



## KERJA PRAKTIK - EF234603

### Implementasi dan Pengembangan Sistem Monitoring Listrik di Departemen Teknik Informatika

PQCX+424, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur  
60117

Periode: 18 Juli 2023 - 25 Desember 2023

Oleh:

Zunia Aswaroh 5025201058  
Ida Bagus Kade Rainata P W 5025201235

**Pembimbing Jurusan**

Ary Mazharuddin S.Kom., M.Comp.Sc.

**Pembimbing Lapangan**

Ridho Rahman Hariadi, S.Kom., M.Sc..

**DEPARTEMEN TEKNIK INFORMATIKA**

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2023



KERJA PRAKTIK - EF234603

## **Implementasi dan Pengembangan Sistem Monitoring Listrik di Departemen Teknik Informatika**

PQCX+424, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur  
60117

Periode: 18 Juli 2023 - 25 Desember 2023

Oleh:

Zunia Aswaroh 5025201058

Ida Bagus Kade Rainata P W 5025201235

## Pembimbing Jurusan

Ary Mazharuddin S.Kom., M.Comp.Sc.

## Pembimbing Lapangan

Ridho Rahman Hariadi, S.Kom., M.Sc..

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2023

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR ISI**

DAFTAR ISI.....	4
DAFTAR GAMBAR.....	8
DAFTAR TABEL.....	10
LEMBAR PENGESAHAN.....	12
KATA PENGANTAR.....	16
BAB I	
PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	1
1.3. Manfaat.....	2
1.4. Rumusan Masalah.....	2
1.5. Lokasi dan Waktu Kerja Praktek.....	3
1.6. Metodologi Kerja Praktek.....	3
1.6.1. Perumusan Masalah.....	3
1.6.2. Studi Literatur.....	3
1.6.3. Analisis dan Perancangan Sistem.....	4
1.6.4. Implementasi Sistem.....	4
1.6.5. Pengujian dan Evaluasi.....	4
1.6.6. Kesimpulan dan Saran.....	4
1.7. Sistematika Laporan.....	4
1.7.1. Bab I Pendahuluan.....	4
1.7.2. Bab II Profil Perusahaan.....	5
1.7.3. Bab III Tinjauan Pustaka.....	5
1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem.....	5
1.7.5. Bab V Implementasi Sistem.....	5
1.7.6. Bab VI Pengujian dan Evaluasi.....	5

1.7.7. Bab VII Kesimpulan dan Saran.....	5
<b>BAB II</b>	
PROFIL INSTITUSI.....	7
2.1. Profil Departemen Teknik Informatika.....	7
2.2. Lokasi.....	7
<b>BAB III</b>	
TINJAUAN PUSTAKA.....	9
3.1. Tegangan Listrik.....	9
3.2. Arus Listrik.....	9
3.3. Daya Listrik.....	10
3.4. NodeMCU ESP8266.....	12
3.5. MQTT (Message Queuing Telemetry Transport).....	13
<b>BAB IV</b>	
ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM.....	16
4.1. Analisis Sistem.....	16
4.1.1. Definisi Umum Aplikasi.....	16
4.1.2. Analisis Kebutuhan.....	16
4.2. Perancangan Infrastruktur Sistem.....	19
4.2.1. Desain Sistem.....	19
4.2.2. Bahan dan Peralatan yang Digunakan.....	21
<b>BAB V</b>	
IMPLEMENTASI SISTEM.....	26
5.1. Implementasi Sensor PZEM-004T.....	26
5.2. Implementasi Kode Arduiono pada NodeMCU ESP8266. 27	
5.2.1. Konfigurasi Library yang Diperlukan.....	28
5.2.2. Penulisan Kode Set Up Koneksi Ke Wifi dan Koneksi ke MQTT Server.....	30
5.2.3. Penulisan Kode Implementasi Pembacaan Data	

pada PZEM-004T .....	34
5.3. Implementasi Homeassistant.....	37
BAB VI	
PENGUJIAN DAN EVALUASI.....	45
6.1. Tujuan Pengujian.....	45
6.2. Kriteria Pengujian.....	45
6.3. Skenario Pengujian.....	45
6.4. Evaluasi Pengujian.....	46
BAB VII	
KESIMPULAN DAN SARAN.....	48
7.1. Kesimpulan.....	48
7.2. Saran.....	48
DAFTAR PUSTAKA.....	50
BIODATA PENULIS I.....	52
BIODATA PENULIS II.....	52

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR GAMBAR**

Gambar 3.1 MQTT (Message Queuing Telemetry Transport)....	13
Gambar 4.1 Desain Arsitektur Sistem.....	19
Gambar 4.2 Desain Wiring Sensor dengan Microcontroller.....	19
Gambar 5.1 Kabel Jumper pada Sensor PZEM-004T.....	26
Gambar 5.2 Kabel Jumper pada ESP8266.....	27
Gambar 5.3 Hasil Akhir Sensor.....	27
Gambar 5.4 Tampilan Preferences.....	28
Gambar 5.5 Library ESP8266 Terinstall.....	29
Gambar 5.6 Library PZEM004Tv30 Terinstall.....	29
Gambar 5.7 IP yang Dihasilkan dari Mosquitto MQTT Server...	30
Gambar 5.8 Menu Settings Homeassitant.....	40
Gambar 5.9 Opsi Integrasi MQTT pada Homeassistant.....	41
Gambar 5.10 Form Konfigurasi MQTT Broker pada Homeassistant.....	41
Gambar 5.11 Konfigurasi Entitas Current yang Dikelompokkan ke Area Panel Gedung A.....	42
Gambar 5.12 List Entitas yang sudah Terkonfigurasi.....	42
Gambar 5.13 Tampilan Dashboard pada Homeassitant.....	43
Gambar 5.14 Tampilan Grafis pada Homeassitant.....	43

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR TABEL**

Tabel 4.1 Alat dan Bahan yang Digunakan dalam Kerja Praktik.....	22
Tabel 6.1 Hasil Evaluasi Pengujian.....	46

## **DAFTAR KODE SUMBER**

Kode Sumber 5.1 Variabel pada Setup Koneksi Wifi dan MQTT Server.....	32
Kode Sumber 5.2 Implementasi Fungsi Setup Wifi dan Reconnect.....	33
Kode Sumber 5.3 Implementasi Pembacaan Data dari Sensor...	37
Kode Sumber 5.4 File configuration.yaml.....	39

*[Halaman ini sengaja dikosongkan]*

## **LEMBAR PENGESAHAN KERJA PRAKTIK**

### **Implementasi dan Pengembangan Sistem Monitoring Listrik di Departemen Teknik Informatika**

Oleh:

Zunia Aswaroh	5025201058
Ida Bagus Kade Rainata P W	5025201235

Disetujui oleh Pembimbing Kerja Praktik:

1. Ary Mazharuddin S.Kom.,  
M.Comp.Sc.  
NIP. 198106202005011003

(Pembimbing Departemen)

2. Ridho Rahman Hariadi,  
S.Kom., M.Sc.  
NIP. 198702132014041001

(Pembimbing Lapangan)

*[Halaman ini sengaja dikosongkan]*

## **Implementasi dan Pengembangan Sistem Monitoring Listrik di Departemen Teknik Informatika**

Nama Mahasiswa	:	Zunia Aswaroh
NRP	:	5025201058
Nama Mahasiswa	:	Ida Bagus Kade Rainata P W
NRP	:	5025201235
Departemen	:	Teknik Informatika FTEIC-ITS
Pembimbing Departemen	:	Ary Mazharuddin S.Kom., M.Comp.Sc.
Pembimbing Lapangan	:	Ridho Rahman Hariadi, S.Kom., M.Sc.

