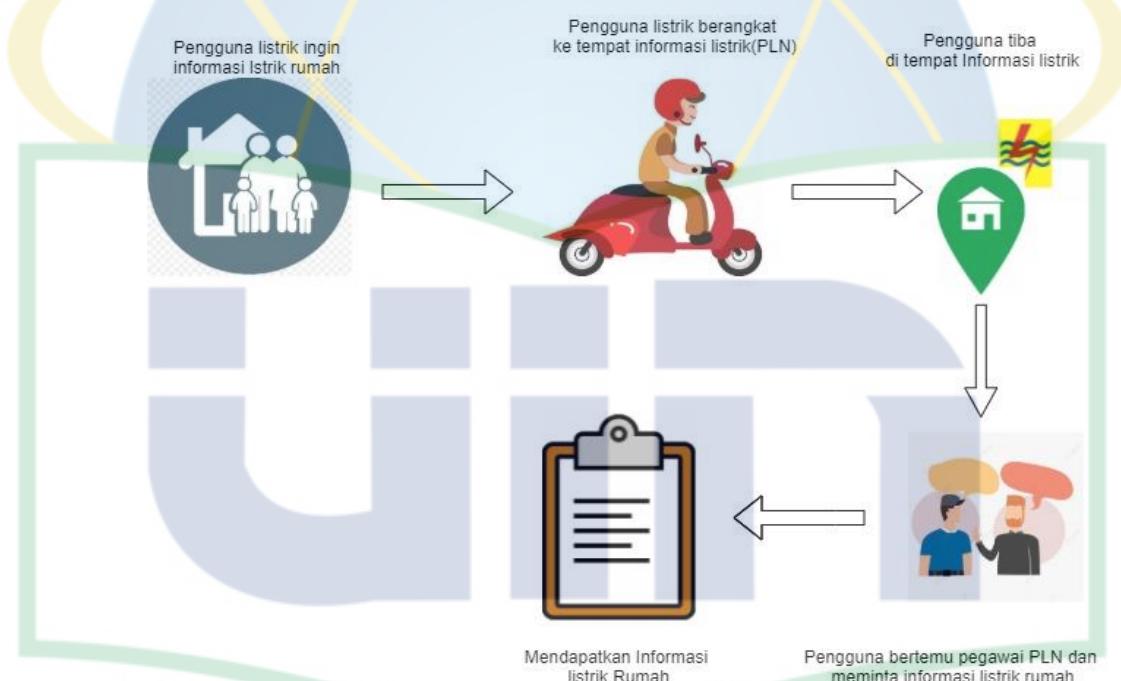
*Tabel 4. 1 Analisis kebutuhan hardware*

No	Komponen	Jumlah	Kegunaan
1.	Modul WiFi ESP8266	1	digunakan berkomunikasi atau kontrol melalui internet
2.	Pzem-004T	3	berfungsi untuk mengukur :Voltage/ Tegangan, Arus, Daya, Frekuensi, Energi dan Power Faktor.

3.	Stop Kontak	1	Berfungsi sebagai sumber daya yang diambil data sensor
4.	Plug(Colokan)	1	Untuk memberikan power di Pzem-004
5.	Kabel	1 Meter	Untuk mehubungan antara Pzem dan Plug
6.	Box Project	1	Sebagai kemasan Hardware
7.	Kabel jumper	20 buah	Sebagai penghubung antara pzem dan esp8266

#### 4.2.3 Analisis Sistem Berjalan

Berdasarkan hasil wawancara dan studi literatur yang dilakukan, dapat dilihat sistem berjalan pada gambar 4.1.



Gambar 4. 1 Analisis Sistem Berjalan

Pada gambar diatas pengguna listrik secara manual mencatat penggunaan listrik dengan membaca meteran listrik pada interval tertentu, misalnya bulanan. Mereka harus mencatat angka meteran listrik awal dan akhir untuk menghitung total penggunaan listrik selama periode tersebut.

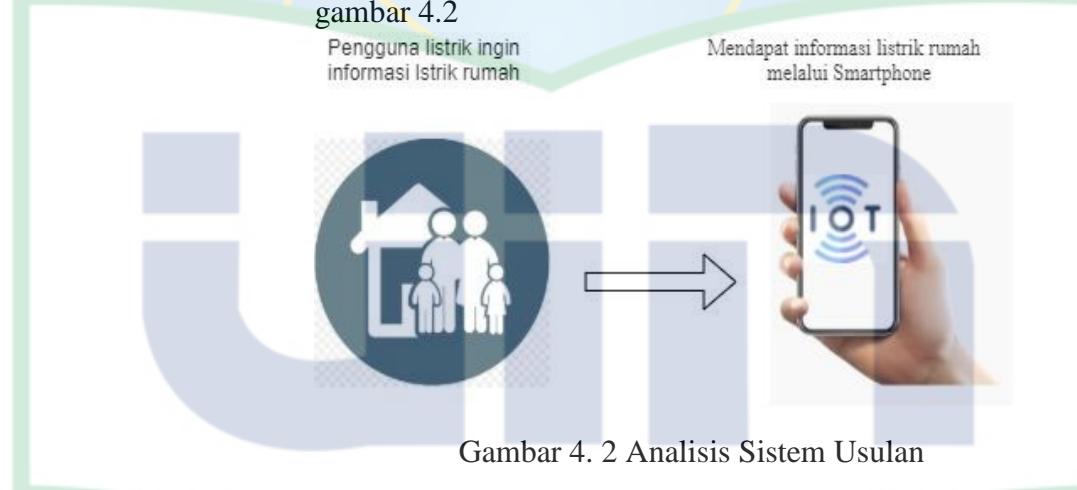
Pengguna listrik harus menghitung sendiri tagihan listrik mereka berdasarkan tarif listrik yang diberlakukan oleh penyedia listrik. Mereka harus menggunakan data penggunaan listrik yang telah dicatat untuk menghitung biaya yang harus

dibayarkan. Jika pengguna listrik ingin mendapatkan informasi lebih lanjut tentang penggunaan listrik mereka, pengguna harus pergi ke kantor PLN atau tempat informasi listrik yang ditunjuk.

Dalam metode konvensional, pengguna listrik tidak memiliki akses langsung atau real-time terhadap data penggunaan listrik. Pengguna hanya dapat mengakses informasi tersebut pada saat mereka membaca meteran atau mengunjungi tempat informasi listrik. Hal ini membatasi pemantauan dan pengendalian penggunaan listrik secara real-time dan pengukuran penggunaan listrik sering kali tidak akurat karena bergantung pada pengambilan manual oleh pengguna. Kesalahan atau ketidaktepatan pengukuran dapat terjadi, yang dapat mempengaruhi akurasi tagihan dan pengendalian penggunaan listrik.

#### 4.2.4 Analisis Sistem Usulan

Agar mempermudah dalam mendapatkan informasi listrik rumah maka Peneliti membuat sistem usulan untuk Pengguna listrik rumah. Berikut adalah sistem susulan dapat dilihat pada gambar 4.2



Gambar 4. 2 Analisis Sistem Usulan

Pada gambar diatas terdapat aplikasi smartphone yang telah dirancang untuk memantau daya listrik rumah dengan menggunakan sistem monitoring daya listrik berbasis Android. Aplikasi ini memungkinkan pengguna listrik untuk melihat informasi dan grafik daya listrik secara jarak jauh tanpa harus berada di lokasi fisik informasi daya listrik.

Dengan menggunakan aplikasi tersebut, pengguna listrik dapat melakukan pemantauan daya listrik rumah mereka melalui smartphone. Pengguna dapat melihat grafik dan tabel analisis dari waktu yang diinginkan, yang memungkinkan pengguna untuk menganalisis pola penggunaan daya listrik, mendeteksi

perubahan yang tidak biasa, dan membuat keputusan yang lebih cerdas terkait penggunaan energi.

Dengan aplikasi berbasis Android ini, pengguna listrik memiliki fleksibilitas untuk mengakses informasi daya listrik secara real-time dan memantau penggunaan daya listrik rumah mereka secara efisien, bahkan dari jarak jauh.

Dengan demikian, pengguna listrik dapat dengan mudah dan praktis memantau dan mengelola penggunaan daya listrik rumah mereka melalui aplikasi smartphone yang telah dirancang khusus. Hal ini memberikan kemudahan, kenyamanan, dan memberikan pengguna informasi yang diperlukan untuk mengoptimalkan penggunaan energi mereka.

#### 4.2.5 Analisa Kebutuhan *Software* dan *Tools*

Selain Hardware yang sudah disebutkan sebelumnya, dibutuhkan juga Software dan Tools untuk mendukung hardware agar berjalan sesuai dengan yang diharapkan. Berikut software yang dibutuhkan dalam pembuatan sistem monitoring daya listrik ini agar dapat berjalan sesuai dengan yang diharapkan.

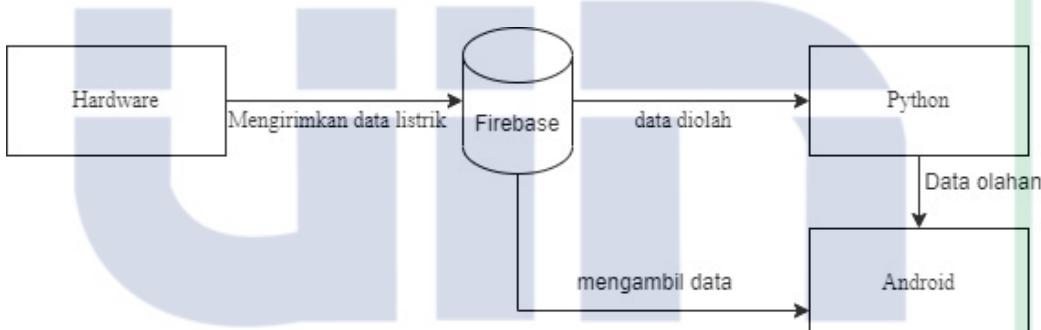
Tabel 4. 2 Kebutuhan *software*

No	Nama <i>Software/Tools</i>	Kegunaan
1.	Arduino IDE	Digunakan untuk menulis kode program yang berisi perintah yang diperlukan untuk diunggah pada mikrokontroler <i>Module WiFi ESP8266</i>
2.	<i>Firebase</i>	Sebagai penyimpanan data listrik dari <i>Module WiFi ESP8266</i> dan dikirimkan ke android
3.	Android Studio	sebagai tempat pengkodean <i>Interface</i> sistem monitoring listrik pengguna listrik rumah
4.	Visual studio Code	tempat pengkodean <i>Fuzzy</i> dan komponen software untuk memproses hasil dari sistem monitoring

5.	Draw.io	Digunakan sebagai Desain berbentuk 2D jalur sistem yang berjalan serta susulan
6.	Fritzing	Digunakan untuk membuat skema dari sistem Monitoring listrik rumah
7.	Pyhton	bahasa pemrograman yang melakukan analisis dan pengolahan data yang diperoleh dari data daya listrik
8.	Flutter	framework pengembangan aplikasi mobile

#### 4.2.6 Analisis kebutuhan Fungsional Sistem

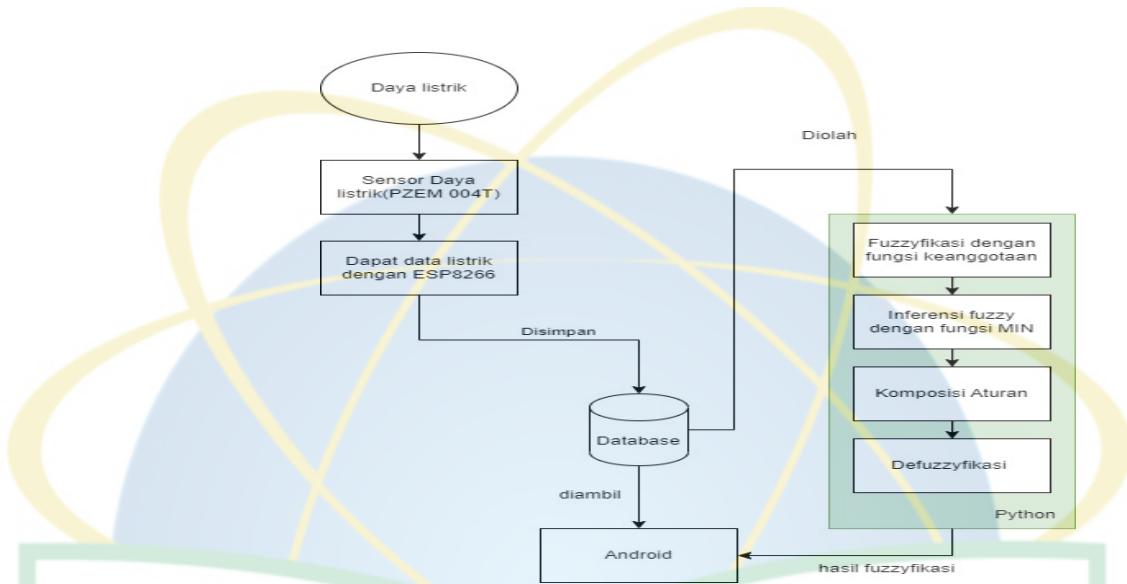
Analisis kebutuhan Fungsional sistem dilakukan dengan mengambar Data flow diagram untuk memahami alur data dan flowchart untuk mengetahui keseluruhan fungsional sehingga peneliti mendapat gambaran mengenai sistem yang akan berjalan secara lebih terperinci, yang akan mempermudah dalam proses pembangunan sistem nantinya, berikut gambaran dari sistem yang akan dibentuk :



Gambar 4. 3 DFD Level 0

Pada gambar Gambar DFD 0 diatas menunjukkan alur sistem monitoring listrik dengan melibatkan hardware, *Firebase*, *Python*, dan aplikasi *Android*. *Hardware* mengirimkan data listrik ke *Firebase*, Data listrik tersebut dikirimkan melalui koneksi internet ke *Firebase* sebagai database cloud. Data listrik yang terkirim ke *Firebase* diakses melalui *API Firebase* dari lingkungan *Python*, Proses pengolahan data dilakukan di *Python*, yang dapat mencakup analisis, Hasil digunakan untuk hasil data olahan ke aplikasi android. Aplikasi *Android* dapat mengakses *Firebase* menggunakan *Firebase SDK* atau *API* untuk mengambil data yang tersimpan di

**Firebase**, Data yang telah diolah di Python dapat diambil langsung oleh aplikasi Android, Aplikasi Android dapat menampilkan data tersebut kepada pengguna, melakukan visualisasi data dan menampilkan data realtime yang didapatkan dari *Firebase*



Gambar 4. 4 DFD Level 1

Pada gambar DFD level 1 yang diatas, pertama dilakukan mendapatkan Data sensor listrik dengan alat PZEM004T yang diambil oleh ESP8266, lalu data tersimpan di *firebase* secara *realtime* yang mendapatkan hasil Arus, Tegangan, *timestamp*, Daya listrik dan harga listrik. Data yang sudah disimpan akan diolah menggunakan Pyhton sebagai proses *Fuzzyifikasi* dan aplikasi android menampilkan data monitoring daya listrik yang akan disesuaikan dengan data Visualisasi dari data pyhton dan tabel untuk mempermudah pengguna membaca data.

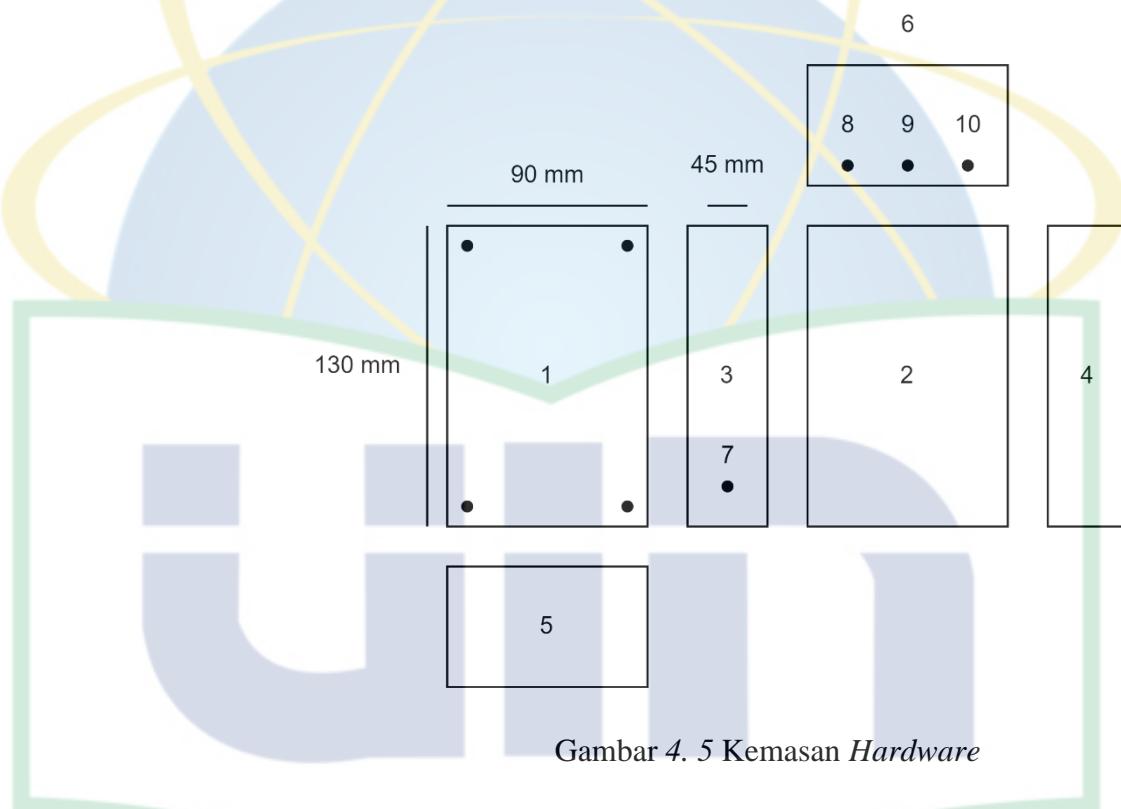
Selain berfungsi sebagai output monitoring daya listrik, Android berfungsi sebagai tempat hasil *Fuzzy* dengan data listrik rumah perhari yaitu KWh, daya listrik, Jumlah perangkat, dan biaya listrik yang akan diolah masuk kedalam keanggotaan *Fuzzy*. tahap inferensi menggunakan fungsi MIN yang akan dilakukan terhadap nilai keanggotaan *Fuzzy*. Jumlah inferensi ini bergantung kepada jumlah aturan *Fuzzy* yang dibuat. Setelah itu masuk ke proses *defuzzyifikasi* menggunakan metode sugeno. Hasil dari perhitungan *Fuzzy* untuk menentukan penggunaan daya listrik rendah, sedang atau tinggi, hasil Proses ini akan didapatkan ketika hari sudah berganti dan menghasilkan grafik diandroid.

Dalam *Data Flow Diagram* Level 2 dijelaskan lebih detail gambaran dari alur sistem monitrong listrik dengan dibagi menjadi 3 alur diagram yaitu DFD Level 2 bagian Hardware, *Firebase* dan android-python pada pembahasan pemodelan cepat.

#### 4.3 Tahap Pemodelan cepat

##### 4.3.1 Kemasan

Desain perancangan kemasan dilakukan pada alat yang sudah direncanakan, berikut rancangan kemasan dari tiap-tiap alat :



Gambar 4. 5 Kemasan *Hardware*

Berikut penjelasan mengnai lubang-lubang dan bagian dari kemasan yang akan dibuat :

1. Bagian depan dari kemasan atau tutup kemasan, empat buah lubang di masing-masing sudut adalah lubang untuk memasukkan baut untuk mengencangkan tutup agar tidak mudah lepas.
2. Bagian belakang dari kemasan.
3. Bagian kanan dari kemasan
4. Bagian kiri dari kemasan.

5. Bagian bawah dari kemasan.
6. Bagian atas dari kemasan.
7. 1 buah lubang untuk menghubung USB ke ESP8266
8. 1 buah lubang untuk memberikan power atau mengaktifkan PZEM004T pertama.
9. 1 buah lubang untuk memberikan power atau mengaktifkan PZEM004T kedua
10. 1 buah lubang untuk memberikan power atau mengaktifkan PZEM004T ketiga

Dan rancangan yang diatas, peneliti melampirakan gambar hasil dari kemasan yang sudah dirancang sesuai dengan skematik secara keseluruan untuk Hardware. Berikut gambar hasil kemasan :

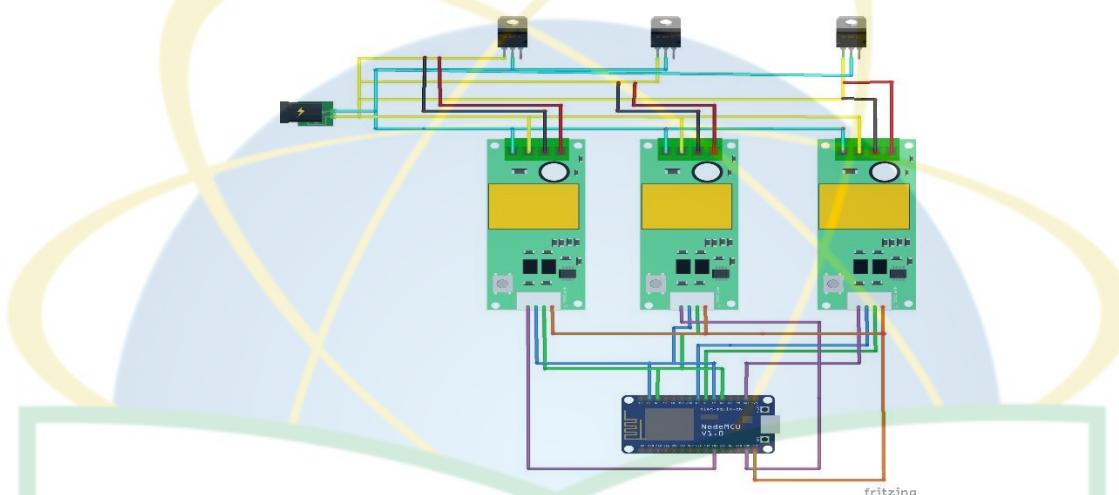


Gambar 4. 6 hasil kemasan

### 4.3.2 Skematik Sistem dengan perangkat Hardware

#### a. Skema Hardware

Dalam skematik penelitian ini akan menjelaskan bagaimana ESP8266 terhubung dengan PZEM yang sudah mengambil data listrik.



