

## **TUGAS AKHIR**

# **PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGUNAKAN THINGSPEAK**



**Disusun Oleh :**

**MUHAMAD NANDA AL HUDA**

**NIM. 1814121009**

**BIDANG STUDI TEKNIK SISTEM TENAGA  
PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS BHAYANGKARA SURABAYA  
2023**



**LEMBAR PERSETUJUAN**

**PENERAPAN WIRELESS SENSOR NETWORK  
(WSN) UNTUK MONITORING DAYA LAMPU  
PENERANGAN JALAN UMUM (LPJU)  
MENGUNAKAN THINGSPEAK**

**Diajukan Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana  
(S1) Teknik Elektro Universitas Bhayangkara Surabaya**


Disusun Oleh :

**MUHAMAD NANDA AL HUDA**

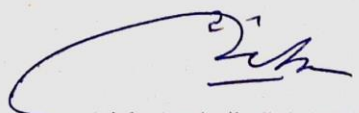
**NIM. 1814121009**

Diperiksa dan disetujui oleh :

Pembimbing I

  
Dr. Bambang Purwahyudi, ST., MT.  
NIDN : 0025057001

Mengetahui,  
Ketua Program Studi Teknik Elektro

  
Richa Watiasih, S.T., M.T.  
NIDN : 0714047601

HALAMAN INI SENGAJA DIKOSONGKAN

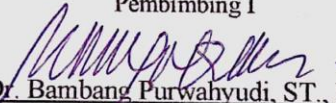
## LEMBAR PENGESAHAN

### PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGUNAKAN THINGSPEAK

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh  
Gelara Sarjana (S1) Teknik Elektro Universitas Bhayangkara  
Surabaya

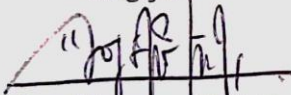
Disusun Oleh :  
**MUHAMAD NANDA AL HUDA**  
NIM : 1814121009

Tanggal Ujian : 06 Juli 2023  
Pembimbing I

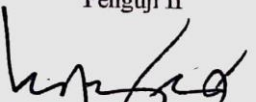
  
Dr. Bambang Purwahyudi, ST., MT.  
NIDN : 0025057001

Disetujui oleh Tim Penguji Tugas Akhir

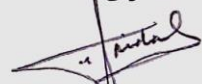
Penguji I

  
Agus Kiswanto, ST., MT  
NIDN: 0715087101


Penguji II

  
Dr. Hasti Afranti, ST., MT  
NIDN: 0730037402

Penguji III

  
Prof. Dr. F. Saidah, MT  
NIDN: 0712066101

  
Mengetahui,  
Dekan Fakultas Teknik  
Dr. Mohammad Ghazi, ST., MT  
NIDN: 0028127003

  
Menyetujui,  
Ketua Program Studi Teknik Elektro  
Richa Watiasih, ST., MT  
NIDN : 0714047601

HALAMAN INI SENGAJA DIKOSONGKAN

## LEMBAR PERNYATAAN KEASLIAN PENELITIAN

Yang bertandatangan di bawah ini :

**Nama** : Muhamad Nanda Al Huda  
**Tempat/ Tanggal lahir** : Kediri/ 07 September 1999  
**NIM** : 1814121009  
**Fakultas / Jurusan** : Teknik / Teknik Elektro

Dengan ini saya menyatakan bahwa Tugas Akhir kami dengan judul **“PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGGUNAKAN THINGSPEAK”** beserta seluruh isinya adalah karya saya sendiri dan bukan merupakan karya tulis orang lain, baik sebagian maupun seluruhnya, kecuali dalam bentuk kutipan yang telah disebutkan sumbernya.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya. Apabila kemudian di temukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini maka saya siap menanggung segala bentuk resiko/sanksi yang berlaku.

Surabaya, 06 Juli 2023

Yang membuat pernyataan,



**(Muhamad Nanda Al Huda)**

**NIM: 1814121009**

HALAMAN INI SENGAJA DIKOSONGKAN



## ABSTRAK

Di era teknologi informasi yang kian maju saat ini, banyak sekali produsen lampu penerangan jalan yang berlomba lomba untuk berinovasi pada bidang IOT untuk optimasi penggunaan lampu PJU. Dimana saat ini banyak perusahaan yang menerapkan pemantauan lampu secara online menggunakan website. Akan tetapi masih banyak isu mengenai kurangnya efisiensi dalam segi biaya. Hal tersebut terjadi karena banyak PJU bersistem IOT tersebut bersifat stand alone atau berdiri sendiri, dimana 1 buah lampu dibekali 1 buah modem sellular yang artinya setiap lampu PJU akan dikenakan biaya berlangganan. Dapat dibayangkan apabila dalam suatu jalan dipasang 100 buah PJU, maka user atau pengguna akan mengeluarkan biaya untuk 100 buah PJU. Lain halnya apabila system pengiriman data antar PJU tersebut kita model seperti topologi jaringan, dimana antar PJU nantinya akan berkomunikasi menggunakan perangkat RF, dan hanya lampu PJU induk yang akan dibekali modem untuk pengiriman data secara online. Maka biaya berlangganan yang akan ditanggung user atau pengguna hanya untuk lampu PJU induk saja. Dengan adanya studi kasus tersebut, maka penulis memanfaatkan system WSN untuk proses komunikasi dan pengiriman data antar PJU dan internet. ESP32 sebagai mikrokontroler dan pengganti modul jaringan sellular, EBYTE LORA E220 sebagai perangkat komunikasi antar PJU, dan PZEM004T sebagai sensor besaran listrik, dan Thingspeak sebagai website penampil data PJU.

Kata Kunci : PJU, WSN, Mikrokontroler, *Internet of Think (IoT)*.

HALAMAN INI SENGAJA DIKOSONGKAN

## **ABSTRACT**

*In the era of increasingly advanced information technology, many street lighting manufacturers are competing to innovate in the IOT field to optimize the use of PJU lamps. Where currently many companies are implementing online lamp monitoring using the website. However, there are still many issues regarding the lack of efficiency in terms of costs. This happens because many PJUs with the IOT system are stand alone, where 1 lamp is equipped with 1 cellular modem, which means that each PJU lamp will be charged a subscription fee. You can imagine if 100 PJUs are installed on a road, then the user or users will pay for 100 PJUs. It's another case if we model the data transmission system between PJUs like a network topology, where between PJUs will communicate using RF devices, and only the main PJU lights will be equipped with a modem for online data transmission. Then the subscription fee that will be borne by the user or users is only for the main PJU lights. With this case study, the authors utilize the WSN system for the process of communication and sending data between PJU and the internet. ESP32 as a microcontroller and replacement for cellular network modules, EBYTE LORA E220 as a communication device between PJUs, and PZEM004T as a sensor for electrical quantities, and Thingspeak as a website for displaying PJU data.*

*Keywords: PJU, WSN, Microcontroller, Internet of Think (IoT).*

HALAMAN INI SENGAJA DIKOSONGKAN

## KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT, atas limpaha nrahmat dan hidayah-Nya sehingga peneliti dapat menyelesaikan Tugas Akhir yang berjudul : **“PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGGUNAKAN THINGSPEAK”**

1. **Allah SWT** yang masih memperkenalkan kepada penulis terlebih lagi atas kesempatan untuk menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Bambang Purwahyudi,ST., MT. Selaku pembimbing tugas akhir
3. Bapak Dr. Mohammad Ghazi ST., MT selaku Dekan Fakultas Teknik Universitas Bhayangkara Surabaya
4. Ibu Richa Watiasih ST., MT selaku Ketua Program Studi Teknik Elektro Universitas Bhayangkara Surabaya
5. Bapak Agus Kiswantonono ST., MT selaku Koordinator Tugas Akhir
6. Bapak Heru selaku staff TU TE yang telah membantu penulis melengkapi administrasi tugas akhir
7. Kedua orang tua dan keluarga yang selalu memberikan doa, semangat, dan dukungan lainnya hingga tugas akhir ini dapat diselesaikan
8. Serta banyak pihak lain yang telah membantu penulis dalam menyelesaikan tugas akhir ini.

Peneliti menyadari sepenuhnya bahwa dalam penulisan Tugas Akhir ini masih banyak kekurangan dan keterbatasan, oleh karena itu peneliti mengharapkan saran yang membangun agar tulisan ini bermanfaat bagi perkembangan ilmu pengetahuan kedepannya.

Surabaya, 06 Juli 2023  
Penulis,

**Muhamad Nanda Al Huda**

HALAMAN INI SENGAJA DIKOSONGKAN

## DAFTAR ISI

SAMPUL DEPAN .....	i
LEMBAR PERSETUJUAN.....	iii
LEMBAR PENGESAHAN.....	v
LEMBAR PERNYATAAN KEASLIAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xix
DAFTAR TABEL.....	xxi
BAB I.....	1
PENDAHULUAN .....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah .....	1
1.3    Batasan Masalah .....	2
1.4    Tujuan Penelitian .....	2
1.5    Metodologi Penelitian.....	2
1.5.1    Studi Literatur.....	2
1.5.2    Perancangan Sistem .....	3
1.5.3    Pembuatan <i>Software</i> sistem .....	3
1.5.4    Pembuatan <i>Hardware</i> Sistem .....	3
1.5.5    Pengujian Sistem, Integrasi dan Pengambilan Data.....	4
1.5.6    Pembuatan Laporan Proyek Akhir .....	4
1.6    Sistematika Penulisan .....	4
BAB II.....	5
TINJAUAN PUSTAKA .....	5
2.1    Lampu PJU.....	5
2.1.1    Jenis lampu PJU.....	5
2.1.2    Perkembangan PJU .....	8
2.2    Sensor PZEM004-T.....	9
2.2.1    Spesifikasi PZEM004-T .....	10
2.3    RF Modul EBYTE E220.....	10
2.3.1    Fitur EBYTE E22 .....	11
2.4    NodeMCU ESP32 .....	12
2.4.1    Spesifikasi.....	13
2.4.2    Implementasi.....	13
2.5    Arduino IDE.....	14

2.5.1	Menulis Sketch .....	14
2.5.2	Sketchbook .....	19
2.5.3	Tabs, Multiple Files, Dan Campilations .....	19
2.5.4	Uploading .....	19
2.5.5	Library .....	19
2.5.6	Serial Monitor .....	19
2.5.7	Preferences.....	20
2.5.8	Language Support .....	20
2.6	Internet Of Things .....	20
2.6.1	Bagian-bagian IOT .....	21
2.6.2	Cara Kerja.....	22
2.6.3	Penerangan.....	23
2.6.4	Manfaat.....	24
2.7	Thingspeak .....	25
2.7.1	Fungsi.....	26
2.7.2	Fitur.....	26
2.7.3	Langkah Penggunaan.....	26
<b>BAB III.....</b>		<b>29</b>
<b>PERANCANGAN SISTEM .....</b>		<b>29</b>
3.1	Perencanaan Sistem .....	29
3.2	Perencanaan dan Perancangan Perangkat Keras Eletronik ..	31
3.2.1	NodeMCU.....	32
3.2.2	Sensor PZEM004T .....	33
3.2.3	RF modul .....	34
3.3	ThingSpeak.....	35
3.4	Sistem WSN .....	35
3.5	Desain Alat.....	36
<b>BAB IV .....</b>		<b>37</b>
<b>ANALISA DAN PENGUJIAN .....</b>		<b>37</b>
4.1	Pengujian PZEM004-T .....	37
4.1.1	Tujuan.....	37
4.1.2	Proses Pengambilan Data.....	37
4.1.3	Hasil dan Analisa .....	37
4.2	Peengujian Pengiriman data RF.....	42
4.2.1	Tujuan.....	42
4.2.2	Proses Pengambilan Data.....	42



4.2.3	Hasil dan Analisa .....	42
4.3	Pengujian Hasil Topologi linier.....	43
4.3.1	Tujuan .....	43
4.3.2	Proses Pengambilan Data.....	44
4.3.3	Hasil dan Analisa .....	44
4.4	Pengujian intensitas cahaya .....	55
4.4.1	Tujuan .....	55
4.4.2	Proses Pengambilan Data.....	55
4.4.3	Hasil dan Analisa.....	55
4.5	Menghitung umur lampu dalam satuan .....	57
4.5.1	Tujuan .....	57
4.5.2	Proses Pengambilan Data.....	57
4.5.3	Hasil dan Analisa .....	57
4.6	Foto Alat.....	57
BAB V .....		59
PENUTUP.....		59
5.1	Kesimpulan.....	59
5.2	Saran.....	59
DAFTAR PUSTAKA .....		61

HALAMAN INI SENGAJA DIKOSONGKAN

## DAFTAR GAMBAR

Gambar 2.1 Lampu PJU LVD.....	5
Gambar 2.2 Lampu TL.....	6
Gambar 2.3 Lampu PJU LED .....	7
Gambar 2.4 Salah satu aplikasi Smart Lighting perusahaan lokal .....	8
Gambar 2.5 Sensor PZEM004-T.....	9
Gambar 2.6 Rekomendasi rangkaian PZEM004-T .....	9
Gambar 2.7 Bentuk fisik modul RF EBYTE E22 .....	11
Gambar 2.8 Modul ESP32 devkit-C .....	12
Gambar 2.9 Tampilan Dari Software Arduino IDE .....	15
Gambar 2.10 Tampilan depan website thingspeak.....	25
Gambar 2.11 Tampilan sign up thingspeak.....	26
Gambar 2.12 Tampilan saat akan membuat chnnel baru .....	27
Gambar 2.13 Pengaturan chnnel baru .....	27
Gambar 2.14 Setting API Key .....	27
Gambar 2.15 Tampilan new Channel.....	28
Gambar 2.16 Tampilan menu aplikasi thingspeak.....	28
Gambar 3.1 Blok diagram sistem.....	29
Gambar 3.2 Flowchart sistem monitoring.....	30
Gambar 3.3 Rancangan skematik.....	31
Gambar 3.4 Hasil Hardware elektronik tampak depan .....	31
Gambar 3.5 Desain layout PCB rangkaian monitoring.....	32
Gambar 3.6 Konfigurasi pin I/O ESP32 ke interface.....	33
Gambar 3.7 Aplikasi Sensor tegangan pada kontroler .....	34
Gambar 3.8 Aplikasi modul sensor arus pada kontroler .....	34
Gambar 3.9 Tampilan data arus aplikasi thingspeak.....	35
Gambar 3.10 Pengiriman data dengan topologi linier.....	35
Gambar 3.11 desain penerangan jalan umum.....	36
Gambar 4.1 percobaan dengan alat watt meter.....	38
Gambar 4.2 Grafik perbandingan PZEM004-T dengan watt meter .....	39
Gambar 4.3 Grafik besar nilai error pembacaan watt.....	39
Gambar 4.4 Percobaan dengan alat ampere meter.....	40
Gambar 4.5 Grafik terbanding PZEM004-T dengan Amperemeter .....	41
Gambar 4.6 Grafik besar nilai error pembacaan arus.....	41
Gambar 4.5 Grafik jangkauan RF.....	43
Gambar 4.6 Monitoring pengiriman data antar PJU .....	45
Gambar 4.7 Data di Aplikasi thingspeak .....	47
Gambar 4.8 Grafik daya base.....	48
Gambar 4.9 Grafik arus base.....	49

Gambar 4.10 Grafik daya client 1.....	50
Gambar 4.11 Grafik arus client 1.....	51
Gambar 4.12 Grafik daya client 2.....	52
Gambar 4.13 Grafik arus client 2.....	53
Gambar 4.14 Data grafis perbandingan intensitas terhadap jarak .....	56
Gambar 4.15 pengambilan data intensitas cahaya pada jarak 50cm.....	56

## DAFTAR TABEL

Tabel 4.1 Perbandingan sensor PZEM004T dengan Watt meter .....	38
Tabel 4.2 Perbandingan sensor PZEM004T dengan ampere meter .....	40
Tabel 4.3 Data pengujian jangkauan RF .....	43
Tabel 4.4 Data daya base.....	48
Tabel 4.5 Data arus base.....	49
Tabel 4.6 Data daya client 1.....	50
Tabel 4.7 Data arus client 1.....	51
Tabel 4.8 Data daya client 2.....	52
Tabel 4.9 Data arus client 2.....	53
Tabel 4.10 Hasil Data perbandingan .....	54
Tabel 4.11 Data intensitas cahaya pada jarak permukaan sorot lampu .	55

HALAMAN INI SENGAJA DIKOSONGKAN

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi infrastruktur lampu penerangan jalan saat ini sudah berkembang cukup maju dengan kehadiran dan tuntutan percepatan inovasi teknologi 4.0.