### **ABSTRAK**

*PT. PLN (persero) adalah perusahaan milik negara yang bergerak di bidang ketenagalistrikan baik dari mulai mengoperasikan pembangkit energi listrik yaitu generator sampai ke konsumen dengan melakukan transmisi ke seluruh wilayah indonesia, konsumen PT.PLN Persero selama ini mendapatkan layanan program listrik prabayar dan pascabayar, namun pada program listrik pascabayar untuk 1 fase PLN harus turun ke rumah pelanggan untuk memonitor pemakaian energi listrik.*

*Pada kerja praktek ini, dirancang sebuah alat monitoring energi listrik terpakai dengan menggunakan MQTT (Message Queueing Telemetry Transport) berbasis mikrokontroler NodeMCU ESP8266MOD yang memiliki kapasitas data yang lebih besar dan pin input yang cukup banyak sehingga berfungsi untuk penambahan sensor sebagaimana diperlukan untuk mencatat dan membaca pemakaian energi listrik secara realtime dan berkala.*

**Kata Kunci : Monitoring Listrik, NodeMCU, IOT**

*[Halaman ini sengaja dikosongkan]*

## **KATA PENGANTAR**

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Implementasi dan Pengembangan Sistem Monitoring Listrik di Departemen Teknik Informatika.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Ary Mazharuddin S.Kom., M.Comp.Sc. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Bapak Ridho Rahman Hariadi, S.Kom., M.Sc. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 23 November 2023  
Zunia Aswaroh dan Ida Bagus Kade Rainata P W

*[Halaman ini sengaja dikosongkan]*

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Sistem monitoring listrik memiliki peranan yang sangat krusial dalam memastikan keberlanjutan operasional dan efisiensi energi di berbagai lingkungan, termasuk di Departemen Teknik Informatika. Seiring dengan perkembangan teknologi, implementasi dan pengembangan sistem monitoring listrik menjadi semakin penting untuk meningkatkan pemahaman, pengendalian, dan pengelolaan konsumsi energi. Departemen Teknik Informatika sebagai entitas yang berkaitan erat dengan teknologi dan informasi, harus memahami dan menerapkan sistem monitoring listrik sebagai langkah proaktif dalam mendukung keberlanjutan, efisiensi, dan tanggung jawab lingkungan.

Peningkatan kebutuhan akan energi listrik dan kepedulian terhadap dampak lingkungan telah mendorong organisasi, termasuk Departemen Teknik Informatika, untuk mencari solusi yang lebih efisien dan berkelanjutan dalam mengelola konsumsi energi. Dengan pelaksanaan sistem monitoring listrik yang canggih, departemen ini dapat secara real-time memantau, menganalisis, dan mengelola penggunaan energi listriknya. Hal ini tidak hanya memberikan keuntungan ekonomis melalui efisiensi energi, tetapi juga memberikan kontribusi positif terhadap upaya mitigasi dampak lingkungan.

#### **1.2. Tujuan**

Tujuan kerja praktek ini adalah menyelesaikan kewajiban nilai kerja praktek sebesar 2 sks dan membantu

Departemen Teknik Informatika untuk memonitoring kebutuhan listrik di gedung Teknik Informatika.

### **1.3. Manfaat**

Manfaat yang diperoleh dengan adanya sistem monitoring listrik di departemen Teknik informatika adalah sebagai berikut:

1. Implementasikan untuk mengurangi konsumsi listrik secara keseluruhan.
2. Dapat membantu dalam memantau tren konsumsi energi dari waktu ke waktu. Dengan data yang secara real-time ini, departemen dapat memprediksi kebutuhan energi masa depan dan merencanakan kapasitas sumber daya yang optimal.
3. Dapat mengidentifikasi area atau perangkat yang menggunakan listrik secara berlebihan.

### **1.4. Rumusan Masalah**

Rumusan masalah dari kerja praktek ini adalah sebagai berikut:

1. Bagaimana implementasi sistem monitoring listrik di Departemen Teknik Informatika dapat meningkatkan efisiensi penggunaan energi?

2. Bagaimana mengimplementasikan pemantauan konsumsi energi secara real-time untuk mendapatkan data yang akurat dan terkini?

## **1.5. Lokasi dan Waktu Kerja Praktek**

Pengerjaan Kerja Praktek ini dilakukan di departemen Teknik Informatika.

Adapun kerja praktek dimulai pada tanggal 18 Juli 2023 hingga 25 Desember 2023.

## **1.6. Metodologi Kerja Praktek**

Metodologi dalam pembuatan buku kerja praktek meliputi:

### **1.6.1. Perumusan Masalah**

Untuk mengetahui kebutuhan dalam pengerjaan kerja praktik tersebut. Kami berdiskusi bersama pak ary dan pak ridho mengenai konsep dan proses untuk pengerjaannya, pada diskusi tersebut diterapkan fitur - fitur apa saja yang akan digunakan pada proses monitoring listrik tersebut. Dibahas juga bagaimana langkah-langkah untuk melakukan instalasi serta implementasi monitoring listrik di Departemen Teknik Informatika. Dan juga disiapkan bahan - bahan yang akan digunakan pada kerja praktik tersebut.

### **1.6.2. Studi Literatur**

Setelah mendapat gambaran bagaimana sistem tersebut berjalan, kami diberitahu tinjauan apa saja yang akan diimplementasikan untuk membuat monitoring listrik tersebut beroperasi. Tinjauan yang dipakai meliputi, Web Home Assistant, MQTT (*Message Queuing Telemetry Transport*), sensor listrik dan lain-lain. Selain itu, kami dijelaskan aturan-aturan

dalam menuliskan konfigurasi agar konfigurasi dapat mudah dipahami oleh pengembang yang lain.

### **1.6.3. Analisis dan Perancangan Sistem**

Setelah konsep dan proses dijelaskan, untuk merancang sistem yang baik perlu adanya sebuah desain model sensor jaringan elektronik. Dimana desain tersebut sebagai acuan berjalannya jaringan listrik.

### **1.6.4. Implementasi Sistem**

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami mengintegrasikan hasil perancangan sistem sensor yang telah dibuat dengan menyambungkan alat - alat dan dikonfigurasi lewat IDE arduino dengan memodifikasi apa saja yang dibutuhkan sesuai fitur - fitur yang diminta, sehingga hasilnya implementasi akan ditampilkan melalui homeassistant serta terdapat beberapa settingan yang harus disesuaikan agar berjalan dengan baik.

### **1.6.5. Pengujian dan Evaluasi**

Setelah proses yang dilakukan telah jadi, perlu adanya evaluasi untuk menguji apakah alat yang dirancang sesuai harapan, jika masih belum sesuai atau perlu menambah fitur, maka dilakukan perbaikan ulang terhadap alat sensor yang telah dibuat sekaligus menambah atau mengurangi pada fitur tersebut.

### **1.6.6. Kesimpulan dan Saran**

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

## **1.7. Sistematika Laporan**

### **1.7.1. Bab I Pendahuluan**

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

**1.7.2. Bab II Profil Perusahaan**

Bab ini berisi gambaran umum Dinas Pendidikan Provinsi Jawa Timur mulai dari profil, lokasi perusahaan.

**1.7.3. Bab III Tinjauan Pustaka**

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

**1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem**

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

**1.7.5. Bab V Implementasi Sistem**

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

**1.7.6. Bab VI Pengujian dan Evaluasi**

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

**1.7.7. Bab VII Kesimpulan dan Saran**

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

## **PROFIL INSTITUSI**

### **2.1. Profil Departemen Teknik Informatika**

Departemen Teknik Informatika di Institut Teknologi Sepuluh Nopember (ITS) Surabaya adalah sebuah entitas akademik yang memiliki peran strategis dalam mendidik, mengembangkan, dan memajukan keilmuan serta teknologi di bidang teknik informatika. Dan juga menjadi pusat unggulan dalam pengembangan teknologi informasi berbasis riset dan inovasi untuk mendukung kebutuhan masyarakat dan industri. Departemen Teknik Informatika ITS Surabaya terus berusaha untuk menjadi pusat keunggulan di bidangnya, menghasilkan lulusan yang siap bersaing di dunia industri dan berkontribusi pada kemajuan teknologi informasi secara global.

