

RANCANG BANGUN ALAT MONITORING DAYA LISTRIK  
BERBASIS *INTERNET OF THINGS (IOT)*



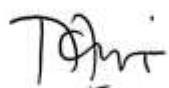
PROGRAM STUDI D-3 TEKNIK ELEKTRONIKA  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI UJUNG PANDANG  
MAKASSAR  
2023

## HALAMAN PENGESAHAN

Laporan Tugas Akhir ini dengan judul "Rancang Bangun Alat Monitoring Daya Listrik Berbasis *Internet of Things (IoT)*" oleh Mhd. Khairul Nizam NIM 32320033 dan Muh. Iksan M.Rahmat NIM 32320044 dinyatakan layak untuk diujikan.

Makassar, Agustus 2023

Pembimbing I



Nur Aminah, S.T., M.T.  
NIP. 196607211990112001

Pembimbing II



Reski Praminasari, S.T., M.T.  
NIP. 198401042014042001



## HALAMAN PENERIMAAN

Pada hari ini, Rabu 6 September 2023, Tim Penguji Ujian Sidang Laporan Tugas Akhir telah menerima dengan baik Laporan Tugas Akhir oleh mahasiswa: Mhd. Khairul Nizam NIM 323 20 033 dan Muh. Iksan M.Rahmat NIM 323 20 044 dengan judul “Rancang Bangun Alat Monitoring Daya Listrik Berbasis *Internet of Things (IoT)*”.

Makassar, September 2023

Tim Penguji Ujian Sidang Tugas Akhir:

1. Kartika Dewi, S.T., M.T.	Ketua	(.....)
2. Sulaeman, S.T., M.T.	Sekretaris	(.....)
3. Zainal Abidin, S.T., M.T.	Anggota	(.....)
4. Bagus Prasetyo, S.Pd.,M.T.	Anggota	(.....)
5. Nur Aminah, S.T., M.T.	Pengarah 1	(.....)
6. Reski Praminasari, S.T., M.T.	Pengarah 2	(.....)

## **KATA PENGANTAR**

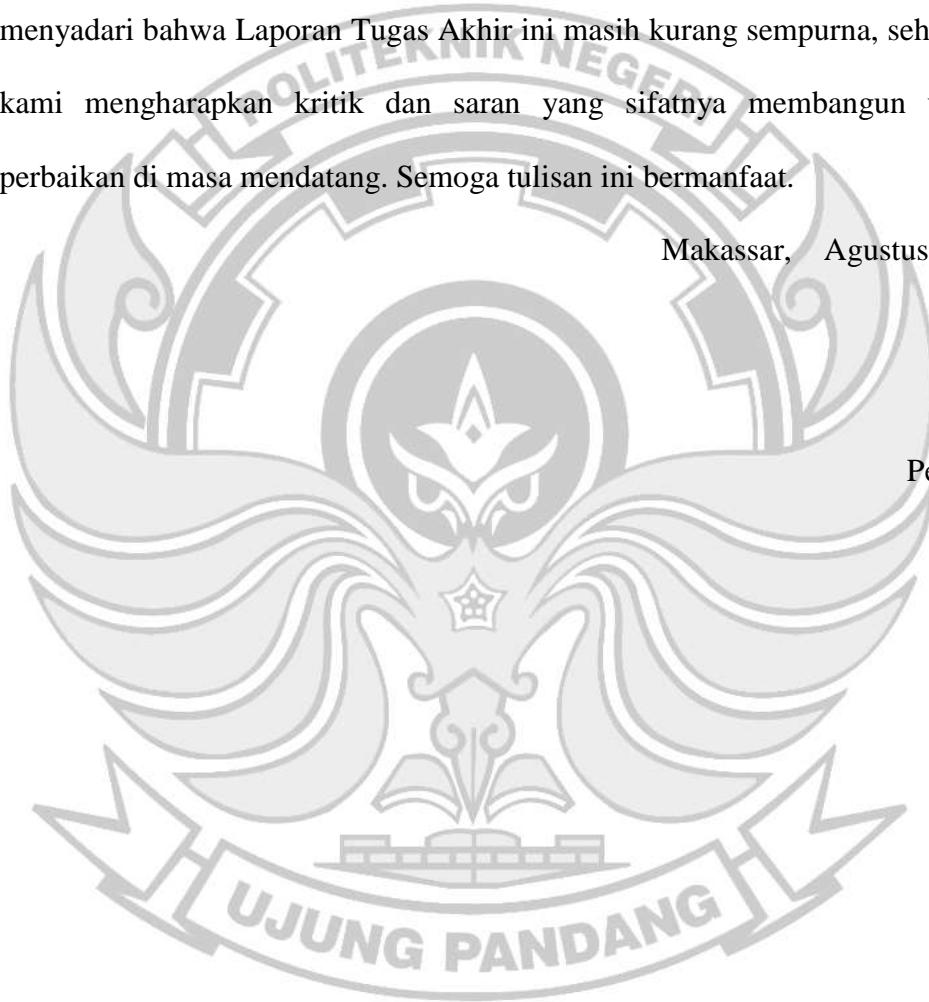
Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, penulisan Laporan Tugas Akhir ini yang berjudul “Rancang Bangun Alat Monitoring Daya Listrik Berbasis *Internet of Things* (IoT)” dapat diselesaikan dengan baik. Laporan Tugas Akhir ini merupakan hasil penelitian yang dilaksanakan mulai Februari sampai Agustus 2023 bertempat di Politeknik Negeri Ujung Pandang. Kesempatan ini penulis menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua kami yang selalu setia mendoakan kami dan memberikan dorongan dan motivasi baik moril maupun material.
2. Bapak Ir. Ilyas Mansur, M.T. sebagai Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. sebagai Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Muh. Chaerur Rijal, S.T., M.T. sebagai Koordinator Program Studi Teknik Elektronika.
5. Ibu Nur Aminah, S.T., M.T. sebagai Pembimbing 1 dan ibu Reski Praminasari, S.T., M.T. sebagai Pembimbing 2 yang telah muncurahkan waktu dan kesempatannya untuk mengarahkan penulis dalam menyelesaikan Laporan Tugas Akhir ini.
6. Wali Kelas 3B D3 Teknik Elektronika Angkatan 2020, Ibu Fitriaty Pangerang, S.T., M.T.

7. Bapak/Ibu Dosen Teknik Elektronika Politeknik Negeri Ujung Pandang yang telah membekali ilmu kepada penulis selama mengikuti proses perkuliahan.
8. Teman-teman kelas 3B Elektronika angkatan 2020 yang telah membantu dalam penyelesaian Laporan Tugas Akhir.
9. Semua pihak yang terkait dalam penulisan Laporan Tugas Akhir. Penulis menyadari bahwa Laporan Tugas Akhir ini masih kurang sempurna, sehingga kami mengharapkan kritik dan saran yang sifatnya membangun untuk perbaikan di masa mendatang. Semoga tulisan ini bermanfaat.

Makassar, Agustus 2023

Penulis



## DAFTAR ISI

HALAMAN PENGESAHAN.....	i
HALAMAN PENERIMAAN .....	<b>Error! Bookmark not defined.</b>
DAFTAR ISI.....	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR .....	viii
DAFTAR LAMPIRAN.....	x
SURAT PERNYATAAN.....	xi
RINGKASAN .....	xii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Ruang Lingkup Kegiatan.....	3
1.4 Tujuan Kegiatan .....	3
1.5 Manfaat Kegiatan .....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Studi Pendahuluan .....	4
2.2 Meteran Listrik Prabayar.....	5
2.3 Segitiga Daya .....	7
2.4 Modul ESP32 DEVKIT V1.....	10
2.5 Modul Sensor PZEM-004T .....	12
2.6 Modul Power Supply .....	14
2.7 Modul Relay .....	16
2.8 MOC3021 .....	18
2.9 Resistor .....	19
2.10 Transistor.....	21
2.11 Dioda .....	24

2.11 <i>Pilot Lamp</i> .....	25
2.12 Internet of things .....	26
2.13 <i>Fremework Laravel</i> .....	27
2.14 MYSQL.....	33
BAB III METODE KEGIATAN .....	<b>Error! Bookmark not defined.</b>
3.1    Tempat dan Waktu Kegiatan .....	<b>Error! Bookmark not defined.</b>
3.2    Alat dan Bahan .....	<b>Error! Bookmark not defined.</b>
3.3    Prosedur Kerja.....	<b>Error! Bookmark not defined.</b>
BAB IV HASIL DAN DESKRIPSI KEGIATAN	<b>Error! Bookmark not defined.</b>
4.1    Hasil Perancangan .....	<b>Error! Bookmark not defined.</b>
4.1.1    Hasil pembuatan dan perakitan mekanik alat.	<b>Error! Bookmark not defined.</b>
4.1.2    Hasil Perancangan Sistem Elektronik .....	<b>Error! Bookmark not defined.</b>
4.1.3    Hasil Akhir Perancangan Alat Secara Keseluruhan.....	<b>Error! Bookmark not defined.</b>
4.2    Pengujian Dan Analisi pada Alat .....	<b>Error! Bookmark not defined.</b>
4.2.1    Pengujian Modul PZEM-004T.....	<b>Error! Bookmark not defined.</b>
4.2.2    Pengujian Pada Modul ESP32 .....	<b>Error! Bookmark not defined.</b>
4.2.3    Pengujian Pada Modul Hi-Link HLK-PM01 .	<b>Error! Bookmark not defined.</b>
4.2.4    Pengujian Pada Modul <i>Relay</i> dan Terminal...	<b>Error! Bookmark not defined.</b>
4.2.5    Pengujian Lampu Pilot.....	<b>Error! Bookmark not defined.</b>
4.3    Analisa Data .....	<b>Error! Bookmark not defined.</b>
4.4    Spesifikasi Alat.....	<b>Error! Bookmark not defined.</b>
BAB V PENUTUP.....	35
5.1    Kesimpulan.....	35
5.2    Saran .....	35
DAFTAR PUSTAKA .....	37
LAMPIRAN .....	40



## DAFTAR TABEL

Tabel 2. 1 Golongan Pelanggan PLN per Juli-September 2023 ..... 6

Tabel 2. 2 Daftar Urutan Daya Listrik yang disediakan PLN..... 9

Tabel 2. 3 Kode Warna Resistor ..... 19

Tabel 2. 4 Kode Toleransi Resistor..... 20

Tabel 3. 1 Daftar Alat yang Digunakan

**Error! Bookmark not defined.**

Tabel 3. 2 Daftar Bahan yang Digunakan

**Error! Bookmark not defined.**

Tabel 4. 1 Pengukuran tegangan pada ESP32

**Error! Bookmark not defined.**

Tabel 4. 2 Pengukuran tegangan pada Modul Hi-Link HLK-PM01

**Error! Bookmark not defined.**

Tabel 4. 3 Pengujian Tegangan Modul Relay

**Error! Bookmark not defined.**

Tabel 4. 4 Pengujian lampu pilot

**Error! Bookmark not defined.**

Tabel 4. 5 Pengukuran daya listrik

**Error! Bookmark not defined.**

## DAFTAR GAMBAR

Gambar 2. 1 Segitiga Daya .....	8
Gambar 2. 2 Bentuk fisik Modul ESP32 DEVKIT V1 .....	10
Gambar 2. 3 Pinout ESP32 .....	12
Gambar 2.4 Modul PZEM-004T .....	13
Gambar 2.5 Ukuran dan fungsi pin modul Hi-Link Pm01 .....	14
Gambar 2. 6 Modul Hi-Link Pm01 .....	15
Gambar 2. 7 Bentuk fisik modul relay .....	16
Gambar 2. 8 Struktur relay .....	17
Gambar 2. 9 Bentuk fisik MOC3021 .....	18
Gambar 2. 10 Schematic MOC3021 .....	18
Gambar 2. 11 Bentuk dan simbol resistor .....	19
Gambar 2. 12 Cara Membaca Kode Warna Resistor .....	21
Gambar 2. 13 Bentuk dan simbol transistor .....	22
Gambar 2. 14 Cara Kerja Transistor .....	24
Gambar 2. 15 Bentuk dan symbol dioda .....	24
Gambar 2. 16 <i>Pilot Lamp</i> .....	26
Gambar 2. 17 Aliran informasi dengan konsep Internet of Things .....	27
Gambar 2. 18 <i>Framework</i> Laravel .....	27
Gambar 2. 19 Struktur folder dari Laravel 8 .....	32
Gambar 3. 1 Diagram Alir Metode Kegiatan	