Permasalahan pengembangan teknologi lampu penerangan jalan umum yang ada saat ini dirasa kurang efisien karena masih menggunakan sistem konsentrator pada 1 panel kontrol yang mengirimkan data teknis menjadi 1 untuk beberapa PJU tanpa adanya identitas pada tiap lampu. Begitu juga dengan teknologi komunikasi PJU berbasis internet dengan sistem Stand Alone atau berdiri sendiri yang memiliki kekurangan pada biaya pembuatan yang cenderung lebih mahal.

Melihat permasalahan tersebut, maka dibutuhkan lampu jalan yang dapat memantau parameter –parameter teknis dari LPJU secara spesifik untuk setiap lampu yang terdaftar dan mengirimkan data-data tersebut beserta status kerusakan lampu jalan secara cepat kepada penanggung jawab perawatan lampu jalan agar keamanan dan kenyamanan pengguna jalan tetap terjaga. Oleh karena itu, akan dirancang sebuah **Penerapan *Wireless Sensor Network* (WSN) Untuk Monitoring Daya Lampu Penerangan Jalan Umum (LPJU) Menggunakan Thingspeak.**

#### 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang maka dapat dirumuskan beberapa permasalahan antara lain sebagai berikut :

1. Bagaimana cara melakukan pengiriman data secara kolektif dari banyak lampu PJU?
2. Bagaimana cara mengolah data daya lampu PJU?
3. Bagaimana keefektifan *monitoring* beberapa Lampu PJU berbasis *aplikasi thingspeak* dengan menggunakan sistem Jaringan Sensor Nirkabel?

### **1.3 Batasan Masalah**

Untuk membatasi ruang lingkup tugas akhir supaya lebih terarah dan penyesuaian dengan kemampuan yang ada maka pembahasan dibatasi beberapa hal, antara lain:

1. Menggunakan 3 buah LPJU dengan 1 Base Station dan 2 Client.
2. Menggunakan topologi linier.
3. Base Station tidak menggunakan jaringan seluler, tetapi menggunakan jaringan WIFI.
4. *Web server* yang digunakan adalah web server tidak berbayar yaitu thingspeak.

### **1.4 Tujuan Penelitian**

Sesuai dengan uraian latar belakang, maka tujuan yang ingin dicapai dari penelitian ini adalah :

1. Mengetahui titik koordinat lampu yang mengalami kerusakan secara tepat.
2. Mengetahui indikasi kerusakan lampu secara *realtime* melalui website.
3. Mengetahui keefektifan *monitoring* LPJU berbasis *Aplikasi thingspeak* yang dipadukan dengan sistem WSN.

### **1.5 Metodologi Penelitian**

Dalam pengerjaan proyek akhir ini penulis menerapkan metodologi melalui beberapa tahap sebagai berikut :

#### **1.5.1 Studi Literatur**

Dalam proses perancangan dan pengujian sistem, perlu mempelajari tentang pengukuran besaran listrik pada lampu PJU dan bagaimana system pengiriman data antar lampu PJU sehingga data dapat disajikan pada server melalui jaringan internet. Pengambilan dan pengumpulan data-data serta dasar teori yang digunakan sebagai acuan dalam penyelesaian proyek akhir antara lain pengetahuan tentang sensor PZEM004T guna pembacaan besaran listrik dan sistem jaringan sensor. Literatur didapatkan dari buku – buku acuan, makalah-makalah acuan, dan beberapa forum diskusi di internet.



### **1.5.2 Perancangan Sistem**

Setelah mempelajari materi-materi yang ada, dimulailah perancangan sistem, penyediaan komponen, uji coba sebelum pembuatan *hardware* elektronik dan sesudah pembuatan *hardware* elektronik, mengujinya dan setelah itu pembuatan media observasi yaitu berupa Lampu PJU yang dimodifikasi dan disesuaikan terhadap *plant* yg dibuat.

Proyek akhir ini dibuat melalui beberapa tahap, diantaranya mendesain dan merancang alat, membuat *hardware* elektronik, pengecekan *hardware* elektronik, integrasi media tanam dan *hardware* elektronik, pembuatan program, pembuatan GUI di *thingspeak*, integrasi seluruh sistem.

### **1.5.3 Pembuatan *Software* sistem**

Setelah perancangan sistem selesai, selanjutnya perancangan perangkat lunak (*software*) dari sistem yang dikontrol. Perancangan perangkat lunak yaitu, membuat algoritma untuk mengakses sensor arus dan tegangan dan membuat tampilan GUI *thingspeak* sebagai perekam data daya, integrasi sistem sensor dan peraga PJU.

### **1.5.4 Pembuatan *Hardware* Sistem**

Pada tahap ini dilakukan pembuatan *hardware system* yang terbagi menjadi 2 yaitu elektronik dan peraga PJU yang pada akhirnya juga diuji. Proyek Akhir ini menggunakan PJU dengan daya sebesar 50watt dengan beberapa modifikasi agar dapat terkoneksi dengan perangkat sensor dan komunikasi. Pada penelitian ini dibuat 3 buah lampu PJU yang memiliki peran masing-masing yaitu Base Station, Client 1 dan Client 2 yang nantinya akan saling mengirimkan data secara realtime.

Sedangkan untuk *hardware* elektronik alat ini meliputi bagian-bagian seperti berikut.

- NodeMCU ESP32.
- Sensor PZEM004T
- Tranceiver RF.

### **1.5.5 Pengujian Sistem, Integrasi dan Pengambilan Data**

Pada tahap ini dilakukan pengujian tiap-tiap bagian dari *system* yang sebelumnya dibuat, setelah dirasa siap selanjutnya seluruh sistem tersebut diintegrasikan dengan rapi dan siap untuk diambil data. Pengujian Poyek Akhir dilakukan untuk mengetahui kelebihan dan kekurangan dari sistem sebagai referensi untuk perbaikan dan pengembangan alat untuk kemudian hari. Untuk Kemudian sistem secara keseluruhan diuji kembali sehingga sistem bekerja dengan baik. Berdasarkan tahapan ini dilakukan pula pengambilan data-data yang dibutuhkan untuk dicantumkan pada laporan Proyek Akhir.

### **1.5.6 Pembuatan Laporan Proyek Akhir**

Pada tahap ini dilakukan pembuatan atau penulisan laporan buku Proyek Akhir. Pada laporan tersebut dijelaskan mengenai semua hal yang berkaitan dengan pengerjaan Proyek Akhir, seperti penjelasan komponen yang dipakai, proses pembuatan alat, sistem kerja alat, data-data hasil pengujian alat, dan lain sebagainya. Diharapkan dengan penulisan laporan ini dapat bermanfaat sebagai bahan acuan dalam pembuatan alat serupa diwaktu yang akan datang.

## **1.6 Sistematika Penulisan**

Sistematika dalam pembahasan penyusunan Proyek Akhir direncanakan sebagai berikut:

- Pada Bab I berisikan latar belakang pembuatan alat pada proyek akhir, tujuan yang ingin dicapai, batasan masalah pada proyek akhir, serta sistematika pembahasan.
- Bab II meliputi literatur-literatur, teori dasar, serta referensi yang berguna sebagai acuan dan landasan dalam perencanaan dan pembuatan proyek akhir.
- Pada Bab III membahas tentang perencanaan dan pembuatan perangkat lunak maupun perangkat keras dari proyek akhir.
- Pada Bab IV berisikan pengujian dan analisa hasil yang diperoleh.
- Pada Bab V berisikan kesimpulan dan juga saran serta masukan setelah melihat analisa dari pengerjaan alat dan pengujian alat.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Lampu PJU

Lampu jalan adalah lampu yang digunakan untuk penerangan jalan pada malam hari sehingga pejalan kaki, pesepeda dan pengendara dapat melihat dengan lebih jelas jalan yang akan dilalui pada malam hari, sehingga dapat meningkatkan keselamatan lalu lintas dan keamanan dari para pengguna jalan. Secara spesifik peranan dari adanya Lampu PJU ini diantaranya:

- Alat bantu navigasi bagi pengguna jalan
- Meningkatkan keamanan suatu lingkungan
- Menghasilkan warna kontra antara obyek dan permukaan jalan
- Membantu keselamatan serta kenyamanan pengguna jalan, terkhusus pada malam hari.

##### 2.1.1 Jenis lampu PJU

- Lampu LVD

Lampu induksi efisiensi tinggi yang dipakai sebagai pengganti lampu penerangan jalan umum ( PJU ) dan lampu sorot. Lampu LVD dengan watt kecil mampu menghasilkan setara dengan lampu Metal Halide ( MHL ) ataupun lampu Sodium dengan daya lebih besar. Selebihnya adalah masa pakai atau umur pemakaian mencapai 100.000 jam. Pada umumnya lampu PJU dipakai jenis induksi LVD 80 watt dengan efisiensi yang tinggi setara dengan lampu merkuri 250w, yang terang dan kuat usia pemakaiannya.



Gambar 2.1 Lampu PJU LVD

Sistem lampu LVD atau Induksi terdiri 3 bagian penting :

- Ballast dengan frekuensi tinggi.
- Kumparan Induksi.
- Lampunya sendiri.

Ballast menghasilkan arus listrik frekuensi tinggi dan memberikan ke kumparan induksi. Arus yang melalui kumparan membuat fluktuatif medan elektromagnet pada lampu yang menarik meng-ionisasi atom - atom gas, mengeluarkan radiasi ultraviolet. Waktu radiasi ultraviolet ini bersentuhan dengan fosfor sehingga menghasilkan cahaya.

Karena lampu induksi ini tidak membutuhkan filamen atau elektroda untuk menyala, maka daya pakai jauh lebih lama daripada lampu pijar konvensional, lampu neon atau lampu dengan media gas tanpa membuat lampu menjadi hitam. Pengoperasian yang jauh efisien dengan depresiasi lumen yang jauh berkurang walau lampu ini sering dinyalakan, bahkan hingga 60.000 jam lampu induksi ini masih memberikan 80% keluaran cahayanya.

- **Lampu Neon TL**

Lampu fluorescent sering disebut lampu TL. Jenis ini banyak dipakai karena daya pakai relatif kecil dibanding lampu bohlam, selain itu lampu TL lebih dingin suhunya pada pemakaian yang sama.



Gambar 2.2 Lampu TL

Lampu TL sudah banyak dipakai luas di masyarakat untuk perumahan atau industri, keuntungan lampu TL ini seperti menghasilkan keluaran cahaya per watt lebih tinggi daripada lampu biasa ( incandescent lamp ). Contohnya pada penelitian mengukur 32 watt lampu TL menghasilkan 1700 lumen pada jarak 1 meter sedangkan 75 watt lampu bohlam biasa ( dengan filamen tungsten ) menghasilkan 1200 lumen. Dengan kata lain perbandingan efisiensi lampu TL dan lampu bohlam 53 : 16. Efisiensi ini diartikan sebagai hasil kekuatan cahaya dibagi dengan daya listrik yang digunakan. Meski lampu TL banyak keuntungan untuk penghematan daya, tetapi juga mempunyai kerugian :

- Biaya yang besar untuk pembelian set lampu TL
- Ruang yang lebih lebar satu set lampu TL
- Lampu LED

Lampu PJU yang didalamnya tersusun dari beberapa LED yang disusun sedemikian rupa dengan lensa sebagai reflektornya sehingga dapat menghasilkan cahaya yang sangat terang dengan nilai lux yang tinggi dengan daya yang jauh lebih rendah dari pada para pendahulunya.

Dengan digunakannya lampu LED, pengaturan daya dapat diatur sesuai yang diinginkan dengan mudah, daya lampu yang digunakan biasanya mulai dari 10 watt hingga 200 watt.



Gambar 2.3 Lampu PJU LED

## 2.1.2 Perkembangan LPJU

Dengan perkembangan teknologi yang makin pesat, perkembangan teknologi lampu PJU juga ikut berkembang dengan pesat, hal ini ditandai dengan adanya sistem Lampu PJU pintar yang dapat memonitor keadaan lampu PJU. Penerangan Jalan Umum Smart System merupakan penerapan dari IoT atau Internet of Things. IoT adalah sebuah konsep dimana internet dapat dihubungkan dengan benda-benda sehingga benda tersebut dapat dikontrol atau digunakan dengan perangkat nirkabel.



Gambar 2.4 Salah satu aplikasi Smart Lighting perusahaan lokal

Lampu PJU Smart System dapat dihidupkan dan dimatikan dengan kendali dari ruang kontrol sehingga lebih efisien dibandingkan dengan cara pengoperasian manual. Apabila ada lampu yang padam karena kerusakan, notifikasi akan muncul di layar ruang monitor sehingga operator ruang kontrol dapat langsung memberikan perintah kerja kepada petugas di lapangan untuk melakukan perbaikan segera. Notifikasi tersebut dapat menunjukkan lokasi dari lampu yang padam, jenis lampu, serta nomor token. Jika lampu sudah diperbaiki, akan muncul pula notifikasi bahwa lampu sudah dapat berfungsi seperti sedia kala. Smart System ini juga dapat menghemat listrik serta biaya yang digunakan

karena lampu dapat dinyalakan dan dimatikan tepat waktu, serta untuk daerah yang aktifitasnya minim pada malam hari, petugas dapat menurunkan pencahayaan serta penggunaan daya.

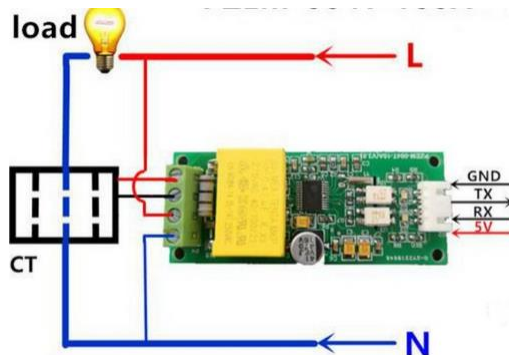
## 2.2 Sensor PZEM004-T

Sensor PZEM-004T adalah sensor yang dapat mengukur Arus, Tegangan, Power dan Energy dari listrik AC. Sensor ini mengeluarkan output dengan komunikasi serial.



Gambar 2.5 sensor PZEM004-T

Gambar berikut ini menunjukkan Typical Application dari modul sensor PZEM004-T.



Gambar 2.6 Rekomendasi rangkaian sensor PZEM004-T

### 2.2.1 Spesifikasi PZEM004-T

Berikut ini adalah spesifikasi dan fitur dari sensor PZM004-T:

- Fungsi pengukuran (voltage / tegangan, current / arus, active power).
- Power button clear / reset Energy (PZEM-004T V2.0)
- Power-down data storage function (cumulative power down before saving)
- Komunikasi Serial TTL
- Pengukuran Power / Daya : 0 ~ 9999kW
- Pengukuran Voltage / Tegangan : 80 ~ 260VAC
- Pengukuran Current / Arus : 0 ~ 100A
- Working voltage: 80 ~ 260VAC
- Rated power: 100A / 22000W
- Working Frequency: 45-65Hz
- Measurement accuracy: 1.0
- Factor daya rentang pengukuran 0.00-1.00
- Resolusi factor daya 0.01
- Akurasi factor daya 1%
- Frekuensi rentang pengukuran 45-65KHz
- Resolusi frekuensi 0.1Hz
- Akurasi frekuensi 0.5%

### 2.3 RF Modul EBYTE E220

E22-400T30D adalah modul Lora (UART) baru berdasarkan chip SX1268 RF SEMTECH. Ini memiliki beberapa mode transmisi yang bekerja pada 410.125 ~ 493.125MHz, teknologi spektrum penyebaran LoRa, output level TTL, kompatibel dengan 3.3V dan 5V IO port voltage.