Gambar 4. 7 Skematik full Sistem dengan Sensor Daya listrik dan ESP8266

Tabel 4. 3 Pin Hardware

PZEM004T	ESP8266
5V	VIN
RX 1 2 3	D1 5 7
TX 1 2 3	D2 6 8
GND 1 2 3	GND 1 2 3

Pada gambar diatas skematik tersebut, terdapat sistem daya listrik yang menggunakan 3 alat sensor PZEM dan modul ESP8266 sebagai pembaca data. Sistem ini dirancang untuk mengambil data dari sumber listrik melalui stop kontak.

Modul ESP8266 berfungsi sebagai pembaca data yang menerima data dari sensor PZEM dan mengirimkannya ke perangkat lain, seperti server atau aplikasi yang menjalankan logika dan analisis data. Modul ESP8266 dapat berkomunikasi dengan sensor PZEM melalui protokol komunikasi yang sesuai, seperti UART.

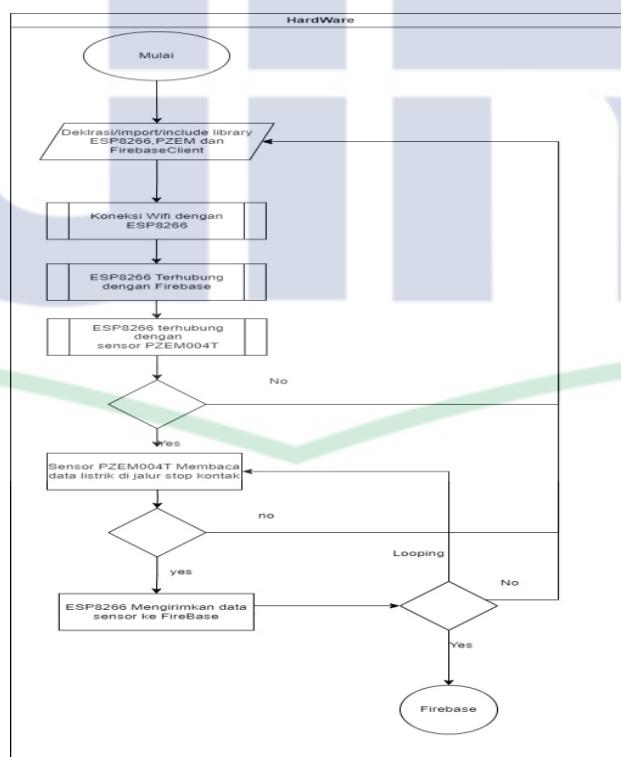
Dengan kombinasi sensor PZEM dan modul ESP8266, sistem dapat membaca dan memantau data daya listrik secara real-time. Data yang dikumpulkan dapat digunakan untuk analisis, pengontrolan, atau pemantauan tingkat daya listrik yang dikonsumsi oleh perangkat yang terhubung.

Terdapat koneksi kabel yang menghubungkan setiap alat sensor PZEM dengan modul ESP8266. Kabel-kabel tersebut digunakan untuk mentransfer data yang diukur oleh sensor PZEM ke modul ESP8266. Selain itu, terdapat kabel power yang digunakan untuk memberikan daya kepada ketiga alat sensor tersebut.

Dengan adanya koneksi yang tepat antara sensor PZEM dan modul ESP8266, modul ESP8266 dapat membaca data yang diukur oleh setiap sensor PZEM secara individual. Data tersebut kemudian dapat dikirimkan ke perangkat lain, seperti server *firebase*, untuk dianalisis, dipantau, atau dikontrol sesuai kebutuhan.

#### b. *Data Flow Diagram Hardware*

Data Flow diagram ini menggambarkan alur *hardware* secara khusus, flowchart ini masuk bagian DFD ke level 2 karena pada pembahasan 4.2.6 menggambarkan DFD level 1, berikut flowchart/DFD level 2 bagian hardware :



*Gambar 4. 8 Flowchart/DFD Level 2 bagian hardware*

Berikut adalah alur skema untuk mendapatkan data listrik dari perangkat hardware seperti PZEM dan ESP8266 menggunakan Arduino IDE:

1) Deklarasi:

- Mulailah dengan mendeklarasikan semua variabel dan pin yang diperlukan dalam program.
- Ini termasuk deklarasi pin untuk komunikasi antara ESP8266 dan PZEM, serta variabel untuk menyimpan data listrik.

2) Koneksi WiFi:

- Menggunakan WiFi library pada Arduino IDE, Anda perlu mengatur koneksi WiFi pada ESP8266.
- Dalam langkah ini, Anda akan mencari jaringan WiFi yang tersedia, memasukkan kredensial WiFi, dan melakukan koneksi ke jaringan tersebut.
- Setelah terhubung, ESP8266 akan mendapatkan alamat IP WiFi.

3) Koneksi ke PZEM:

- Menggunakan library yang sesuai untuk komunikasi serial (SoftwareSerial atau Serial1 pada ESP8266), peneliti menghubungkan ESP8266 dengan PZEM.
- Tentukan pin yang akan digunakan untuk komunikasi serial dan inisialisasi objek serial.
- Mulai komunikasi dengan PZEM untuk membaca data sensor listrik seperti tegangan, arus, dan daya.

4) Koneksi ke *Firebase*:

- Untuk mengirim data listrik ke *Firebase*, peneliti perlu menghubungkan ESP8266 dengan *Firebase* menggunakan *Firebase* library atau REST API.
- Berikan informasi otentikasi *Firebase* seperti token atau kunci akses.
- Buatlah koneksi dengan *Firebase* dan tentukan path atau node *Firebase* yang akan digunakan untuk menyimpan data listrik.

5) Proses Pembacaan Data PZEM:

- Dalam loop utama program, baca data sensor listrik dari PZEM menggunakan metode yang disediakan oleh library PZEM.

- Baca data tegangan, arus, dan daya dari PZEM dan simpan dalam variabel yang sesuai.

6) Pengiriman Data ke *Firebase*:

- Setelah mendapatkan data listrik dari PZEM, kirim data tersebut ke *Firebase* menggunakan fungsi yang disediakan oleh *Firebase* library atau melalui permintaan REST API.
- Format data sesuaikan dengan struktur yang diharapkan oleh *Firebase*.

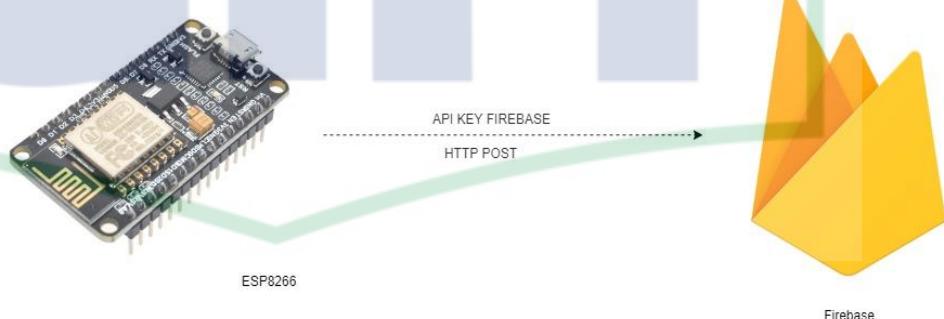
7) Pengulangan:

- Setelah mengirim data ke *Firebase*, program akan kembali ke langkah 5 dan terus membaca data sensor listrik secara berulang.
- Pastikan ada penundaan atau pengaturan waktu yang sesuai antara pembacaan data untuk mengontrol frekuensi pengambilan data.

Selama menjalankan program ini, ESP8266 akan terus membaca data dari PZEM, mengirimkannya ke *Firebase*, dan mengulang proses tersebut. Dengan demikian, penilit dapat mendapatkan data listrik secara realtime dan menyimpannya dalam database *Firebase* untuk pengolahan dan visualisasi lebih lanjut.

#### 4.3.3 Skematik dengan *Firebase*

a. Skematik hubungan antara modul ESP8266 dengan *Firebase*



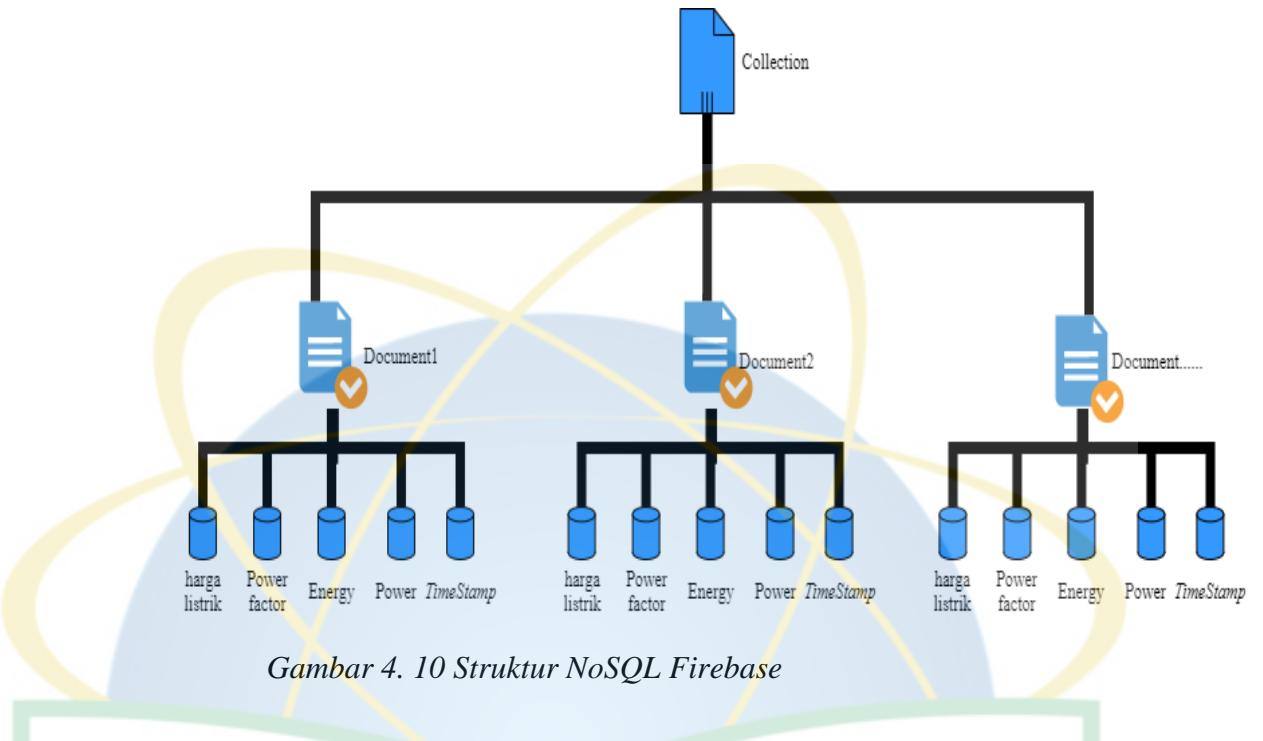
*Gambar 4. 9 Skema Firebase dan ESP8266*

Pada gambar diatas menunjuk skema yang dilakukan antara ESP8266 dan arduinio,berikut penjelasanya :

- ESP8266 terhubung ke jaringan WiFi menggunakan SSID dan kata sandi yang telah ditentukan sebelumnya. Ini memungkinkan ESP8266 untuk terhubung ke internet.
- Sensor Listrik: ESP8266 membaca data listrik dari sensor yang terhubung ke pin yang sesuai pada ESP8266. Sensor ini dapat mengukur nilai listrik seperti arus, tegangan, atau daya, dan mengubahnya menjadi nilai analog yang dapat dibaca oleh ESP8266.
- *Firebase API Key*: peneliti mendapatkan API Key *Firebase* peneliti (dengan API key : "AIzaSyCkiDrj-gBX7\_xp-6rHqn-KeKNgBFfW2uw") dari konsol *Firebase*. API Key ini digunakan untuk mengotentikasi permintaan ke *Firebase*.
- Kode Arduino IDE: Dalam kode Arduino IDE, peneliti menggunakan fungsi *HTTP POST* untuk mengirimkan data listrik ke *Firebase*. Peneliti membentuk permintaan *HTTP POST* ke URL *Firebase* yang sesuai, dan melampirkan API Key sebagai parameter otorisasi dan data listrik sebagai *payload* permintaan.
- *Firebase*: *Firebase* akan menerima permintaan *HTTP POST* yang dikirimkan oleh ESP8266. *Firebase* akan memproses data yang diterima dan menyimpannya dalam struktur database yang sesuai. *Firebase* juga dapat mengirimkan respon *HTTP* ke ESP8266 untuk memberi tahu hasil pemrosesan data.

#### b. Struktur *Firebase*

Pada tahap ini peneliti mendapatkan data listrik yang dikirimkan oleh ESP8266 melalui arduino IDE ke CloudFirestore, berikut struktur NoSQL *Firebase*



Gambar 4. 10 Struktur NoSQL Firebase

Pada gambar 4.7, terdapat bentuk struktur NoSQL *Firebase* yang digunakan dalam sistem monitoring daya listrik. *Firebase* digunakan sebagai basis data untuk menyimpan data secara real-time dan memungkinkan pembaruan data secara otomatis.

Struktur NoSQL *Firebase* dapat terdiri dari koleksi (collection) dan dokumen (document). Pada kasus ini, dokumen berisi data listrik seperti harga listrik, power, factor, energy, dan timestamp (waktu pengambilan data).

Ketika ESP8266 membaca data dari alat sensor PZEM, data tersebut akan diakses dan dimasukkan ke dalam dokumen yang sesuai di *Firebase*. Misalnya, setiap kali data listrik baru diterima, data tersebut akan diperbarui secara langsung ke dokumen yang sesuai dalam *Firebase*.

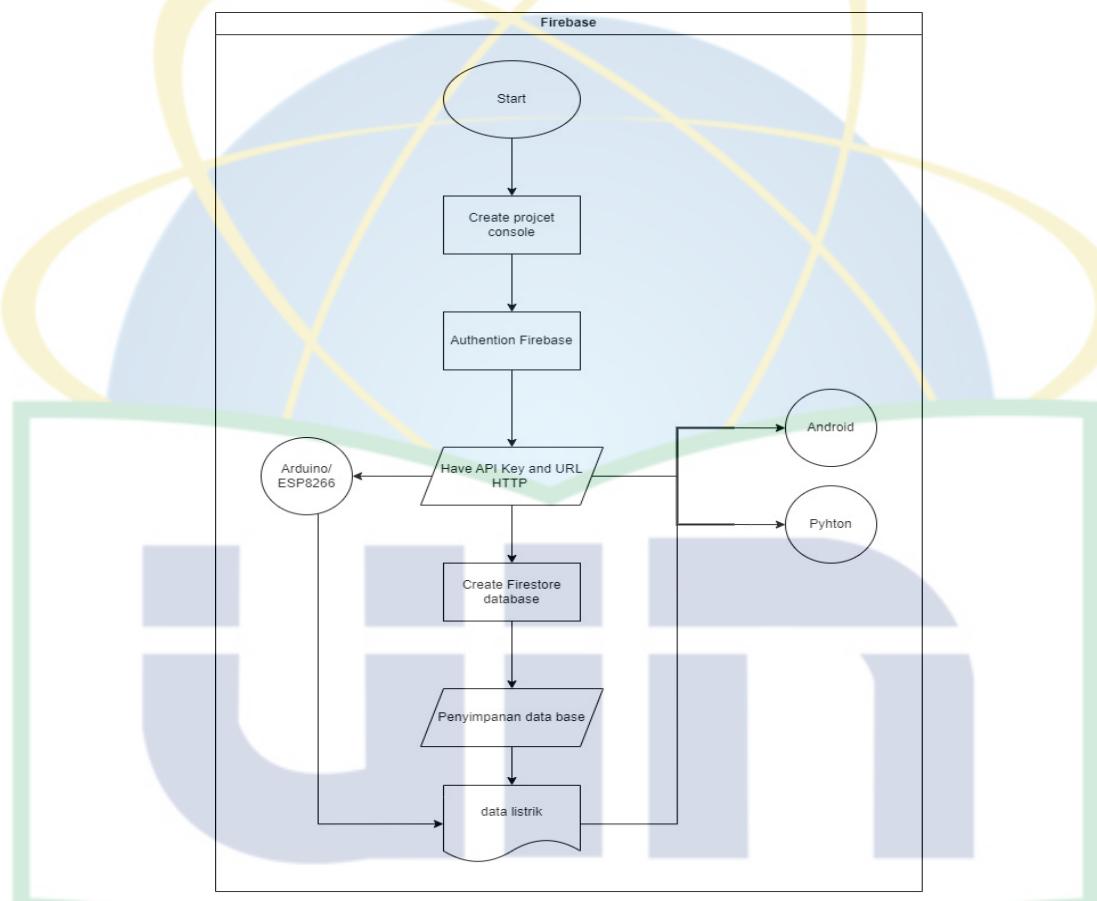
Dengan menggunakan struktur NoSQL *Firebase*, data listrik akan selalu diperbarui secara real-time sesuai dengan pembaruan yang dilakukan oleh ESP8266. Hal ini memungkinkan pengguna atau sistem lain yang terhubung dengan *Firebase* untuk melihat data listrik terkini tanpa perlu melakukan permintaan secara manual.

Dengan adanya integrasi *Firebase*, data listrik dapat disimpan, diakses, dan diperbarui dengan mudah. *Firebase* juga menyediakan fitur-fitur lain seperti sinkronisasi data secara otomatis antar perangkat yang

terhubung dan penyediaan API untuk integrasi dengan aplikasi atau sistem lain.

### b. . Data Flow Diagram Firebase

*Data Flow diagram* ini menggambarkan alur *firebase* secara dalam, flowchart ini masuk bagian DFD ke level 2 karena pada pembahasan 4.2.6 menggambarkan DFD level 1, berikut flowchart *Firebase* :



Gambar 4. 11 DFD level 2 bagian firebase

Berikut adalah alur skema *Firebase* yang mencakup proses pembuatan akun *Firebase*, pengaturan proyek *Firebase* Console, konfigurasi autentikasi *Firebase*, dan penggunaan Firestore sebagai database penyimpanan data listrik:

#### 1) Membuat Akun *Firebase*:

- Langkah pertama adalah membuat akun *Firebase* dengan mengunjungi situs web *Firebase* dan mengikuti proses pendaftaran.
- Setelah berhasil mendaftar, peniliti akan memiliki akses ke *Firebase* Console.

2) Membuat Proyek di *Firebase* Console:

- Di *Firebase* Console, buat proyek baru dengan memberikan nama proyek yang relevan.
- Proyek *Firebase* akan berfungsi sebagai tempat untuk mengelola semua sumber daya terkait aplikasi, seperti database, autentikasi, dan penyimpanan.

3) Konfigurasi Autentikasi *Firebase*:

- Aktifkan layanan autentikasi *Firebase* yang sesuai untuk aplikasi peniliti (misalnya, email/password, Google, atau Facebook).
- Dalam langkah ini, peniliti akan mengatur metode otentikasi yang akan digunakan oleh pengguna untuk mengakses aplikasi Android.

4) Mendapatkan API Key:

- Dalam *Firebase* Console, navigasikan ke pengaturan proyek dan dapatkan API Key *Firebase*.
- API Key ini akan digunakan dalam implementasi ESP8266 dan aplikasi Android untuk mengautentikasi dan mengirim data ke *Firebase*.

5) Konfigurasi *Firebase* di ESP8266:

- Di dalam program Arduino IDE atau platform pengembangan ESP8266, tambahkan library *Firebase* dan mengatur koneksi ke *Firebase* menggunakan API Key yang telah diperoleh.
- Dalam program ESP8266, Anda akan mengonfigurasi autentikasi dan koneksi ke Firestore atau Realtime Database *Firebase*.

6) Konfigurasi *Firebase* di Python:

- Untuk pengolahan data listrik di Python, peneliti menginstal library *Firebase* dan menggunakan API Key yang sama untuk mengonfigurasi koneksi ke *Firebase*.
- Dalam skrip Python, peneliti melakukan operasi pembacaan dan pemrosesan data dari Firestore atau Realtime Database *Firebase*.

7) Membuat Firestore Database:

- Di *Firebase* Console, buat Firestore Database baru sebagai tempat penyimpanan data listrik dari perangkat keras.
- Tentukan struktur database yang sesuai, seperti koleksi dan dokumen yang akan digunakan untuk menyimpan data listrik.

Setelah mengikuti langkah-langkah di atas, peniliti memiliki konfigurasi *Firebase* yang siap digunakan untuk mengirim, menyimpan, dan memproses data listrik dari perangkat keras menggunakan ESP8266, Python, dan aplikasi Android. *Firebase* akan berfungsi sebagai server dan database untuk aplikasi, memungkinkan peniliti mengakses data secara realtime dan mengelola penggunaan.