**Error! Bookmark not defined.**

Gambar 3. 2 Diagram Blok

**Error! Bookmark not defined.**

Gambar 3. 3 Rangkaian Skematik

**Error! Bookmark not defined.**

Gambar 3. 4 *Layout PCB*

**Error! Bookmark not defined.**

Gambar 3. 5 Desain Alat Tampak Atas

**Error! Bookmark not defined.**

Gambar 3. 6 Desain Alat Tampak Luar Dalam

**Error! Bookmark not defined.**

Gambar 3. 7 Desain Alat Tampak Depan

**Error! Bookmark not defined.**

Gambar 3. 8 Desain Alat Tampak Samping

**Error! Bookmark not defined.**

Gambar 3. 9 *Flowchart* Tampilan *Dashboard* website

**Error! Bookmark not defined.**

Gambar 3. 10 *Flowchart* Input token

**Error! Bookmark not defined.**

Gambar 3. 11 *Flowchart* cek token

**Error! Bookmark not defined.**

Gambar 3. 12 Tampilan awal dan login

**Error! Bookmark not defined.**

Gambar 3. 13 Tampilan halaman *dasboard*

**Error! Bookmark not defined.**

Gambar 3. 14 Tampilan Top Up pulsa

**Error! Bookmark not defined.**

Gambar 3. 15 Tampilan Cek Pulsa

**Error! Bookmark not defined.**

Gambar 4. 1 Tampilan Tampak Atas

**Error! Bookmark not defined.**

Gambar 4. 2 Tampilan Tampak Depan

**Error! Bookmark not defined.**

Gambar 4. 3 Tampilan Tampak Samping

**Error! Bookmark not defined.**

Gambar 4. 4 Jalur Papan PCB

**Error! Bookmark not defined.**

Gambar 4. 5 Pengujian pada Modul PZEM-004T

**Error! Bookmark not defined.**

## DAFTAR LAMPIRAN

Lampiran 1 Proses Pebuatan .....	40
Lampiran 2 Web Monitoring .....	41
Lampiran 3 Pengambilan data.....	42
Lampiran 4 Daftar Waktu Kegiatan.....	44
Lampiran 5 Listing Program Menggunakan Arduino IDE .....	65



## SURAT PERNYATAAN

Kami yang bertanda tangan dibawah ini:

Nama : Mhd. Khairul Nizam / Muh. Ikhsan M Rahmat

Nim : 32320033 / 32320044

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Laporan Tugas Akhir ini yang berjudul **Rancang Bangun Alat Monitoring Daya Listrik Berbasis *Internet of Things* (IoT)** merupakan gagasan dan hasil karya kami sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun. Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam Laporan Tugas Akhir ini. Jika pernyataan saya tersebut di atas tidak benar, kami siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 2023

Mhd. Khairul Nizam

NIM 32320033

Muh. Ikhsan M Rahmat

NIM 32320044

## RANCANG BANGUN ALAT MONITORING DAYA LISTRIK

### BERBASIS *INTERNET OF THINGS* (IOT)

#### RINGKASAN

Manajemen energi listrik dapat dilakukan dengan melihat konsumsi energi listrik pada monitor alat dan kemudian mematikan peralatan-peralatan elektronik yang tidak digunakan sehingga biaya penggunaan energi listrik tidak membengkak. Agar bisa melakukan manajemen listrik yang lebih baik dalam kasus rumah tangga, maka diperlukan suatu perangkat elektronik yang dapat mempermudah pekerjaan manusia dengan memanfaatkan kemajuan teknologi. Penelitian ini bertujuan untuk merancang sistem secara *real-time* untuk memonitoring penggunaan daya listrik yang berbasis *Internet of Things* (IoT). Dengan merancang alat yang dapat mengukur tegangan, arus, dan daya listrik serta dapat dimonitoring pada *webserver*. Sistem real-time monitoring dibuat dengan modul ESP32 yang mengirim data ke database *webserver* monitoringlistrik.online, jika data dipicu oleh modul ESP32 maka server akan meminta data dari modul sensor PZEM-004T. Setelah meminta data, modul ESP32 mengirim data String ke *webserver* dan akan disimpan ke database, dashboard pada website akan terupdate otomatis setelah terdapat data baru yang masuk. Sistem monitoring daya listrik berbasis IoT dapat melakukan monitoring dan pengukuran tegangan, arus, dan daya listrik dengan rata-rata tingkat keakuratan sensor di atas 95%. Aplikasi *web* yang dibuat dapat menampilkan data secara *real-time* dan dapat diakses dimana saja.

**Kata kunci** : Modul ESP32, Modul Sensor PZEM-004T, *website*, *real-time*.

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan Teknologi yang semakin pesat, mengharuskan perangkat-perangkat elektronik yang dibuat dan dirancang membantu kebutuhan manusia dalam kehidupan sehari-hari. Perangkat-perangkat elektronik sangat mudah dijumpai dikehidupan manusia, sehingga kehadiran perangkat-perangkat elektronik membantu pekerjaan manusia menjadi lebih cepat, efisien, dan efektif. Pada pemakaian perangkat-perangkat elektronik tersebut tentunya memerlukan pasokan energi listrik dan sesuai dengan Undang – Undang Nomor 30 Tahun 2009 tentang ketenagalistrikan yang mengatur mengenai peningkatan tarif dasar listrik dalam rumah tangga maupun industri kecil yang dikarenakan pencabutan subsidi listrik.

Energi listrik merupakan suatu kebutuhan penting bagi manusia dan penggunaan energi listrik yang cukup banyak terdapat pada sektor rumah tangga. Penggunaan daya listrik bergantung pada pemakaian. Semakin banyak peralatan yang digunakan maka daya yang terpakai juga akan semakin besar sehingga dapat menyebabkan beban arus yang berlebih (Ardiansyah, 2019). Perlunya dilakukan manajemen konsumsi energi listrik dalam pemakaian, namun dilakukannya manajemen energi listrik tidak cukup hanya menggunakan Kwh meter saja, karena menggunakan Kwh meter hanya bertugas untuk memonitor dan membatasi penggunaan listrik secara keseluruhan di rumah. Oleh karena itu, dibutuhkan kesadaran untuk benar-benar menghemat dalam pemakaian energi listrik. Manajemen energi listrik dapat dilakukan dengan melihat konsumsi energi listrik pada monitor alat dan kemudian mematikan peralatan-peralatan elektronik yang

tidak digunakan sehingga biaya penggunaan energi listrik tidak membengkak. Agar bisa melakukan manajemen listrik yang lebih baik dalam kasus rumah tangga, maka diperlukan suatu perangkat elektronik yang dapat mempermudah pekerjaan manusia dengan memanfaatkan kemajuan teknologi (Anantama dkk.,2020)

Dalam penghematan energi listrik dibutuhkan sebuah alat pemonitoring penggunaan energi listrik pada perangkat-perangkat elektronik seperti televisi, kulkas, penanak nasi, mesin cuci, dan pendingin ruangan. Terpasangnya alat monitoring akan memudahkan pengguna dalam melakukan penghematan penggunaan energi listrik dengan menampilkan daya, tegangan, arus, dan biaya pemakaian perangkat elektronik tersebut. Untuk mengatasi permasalahan tersebut maka dirancang sebuah alat rancang bangun alat monitoring daya listrik berbasis IoT. Terciptanya alat ini diharapkan akan mempermudah pengguna energi listrik untuk mengetahui berapa besar daya pengguna pada Kwh listrik rumah. Sehingga para pengguna energi listrik dapat menghemat penggunaanya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dari perancangan ini adalah sebagai berikut:

- a. Bagaimana sistem monitoring ini dapat mengukur tegangan, arus, dan daya listrik?
- b. Bagaimana membuat sistem monitoring *real-time* untuk tegangan, arus, dan daya listrik?

### **1.3 Ruang Lingkup Kegiatan**

Penelitian ini dibatasi pada ruang lingkup yang terdiri dari:

- a. Hal yang dimonitoring pada alat ini yaitu daya yang ada pada Kwh meteran rumah.
- b. Menggunakan *device* yaitu: laptop dan *Website Monitoringlistrik.online*

### **1.4 Tujuan Kegiatan**

Adapun tujuan yang ingin dicapai dari perancangan ini adalah sebagai berikut:

- a. Merancang alat monitoring daya listrik berbasis *Internet of Things* (IoT).
- b. Membuat sistem monitoring tegangan, arus, dan daya listrik secara *real-time*.

### **1.5 Manfaat Kegiatan**

Adapun manfaat dari perancangan kali ini yaitu:

- a. Memberikan kemudahan dalam monitoring penggunaan daya listrik.
- b. Meminimalisir terjadinya suatu kondisi yang tidak diinginkan (harga biaya listrik yang menjadi besar).

## BAB II TINJAUAN PUSTAKA

### 2.1 Studi Pendahuluan

Tinjauan Pustaka ini bertujuan untuk menerangkan beberapa hasil penelitian sebelumnya yang terkait dengan monitoring daya listrik. Monitoring daya listrik berbasis IoT (*Internet of Things*) dengan menggunakan perangkat laptop.

Dalam penelitian ini penulis merancang dan menghasilkan alat untuk pengukuran pemakaian beban listrik rumah tangga pada setiap pemakaian beban menggunakan Arduino “Pendeteksi Pemakaian Beban Listrik Rumah Tangga” (Ardianto dkk 2019). Persamaan yang peneliti terdahulu teliti dengan penelitian ini yaitu pengukuran beban listrik pada rumah tangga.

Penelitian yang dilakukan oleh Irwan Dinata (2015) dengan judul Implementasi *Wireless Monitoring Energi Listrik Berbasis Web Database*. Penelitian ini bertujuan untuk menggantikan sistem pengukuran energi listrik secara manual dan konvensional. Penelitian ini menggunakan 4 (empat) bagian utama yaitu sensor, *processor*, display, dan network. Pada bagian sensor terdiri dari *Current Transformer* dan AC to AC Power Adaptor. Kemudian *Processor* digunakan oleh arduino UNO untuk mengolah hasil kerja pada sensor. Display menampilkan data keluaran secara *real-time*. Sedangkan pada network terdiri dari 3G Router, Ethernet Shield, 3G Modem sebagai komunikasi ke Server Database untuk penyimpanan dan pengolahan data. Dari hasil pengujian dengan beban 120 Watt menunjukan hasil dari Vrms yang ditampilkan pada LCD sebesar 218 Volt, hasil pengukuran dari *Clamp Meter* sebesar 216 Volt. Sedangkan pada arus menunjukan hasil dari I rms sebesar 0.44 ampere

hasil pengukuran dari Clamp Meter sebesar 0.5 ampare. Selain itu peneltian ini berhasil mengirimkan pengukuran setiap saat ke Database Server pada aplikasi *Wireless energy Monitoring* yang telah dirancang. Persamaan yang peneliti terdahulu teliti dengan penelitian ini yaitu monitoring pemakaian daya listrik.