### **2.2. Lokasi**

Jalan Teknik Kimia - Gedung Departemen Teknik Informatika Kampus Institut Teknologi Sepuluh Nopember Surabaya Jalan Raya ITS, Sukolilo, Surabaya 60111, Indonesia

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1. Tegangan Listrik**

Tegangan yaitu beda potensial (voltage) adalah kerja yang dilakukan untuk menggerakkan satu muatan (sebesar satu coulomb) pada elemen atau komponen dari satu terminal atau kutub ke terminal atau kutub lainnya, beda potensial jika menggerakkan atau memindahkan muatan sebesar satu coulomb dari satu terminal ke terminal lainnya. Keterkaitan antara kerja yang dilakukan sebenarnya adalah energi yang dikeluarkan, sehingga pengertian dapat disederhanakan bahwa tegangan adalah energi per satuan muatan. ( Irwan Dinata, 2015)

Secara matematis:

Pada tegangan searah (DC)

$$V = \frac{I}{R} \text{ (volt)} \quad (3.1)$$

Pada tegangan bolak balik (AC)

$$V = \frac{p}{I} \cdot \cos \phi \quad (3.2)$$

#### **3.2. Arus Listrik**

Sumber listrik PLN merupakan sumber energi listrik dengan arus bolak balik atau Alternating current (AC) yang dihasilkan dari generator AC pembangkit listrik baik itu pembangkit listrik bertenaga diesel (PLTD), pembangkit listrik tenaga air (PLTA) ataupun pembangkit listrik lainnya yang menghasilkan sumber listrik dengan arus bolak balik. Sumber listrik AC menghasilkan tegangan dan arus dengan tegangan serta polaritasnya selalu berubah-ubah dari polaritas positif ke polaritas negatif atau sebaliknya secara periodik terhadap fungsi waktu, dengan bentuk gelombang berupa gelombang sinus, square, dan segitiga. Untuk sumber listrik dari PLN sendiri berupa gelombang

sinus, sementara untuk gelombang square dan segitiga biasanya banyak digunakan di inverter.

Arus listrik merupakan suatu elektron yang mengalir pada suatu penghantar yang digerakkan oleh beda potensial (tegangan) besarnya hantaran pada kawat tersebut hanya bergantung pada adanya elektron bebas, karena muatan inti dan elektron pada lintasan dalam terikat erat pada struktur kristal. Pada dasarnya di dalam kawat terdapat aliran elektron dalam jumlah yang sangat besar, jika jumlah elektron bergerak ke kanan dan ke kiri sama besar maka seolah olah tidak terjadi apa apa. Namun jika ujung sebelah kanan kawat menarik elektron sedangkan ujung sebelah kiri melepaskannya maka akan terjadi aliran elektron ke kanan. Aliran elektron yang selanjutnya disebut dengan arus listrik. Besarnya arus listrik diukur dengan satuan banyaknya elektron per detik, namun demikian ini bukan satuan yang praktis karena harganya terlalu kecil. Untuk menggerakan arus dibutuhkan tegangan atau beda potensial yang dihasilkan generator. (Muhammad Fitra, 2022).

Adapun rumus arus pada tegangan searah yaitu:

$$I = \frac{V}{R} \quad (3.3)$$

Dan rumus arus pada tegangan bolak balik yaitu:

$$I = \frac{P}{V} \cdot \cos \cos phi \quad (3.4)$$

### 3.3. Daya Listrik

Menurut sejarahnya, penggunaan konsep daya semu (apparent power) dan faktor daya (power factor) diperkenalkan oleh kalangan industri penyedia daya listrik, yang bisnisnya memindahkan energi listrik dari satu titik ke titik lain. Efisiensi proses pemindahan daya listrik ini terkait langsung dengan biaya energi listrik yang pada gilirannya menjelma menjadi biaya yang harus dibayarkan oleh

konsumen. Hal yang mempengaruhi perpindahan energi listrik tersebut adalah faktor daya. Untuk mencapai efisiensi pemindahan energi 100% maka rangkaian harus memiliki nilai faktor daya sebesar 1. Namun hal ini sulit dicapai karena adanya rugi-rugi yang ditimbulkan oleh penghantar listrik dan juga beban listrik, terutama beban induktif. Perbandingan antara daya nyata (watt) terhadap perkalian arus dan tegangan disebut faktor daya ( $pF$ ). (Muhammad Fitra, 2022)

Daya pada arus bolak-balik atau *alternating current* (ac) ada 3 macam yaitu daya aktif, daya reaktif dan daya nyata.

### 1. Daya Aktif

Daya aktif digunakan secara umum oleh konsumen. Daya aktif inilah yang biasanya dapat dikonversikan dalam bentuk kerja. Satuan daya aktif dinyatakan dalam watt. Daya aktif (*real power*), didapat dari persamaan:

$$P = V \cdot I \cdot \cos\theta [kW] \quad (3.5)$$

### 2. Daya Reaktif

Daya reaktif adalah jumlah daya yang diperlukan untuk pembentukan medan magnet. Dari pembentukan medan magnet. Maka akan terbentuk fluks magnet. Satuan daya reaktif dinyatakan dalam

Var. Daya reaktif (*reactive power*), didapat dari persamaan :

$$Q = V \cdot I \cdot \sin \phi \text{ (Var)} \quad (3.6)$$

### 3. Daya Semu

Daya semu adalah penjumlahan geometris dari daya aktif dan daya reaktif. Daya semu merupakan daya yg di produksi oleh perusahaan sumber listrik untuk di distribusikan ke konsumen. Satuan daya semu ini dinyatakan dalam VA.(R A G ramadhianti, 2018) Daya semu didapat dari persamaan berikut:

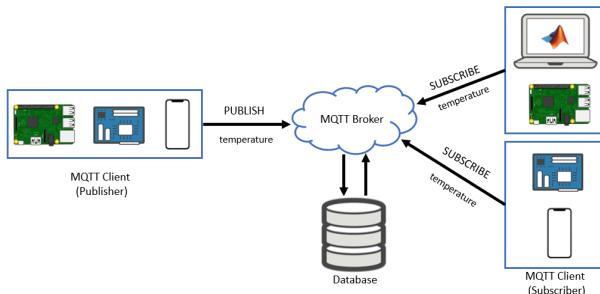
$$S = V \cdot I \text{ (VA)} \quad (3.7)$$

## 3.4. NodeMCU ESP8266

NodeMCU ESP8266 merupakan salah satu *device open source* yang digunakan pada *platform IoT* dimana memiliki ukuran board yang kecil dengan panjang 4.83 cm, lebar 2.54 cm, dan berat sekitar 7 gram. Selain ukurannya yang kecil NodeMCU ESP8266 sendiri sudah dilengkapi fitur wifi dan firmware yang tentunya juga bersifat open source dari segi biaya penggunaan NodeMCU ESP8266 juga lebih hemat dibandingkan dengan menggunakan *Arduino Uno* yang belum memiliki modul wifi dan hanya berupa mikrokontroler belum berbasis IoT sehingga perlu adanya perangkat tambahan seperti modul *wifi shield* agar bisa terhubung ke dalam sistem IoT. (Ari Kurnianto, 2022)

### 3.5. MQTT (*Message Queuing Telemetry Transport*)

*Message Queuing Telemetry Transport* (MQTT) merupakan jenis protokol jaringan pada IoT (*Internet of Things*) yang dimana berfungsi sebagai komunikasi yang bersifat *machine to machine* atau M2M dapat bekerja pada *bandwidth* yang rendah sangat cocok untuk digunakan dalam *internet of things* (IoT) yang dapat bekerja dalam energi dan media penyimpanan yang minimum serta sifatnya yang ringan dalam pengiriman pesan (*lightweight message*) pada header berukuran kecil yaitu 2 bytes. MQTT pertama kali dikembangkan oleh Andy Stanford-Clark dari IBM dan Arlen Nipper dari Arcom pada tahun 1999 dan dengan tujuan untuk meminimalisir daya dari proses serta meminimalkan penggunaan *bandwidth*. Seperti ditunjukkan pada gambar 3.1



**Gambar 3.1** MQTT (Message Queuing Telemetry Transport)  
(sumber: <https://images.app.goo.gl/bMbVYGvFwxxiLYbh7>)

Pada protokol MQTT (Message Queuing Telemetry Transport) sendiri menggunakan metode komunikasi publish/subscribe yang dimana publish berfungsi sebagai pengirim pesan dan subscribe sebagai penerima pesan menggunakan protokol MQTT sendiri dimana dalam MQTT

terdapat broker yang berfungsi sebagai topic yang dikirimkan oleh publisher untuk diteruskan ke subscribe berdasarkan permintaan dari pengguna. (Z. B. Abilovani, 2018)

*[Halaman ini sengaja dikosongkan]*

## **BAB IV**

# **ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM**

### **4.1. Analisis Sistem**

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun infrastruktur sistem sensor listrik Teknik Informatika yaitu analisis dari infrastruktur sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.