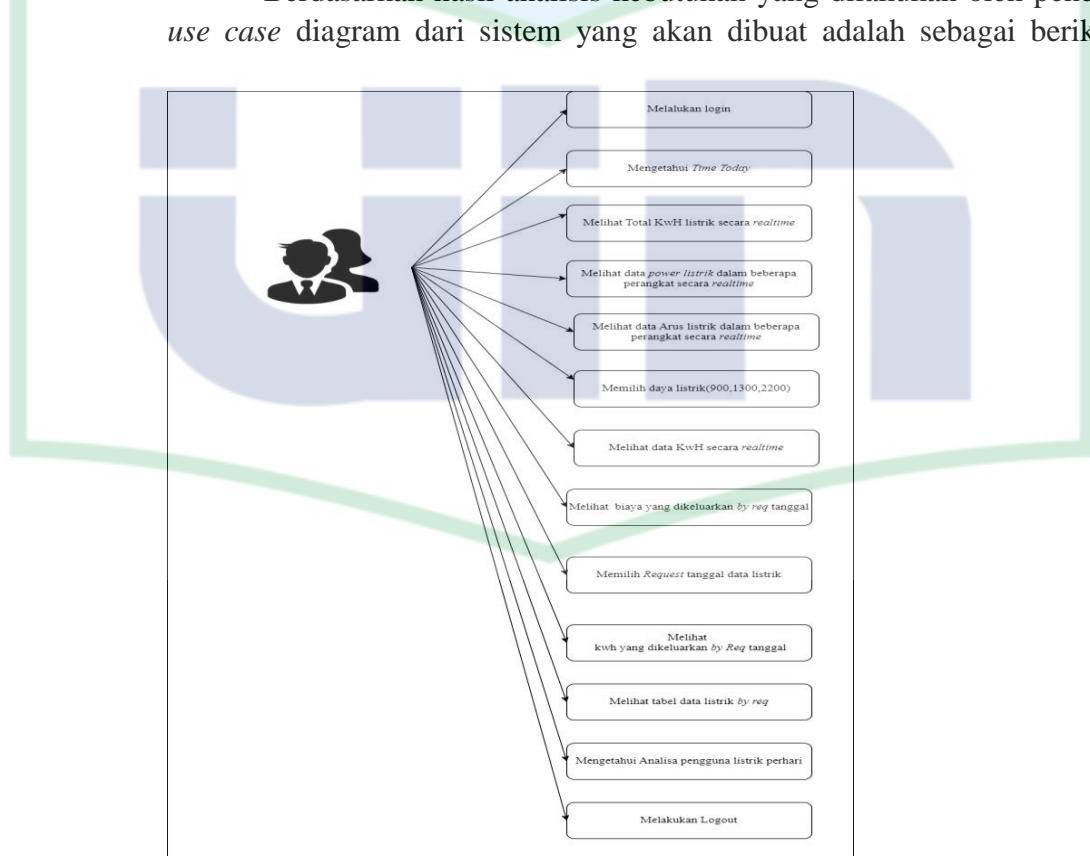
#### 4.3.4 Skematik dengan android dan Pyhton

Pada Skematik dengan android peneliti memberikan informasi terkait dengan tampilan antarmuka yang akan digunakan oleh pengguna monitoring listrik untuk memonitoring data yang sudah tersimpan di *Firebase*.

##### a. Use case

*Use Case Diagram* merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. *Use Case Diagram* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.

Berdasarkan hasil analisis kebutuhan yang dilakukan oleh peneliti *use case* diagram dari sistem yang akan dibuat adalah sebagai berikut:

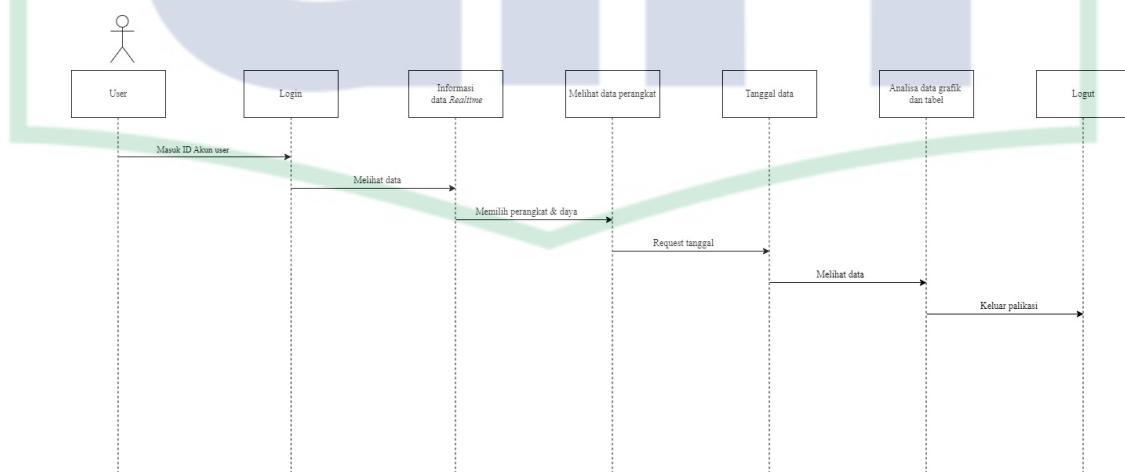


Gambar 4. 12 Use Case

Pada gambar diatas ialah gambaran skenario dari interaksi antara pengguna listrik dengan sistem aplikasi android yang dibuat oleh peneliti dengan beberapa fitur seperti Login; Pengguna dapat melakukan login ke aplikasi menggunakan username dan password yang valid. Ini memungkinkan pengguna untuk mengakses fitur-fitur yang terkait dengan data listrik,Melihat Data Realtime; Pengguna dapat melihat data realtime seperti arus dan power yang dikumpulkan dari sensor-sensor PZEM. Data ini akan ditampilkan secara aktual dan diperbarui secara otomatis,Mengetahui Energy atau kWh Hari Ini: Pengguna dapat melihat informasi tentang total energi atau kWh yang dikonsumsi pada hari yang sedang berjalan. Ini memberikan gambaran tentang penggunaan listrik pada hari tertentu,Analisis Data Listrik; Pengguna dapat mengakses tabel data listrik yang mencakup informasi seperti arus, power, faktor daya, dan energi. Pengguna dapat melakukan pencarian berdasarkan tanggal tertentu dan melihat total listrik sesuai dengan permintaan tanggal yang dimasukkan.,Grafik Analisis Pengguna: Pengguna dapat melihat grafik yang menganalisis penggunaan listrik mereka. Grafik ini dapat memberikan informasi tentang golongan pengguna listrik, seperti rendah, sedang, atau tinggi. Grafik tersebut dapat disesuaikan berdasarkan rentang tanggal yang dipilih oleh pengguna.

#### b. Squence diagram user

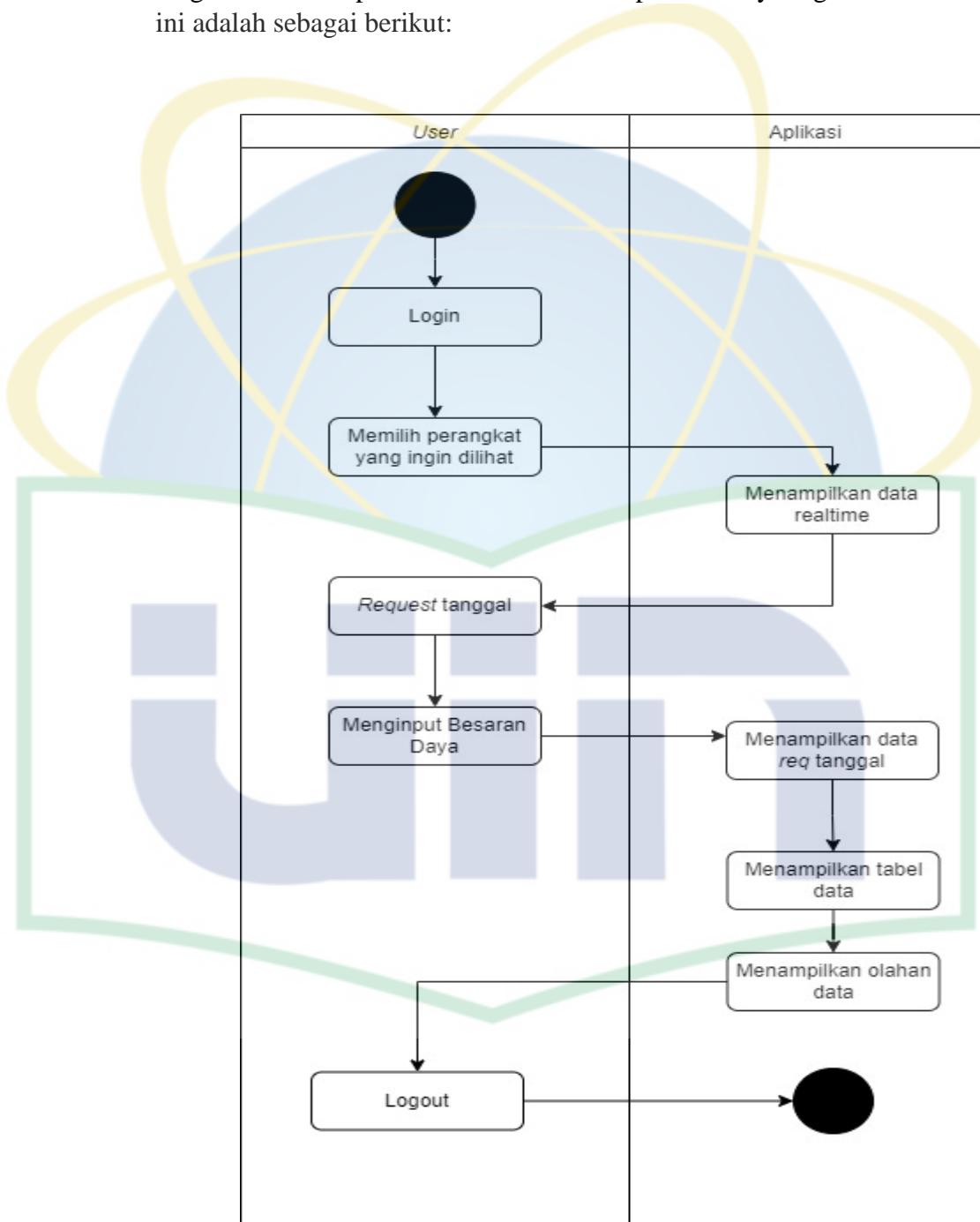
Squence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar berupa message terhadap waktu. Pembuatan squence diagram bertujuan agar rancangan lebih mudah dan terarah. Interaksi-interaksi yang terjadi dalam aplikasi yang dihasilkan sistem ini adalah



Gambar 4. 13 Squence diagram user

#### c. Activity Diagram

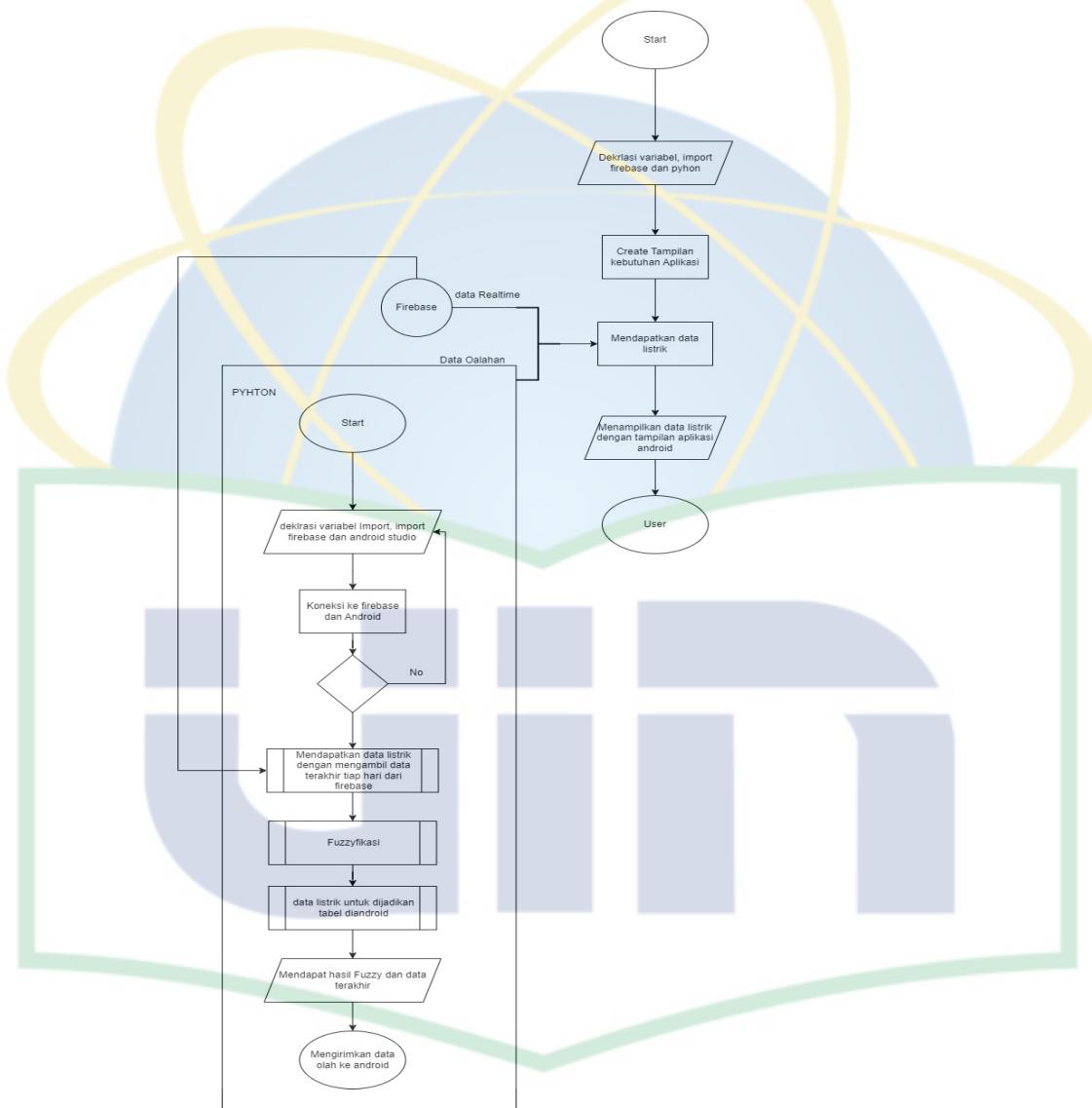
Activity Diagram adalah representasi grafis dari seluruh tahapan alur kerja yang mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem. Adapun activity diagram dari sistem ini adalah sebagai berikut:



Gambar 4. 14 Activity diagram

d. *Data Flow Diagram Android*

DFD tersebut memberikan gambaran umum tentang alur data antara Android, Python, dan *Firebase* dalam sistem monitoring listrik. Berikut Gambaran DFD level 2 :



*Gambar 4. 15 Data Flow Diagram Android*

Berikut adalah gambaran alur skema Data Flow Diagram (DFD) untuk sistem monitoring listrik menggunakan Android dan Python:

- 1) Proses di Android:

- Android Studio digunakan untuk mengembangkan aplikasi Android untuk sistem monitoring listrik.
- Aplikasi akan berinteraksi dengan pengguna dan menampilkan data monitoring listrik secara visual.
- Aplikasi juga akan mengirim permintaan ke server *Firebase* untuk mengambil data listrik secara real-time.

## 2) Proses di Python:

- Python digunakan untuk pemrosesan data listrik yang diambil dari *Firebase*.
- Python akan terhubung ke *Firebase* menggunakan library *Firebase* dan mengambil data dari Firestore atau Realtime Database.
- Data listrik akan diproses menggunakan algoritma *Fuzzy logic* atau metode lain sesuai dengan kebutuhan.
- Hasil pemrosesan akan digunakan untuk visualisasi data atau pengambilan keputusan.

## 3) Interaksi antara Android dan Python:

- Android akan mengirimkan permintaan ke server *Firebase* untuk mengambil data listrik secara real-time.
- Python akan mengambil data tersebut dari *Firebase* dan melakukan pemrosesan sesuai dengan logika yang ditentukan.
- Hasil pemrosesan akan dikirimkan kembali ke Android untuk ditampilkan kepada pengguna.

## 4) Tampilan dan Pengolahan Data di Android:

- Android akan menampilkan data monitoring listrik dalam bentuk grafik, tabel dan.
- Pengguna dapat melihat data secara real-time dan memantau penggunaan listrik.

## 5) Pemrosesan Data di Python:

- Python akan menggunakan data yang diambil dari *Firebase* untuk pemrosesan lebih lanjut.
- Pemrosesan bisa termasuk perhitungan statistik, klasifikasi, atau metode lainnya sesuai dengan kebutuhan aplikasi.

## 6) Integrasi dengan *Firebase*:

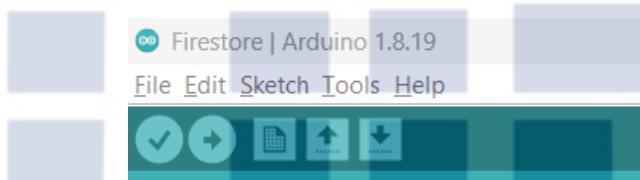
- Baik Android maupun Python akan terhubung ke *Firebase* untuk mengambil dan menyimpan data listrik.
- *Firebase* akan berfungsi sebagai server dan database untuk sistem monitoring listrik.

## 4.4 Kontruksi

Konstruksi dilakukan setelah Perangkat keras terbentuk. Pengkodean dilakukan pada Visual Code dengan menggunakan Bahasa Python yang diterakapkan di android untuk menjalankan sistem monitoring listrik dan memenuhi kebutuhan fungsionalitas yang telah ditetapkan pada proses perencanaan. Berikut ini adalah langkah-langkah pada tahapan konstruksi.

### 4.4.1 Pengkodean Arduino IDE

Dalam Pengkodean di *Arduino IDE* terdapat beberapa sistem yang dijalankan , yaitu mendapatkan data listrik dari sensor PZEM 004T ,Mengirimkan data ke *Firebase* dan Menetapkan atau mendeklarasikan internet ke ESP8266 sebagai sistem pengiriman daya ke *Firebase*. Hal pertama membuka aplikasi *Arduino IDE* lalu klik new project, seperti gambar dibawah ini



Gambar 4. 16 Tampilan Arduino IDE

- a. Pengkodean Arduino IDE dengan ESP8266 dan PZEM004T

Untuk mendeklasikan internet sebagai deklarasi WiFi dibutuhkan agar modul dapat terhubung ke jaringan WiFi dan dapat mengirim dan menerima data melalui koneksi tersebut.

Dengan mendeklarasikan koneksi WiFi pada ESP8266 yang sudah ditentukan nama jaringan WiFi (SSID) dan kata sandi (password) dari jaringan yang akan dihubungkan, maka modul akan dapat melakukan proses autentikasi dan otentikasi terhadap jaringan WiFi. Setelah terhubung dengan jaringan WiFi, modul ESP8266 dapat digunakan untuk mengirim dan menerima data melalui koneksi WiFi tersebut ke *firebase* yang didapatkan dari sensor PZEM004T .Berikut pekodean di *Arduino IDE* dengan ESP8266

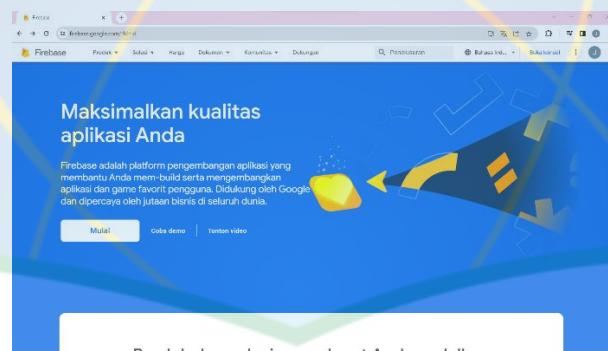
dan PZEM004T.Segala kode program Arduino IDE disimpan dalam git yang dapat diakses :

<https://github.com/muzakirjohn/SistemMonitoringpartarduinoIDE>

#### 4.4.2 Pembuatan *Firebase*

Dalam sistem monitoring listrik *Internet of Things* terdapat *firebase* untuk menyimpan data dari sensor listrik yang peneliti rancang, *firebase* tersebut memiliki tahapan atau proses untuk bisa menjadi pusat penyimpanan secara realtime, berikut tahapan pembuatan *firebase* :

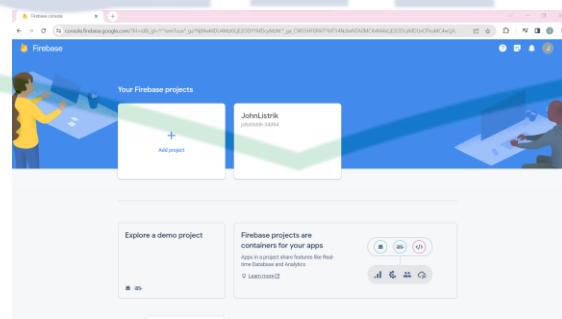
##### a. Pembuatan console *firebase*



Gambar 4. 17 Tahap pembuatan *Firebase* part 1

Pertama yang dilakukan yaitu membuat atau klik "Mulai" pada gambar diatas untuk pembuatan *firebase*.