Menurut Nusa dkk (2015) dalam penelitiannya membahas mengenai Pengukuran Daya Listrik *Real Time* dengan Menggunakan Modul Sensor Arus ACS 712. Dilakukannya penelitian ini sehingga dapat menghemat penggunaan energi listrik dengan cara mengetahui daya pada penggunaan perangkat-perangkat elektronik, sehingga dapat dilakukannya penghematan energi listrik. Dari hasil pengujian yang dilakukan pada perangkat-perangkat elektronik menunjukan hasil bahwa ketika tegangan pada sistem terlalu tinggi/rendah maka akan mengganggu dan merusak beban. Kondisi pada beban memiliki nilai variatif tersendiri, bila beban mengalami kenaikan maka tegangan di ujung penerimaan akan menurun dan sebaliknya. Saat pengujian nilai arus mengalami perubahan yang disebabkan oleh beban induktif dan kapasitif. Persamaan yang peneliti terdahulu teliti dengan penelitian ini yaitu monitoring pemakaian daya listrik dengan keunggulan dapat memonitoring dan mengontrol melalui *website*.

## 2.2 Meteran Listrik Prabayar

Meteran listrik merupakan jembatan yang memiliki fungsi sebagai penghubung antara pengguna dengan pihak penyedia jasa, meteran listrik biasa digunakan sebagai pusat kendali energi listrik, kemudian untuk melihat total kWh tersisa dan juga biasa digunakan untuk mengisi ulang pulsa atau token yang akan dikonversikan menjadi saldo kWh, kemudian terdapat kata “Prabayar” merupakan

istilah sistem pembayaran yang dilakukan sebelum menerima layanan dari penyedia jasa (Sari, M., & Prasetyo, D. E. 2002). Meteran listrik yang dihubungkan ke banyak rumah tentunya memiliki beberapa tipe yang berbeda, pemerintah menetapkan harga per-kWh yang berbeda berdasarkan kondisi ekonomi dari pengguna. Pada tabel 2.1 terdapat pembagian harga yang disesuaikan dengan kondisi ekonomi dari pengguna.

Tabel 2. 1 Golongan Pelangan PLN per Juli-September 2023

No	Golongan	Daya	Harga per-kWh
1	R-1/TR (Tegangan Rendah)	900 VA	Rp. 1.352,-
2	R-1/TR	1.300 VA	Rp. 1.444,70,-
3	R-1/TR	2.200 VA	Rp. 1.444,70,-
4	R-2/TR	3.500-5.500 VA	Rp. 1.699,53,-
5	R-3/TR	6.600 VA	Rp. 1.699,53,-
6	B-2/TR	6.600 VA-200KVA	Rp. 1.444,70,-
7	B-3/TM (Tegangan Menengah)	200 KVA	Rp. 1.114,74,-
8	I-3/TM	200 KVA	Rp. 1.114,74,-
9	I-4/TT (Tegangan Tinggi)	30.000 KVA	Rp. 996,74,-
10	P-1/TR	6.600 VA-200 KVA	Rp. 1.699,53,-
11	P-2/TM	200 KVA	Rp. 1.522,88,-
12	P-3/TR	Penerangan Jalan Umum	Rp. 1.699,53,-
13	L/TR, TM, TT	-	Rp. 1.644,52,-

Pada tabel 2.1, dapat disimpulkan jika penggunaan daya listrik memiliki harga yang berbeda berdasarkan golongan tersebut, oleh karena itu maka akan diambil harga dengan mayoritas tertinggi, yaitu Rp. 1.699,53,- sebagai tolak ukur terkait pembuatan alat, setelah menentukan tarif yang digunakan, nilai tersebut akan diolah oleh sistem supaya memberikan konversi menjadi Rupiah dari Kwh yang

digunakan. Untuk membangun sistem tersebut tentu memerlukan konsep *Internet of Things* (IoT) dan komponen lainnya.

### 2.3 Segitiga Daya

Daya merupakan energi yang dikeluarkan untuk melakukan usaha. Pada sistem tenaga listrik, daya merupakan jumlah yang digunakan untuk melakukan kerja atau usaha. Daya memiliki satuan Watt, dimana merupakan perkalian dari Tegangan (volt) dan arus (ampere). Daya dinyatakan dalam  $P$ , Tegangan dinyatakan dalam  $V$  dan Arus dinyatakan dalam  $I$ , sehingga besarnya daya dinyatakan :

$$P = V \times I$$

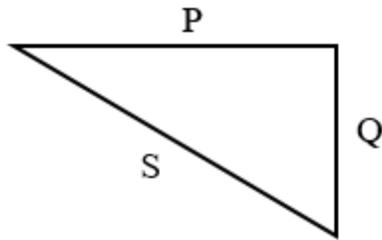
$$P = \text{Volt} \times \text{Ampere} \times \text{Cos } \varphi$$

Segitiga daya adalah sebuah segitiga siku-siku (trigonometri) yang digunakan untuk memudahkan dalam menghitung daya aktif, daya semu dan daya reaktif.

Pada dasarnya daya listrik dibagi menjadi tiga yaitu :

1. Daya nyata / daya aktif dengan satuan W (WATT), dan simbol  $P$
2. Daya semu dengan satuan VA (Volt Ampere), dan simbol  $S$
3. Daya reaktif VAR (Volt Ampere Reaktif), dan simbol  $Q$

Ketiga daya ini terdapat pada listrik satu fasa maupun listrik tiga fasa dan saling berkaitan atau berhubungan satu sama lain. Nilai dari ketiga daya ini juga sangat dipengaruhi oleh cos phi ( $\cos \varphi$ ). Pada listrik 1 fasa 220 Volt PLN biasanya  $\cos \varphi$  ini nilainya ditetapkan 0,8.



Gambar 2. 1 Segitiga Daya

Dimana berlaku hubungan

$$S=V \cdot I$$

$$P=S \cdot \cos \phi$$

$$Q=S \cdot \sin \phi$$

### 2.3.1 Daya Aktif (Daya Nyata/Real Power)

Daya aktif adalah suatu daya yang sesungguhnya terpakai untuk melakukan kerja terhadap beban atau merupakan daya yang sesungguhnya dibutuhkan beban. Daya ini digunakan untuk mengubah suatu energi listrik menjadi bentuk energi lain. Misalkan pada sebuah lampu dimana ada konversi energi listrik menjadi energi cahaya. Satuan dari daya aktif adalah Watt dan daya aktif ini bisa terjadi pada beban induktif maupun beban resistif.

Rumus segitiga daya pada listrik satu phasa

$$P = V \times I \times \cos \phi$$

Rumus segitiga daya pada listrik tiga phasa

$$P = \sqrt{3} \times V \times I \times \cos \phi$$

### 2.3.2 Daya Semu

Merupakan keseluruhan kapasitas daya yang belum terpakai. Kapasitas daya ini yang disediakan oleh PLN dengan satuan VA (Volt Ampere). Pada persamaan segitiga daya bisa dilihat bahwa daya semu ini tidak terdapat cos.

Tabel 2. 2 Daftar Urutan Daya Listrik yang disediakan PLN

No	Daya Terpasang (Volt Ampere)	Arus (Ampere)
1	250	1,2
2	450	2
3	900	4
4	1.300	6
5	2.200	10
6	3.500	16
7	4.400	20
8	5.500	25
9	7.700	35
10	11.000	50

Rumus segitiga daya pada listrik satu phasa

$$S = V \times I$$

Rumus segitiga daya pada listrik tiga phasa

$$S = V \times I \times \sqrt{3}$$

### 2.3.3 Daya Reaktif

Daya reaktif adalah sebuah daya yang terserap untuk pembentukan medan magnet. Daya ini ditimbulkan oleh beban induktif seperti transformator, motor, dan lain lain. Beban induktif disebabkan oleh lilitan kawat atau kumparan yang digunakan untuk membangkitkan medan magnet agar peralatan listrik dapat bekerja dengan baik. Satuan dari Daya Reaktif ini adalah VAR (Volt Ampere Reaktif).

Rumus segitiga daya pada listrik satu phasa

$$Q = V \times I \times \sin \phi$$

Rumus segitiga daya pada listrik tiga phasa

$$Q = V \times I \times \sin \phi \times \sqrt{3}$$

#### 2.4 Modul ESP32 DEVKIT V1

Mikrokontroler ini merupakan pusat atau inti dari sistem yang berbentuk fisik atau *hardware* dari sistem ini, dengan adanya mikrokontroler ini dapat mempermudah untuk mengendalikan sistem secara terpusat dengan lebih baik. Modul ESP32 ini memiliki beberapa varian yang bermacam-macam untuk digunakan sesuai kebutuhan. Pada sistem atau alat yang dikembangkan ini, dibutuhkan Modul ESP32 dengan adanya Wi-Fi pada mikrokontroler yang dapat memberikan kemampuan pada mikrokontroler Modul ESP32 ataupun komponen lainnya untuk melakukan pemantauan dan pengendalian jarak jauh, sehingga pada tahap selanjutnya hanya perlu menguasai pada proses pembangunan (Annisa, T. T. 2022).



Gambar 2. 2 Bentuk fisik Modul ESP32 DEVKIT V1

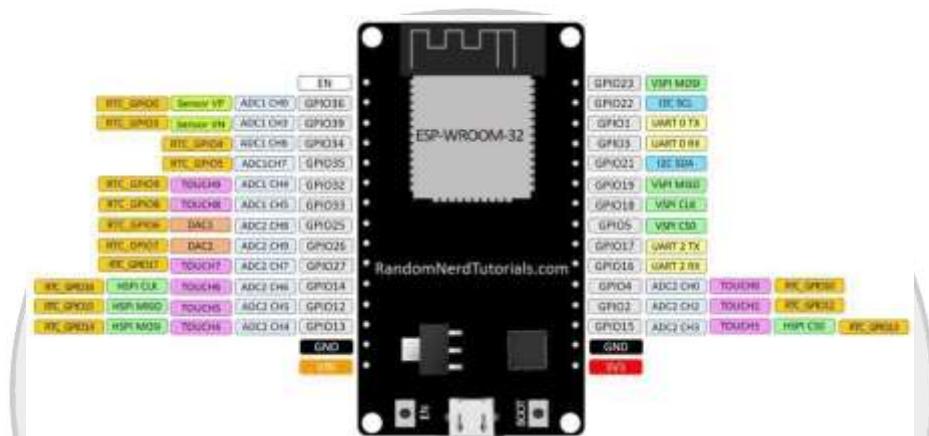
Gambar 2.2 merupakan tampilan dari Modul ESP32, dari mikrokontroler ini terdapat beberapa spesifikasi yang tercantum dalam komponen ini, diantaranya:

1. *Prosesor: Xtensa dual-core (or single-core) 32-bit LX6 microprocessor, operating at 160 or 240 MHz.*
2. *Memori: 520 KB SRAM.*
3. *Wireless connectivity: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE (shares the radio with Wi-Fi).*
4. *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.*
5. *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).*

Kemudian untuk mikrokontroler ESP32 memiliki berbagai macam jenis port yang tersedia di setiap pin, masing-masing pin dipetakan sebagai berikut:

1. Usb port: Micro USb
2. Jumlah pin: 30 meliputi pin tegangan dan GPIO
3. 15 pin ADC (*analog to digital converter*)
4. 3 buah UART Interface

5. 3 buah SPI Interface
  6. 2 buah 12C Interface
  7. 2 buah pin DAC (*digital to analog converter*)
  8. Power *Input*: 5V DC
  9. Ukuran module: 47 x 24 mm



Gambar 2. 3 Pinout ESP32

Penggunaan dari mikrokontroler ESP32 sebagai pusat untuk pengendalian sistem dapat memberikan perintah kepada berbagai komponen yang terhubung dengan mikrokontroler ESP32, masing-masing komponen dapat berjalan secara bersamaan dan data yang dihasilkan dapat lebih *flexible*.