#### **4.1.1. Definisi Umum Aplikasi**

Secara umum, sistem monitoring listrik ini merupakan sistem dimana informasi kelistrikan seperti Tegangan (V), Kuat Arus (A), Daya (W), Energi (kWh), Frekuensi (Hz), dan *Power Factor* dapat monitor melalui suatu aplikasi. Aplikasi ini melibatkan sensor listrik (PZEM-004T) yang dihubungkan dengan *micro-controller* (ESP8266). Data yang didapatkan dari sensor ini kemudian dikirim ke suatu *message broker* (MQTT). Selanjutnya, aplikasi berbasis web (*Home Assistant*) mengambil data tersebut melalui *message broker* dan mengolah data tersebut untuk ditampilkan kepada *client*.

#### **4.1.2. Analisis Kebutuhan**

Dalam analisis kebutuhan ini, dijabarkan beberapa *use case* dalam sistem monitoring listrik yang dikerjakan. Analisis kebutuhan ini dibagi menjadi 2 bagian, yaitu bagian sensor listrik dan bagian aplikasi.

##### **1. Sensor Listrik**

Ada beberapa *use case* yang dibutuhkan dalam perakitan sensor listrik ini, yaitu:

### **[UC01]: Membaca Tegangan (V)**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai tegangan dalam satuan *volt* (V) dan mengirimkan data ini ke MQTT server.

### **[UC02]: Membaca Kuat Arus (A)**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai kuat arus dalam satuan *ampere* (A) dan mengirimkan data tersebut ke MQTT server.

### **[UC03]: Membaca Daya (W)**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai daya dalam satuan *watt* (W) dan mengirimkan data tersebut ke MQTT server.

### **[UC04]: Membaca Energi (kWh)**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai energi dalam satuan *kilowatt per hour* (kWh) dan mengirimkan data tersebut ke MQTT server.

### **[UC05]: Membaca Frequency (Hz)**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai frekuensi dalam satuan *hertz* (Hz) dan mengirimkan data tersebut ke MQTT server.

### **[UC06]: Membaca Power Factor**

*Use case* ini menjelaskan bahwa sensor listrik harus mampu membaca nilai *power factor* mengirimkan data tersebut ke MQTT server.

## **2. Aplikasi**

Perancangan aplikasi berbasis web ini secara garis besar hanya melibatkan proses pembacaan data dari MQTT server dan menampilkannya dalam bentuk grafik.

### **[UC07]: Membaca Tegangan (V) dari MQTT Server**

*Use case ini menjelaskan pembacaan tegangan (V) dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC08]: Membaca Kuat Arus (A) dari MQTT Server**

*Use case ini menjelaskan pembacaan kuat arus (A) dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC09]: Membaca Daya (W) dari MQTT Server**

*Use case ini menjelaskan pembacaan Daya (W) dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC10]: Membaca Energi (kWh) dari MQTT Server**

*Use case ini menjelaskan pembacaan Energi (kWh) dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC11]: Membaca Frequency (Hz) dari MQTT Server**

*Use case ini menjelaskan pembacaan Frekuensi (Hz) dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC12]: Membaca Power Factor dari MQTT Server**

*Use case ini menjelaskan pembacaan Power Factor dari MQTT server ke dalam bentuk entitas home assistant.*

### **[UC13]: Menampilkan Data Sensor ke Dalam Bentuk Grafik**

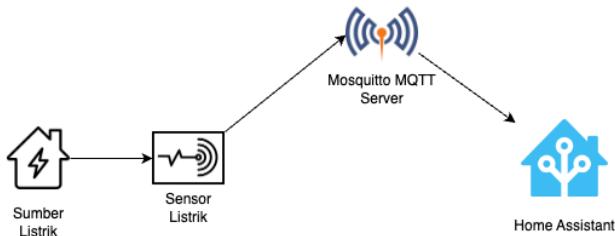
*Use case ini merupakan tahap dimana data yang diterima dari MQTT server ditampilkan dalam bentuk grafik pada Home Assistant.*

Data-data tersebut meliputi Tegangan (V), Kuat Arus (A), Daya (W), Energi (kWh), Frekuensi (Hz), dan Power Factor.

## 4.2. Perancangan Infrastruktur Sistem

### 4.2.1. Desain Sistem

Desain Sistem Monitoring Listrik Teknik Informatika ini dimulai dari proses perakitan sensor listrik, pengiriman data sensor dari *microcontroller* ke MQTT server Mosquitto, *set up Home Assistant* pada server, dan pembacaan data pada *Home Assistant*. Secara garis besar, arsitektur sistem ini dapat dilihat pada **Gambar 4.1**.

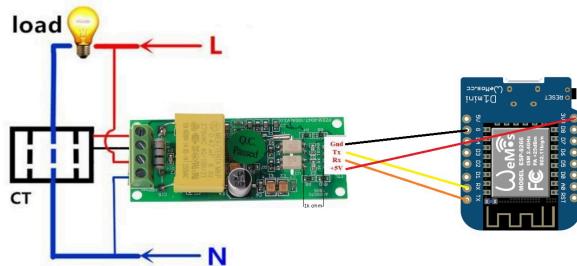


**Gambar 4.1** Desain Arsitektur Sistem

**Gambar 4.1** menunjukkan entitas-entitas apa saja yang perlu ada agar sistem ini dapat berjalan. Tentu saja, entitas yang perlu ada dalam suatu sistem monitoring listrik adalah sumber listrik. Dalam kerja praktik ini, sumber listrik terfokus pada listrik di Teknik Informatika ITS Surabaya.

Selanjutnya, entitas Sensor Listrik yang berfungsi sebagai alat untuk membaca listrik yang keluar dari entitas Sumber Listrik. Perlu diingat bahwa sensor yang digunakan disini adalah PZEM-004T yang dihubungkan dengan *microcontroller* ESP8266. Sensor listrik ini memiliki kemampuan untuk

membaca arus listrik maksimal 100A dan dengan tegangan maksimal 260V. Kemampuan sensor ini sudah disesuaikan dengan sumber listrik yang akan dibaca datanya. Hal ini penting dilakukan agar tidak terjadi kerusakan fatal pada sumber listrik yang dapat berakibat pada tidak berjalannya aktivitas kampus. Jika sensor listrik sudah sesuai, proses perakitan dapat dilakukan. Perakitan sensor listrik melibatkan proses penghubungan antara sensor PZEM-004T dengan *microcontroller* ESP8266. **Gambar 4.2** menunjukkan desain *wiring* antara sensor dengan *microcontroller*. Sesuai dengan gambar, sensor harus dihubungkan dengan pin-pin yang sesuai pada *microcontroller* melalui kabel *jumper*.



**Gambar 4.2** Desain *Wiring* Sensor dengan *Microcontroller*

Setelah pin-pin yang bersesuaian terhubung, langkah selanjutnya adalah mengonfigurasi *microcontroller* melalui Arduino IDE. Di tahap ini, proses penghubungan *microcontroller* ke *wifi*, pembacaan data dari sensor, dan pengiriman data ke *mqtt* server akan dilakukan dalam bentuk penulisan kode.

Entitas berikutnya adalah *MQTT* Server. *Broker MQTT* yang digunakan adalah Mosquitto karena

broker ini gratis dan memiliki kecepatan transfer data yang cepat. Alasan menggunakan *broker* pihak ketiga adalah karena kemudahan dalam proses *set up*. Di tahap ini, data dari sensor akan dikirim ke Mosquitto server melalui *microcontroller*. Proses pengiriman data melibatkan tahapan *subscribe* dan *publish* pada sebuah *topic*. *Subscribe* merupakan tahap dimana data akan dibaca, *publish* adalah tahap dimana data akan dikirim, dan *topic* merupakan saluran dimana suatu data akan dibaca atau dikirim (*tunnel*).

Entitas terakhir yaitu *Home Assistant*. *Home Assistant* akan melakukan *subscribe* ke *topic-topic* yang sudah dibuat pada MQTT server. *Topic-topic* yang ada akan berjumlah sesuai dengan jenis data yang dikirim. Misalnya terdapat topic yang bernama “listrik-out-informatika/power”. Ini berarti, dalam *topic* tersebut, data *power* (Daya) dari sensor dikirim. Proses *subscribe* dilakukan pada konfigurasi entitas *home assistant*. Entitas merupakan representasi setiap data yang diterima dari sensor. Proses instalasi *home assistant* dilakukan berbasis docker yang kemudian akan di-host dalam suatu server. Hal ini dilakukan dengan alasan isolasi dan agar *homeassistant* dapat diakses dari lokasi manapun.