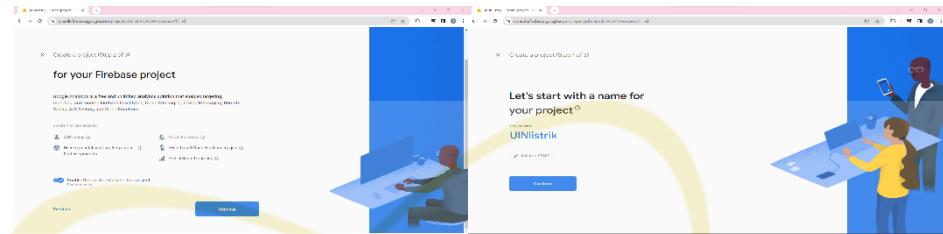
- b. Setelah masuk, Anda akan melihat tombol "Tambah proyek" atau "Buat proyek baru". Klik tombol tersebut.



Gambar 4. 18 Tahap pembuatan *Firebase* part 2

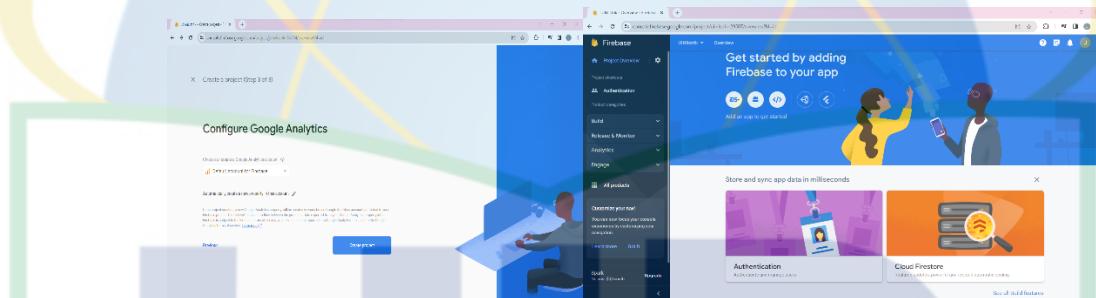
- c. Pada jendela pop-up, berikan nama untuk proyek *Firebase*.peneliti juga dapat memilih untuk mengaktifkan Google Analytics untuk

proyek peneliti.



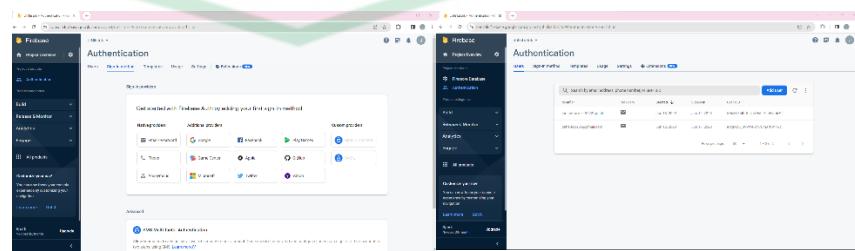
Gambar 4. 19 Tahap pembuatan *Firebase* part 3

- d. Setelah peneliti memberikan nama untuk proyek, klik tombol "Buat proyek". *Firebase* akan membuat proyek peneliti dan mengarahkan peneliti ke tampilan proyek *Firebase* yang baru dibuat. Di sini, peneliti akan menemukan berbagai layanan *Firebase* yang dapat peneliti gunakan.



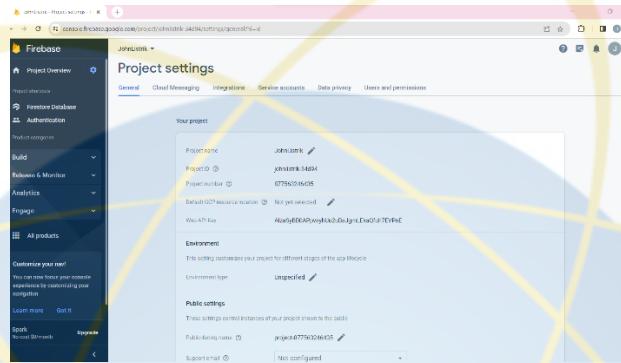
Gambar 4. 20 Tahap pembuatan *Firebase* part 4

- e. Di panel sebelah kiri *Firebase* Console, pilih opsi "Authentication" atau "Otentikasi". Ini akan membuka halaman pengaturan otentikasi. Pada halaman pengaturan otentikasi, peneliti akan menemukan berbagai opsi untuk mengkonfigurasi dan mengelola metode otentikasi yang tersedia. Beberapa metode otentikasi yang didukung oleh *Firebase* termasuk Email/Password yang peneliti pilih sebagai metode otentifikasi



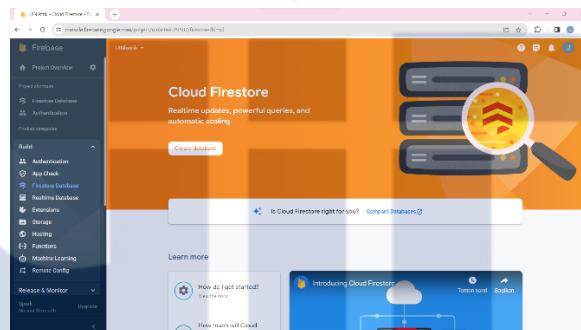
Gambar 4. 21 Tahap pembuatan *Firebase* part 5

- f. Selama proses konfigurasi, peneliti mendapatkan tambahan seperti ID klien OAuth, kunci API, atau konfigurasi tambahan yang diperlukan oleh masing-masing metode otentikasi dan komponen tersebut digunakan ketika pengiriman dari alat hardware monitoring listrik ke *firebase* dan permintaan pengambilan data *firebase* ke aplikasi/python .



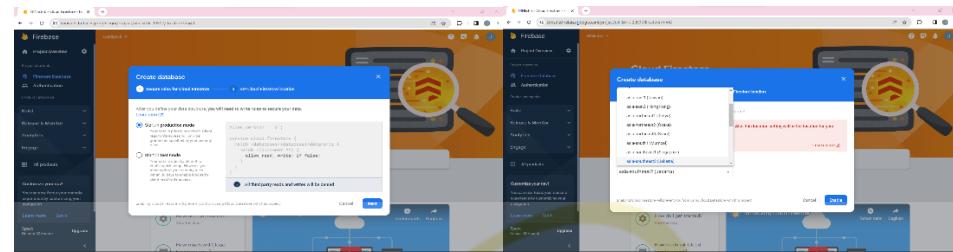
Gambar 4. 22 Tahap pembuatan *Firebase* part 6

- g. Lalu Di panel sebelah kiri *Firebase* Console, pilih opsi "Firestore Database" atau "Database". Ini akan membuka halaman pengaturan Firestore.



Gambar 4. 23 Tahap pembuatan *Firebase* part 7

- h. Pada halaman pengaturan Firestore, klik tombol "Buat database" atau "Mulai koleksi". peneliti akan diberikan opsi untuk memilih mode database: "*Production mode*" atau "*Test mode*". Pilih mode yang sesuai dengan kebutuhan peneliti.peneliti memilih "*Production mode*".



Gambar 4. 24 Tahap pembuatan *Firebase* part 9

#### 4.4.3 Pengkodean *Fuzzy Logic* dengan Metode Sugeno

Berikut merupakan tahapan-tahapan kode untuk mengimplementasikan *Fuzzy logic* menggunakan metode sugeno.

Berikut kode *Fuzzy* yang dilakukan di Visual code dengan Bahasa Python dan penjelasannya ;

- Kode deklarasi Trasmitter *firebase* yaitu untuk menerima dan mengirimkan data ke *firebase*:

```
from flask import Flask, render_template, request, redirect, url_for, Response
import json
import firebase_admin
from firebase_admin import credentials
from firebase_admin import firestore
import time
from datetime import datetime
import pandas as pd
import os

cred = credentials.Certificate("./monitoring.json")
a.     firebase_admin.initialize_app(cred)
```

- deklarasi variabel input *Fuzzy* :

```
def FuzzyLogic(Power, jumlahperangkat=1, HasilDaya=0, stopwatch=0, biayalistrik=0):
    # input Fuzzy
    daya_listrik = float(Power)
    Sum_perangkat = float(jumlahperangkat)
    Daya = float(HasilDaya)
```

```
waktu = float(stopwatch)  
biaya = float(biayalistrik)
```

c. Deklarasi variabel himpunan *Fuzzy* untuk masing masing variabel:

```
Power = [0, 0, 0]
jumlahperangkat = [0, 0, 0]
HasilDaya = [0, 0, 0]
stopwatch = [0, 0, 0]
biayalistrik = [0, 0]
```

d. deklarasi variabel implikasi dari aturan yang telah dibuat:

```
[[[0, 0], [0, 0], [0, 0]]]]  
]
```

- e. Deklarasi variabel yang digunakan pada tahap deFuzzyifikasi sebagai faktor perkalian dari deFuzzyifikasi :

```
penggunaan_Rendah = 0  
penggunaan_normal = 1  
penggunaan_tinggi = 2
```

- f. Mengisi domain atau nilai keanggotan energi yang sudah ditentukan dengan tabel dan gambar kurva keanggotan landasan teori Fuzzy

```
# Fuzzyfication  
# sensor energy listrik  
# RENDAH 0 0,66 1  
# SEDANG 0,833 1,66 2,5  
# TINGGI 2,33 3,66 4  
  
# #energylistrik rendah  
if energy_total < 0.66:  
    energy[0] = 1  
elif energy_total < 1:  
    energy[0] = (1 - energy_total)/(1 - 0.83)  
else:  
    energy[0] = 0  
  
# energylistrik Sedang  
if energy_total < 0.83:  
    energy[1] = 0  
elif energy_total < 1.66:  
    energy[1] = (energy_total-0.83)/(1.6 - 0.83)  
elif energy_total < 2.5:  
    energy[1] = (2.5-energy_total)/(2.5 - 1.6)  
else:  
    energy[1] = 0  
  
# energylistrik Tinggi  
if energy_total < 2.33:  
    energy[2] = 0  
elif energy_total < 3.66:  
    energy[2] = (energy_total-3.66)/(3.66 - 2.33)  
else:
```

```
energy[2] = 1
```

- g. Mengisi domain atau nilai keanggotan jumlah perangkat yang sudah ditentukan dengan tabel dan gambar kurva keanggotan pada landasan teori *Fuzzy*

```
if Sum_perangkat < 2:
    jumlahperangkat[0] = 1
elif Sum_perangkat < 4:
    jumlahperangkat[0] = (4 - Sum_perangkat)/(4 - 2)
else:
    jumlahperangkat[0] = 0

# jumlahperangkat umah sedang
if Sum_perangkat < 2:
    jumlahperangkat[1] = 0
elif Sum_perangkat < 4:
    jumlahperangkat[1] = (Sum_perangkat - 2)/(4 - 2)
elif Sum_perangkat < 6:
    jumlahperangkat[1] = 1
elif Sum_perangkat < 8:
    jumlahperangkat[1] = (8 - Sum_perangkat)/(8 - 6)
else:
    jumlahperangkat[1] = 0

# jumlahperangkat umah Banyak
if Sum_perangkat < 6:
    jumlahperangkat[2] = 0
elif Sum_perangkat < 8:
    jumlahperangkat[2] = (Sum_perangkat-6)/(8 - 6)
else:
    jumlahperangkat[2] = 1
```

- h. Mengisi domain atau nilai keanggotan daya listrik yang sudah ditentukan dengan tabel dan gambar kurva keanggotan pada landasan teori *Fuzzy* :

```
# inputan Daya Listrik
# Rendah  x  400 900
# Normal   400 900 1400
# Tinggi   900 1400  x

# Daya rendah
```

```

if Daya < 400:
    HasilDaya[0] = 1
elif Daya < 900:
    HasilDaya[0] = (900 - Daya)/(900 - 400)
else:
    HasilDaya[0] = 0

# HasilDayasedang
if Daya < 400:
    HasilDaya[1] = 0
elif Daya < 900:
    HasilDaya[1] = (Daya-400)/(900 - 400)
elif Daya < 1400:
    HasilDaya[1] = (1400 - Daya)/(1400-900)
else:
    HasilDaya[1] = 0

# HasilDaya tinggi
if Daya < 900:
    HasilDaya[2] = 0
elif Daya < 1400:
    HasilDaya[2] = (Daya-900)/(1400-900)
else:
    HasilDaya[2] = 1

```

- Mengisi domain atau nilai keanggotaan waktu penggunaan yang sudah ditentukan dengan tabel dan gambar kurva keanggotaan pada landasan teori Fuzzy:

```

# Inputan waktu Penggunaan
# Cepat 0-6 8
# Normal 6 8-12 14
# Lama 12 14-24 24

if waktu < 6:
    stopwatch[0] = 1
elif waktu < 8:
    stopwatch[0] = (8 - waktu)/(8 - 6)
else:
    stopwatch[0] = 0

# stopwatchumah sedang
if waktu < 6:

```

```

stopwatch[1] = 0
elif waktu < 8:
    stopwatch[1] = (waktu - 6)/(8 - 6)
elif waktu < 10:
    stopwatch[1] = 1
elif waktu < 12:
    stopwatch[1] = (12 - waktu)/(12 - 10)
else:
    stopwatch[1] = 0

# stopwatchumah Banyak
if waktu < 10:
    stopwatch[2] = 0
elif waktu < 12:
    stopwatch[2] = (waktu-10)/(12 - 10)
else:
    stopwatch[2] = 1

```

- j. Mengisi domain atau nilai keanggotan biaya listrik penggunaan yang sudah ditentukan dengan tabel dan gambar kurva keanggotan pada landasan teori *Fuzzy*:

```

# biaya listrik murah
if biaya < 4000:
    biayalistrik[0] = 1
elif biaya < 7000:
    biayalistrik[0] = (7000 - biaya)/(7000 - 4000)
else:
    biayalistrik[0] = 0

# biaya mahal
if biaya < 4000:
    biayalistrik[1] = 0
elif biaya < 7000:
    biayalistrik[1] = (biaya - 4000)/(7000 - 4000)
else:
    biayalistrik[0] = 1

```

- k. Berikut pengkodean deklarasi variabel pembagi tahap *deFuzzy*, lalu pembuatan aturan *Fuzzy* dengan fungsi implikasi menggunakan min() dan penjumlahan variabel pembagi untuk dapat digunakan pada *deFuzzifikasi*:

```
deFuzzy = 0
```

```

for i in range(3):
    for j in range(3):
        for k in range(3):
            for l in range(3):
                for m in range(2):
                    print('i:', i, ' j:', j, ' k:', k, 'l:', l,
'm:', m)
                    rules[i][j][k][l][m] = min(
                        stopwatch[j], Power[i], HasilDaya[k],
jumlahperangkat[l], biayalistrik[m])
                    deFuzzy += rules[i][j][k][l][m]

```

1. Kode Tahap berdasarkan aturan yang telah dibuat apakah data varibel itu masuk dalam output pengguna listrik rendah, sedang dan tinggi, hasil deFuzzy akan disimpan kedalam variabel z.

```

z = (
# 1
(rules[0][0][0][0][0] * penggunaan_Rendah)+(rules[0][0][1][0][0] *
penggunaan_Rendah)+(rules[0][0][2][0][0] * penggunaan_Rendah) +
(rules[0][1][0][0][0] * penggunaan_Rendah)+(rules[0][1][1][0][0] *
penggunaan_Rendah)+(rules[0][1][2][0][0] * penggunaan_Rendah) +
(rules[0][2][0][0][0] * penggunaan_Rendah)+(rules[0][2][1][0][0] *
penggunaan_Rendah)+(rules[0][2][2][0][0] * penggunaan_Rendah) +
(rules[1][0][0][0][0] * penggunaan_Rendah)+(rules[1][0][1][0][0] *
penggunaan_Rendah)+(rules[1][0][2][0][0] * penggunaan_Rendah) +
(rules[1][1][0][0][0] * penggunaan_normal)+(rules[1][1][1][0][0] *
penggunaan_normal)+(rules[1][1][2][0][0] * penggunaan_normal) +
(rules[1][2][0][0][0] * penggunaan_normal)+(rules[1][2][1][0][0] *
penggunaan_normal)+(rules[1][2][2][0][0] * penggunaan_normal) +
(rules[2][0][0][0][0] * penggunaan_Rendah)+(rules[2][0][1][0][0] *
penggunaan_Rendah)+(rules[2][0][2][0][0] * penggunaan_Rendah) +
(rules[2][1][0][0][0] * penggunaan_normal)+(rules[2][1][1][0][0] *
penggunaan_normal)+(rules[2][1][2][0][0] * penggunaan_normal) +
(rules[2][2][0][0][0] * penggunaan_tinggi)+(rules[2][2][1][0][0] *
penggunaan_tinggi)+(rules[2][2][2][0][0] * penggunaan_tinggi) +
# 2
(rules[0][0][0][1][0] * penggunaan_Rendah)+(rules[0][0][1][1][0] *
penggunaan_Rendah)+(rules[0][0][2][1][0] * penggunaan_Rendah) +
(rules[0][1][0][1][0] * penggunaan_Rendah)+(rules[0][1][1][1][0] *
penggunaan_Rendah)+(rules[0][1][2][1][0] * penggunaan_Rendah) +
(rules[0][2][0][1][0] * penggunaan_Rendah)+(rules[0][2][1][1][0] *
penggunaan_Rendah)+(rules[0][2][2][1][0] * penggunaan_Rendah) +

```

```

(rules[1][0][0][1][0] * penggunaan_Rendah)+(rules[1][0][1][1][0] *
penggunaan_Rendah)+(rules[1][0][2][1][0] * penggunaan_Rendah) +
(rules[1][1][0][1][0] * penggunaan_normal)+(rules[1][1][1][1][0] *
penggunaan_normal)+(rules[1][1][2][1][0] * penggunaan_normal) +
(rules[1][2][0][1][0] * penggunaan_normal)+(rules[1][2][1][1][0] *
penggunaan_normal)+(rules[1][2][2][1][0] * penggunaan_normal) +
(rules[2][0][0][1][0] * penggunaan_Rendah)+(rules[2][0][1][1][0] *
penggunaan_Rendah)+(rules[2][0][2][1][0] * penggunaan_Rendah) +
(rules[2][1][0][1][0] * penggunaan_normal)+(rules[2][1][1][1][0] *
penggunaan_normal)+(rules[2][1][2][1][0] * penggunaan_normal) +
(rules[2][2][0][1][0] * penggunaan_tinggi)+(rules[2][2][1][1][0] *
penggunaan_tinggi)+(rules[2][2][2][1][0] * penggunaan_tinggi) +

# 3

(rules[0][0][0][2][0] * penggunaan_Rendah)+(rules[0][0][1][2][0] *
penggunaan_Rendah)+(rules[0][0][2][2][0] * penggunaan_Rendah) +
(rules[0][1][0][2][0] * penggunaan_Rendah)+(rules[0][1][1][2][0] *
penggunaan_Rendah)+(rules[0][1][2][2][0] * penggunaan_Rendah) +
(rules[0][2][0][2][0] * penggunaan_Rendah)+(rules[0][2][1][2][0] *
penggunaan_Rendah)+(rules[0][2][2][2][0] * penggunaan_Rendah) +
(rules[1][0][0][2][0] * penggunaan_Rendah)+(rules[1][0][1][2][0] *
penggunaan_Rendah)+(rules[1][0][2][2][0] * penggunaan_Rendah) +
(rules[1][1][0][2][0] * penggunaan_normal)+(rules[1][1][1][2][0] *
penggunaan_normal)+(rules[1][1][2][2][0] * penggunaan_normal) +
(rules[1][2][0][2][0] * penggunaan_normal)+(rules[1][2][1][2][0] *
penggunaan_normal)+(rules[1][2][2][2][0] * penggunaan_normal) +
(rules[2][0][0][2][0] * penggunaan_Rendah)+(rules[2][0][1][2][0] *
penggunaan_Rendah)+(rules[2][0][2][2][0] * penggunaan_Rendah) +
(rules[2][1][0][2][0] * penggunaan_normal)+(rules[2][1][1][2][0] *
penggunaan_normal)+(rules[2][1][2][2][0] * penggunaan_normal) +
(rules[2][2][0][2][0] * penggunaan_tinggi)+(rules[2][2][1][2][0] *
penggunaan_tinggi)+(rules[2][2][2][2][0] * penggunaan_tinggi) +

# 1

(rules[0][0][0][0][1] * penggunaan_Rendah)+(rules[0][0][1][0][1] *
penggunaan_Rendah)+(rules[0][0][2][0][1] * penggunaan_Rendah) +
(rules[0][1][0][0][1] * penggunaan_Rendah)+(rules[0][1][1][0][1] *
penggunaan_Rendah)+(rules[0][1][2][0][1] * penggunaan_Rendah) +
(rules[0][2][0][0][1] * penggunaan_Rendah)+(rules[0][2][1][0][1] *
penggunaan_Rendah)+(rules[0][2][2][0][1] * penggunaan_Rendah) +
(rules[1][0][0][0][1] * penggunaan_Rendah)+(rules[1][0][1][0][1] *
penggunaan_Rendah)+(rules[1][0][2][0][1] * penggunaan_Rendah) +
(rules[1][1][0][0][1] * penggunaan_normal)+(rules[1][1][1][0][1] *
penggunaan_normal)+(rules[1][1][2][0][1] * penggunaan_normal) +
(rules[1][2][0][0][1] * penggunaan_normal)+(rules[1][2][1][0][1] *
penggunaan_normal)+(rules[1][2][2][0][1] * penggunaan_normal) +