## 2.5 Modul Sensor PZEM-004T

Komponen ini merupakan komponen yang menghubungkan arus listrik dengan alat, komponen ini dapat memberikan data terkait arus listrik ke dalam sistem atau alat yang terhubung. Data yang diteruskan akan terus berkembang dengan menyesuaikan kondisi secara langsung dari arus listrik yang mengalir, kemudian perkembangan data yang diunggah ke mikrokontroler dapat diolah kembali dengan menggunakan beberapa komponen lain untuk melakukan

perubahan data untuk dapat divisualisasikan ke dalam bentuk yang lebih mudah untuk dipahami (Alipudin, A. M. 2018). Komponen ini memiliki peran untuk mengambil data tertentu dari arus listrik, diantaranya adalah *Energy* yang memberi informasi terkait jumlah energi listrik yang telah digunakan, besaran dari *Energy* ini menggunakan *kWh*, kemudian *Power* memberikan informasi terkait daya yang telah digunakan, besaran dari *Power* ini menggunakan *Watt*, kemudian *Voltage* memberikan informasi terkait tegangan yang digunakan, besaran dari *Voltage* ini menggunakan *Volt*, dan yang terakhir *Current* memberikan informasi terkait arus yang digunakan, besaran dari *Current* ini menggunakan *Ampere*. Nilai-nilai yang dihasilkan dari komponen ini dapat diolah secara fleksibel melalui mikrokontroler atau platform lain yang sudah terhubung, kemudian nilai tersebut dapat ditampilkan melalui komponen lain untuk mempermudah dalam visualisasi data.



Gambar 2.4 Modul PZEM-004T

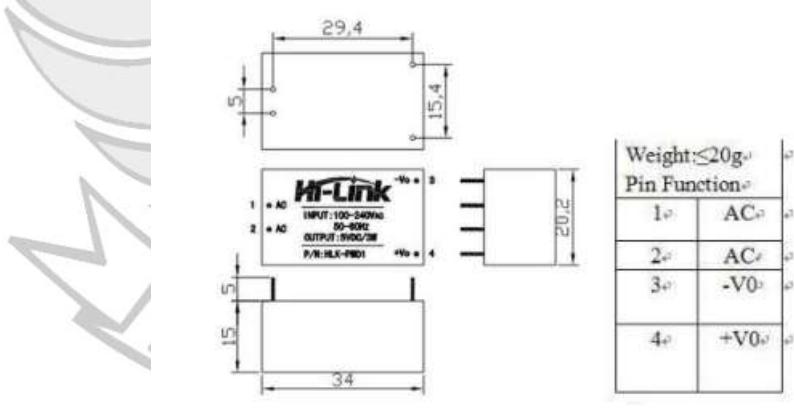
Gambar 2.4 merupakan tampilan dari Modul PZEM-004T, komponen ini terdiri dari dua alat yang memiliki tugas berbeda, yang pertama adalah sensor dengan bentuk seperti lingkaran cincin yang digunakan sebagai tempat untuk memasukkan kabel dengan arus negatif, kemudian yang kedua adalah pengolah data dari sensor yang nantinya menerima data dari sensor kemudian mengirimkan

data ke mikrokontroler untuk diolah kembali. Berikut adalah spesifikasi dari PZEM-004T:

1. *Working voltage: 80 ~ 260VAC*
2. *Rated power: 100A / 22000W*
3. *Working frequency: 45~65Hz*
4. *Measurement accuracy: 1.0*

## 2.6 Modul Power Supply

Hi-Link HLK-PM01 adalah *Power Supply* Mini yang mengkonversi tegangan AC 220V ke tegangan DC 5V 600 mA (3W). Dengan ukuran yang kecil dan *compact*, *Power Supply* ini cocok untuk *project* yang membutuhkan daya yang tidak terlalu besar. Ukuran dan fungsi pin modul Hi-Link Pm01 ditunjukkan gambar 2.5 dan Modul Hi-Link Pm01 ditunjukkan gambar 2.6.



Gambar 2.5 Ukuran dan fungsi pin modul Hi-Link Pm01



Gambar 2. 6 Modul Hi-Link Pm01

Modul Hi-Link mempunyai fitur sebagai berikut.

1. Sangat tipis, dan sangat kecil
2. Semua input tegangan (AC: 90 ~ 264V)
3. tingkat kebisingan rendah
4. Konsumsi daya rendah
5. Pelindungan terhadap beban lebih dan hubung singkat

Karakteristik listrik

1. Nilai tegangan input: 100-240 VAC
2. Rentang tegangan input: 90-264 VAC
3. Tegangan input maksimum:  $\leq 270$  VAC
4. Arus input maksimum:  $\leq 0.2$  A
5. Masukan lonjakan arus:  $\leq 10$  A
6. Masukkan awal yang lambat:  $\leq 50$  ms
7. *Input Low Voltage Efficiency:*  $V_i = 110$  VAC  $\leq$  *Output penuh - load*  $\geq 69\%$
8. *Input Efisiensi Tegangan Tinggi:*  $V_i = 220$  VAC  $\leq$  *Output penuh - load*  $\geq 70\%$

## 2.7 Modul Relay



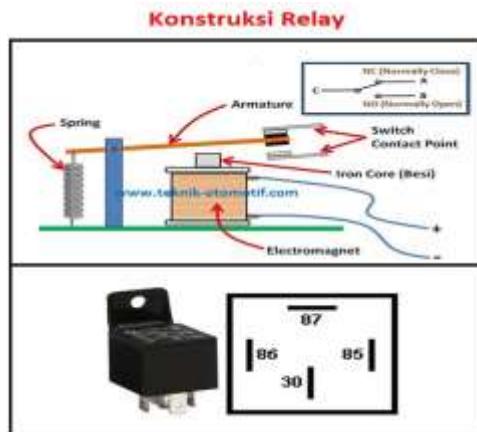
Gambar 2. 7 Bentuk fisik modul relay

Modul relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* yang terdiri dari dua bagian utama yakni electromagnet (*Coil*) dan mekanikal (seperangkat kontak saklar/*switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan relay yang menggunakan *electromagnet* 5V dan 50mA mampu menggerakkan armature relay yang berfungsi sebagai saklarnya untuk menghantarkan listrik 220V 2A.

### 2.6.1 Prinsip kerja Relay

Pada dasarnya relay terdiri dari 4 komponen dasar yaitu:

1. *Electromagnet* (*Coil*)
2. Armature
3. *Switch Contact Point* (Saklar)
4. Spring



Gambar 2. 8 Struktur relay

Titik Kontak (Contact Point) relay terdiri dari dua jenis yaitu:

1. *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *close* (tertutup)
2. *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *open* (terbuka)

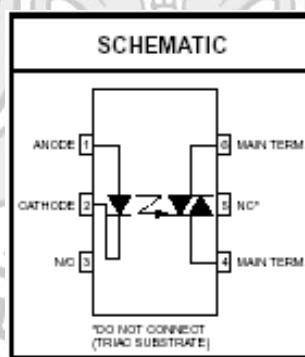
Berdasarkan gambar 2.8 di atas, sebuah inti besi (*Iron Core*) yang dililit oleh sebuah kumparan koil yang berfungsi untuk mengendalikan besi tersebut. Apabila kumparan koil diberikan arus listrik, maka akan timbul gaya *electromagnet* yang kemudian menarik *armature* untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantar arus listrik di posisi barunya (NO). Posisi dimana *armature* tersebut berada sebelumnya (NC) akan menjadi open atau tidak terhubung. Pada saat tidak dialiri arus listrik, *armature* akan kembali lagi ke posisi awal (NC). Koil yang digunakan oleh relay untuk menarik kontak poin ke posisi *close* pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

## 2.8 MOC3021



Gambar 2. 9 Bentuk fisik MOC3021

MOC3021(*optoisolator*) adalah rangkaian yang menurut ilmu optik sebagai alat yang digunakan untuk atau sebagai *driver triac*. Alat ini berisi AlGaAs dioda yang memancarkan infra merah dan cahaya mengaktifkan silicon sebagai *switch* dari dua tempat, yang mana berfungsi seperti suatu triac. Alat ini dibuat atau dirancang untuk antarmuka antara kendali elektronik dan daya *triac* ke kendali beban resistive dan induktif untuk pengoperasian tegangan 115/240 VAC.

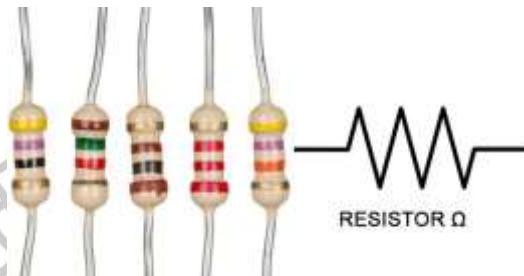


Gambar 2. 10 Schematic MOC3021

Prinsip kerja *optoisolator* adalah apabila diac yang ada didalamnya aktif maka diac akan mengalirkan arus listrik. *Optoisolator* ini aktif apabila *infrared* memberikan input pada diac. *Infrared* akan aktif apabila *infrared* mendapatkan tegangan dan arus sebesar tegangan dan arus minimum yang dibutuhkan untuk

menyalakan *infrared*. *Optoisolator* ini dibuat kompatibel dengan mikrokontroler karena arus yang dibutuhkan untuk mengaktifkan *infrared* dapat dipenuhi oleh mikrokontroler sehingga tidak memerlukan *driver* tambahan.

## 2.9 Resistor



Gambar 2. 11 Bentuk dan simbol resistor

Resistor berfungsi untuk menghambat arus yang akan melewati rangkaian. Satuan hambatan adalah ohm ( $\Omega$ ). Untuk mengetahui nilai hambatan dari sebuah resistor kita bisa membaca kode warna yang ada pada resistor tersebut, atau bisa juga dengan mengukur nilai hambatannya dengan menggunakan alat ukur ohm meter. Resistor mempunyai kode warna yang masing-masing warna mempunyai nilai sendiri-sendiri, seperti ditunjukkan pada tabel 2.3 dan tabel 2.4.