#### 4.2.2.

#### Bahan dan Peralatan yang Digunakan

Dalam pembuatan sistem monitoring listrik ini, ada beberapa alat dan bahan yang diperlukan sesuai dengan **Tabel 4.1**. Alat dan bahan kebanyakan digunakan pada saat proses perakitan sensor listrik. Hal ini dikarenakan tahap tersebutlah yang dilakukan dilakukan secara langsung.

**Tabel 4.1** Alat dan Bahan yang Digunakan dalam Kerja Praktik

No	Bahan/Alat	Fungsi
1	PZEM-004T Sensor <i>(With Clamp)</i>	Sensor listrik. Membaca tegangan, kuat arus, daya, energi, frekuensi, dan <i>power factor</i> .
2	<i>Clamp</i> Sensor.	Untuk menangkap aliran listrik yang keluar.
3	Server (Lab Arsitektur Komputer dan Jaringan)	Untuk meng-host <i>Home Assistant</i> docker agar bisa diakses dengan browser dari tempat manapun.
4	Kabel <i>Jumper Male</i>	Untuk menghubungkan sensor

---

*Female*

dengan NodeMCU ESP8266.



- 
- 5    NodeMCU ESP8266  
(*Microcontroller*)



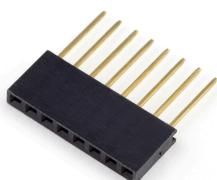
Sebagai media untuk menerima data dari sensor. Mengirimkan data ke mqtt server.

- 
- 6    Kabel Micro USB

Kabel penghubung NodeMCU ESP8266 ke Laptop.



- 
- 7    *Connector Pin*



Alat penghubung pin-pin pada NodeMCU ESP8266 ke kabel *jumper*.

---

8. Solder



Alat untuk menempel *connector pin* ke NodeMCU ESP8266.

---

9 Kabel Listrik



Untuk menghubungkan *clamp* listrik ke sensor dan menghubungkan kabel sumber listrik ke sensor.

---

10 Laptop



Untuk menulis program arduino yang akan diinstal ke dalam NodeMCU ESP8266.

---

*[Halaman ini sengaja dikosongkan]*

## BAB V

### IMPLEMENTASI SISTEM

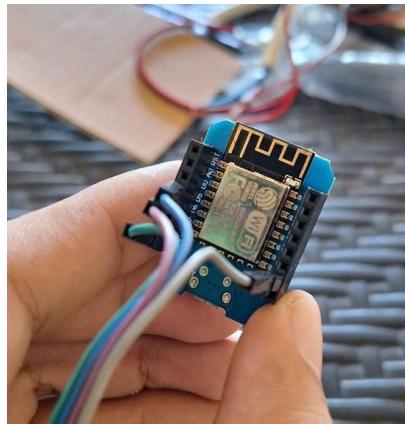
Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian implementasi sensor PZEM-004T, implementasi kode arduino pada ESP8266, dan implementasi *home assistant*.

#### 5.1. Implementasi Sensor PZEM-004T

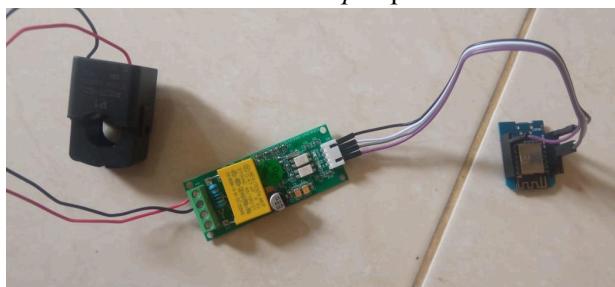
Implementasi PZEM-004T mengacu pada proses perakitan sensor. *Step* ini dimulai dari proses penempelan *connector pin* dengan solder pada NodeMCU ESP8266. Dilanjutkan dengan menghubungkan sensor ke NodeMCU ESP8266 dengan kabel *jumper* sesuai dengan **Gambar 4.2**, **Gambar 5.1** dan **Gambar 5.2** menunjukkan kabel *jumper* pada sensor dan kabel *jumper* pada ESP8266. Selanjutnya, adalah penghubungan *clamp* dengan sensor. Hal ini juga mengacu pada **Gambar 4.2**. Hasil akhir dari perakitan sensor ini ditunjukkan pada **Gambar 5.3**.



**Gambar 5.1** Kabel *Jumper* pada Sensor PZEM-004T



**Gambar 5.2** Kabel Jumper pada ESP8266



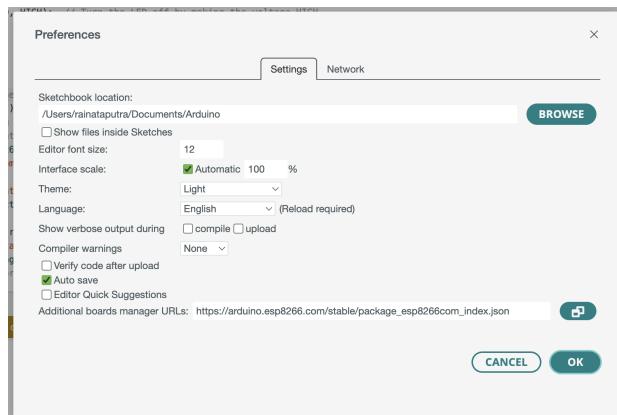
**Gambar 5.3** Hasil Akhir Sensor

## 5.2. Implementasi Kode Arduino pada NodeMCU ESP8266

Bagian ini berisi tentang implementasi kode *microcontroller* NodeMCU ESP8266, dimulai tahap pembacaan data dari sensor, *set up* koneksi *wifi*, *set up* koneksi MQTT Server, dan pengiriman data ke MQTT Server Mosquitto.

### 5.2.1. Konfigurasi *Library* yang Diperlukan

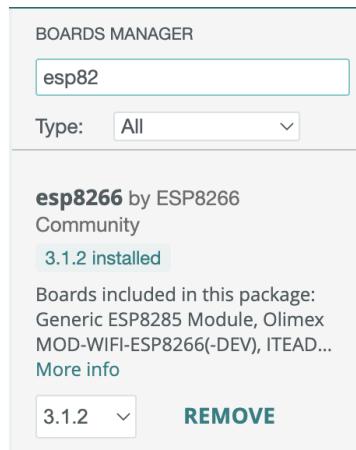
Sebelum masuk ke tahap ini, pastika Arduino IDE sudah ter-*install* pada *device* yang digunakan. Selanjutnya, NodeMCU ESP8266 *board manager* perlu diunduh agar dapat menggunakan *microcontroller*. Hal ini dilakukan dengan pergi ke **Preferences** pada ArduinoIDE dan pada bagian “**Additional boards manager URLs**”, masukkan alamat *package board manager* ESP8266 di [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json). **Gambar 5.4** menunjukkan menu *Preferences*.



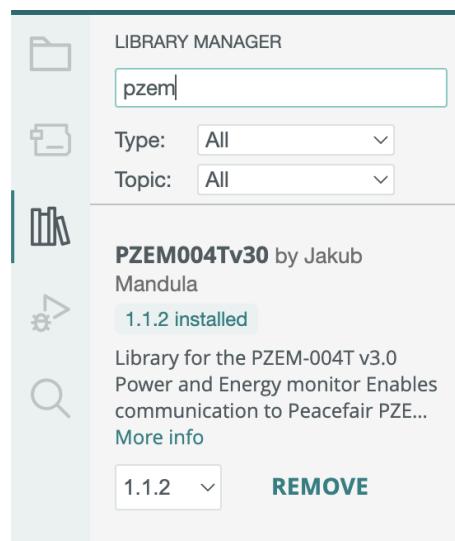
**Gambar 5.4** Tampilan *Preferences*

Selanjutnya, perlu mengunduh *package* yang sudah diunggah dengan cara pergi ke **Tools -> Board-> Board Manager** dan unduh *library* sesuai dengan **Gambar 5.5**. Pada kasus ini, penulis sudah mengunduh *library*. Setelah itu, perlu dilakukan *instalasi library* PZEM-004T agar pembacaan data dari sensor dapat digunakan. Pergi ke **Tools-> Manage Libraries** dan cari *library*