Gambar 2.7 Bentuk fisik modul RF EBYTE E22

### 2.3.1 Fitur EBYTE E22

- Berdasarkan teknologi SX1268 LoRa, ini memungkinkan jarak yang lebih jauh dan kemampuan anti-interferensi yang lebih baik.
- Jaringan relai otomatis, multi-stage relay cocok untuk komunikasi jarak jauh, beberapa jaringan yang berjalan di area yang sama berjalan secara bersamaan;
- Pengguna untuk mengatur kunci komunikasi mereka sendiri dan tidak dapat dibaca, yang sangat meningkatkan kerahasiaan data pengguna;
- RSSI untuk mengevaluasi kualitas sinyal, meningkatkan jaringan komunikasi, dan jangkauan;
- Dengan LBT untuk memantau kebisingan lingkungan saluran sebelum mengirim data, dan untuk meningkatkan komunikasi;
- Konfigurasi parameter nirkabel, mengirim paket data perintah secara nirkabel, mengkonfigurasi atau membaca parameter modul nirkabel dari jarak jauh;
- Wake-on-air, yaitu fungsi konsumsi daya ultra-rendah, cocok untuk aplikasi bertenaga baterai;
- Dengan transmisi titik tetap, transmisi siaran, dan pemantauan saluran;
- ISM 433MHz dan 470Mhz bebas lisensi global untuk pembacaan meter
- Dalam mode tidur nyenyak, konsumsi daya adalah 2uA ;
- Dengan PA+LNA, jarak komunikasi yang diuji hingga 6 km ;
- Parameter disimpan setelah power-off. Setelah power-on, modul akan bekerja sesuai dengan parameter yang ditetapkan.
- Desain pengawas efisiensi tinggi, setelah pengecualian terjadi, modul akan secara otomatis memulai ulang dan terus bekerja sesuai dengan pengaturan parameter sebelumnya;
- Tingkat tanggal tayang 0.3kbps~62.5kbps;
- Catu daya 3.3V ~ 5.5V, catu daya lebih dari 5.0 V dapat menjamin kinerja terbaik ;
- Desain standar kelas industri, mendukung -40 ~ 85 ° C untuk bekerja dalam waktu lama
- Antarmuka SMA-K untuk antena eksternal.

## 2.4 NodeMCU ESP32

Mikrokontroler ESP32 dibuat oleh perusahaan bernama Espressif Systems. Salah satu kelebihan yang dimiliki oleh ESP32 yaitu sudah terdapat Wi-Fi dan Bluetooth di dalamnya, sehingga akan sangat memudahkan ketika kita belajar membuat sistem IoT yang memerlukan koneksi wireless. Mikrokontroler ESP32 memiliki keunggulan yaitu sistem berbiaya rendah, dan juga berdaya rendah dengan modul WiFi yang terintegrasi dengan chip mikrokontroler serta memiliki bluetooth dengan mode ganda dan fitur hemat daya menjadikannya lebih fleksibel.



Gambar 2.8 Modul ESP32 devkit-C

### 2.4.1 Spesifikasi

ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Pada mikrokontroler ini sudah tersedia modul WiFi dalam chip sehingga sangat mendukung untuk membuat sistem aplikasi Internet of Things. ESP32 sendiri tidak jauh berbeda dengan ESP8266 yang familiar di pasaran, hanya saja ESP32 lebih kompleks dibandingkan ESP8266. Berikut ini merupakan spesifikasi yang dimiliki oleh mikrokontroler ESP32 :

- Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.
- Memori: 520 KB SRAM.

- Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).
- Peripheral I/O: 12-bit SAR ADC (up to 18 channels), 2x 8-bit DACs, 10x touch sensors (capacitive sensing GPIOs), 4x SPI, 2x I2S interfaces, 2x I2C interfaces, 3x UART, SD/SDIO/CE-ATA/MMC/eMMC host controller, SDIO/SPI slave controller, Ethernet MAC interface, CAN bus 2.0, infrared remote controller (TX/RX, up to 8 channels), motor PWM, LED PWM (up to 16 channels), hall effect sensor, ultra low power analog pre-amplifier.
- Security : IEEE 802.11 standard security, secure boot, flash, encryption, 1024-bit, OTP (up to 768-bit for customers), cryptographic hardware acceleration (AES, SHA-2, RSA, ECC), random number generator (RNG).

### 2.4.2 Implementasi

Mikrokontroler ESP32 banyak digunakan dan sangat mendukung untuk pembuatan sistem aplikasi Internet of Things (IoT). Berikut ini merupakan contoh penggunaannya, sebagai berikut :

- Smart Security, Salah satu contoh perangkat smart security yang paling umum adalah kunci rumah yang menggunakan biometrik seperti sidik jari atau kode otentikasi. kegunaan alat tersebut untuk mengurangi pembobolan, kemalingan rumah, dan memperketat keamanan rumah dari orang yang ingin berbuat jahat. Selain itu juga untuk mengatasi beberapa ‘kecelakaan’ kecil seperti kehilangan kunci, akses rumah oleh tamu, dan akses ruangan yang memerlukan otorisasi khusus seperti kamar orang tua.
- Smart City, Di skala yang lebih luas, teknologi IoT juga menyentuh urusan tata kota. kegunaannya smart city ini untuk mengintegrasikan semua permasalahan kota menjadi satu. contohnya seperti CCTV, lampu lalu lintas, sampai integrasi sistem transportasi dalam kota. Semuanya bisa diakses langsung lewat smartphone yang terhubung ke internet.

## 2.5 Arduino IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui

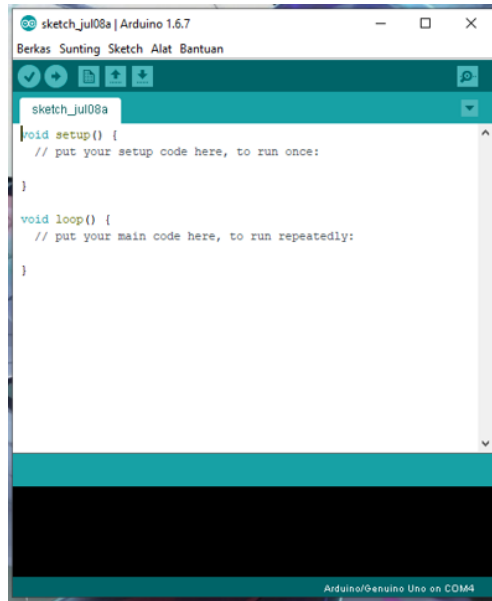
sintaks pemrograman[6]. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

### 2.5.1 Menulis Sketch

Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai *sketch*. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *.ino*. Teks editor pada Arduino Software memiliki fitur” seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan kamu dalam menulis kode program.

Pada Software Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.



Gambar 2.9 tampilan dari Software Arduino IDE.

Tombol- tombol pada toolbar, sebagai berikut:

- *Verify*  
Befungsi untuk melakukan checking kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum
- *Upload*  
Befungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh mesih alias si Arduino.
- *New*  
Befungsi untuk membuat *Sketch* baru
- *Open*  
Befungsi untuk membuka *sketch* yang pernah kamu buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke Arduino.
- *Save*  
Befungsi untuk menyimpan *Sketch* yang telah kamu buat.
- *Serial Monitor*  
Befungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan sketch pada port serialnya.

Serial Monitor ini sangat berguna sekali ketika kamu ingin membuat program atau melakukan *debugging* tanpa menggunakan LCD pada Arduino. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan error.

Fitur-fitur pada menuubar, sebagai berikut

- File

Memiliki sub menu sebagai berikut:

- ✓ **New**, berfungsi untuk membuat membuat sketch baru dengan bare minimum yang terdiri void setup() dan void loop().
- ✓ **Open**, berfungsi membuka sketch yang pernah dibuat di dalam drive.
- ✓ **Open Recent**, merupakan menu yang berfungsi mempersingkat waktu pembukaan file atau sketch yang baru-baru ini sudah dibuat.
- ✓ **Sketchbook**, berfungsi menunjukkan hirarki *sketch* yang kamu buat termasuk struktur foldernya.
- ✓ **Example**, berisi contoh-contoh pemrograman yang disediakan pengembang Arduino, sehingga kamu dapat mempelajari program-program dari contoh yang diberikan.
- ✓ **Close**, berfungsi menutup jendela Arduino IDE dan menghentikan aplikasi.
- ✓ **Save**, berfungsi menyimpan *sketch* yang dibuat atau perubahan yang dilakukan pada *sketch*
- ✓ **Save as...**, berfungsi menyimpan *sketch* yang sedang dikerjakan atau *sketch* yang sudah disimpan dengan nama yang berbeda.
- ✓ **Page Setup**, berfungsi mengatur tampilan page pada proses pencetakan.
- ✓ **Print**, berfungsi mengirimkan file sketch ke mesin cetak untuk dicetak.
- ✓ **Preferences**, disini kam dapat merubah tampilan *interface* IDE Arduino.
- ✓ **Quit**, berfungsi menutup semua jendela Arduino IDE. *Sketch* yang masih terbuka pada saat tombol *Quit* ditekan, secara otomatis akan terbuka pada saat Arduino IDE dijalankan.

- Edit

Memiliki sub menu sebagai berikut:

- ✓ **Undo/Redo**, berfungsi untuk mengembalikan perubahan yang sudah dilakukan pada *Sketch* beberapa langkah mundur dengan *Undo* atau maju dengan *Redo*.

- ✓ **Cut**, berfungsi untuk *remove* teks yang terpilih pada editor dan menempatkan teks tersebut pada *clipboard*.
- ✓ **Copy**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard*.
- ✓ **Copy for Forum**, berfungsi melakukan *copy* kode dari editor dan melakukan *formatting* agar sesuai untuk ditampilkan dalam forum, sehingga kode tersebut bisa digunakan sebagai bahan diskusi dalam forum.
- ✓ **Copy as HTML**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard* dalam bentuk atau format HTML. Biasanya ini digunakan agar code dapat *diembedded* pada halaman web.
- ✓ **Paste**, berfungsi menyalin data yang terdapat pada *clipboard*, kedalam editor.
- ✓ **Select All**, berfungsi untk melakukan pemilihan teks atau kode dalam halaman editor.
- ✓ **Comment/Uncomment**, berfungsi memberikan atau menghilangkan tanda // pada kode atau teks, dimana tanda tersebut menjadikan suatu baris kode sebagai komen dan tidak disertakan pada tahap kompilasi.
- ✓ **Increase/Decrease Indent**, berfungsi untuk mengurangi atau menambahkan indentasi pada baris kode tertentu. Indentasi adalah “tab”.
- ✓ **Find**, berfungsi memanggil jendela window *find and replace*, dimana kamu dapat menggunakannya untuk menemukan variabel atau kata tertentu dalam program atau menemukan serta menggantikan kata tersebut dengan kata lain.
- ✓ **Find Next**, berfungsi menemukan kata setelahnya dari kata pertama yang berhasil ditemukan.
- ✓ **Find Previous**, berfungsi menemukan kata sebelumnya dari kata pertama yang berhasil ditemukan.
- Sketch
  - Memiliki sub menu sebagai berikut:
  - ✓ **Verify/Compile**, berfungsi untuk mengecek apakah *sketch* yang kamu buat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang kamu buat akan dikompile kedalam bahasa mesin.
  - ✓ **Upload**, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.
  - ✓ **Upload Using Programmer**, menu ini berfungsi untuk menuliskan *bootloader* kedalam IC Mikrokontroler Arduino. Pada kasus ini

kamu membutuhkan perangkat tambahan seperti *USBasp* untuk menjembatani penulisan program *bootloader* ke IC Mikrokontroler.

- ✓ **Export Compiled Binary**, berfungsi untuk menyimpan file dengan ekstensi **.hex**, dimana file ini dapat disimpan sebagai arsip untuk di upload ke board lain menggunakan tools yang berbeda.
- ✓ **Show Sketch Folder**, berfungsi membuka folder *sketch* yang saat ini dikerjakan.
- ✓ **Include Library**, berfungsi menambahkan library/pustaka kedalam *sketch* yang dibuat dengan menyertakan sintaks `#include` di awal kode. Selain itu kamu juga bisa menambahkan library eksternal dari file **.zip** kedalam Arduino IDE.
- ✓ **Add File...**, berfungsi untuk menambahkan file kedalam *sketch* arduino (file akan dikopikan dari drive asal). File akan muncul sebagai tab baru dalam jendela *sketch*.

- Tools

Memiliki sub menu sebagai berikut:

- ✓ **Auto Format**, berfungsi melakukan pengatran format kode pada jendela editor
- ✓ **Archive Sketch**, berfungsi menyimpan *sketch* kedalam file **.zip**
- ✓ **Fix Encoding & Reload**, berfungsi memperbaiki kemungkinan perbedaan antara pengkodean peta karakter editor dan peta karakter sistem operasi yang lain.
- ✓ **Serial Monitor**, berfungsi membuka jendela serial monitor untuk melihat pertukaran data.
- ✓ **Board**, berfungsi memilih dan melakukan konfigurasi board yang digunakan.
- ✓ **Port**, memilih port sebagai kanal komunikasi antara software dengan hardware.
- ✓ **Programmer**, menu ini digunakan ketika kamu hendak melakukan pemrograman chip mikrokontroler tanpa menggunakan koneksi Onboard USB-Serial. Biasanya digunakan pada proses *burning bootloader*.
- ✓ **Burn Bootloader**, mengizinkan kamu untuk mengkopikan program *bootloader* kedalam IC mikrokontroler

- Help

Disini kamu bisa mendapatkan bantuan terhadap kegalauanmu mengenai pemrograman. Menu help berisikan file-file dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya. Selain itu pada menu help juga diberikan link untuk



menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.

### 2.5.2 Sketchbook

Arduino Software IDE, menggunakan konsep *sketchbook*, dimana *sketchbook* menjadi standar peletakan dan penyimpanan file program. Sketch yang telah kamu buat dapat dibuka dengan dari **File -> Sketchbook**, atau dengan menu **Open**.

### 2.5.3 Tabs, Multiple Files, dan Compilations

Mekanisme ini memungkinkan kamu untuk melakukan manajemen sketch, dimana lebih dari satu file dibuka dalam tab yang berbeda.

### 2.5.4 Uploading

Merupakan mekanisme untuk mengkopikan file **.hex** atau file hasil kompilasi ke dalam IC mikrokontroler Arduino. Sebelum melakukan *uploading*, yang perlu kamu pastikan adalah jenis board yang kamu gunakan dan COM Ports dimana keduanya terletak pada menu **Tools -> Board** dan **Tools -> Port**.

### 2.5.5 Library

Library/ Pustaka merupakan file yang memberikan fungsi ekstra dari sketch yang kamu buat, semisal agar Arduino dapat bekerja dengan *hardware* tertentu dan melakukan proses manipulasi data. Untuk menginstal *Library* pihak ketiga alias *Library* bukan dari Arduino, dapat dilakukan dengan Library Manager, Import file **.zip**, atau kopi paste secara manual di folder libraries pada Documents di platform Windows.

### 2.5.6 Serial Monitor

Serial monitor merupakan suatu jendela yang menunjukkan data yang dipertukarkan antara arduino dan komputer selama beroperasi, sehingga kamu bisa menggunakan serial monitor ini untuk menampilkan nilai hasil operasi atau pesan *debugging*. Selain melihat data, kamu juga bisa mengirimkan data ke Arduino melalui serial monitor ini, caranya

dengan memasukkan data pada *text box* dan menekan tombol *send* untuk mengirimkan data. Hal penting yang harus kamu perhatikan adalah menyamakan *baudrate* antara serial monitor dengan Arduino board. Untuk menggunakan kemampuan komunikasi serial ini, pada Arduino, di bagian fungsi void setup(), diawali dengan instruksi **Serial.begin** diikuti dengan nilai *baudrate*.