Tabel 2. 3 Kode Warna Resistor

Warna	Nilai
Hitam	0
Coklat	1
Merah	2
Orange	3
Kuning	4
Hijau	5
Biru	6
Ungu	7
Abu-abu	8
Putih	9

Tabel 2. 4 Kode Toleransi Resistor

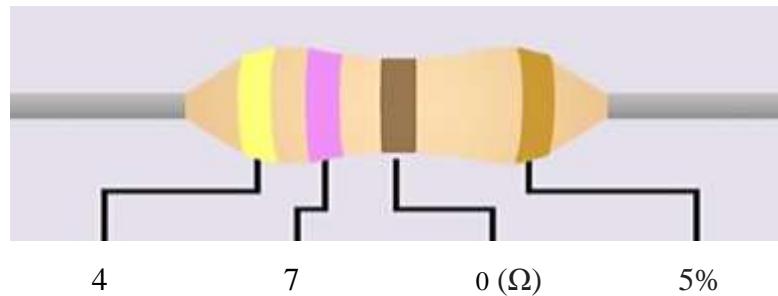
Warna	Nilai
Emas	5%
Perak	10%
Tanpa warna	20%

Adapun cara-cara membaca kode warna pada resistor adalah sebagai berikut :

1. Warna yang pertama adalah angka yang pertama.
2. Warna yang ke dua adalah angka yang kedua.
3. Warna yang ke tiga adalah jumlah nol (0) yang mengikuti angka ke dua.

Contoh perhitungan misalnya ada sebuah resistor dengan kode warna kuning, ungu, coklat, dan emas. Cara mengetahui nilai hambatan pada resistor dengan warna tersebut adalah sebagai berikut :

Warna pertama kuning = 4, maka kita tuliskan angka pertama adalah 4. Warna kedua ungu = 7, maka setelah angka pertama, kita tulis angka kedua yaitu angka 7, sehingga menjadi 47. Selanjutnya warna ketiga coklat = 1 berarti jumlah nol (0) yang mengikuti angka kedua ada 1 (0), sehingga menjadi  $470 = 470 \Omega$ . Warna terakhir emas = 5 % berarti nilai toleransi hambatan sebesar 5 %. Dengan demikian kita mengetahui bahwa resistor tersebut mempunyai nilai hambatan sebesar  $470 \Omega$ , dengan toleransi hambatan sebesar 5 %. Untuk lebih jelasnya dapat dilihat pada gambar 2.12 berikut ini :



Gambar 2. 12 Cara Membaca Kode Warna Resistor

Dalam teori hambatan kita mengenal hukum ohm yang menjelaskan bahwa:

$V = I \times R$

Keterangan :

$V$  = Tegangan ( Volt )

$I$  = Arus ( Ampere )

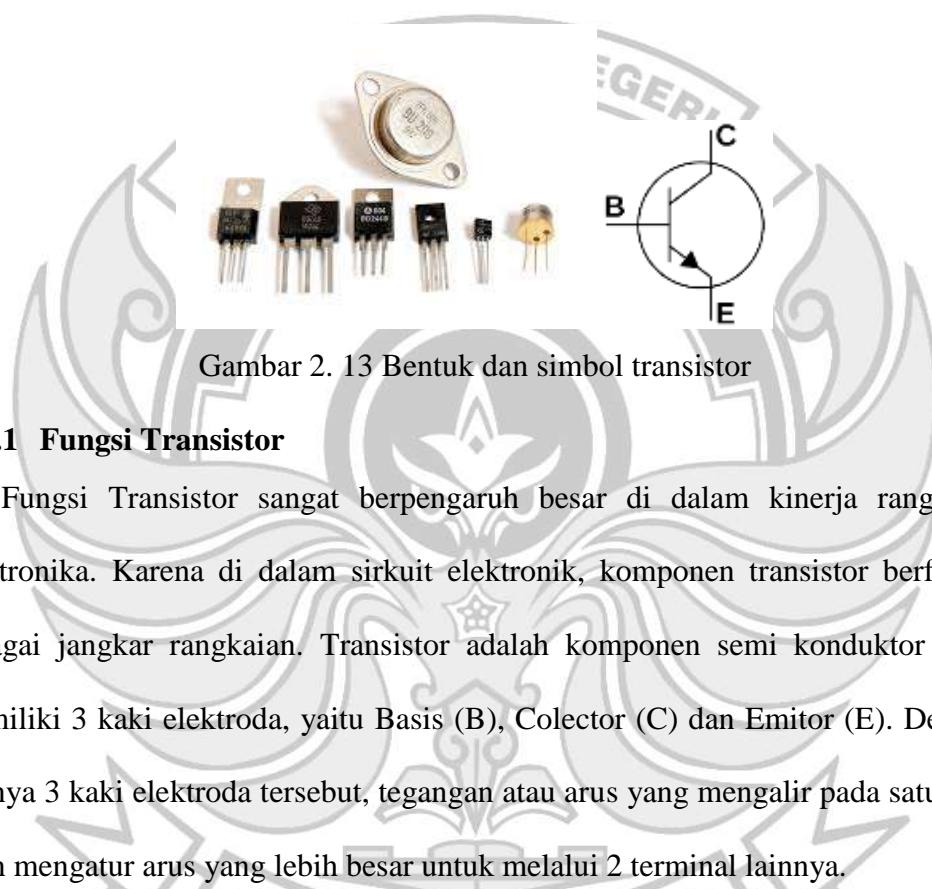
$R$  = Hambatan ( Ohm )

Yakni bahwa nilai tegangan berbanding lurus dengan nilai arus dan berbanding terbalik dengan nilai hambatan.

## 2.10 Transistor

Transistor adalah komponen elektronika semikonduktor yang memiliki 3 kaki elektroda, yaitu Basis (Dasar), Kolektor (Pengumpul) dan Emitor (Pemancar). Komponen ini berfungsi sebagai penguat, pemutus dan penyambung (switching), stabilitasi tegangan, modulasi sinyal dan masih banyak lagi fungsi lainnya. Selain itu, transistor juga dapat digunakan sebagai kran listrik sehingga dapat mengalirkan listrik dengan sangat akurat dan sumber listriknya. Transistor sebenarnya berasal dari kata “transfer” yang berarti pemindahan dan “resistor” yang berarti penghambat. Dari kedua kata tersebut dapat kita simpulkan, pengertian transistor adalah pemindahan atau peralihan bahan setengah penghantar menjadi suhu

tertentu. Transistor pertama kali ditemukan pada tahun 1948 oleh William Shockley, John Barden dan W.H, Brattain. Tetapi, komponen ini mulai digunakan pada tahun 1958. Jenis Transistor terbagi menjadi 2, yaitu transistor tipe P-N-P dan transistor N-P-N. Untuk lebih jelasnya dapat dilihat pada gambar 2.13 berikut ini :



Gambar 2. 13 Bentuk dan simbol transistor

### 2.9.1 Fungsi Transistor

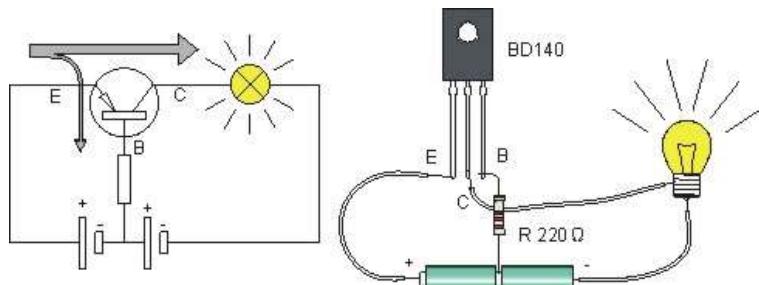
Fungsi Transistor sangat berpengaruh besar di dalam kinerja rangkaian elektronika. Karena di dalam sirkuit elektronik, komponen transistor berfungsi sebagai jangkar rangkaian. Transistor adalah komponen semi konduktor yang memiliki 3 kaki elektroda, yaitu Basis (B), Colector (C) dan Emitor (E). Dengan adanya 3 kaki elektroda tersebut, tegangan atau arus yang mengalir pada satu kaki akan mengatur arus yang lebih besar untuk melalui 2 terminal lainnya.

Fungsi transistor lainnya, sebagai penguat amplifier. Sebagai pemutus dan penyambung (*switching*). Sebagai pengatur stabilitas tegangan. Sebagai peratas arus. Dapat menahan sebagian arus yang mengalir. Menguatkan arus dalam rangkaian. Sebagai pembangkit frekuensi rendah ataupun tinggi. Jika kita lihat dari susunan semi konduktor, Transistor dibedakan lagi menjadi 2 bagian, yaitu Transistor PNP dan Transistor NPN. Untuk dapat membedakan kedua jenis

tersebut, dapat kita lihat dari bentuk arah panah yang terdapat pada kaki emitornya. Pada transistor PNP arah panah akan mengarah ke dalam, sedangkan pada transistor NPN arah panahnya akan mengarah ke luar. Saat ini transistor telah mengalami banyak perkembangan, karena sekarang ini transistor sudah dapat kita gunakan sebagai memory dan dapat memproses sebuah getaran listrik dalam dunia prosesor komputer.

### **2.9.2 Cara Kerja Transistor**

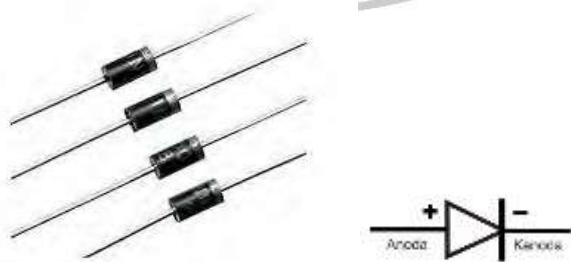
Cara Kerja Transistor cukup menarik untuk dibahas, karena macam dan fungsinya yang unik. Secara harfiah sendiri transistor merupakan gabungan dari dua kata yaitu transfer dan resistor yang dapat diartikan secara bebas sebagai pengalir arus atau pengatur aliran arus. Triode merupakan istilah yang memiliki arti tiga elektroda, dan di dalam resistor sendiri memang memiliki tiga elektroda tersebut, yaitu basis atau dasar, emitor atau pemancar dan kolektor atau pengumpul. Transistor dapat mengalirkan arus listrik atau juga menguatkan tegangan dikarenakan memiliki ketiga elektroda tersebut. Fungsi lain dari transistor adalah sebagai saklar pemutus dan penyambung aliran listrik ketika pada dasar atau basis diberikan arus yang sangat besar. Untuk cara kerja dari transistor sendiri tergantung dari transistor jenis apa yang digunakan. Untuk lebih jelasnya dapat dilihat pada gambar 2.14 berikut ini :



Gambar 2. 14 Cara Kerja Transistor

## 2.11 Dioda

Dioda adalah komponen aktif dua kutub yang pada umumnya bersifat semikonduktor, yang memperbolehkan arus listrik mengalir ke satu arah (kondisi panjar maju) dan menghambat arus dari arah sebaliknya (kondisi panjar mundur). Dioda dapat disamakan sebagai fungsi katup di dalam bidang elektronika. Dioda sebenarnya tidak menunjukkan karakteristik kesearahan yang sempurna, melainkan mempunyai karakteristik hubungan arus dan tegangan kompleks yang tidak linier dan seringkali tergantung pada teknologi atau material yang digunakan serta parameter penggunaan. Beberapa jenis dioda juga mempunyai fungsi yang tidak ditujukan untuk penggunaan penyearah. Dioda yang dipakai pada rangkaian *power supply* pada alat ini adalah dioda tipe 1N4007. Spesifikasi arus : 1A. Input tegangan puncak :1000V. Dibawah ini pada gambar 2.15 adalah bentuk fisik dari Dioda 1N4007.