```

```

(rules[2][0][0][0][1] * penggunaan_Rendah)+(rules[2][0][1][0][1] *
penggunaan_Rendah)+(rules[2][0][2][0][1] * penggunaan_Rendah) +
(rules[2][1][0][0][1] * penggunaan_normal)+(rules[2][1][1][0][1] *
penggunaan_normal)+(rules[2][1][2][0][1] * penggunaan_normal) +
(rules[2][2][0][0][1] * penggunaan_tinggi)+(rules[2][2][1][0][1] *
penggunaan_tinggi)+(rules[2][2][2][0][1] * penggunaan_tinggi) +  

# 2  

(rules[0][0][0][1][1] * penggunaan_Rendah)+(rules[0][0][1][1][1] *
penggunaan_Rendah)+(rules[0][0][2][1][1] * penggunaan_Rendah) +
(rules[0][1][0][1][1] * penggunaan_Rendah)+(rules[0][1][1][1][1] *
penggunaan_Rendah)+(rules[0][1][2][1][1] * penggunaan_Rendah) +
(rules[0][2][0][1][1] * penggunaan_Rendah)+(rules[0][2][1][1][1] *
penggunaan_Rendah)+(rules[0][2][2][1][1] * penggunaan_Rendah) +
(rules[1][0][0][1][1] * penggunaan_Rendah)+(rules[1][0][1][1][1] *
penggunaan_Rendah)+(rules[1][0][2][1][1] * penggunaan_Rendah) +
(rules[1][1][0][1][1] * penggunaan_normal)+(rules[1][1][1][1][1] *
penggunaan_normal)+(rules[1][1][2][1][1] * penggunaan_normal) +
(rules[1][2][0][1][1] * penggunaan_normal)+(rules[1][2][1][1][1] *
penggunaan_normal)+(rules[1][2][2][1][1] * penggunaan_normal) +
(rules[2][0][0][1][1] * penggunaan_Rendah)+(rules[2][0][1][1][1] *
penggunaan_Rendah)+(rules[2][0][2][1][1] * penggunaan_Rendah) +
(rules[2][1][0][1][1] * penggunaan_normal)+(rules[2][1][1][1][1] *
penggunaan_normal)+(rules[2][1][2][1][1] * penggunaan_normal) +
(rules[2][2][0][1][1] * penggunaan_tinggi)+(rules[2][2][1][1][1] *
penggunaan_tinggi)+(rules[2][2][2][1][1] * penggunaan_tinggi) +  

# 3  

(rules[0][0][0][2][1] * penggunaan_Rendah)+(rules[0][0][1][2][1] *
penggunaan_Rendah)+(rules[0][0][2][2][1] * penggunaan_Rendah) +
(rules[0][1][0][2][1] * penggunaan_Rendah)+(rules[0][1][1][2][1] *
penggunaan_Rendah)+(rules[0][1][2][2][1] * penggunaan_Rendah) +
(rules[0][2][0][2][1] * penggunaan_Rendah)+(rules[0][2][1][2][1] *
penggunaan_Rendah)+(rules[0][2][2][2][1] * penggunaan_Rendah) +
(rules[1][0][0][2][1] * penggunaan_Rendah)+(rules[1][0][1][2][1] *
penggunaan_Rendah)+(rules[1][0][2][2][1] * penggunaan_Rendah) +
(rules[1][1][0][2][1] * penggunaan_normal)+(rules[1][1][1][2][1] *
penggunaan_normal)+(rules[1][1][2][2][1] * penggunaan_normal) +
(rules[1][2][0][2][1] * penggunaan_normal)+(rules[1][2][1][2][1] *
penggunaan_normal)+(rules[1][2][2][2][1] * penggunaan_normal) +
(rules[2][0][0][2][1] * penggunaan_Rendah)+(rules[2][0][1][2][1] *
penggunaan_Rendah)+(rules[2][0][2][2][1] * penggunaan_Rendah) +
(rules[2][1][0][2][1] * penggunaan_normal)+(rules[2][1][1][2][1] *
penggunaan_normal)+(rules[2][1][2][2][1] * penggunaan_normal) +
(rules[2][2][0][2][1] * penggunaan_tinggi)+(rules[2][2][1][2][1] *
penggunaan_tinggi)+(rules[2][2][2][2][1] * penggunaan_tinggi)/deFuzzy

```

- m. Menentukan hasil pengambilan keputusan dari perhitungan z atau deFuzzyifikasi

```

print("Z adalah : "+str(z))

if z <= 0.67:
    return {
        "fuzy": round(z, 2),
        "text": "Penggunaan Rendah"
    }
elif z <= 1.33:
    return {
        "fuzy": round(z, 2),
        "text": "Penggunaan Sedang"
    }
else:
    return {
        "fuzy": round(z, 2),
        "text": "Penggunaan Tinggi"
}

```

Berikut kode Fuzzy yang dilakukan di Visual code dengan Bahasa Python yang sudah dibuat git yang dapat diakses publik;

<https://github.com/muzakirjohn/FuzzySugeniMonitoringlistrik>

#### 4.4.4 Pengkodean aplikasi android Sistem monitoring lsitrik

Pada tahap pengkodean aplikasi android seluruh rancangan diimplementasikan langsung. Sesuai dengan rancangan-rancangan yang telah dibuat. Pengkodean ini menggunakan Bahasa utama yaitu dart versi 3 dengan struktur penulisan *framework* yaitu Flutter versi 3, sehingga aplikasi ini disebut aplikasi berbasis Android. Segala kode program disimpan dan dapat diakses dalam git:

<https://github.com/muzakirjohn/MonitrongAndroidFlutter>

#### 4.5 Evaluasi dengan *Black Box Testing*

Setelah melakukan tahap pengkodean, peneliti melakukan pengujian terhadap hasil implementasi sistem. pengujian yang akan dilakukan ialah pengujian *black box testing* dalam uji hasil dari Fuzzy dan uji kesesuaian output Penguna listrik dari input yang diberikan.

#### 4.5.1 Pengujian Ardunio IDE

Pengujian *black box testing* pada Arduino IDE dengan menggunakan ESP8266, PZEM, dan mengirimkan data ke *Firebase* adalah langkah yang baik untuk memastikan bahwa sistem berfungsi dengan benar dan memenuhi kebutuhan fungsional , berikut tabel pengujian Ardunio IDE dengan *black box testing* :

Tabel 4. 4 Pengujian Ardunio IDE

	Fungsi	Input	Output diharapkan	OutPut hasil	Review Hasil
1	ESP8266	<i>include username</i> dan <i>password WiFi</i>	Terkoneksi dengan IP WiFi.	✓ Berhasil	Berhasil Terkoneksi dengan IP
		<i>Include sensor PZEM004T</i> 3	Terhubung dengan Sensor PZEM004T	✓ Berhasil	Berhasil terhubung dengan 3 sensor
			Mendapatkan data sensor listrik dari PZEM	✓ Berhasil	Berhasil mendapatkan data sensor listrik
			Mendapatkan data dengan <i>realtime</i>	✓ Berhasil	Berhasil Mendapatkan data dengan <i>realtime</i>
2	<i>Firebase</i>	<i>Include</i> dan deklarasi <i>Auth firebase</i> dengan <i>library</i> -nya menggunakan <i>API key</i> dan <i>HTTP</i>	Terhubung dengan ESP8266	✓ Berhasil	Berhasil terhubung dengan <i>Firebase</i>
			Mengirimkan data sensor dari ESP8266	✓ Berhasil	Data sensor terkirim
3	PZEM004T	<i>Include sensor</i> 3	Terhubung dengan ESP8266	✓ Berhasil	Berhasil terhubung dengan ESP8266

		deklarasi variable komponen sensor PZEM004T	Memabaca sensor dari arus listrik di stop kontak yang sudah di SetUp	✓ Berhasil	Berhasil Membaca dari arus listrik di stop kontak
			Memabaca sensor secara realtime	✓ Berhasil	Berhasil Membaca sensor secara realtime

#### 4.5.2 Pengujian *Firebase*

Pengujian black box testing pada *Firebase* dalam penelitian sistem monitoring listrik bertujuan untuk memastikan bahwa *Firebase* berfungsi dengan baik dalam menyimpan, mengambil, dan memproses data monitoring listrik secara akurat dan efisien. Berikut adalah tabel pengujian yang dilakukan:

Tabel 4. 5 Pengujian Blcak box testing *Firebase*

No	Kasus Uji	Deskripsi	Hasil Pengujian	Keterangan
1	Integrasi ESP8266 dan aplikasi android	Mengecek keterhubungan antara ESP8266, <i>Firebase</i> , dan aplikasi Android	Koneksi berhasil terjalin	<i>Firebase</i> dapat terhubung dengan ESP8266 dan aplikasi Android
2	Pengiriman Data Sensor ke <i>Firebase</i>	Mengirimkan data sensor ke <i>Firebase</i> Firestore	Data sensor diterima	Data sensor berhasil dikirimkan ke <i>Firebase</i> Firestore
3	Pembaruan Data Monitoring secara realtime dan autimatis	Memperbarui data monitoring di Firestore	Data berhasil diperbarui	Data monitoring diperbarui di Firestore

4	Query Data Monitoring	Melakukan query untuk mendapatkan data monitoring dari Firestore	Data berhasil ditemukan	Query berhasil mengembalikan data monitoring
5	Grafik Visualisasi Data	Menampilkan grafik visualisasi data monitoring listrik dari Firestore	Grafik berhasil ditampilkan	Grafik visualisasi data monitoring listrik muncul dengan benar
6	Tampilan Tabel Data	Menampilkan tabel data monitoring listrik dari Firestore	Tabel berhasil ditampilkan	Tabel data monitoring listrik ditampilkan dengan benar
7	Keterhubungan dengan Aplikasi Android	Mengecek keterhubungan antara <i>Firebase Firestore</i> dan aplikasi Android	Koneksi berhasil terjalin	Aplikasi Android dapat terhubung dan berkomunikasi dengan <i>Firebase</i>
8	Integrasi dengan Komponen ESP8266	Mengintegrasikan <i>Firebase Firestore</i> dengan komponen esp8266	Integrasi berhasil dilakukan	<i>Firebase Firestore</i> dapat berintegrasi dengan komponen hardware eksternal
9	Error Handling	Menguji penanganan kesalahan dan kesalahan yang terjadi	Aplikasi mengatasi kesalahan dengan baik	Pesan kesalahan yang tepat ditampilkan dan masalah diatasi dengan baik
10	Skalabilitas <i>Firebase Firestore</i>	Menguji kemampuan <i>Firebase Firestore</i> dalam menangani jumlah data yang besar	<i>Firebase</i> mampu menangani	<i>Firebase Firestore</i> dapat menangani dengan baik jumlah data yang besar

#### 4.5.2 Pengujian Aplikasi Android

Pengujian black box pada aplikasi Android adalah metode pengujian di mana aplikasi diuji tanpa pengetahuan tentang struktur internal atau implementasi kode aplikasi. Pada dasarnya, pengujian ini bertujuan untuk menguji aplikasi dari sudut pandang pengguna akhir, tanpa memperhatikan detail implementasi internal.

Tabel 4. 6 Pengujian Aplikasi Android

No	Kasus Uji	Deskripsi	Hasil Pengujian	Keterangan
1	Login dengan akun valid	Memasukkan username dan password yang valid	Berhasil login	Pengguna dapat masuk ke dalam aplikasi
2	Login dengan akun tidak valid	Memasukkan username dan password yang tidak valid	Gagal login	Aplikasi menampilkan pesan kesalahan atau validasi gagal
3	Pemilihan 3 sampel alat yang dicolok	Memilih Alat yang akan memperoleh hasil data listrik secara realtime	Data listrik ditampilkan sesuai dengan pilihan	aplikasi menampilkan data listrik dari pilihan alat
4	Tampilan Data Listrik Realtime	Memperoleh data listrik secara realtime	Data listrik ditampilkan dengan benar	Data listrik diperbarui secara real-time dengan benar
5	Pemilihan daya listrik	Memilih daya listrik (900,1300,2200) untuk menentukan perkalian biaya listrik	Biaya listrik mengikuti tarif dari daya listrik yang dipilih	Aplikasi menampilkan data biaya listrik sesuai pilihan daya listrik
6	Pemilihan Tanggal Data Listrik	Memilih tanggal untuk melihat data listrik	Data listrik sesuai dengan tanggal yang dipilih	Aplikasi menampilkan data listrik sesuai

				permintaan pengguna
7	Grafik Output <i>Fuzzy Sugeno</i>	Menampilkan grafik hasil output <i>Fuzzy Sugeno</i> (rendah, sedang, tinggi)	Grafik ditampilkan dengan benar	Grafik menggambarkan kategori data listrik dengan jelas
8	Tabel Data Listrik sesuai Permintaan Tanggal	Melihat tabel data listrik berdasarkan permintaan tanggal	Tabel data listrik ditampilkan sesuai permintaan pengguna	Aplikasi menampilkan data listrik dalam bentuk tabel sesuai permintaan
9	Perhitungan Jumlah kWh dan Biaya Listrik	Melihat jumlah kWh dan biaya listrik berdasarkan data yang ditampilkan	Perhitungan yang benar ditampilkan	Aplikasi menghitung jumlah kWh dan biaya listrik dengan benar
10	Logout aplikasi android	Menekan button “logout” untuk menutup aplikasi	Berhasil Menutup aplikasi	Aplikasi ditutup dengan baik
11	Responsivitas UI	Berinteraksi dengan elemen UI seperti tombol dan tampilan lainnya	UI merespons dengan tidak begitu cepat ketika permintaan data yang dianalisis banyak.	Tampilan dan respons aplikasi sesuai dengan aksi pengguna

#### 4.5.3 Pengujian *Fuzzy Logic*

*black box testing* akan dilakukan pada *Fuzzy sugeno* dengan membandingkan output yang diharapkan terhadap output hasil uji berdasarkan input case yang diberikan. Output yang diharapkan didapatkan dengan menggunakan perhitungan manual sedangkan output hasil uji didapat berdasarkan hasil komputasi dari *Python visual studio code*.

Tabel 4. 7 black box testing

Uji ke	Data Uji					Output Fuzzy yang dihitung manual (sesuai rumus)	Output hasil uji Fuzzy
	Waktu	Energi	Daya	Biaya listrik	Jumlah perangkat		
1	7	0.7	900	2500	2	0	0
2	24	2.134	900	2500	3	1.53	1.53
3	5	4.87	900	7000	1	0	0
4	24	5.87	1300	7000	1	2	2
5	24	5.87	900	5000	3	2	2
6	11	2.56	900	5000	2	1.5	1.5

Berdasarkan hasil Uji variable input sistem monitoring, keenam hasil tes yang diharapkan sudah bernilai sama dengan yang dihasilkan oleh perhitungan oleh *python Visual code*. Ini menunjukkan bahwa *logika Fuzzy* dengan metode sugeno yang berada pada *python Visual code* tersebut telah sesuai dengan yang diharapkan.

Perhitungan Manual untuk Data Uji ke 2 pada sistem monitoring listrik adalah sebagai berikut;

Diketahui *Crisp input*:

Waktu	: 24
Energi	: 2.134
Daya	: 900
Biaya lsitrik	: 2500
Jumlah Perangkat	: 3

a. Hitungan *Fuzzy* input variable

- Hitungan nilai keanggotaan waktu

Crisp input = 24

Input Fuzzy Waktu Lama = 1

- Hitungan nilai keanggotaan KwH

Crisp input = 2.134

Input Fuzzy KwH sedang =

$$= \frac{2.5 - 2.134}{2.5 - 1.66}, \quad 1.66 \leq x \leq 2.5$$

$$= \frac{0.366}{0.84}$$

$$= 0.44$$

Input Fuzzy KwH Tinggi =

$$= \frac{2.134 - 1.66}{2.5 - 1.66}, \quad 1.66 \leq x \leq 2.5$$

$$= \frac{0.474}{0.84}$$

$$= 0.56$$

- Hitungan nilai keanggotaan Daya listrik

Crisp input = 900

Input Fuzzy daya listrik rendah = 1

- Hitungan nilai keanggotaan Biaya listrik

Crisp input = 2500

I Input Fuzzy Biaya listrik Normal = 1

- Hitungan nilai keanggotaan jumlah perangkat

Crisp input = 3

Input Fuzzy jumlah perangkat sedikit =

$$\begin{aligned}
 &= \frac{4-3}{4-2} \\
 &= \frac{1}{2} \\
 &= 0.5
 \end{aligned}$$

Input Fuzzy jumlah perangkat sedang =

$$\begin{aligned}
 &= \frac{3-2}{4-2} \\
 &= \frac{1}{2} \\
 &= 0.5
 \end{aligned}$$

#### b. Inferensi Fuzzy

Pada tahap ini Fuzzy input akan diproses di tahapan aturan Fuzzy dengan fungsi operator *min*, dari 162 rules didapatkan yang sudah diinferensikan maka rules yang digunakan hanya 4 rules dengan kondisi input Fuzzy yang didapatkan pada proses sebelumnya, berikut rules Fuzzy :

Tabel 4. 8 Rules pengujian Fuzzy

Rules	Waktu		Daya		KWH		Biaya		Jumlah Perangkat		PENGGUNAAN
22	Lama (1)	Jika	Rendah (1)	Jika	SEDANG (0.44)	Jika	Normal (1)	Jika	Sedikit (0.5)	Maka	SEDANG(0.44)
25	Lama (1)	Jika	Rendah (1)	Jika	TINGGI (0.56)	Jika	Normal (1)	Jika	Sedikit (0.5)	Maka	TINGGI(0.5)
52	Lama (1)	Jika	Rendah (1)	Jika	TINGGI (0.56)	Jika	Normal (1)	Jika	Sedang (0.5)	Maka	TINGGI(0.5)
49	Lama (1)	Jika	Rendah (1)	Jika	SEDANG (0.44)	Jika	Normal (1)	Jika	Sedang (0.5)	Maka	SEDANG(0.44)

Dari hasil rules Fuzzy peneliti mendapatkan Fuzzy output yaitu :

Tabel 4. 9 Fuzzy Output pengujian

Penggunaan	Fuzzy Output
SEDANG(1)	0.44( $\alpha_1$ )
TINGGI(2)	0.5( $\alpha_2$ )

#### c. DeFuzzifikasi

$$Z = \frac{(\alpha_1 \times Z_1) + \dots + (\alpha_{162} \times Z_{162})}{\alpha_1 + \dots + \alpha_{162}}$$

$$Z = \frac{(0.44 \times 1) + (0.5 \times 2)}{0.44 + 0.5}$$

$$Z = \frac{0.585 + 1}{0.94}$$

$$Z = \frac{1.5852}{0.94}$$

$$Z = 1.53$$

Pada teori landasan *defuzzifikasi* bahwa  $Z = 1.53$  berada diposisi atau *range* jenis pengguna tinggi karena masuk dalam *range*  $1.33 < x < 2$ .

#### 4.5.4 Pengujian Output Terhadap Input Berdasarkan Aturan Fuzzy

Testing ini dilakukan untuk melihat apakah input dari semua data dapat menghasilkan output kategori pengguna listrik yang sesuai. Input tanah akan dibagi kedalam dua kategori yaitu kering dan basah. Batasan-batasan pada kategori input ini dapat dilihat pada tabel 4.3 pada pembahasan implementasi logika *Fuzzy*. Input ini akan dicocokkan kedalam aturan *Fuzzy* untuk menentukan nilai dari output yang diharapkan. Output hasil uji didapatkan saat uji coba alat dijalankan.

Tabel 4. 10 Output terhadap input berdasarkan aturan Fuzzy

Uji ke	Data Uji					Output Pengguna listrik diharapkan	Output Pengguna listrik uji
	Waktu	Energi	Daya	Biaya listrik	Jumlah perangkat		
1	7	0.7	900	2500	2	Pengguna Rendah	Pengguna Rendah
2	24	2.134	900	2500	3	Pengguna Tinggi	Pengguna Tinggi
3	5	4.87	900	7000	1	Pengguna Rendah	Pengguna Rendah
4	24	5.87	1300	7000	1	Pengguna Tinggi	Pengguna Tinggi
5	24	5.87	900	5000	3	Pengguna Tinggi	Pengguna Tinggi
6	11	2.56	900	5000	2	Pengguna Tinggi	Pengguna Tinggi

Berdasarkan hasil testing pada hasil python visual code output pengguna listrik yang diharapkan telah sesuai dengan yang dihasilkan oleh python visual code. Seperti ketika waktu lama , energi listrik tinggi , daya sedang , jumlah perangkat banyak,biaya listrik dan Output pengguna

listrik Tinggi sesuai dengan aturan *Fuzzy* yang telah dibuat. Dari keenam testing yang dilakukan pada sistem tersebut. Semuanya menghasilkan hasil yang sesuai dengan yang diharapkan. Ini menunjukkan bahwa alat ini sudah sesuai dengan yang dibutuhkan dan dapat beroperasi dengan baik.