### 2.5.7 Preferences

*Preferences* mengatur tentang beberapa hal dalam penggunaan Arduino Software IDE, seperti ukuran font, lokasi dimana menyimpan sketbook, bahasa yang digunakan pada Arduino Software IDE, dan masih banyak lagi. Kamu bisa mengatur preferences pada menu file yang dapat dijumpai pada platform Windows dan Linux.

### 2.5.8 Language Support

*Language Support* merupakan pilihan bahasa yang dapat disesuaikan pada Software Arduino IDE. Bahasa Indonesia sudah ada loh. Language Support ini dapat ditemukan pada menu **file -> preferences** atau dengan menekan **Ctrl+Comma**

## 2.6 Internet Of Things

Internet of things adalah suatu konsep atau program dimana sebuah objek memiliki kemampuan untuk mentransmisikan atau mengirimkan data melalui jaringan tanpa menggunakan bantuan perangkat komputer dan manusia. Internet of things atau sering disebut dengan IoT saat ini mengalami banyak perkembangan.

Perkembangan IoT dapat dilihat mulai dari tingkat konvergensi teknologi nirkabel, microelectromechanical (MEMS), internet, dan QR (*Quick Responses*) Code. IoT juga sering diidentifikasi dengan RFID (*Radio Frequency Identification*) sebagai metode komunikasi.

Selain itu, juga mencakup teknologi berbasis sensor, seperti teknologi nirkabel, QR Code yang sering kita jumpai. Kemampuan dari IoT sendiri tidak perlu diragukan lagi. Banyak sekali teknologi yang telah menerapkan sistem IoT, sebagai contoh sensor cahaya, sensor suara dari teknologi Google terbaru, yaitu Google Ai, dan Amazon Alexa.

Dan yang terbaru saat ini, penerapan Smart City yang sudah dilakukan di beberapa negara maju, seperti China dan Jerman. Sehingga, segala

bentuk aktivitas penduduk suatu kota dapat termonitoring dengan baik oleh sistem dengan jaringan basis data berskala besar.

### 2.6.1 Bagian-bagian IoT

Setelah mengenal apa itu internet of things, selanjutnya masuk pada pembahasan mengenai unsur – unsur IoT. Setidaknya, terdapat lima unsur pembentuk dari internet termasuk juga kecerdasan buatan, konektivitas, sensor, dan lain sebagainya. Berikut merupakan penjabarannya:

- Artificial Intelligence (AI)

Artificial Intelligence (AI) atau dalam bahasa Indonesia berarti kecerdasan buatan merupakan merupakan sebuah penemuan yang dapat memberikan kemampuan bagi setiap teknologi atau mesin untuk berpikir (menjadi “smart”). Jadi, AI disini dilakukan dengan mengumpulkan berbagai data, pemasangan jaringan, dan pengembangan algoritma dari kecerdasan buatan.

Sehingga, dari yang awalnya sebuah mesin hanya dapat melaksanakan perintah dari pengguna secara langsung, sekarang dapat melakukan berbagai aktivitas sendiri tanpa menunggu instruksi dari pengguna. Misalnya saja, teknologi AI yang diterapkan pada robot pelayan di sebuah restoran di Jepang.

Dimana, kemampuan robot tersebut dapat berpikir layaknya seorang pelayan manusia asli. Karena di dalam sistem kendali robot tersebut telah menggunakan bantuan AI. Dengan mencakup berbagai sumber data dan informasi secara lengkap dan algoritma yang kompleks.

- Konektivitas

Konektivitas atau biasa disebut dengan hubungan koneksi antar jaringan. Di dalam sebuah sistem IoT yang terdiri dari perangkat kecil, setiap sistem akan saling terhubung dengan jaringan. Sehingga dapat menciptakan kinerja yang lebih efektif dan efisien.

Untuk standar biaya pemasangan jaringan tidak selalu membutuhkan jaringan yang besar dan biaya yang mahal. Anda juga dapat merancang sistem perangkat dengan menggunakan jaringan yang lebih sederhana dengan biaya yang lebih murah.

- Perangkat ukuran kecil

Di dalam perkembangan teknologi masa kini, semakin kecil sebuah perangkat maka akan menghasilkan biaya yang lebih sedikit,

namun efektifitas dan skalabilitas menjadi tinggi. Sehingga di masa yang akan datang, manusia dapat lebih mudah menggunakan perangkat teknologi berbasis IoT dengan nyaman, tepat, dan efisien.

- **Sensor**  
Sensor merupakan unsur yang menjadi pembeda dari IoT dengan mesin canggih yang lain. Dengan adanya sensor, mampu untuk mendefinisikan sebuah instrumen, yang mana dapat mengubah IoT dari jaringan standar yang cenderung pasif menjadi sistem aktif yang terintegrasi dengan dunia nyata.
- **Keterlibatan Aktif**  
Banyak mesin modern yang masih menggunakan keterlibatan (*engagement*) secara pasif. Namun, yang menjadi pembeda dari mesin yang lain, IoT telah menerapkan metode paradigma aktif dalam berbagai konten, produk, serta layanan yang tersedia.

### **2.6.2 Cara Kerja**

Cara kerja internet of things adalah memanfaatkan sebuah argumentasi dari algoritma bahasa pemrograman yang telah tersusun. Dimana, setiap argumen yang terbentuk akan menghasilkan sebuah interaksi yang akan membantu perangkat keras atau mesin dalam melakukan fungsi atau kerja.

Sehingga, mesin tersebut tidak memerlukan bantuan dari manusia lagi dan dapat dikendalikan secara otomatis. Faktor terpenting dari jalannya program tersebut terletak pada jaringan internet yang menjadi penghubung antar sistem dan perangkat keras. Tugas utama dari manusia adalah menjadi pengawas untuk memonitoring setiap tindakan dan perilaku dari mesin saat bekerja.

Kendala terbesar dari pengembangan Internet of things adalah dari sisi sumber daya yang cukup mahal, serta penyusunan jaringan yang sangat kompleks. Biaya pengembangan juga masih terlampau mahal dan tidak semua kota atau negara telah menggunakan IoT sebagai kebutuhan primer mereka.

### 2.6.3 Penerapan

Banyak sekali contoh dari penerapan IoT dalam kehidupan sehari – hari yang tanpa anda sadari sangat dekat dengan anda. Berikut merupakan beberapa contoh bidang yang telah menerapkan teknologi IoT.

- Kesehatan

Contoh internet of things yang pertama dalam bidang kesehatan. Saat ini, banyak sekali teknologi *advanced* yang dapat membantu kinerja dari dokter maupun tenaga medis. IoT juga membuat sebuah terobosan baru dalam pengembangan mesin dan alat medis untuk mendukung kinerja dari tenaga medis agar lebih efektif, tepat, dan mengurangi resiko kesalahan.

Salah satu contoh dari keberadaan IoT dalam dunia kesehatan adalah membantu dalam proses pendataan detak jantung, mengukur kadar gula tubuh, mengecek suhu tubuh dan lain sebagainya. Data yang diperoleh akan disimpan dalam penyimpanan data berskala besar.

Saat ini lebih dikenal dengan big data. Dengan menggunakan big data mampu membaca informasi dan data yang berupa angka atau teks secara cepat, dan efisien. Tenaga medis tidak perlu lagi untuk mencatat secara manual, karena semua informasi dapat ditampung dalam basis data dan akan dikirimkan pada mesin IoT untuk menjalankan tugas sesuai dengan algoritma yang dikembangkan.

- Energi

Dalam bidang energi, terdapat bervariasi permasalahan yang timbul. Mulai dari polusi atau pencemaran, pemborosan, dan berkurangnya pasokan sumber daya. Oleh karena itu, dengan adanya IoT sendiri mampu untuk mengurangi beberapa resiko tersebut. Misalnya saja, dengan penerapan sensor cahaya mampu untuk mengurangi penggunaan energi listrik.

Dengan sensor tersebut, mampu menangkap partikel cahaya, sehingga saat cahaya tersebut banyak maka lampu akan mati. Namun, saat tidak ada pasokan cahaya, maka lampu akan otomatis menyala.

Kemudian, juga dapat menerapkan pada fungsi penjadwalan yang dilakukan pada mesin oven, mesin pemanas yang telah terintegrasi dengan jaringan internet. Dan contoh konkret yang sering kita jumpai adalah pada smart TV yang telah menerapkan IoT untuk

metode pencarian channel disesuaikan dengan pilihan pengguna (*user*).

- **Transportasi**

Teknologi cerdas juga telah mencapai bidang transportasi umum. Biasanya, anda selalu mengendarai sebuah mobil sendiri sesuai dengan aturan dan kemampuan berkendara yang telah anda pelajari. Namun, apakah anda sudah mengetahui saat ini ada penemuan terbaru, dimana anda dapat menjalankan mobil tanpa mengemudi sendiri.

Mobil tersebut dapat berjalan sendiri sesuai dengan prosedur dan terprogram dengan baik. Jadi, anda dapat merasakan sensasi seperti pada sistem autopilot di pesawat. Tahap pengembangan kendaraan tersebut masih diujicobakan di beberapa negara maju.

Selain kendaraan, sistem lalu lintas juga termasuk dalam cakupan internet of things. Dengan IoT, mampu untuk mengontrol berbagai sistem lalu lintas saat kondisi macet maupun sepi. Sehingga, mampu mengurangi resiko angka kecelakaan dan pelanggaran lalu lintas yang terjadi.

- **Lingkungan Umum**

Contoh internet of things yang terakhir yaitu dalam bidang lingkungan umum. Dimana segala aktivitas manusia, tumbuhan, maupun hewan dapat dipantau dan diawasi dengan menggunakan teknologi IoT. Misalnya saja, untuk melakukan penelitian kualitas air harus dibutuhkan sumber informasi yang akurat dan terpercaya.

Dengan bantuan internet of things, mampu untuk mencari sumber data secara valid dan cepat. Tidak hanya itu, cakupan wilayah geografis yang disajikan juga cukup luas dan dapat menjangkau lebih banyak daerah. Dengan bantuan big data, permasalahan mengenai kecepatan transfer data dan pembacaan data data tertutupi dengan baik.

#### **2.6.4 Manfaat**

Setelah mengetahui dengan rinci mengenai contoh internet of things, berikutnya masuk pada pembahasan mengenai manfaat internet of things. Manfaat disini dapat dibagi menjadi tiga bagian.

- Memudahkan proses konektivitas  
Manfaat IoT yang pertama adalah memudahkan dalam proses konektivitas antar perangkat atau mesin. Semakin koneksi antar jaringan baik, maka sistem perangkat dapat berjalan dengan lebih cepat dan fleksibel.  
Anda mungkin masih banyak yang menggunakan alat konvensional, namun apabila anda mencoba untuk mengoperasikan sebuah sistem secara terpusat hanya melalui perangkat mobile, maka jawabannya yang pasti adalah dengan menggunakan teknologi cerdas.
- Ketercapaian efisiensi  
Manfaat internet of things yang kedua adalah tercapainya efisiensi kerja. Semakin banyak konektivitas jaringan yang terbentuk, semakin kecil pula jumlah penurunan waktu untuk melakukan tugas. Sehingga, aktivitas dan kinerja manusia menjadi lebih terbantu dengan adanya IoT.
- Meningkatkan efektivitas monitoring kegiatan  
Dengan menggunakan internet of things, efektivitas untuk mengontrol dan monitoring sebuah pekerjaan menjadi lebih mudah. Selain itu, teknologi cerdas juga mampu untuk memberikan rekomendasi atau alternatif pekerjaan yang lebih mudah bagi pengguna.

## 2.7 ThingSpeak

Thingspeak.com merupakan platform Internet of Things dibagian cloud dimana kita dapat mengirim atau menerima suatu data dengan protokol komunikasi HTTP dan juga dapat menampilkan nilai data melalui dashboard gratis yang diberikan[11].



Gambar 2.10 Tampilan depan website thingspeak

### 2.7.1 Fungsi

Thingspeak.com berfungsi sebagai pengumpul data yang berasal dari perangkat node berupa sensor-sensor yang sudah terhubung ke internet dan juga memungkinkan pengambilan data dari perangkat lunak untuk keperluan visualisasi ,notifikasi, kontrol dan analisis historis data.

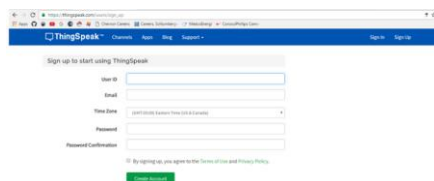
Unsur utama dari ThingSpeak adalah saluran, yang berisi bidang data, bidang lokasi, dan bidang status. Setelah membuat saluranThingSpeak, kita bisa menulis data ke saluran proses dan melihat hasil data lewat MATLAB. Dari situ dapat dilihat reaksi terhadap data dengan *tweet* dan alert lainnya.

### 2.7.2 Fitur

- Open API
- Real-time data collection
- Geolocation data
- Data processing
- Data Visualizations
- Device status messages
- Plugins

### 2.7.3 Langkah Penggunaan

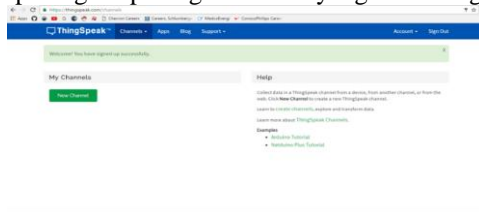
- Kita harus memiliki akun Thingspeak.com dengan mengisi User ID, Email dan juga Password. Pastikan email yang digunakan aktif karena digunakan untuk konfirmasi akun dan pemberitahuan lebih lanjut.



Gambar 2.11 Tampilan sign up thingSpeak

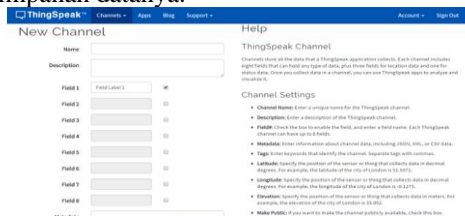


- Membuat Channel baru dengan cara menekan tombol "New Channel" yang nantinya ini digunakan untuk menampilkan data yang dikirim dari perangkat-perangkat sensor yang terhubung ke internet.



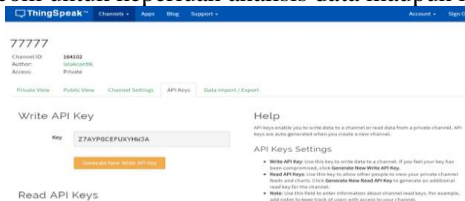
Gambar 2.12 Tampilan saat akan membuat channel baru

- Langkah ketiga adalah mengisi rincian untuk saluran yang baru kita buat. Nama saluran bebas dalam pelabelannya dan ini sekaligus menjadi nama data tersimpan atau menjadi proyek kerja kita . Kita boleh mengkosongkan beberapa field jika memang tidak digunakan dalam penyimpanan datanya.



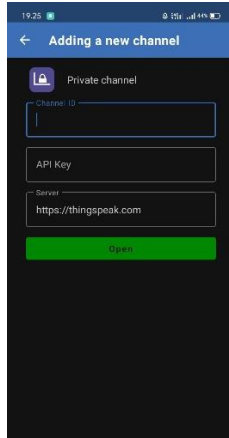
Gambar 2.13 Pengaturan channel baru

- Langkah keempat kita bisa menggunakan API Key yaitu API Write yang berguna untuk keperluan mengirim data ke thingspeak.com dan juga API Read yang digunakan untuk mengambil data dari thingspeak.com untuk keperluan analisis data maupun kontrol.



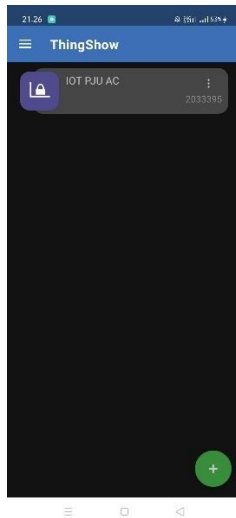
Gambar 2.14 Setting API Key

- Langkah kelima kita menggunakan aplikasi Thingspeak yang bernama Thingshow bisa dicari digoogle atau playstore, lalu masukkan ide dan api key.