Gambar 2. 15 Bentuk dan symbol dioda

## 2.11 Pilot Lamp

Sebuah *Pilot lamp* atau dalam bahasa indonesia lampu pilot merupakan sebuah lampu LED yang biasa digunakan sebagai lampu indikator dalam rangkaian sebuah alat atau mesin. *Pilot lamp* tersebut dapat bekerja sebagai mestinya jika dialiri daya AC sebesar 220 VAC dengan toleransi 110 – 240 VAC. Warna yang dihasilkan *Pilot lamp* ini adalah lampu putih. Karena fungsinya sebagai lampu indikator, *Pilot lamp* ini dibuat warna warni sinarnya dengan menambahkan penutup kaca yang berwarna sehingga tampak dari luar berwarna sinar yang dihasilkan. Biasanya warna Pilot lamp ini ada 3 macam merah, hijau, kuning.

Dalam control magnetik alat ini tergolong sebagai sinyal output yang berperan sebagai lampu indikator yang mengindikasikan menunjukan apakah rangkaian itu telah aktif. *Output* dari control magnetik tersebut dihubungkan ke *pilot lamp* ini jika rangkaian tersebut sudah benar maka ketika rangkaian aktif alat ini akan aktif (menyalा). Ketika *Pilot lamp* tersebut menyalा kita dapat mengetahui bahwa rangkaian control magnetik tersebut sudah benar atau aktif. Karena fungsinya sebagai lampu indikator *pilot lamp* ini akan bekerja jika dan hanya jika mendapat aliran listrik. Seperti telah kita ketahui, *pilot lamp* tersebut sangat banyak digunakan, dalam sebuah operation panel bisa kita jumpai beberapa *Pilot lamp* jumlahnya tergantung dari keperluan, dengan warna warna yang dimiliki *pilot lamp* tersebut dapat mengindikasikan indikator yang berbeda. Biasanya lampu warna merah menunjukkan rangkaian tersebut tidak aktif, lampu warna hijau menunjukkan rangkaian itu aktif.



Gambar 2. 16 Pilot Lamp

## 2.12 Internet of things

*Internet of things* merupakan jaringan infrastruktur global yang dinamis yang memiliki kemampuan konfigurasi berdasarkan standar protokol komunikasi dengan memiliki sistem identitas, atribut fisik, karakter kuat, dan antarmuka cerdas yang terhubung dan terintegrasi ke dalam sebuah jaringan informasi. Sederhananya IoT dapat menyatukan dunia virtual teknologi informasi dengan benda riil di dunia nyata. IoT dapat bekerja dengan mengambil data dari pembacaan sensor yang diletakkan pada benda di dunia nyata kemudian dikirimkan ke server. Adapun sensor-sensor yang bisa terhubung ke jaringan interner seperti sensor tegangan, arus, RFID, atau sensor lainnya yang layaknya seperti indera manusia seperti sensor cahaya, gerak, tekanan, suara, dan lain-lain. Pada sistem IoT, benda yang terhubung sensor diberikan kemampuan untuk melakukan reaksi yang diperintahkan oleh server melalui kontroler.



Gambar 2. 17 Aliran informasi dengan konsep Internet of Things

Konsep IoT mencangkup 3 elemen utama yaitu : Koneksi internet, benda fisik atau nyata yang telah diintegrasikan pada modul sensor, dan pusat data pada server untuk menyimpan data atau informasi dari aplikasi. Benda-benda yang terkoneksi ke jaringan internet akan menghimpun data yang kemudian terkumpul yang kemudian diolah, setelah diolah akan dianalisa baik oleh instansi pemerintah, perusahaan terkait, maupun instansi lainnya yang kemudian dimanfaatkan bagi kepentingan masing-masing.

### 2.13 Framework Laravel



Gambar 2. 18 Framework Laravel

*Framework* adalah struktur konseptual dasar yang berisi kumpulan fungsi untuk tujuan tertentu yang sudah siap untuk digunakan, sehingga pembuatan

aplikasi dapat dilakukan dengan lebih cepat karena kode programnya tidak dibuat dari awal.

Beberapa alasan dari digunakannya *framework* dalam membuat aplikasi adalah sebagai berikut.

- a. Aplikasi akan memiliki standar pemrograman yang universal.
- b. Menghindari *repetitive work*.
- c. Memudahkan dalam *team work*.
- d. Memudahkan dalam *maintenance* dan pengembangan aplikasi di masa mendatang.
- e. Hemat waktu dan biaya.

Laravel merupakan *Framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti *framework* PHP lainnya, Laravel dibangun dengan basis MVC (Model-View-Controller). Laravel dilengkapi command line tool yang bernama “Artisan” yang bisa digunakan untuk *packging bundle* dan instalasi bundle. *Framework* Laravel dibuat oleh Taylor Otwell, proyek Laravel dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang up-to-date dengan versi PHP. Mengembangkan *framewrok* yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut, Otwell membuat sendiri *framework* dengan nama Laravel. Oleh karena itu Laravel mensyaratkan PHP versi 5.3 keatas. (Rohman, 2014). *Framewok* Laravel juga memiliki beberapa keunggulan sebagai berikut:

- a. Menggunakan *Command Line Interface* (CLI) Artisan.
- b. Menggunakan *package manager* PHP *Composer*.
- c. Penulisan kode program lebih singkat, mudah dimengerti, dan ekspresif.

Kemudian untuk cara instalasi *framework* Laravel dapat dilakukan dengan 3 cara yaitu:

Melalui Installer Laravel.

- a. Menggunakan Composer dengan mengetikkan perintah `create-project`.
- b. Download source code Laravel secara lengkap melalui GitHub dengan alamat <https://github.com/laravel/laravel/>.

Untuk menggunakan Laravel versi 8 komputer atau server yang digunakan harus memenuhi persyaratan sebagai berikut:

- a. PHP  $\geq 8.1$
- b. OpenSSL PHP Extension
- c. PDO PHP Extension
- d. Mbstring PHP Extension
- e. Tokenizer PHP Extension
- f. XML PHP Extension

Fitur *framework* Laravel yang ditekankan pada penelitian ini adalah Blade, Migration, Eloquent ORM, Resource Controller, dan Middleware. Berikut adalah penjelasan mengenai lima fitur tersebut:

a. *Blade*

adalah template engine. Pada dasarnya *Blade* adalah *view* namun dengan menggunakan *Blade* akan mempermudah untuk mengatur tampilan website

dan menampilkan data. Cara untuk membuat file *view* menjadi file *Blade* adalah dengan menambahkan ekstensi *.blade.php* pada file *view*. Dan cara untuk memanggil file *Blade* sama dengan cara untuk memanggil file *view* biasa.

*b. Migration*

adalah fitur yang menyediakan cara baru untuk membuat *database*. Dengan menggunakan *migration* cara membuat *database* melalui *Command Line Interface (CLI)* *database* atau dengan menggunakan aplikasi *database manager* digantikan dengan menggunakan *class*. Tahapan menggunakan *migration* adalah membuat *class* kemudian melakukan perintah *migrate* melalui *Command Line Interface (CLI)* *artisan*.

Keuntungan menggunakan *migration* adalah *class* yang dibuat bisa dipakai untuk membuat *database* pada berbagai macam *Relation Database Management System (RDBMS)* yang didukung oleh Laravel. Sebagai contoh misalnya aplikasi yang digunakan selama ini menggunakan *database MySQL*, kemudian karena alasan pengembangan aplikasi maka akan dilakukan penggantian *database* ke PostgreSQL. Dalam proses penggantian tersebut tidak perlu membuat *class* lagi, tinggal melakukan perintah *migrate* melalui *Command Line Interface (CLI)* *artisan*.

Keuntungan lain dari menggunakan *migration* adalah semua perubahan yang dilakukan pada *database* akan disimpan pada suatu tabel. Sehingga bisa dilakukan pembatalan (*rollback*) pada *database* jika melakukan perubahan yang tidak benar.

### c. *Eloquent ORM*

adalah implementasi dari *ActiveRecord* yang digunakan untuk mengatur relasi antar tabel di *database*. Pada *Eloquent ORM* tabel direpresentasikan dalam bentuk kelas dan data yang tersimpan didalam tabel direpresentasikan dalam bentuk objek. Relasi yang dapat diatur menggunakan *Eloquent ORM* adalah sebagai berikut:

1. *One-to-One* yaitu relasi satu ke satu. Pada relasi ini digunakan *method hasOne* dan *belongsTo*.
2. *One-to-Many* yaitu relasi satu ke banyak. Pada relasi ini digunakan *method hasMany* dan *belongsTo*.
3. *Many-to-One* yaitu relasi banyak ke satu. Pada relasi ini digunakan *method belongsTo* dan *hasMany*.
4. *Many-to-Many* yaitu relasi banyak ke banyak. Pada relasi ini digunakan *method belongsToMany*.

### d. *Resource Controller*

adalah fitur yang digunakan untuk mempercepat pembuatan *controller*. Sebagai contoh misalnya ada *controller* yang menangani semua *HTTP request* terhadap data sensor.

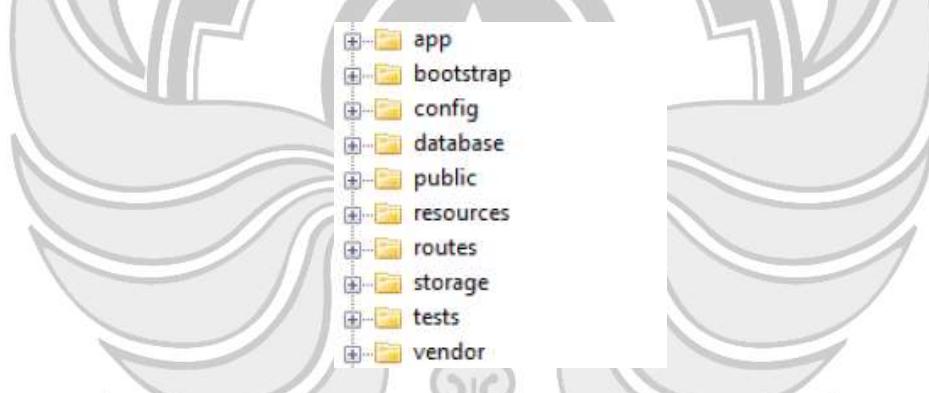
### e. *Middleware*

adalah fitur yang menyediakan mekanisme untuk memfilter *HTTP request* yang masuk ke aplikasi. Laravel memiliki beberapa *Middleware* yaitu *Authenticate*, *EncryptCookies*, *RedirectIfAuthenticated*, dan *VerifyCsrfToken*.

Sebagai pembahasan akan dibahas *Middleware Authenticate*. *Middleware* tersebut akan memeriksa apakah *user* sudah *login* atau belum. Jika *user* sudah *login* maka *request* akan dilanjutkan ke halaman yang dikehendaki oleh *user*. Tetapi jika *user* belum *login* maka *Middleware Authenticate* akan mengarahkan *user* ke halaman *login*.

Jika *Middleware* yang sudah ada pada Laravel kurang sesuai dengan kebutuhan ataupun tidak sesuai dengan kebutuhan maka dapat dibuat sendiri *Middleware* yang sesuai dengan kebutuhan.

Struktur folder dari *framework* Laravel 8 yang masih *default* dapat dilihat pada gambar 2.19



Gambar 2. 19 Struktur folder dari Laravel 8

1. Folder *app* adalah folder yang berisi kode program inti dari aplikasi yang akan dibuat. *Model* dan *controller* tersimpan pada folder ini.
2. Folder *bootstrap* adalah folder yang berisi konfigurasi *autoloading* dan terdapat juga folder *cache* yang menyimpan file-file yang dihasilkan secara otomatis oleh Laravel untuk mengoptimasi kinerja dari sistem yang dihasilkan.
3. Folder *config* adalah folder yang berisi semua file konfigurasi aplikasi.