“PZEM004Tv30” oleh Jakub Mandula, sesuai dengan **Gambar 5.6.**



**Gambar 5.5** Library ESP8266 Terinstall



**Gambar 5.6** Library PZEM004Tv30 Terinstall

### 5.2.2. Penulisan Kode *Set Up* Koneksi Ke *Wifi* dan Koneksi ke MQTT Server

Tahap pertama pada bagian ini adalah *import library* yang diperlukan. Setelah *importing library*, berikutnya dilakukan konfigurasi MQTT Server. MQTT *broker* yang digunakan adalah Mosquitto. Perlu didapatkan IP dari *broker* dimana data akan terkirim. Hal ini dilakukan dengan menulis *command* “ping test.mosquitto.org” pada terminal atau cmd. **Gambar 5.7** menunjukkan IP yang didapat pada proses ini.

```
rainataputra@rainatas-MacBook-Pro ~ % ping test.mosquitto.org
PING test.mosquitto.org (91.121.93.94): 56 data bytes
64 bytes from 91.121.93.94: icmp_seq=0 ttl=49 time=281.342 ms
64 bytes from 91.121.93.94: icmp_seq=1 ttl=49 time=286.673 ms
```

**Gambar 5.7** IP yang Dihasilkan dari Mosquitto MQTT Server

Selanjutnya, perlu didefinisikan konstanta yang penting untuk digunakan. Pada **Kode Sumber 5.1**, konstanta pertama yang perlu didefinisikan adalah “ssid” yang menunjukkan nama *wifi* yang akan terhubung dengan *microcontroller*. Selanjutnya konstanta “password” yang menunjukkan sandi *wifi*. Selanjutnya, “mqtt\_server” merupakan IP address server MQTT yang di dapat dari Mosquitto. Ini merupakan alamat dimana data sensor akan dikirim.

Selain konstanta, terdapat juga variabel penting dalam **Kode Sumber 5.1**. Variabel pertama adalah “espClient” yang bertipe “WiFiClient” yang berfungsi sebagai media untuk menghubungkan *microcontroller* ke WiFi. Berikutnya adalah variabel “client(espClient)” yang bertipe “PubSubClient” yang digunakan untuk membuat MQTT *client* agar bisa terhubung ke MQTT *broker*.

Konstanta “MSG\_BUFFER\_SIZE” dan variabel “msg” yang digunakan untuk menampung data yang masuk.

Selanjutnya, dibuat dua buah fungsi, yaitu fungsi “setup\_wifi” yang berfungsi untuk menghubungkan *microcontroller* ke *wifi* dan fungsi “reconnect” yang berfungsi untuk menghubungkan *client* ke *mqtt server* secara terus menerus hingga terhubung. Fungsi “setup\_wifi” dipanggil di dalam fungsi “setup” utama *microcontroller* dan fungsi “reconnect” dipanggil dalam fungsi “loop” utama *microcontroller*. **Kode Sumber 5.2** menunjukkan definisi fungsi “setup wifi” dan “reconnect” serta pemanggilannya pada fungsi-fungsi utama tadi. Perlu diingat bahwa fungsi “setup” pada arduino merupakan fungsi pertama yang dijalankan. Setelah fungsi “setup” selesai dijalankan, fungsi “loop” akan dijalankan secara terus menerus.

*Port* yang digunakan untuk menghubungkan *client* ke *MQTT Server* adalah *port* 1883. Proses koneksi ke *MQTT server* Mosquitto ditunjukkan pada fungsi “setup” pada kalimat “client.setServer(mqtt\_server, 1883);”

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <PZEM004Tv30.h>

const char* ssid = "Tselhome-972E";
const char* password = "81890541";
const char* mqtt_server = "91.121.93.94";
.....
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
```

```
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
....
```

### Kode Sumber 5.1 Variabel pada *Setup* Koneksi *Wifi* dan MQTT Server

```
void setup_wifi() {

    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    randomSeed(micros());

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID
```

```

String clientId = "ESP8266Client-";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str())) {
    Serial.println("connected");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
}
}

void setup() {
pinMode(BUILTIN_LED, OUTPUT);      // Initialize
the BUILTIN_LED pin as an output
Serial.begin(115200);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

void loop() {

if (!client.connected()) {
    reconnect();
}
client.loop();
.....
}

```

**Kode Sumber 5.2** Implementasi Fungsi *Setup Wifi* dan *Reconnect*

### 5.2.3. Penulisan Kode Implementasi Pembacaan Data pada PZEM-004T

Setelah menulis kode koneksi *wifi* dan MQTT server, langkah selanjutnya adalah menulis implementasi kode pembacaan ada pada sensor PZEM-004T. Hal ini dilakukan dengan *library* PZEM004Tv30.h yang sudah diimpor. Implementasi ini dilakukan pada fungsi “loop” utama *microcontroller*. Tujuannya adalah pembacaan hanya dapat dilakukan ketika *wifi* sudah terhubung dan *client* sudah terhubung ke MQTT server. **Kode Sumber 5.3** menunjukkan implementasi pembacaan data listrik pada sensor. Bagian pertama yang perlu diperhatikan adalah konstanta *topic* seperti “POWERTOPIC”, “ENERGYTOPIC” dan seterusnya. Konstanta ini berfungsi sebagai *topic* atau *channel* dimana data dari sensor akan dikirim. Cara kerjanya adalah ketika ingin mengirim data, maka *client* mem-*publish* data melalui *topic* yang sesuai. Misalnya, data power akan dikirim melalui *topic* “POWERTOPIC” yang nilainya dalam kasus ini adalah “listrik-out-informatika/power”. Sama seperti konsep pengiriman, ketika *client* ingin menerima data, maka *client* harus men-*subscribe* pada *topic* yang bersangkutan. Misalnya, ketika *client* ingin menerima data power, *client* harus *subscribe* pada konstanta “listrik-out-informatika/power” pada konstanta “POWERTOPIC”.

Selanjutnya, terdapat inisialisasi variabel “PZEM004Tv30 pzem(14, 12, 0x01);”. Kode ini menunjukkan nomor 14 dan 12 yang menandakan dimana pin TX dan RX berada. Pada sensor pin TX dan RX terhubung pada pin D5 (nomor 14) dan D6 (nomor 12)

pada *microcontroller*. Setelah proses *setup* PZEM004T selesai, langkah berikutnya adalah mendefinisikan variabel yang akan menampung seluruh jenis data pada sensor. Variabel ini bertipe *float* karena data yang dikirim seluruhnya berbentuk desimal. Adapun variabel yang akan dibaca adalah *power*, *energy*, *voltage*, *current*, *frequence*, dan *pf* (*power factor*). Setelah variabel didefinisikan, data dapat dibaca pada fungsi “loop” utama dengan menggunakan variabel “*pzem*” yang sudah didefinisikan tadi. Misalnya, jika ingin membaca *power*, sintaks yang digunakan adalah “Power = *pzem.power()*”. Nilai yang akan diberikan adalah nilai *power* dalam bentuk Watt. Apabila nilai tidak tersedia, nilai yang dikembalikan adalah Nan. Langkah ini dilakukan untuk semua jenis data yang dibaca. Jika seluruh data sudah dibaca, langkah selanjutnya adalah mengirim data ke MQTT Server. Pengiriman data ini dilakukan dengan variabel client yang bertipe PubSubClient dengan fungsi *publish*. Misalnya, jika client ingin mengirim data *power*, sintaks yang digunakan adalah “snprintf (msg, MSG\_BUFFER\_SIZE, "%lf", Power);”, lalu “client.publish(POWERTOPIC, msg);”. *Publishing* data dilakukan untuk seluruh jenis data sensor.