Gambar 2.15 tampilan new channel

- Langkah keenam menu tampilan monitoring pju terlihat dihalaman utama.



Gambar 2.16 tampilan menu aplikasi thingspeak

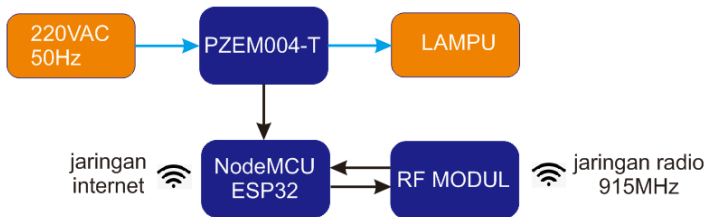
## BAB III

### PERANCANGAN SISTEM

#### 3.1 Perencanaan Sistem

Perangkat keras secara garis besar terdiri dari NodeMCU ESP32, sensor besaran listrik PZEM004-T, dan perangkat RF untuk komunikasi antar PJU. Untuk perangkat lunak menggunakan bahasa C pada Arduino IDE, karena cukup mudah digunakan ketimbang bahasa *assembly* dan sudah familiar untuk digunakan. Sedangkan untuk *Graphical User Interface* pada website, digunakan tampilan bawaan dari *broker iot* yaitu thingspeak.

Perencanaan sistem dalam mengerjakan proyek akhir ini dikerjakan berdasarkan blok diagram yang telah dibuat seperti Gambar 3.1 dibawah ini :

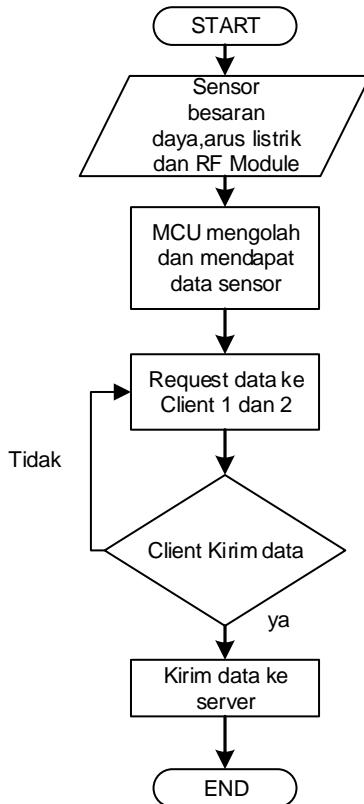


Gambar 3.1 Blok diagram sistem

Berdasarkan pada blok diagram di atas secara keseluruhan gambaran sistem yang dibangun pada perancangan dan pembuatan alat ini terdiri dari beberapa bagian yaitu: *input*, pemroses data, dan *output*.

- a. Blok pertama adalah blok *input* sistem yang terdiri dari sensor besaran listrik PZEM004-T yang dibutuhkan untuk pengukuran daya pada tiap titik pemasangan lampu yang dirangkai secara parallel terhadap sumber listrik untuk pengukuran tegangan dan sensor koil digunakan untuk pengukuran arus AC.
- b. Blok kedua adalah blok mikrokontroler atau dapat kita sebut *nodeMCU*, yang berfungsi untuk menjalankan atau mengeksekusi perintah, mengolah data dari *input* yang berupa sensor dan berbagai aktuator yang digunakan pada sistem ini.

- c. Blok ketiga adalah bagian dari *output*, pada penelitian ini bagian output untuk lampu PJU Base Station berupa data yang dikirimkan melalui jaringan internet, sedangkan pada client berupa perangkat RF untuk komunikasi antar lampu PJU.



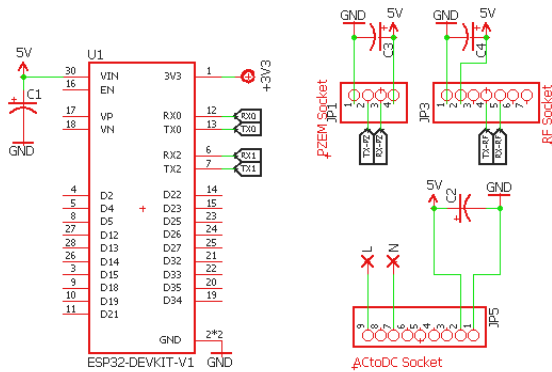
Gambar 3.2 Flowchart Sistem monitoring

Lampu PJU dinyalakan langsung menggunakan sumber tegangan ac 220Volt 50hz. Dari koneksi antara sumber tegangan ke beban tersebut dipasangkan sebuah sensor yang akan menghasilkan data daya dalam bentuk parsing serial UART. Data tersebut kemudian disimpan dan dikirimkan ke lampu PJU base station, saat data dari semua PJU sudah terkumpul maka data tersebut secara utuh dikirimkan ke Thingspeak

melalui jaringan internet. Untuk kebutuhan *logging* data dapat kita lihat maupun simpan pada server thingspeak.

### 3.2 Perencanaan dan Perancangan Perangkat Keras Elektronik

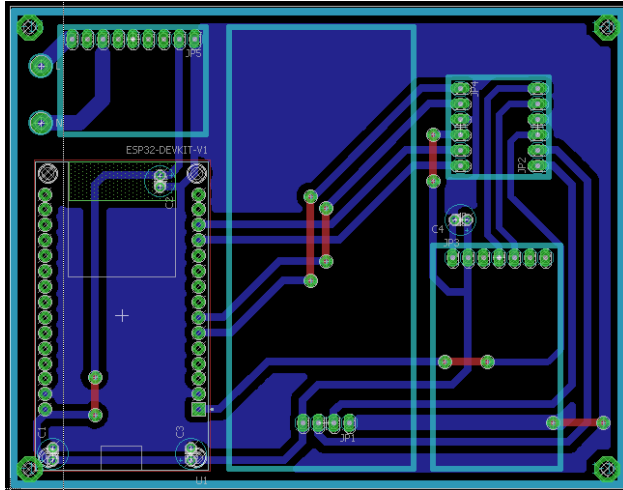
Perangkat keras elektronik direncanakan terdiri dari minimum sistem nodeMCU, sensor PZEM004T, AC to DC converter, dan RF modul.



Gambar 3.3 Rancangan skematik



Gambar 3.4 Hasil Hardware elektronik tampak depan

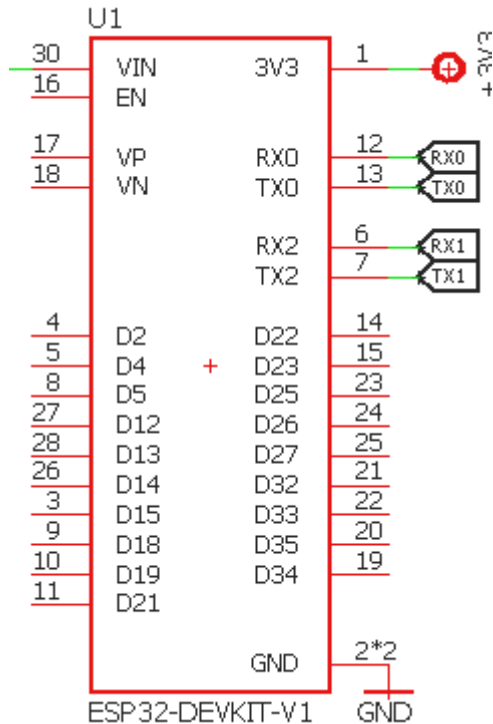


Gambar 3.5 Desain layout PCB rangkaian monitoring

### 3.2.1 NodeMCU

Sebagai pengolah data, sistem minimum sangat diperlukan, dalam sistem minimum terdapat IC mikrokontroler dan komponen pendukung lainnya. Dalam alat ini mikrokontroler yang digunakan ialah ESP32. Alat yang dibuat ini membutuhkan beberapa port komunikasi serial untuk RF dan untuk sensor PZEM004T. dengan kebutuhan tersebut maka digunakan ESP32 yang memiliki port serial ganda dan modul wifi yang sudah tertanam dalam hardware.

Konfigurasi port input/output dari mikrokontroler adalah sebagai berikut:



Gambar 3.6 Konfigurasi pin I/O ESP32 ke interface

Dari gambar 3.7 dapat kita perhatikan bahwa ada 2 buah port serial yang digunakan, port serial 0 digunakan untuk bertukar data dengan sensor PZEM004T, sedangkan port serial 1 digunakan untuk system komunikasi dengan perangkat RF yang digunakan sebagai pengirim data antara lampu PJU satu dengan yang lainnya.

### 3.2.2 Sensor PZEM004T

Sensor yang digunakan ini melakukan sensing melalui pengukuran langsung terhadap jala-jala listrik dan melalui CT (Current Tranformer). Dari kedua kombinasi pengukuran tersebut menghasilkan turunan besaran listrik lainnya. Di dalam modul yang digunakan, data dari perangkat

sensor sudah diolah secara langsung di dalam modul menggunakan mikrokontroler tersendiri yang kemudian data hasil pengolahan tersebut dikemas dalam sebuah paket data serial UART yang nantinya ada diminta dan diterima oleh NodeMCU ESP32.



Gambar 3.7 Aplikasi Sensor tegangan pada kontroler

Modul sensor tersebut dihubungkan dengan sumber tegangan 5volt. Daftar komponen yang digunakan untuk sebagai berikut

- PZEM004T & CT
- Kapasitor filter 10uf/16volt

### 3.2.3 RF modul

Modul RF yang digunakan dalam penelitian ini adalah EBYTE E220 dengan frekuensi kerja yang di set pada 915MHz dengan LoRa sebagai pemroses sinyal dan keluaran data sudah berupa data serial UART yang nantinya akan mengumpankan data ke NodeMCU ESP32.



Gambar 3.8 aplikasi modul sensor arus pada kontroler

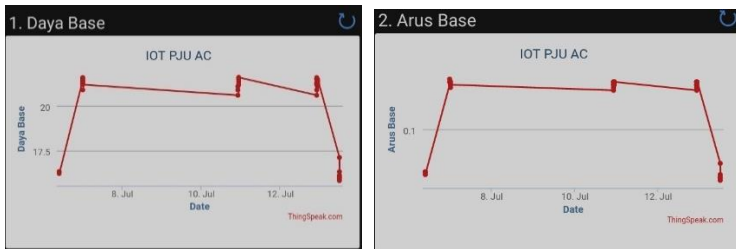
Modul sensor tersebut dihubungkan dengan sumber tegangan 5volt. Daftar komponen yang digunakan untuk sebagai berikut

- EBYTE E220-900T22D LoRa
- Kapasitor filter 10uf/16volt



### 3.3 ThingSpeak

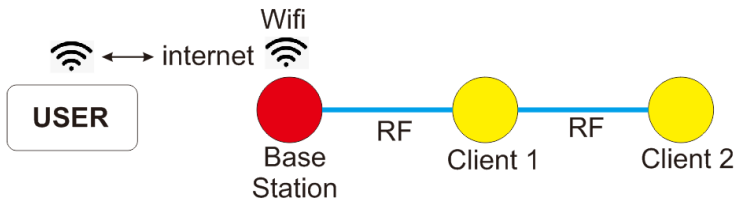
Dalam penelitian ini untuk memonitoring nilai tegangan dan arus digunakan sebuah interface berupa sebuah website yang nantinya akan menyimpan dan menampilkan data yang dikirim oleh perangkat keras. Selain dari website, juga dapat diakses melalui apk pada smartphone, pada penelitian ini digunakan aplikasi pihak ke 3 yaitu Thingshow yang mengambil data dari thingspeak dengan cara memasukkan channel ID dari field yang buat di thingspeak.



Gambar 3.9 Tampilan data daya, arus pada aplikasi thingspeak

### 3.4 Sistem WSN

*Wireless sensor network* adalah sebuah kumpulan node yang dapat berupa sensor yang akan melakukan pengambilan data pada parameter ukur dan kemudian dikirimkan pada sebuah node sentral atau sebuah server untuk dilakukan pengolahan data. Node-node yang ada pada WSN merupakan sensor yang diletakkan pada titik-titik pada sebuah area yang ingin diketahui besarnya.



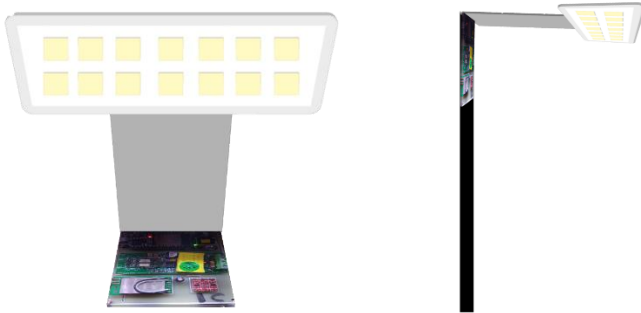
Gambar 3.10 Pengiriman data dengan topologi Linier

Topologi jaringan yang digunakan dalam penelitian ini ialah topologi Linier, dimana data hasil pengolahan sensor dikirimkan dari

client terakhir hingga client pertama atau sebaliknya, hingga data tersebut diterima oleh *Base Station* secara utuh.

### 3.5 Desain alat

Desain adalah suatu perencanaan yang dilakukan sebelum pembuatan suatu objek, sistem, komponen, atau struktur.



Gambar 3.11 Desain penerangan jalan umum

## **BAB IV**

### **ANALISA DAN PENGUJIAN**

Setelah perencanaan dan pembuatan alat dilakukan maka dilaksanakan pengujian untuk mengetahui sistem yang sudah dirancang berjalan sesuai atau tidak dengan teori maupun hasil perhitungan. Pengujian tersebut dilakukan dalam beberapa tahap. Pengujian pertama dilakukan secara terpisah, kemudian pengujian secara keseluruhan atau sistem yang sudah terintegrasi. Pengujian yang dilakukan secara terpisah adalah :

1. Pengujian PZEM004-T
2. Pengujian Pengiriman data RF
3. Pengujian hasil topologi Linier
4. Pengujian intensitas cahaya

#### **4.1 Pengujian PZEM004-T**

##### **4.1.1 Tujuan**

- Untuk mengetahui karakteristik sensor PZEM004-T.
- Untuk memastikan data dari sensor dapat terbaca oleh mikrokontroler

##### **4.1.2 Proses Pengambilan Data**

Data diambil dengan cara menyalakan lampu PJU yang kemudian hasil dari pembacaan sensor di tampilkan pada serial monitor untuk dibandingkan dengan watt meter yang terpasang pada ujung steker dari lampu PJU. Pengambilan data sensor PZEM00T dilakukan pada salah satu lampu PJU saja.

##### **4.1.3 Hasil dan Analisa**

PZEM004T merupakan sensor praktis yang secara langsung melakukan pengolahan data di dalam modul, sehingga data yang dikeluarkan sudah berupa data jadi berbentuk serial UART yang mudah diakses. Dari data yang ada menunjukkan rata-rata presentase error yang ditunjukkan oleh pzem004t ialah sebesar 0,99% dari 10 percobaan yang diambil. nilai error terbesar yaitu 1.27 dengan perbedaan pembacaan

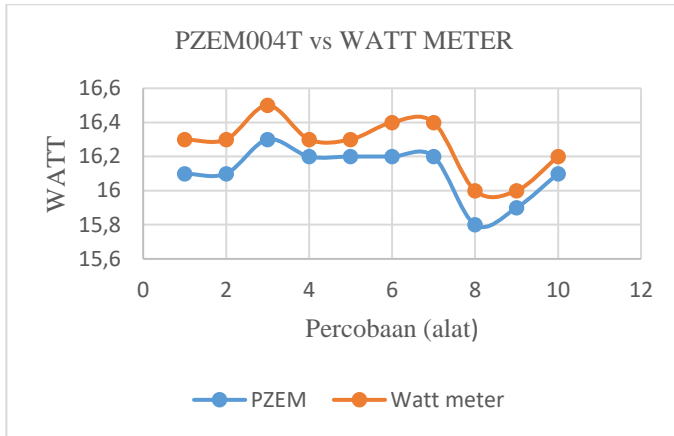
sebesar 0,2 watt dan error terkecil sebesar 0.62 dengan selisih pembacaan sebesar 0,1 watt. Jika dilihat dari spesifikasi yang diterbitkan oleh produsen sensor, resolusi pengukuran daya sebesar 0,1watt dengan akurasi sebesar 0,5%. Jika dibandingkan dengan rata-rata error dari pembacaan sensor maka sensor yang digunakan dalam penelitian ini memenuhi kriteria.