4. Folder *database* adalah folder yang berisi file *database migration* dan *seeds*.
5. Folder *public* adalah folder yang berisi file *index.php*. File tersebut digunakan sebagai *entry point* untuk menangani semua *request* yang masuk ke aplikasi. Pada folder ini juga dapat disimpan beberapa aset dari aplikasi seperti gambar, *JavaScript*, dan *CSS*.
6. Folder *resources* adalah folder yang berisi file *view* dari aplikasi yang dibuat. Selain itu terdapat juga file *language* yang digunakan aplikasi.
7. Folder *routes* adalah folder yang berisi file yang digunakan untuk mendefinisikan semua *route* ke aplikasi. Secara *default* ada tiga file *route* yang disediakan Laravel yaitu *api.php*, *console.php*, dan *web.php*.
8. Folder *storage* adalah folder yang berisi *template Blade* yang dikompilasi, file *session*, file *cache*, dan file lainnya yang dihasilkan secara otomatis oleh Laravel.
9. Folder *tests* adalah folder yang berisi semua file *test* yang dibuat untuk aplikasi.
10. Folder *vendor* adalah folder yang menyimpan semua *library* yang digunakan.

## 2.14 MYSQL

Menurut Nugroho (2013), “MySQL adalah software atau program Database Server”. Sedangkan SQL adalah bahasa pemrogramannya, bahasa permintaan (query) dalam database server termasuk dalam MySQL itu sendiri. SQL juga dipakai dalam software database server lain, seperti SQL Server, Oracle, PostgreSQL dan lainnya. Saat ini pengembangan MySQL berada dibawah naungan

perusahaan MySQL AB. Sebagai software DBMS, MySQL memiliki sejumlah fitur seperti:

- a. Multiplatform MySQL tersedia pada beberapa platform (Windows, Linux, Unix, dan lain-lain).
- b. Andal, cepat, dan mudah digunakan MySQL tergolong sebagai database server yang andal, Dapat menangani database yang besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses database, dan sekaligus mudah untuk digunakan.
- c. Jaminan keamanan akses MySQL mendukung pengamanan database dengan berbagai kriteria pengaksesan.
- d. Dukungan SQL, SQL merupakan standar dalam pengaksesan database relasional. Pengetahuan SQL akan memudahkan siapa pun menggunakan MySQL.

## BAB V PENUTUP

### 5.1 Kesimpulan

Setelah menyelesaikan seluruh tahapan pelaksanaan pembuatan alat hingga melakukan pengambilan data, maka kesimpulan yang dapat diambil sebagai berikut:

1. Sistem monitoring dapat bekerja dengan baik dimana mikrokontroler yang digunakan terhubung ke sistem *real-time* pada *website* hosting database. Modul ESP32 ini memerlukan internet untuk dapat terhubung ke *server database*, modul ini juga menerima data modul sensor PZEM-004T untuk membaca tegangan, arus, dan daya listrik.
2. Sistem *real-time* monitoring dibuat dengan modul ESP32 yang mengirim data ke *database webserver* monitoringlistrik.online, jika data dipicu oleh modul ESP32 maka *server* akan meminta data dari modul sensor PZEM-004T. Setelah meminta data, modul ESP32 mengirim data *String* ke *webserver* dan akan disimpan ke *database*, *dashboard* pada *website* akan terupdate otomatis setelah terdapat data baru yang masuk.

### 5.2 Saran

Dari hasil pengujian dan pembacaan terdapat beberapa saran yang dapat diaplikasikan untuk pengembangan alat ini, maka dapat diberikan saran sebagai berikut :

1. Sistem dapat dikembangkan sehingga memungkinkan untuk menampilkan data penggunaan listrik yang ada di meteran listrik.

2. Sistem dapat dikembangkan untuk meningkatkan kinerja yang stabil dalam jangka waktu yang panjang.



## DAFTAR PUSTAKA

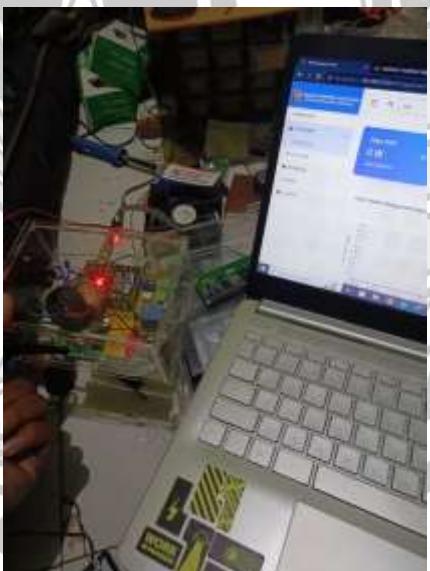
- Amir, A., Marwanto, A., & Nugroho, D. (2018). Rancang Bangun Purwarupa Alat Monitoring Dan Kontrol Beban Satu Fasa Berbasis Iot (Internet of Things). *Transmisi*, 20(1), 29. <https://doi.org/10.14710/transmisi.20.1.29-33>
- Anantama, A., Apriyantina, A., Samsugi, S., & Rossi, F. (2020). Alat Pantau Jumlah Pemakaian Daya Listrik Pada Alat Elektronik Berbasis Arduino Uno. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1), 29. <https://doi.org/10.33365/jtst.v1i1.712>
- Aprilianto, A. (2021). Implementasi Framework Laravel pada Rancang Bangun Website IAKN Palangka Raya dengan Metode Prototype. *Jurnal Sains Komputer Dan Teknologi Informasi*, 3(2), 87–96. <https://doi.org/10.33084/jsakti.v3i2.2238>
- Ardianto, F., Eliza, E., & Saputra, R. (2019). Pendekripsi Pemakaian Beban Listrik Rumah Tangga. *Jurnal Surya Energy*, 4(1), 338–344. <https://doi.org/10.32502/jse.v4i1.1961>
- Berlianti, R., & Fibriyanti, F. (2020). Perancangan Alat Pengontrolan Beban Listrik Satu Phasa Jarak Jauh Menggunakan Aplikasi Blynk Berbasis Arduino Mega. *SainETIn*, 5(1), 17–26. <http://journal.unilak.ac.id/index.php/SainETIn/article/view/6398>
- Danang Danang, & Siswanto Siswanto. (2019). Konsep Pengendali Lampu Penerangan Rumah Dari Jarak Jauh Menggunakan Gelombang Frekuensi. *Elkom : Jurnal Elektronika Dan Komputer*, 12(2), 34–49. <https://doi.org/10.51903/elkom.v12i2.99>

- Desma Aipina, & Harry Witriyono. (2022). Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web. *Jurnal Media Infotama*, 18(1), 36–42.
- Farisy, S. Al, Siswono, I. H., & Pendahuluan, I. (n.d.). *Rumah Berbasis Nodemcu Esp8266 Menggunakan Aplikasi*.
- Fikriyuda, G., & Rachman, M. I. (2015). *Tugas Akhir-Te 145561*.
- Furqon, A., Prasetijo, A. B., & ... (2019). Rancang Bangun Sistem Monitoring dan Kendali Daya Listrik pada Rumah Kos Menggunakan NodeMCU dan Firebase Berbasis Android. *Techné: Jurnal Ilmiah* ....  
<http://ojs.jurnaltechne.org/index.php/techne/article/view/202>
- Hanur, M. F. A. (2016). *Rancang Bangun Alat Pemutus KWH Meter Sebagai Proteksi Berbasis Arduino*. 54.
- Industri, F. T., & Indonesia, U. I. (2022). *Sistem Monitoring Meteran Listrik*.
- Manurung, B. D. (2021). RANCANG BANGUN ALAT PEMANTAU DAN PENGENDALI DAYA LISTRIK BERBASIS INTERNET OF THING. ....  
Karya Ilmiah Mahasiswa Fakultas Sains Dan ....  
<https://journal.pancabudi.ac.id/index.php/fastek/article/download/1832/1672>
- Pramono, A., Fakhrurozi, R. F., Bangkit, J. P., & Lâ, T. J. (2020). Prototype Sistem Kontrol Instalasi Listrik Rumah Tinggal Menggunakan Android Berbasis Arduino Mega. *Iteks*, 1, 43–50. <https://www.ejournal.stt-wiworotomo.ac.id/index.php/iteks/article/view/288>
- Rdiansyah, A. (2020). Monitoring Daya Listrik Berbasis IoT (Internet of Things). *Universitas Islam Indonesia*. <https://dspace.uii.ac.id/handle/123456789/23561>

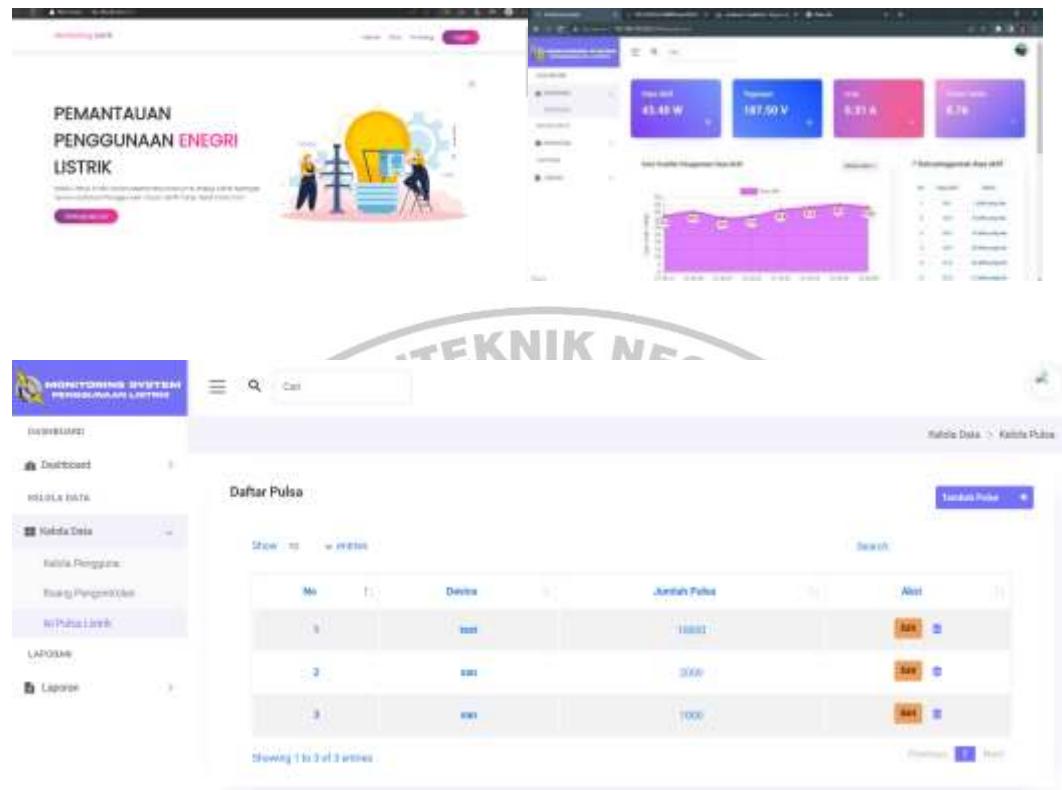
- Reyval, D. (2022). Elektronika Dasar Transistor Dan Cara Kerjanya. *Jurnal Portal Data*, 2(4), 1–9. <http://portaldatal.org/index.php/portaldatal/article/view/121>
- Risanty, R. D., & Arianto, L. (2017). Rancang Bangun Sistem Pengendalian Listrik Ruangan Dengan Menggunakan Atmega 328 Dan Sms Gateway Sebagai Media Informasi. *JUST IT: Jurnal Sistem Informasi, Teknologi* ....  
<https://jurnal.umj.ac.id/index.php/just-it/article/view/1070>
- Saepudin Nirwan, & Hafidz MS. (2020). 871-Article Text-1481-1-10-20200824. *Rancang Bangun Aplikasi Untuk Prototipe Sistem Monitoring Konsumsi Energi Listrik Pada Peralatan Elektronik Berbasis Pzem-004T*, 12(2), 23–24.
- Suryono, S., & Supriyati, S. (2017). RANCANG BANGUN PENGONTROL LAMPU LISTRIK MENGGUNAKAN ANDROID DILENGKAPI DENGAN SAKLAR MANUAL. *Orbith: Majalah Ilmiah Pengembangan* ....  
<https://jurnal.polines.ac.id/index.php/orbith/article/view/957>
- Yulistiyanto, A. (2019). *SISTEM MONITORING ENERGI LISTRIK PADA MOTOR LISTRIK OVEN* 7 PT. RECIKTT BENCKISER MENGGUNAKAN IoT (INTERNET of THINGS).  
<http://repository.unissula.ac.id/id/eprint/14767%0Ahttp://repository.unissula.ac.id/14767/6/Lampiran.pdf>