```
....  
// MQTT BROKER SETTINGS  
#define INTOPIC "LISTRIK_BROKER_ITS"  
#define POWERTOPIC  
"listrik-out-informatika/power"  
#define ENERGYTOPIC  
"listrik-out-informatika/energy"  
#define VOLTAGETOPIC
```

```

"listrik-out-informatika/voltage"
#define CURRENTTOPIC
"listrik-out-informatika/current"
#define FREQUENCYTOPIC
"listrik-out-informatika/frequency"
#define PFTOPIC "listrik-out-informatika/pf"

//inisialisasi objek untuk pzem (sensor arus)
PZEM004Tv30 pzem(14, 12, 0x01); // D5 = 14 , D6 =
12
//Variabel penampung untuk nilai sesnsor arus
float Power, Energy, Voltage, Current, Frequency,
Pf;

.....

void loop() {

    //baca nilai power (kW)
    Power = pzem.power();
    //jika gagal membaca power
    if(isnan(Power)){
        Serial.println("Gagal membaca Power");
    }
    else{
        Serial.print("Power :");
        Serial.print(Power);
        Serial.println("W");
    }

    // baca nilai Energy (Daya/jam)
    Energy = pzem.energy();
    //jika gagal membaca energy
    if(isnan(Energy)){
        Serial.println("Gagal membaca Energy");
    }
}

```

```

else{
    Serial.print("Energy :");
    Serial.print(Energy);
    Serial.println("kWh");
}
.....
snprintf (msg, MSG_BUFFER_SIZE, "%lf", Power);
client.publish(POWERTOPIC, msg);
snprintf (msg, MSG_BUFFER_SIZE, "%lf", Energy);
client.publish(ENERGYTOPIC, msg);
.....
}

```

**Kode Sumber 5.3** Implementasi Pembacaan Data dari Sensor

### 5.3. Implementasi *Homeassistant*

Jenis *Homeassistant* yang digunakan pada kerja praktek ini adalah *homeassistant container*. *Container* ini dijalankan melalui *docker*. Alasan digunakannya *container* adalah agar sistem *homeassistant* terisolasi. *Container* ini di-host pada server Arsitektur Komputer dan Jaringan Teknik Informatika Institut Teknologi Sepuluh Nopember. Langkah pertama yang perlu dilakukan adalah membuat entitas sensor pada *homeassistant*. Caranya adalah pergi ke *folder home assistant -> config* dan buka *file configuration.yaml*. Di *file* inilah entitas sensor didefinisikan. **Kode Sumber 5.4** menunjukkan isi dari *file configuration.yaml*. Hal yang perlu diperhatikan pada *file* ini adalah setiap sensor yang didefinisikan berada dalam blok “*mqtt*”. Hal ini penting

agar integrasi pada MQTT server dapat berjalan dengan lancar.

```
# Example configuration.yaml entry
mqtt:
  sensor:
    - name: "Informatika Power 1"
      state_topic:
        "listrik-out-informatika/power"
        icon: mdi:home-lightning-bolt
        unique_id: "power-sensor-1"
        device_class: "energy"
        unit_of_measurement: "W"
        payload_not_available: "nan"
    - name: "Informatika Energy 1"
      state_topic:
        "listrik-out-informatika/energy"
        icon: mdi:transmission-tower
        unique_id: "energy-sensor-1"
        device_class: "energy"
        unit_of_measurement: "kWh"
        state_class: "total"
        payload_not_available: "nan"
    - name: "Informatika Voltage 1"
      state_topic:
        "listrik-out-informatika/voltage"
        icon: mdi:transmission-tower
        unique_id: "voltage-sensor-1"
        device_class: "energy"
        unit_of_measurement: "V"
```