#### 4.5.5 Pengujian Lokasi Sistem monitoring listrik

Testing ini dilakukan untuk menguji sejauh mana pengguna bisa memantau daya listrik dari alat hardware dengan posisi hardware dirumah pengguna listrik dengan menggunakan blackbox testing

Tabel 4. 11 Pengujian Lokasi

4 Lokasi	Uji Kasus	Deskripsi	Hasil Pengujian
1.Jalan raya tengah No 39 RT004/RW009 Kelurahan tengah, kec kramat jati, Jakarta timur	<i>Firebase</i>	Memperoleh data listrik secara realtime	Berhasil memperoleh data realtime di lokasi
2. (Kos 1001) Jln. Inpres No. 80 Cirendeuy, Ciputat Timur, Tangerang Selatan, Banten		Melihat volume data sensor listrik di panel view <i>firebase</i>	Pengguna dapat melihat volume data di panel view <i>firebase</i> sesuai lokasi dengan data realtime
3. Kampus 1 UIN Jakarta ,Tangerang selatan , Banten		Melihat grafik monitor rules di <i>firebase</i>	Pengguna dapat melihat grafik rules di <i>firebase</i> dengan data realtime
4. Jalan Bougenville Atas RT 03 RW 01 No. 97 Serua Ciputat Tangerang Selatan			

	Aplikasi android	memonitoring data listrik(KwH,power,arus) dengan realtime	Berhasil monitor data listrik dengan realtime
		Memilih tanggal untuk melihat data yang dianalisis atau diperlihatkan	Dapat memilih tanggal untuk melihat data analisis
		Mengetahui analisis data berupa grafik dan tabel	Berhasil Mengetahui analisis data berupa grafik dan tabel
		Mengetahui total KwH dan biaya listrik sesuai <i>req</i> tanggal	Dapat mengetahui total KwH dan biaya listrik by <i>req date</i>

#### 4.5.6 Pengujian Gangguan Sistem

Testing ini dilakukan untuk menguji gangguan yang terjadi pada sistem yaitu berupa alat *hardware*, *firebase* dan aplikasi android yang berada pada di rumah pengguna listrik dengan menggunakan *blackbox testing* , berikut tabel pengujian sistem alat hardware;

Tabel 4. 12 Pengujian gangguan sistem

No	Gangguan	Uji Kasus	Deskripsi	Hasil Pengujian
1	Listrik Padam	Alat <i>Hardware</i>	Terjadi pemadaman pada listrik rumah tangga	<i>alat hardware</i> ikut padam dikarena alat tersebut membutuhkan <i>supplier</i> daya listrik
				Alat <i>hardware</i> tidak membaca listrik yang

				masuk dikarenakan pemadaman listrik
				Alat <i>hardware/microntroller</i> Tidak mengirimkan data ke <i>firebase</i>
	Firebase	Terjadi pemadaman pada listrik rumah tangga		<i>Firebase</i> tidak mendapatkan data kiriman dari <i>microcontroller</i> secara <i>realtime</i>
	Aplikasi Android	Terjadi pemadaman pada listrik rumah tangga		Tidak dapat melihat data secara <i>realtime</i> namun bisa melakukan analisis dengan tanggal sebelumnya
Listrik Pasca padam	Alat <i>hardware</i>	Alat <i>hardware</i> peneliti pasca pemadaman listrik		Alat hardware hidup kembali tetapi tidak berfungsi
		Melakukan tindakan penormalan fungsi alat hardware agar berfungsi kembali jika alat tersebut di <i>off</i> – kan di saklarnya dan dicabut juga <i>supplier</i> daya alat harwarenya, lalu colok kembali <i>supplier</i> daya alat harwarenya lalu di <i>ON</i> kan		<i>Micrtoller</i> dapat hasil sensor
	Firebase	Terjadi listrik pasca padam dan sudah melakukan penormalan kembali di hardware		<i>Sensor</i> dapat membaca data listrik <i>Micrtoller</i> dapat mengirimkan data liustrik ke <i>firebase</i>
				<i>Firebase</i> mendapatkan data terusan dari sebelum padam listrik dan data baru secara <i>realtime</i> .

		Aplikasi android	Terjadi listrik pasca padam dan sudah dialihkan fungsi kembali di hardware	dapat melihat data secara <i>realtime</i>
WiFi rumah pengguna tidak aktif atau bermasalah		Alat <i>Hardware</i>	Wifi mati berpengaruh pada alat <i>hardware</i>	Alat <i>hardware</i> atau <i>microcontroller</i> tidak terhubung dengan internet <i>Microcontroller</i> tidak bisa memebaca sensor listrik <i>Microcontroller</i> tidak bisa mengirimkan data ke <i>firebase</i> Sensor tidak bisa membaca tetapi aktif
		<i>Firebase</i>	<i>Microcontroller</i> tidak terhubung internet	<i>Firebase</i> tidak mendapatkan data <i>realtime</i>
		Aplikasi android	Akibat dari WiFi local bermasalah atau tidak aktif di rumah pengguna berdampak di aplikasi android	Aplikasi android tidak dapat memberikan pantauan secara <i>realtime</i>
WiFi rumah pengguna aktif kembali		Alat <i>Hardware</i>	Wifi rumah pengguna aktif dan berpengaruh ke alat <i>hardware</i>	Alat <i>hardware</i> atau <i>microcontroller</i> terhubung dengan internet secara automatis <i>Microcontroller</i> bisa memebaca sensor listrik <i>Microcontroller</i> bisa mengirimkan data ke <i>firebase</i> Sensor bisa membaca
		<i>Firebase</i>	Wifi rumah pengguna aktif dan berpengaruh ke <i>firebase</i>	<i>Firebase</i> mendapatkan data <i>realtime</i>

		Aplikasi android	Wifi rumah pengguna aktif dan berpengaruh ke aplikasi android	Aplikasi android tidak dapat memberikan pantauan secara <i>realtime</i>
--	--	------------------	---	---



## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Sistem Monitoring listrik

Sistem monitoring listrik yang telah dibangun memiliki landasan teori yang kuat dan telah berhasil dijalankan dengan sukses. Melalui sistem ini, pengguna listrik dapat memperoleh informasi tentang penggunaan energi listrik dalam beberapa hari sesuai dengan tanggal yang diminta. Selain itu, pengguna juga dapat mengetahui harga listrik, jumlah perangkat yang digunakan, dan waktu pemakaian melalui tabel yang tersedia dalam sistem monitoring. Total tagihan listrik juga dapat ditampilkan kepada pengguna.

Sistem monitoring ini juga memberikan data realtime kepada pengguna, yang memungkinkan mereka untuk melihat informasi aktual tentang penggunaan listrik saat ini. Selain itu, pengguna dapat mengetahui kategori pengguna listrik yang mereka masuki. Kategori ini ditentukan melalui proses *Fuzzy logic* menggunakan metode Sugeno, yang mempertimbangkan energi, waktu, daya listrik, jumlah perangkat, dan biaya listrik yang telah dikumpulkan oleh sistem monitoring pada setiap harinya.

Dengan demikian, sistem monitoring listrik ini memberikan pengguna listrik kemampuan untuk mengelola dan memantau penggunaan listrik mereka dengan lebih efektif. Mereka dapat melihat data historis, melacak biaya listrik, dan memahami kategori penggunaan listrik mereka. Hal ini memungkinkan pengguna untuk mengambil langkah-langkah yang tepat dalam mengoptimalkan penggunaan energi dan mengontrol pengeluaran mereka.

#### 5.2 Internet Of Things

Hasil yang diperoleh melibatkan implementasi sistem monitoring daya listrik berbasis IoT dengan menggunakan algoritma *Fuzzy Logic* Sugeno dan *Firebase* sebagai platform penyimpanan data. Berikut adalah deskripsi hasil penelitian yang dapat dijelaskan, Sistem monitoring daya listrik diimplementasikan menggunakan perangkat keras ESP8266 sebagai mikrokontroler dan sensor daya listrik untuk mengukur konsumsi listrik. Sistem ini juga mencakup perangkat lunak Android sebagai antarmuka pengguna.

ESP8266 terhubung ke *Firebase* menggunakan API Key yang disediakan oleh *Firebase*. API Key ini memungkinkan ESP8266 untuk mengirimkan data listrik ke *Firebase* secara langsung. ESP8266 menggunakan sensor daya listrik untuk mengukur konsumsi listrik. Data yang dikumpulkan kemudian diunggah ke *Firebase*, di mana *Firebase* menyimpan data tersebut dalam format yang sesuai untuk pengelolaan dan analisis lebih lanjut. Data listrik yang telah dikumpulkan

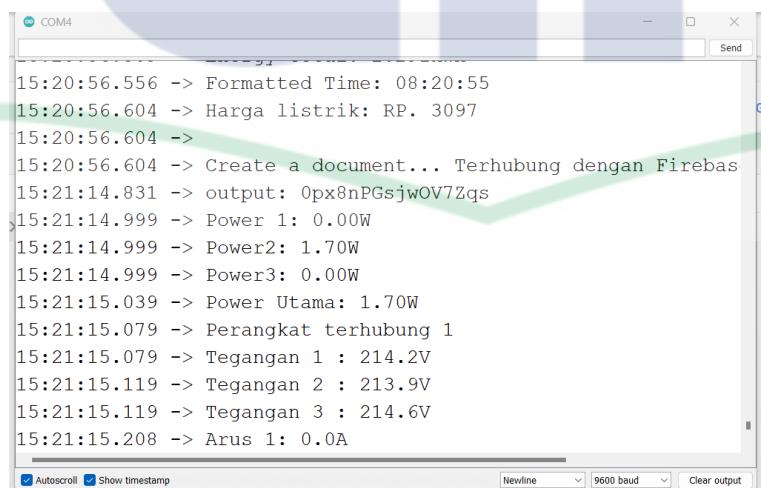
dianalisis menggunakan algoritma *Fuzzy Logic* Sugeno. Algoritma ini menggunakan aturan *Fuzzy* untuk memproses data input dan menghasilkan output yang berhubungan dengan penggunaan listrik.

Sistem ini juga mencakup aplikasi Android yang berfungsi sebagai antarmuka pengguna. Aplikasi ini memungkinkan pengguna untuk memantau data listrik secara real-time dan melihat data historis. Pengguna dapat melihat informasi tentang penggunaan listrik saat ini atau memilih rentang waktu tertentu untuk melihat data historis. Hasil penelitian dievaluasi untuk mengukur kinerja sistem monitoring daya listrik. Evaluasi ini meliputi akurasi pengukuran, respons sistem terhadap perubahan kondisi listrik, dan kegunaan antarmuka pengguna Android. Manfaat dari sistem ini meliputi pemantauan efisiensi energi, identifikasi pola penggunaan yang tidak efisien, dan potensi penghematan energi.

Penelitian ini juga mengidentifikasi beberapa pembatasan dalam implementasi sistem, seperti keterbatasan daya komputasi ESP8266 atau skalabilitas sistem untuk penggunaan yang lebih luas. Selain itu, penelitian ini menyoroti peluang pengembangan di masa depan, seperti integrasi dengan teknologi lain atau pengembangan algoritma *Fuzzy* yang lebih canggih. Dengan demikian, hasil penelitian ini memberikan kontribusi dalam pengembangan sistem monitoring daya listrik IoT dengan memanfaatkan algoritma *Fuzzy Logic* Sugeno dan platform *Firebase* untuk penyimpanan data.

### 5.3 Hasil Arduino IDE

Hasil Arduino Ide ini berupa hasil dari pengkodean di dalam aplikasi Arduino IDE yang berupa data yang diambil dari sensor listrik yaitu *PZEM004T* yang dikoneksikan di ESP8266 sebagai mikrokontroller. Berikut hasil dari Arduino IDE berupa gambar Serial Monitor:



```

COM4
15:20:56.556 -> Formatted Time: 08:20:55
15:20:56.604 -> Harga listrik: RP. 3097
15:20:56.604 ->
15:20:56.604 -> Create a document... Terhubung dengan Firebas
15:21:14.831 -> output: 0px8nPgsjwOV7Zqs
15:21:14.999 -> Power 1: 0.00W
15:21:14.999 -> Power2: 1.70W
15:21:14.999 -> Power3: 0.00W
15:21:15.039 -> Power Utama: 1.70W
15:21:15.079 -> Perangkat terhubung 1
15:21:15.079 -> Tegangan 1 : 214.2V
15:21:15.119 -> Tegangan 2 : 213.9V
15:21:15.119 -> Tegangan 3 : 214.6V
15:21:15.208 -> Arus 1: 0.0A

```

The screenshot shows the Arduino Serial Monitor window titled "COM4". It displays a series of text messages representing sensor data. The messages include formatted time, electricity price, a note about connecting to Firebase, and various power and voltage measurements. At the bottom of the monitor, there are checkboxes for "Autoscroll" and "Show timestamp", and dropdown menus for "Newline", "9600 baud", and "Clear output".

Gambar 5. 1 Serial monitor part 1

```

COM3
20:06:47.340 -> Connecting to Wi-Fi.....
20:06:53.454 -> Connected with IP: 192.168.1.15
20:06:53.502 ->
20:06:53.502 -> Firebase Client v4.3.4
20:06:53.502 ->
20:06:53.502 -> Token info: type = id token (GITKit token), status = on request
20:06:57.607 -> Token info: type = id token (GITKit token), status = ready
20:06:57.646 -> output: XlawdftxImHjufxG

```

*Gambar 5. 2 Serial Monitor Part 2*

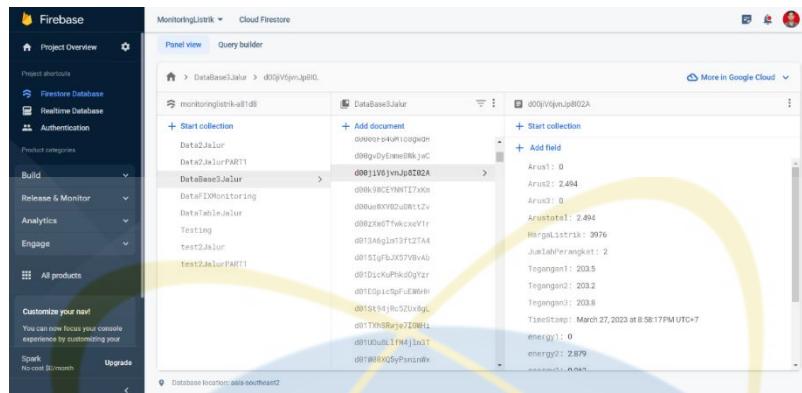
Pada gambar diatas, gambar yang dimaksud adalah serial Monitor yang berjalan di Arduino IDE, menampilkan output yang dihasilkan dari kode yang telah dikodekan di dalamnya. Serial Monitor ini menunjukkan bahwa Anda telah berhasil menampilkan daya, energi, dan arus listrik yang diambil secara langsung oleh sensor PZEM004T. Serial Monitor akan menampilkan data yang diambil oleh sensor PZEM004T, seperti daya (*power*), energi, dan arus listrik. Data ini memberikan gambaran tentang konsumsi listrik yang sedang diukur.

Serial Monitor juga memberikan informasi bahwa ESP8266 telah terhubung dengan jaringan WiFi yang ditunjukkan oleh pesan yang muncul. Ini menandakan bahwa modul ESP8266 dapat terhubung ke internet melalui WiFi. Selain itu, serial Monitor juga memberikan informasi bahwa data yang didapatkan akan dikirimkan ke *Firebase*. Ini menunjukkan bahwa pengguna telah mengatur koneksi antara ESP8266 dan *Firebase* untuk mengirimkan data listrik yang diukur.

Serial Monitor adalah alat penting dalam proses pengembangan dan pemecahan masalah dalam pengkodean Arduino. Melalui Serial Monitor, Anda dapat memantau dan memeriksa data yang dikirimkan dan menerima pesan untuk memvalidasi koneksi dan pengukuran yang dilakukan.

#### 5.4 Hasil *Firebase*

Pada hasil ini peneliti menunjukkan data listrik yang sudah masuk di *Firebase*. Berikut gambar hasil *Firebase* :

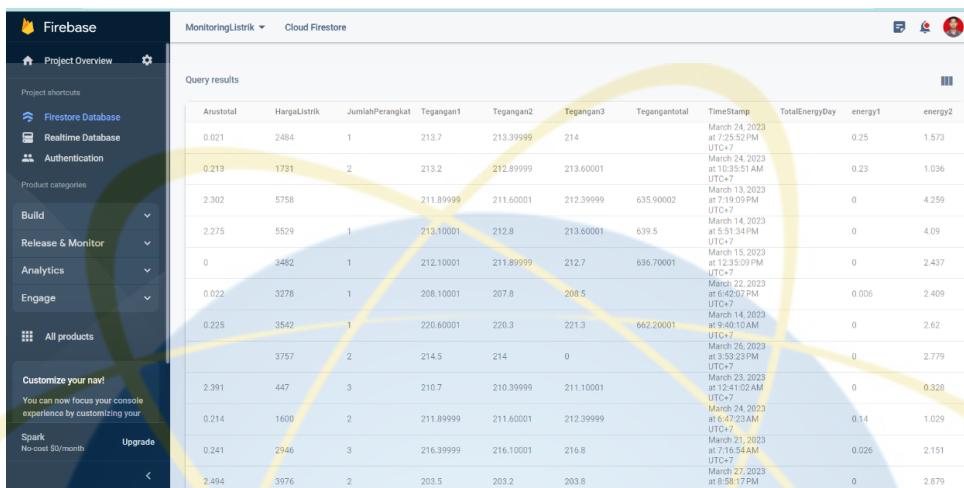


Gambar 5. 3 Hasil *Firebase* (panel view)

menggambarkan tampilan panel view dalam *Firebase* yang menunjukkan dokumen yang dibuat secara otomatis melalui koneksi dengan Arduino IDE. Tampilan ini memberikan bukti bahwa data listrik yang dikirimkan dari ESP8266 dengan sensor PZEM004T berhasil diterima oleh *Firebase* dan diubah menjadi dokumen secara real-time. Tampilan panel view menampilkan daftar dokumen yang dibuat secara otomatis. Setiap dokumen mewakili data listrik yang dikirimkan oleh ESP8266 dengan sensor PZEM004T. Dokumen-dokumen ini dapat disusun berdasarkan waktu penerimaan atau parameter lainnya yang relevan.

Saat pengguna memilih dokumen tertentu, tampilan panel view dapat menampilkan isi dokumen tersebut. Ini termasuk informasi seperti nilai daya (power), arus (current), tegangan (voltage), atau parameter lainnya yang diukur oleh sensor PZEM004T. Informasi ini dapat disajikan dalam bentuk tabel atau tampilan yang terstruktur sesuai dengan kebutuhan aplikasi. Real-time Update: Tampilan tersebut menunjukkan bahwa dokumen-dokumen ini diperbarui secara real-time ketika ada data baru yang diterima dari ESP8266. Dengan demikian, pengguna

dapat melihat data listrik terkini dalam dokumen-dokumen tersebut tanpa perlu melakukan input manual.



The screenshot shows the Firebase Cloud Firestore interface with a table titled "Query results". The table contains data from a collection named "MonitoringListrik". The columns are: ArusTotal, HargaListrik, JumlahPerangkat, Tegangan1, Tegangan2, Tegangan3, TeganganTotal, TimeStamp, TotalEnergyDay, energy1, and energy2. The data rows represent various measurements over time, with timestamps indicating when each entry was recorded.

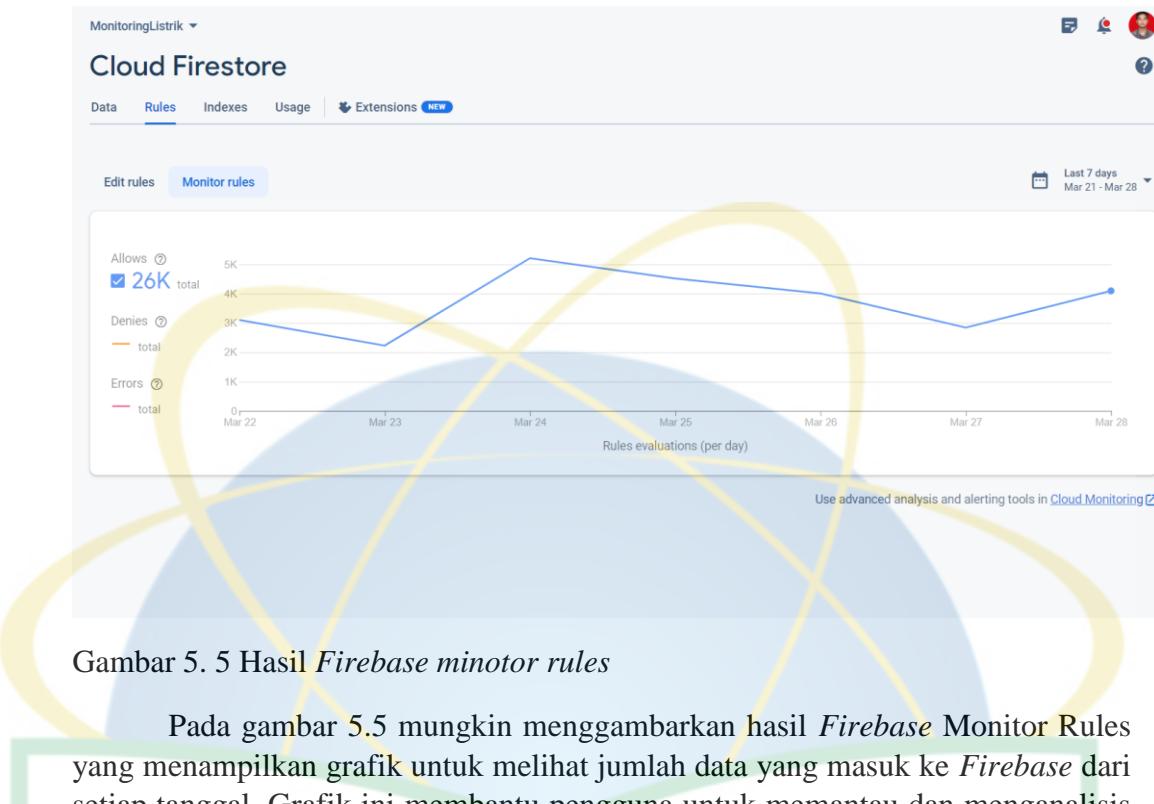
ArusTotal	HargaListrik	JumlahPerangkat	Tegangan1	Tegangan2	Tegangan3	TeganganTotal	TimeStamp	TotalEnergyDay	energy1	energy2
0.021	2484	1	213.7	213.39999	214		March 24, 2023 at 7:25:52 PM UTC+7	0.25	1.573	
0.213	1731	2	213.2	212.89999	213.60001		March 24, 2023 at 7:35:51 AM UTC+7	0.23	1.036	
2.302	5758		211.89999	211.60001	212.39999	635.90002	March 13, 2023 at 7:19:09 PM UTC+7	0	4.259	
2.275	5529	1	213.10001	212.8	213.80001	639.5	March 14, 2023 at 5:51:34 PM UTC+7	0	4.09	
0	3482	1	212.10001	211.89999	212.7	636.70001	March 15, 2023 at 12:30:09 PM UTC+7	0	2.437	
0.022	3278	1	208.10001	207.8	208.5		March 22, 2023 at 4:42:07 PM UTC+7	0.006	2.409	
0.225	3542	1	220.60001	220.3	221.3	662.20001	March 14, 2023 at 10:10:00 AM UTC+7	0	2.62	
	3757	2	214.5	214	0		March 26, 2023 at 3:53:23 PM UTC+7	0	2.779	
2.391	447	3	210.7	210.39999	211.10001		March 23, 2023 at 12:41:02 AM UTC+7	0	0.328	
0.214	1600	2	211.89999	211.60001	212.39999		March 24, 2023 at 6:47:23 AM UTC+7	0.14	1.029	
0.241	2946	3	216.39999	216.10001	216.8		March 21, 2023 at 10:54:44 AM UTC+7	0.026	2.151	
2.494	3976	2	203.5	203.2	203.8		March 27, 2023 at 8:58:17 PM UTC+7	0	2.879	

Gambar 5.4 Hasil Firebase (*Query builder*)

Gambar 5.4 mungkin menggambarkan tampilan tabel yang menampilkan data listrik sesuai dengan dokumen yang diberikan. Tampilan tabel ini memberikan gambaran yang lebih terstruktur dan terorganisir tentang data yang diterima oleh *Firebase* dari ESP8266 dengan sensor PZEM004T. Tampilan tabel akan mencakup beberapa kolom yang mewakili atribut atau parameter data listrik yang diukur. Ini bisa termasuk kolom seperti tanggal, *timestamp*, daya (*power*), arus (*current*), tegangan (*voltage*), atau atribut lainnya yang relevan dengan sistem monitoring listrik.