Tabel 4.1 Perbandingan sensor pzem004t dengan watt meter

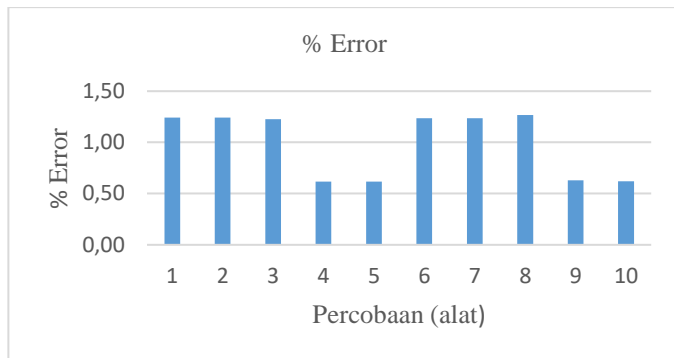
<b>Percobaan (alat)</b>	<b>PZEM (watt)</b>	<b>Watt meter</b>	<b>% Error</b>
1	16.1	16.3	1.24
2	16.1	16.3	1.24
3	16.3	16.5	1.23
4	16.2	16.3	0.62
5	16.2	16.3	0.62
6	16.2	16.4	1.23
7	16.2	16.4	1.23
8	15.8	16	1.27
9	15.9	16	0.63
10	16.1	16.2	0.62



Gambar 4.1 percobaan dengan alat watt meter



Gambar 4.2 Grafik perbandingan PZEM004T dengan Watmeter



Gambar 4.3 Grafik besar nilai error pembacaan watt

$$\% Error = \frac{PZEM - Watt Meter}{Watt Meter} \times 100\% \quad (4.1)$$

$$= \frac{16,3 - 16,1}{16,3}$$

$$= \frac{-0,2}{16,3} = 0,01$$

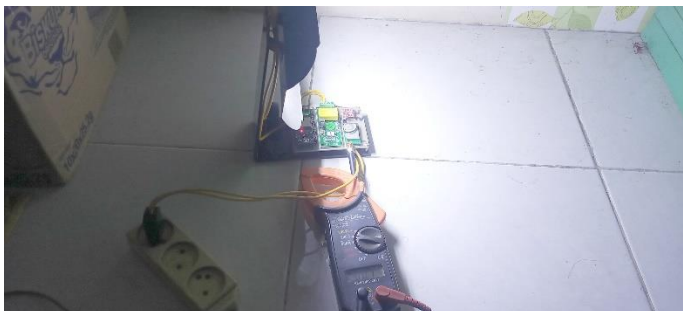
$$= 0,01 \times 100$$

$$= 0,01 \% Error$$

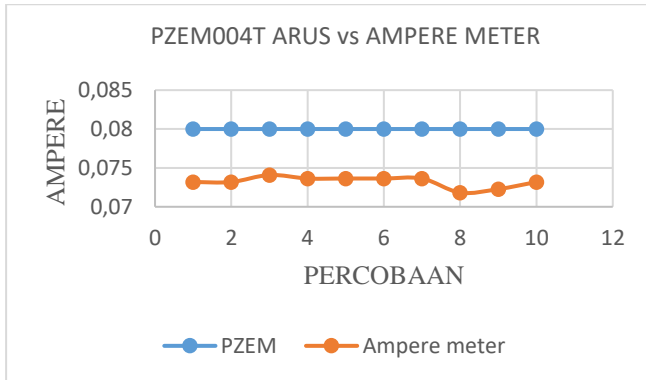
Pengukuran nilai Arus pada beban lampu dilakukan pada lampu PJU base station menghasilkan nilai yang sangat konstan pada besarnya arus yaitu 0.08 Ampere pada semua percobaan, sedangkan yang ditunjukkan oleh ampere meter cukup fluktuatif dengan selisih nilai 0,006A hingga 0,008A atau 6mA hingga 8mA. Dengan resolusi pengukuran hingga 0,1mA, hasil dari pembacaan pzem004T pada beban yang sama menunjukkan hasil yang stabil.

Tabel 4.2 Perbandingan sensor pzem004t dengan ampere meter

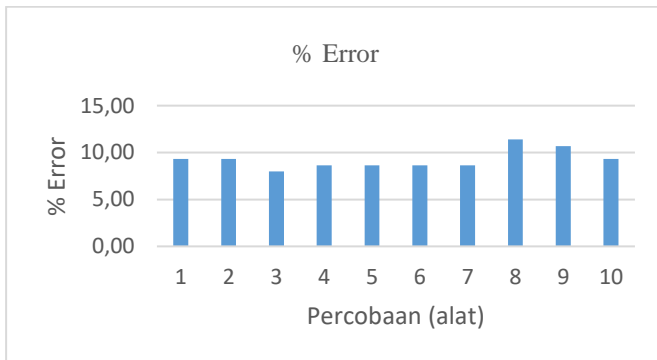
<b>Percobaan (alat)</b>	<b>PZEM (Ampere)</b>	<b>Ampere meter</b>
1	0.08	0.073
2	0.08	0.073
3	0.08	0.074
4	0.08	0.074
5	0.08	0.074
6	0.08	0.074
7	0.08	0.074
8	0.08	0.072
9	0.08	0.072
10	0.08	0.073



Gambar 4.4 percobaan dengan alat ampere meter



Gambar 4.5 Grafik terbandingan PZEM004T dengan Amperemeter



Gambar 4.6 Grafik besar nilai error pembacaan arus

$$\% Error = \frac{PZEM - \text{Ampere Meter}}{\text{Ampere Meter}} \times 100\% \quad (4.2)$$

$$= \frac{0,08 - 0,073}{0,073}$$

$$= \frac{0,08}{0,073} = 0,09$$

$$= 0,09 \times 100\%$$

$$= 0,09 \% \text{ Error}$$

## **4.2 Pengujian Pengiriman data RF**

### **4.2.1 Tujuan**

- Untuk mengetahui seberapa jauh RF dapat mengirimkan data

### **4.2.2 Proses Pengambilan Data**

Data diambil dengan menggunakan 2 buah PJU dengan 1 buah PJU sebagai pengirim data dan PJU lain sebagai penerima data. Data diambil dari jarak 100 meter hingga 1000 meter dengan kondisi jalan yang lurus, dimana PJU pengirim data dibawa menjauh dari PJU penerima data.

### **4.2.3 Hasil dan Analisa**

Modul RF menggunakan EBYTE dengan frekuensi 915Mhz memiliki jangkauan hingga 5Km dengan kondisi area terbuka. Data yang dikirimkan sejumlah 7 karakter yang diawali karakter # dan diakhiri karakter \*. Pengambilan data diuji pada setiap jarak 100 meter. Pada jarak 100 hingga 700 meter, data yang diterima penuh tanpa terpotong berjumlah utuh 7 karakter. Saat pengujian mulai di 800 hingga 1000 meter, data tidak utuh dan sama sekali tidak mengirimkan data. Hal tersebut terjadi karena banyak faktor seperti adanya kendaraan, kondisi medan yang tinggi rendah, gain antenna dan ketinggian antena. Melihat dari kondisi lampu penerangan jalan yang ada di setiap jalan yang dipasang dengan jarak cukup dekat dengan jarak di bawah 500 meter, maka RF yang digunakan ini cukup untuk digunakan sebagai piranti yang baik untuk pengiriman data antar PJU.

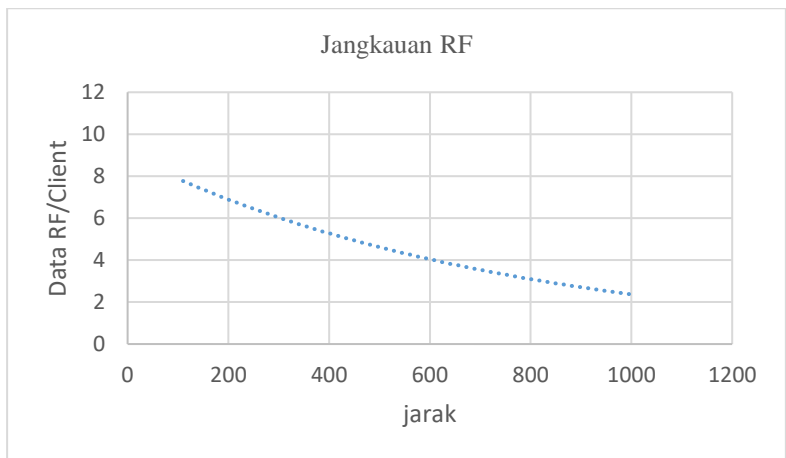


Tabel 4.3 Data pengujian jangkauan RF

Jarak	Data RF/Client	Hasil
100 meter	#REQ1*	OK
200 meter	#REQ1*	OK
300 meter	#REQ1*	OK
400 meter	#REQ1*	OK
500 meter	#REQ1*	OK
600 meter	#REQ1*	OK
700 meter	#REQ1*	OK
800 meter	#REQ1	-
900 meter	-	-
1000 meter	-	-

Keterangan :

1. #REQ1\* = data client telah dikirim.
2. OK = data client berhasil diterima.
3. “-” = data tidak ada.



Gambar 4.5 grafik jangkauan RF

### 4.3 Pengujian hasil topologi linier

#### 4.3.1 Tujuan

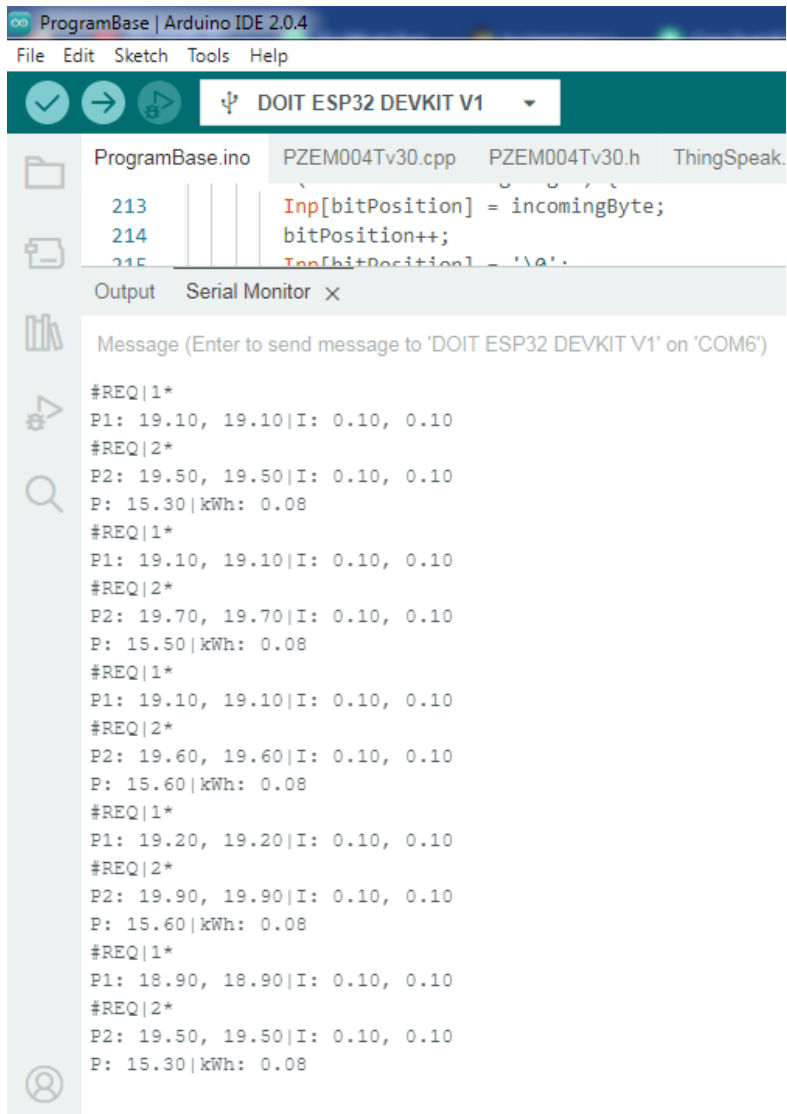
Untuk mengetahui apakah pengiriman data antar PJU dengan topologi linier efektif atau tidak

### **4.3.2 Proses Pengambilan Data**

Data diambil dengan menyalakan bersamaan 3 buah lampu PJU. Dengan 1 PJU sebagai base station dan 2 PJU sebagai client yang nantinya akan menyampaikan datanya secara beruntun sesuai urutannya. Dan dimonitoring melalui serial terminal pada base station dan secara keseluruhan pada website thingspeak.

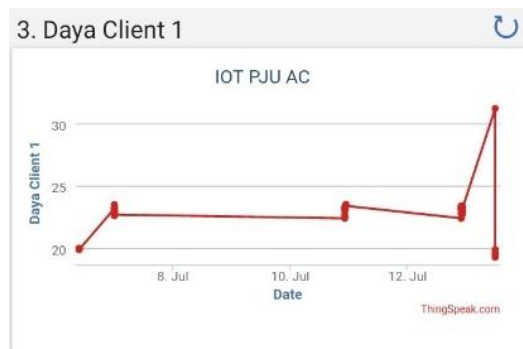
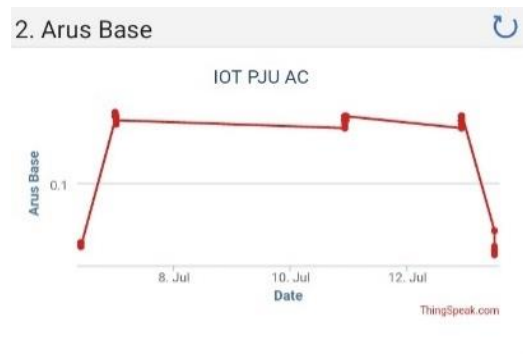
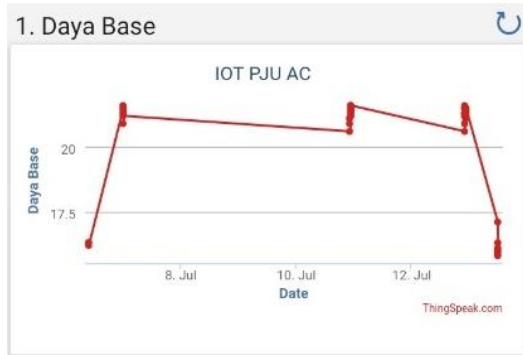
### **4.3.3 Hasil dan Analisa**

Proses pengiriman data dari setiap client dimulai dari perintah base station untuk meminta data dari tiap-tiap PJU client. Data diminta secara berurutan oleh base. Untuk meminta data dari client 1 maka base station akan mengirimkan perintah #REQ|1\* dan Client 1 akan mengirimkan data daya dan daya aktif dengan susunan P1: Data daya|kwh:Data daya aktif. Sedangkan untuk meminta data dari client 2 maka base akan mengirimkan perintah #REQ|2\* kepada client1 dan meneruskannya ke client 2, saat perintah sudah diterima, data dari client 2 akan dilewatkan melalui client 1 dan diteruskan langsung ke Base station. Hal ini dimaksudkan untuk meminimalkan adanya data yang tidak terkirim karena RF tidak mampu menjangkau jarak dari posisinya ke base station.