```

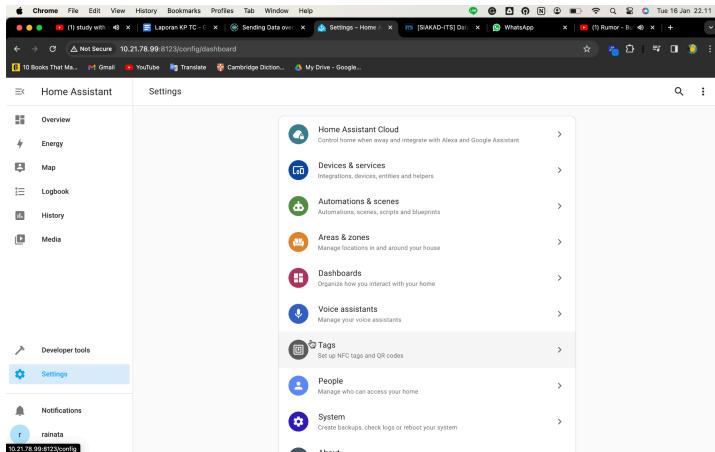
state_class: "total"
payload_not_available: "nan"
- name: "Informatika Current 1"
  state_topic:
"listrik-out-informatika/current"
  icon: mdi:transmission-tower
  unique_id: "current-sensor-1"
  device_class: "energy"
  unit_of_measurement: "A"
  state_class: "total"
  payload_not_available: "nan"
- name: "Informatika Frequency 1"
  state_topic:
"listrik-out-informatika/frequency"
  icon: mdi:transmission-tower
  unique_id: "frequency-sensor-1"
  device_class: "energy"
  unit_of_measurement: "Hz"
  state_class: "total"
  payload_not_available: "nan"
- name: "Informatika Power Factor 1"
  state_topic:
"listrik-out-informatika/pf"
  icon: mdi:transmission-tower
  unique_id: "pf-sensor-1"
  payload_not_available: "nan"

```

#### **Kode Sumber 5.4 File configuration.yaml**

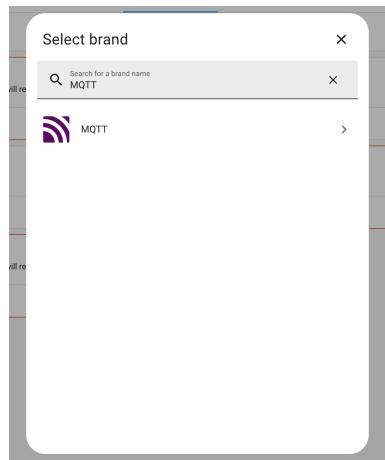
Setelah seluruh entitas sensor sudah didefinisikan pada *file configuration.yaml*, langkah berikutnya adalah proses integrasi MQTT di *home assistant*. Caranya pergi ke menu

**Settings -> Devices and services**, sesuai lalu tekan tombol **Add integration** pada bagian kanan bawah. Selanjutnya pilih MQTT dan klik opsi MQTT. Hal ini sesuai dengan **Gambar 5.9**.



**Gambar 5.8** Menu *Settings Home Assistant*

Setelah menu MQTT dipilih, klik opsi **Configure** dan masukkan IP server Mosquирро serta port yang sesuai. Dalam kasus ini, IP adalah **91.121.93.94** dan port adalah 1883. Hal ini sesuai dengan apa yang ditunjukkan pada **Gambar 5.10**. Setelah dijalankan, akan muncul sejumlah entitas yang sesuai dengan apa yang didefinisikan pada *configuration.yaml* di **Kode Sumber 5.4**. Langkah selanjutnya yang perlu dilakukan adalah mengonfigurasi setiap entitas yang ada. Opsi ini dilakukan untuk mengelompokkan setiap entitas sesuai dengan lokasi terinstallnya sensor. Misalnya, karena sensor ini dipasang pada panel A, maka setiap entitas dari sensor akan dikelompokkan ke dalam area “Panel A”. Hal ini ditunjukkan pada **Gambar 5.11**.

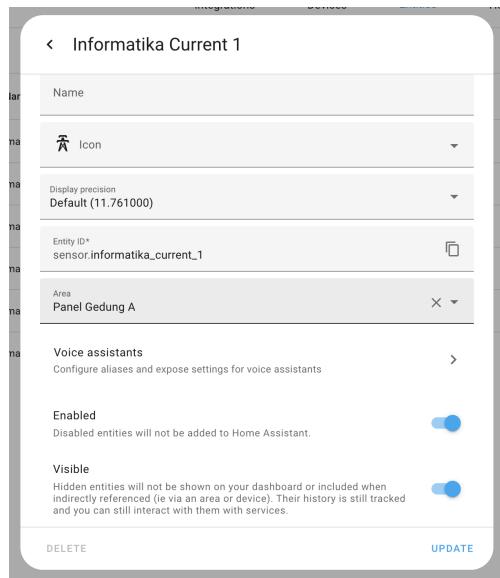


**Gambar 5.9** Opsi Integrasi MQTT pada *Homeassistant*

A screenshot of a configuration form titled "Broker options". The form has a header with a close button ("X") and a sub-header "Please enter the connection information of your MQTT broker.". It contains several input fields: "Broker\*" with the value "91.121.93.94", "Port\*" with the value "1883", "Username" (empty), and "Password" (empty). To the right of the password field is a visibility toggle icon. At the bottom right of the form is a blue "NEXT" button.

**Gambar 5.10** Form Konfigurasi MQTT Broker pada *Homeassistant*

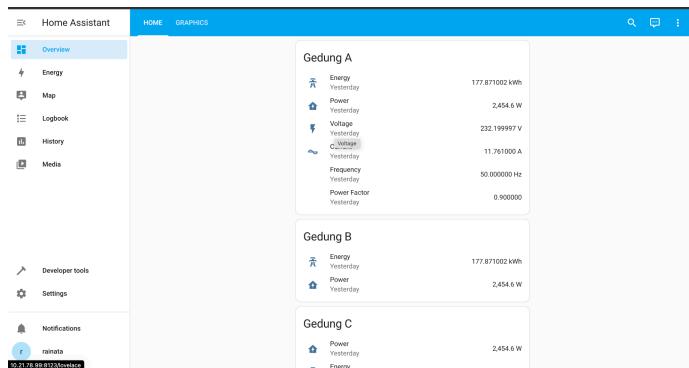
Jika semua sudah diatur, *list entitas* akan tampak seperti **Gambar 5.12**. Pengelompokan berdasarkan area akan mempermudah proses pemantauan nantinya.



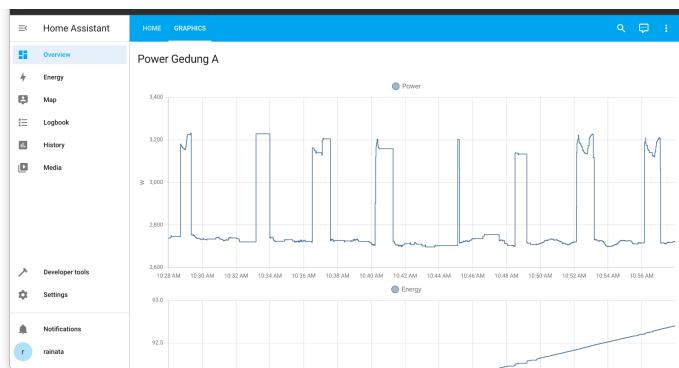
**Gambar 5.11 Konfigurasi Entitas *Current* yang Dikelompokkan ke Area Panel Gedung A**

<input type="checkbox"/>	Name	Entity ID	Integration	Area	Disabled by	Status
<input type="checkbox"/>	Informatika ..	sensor.informatika_current_1	MQTT	Panel Gedung A	—	—
<input type="checkbox"/>	Informatika ..	sensor.informatika_energy	MQTT	Panel Gedung A	—	—
<input type="checkbox"/>	Informatika ..	sensor.informatika_frequency_1	MQTT	Panel Gedung A	—	—
<input type="checkbox"/>	Informatika ..	sensor.informatika_power	MQTT	Panel Gedung A	—	—
<input type="checkbox"/>	Informatika ..	sensor.informatika_power_factor_1	MQTT	Panel Gedung A	—	—
<input type="checkbox"/>	Informatika ..	sensor.informatika_voltage_1	MQTT	Panel Gedung A	—	—

**Gambar 5.12 List Entitas yang sudah Terkonfigurasi**  
Jika seluruhnya sudah dilakukan, data sudah siap ditampilkan pada *home assistant*. **Gambar 5.13** dan **Gambar 5.14** menunjukkan tampilan pada *dashboard* dan tampilan grafis data.



**Gambar 5.13** Tampilan Dashboard pada Home Assistant



**Gambar 5.14** Tampilan Grafis pada Home Assistant

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap sensor listrik yang sudah terpasang di Teknik Informatika Institut Teknologi Sepuluh Nopember. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi pada alat.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap sensor listrik yang sudah terpasang di Teknik Informatika guna melihat apakah sensor dapat mengirim data yang sesuai dan akurat.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

- a. Kemampuan sensor dalam mengirim data sepanjang hari.
- b. Kemampuan sensor untuk mengirim data yang benar dan akurat.
- c. Kemampuan sensor untuk mengirim data kurang dari 5 detik.

#### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan memantau performa sensor pada aplikasi *Home Assistant*. Metode pengamatan yang dilakukan untuk melihat apakah sensor memenuhi kriteria adalah:

- a. Melihat pada histori grafik apakah data selalu ada pada setiap *timestamp*
- b. Melihat kesesuaian data yang dikirim dengan aktivitas listrik pada teknik informatika.

Misalnya, pada siang hari akan ada peningkatan daya karena aktivitas listrik berlangsung paling tinggi pada waktu tersebut.

- c. Melihat waktu *update* setiap entitas.

#### 6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sensor sesuai dengan skenario yang sudah ditentukan. **Tabel 6.1** di bawah ini menjelaskan hasil uji coba tersebut.

**Tabel 6.1.** Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Sensor dapat mengirim data sepanjang hari	Terpenuhi
Sensor mengirim data yang benar dan akurat	Terpenuhi
Sensor untuk mengirim data kurang dari 5 detik	Terpenuhi

*[Halaman ini sengaja dikosongkan]*

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan kerja praktik alat monitoring listrik ini adalah :

- a. Pemantauan listrik dapat meningkatkan efisiensi karena energi yang keluar dapat terpantau melalui sistem sehingga jika terjadi suatu anomali, dapat ditangani secara langsung.
- b. Proses implementasi sistem pemantauan listrik dimulai dari proses perakitan sensor, *set-up* MQTT Server, penulisan kode pada Arduino IDE hingga, *set-up homeassistant*, dan instalasi sensor pada panel listrik.

#### **7.2. Saran**

Saran untuk perancangan alat monitoring sensor berikutnya adalah :

- a. Sebaiknya digunakan *box* pelindung pada sensor agar lebih aman dan mampu melindungi pemasang sensor serta potensi-potensi interupsi di saluran listrik lainnya.
- b. Dianjurkan untuk memasang penggerat pada *pin* kabel *jumper* dengan sensor. Hal ini untuk menghindari adanya kemungkinan putusnya kabel dari pin.

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR PUSTAKA**

- [1] Irwan Dinata, Wahri Sunanda. 2015. “IMPLEMENTASI WIRELESS MONITORING ENERGI LISTRIK BERBASIS WEB DATABASE”, Jurusan Teknik Elektro, Fakultas Teknik Universitas Bangka Belitung, Vol: 4, No.1, Maret 2015 ISSN: 2302-294. [Accessed 02 Januari 2024]
- [2] Muhammad Fitra Zambak. 2022. “Monitoring Pemakaian Listrik Berbasis Mikrokontroler”, ISBN : 978-623-408-057-5 [Accessed 02 Januari 2024]
- [3] Ari Kurnianto, Joseph Dedy Irawan, FX. Ari Wibisono. 2022. “PENERAPAN IOT (INTERNET OF THINGS) UNTUK CONTROLLING LAMPU MENGGUNAKAN PROTOKOL MQTT BERBASIS WEB” , JATI (Jurnal Mahasiswa Teknik Informatika), Vol. 6 No. 2, September 2022. [Accessed 05 Januari 2024]
- [4] Z. B. Abilovani, W. Yahya, dan F. A. Bakhtiar, “Implementasi Protokol MQTT Untuk Sistem Monitoring Perangkat IoT,” J. Pengemb. Teknol. Inf. dan Ilmu Komputer., vol. 2, no. 12, hal. 7521 7527, 2018, [Accesses 07 Januari 2024]
- [5] Raden ajeng gusti ramadhianti, Ir.Cok indra parth, I G A pt raka agung. 2018. “Rancang bangun monitoring energi listrik menggunakan sms berbasis mikrokontroler atmega 328” . E-jurnal SPEKTRUM Vol.5, no.1 [ Accessed 02 Januari 2024]

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS I**

Nama : Zunia Aswaroh  
Tempat, Tanggal Lahir : Lamongan, 25 Februari 2002  
Jenis Kelamin : Perempuan  
Telepon : +6285785346461  
Email : zuniaaswaroh@gmail.com

AKADEMIS

## **BIODATA PENULIS II**

Nama : Ida Bagus Kade Rainata Putra Wibawa  
Tempat, Tanggal Lahir : Denpasar, 11 November 2001  
Jenis Kelamin : Laki-Laki  
Telepon : +6282235012819  
Email : kriegcom77@gmail.com

AKADEMIS