Setiap baris dalam tabel akan mewakili entri data tunggal yang diterima dari ESP8266. Baris ini akan berisi nilai-nilai yang sesuai dengan kolom-kolom yang telah disebutkan sebelumnya. Tampilan tabel mungkin juga menyediakan fasilitas untuk membangun dan menjalankan kueri terhadap data yang ada. *Query builder* memungkinkan pengguna untuk melakukan pencarian, pengurutan, atau pemfilteran data sesuai kebutuhan.

Tampilan tabel ini memberikan cara yang lebih terstruktur dan dapat dianalisis untuk menampilkan data listrik yang diterima oleh *Firebase*. Dengan melihat tabel ini, pengguna dapat dengan mudah memahami dan menganalisis informasi listrik yang diukur sesuai dengan dokumen yang diberikan.



Gambar 5. 5 Hasil *Firebase monitor rules*

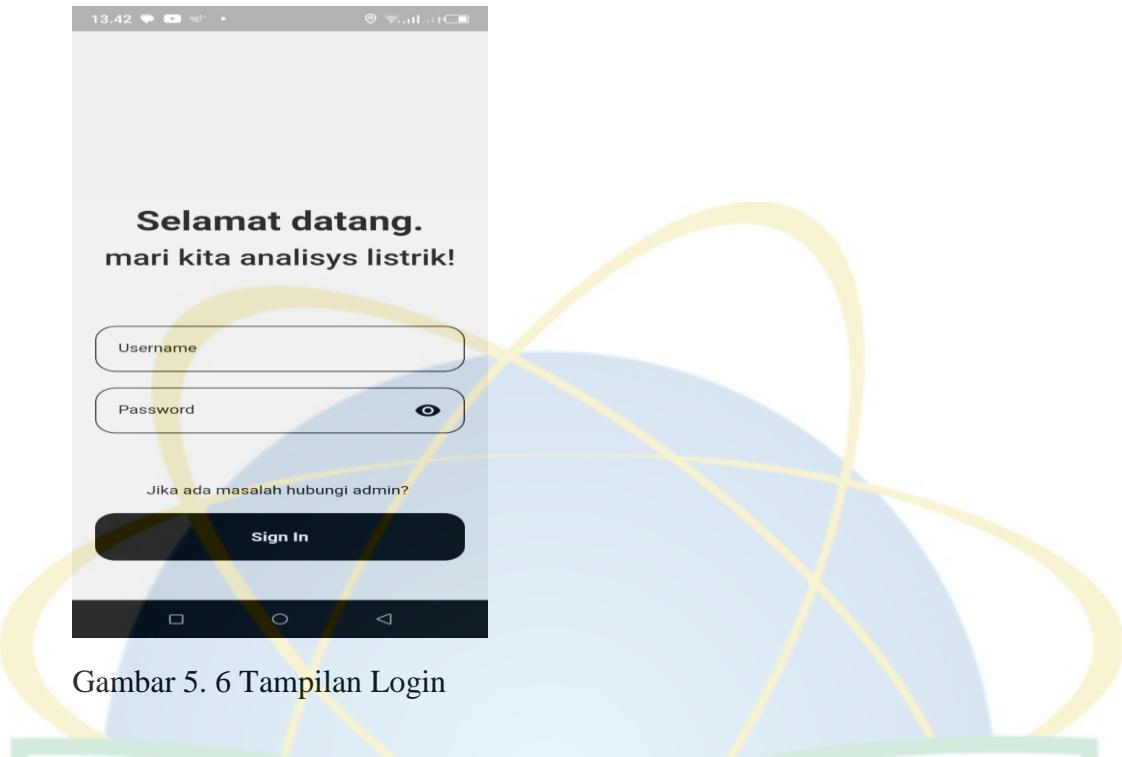
Pada gambar 5.5 mungkin menggambarkan hasil *Firebase Monitor Rules* yang menampilkan grafik untuk melihat jumlah data yang masuk ke *Firebase* dari setiap tanggal. Grafik ini membantu pengguna untuk memantau dan menganalisis pola data yang diterima oleh *Firebase* dari ESP8266 dengan sensor PZEM004T. Grafik akan memiliki sumbu X yang mewakili tanggal dan sumbu Y yang mewakili jumlah data. Sumbu X akan menunjukkan tanggal-tanggal tertentu, sedangkan sumbu Y akan menunjukkan jumlah data yang masuk ke *Firebase* pada tanggal tersebut.

Grafik tersebut mungkin berupa grafik batang atau grafik garis, tergantung pada preferensi visualisasi data yang digunakan. Grafik batang akan menunjukkan jumlah data secara terpisah untuk setiap tanggal, sedangkan grafik garis akan menunjukkan tren atau perubahan jumlah data dari waktu ke waktu.

Grafik dapat menyertakan informasi tooltip yang muncul saat pengguna mengarahkan kursor ke titik data pada grafik. Tooltip ini dapat memberikan informasi lebih rinci tentang jumlah data yang masuk pada tanggal tertentu, memudahkan pengguna untuk memperoleh detail lebih lanjut saat menganalisis data.

## 5.5 Hasil Android

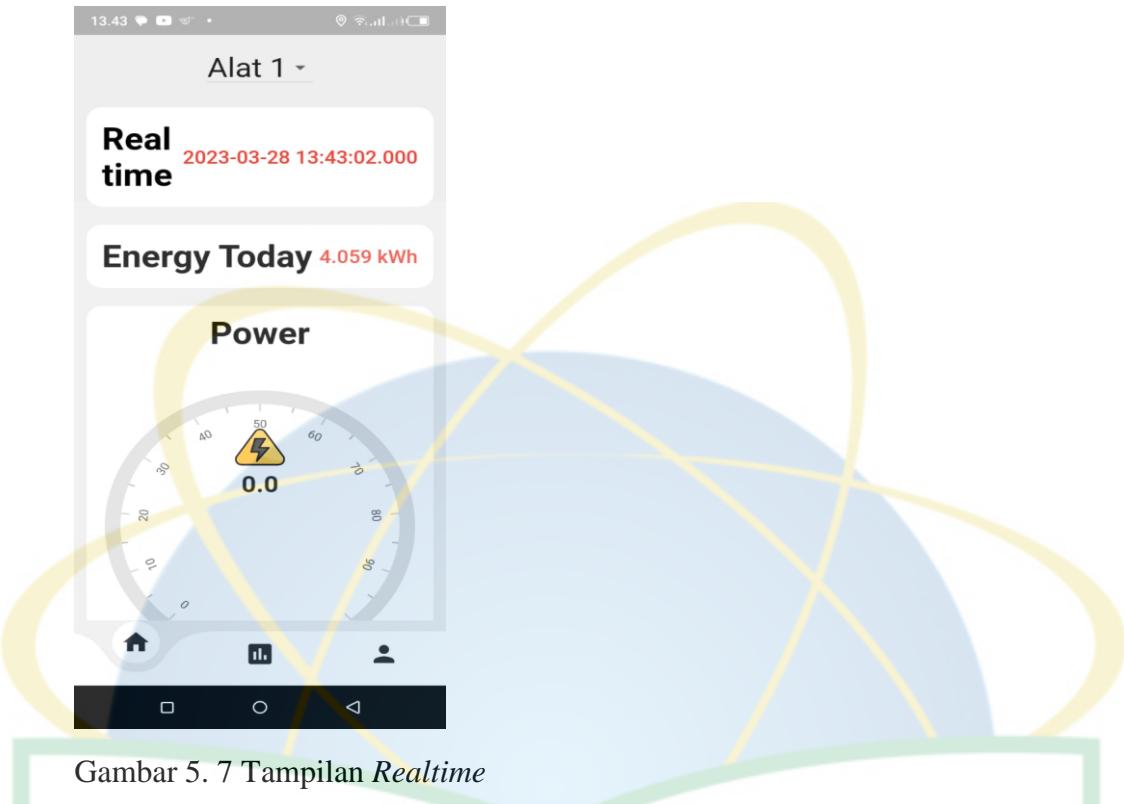
Berikut adalah hasil dari *interface* dan *experience* yang peneliti buat untuk melihat hasil dari sistem monitoring listrik :



Gambar 5. 6 Tampilan Login

Gambar 5.6 menggambarkan tampilan login yang memungkinkan pengguna memasukkan username dan password yang sudah disetup untuk masuk ke layar utama aplikasi monitoring listrik. Tampilan login akan mencakup dua kolom, yaitu kolom "Username" dan kolom "Password". Pengguna dapat memasukkan username yang telah ditentukan untuk akun pengguna di kolom "Username" dan memasukkan password yang sesuai di kolom "Password".

Di bawah formulir login, biasanya terdapat tombol "Sing In" yang memungkinkan pengguna memproses informasi login yang telah dimasukkan. Setelah pengguna mengklik tombol "Login", aplikasi akan memeriksa apakah username dan password yang dimasukkan sesuai dengan kode program android.

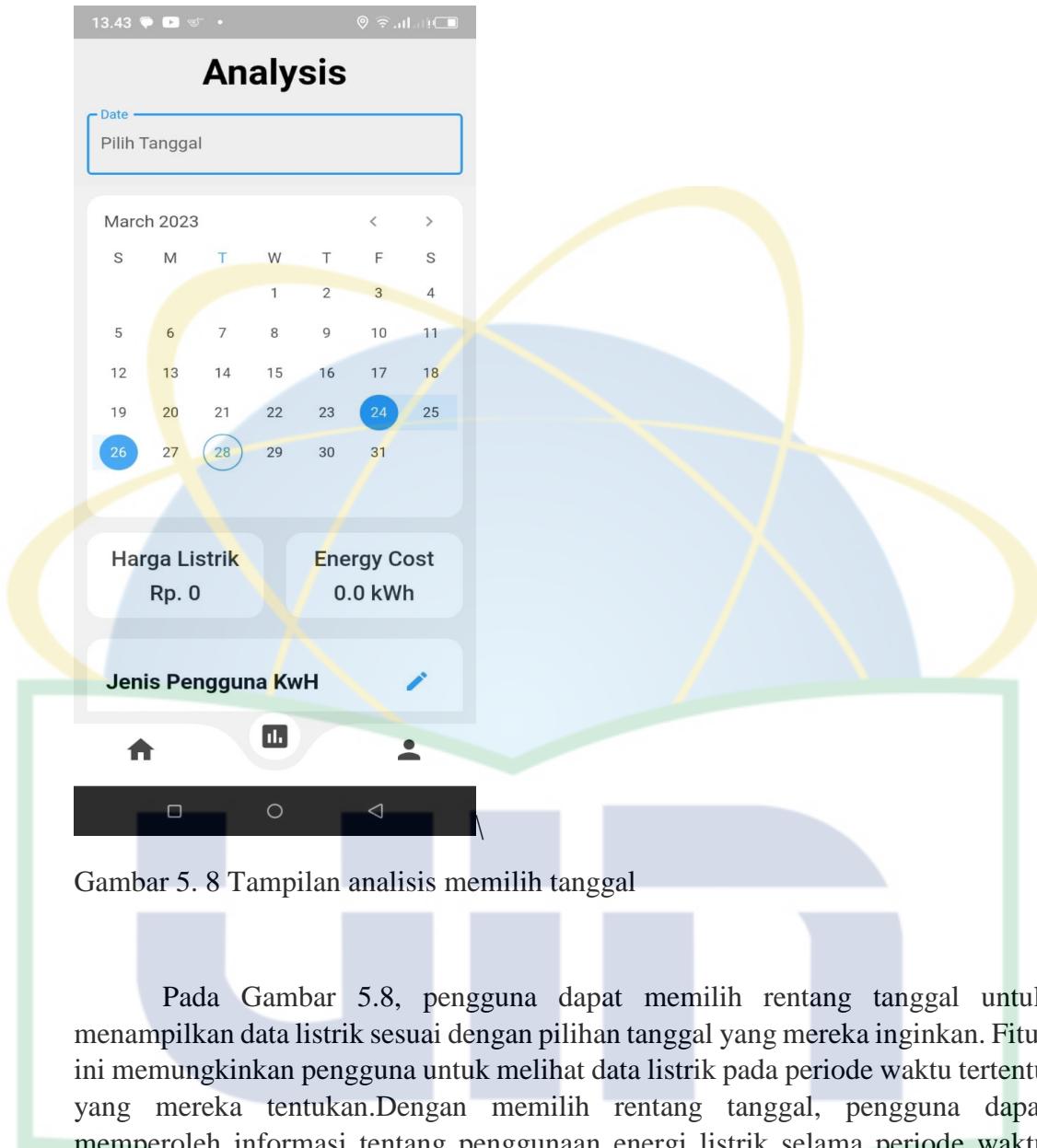


Gambar 5. 7 Tampilan *Realtime*

Pada Gambar 5.7, pengguna dapat memantau data listrik secara real-time untuk hari ini. Hal ini berarti pengguna dapat melihat informasi terkini tentang penggunaan energi listrik yang sedang berlangsung pada saat itu. Informasi yang ditampilkan dapat berupa nilai energi saat ini, seperti arus, tegangan, atau daya yang dikonsumsi pada waktu tersebut.

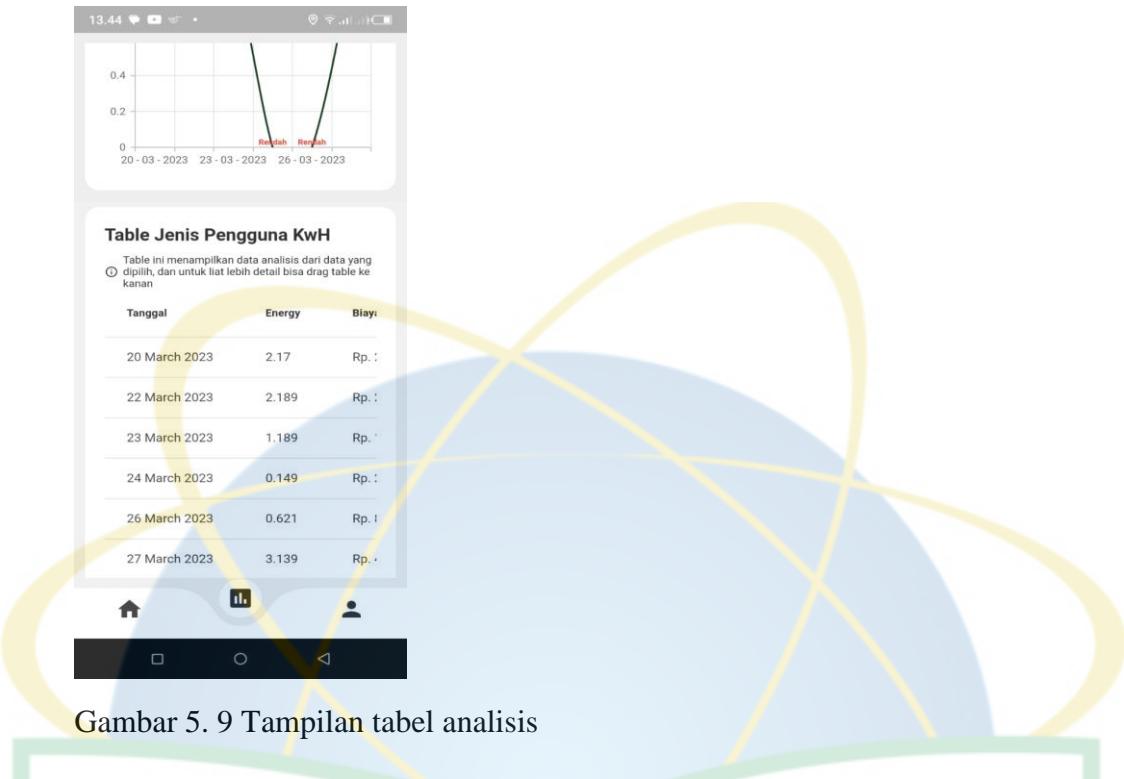
Dengan memantau data listrik secara real-time, pengguna dapat memiliki pemahaman yang lebih baik tentang pola penggunaan energi pengguna. pengguna dapat melihat seberapa banyak energi yang digunakan pada saat tertentu dan dapat mengambil tindakan yang sesuai, seperti mengoptimalkan penggunaan perangkat listrik atau mengurangi konsumsi energi jika diperlukan.

Dengan informasi real-time ini, pengguna dapat dengan mudah melihat apakah ada perubahan yang signifikan dalam penggunaan energi listrik mereka dan dapat mengambil langkah-langkah yang diperlukan untuk mengelola konsumsi energi secara efisien.



Pada Gambar 5.8, pengguna dapat memilih rentang tanggal untuk menampilkan data listrik sesuai dengan pilihan tanggal yang mereka inginkan. Fitur ini memungkinkan pengguna untuk melihat data listrik pada periode waktu tertentu yang mereka tentukan. Dengan memilih rentang tanggal, pengguna dapat memperoleh informasi tentang penggunaan energi listrik selama periode waktu tersebut. Pengguna dapat melihat pola penggunaan energi selama beberapa hari, minggu, atau bulan tertentu.

Fitur ini sangat berguna untuk menganalisis pola penggunaan energi dalam jangka waktu tertentu, seperti mencari tahu penggunaan energi rata-rata setiap hari dalam sebulan, membandingkan konsumsi energi pada bulan yang berbeda, atau melihat perubahan pola penggunaan energi seiring waktu. Dengan adanya pilihan rentang tanggal, pengguna dapat memperoleh wawasan yang lebih komprehensif tentang penggunaan energi listrik mereka dan dapat mengidentifikasi tren atau pola tertentu yang mungkin mempengaruhi konsumsi energi mereka.



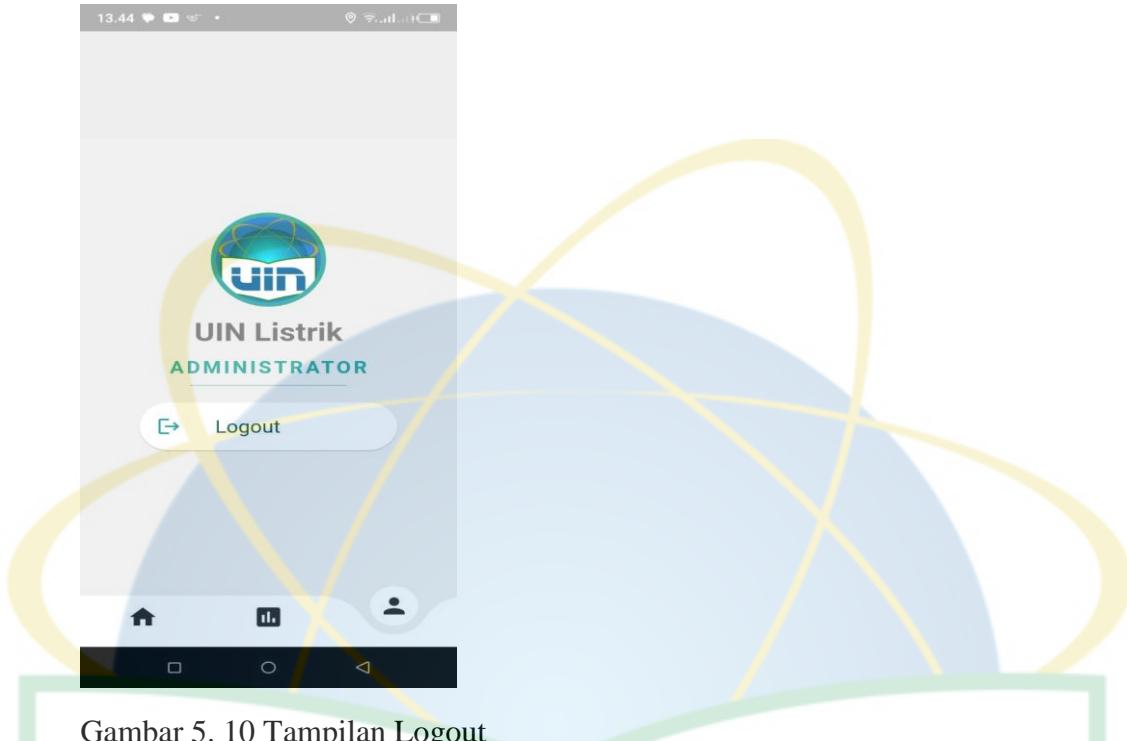
Gambar 5. 9 Tampilan tabel analisis

Gambar 5.10 menggambarkan tabel data yang menampilkan informasi terperinci tentang penggunaan listrik pada tanggal yang dipilih oleh pengguna. Tabel ini dirancang untuk memberikan pengguna listrik dengan gambaran yang lebih mendetail tentang penggunaan energi mereka pada tanggal tertentu.