Gambar 4.6 Monitoring pengiriman data antar PJU

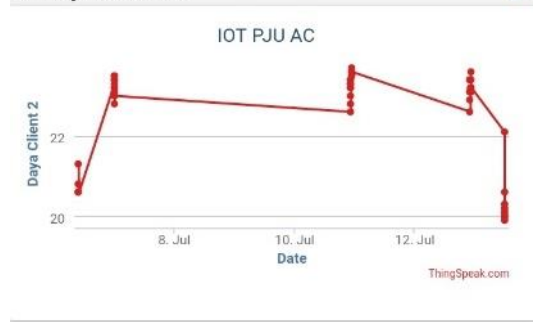
Data semua client yang diterima beserta data sensor dari base station disimpan sementara dan dikumpulkan menjadi 1 dan dikirimkan ke thingspeak seperti yang ditampilkan pada Gambar 4.7



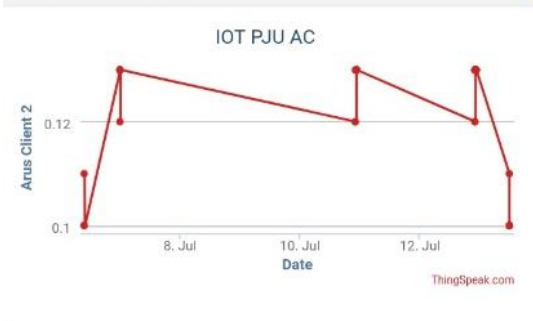
#### 4. Arus Client 1



#### 5. Daya Client 2



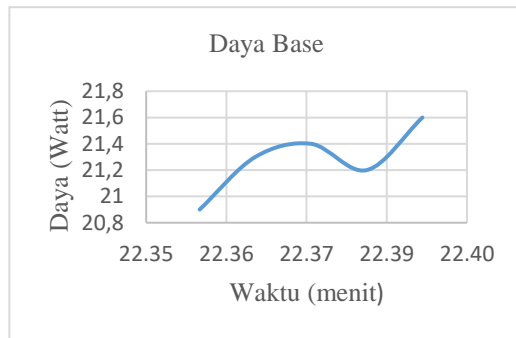
#### 6. Arus Client 2



Gambar 4.7 Data di aplikasi thingspeak

Tabel 4.4 Data Daya Base

Daya Base	
Waktu (Menit)	Daya (Watt)
22:36	20,9
22:37	21,3
22:38	21.4
22:39	21,2
22:40	21,6



Gambar 4.8 Grafik Daya Base

Perhitungan Daya, Arus base dan client menggunakan rumus :

$$P = V.I \quad (4.3)$$

$$I = \frac{P}{V} \quad (4.4)$$

Daya Base

Sebuah PJU memerlukan tegangan 220V dan arus listrik sebesar 0,116 Ampere untuk menyalahkannya. Berapa daya listrik yang dikonsumsi ?

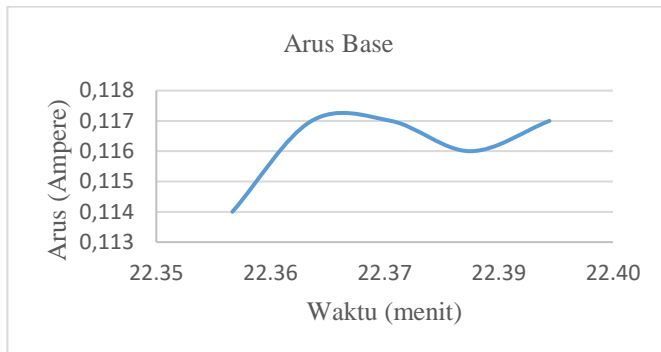
$$P = V \times I = 220V \times 0,116 A$$

$$P = 25,0 \text{ Watt}$$

Jadi PJU tersebut akan mengonsumsi daya listrik sebesar 25,0 Watt.

Tabel 4.5 Data Arus Base

Arus Base	
Waktu (Menit)	Arus (Ampere)
22:36	0,114
22:37	0,117
22:38	0,117
22:39	0,116
22:40	0,117



Gambar 4.9 Grafik Arus Base

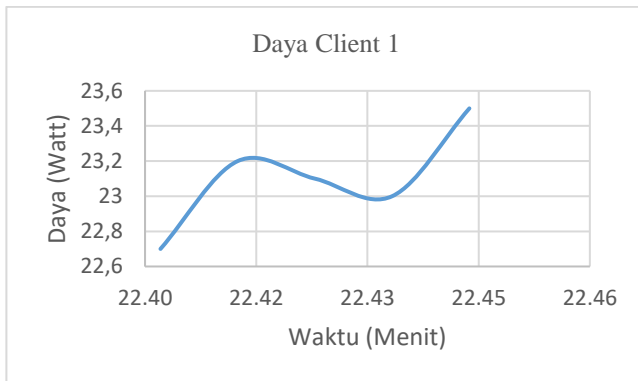
Arus Base

$$I = \frac{20,9}{220} = 0,095 \text{ Ampere}$$

Jadi PJU pada Base Station akan menghantar Arus listrik sebesar 0,095 Ampere.

Tabel 4.6 Data Daya Client 1

Daya Client 1	
Waktu (Menit)	Daya (Watt)
22:41	22,7
22:42	23,2
22:43	23,1
22:44	23
22:45	23,5



Gambar 4.10 Grafik Daya Client 1  
Daya Client 1

Sebuah PJU memerlukan tegangan 220V dan arus listrik sebesar 0,13 Ampere untuk menyalahkannya. Berapa daya listrik yang dikonsumsi ?

$$P = V \times I = 220V \times 0,12 A$$

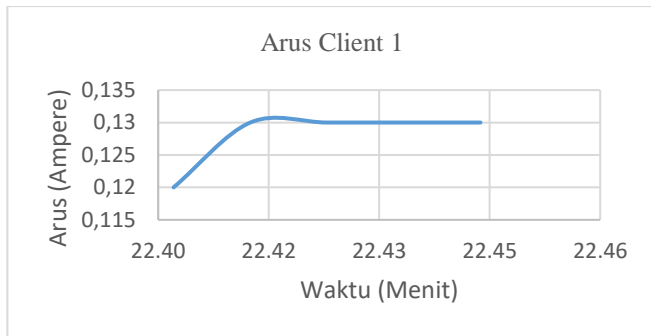
$$P = 26,4 \text{ Watt}$$

Jadi PJU tersebut akan mengonsumsi daya listrik sebesar 26,4 Watt.



Tabel 4.7 Data Arus Client 1

Arus Client 1	
Waktu (Menit)	Arus (Ampere)
22:41	0,12
22:42	0,13
22:43	0,13
22:44	0,13
22:45	0,13



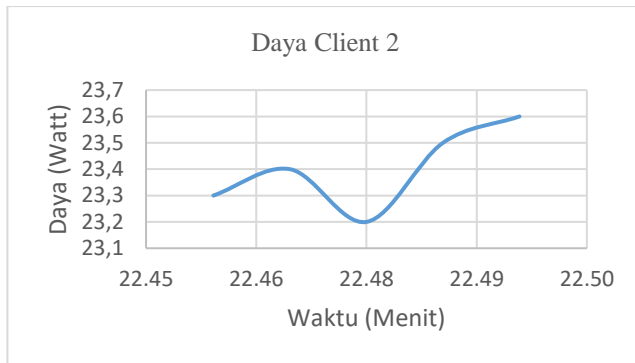
Gambar 4.11 Grafik Arus Client 1  
Arus Client 1

$$I = \frac{22,7}{220} = 0,10 \text{ Ampere}$$

Jadi PJU pada Client 1 akan menghantar Arus listrik sebesar 0,10 Ampere.

Tabel 4.8 Data Daya Client 2

Daya Client 2	
Waktu (Menit)	Daya (Watt)
22:46	23,3
22:47	23,4
22:48	23,2
22:49	23,5
22:50	23,6



Gambar 4.12 Grafik Daya Client 2

- Daya Client 2

Sebuah PJU memerlukan tegangan 220V dan arus listrik sebesar 0,13 Ampere untuk menyalahkannya. Berapa daya listrik yang dikonsumsi ?

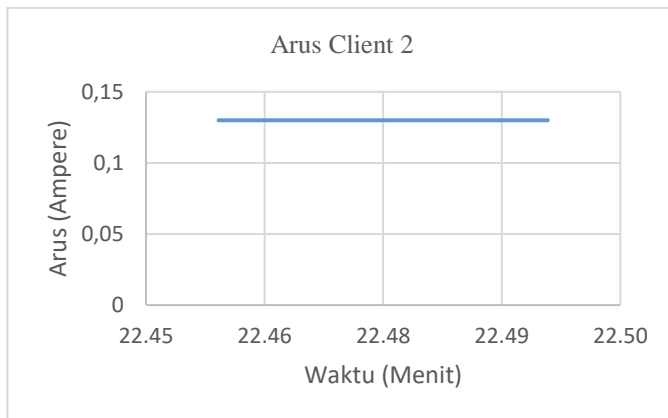
$$P = V \times I = 220V \times 0,13 A$$

$$P = 28,6 \text{ Watt}$$

Jadi PJU tersebut akan mengonsumsi daya listrik sebesar 28,6 Watt.

Tabel 4.9 Data Arus Client 2

Arus Client 2	
Waktu (Menit)	Arus (Ampere)
22:46	0,13
22:47	0,13
22:48	0,13
22:49	0,13
22:50	0,13



Gambar 4.13 Grafik Arus Client 2

Arus Base

$$I = \frac{23,3}{220} = 0,10 \text{ Ampere}$$

Jadi PJU pada Client 2 akan menghantar Arus listrik sebesar 0,10 Ampere.

Tabel 4.10 Hasil Data perbandingan Thingspeak dan Perhitungan menggunakan rumus

No	Data thingspeak		Perhitungan		Alat	
	Daya (Watt)	Arus (Ampere)	Daya (Watt)	Arus (Ampere)	Watt Meter	Ampere Meter
1	20,9	0,114	25,0	0,095	25	0,1
2	21,3	0,117	25,7	0,096	25,7	0,1
3	22,7	0,12	26,4	0,10	26,1	0,1
4	23,2	0,13	28,6	0,10	27,1	0,1
5	23,3	0,13	28,6	0,10	27,8	0,1
6	23,4	0,13	28,6	0,10	27,9	0,1

Untuk nilai nilai sensor menggunakan PZEM karena sensor PZEM-004t adalah sensor digital dengan nilai yang sudah instan tersedia dari modul sensor, sedangkan untuk nilai dari alat ukur adalah nilai langsung yang tertera pada alat tersebut.

## 4.4 Pengujian intensitas cahaya

### 4.4.1 Tujuan

Untuk mengetahui berapa nilai lux lampu yang digunakan terhadap jarak media sorot

### 4.4.2 Proses Pengambilan Data

Lampu PJU dinyalakan dan ditempatkan pada tempat yang tetap, sedangkan lux meter akan digeser tiap 50 cm dan dilakukan pencatatan data..

### 4.4.3 Hasil dan Analisa

Data diambil pada tiap 50 cm, dari jarak 10 cm hingga 500 cm. Dengan jarak paling dekat 10cm menunjukkan nilai lux sebesar 2910, sedangkan pda jarak 500cm menunjukkan nilai lux 42. Hal tersebut menunjukkan bahkan besarnya lux meter berbanding terbalik dengan jarak. Makin jauh jarak alat ukur dengan lampu, maka makin kecil juga intensitas cahaya yang diterima suatu permukaan.

Tabel 4. 11 Data intensitas cahaya terhadap jarak permukaan sorot lampu

Jarak	Lux meter
10 cm	2910
50 cm	2450
100 cm	500
150 cm	230
200 cm	150
250 cm	130
300 cm	95
350 cm	70
400 cm	54
450 cm	48
500 cm	42



Gambar 4.14 Data grafis perbandingan intensitas terhadap jarak



Gambar 4.15 pengambilan data intensitas cahaya pada jarak 50cm

## 4.5 Menghitung umur lampu dalam satuan

### 4.5.1 Tujuan

Untuk mengetahui umur lampu pju dalam satuan “Tahun”

### 4.5.2 Proses pengambilan Data

Data diambil dengan mengubah kesatuan tahun dari umur lampu pju.

### 4.5.3 Hasil Dan Analisa

Umur lampu = 20.000 Jam

Asumsi, digunakan setiap hari selama 12 Jam

Maka umur lampunya :

$$= \frac{20.000}{12} = 1.666 \text{ Hari} = \frac{1.666}{365} = 4,5 \text{ Tahun}$$

Jadi umur lampu yang masa pakainya selama 20.000 jam dan digunakan rata-rata 12 jam perhari adalah 4,5 tahun.

## 4.6 Foto Alat



HALAMAN INI SENGAJA DIKOSONGKAN



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah melakukan percobaan dan penelitian terhadap alat yang dibuat maka menghasilkan kesimpulan sebagai berikut:

- a. Aplikasi IoT pada sistem monitoring menggunakan jaringan internet untuk melakukan pengiriman data ke server dengan melalui nomor API.
- b. PZEM004T cukup akurat dan praktis digunakan sebagai sensor parameter listrik.
- c. Jangkauan jarak penerimaan data dari modul RF tergantung kondisi lingkungan, missal kondisi pepohonan, kondisi ketinggian dan interferensi dari kendaraan yang lewat.
- d. System topologi jaringan linier cocok digunakan untuk system pengiriman data antar PJU, tetapi memiliki kelemahan apabila perantara salah satu client mengalami kerusakan maka data tidak akan terkirim. Untuk penelitian berikutnya dapat ditambahkan system proteksi pengiriman apabila terjadi suatu kerusakan pada salah satu client.

#### **5.2 Saran**

Berikut ini adalah saran-saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini :

Untuk pengembangan selanjutnya diharapkan metode pengiriman Data lebih cepat dan terdapat indikasi saat pengiriman data/terima data.

HALAMAN INI SENGAJA DIKOSONGKAN

## DAFTAR PUSTAKA

- [1] M. Syaifudin, F. Rofii, and A. Qustoniah, “Rancang Bangun Sistem Monitoring Tempat Sampah Rumah Tangga Dan Penerangan Jalan Berbasis Wireless Sensor Network (Wsn),” *Transmisi*, vol. 20, no. 4, p. 158, 2019
- [2] R. L. Nurbed, R. Munadi, and R. Mayasari, “Prototype Smart Street Lighting Di Wireless Sensor Network ( Prototype Smart Street Lighting in Wireless Sensor Network ),” pp. 1–9.
- [3] S. R. Hikmawan and E. A. Suprayitno, “Rancang Bangun Lampu Penerangan Jalan Umum (PJU) Menggunakan Solar Panel Berbasis Android (Aplikasi di Jalan Parkiran Kampus 2 Umsida),” *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 3, no. 1, pp. 9–17, Jul. 2018.
- [4] A. Romadlan, “Rancang Bangun Sistem Monitoring Lampu Jalan berbasis Internet of Things (IoT) menggunakan Arduino Mega 2560,” Universitas Mercu Buana, 2019.
- [5] I. G. A. Putra, A. A. N. Amrita, and I. M. A. Suyadnya, “Rancang Bangun Alat Monitoring Kerusakan Lampu Penerangan Jalan Umum Berbasis Mikrokontroler dengan Notifikasi SMS,” *J. Comput. Sci. Informatics Eng.*, vol. 2, no. 2, pp. 90–99, Dec. 2018.
- [6] E. Ihsanto and M. Dawud, “Sistem Monitoring Lampu Penerangan Jalan Umum Menggunakan Mikrokontroler Arduino dan Sensor LDR dengan Notifikasi SMS,” *J. Teknol. Elektro*, vol. 7, no. 2, May 2016.
- [7] Nasution, H. (2021, Oktober). Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan. *Jurnal ELKHA*, 4, 4. Retrieved Juni 24, 2022
- [8] R. Ferza and M. A. Pranasari, “Inovasi Kebijakan Pengelolaan Penerangan Jalan Umum (PJU) di Kabupaten Sidoarjo: Isu dan Tantangan,” *Matra Pembaruan*, vol. 4, no. 1, pp. 1–11, May 2020.

- [9] Norazizi and Adam (2019), “Sistem Monitoring Lampu Penerangan Jalan Umum berbasis SMS,” *J. Ilm. Flash*, vol. 5, no. 1, pp. 23–28.
- [10] Adani, M. R. (2020, November 23). *Mengenal Apa Itu Internet of Thing dan Contoh Penerapannya*. Retrieved Juni 24, 2022, from Sekawan Media:
- [11] Maulana, K. Y. (2021, Desember 9). *Thingspeak.com, Web Server IoT Gratis Buat Kamu Yang Suka Gratisan*. Retrieved Juni 24, 2024, from ANAK TEKNIK INDONESIA:
- [12] D. P. Buwana, S. Setiawidayat, and M. Mukhsin (2018), “Sistem Pengendalian Lampu Penerangan Jalan Umum (PJU) Melalui Jaringan Internet Berbasis Android,” *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 3, no. 3, pp. 287–292.

## LAMPIRAN

```
#include <WiFi.h>

#include "ThingSpeak.h" // Library Thingspeak

#include "PZEM004Tv30.h" // Library PZEM004Tv30

#include <stdio.h>

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

/*****

* ESP32 initialization

* -----

* The ESP32 HW Serial interface can be routed to any GPIO pin

* Here we initialize the PZEM on Serial2 with RX/TX pins 16 and 17

*/

#if defined(ESP32)

  #if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)

    #define PZEM_RX_PIN 16//3//16

    #define PZEM_TX_PIN 17//1//17

  #endif

  #define PZEM_SERIAL Serial2

  #define CONSOLE_SERIAL Serial
```

```

    PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN,
PZEM_TX_PIN);

#endif

#define BS

//#define CL

//#define ID 2

#define Detik 1000

#define TOMenit 60

#define MaxClient 2

#define TOWait 3

uint8_t detik = 0, Client = 1, Wait = 0;

bool WaitDataClient, NextDataClient, GetDataClient, TimeToSend,
Kirim = false;

#define msglength 35

bool started = false, ended = false;

char incomingByte ;

char Inp[msglength];

char datasend[msglength];

byte bitPosition;

char ssid[] = "Pohon Belimbing";

```

```

char password[] = "klentang25";
WiFiClient client;
unsigned long myChannelNumber = 2033395;
char WriteKey[] = "NDH8XH7J2I65IOWU";
unsigned long prevmillis = 0;
//float voltage, current, power;
//float current1, voltage1, power1;
//float current2, voltage2, power2;
float power, KWh;
float power1, KWh1;
float power2, KWh2;
uint8_t WattCount = 0;
float kwh;

// Fungsi untuk membaca tegangan dari PZEM004Tv30
float cekvolt() {
    float voltage = pzem.voltage();
    if (voltage >= 0.0) {
    }
    else {
        voltage = 0.0;
    }
    return voltage;
}

```

```
}
```

```
// Fungsi untuk membaca arus dari PZEM004Tv30
```

```
float cekcurrent() {  
    float current = pzem.current();  
    if (current >= 0.0) {  
    }  
    else {  
        current = 0.0;  
    }  
    return current;  
}
```

```
// Fungsi untuk membaca daya dari PZEM004Tv30
```

```
float cekpower() {  
    float power = pzem.power();  
    if (power >= 0.0) {  
    }  
    else {  
        power = 0.0;  
    }  
    return power;  
}
```



```

// Fungsi untuk membaca kWh dari PZEM004Tv30
float cekenergy() {
    float power = pzem.energy();
    if (power >= 0.0) {
    }
    else {
        power = 0.0;
    }
    return power;
}

// Format Data dari Client = #CL|ID|Tegangan|Arus|Daya*
// Format Data dari BS = #REQ|ID*
void CekData() {
//  CONSOLE_SERIAL.printf("In %s\r\n", Input);
    char* dt[10] = {"0"};
    memset(dt, 0, sizeof(dt));
    char* ptr = strtok(Inp, "#|*");
    byte i = 0;
    while(ptr) {
        dt[i] = ptr;
        ptr = strtok(NULL, "#|*");
    }
}

```

```

    i++;
}
delay(100);
// for(uint8_t x = 0; x <= i; x++) {
//  CONSOLE_SERIAL.printf("In %d %s\r\n",x, dt[x]);
//  CONSOLE_SERIAL.printf("In 1 %s\r\n", dt[0]);
//  CONSOLE_SERIAL.printf("In 2 %s\r\n",dt[1]);
// }

#ifdef BS
if (memcmp(dt[0], "CL", strlen("CL")) == 0) {
    uint16_t CekID= strtoul(dt[1], NULL, 10);

    if (CekID == 1) {
//      voltage1 = strtoul(dt[2], NULL, 10); // (float)atof(dt[1]);
//      current1 = strtoul(dt[3], NULL, 10); // (float)atof(dt[2]);
        power1 = atof(dt[2]); // (float)atof(dt[3]);
        KWh1 = atof(dt[3]); // (float)atof(dt[3]);
//      power1 = *(uint8_t*)(dt[2]);
//      KWh1 = *(uint8_t*)(dt[3]);
        CONSOLE_SERIAL.printf("P1: %s, %0.2f[kWh: %s, %0.2f\r\n",
dt[2], power1, dt[3], KWh1);
    }

    if (CekID == 2) {

```

```

//     voltage2 = strtoul(dt[2], NULL, 10); // (float)atof(dt[1]);
//     current2 = strtoul(dt[3], NULL, 10); // (float)atof(dt[2]);
    power2 = atof(dt[2]); // (float)atof(dt[3]);
    KWh2 = atof(dt[3]); // (float)atof(dt[3]);
//     power2 = *(uint8_t*)(dt[2]);
//     KWh2 = *(uint8_t*)(dt[3]);
    CONSOLE_SERIAL.printf("P2: %s, %0.2f[kWh: %s, %0.2f\r\n",
dt[2], power2, dt[3], KWh2);
    }
    WaitDataClient = false;
    NextDataClient = true;
// }
}
#endif

#ifdef CL
    if (memcmp(dt[0], "REQ", strlen("REQ")) == 0) {
        uint16_t CekID = strtoul(dt[1], NULL, 10);
//     uint8_t CekID = *(uint8_t*)(dt[1]);
        if (CekID == ID) {
            sprintf(datasend, "#CL|%d|%0.2f|%0.2f*\r\n", CekID, power, KWh);
            CONSOLE_SERIAL.print(datasend);
        }
    }
}

```

```

#endif

    started = false;

    ended = false;

}

void setup() {

    CONSOLE_SERIAL.begin(9600);

    // pzem.begin(9600, SERIAL_8N1, RXD, TXD);

    // PZEM_SERIAL.begin(9600,SERIAL_8N1,PZEM_RX_PIN,
    PZEM_TX_PIN);

    delay(2000);

#ifdef BS

    WiFi.mode(WIFI_STA);

    uint8_t x = 0;

    do {

        if(WifiStatus()) {

            break;

        }

        else {

            x++;

            delay(1000);

        }

    }

    while(x < 5);

```

```

    ThingSpeak.begin(client);

    delay(1000);

    CONSOLE_SERIAL.printf("BASE\r\n");

    delay(1000);

#endif

#ifdef CL

    delay(1000);

    CONSOLE_SERIAL.printf("Client %d\r\n", ID);

    delay(1000);

#endif

}

uint8_t waiting = 0;

void loop() {

    if (CONSOLE_SERIAL.available() > 0) {

        incomingByte = CONSOLE_SERIAL.read();

        if(incomingByte == '#') {

            memset(Inp, 0 , sizeof(Inp));

            started = true;

            bitPosition = 0;

            Inp[bitPosition] = '\0';

        }

        else if ((incomingByte == '*'))

```

```

    { ended = true; }
else {
    if (bitPosition < msglength) {
        Inp[bitPosition] = incomingByte;
        bitPosition++;
        Inp[bitPosition] = '\0';
    }
}

#ifdef BS
    if(started && ended) {
//  CONSOLE_SERIAL.print("Inp: ");
CONSOLE_SERIAL.println(Inp);

        if(WaitDataClient) {
            CekData();
        }
    }

    if(Kirim) {
        SendToThinspeak();
        Kirim = false;
    }
#endif
72

```

```

#ifdef CL

    if(started && ended) {

//  CONSOLE_SERIAL.print("Inp: ");
CONSOLE_SERIAL.println(Inp);

        CekData();

    }

#endif

    if(millis() - prevmillis >= Detik) {

        prevmillis = millis();

        detik++;

        if(!started && !ended) {

//  voltage = cekvolt();

//  current = cekcurrent();

        power = cekpower();

        KWh = cekenergy();

        }

//  CONSOLE_SERIAL.printf("P: %0.2f|kWh: %0.2f\r\n", power,
KWh);

        if(detik > TOMenit) {

            detik = 0; GetDataClient = true;

            kwh += power * ((TOMenit / 1000.0) / 60);

//  KWh = kwh / WattCount;

        }

```

```

#ifdef BS

    if(GetDataClient) {

        waiting++;

        if(NextDataClient || Wait == TOWait) { Client++; Wait = 0;
NextDataClient = false; }

        if(Client <= MaxClient && Wait < TOWait && waiting > 3) {
CONSOLE_SERIAL.printf("#REQ|%d*\r\n", Client); Wait++; waiting
= 0; }

        if(Client > MaxClient) {

            Client = 1;

            Wait = 0;

            GetDataClient = false;

            waiting = 0;

            CONSOLE_SERIAL.printf("P: %0.2f|kWh: %0.2f\r\n", power,
KWh);

            TimeToSend = true;

        }

        WaitDataClient = true;

    }

    if(TimeToSend) {

        if(WifiStatus()) {

            TimeToSend = false;

            Kirim = true;

```



```

    }
    else {
        TimeToSend = false;
    }
}
#endif

#ifdef CL
    if(GetDataClient) {
//    voltage = cekvolt();
//    current = cekcurrent();
//    power = cekpower();
        GetDataClient = false;
    }

#endif

}

}

void SendToThinspeak() {
    ThingSpeak.setField(1, power);
    ThingSpeak.setField(2, KWh);
    ThingSpeak.setField(3, power1);
    ThingSpeak.setField(4, KWh1);
}

```

```

ThingSpeak.setField(5, power2);
ThingSpeak.setField(6, KWh2);
// ThingSpeak.setField(7, voltage2);
// ThingSpeak.setField(8, current2);
// ThingSpeak.setField(9, power2);

int x = ThingSpeak.writeFields(myChannelNumber, WriteKey);
// if(x == 200) { Serial.println("Channel update successful."); }
// else { Serial.println("Problem updating channel. HTTP error code " +
String(x)); }
}

bool WifiStatus() {
/* // Jika harus dipastikan sampai terkoneksi dulu
if(WiFi.status() != WL_CONNECTED){
Serial.print("Attempting to connect to SSID: ");
Serial.println(SECRET_SSID);
while(WiFi.status() != WL_CONNECTED){
WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change
this line if using open or WEP network
Serial.print("."); //delay(1000);
}
Serial.println("\nConnected.");
}
}

```

```
*/  
/* // Jika hanya cek saja sekali, lalu kembali ke loop  
*/  
WiFi.begin(ssid, password);  
if(WiFi.status() != WL_CONNECTED) {  
// Serial.println("Connecting to WiFi...");  
return false;  
}  
else {  
// Serial.println("Connected to WiFi");  
return true;  
}  
}
```

HALAMAN INI SENGAJA DIKOSONGKAN

# SURAT TUGAS



YAYASAN BRATA BHAKTI DAERAH JAWA TIMUR  
**UNIVERSITAS BHAYANGKARA SURABAYA**  
**FAKULTAS TEKNIK**

Kampus : Jl. A. Yani 114 Surabaya Telp. 031 - 8285602, Fax. 031 - 8285601

## SURAT TUGAS

Nomor : TUG / AA. /FTK/10/2022

- Pertimbangan :
- Bahwa dalam rangka kelancaran pelaksanaan bimbingan tugas akhir mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya.
  - Bahwa sehubungan dengan hal tersebut diatas maka dipandang perlu untuk mengeluarkan Surat Tugas ini.
- Dasar :
- Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 17 Tahun 2013 tentang Jabatan Fungsional dan Angka Kreditnya, sebagaimana telah diubah dengan Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Nomor 46 Tahun 2013.
  - Buku Pedoman Akademik Universitas Bhayangkara Surabaya Tahun Akademik 2021 / 2022.
  - Kurikulum Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya.

## DITUGASKAN

- Kepada Untuk :
- Dr. Bambang Purwahyudi, ST., MT. (Pembimbing tunggal)
- Membimbing tugas akhir mahasiswa Program Studi Teknik Elektro Fakultas Teknik Universitas Bhayangkara Surabaya atas nama MUHAMAD NANDA AL HUDA (1814121009) berjudul *PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGGUNAKAN THINGSPEAK*.
  - Melaksanakan tugas ini dengan seksama dan penuh rasa tanggung jawab.
  - Tugas ini berlaku sejak tanggal ditetapkan sampai selesainya kegiatan.
- Selesai.

Dikeluarkan Di : Surabaya

Pada Tanggal : 18 Oktober 2022

Dekan,



Dr. Mohammad Ghazi, S.T., M.T.

HALAMAN INI SENGAJA DIKOSONGKAN

## PROFIL PENULIS



NIM : 1814121009  
NAMA : Muhamad Nanda Al Huda  
Tempat, Tanggal Lahir : Kediri, 07 September 1999  
Program Studi : Teknik Elektro ( Sistem Tenaga)  
Alamat Rumah : Jl. Mangun Sari Bohar Timur Rt10  
Pekerjaan :  
Alamat Kantor :  
Nomor Ponsel : 0895619778112  
Email : [nandaalhuda799@gmail.com](mailto:nandaalhuda799@gmail.com)

Judul Tugas Akhir : PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPJU) MENGGUNAKAN THINGSPEAK

Pembimbing : Dr. Bambang Purwahyudi, ST., MT  
Tanggal Daftar : 27 Juni 2023  
Tanggal Ujian : 06 Juli 2023

**Surabaya, 06 Juli 2023**  
**Penulis,**

**(Muhamad Nanda Al Huda)**

HALAMAN INI SENGAJA DIKOSONGKAN



# FORM BIMBINGAN TUGAS AKHIR



YAYASAN BRATA BHAKTI DAERAH JAWA TIMUR  
UNIVERSITAS BHAYANGKARA SURABAYA  
FAKULTAS TEKNIK

## PROGRAM STUDI TEKNIK ELEKTRO

Kampus: Jl. Ahmad Yani 114 Surabaya Telp. 031 - 8285602 , Fax. 031 - 8285601

### FORM BIMBINGAN TUGAS AKHIR

Nama : MUHAMAD NANDA AL HUDA

NIM : 1814121009

Judul Tugas Akhir : PENERAPAN WIRELESS SENSOR NETWORK (WSN) UNTUK  
MONITORING DAYA LAMPU PENERANGAN JALAN UMUM (LPIJU)  
MENGUNAKAN THINGSPEAK

Pembimbing 1 : Dr. Bambang Purwahyudi, ST., MT.

NIDN : 0025057001

Pembimbing 2 :

NIDN :

NO	TANGGAL	CATATAN REVISI	ACC
1	10-01-2023	BAB I (Latar Belakang, Rumusan Masalah, Tujuan Masalah, Serta Penulisan yang benar).	<i>Mu</i>
2	25-01-2023	BAB II (Perbaiki Penulisan ketidapan gambar)	<i>Mu</i>
3	28-02-2023	BAB III (Perbaiki Blok Diagram dan Flowchart)	<i>Mu</i>
4	14-03-2023	BAB III dan BAB IV (Flowchart dan Perbaiki Penulisan Data)	<i>Mu</i>
5	08-05-2023	BAB IV (Hasil Data)	<i>Mu</i>
6	22-05-2023	BAB V (Perbaikan Penulisan, tambah Daftar Pustaka)	<i>Mu</i>
7	20-06-2023	BAB V (Daftar Pustaka acc)	<i>Mu</i>
8	27-06-2023	Perbaiki Daftar ISI	<i>Mu</i>

Surabaya, 17 Oktober 2022

Disetujui pada tanggal :

...27...10...2023

untuk mengikuti ujian: **Proposal / Tugas Akhir\***

Dosen pembimbing 1,

Dr. Bambang Purwahyudi, ST., MT.

Disetujui pada tanggal :

.....

untuk mengikuti ujian: **Proposal /  
Tugas Akhir\***

Dosen pembimbing 2,