Tabel tersebut biasanya memiliki beberapa kolom yang mencakup informasi seperti waktu penggunaan, jumlah energi yang dikonsumsi (dalam satuan tertentu, seperti kilowatt-hour), dan juga biaya energi terkait.

Pengguna listrik dapat memilih tanggal tertentu untuk melihat tabel data tersebut dan mendapatkan wawasan yang lebih terperinci tentang penggunaan energi pengguna pada hari itu. Informasi ini dapat membantu pengguna memahami pola penggunaan energi mereka, mengidentifikasi sumber penggunaan energi yang signifikan, dan membuat keputusan yang lebih bijaksana dalam hal penggunaan energi di masa depan.

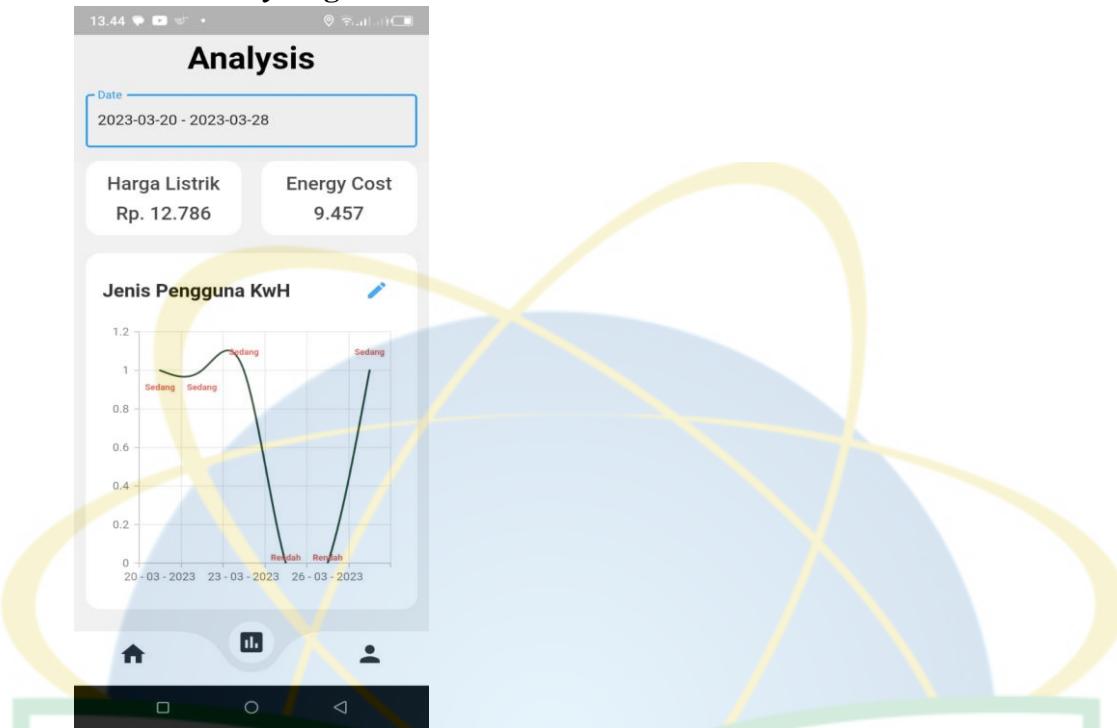
Dengan melihat tabel data ini, pengguna listrik dapat menganalisis tren dan pola penggunaan energi harian mereka, mengidentifikasi waktu atau kegiatan yang mempengaruhi penggunaan energi, dan mengambil tindakan yang tepat untuk mengoptimalkan penggunaan energi mereka. Tabel data seperti ini dapat menjadi alat yang berguna dalam manajemen energi, membantu pengguna listrik dalam memantau, mengontrol, dan mengelola penggunaan energi mereka dengan lebih efisien.



Gambar 5. 10 Tampilan Logout

Gambar 5.11 mungkin menggambarkan tampilan logout dalam aplikasi sistem monitoring listrik. Hanya meklik button logout untuk proses keluar dari aplikasi.Tampilan logout akan mencakup tombol "Logout" yang terletak di tempat yang terlihat dengan jelas, seperti di bagian atas atau bawah layar, atau dalam menu pengaturan atau profil pengguna. Pengguna cukup mengklik tombol tersebut untuk memulai proses logout.

## 5.6 Hasil Fuzzy Sugeno



Gambar 5. 11 Tampilan Fuzzy atau analisis grafik

Gambar 5.9 menggambarkan grafik timeseries yang menunjukkan golongan listrik dalam penggunaan harian berdasarkan tanggal yang dipilih. Grafik ini dibuat menggunakan output dari algoritma *Fuzzy Sugeno*. Output tersebut kemudian dikelompokkan menjadi kategori rendah, sedang, atau tinggi.

Grafik tersebut menampilkan sumbu X yang mewakili tanggal dan sumbu Y yang mewakili golongan listrik. Garis atau titik pada grafik menunjukkan kategori golongan listrik pada tanggal tertentu. Misalnya, garis atau titik di bagian bawah grafik menunjukkan golongan listrik rendah, garis atau titik di tengah menunjukkan golongan listrik sedang, dan garis atau titik di bagian atas menunjukkan golongan listrik tinggi.

Dengan melihat grafik ini, pengguna listrik dapat memahami dan memonitor pola penggunaan listrik mereka seiring dengan perubahan tanggal. Pengguna dapat melihat apakah penggunaan listrik pengguna cenderung rendah, sedang, atau tinggi pada tanggal tertentu, dan dengan demikian dapat mengambil tindakan yang sesuai, seperti mengurangi penggunaan listrik saat kategori golongan tinggi atau mengoptimalkan penggunaan listrik saat kategori golongan rendah. Grafik timeseries ini membantu pengguna listrik untuk mendapatkan wawasan visual tentang pola penggunaan listrik mereka dan dapat menjadi alat yang berguna dalam mengelola dan mengoptimalkan penggunaan energi.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa skripsi ini berhasil mengimplementasikan sistem monitoring daya listrik *internet of things* menggunakan algoritma *Fuzzy logic* sugeno dan *firebase* berbasis android. kesimpulan dari penelitian ini adalah sebagai berikut:

1. Sistem monitoring daya listrik *IoT* yang menggunakan *Firebase* dan algoritma *Fuzzy Logic* Sugeno berbasis android telah berhasil diimplementasikan dengan baik. Sistem ini dapat mengumpulkan data penggunaan daya listrik secara realtime dan mengolahnya dengan algoritma *Fuzzy Logic* untuk memberikan informasi yang berguna kepada pengguna.
2. Penggunaan *Firebase* sebagai database realtime memungkinkan pengiriman data yang cepat dan akurat antara perangkat ESP8266 dan aplikasi Android. Hal ini memungkinkan pengguna untuk melihat informasi penggunaan daya listrik secara realtime melalui aplikasi.
3. Algoritma *Fuzzy Logic* Sugeno digunakan untuk menganalisis data penggunaan daya listrik dan memberikan informasi yang lebih intuitif dan mudah dipahami oleh pengguna dengan menampilkan hasil berupa grafik. Dengan adanya algoritma ini, pengguna dapat melihat pola penggunaan daya listrik yang efisien atau tidak efisien dan mengambil langkah-langkah yang sesuai.
4. Aplikasi Android sebagai antarmuka pengguna memberikan kemudahan dalam mengakses informasi penggunaan daya listrik. Pengguna dapat melihat data secara realtime, melihat data historis, dan melihat analisis data sesuai *reqdate* pengguna listrik.
5. Penelitian ini menunjukkan bahwa sistem monitoring daya listrik *IoT* memiliki potensi besar dalam membantu pengguna untuk mengelola dan mengoptimalkan penggunaan daya listrik. Dengan informasi yang akurat dan realtime, pengguna dapat membuat keputusan yang lebih bijaksana dalam penggunaan energi dan mengurangi pemborosan daya listrik.

Secara keseluruhan, penelitian ini membuktikan bahwa sistem monitoring daya listrik berbasis *IoT* dengan menggunakan *Firebase* dan algoritma *Fuzzy Logic* Sugeno dapat memberikan manfaat yang signifikan dalam mengoptimalkan penggunaan energi. Implementasi dan pengembangan lebih lanjut dari sistem ini dapat menjadi langkah awal dalam mewujudkan rumah atau bangunan pintar yang efisien energi di masa depan.

## 6.2 Saran

Berikut adalah beberapa saran untuk pengembangan skripsi SISTEM MONITORING DAYA LISTRIK INTERNET OF THINGS (IoT) MENGGUNAKAN ALGORITMA FUZZY LOGIC SUGENO DAN FIREBASE BERBASIS ANDROID

1. Pengembangan lebih lanjut pada aplikasi mobile: Pengembangan aplikasi mobile dapat dilakukan dengan menambahkan fitur baru seperti notifikasi ketika daya listrik terlalu tinggi atau terlalu rendah, Mematikan/menghidupkan aliran listrik dari jarak jauh, dan fitur pemberitahuan ketika terjadi pemadaman listrik.
2. Pengembangan algoritma: Algoritma *Fuzzy Logic* Sugeno yang digunakan dalam skripsi ini dapat ditingkatkan lagi dengan memperhatikan beberapa parameter yang mempengaruhi pengukuran daya listrik seperti kualitas arus listrik, tegangan, dan faktor daya.
3. Pengujian lebih lanjut: Selain pengujian yang telah dilakukan pada skripsi ini, pengujian pada skala yang lebih besar dapat dilakukan untuk memastikan keandalan sistem pada lingkungan yang lebih kompleks dan memperbaiki masalah yang mungkin terjadi.
4. Pengembangan pada aspek keamanan: Sistem monitoring daya listrik dapat menjadi target untuk serangan siber, oleh karena itu pengembangan pada aspek keamanan harus ditingkatkan seperti enkripsi data dan autentikasi pengguna.

Dengan mengikuti saran-saran tersebut, diharapkan pengembangan skripsi ini dapat lebih maksimal dan memberikan manfaat yang lebih besar dalam mengoptimalkan penggunaan daya listrik dan membantu mengurangi biaya energi listrik.

## Daftar Pustaka

- Dwisyaputra, I., Yudhi, Y., Anggrainy, K., & Novaldy, S. (2021). Kontrol dan Monitoring Stop Kontak Berbasis Android. *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, 4(1), 23. <https://doi.org/10.24853/resistor.4.1.23-28>
- Fitriyah, Q., Vira Putri, T., Wirangga, A., & Prihadi Eko Wahyudi, M. (2021). Rancang Bangun Alat Monitoring Energi Listrik Pada Kulkas Berbasis Smartphone. *Jurnal Integrasi /*, 122(2), 122–126.
- ICES. (2021). *No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title* (Issue March).
- Ilham Firman Maulana. (2020). Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(5), 854–863. <https://doi.org/10.29207/resti.v4i5.2232>
- Ma, matul, & Hamdani, E. (2019). *Prototipe Sistem Monitoring Pemakaian Daya Listrik Rumah Tangga Menggunakan Android*. 4(2), 2541–5867. <https://doi.org/10.37253/telcomatics.v4i2.613>
- Mitha Djaksana, Y., Eresha Jl Raya Puspittek, S., Serpong, K., & Tangerang Selatan, K. (2020). PERANCANGAN SISTEM MONITORING DAN KONTROLING PENGGUNAAN DAYA LISTRIK BERBASIS ANDROID. *Jurnal Sistem Informasi Dan Teknologi Informasi*, 2(3), 13–24.
- Murdiyat, P., Handri Gunanto, L., Teknik Elektro, J., & Negeri Samarinda, P. (2021). *Rancang Bangun Sensor Node Untuk Sistem Monitoring Energi Listrik Nirkabel Pada Gedung Dalam Kampus Politeknik Negeri Samarinda*. 2(1). <https://doi.org/xx.xxxx>
- Name, P., Count, W., Count, C., Count, P., Size, F., Date, S., & Date, R. (2023). *LOGIKA FUZZY TAHANI by INDAH WAHY UNI 48.pdf* Indah Wahyuni.
- Nasional, S., Elektro, T., Informasi, S., Informatika, T., Sari, S. Z., Putra, D. S., & Iqbal, A. (2021). *Smart Metering untuk Pengidentifikasi Jenis Beban Listrik*. 175–182.
- Online, T. (2023). *Jurnal Politeknik Caltex Riau Alat Monitoring dan Notifikasi Penggunaan Daya Listrik Rumah Tangga Berbasis Internet Of Things*. 8(2), 408–420.
- Pela, M. F., & Pramudita, R. (2021). SISTEM MONITORING PENGGUNAAN DAYA LISTRIK BERBASIS INTERNET OF THINGS PADA RUMAH DENGAN MENGGUNAKAN APLIKASI BLYNK. *Infotech: Journal of Technology Information*, 7(1), 47–54. <https://doi.org/10.37365/jti.v7i1.106>
- PLN. (2021). *Cerita Sampul*.

- Rdiansyah, A. (2020). Monitoring Daya Listrik Berbasis IoT (Internet of Things). *Universitas Islam Indonesia.* <https://dspace.uii.ac.id/handle/123456789/23561>
- Rizky, R., Hidayat, T., Hardianto, A., & Hakim, Z. (2020). Penerapan Metode Fuzzy Sugeno Untuk pengukuran Keakuratan Jarak Pada Pintu Otomatis di CV. Bejo Perkasa. *Jurnal Teknik Informatika Unika St. Thomas (JTIUST)*, 05(01), 33–42.
- Samallo, R. A., Andryana, S., Titi, R., & Sari, K. (2020). Android-Based Monitoring and Control of Electricity Consumption. *Jurnal Mantik*, 4(1). <https://iocscience.org/ejournal/index.php/mantik>
- Suprapto, H., & Simanjuntak, P. (2020). *Fuzzy Logic* Untuk Memprediksi Pemakaian Listrik Menggunakan Metode Mamdani. *Computer and Science* ..., 2(1), 29–37.
- Tanto, T. Al, & Riswanto, R. (2022). Kajian Suhu Permukaan Laut (SPL) Menggunakan Analisis Deret Waktu di Perairan Laut Banda. *Jurnal Kelautan: Indonesian Journal of Marine Science and Technology*, 15(3), 270–279. <https://doi.org/10.21107/jk.v15i3.14386>
- Tukadi, Widodo, W., Ruswiensari, M., & Qomar, A. (2019). Monitoring Pemakaian Daya Listrik Secara Realtime Berbasis Internet Of Things. In *Seminar Nasional Sains dan Teknologi Terapan VII 2019*. <https://ejurnal.itats.ac.id/sntekpan/article/download/659/468>
- Vieri, K. Del. (2021). *Implementasi Fuzzy Inference System Pada Sistem Penyiram Tanaman Otomatis Mesin Paralel*.
- Widiasari, C. (2020). Sistem Monitoring Daya Listrik dan Pengontrolan Perangkat Elektronik Berbasis IoT. *Seminar Nasional Teknologi Informasi Komunikasi Dan Industri*, 342–349.
- Zaen, S. L., & Rozaq, A. (2021). SISTEM MONITORING PEMAKAIAN ENERGI LISTRIK RUMAH TANGGA BERBASIS WEB. In *Jurnal ELKON* (Vol. 01).

## Lampiran

### WAWANCARA PENGGUNA LISTRIK

a. Identitas diri :

- 1) Nama : Muhammad Solihin
- 2) Pekerjaan : Pegawai Negeri Sipil (PNS)
- 3) Alamat : Jalan raya tengah, Kelurahan Tengah , kecamatan kramat jati, Jakarta timur
- 4) Status : Menikah

b. Pertanyaan peneliti

1. Peneliti : Apa alasan utama Anda menggunakan listrik sebagai sumber energi di rumah Anda?

informan: Saya menggunakan listrik karena kepraktisannya. Listrik dapat dengan mudah diakses, bersih, dan dapat digunakan untuk berbagai kebutuhan seperti penerangan, peralatan elektronik, dan sistem pendingin udara.

2. Peneliti: Bagaimana Anda memantau dan mengelola penggunaan listrik di rumah Anda saat ini?

Informan: Saya saat ini menggunakan metode manual untuk memantau penggunaan listrik, seperti mencatat meteran listrik secara berkala. Namun, saya juga memiliki beberapa perangkat pintar yang membantu mengontrol penggunaan listrik, seperti pengatur waktu dan alat penghemat energi.

3. peneliti: Apakah Anda memiliki tantangan atau masalah terkait dengan penggunaan listrik yang ingin Anda selesaikan?

informan: Saya menghadapi masalah dengan biaya listrik yang tinggi. Saya ingin dapat mengidentifikasi peralatan atau kebiasaan yang menyebabkan penggunaan listrik yang tinggi agar dapat mengambil langkah-langkah untuk menghemat energi dan mengurangi biaya.

4. peneliti: Apa pendapat Anda tentang adopsi teknologi terkait penggunaan listrik yang lebih pintar, seperti smart meter atau sistem monitoring listrik ?

informan: Saya melihat potensi besar dalam teknologi tersebut. Dengan menggunakan smart meter atau sistem monitoring energi, kita dapat melacak dan memahami dengan lebih baik bagaimana kita menggunakan listrik, serta mengidentifikasi pola penggunaan yang tidak efisien. Hal ini

dapat membantu kita mengambil tindakan yang tepat untuk mengoptimalkan penggunaan energi dan mengurangi biaya.

5. peneliti: Bagaimana Anda mengelola tagihan listrik bulanan Anda dan apakah Anda tertarik untuk menggunakan solusi yang lebih cerdas dalam mengontrol dan mengelola penggunaan energi?

informan: Saat ini, saya menggunakan metode tradisional untuk mengelola tagihan listrik, yaitu membayar sesuai dengan jumlah yang tertera di tagihan bulanan. Namun, saya tertarik untuk menggunakan solusi yang lebih cerdas, seperti aplikasi mobile atau sistem monitoring energi, yang dapat membantu saya mengontrol penggunaan listrik dan memahami faktor-faktor yang mempengaruhi tagihan listrik.

6. Peneliti : Apa manfaat utama dari sistem monitoring pengguna listrik secara online?

Informan : Sistem monitoring pengguna listrik secara online memberikan kemudahan dan aksesibilitas bagi pengguna untuk melihat dan memantau penggunaan daya listrik mereka secara real-time. Hal ini memungkinkan pengguna untuk mengidentifikasi dan mengelola konsumsi daya mereka dengan lebih efisien, serta mengurangi biaya listrik yang tidak perlu.

7. Peneliti :Apa saja data yang biasanya dipantau dalam sistem monitoring pengguna listrik secara online?

Informan: Data yang biasanya dipantau meliputi informasi seperti daya listrik yang digunakan, tegangan listrik, arus listrik, faktor daya, dan informasi lainnya terkait penggunaan energi. Data ini dapat membantu pengguna dalam memahami pola penggunaan listrik mereka dan mengambil tindakan yang sesuai untuk mengoptimalkan penggunaan energi.

8. Peneliti: Bagaimana sistem monitoring pengguna listrik secara online dapat membantu dalam mendeteksi kemungkinan kebocoran daya atau masalah listrik lainnya?

Informan: Dengan sistem monitoring pengguna listrik secara online, pengguna dapat memantau pola penggunaan daya mereka secara terus-menerus. Jika ada peningkatan yang tidak wajar dalam konsumsi daya, sistem dapat memberikan peringatan kepada pengguna tentang kemungkinan adanya kebocoran daya atau masalah listrik lainnya. Hal ini memungkinkan pengguna untuk mengambil tindakan segera dan mencegah kerugian yang lebih besar.

9. Peneliti: Bagaimana keamanan data pengguna dijaga dalam sistem monitoring pengguna listrik secara online?

informan: Keamanan data pengguna sangat penting dalam sistem monitoring pengguna listrik secara online. Untuk menjaga keamanan data, sistem biasanya menggunakan enkripsi data dan protokol keamanan yang kuat. Selain itu, akses ke data pengguna juga diatur dengan menggunakan otentikasi dan otorisasi yang tepat. Dengan langkah-langkah keamanan ini, data pengguna dapat terlindungi dengan baik.

10. Peneliti : Apa keuntungan utama dari menggunakan sistem monitoring pengguna listrik secara online dibandingkan dengan metode tradisional?

Informan : Salah satu keuntungan utama adalah aksesibilitas yang lebih baik. Dengan sistem monitoring pengguna listrik secara online, pengguna dapat dengan mudah memantau penggunaan daya listrik mereka melalui perangkat mobile atau web kapan pun dan di mana pun. Selain itu, sistem ini juga memberikan kemampuan untuk menganalisis data penggunaan daya secara real-time, memberikan informasi yang lebih akurat dan bermanfaat bagi pengguna.