## LAMPIRAN

Lampiran 1 Proses Pebuatan

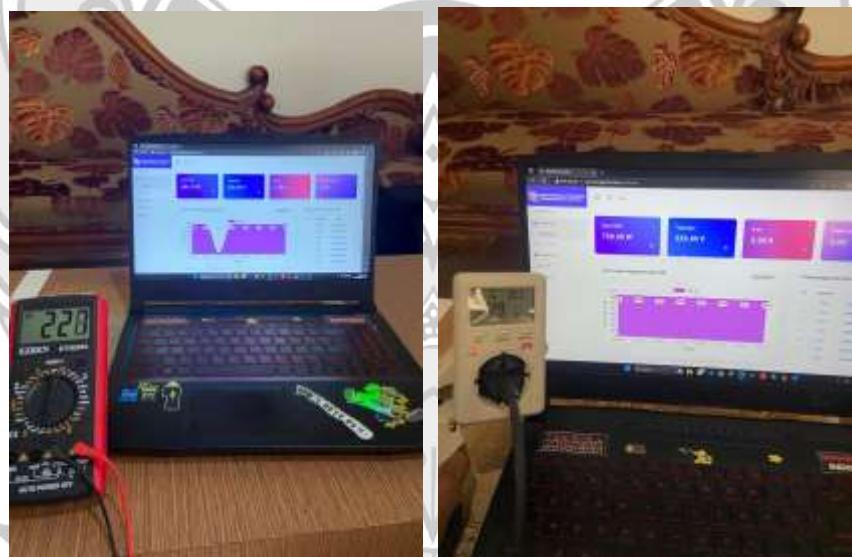
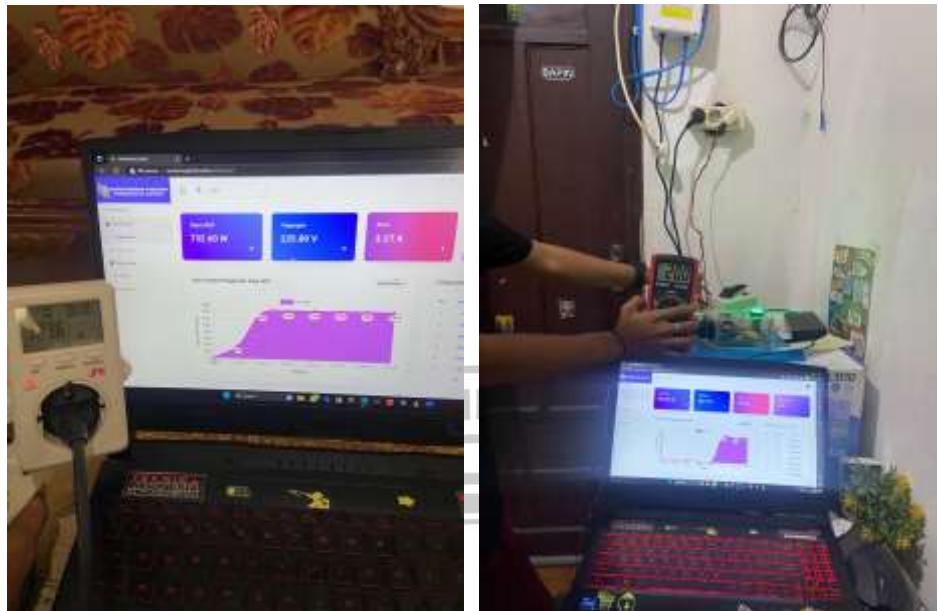


## Lampiran 2 Web Monitoring



### Lampiran 3 Pengambilan data





Lampiran 4 Daftar Waktu Kegiatan

Rencana Kegiatan	Jan				Feb				Mar				Apr				Mei				Jun				Jul				Agu				Sep				
	Minggu Ke-				Minggu Ke-				Minggu Ke-				Minggu Ke-				Minggu Ke-				Minggu Ke-				Minggu Ke-				Minggu Ke-								
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4					
Penyusunan Proposal																																					
Seminar Proposal																																					
Pembuatan Alat																																					
Pengujian Alat																																					
Penyusunan Laporan																																					
Seminar Hasil																																					

## Lampiran 5 Listing Program Menggunakan Arduino IDE

/////////// Header //////////

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <EEPROM.h>      // Simpan data ke eeprom
#include <PZEM004Tv30.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#define EEPROM_SIZE 4
PZEM004Tv30 pzem(Serial2, 16, 17);
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);
///////////
unsigned long wktkontrol = 0;
unsigned long interval_kontrol = 1000;
bool kontrolStat = false;
unsigned long previousMillis = 0;
const long interval = 3000;
unsigned long befMillis = 0;
const long intv = 1000;
unsigned long pulMillis = 0;
const long int_pul = 1000;
unsigned long wktsend = 0;
unsigned long interval_send = 7000;
bool sendStat = false;
const char* ssid = " "; //Nama WiFi
const char* password = " "; //Password WiFi
```

```
const char* host = "monitoringlistrik.online";//isi lokal
String url1;
long lastMillis;
int beban,modeNya;
float kwhSisa,samakanSisa,token,teg_rms,arus_rms,faktor_daya,daya_aktif;
float pulsa;
unsigned long bot_lasttime;
const unsigned long BOT_MTBS = 1000;
String userId=" ";
int kodeBeban;
void setup() {
  Serial.begin(9600);
  pinMode(32,OUTPUT);
  pinMode(13,OUTPUT);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
```

```

EEPROM.begin(EEPROM_SIZE);
storedx = EEPROM.read(0);
storedpul = EEPROM.read(3);
Serial.print("kWh Terakhir: ");
Serial.println(storedx);
Serial.print("Pulsa Terakhir: ");
Serial.println(storedpul);
// pzem.resetEnergy();
}
void kirim(){
Serial.print("connecting to ");
Serial.print(host);
WiFiClient data_kirim;
if (!data_kirim.connect(host, 80 /*httpPort*/)) {
Serial.println("connection failed");
return;
}
//alamat url yg di request di laravel, url harus sesuai
String url = "/esp/"
+ String(teg_rms) + "/"
+ String(arus_rms) + "/"
+ String(faktor_daya) + "/"
+ String(daya_aktif) + "/"
+ String(enr, 7) + "/"
+ String(ener, 7);
Serial.print(" \nRequesting URL: ");
Serial.println(url);
data_kirim.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host +
"\r\n" + "Connection: close\r\n\r\n");
unsigned long timeout = millis();

```

```
while (data_kirim.available() == 0)
{
    if (millis() - timeout > 7000)
    {
        Serial.println(">>> Client Timeout !");
        data_kirim.stop(); //Stop client
        return;
    }
}

void loop()
{
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float energy = pzem.energy();
    float frequency = pzem.frequency();
    float pf = pzem.pf();

    if(isnan(voltage))    Serial.println("Error reading voltage");
    else if (isnan(current)) Serial.println("Error reading current");
    else if (isnan(power)) Serial.println("Error reading power");
    else if (isnan(energy)) Serial.println("Error reading energy");
    else if (isnan(frequency))Serial.println("Error reading frequency");
    else if (isnan(pf))    Serial.println("Error reading power factor");
    else {
        teg_rms=voltage;
        arus_rms=current;
        daya_aktif=power;
        faktor_daya=pf;
        energi=energy;
        ener=energi;
        kwhSisa=(token/1444)-energi;
    }
}
```

```

    }

    if(modeNya==1){ // Mode Otomatis

        if(kwhSisa>0){

            digitalWrite(32,1); // beban on

            digitalWrite(13,1);

        }

        else {

            digitalWrite(32,0); // Beban off

            digitalWrite(13,0);

        }

    }

    else { // mode Manual

        digitalWrite(32,beban);

        digitalWrite(13,beban);

    }

    if(beban==0 && kodeBeban==0){ // Kirim pesan ke telegram

        kodeBeban=1;

        bot.sendMessage(userId, "Beban OFF", "");

    }

    if(beban==1 && kodeBeban==1){ // Kirim pesan ke telegram

        kodeBeban=0;

        bot.sendMessage(userId, "Beban ON", "");

    }

    if(kwhSisa>8 &&kwhSisa<=10 && kwhSisa!=samakanSisa){

        samakanSisa=kwhSisa;

        bot.sendMessage(userId, "Sisa Listrik sebesar 10kwh", "");

    }

    if(kwhSisa<1 && (kwhSisa!=samakanSisa)){

        samakanSisa=kwhSisa;
    }
}

```

```

bot.sendMessage(userId, "Pulsa Listrik Anda Habis", "");
}

Serial.print("Sisa Pulsa = ");
Serial.println(kwhSisa);
Serial.print("energi = ");
Serial.println(energi, 7);

if((millis()-lastMillis)>5000){
    lastMillis=millis();
    kirim();
    kontrol();
}
bacaPesan();
}

void postData() {
    int httpCode1; // variable used to get the response http code after calling api
    HTTPClient httpo; // Declare object of class HTTPClient
    httpo.begin(url1); // begin the HTTPClient object with generated url
    httpCode1 = httpo.POST(url1); // Finally Post the URL with this function and it
    will store the http code
    if (httpCode1 == 200) // Check if the response http code is 200
    {
        Serial.println("Sent ok."); // print message sent ok message
    }
    else // if response HTTP code is not 200 it means there is some
    error.
}

httpo.end(); // After calling API end the HTTP client object.
}

void handleNewMessages(int numNewMessages){
    Serial.print("handleNewMessages ");
}

```

```

Serial.println(numNewMessages);
for (int i = 0; i < numNewMessages; i++)
{
    String chat_id = bot.messages[i].chat_id;
    String text = bot.messages[i].text;
    if (text == "cek pulsa"){
        char ruangan[30];
        long int puluhanA=kwhSisa;
        long int satuanA=kwhSisa*100-puluhanA*100;
        sprintf(ruangan,"Sisa Pulsa %li.%02likwh",puluhanA,satuanA);
        bot.sendMessage(chat_id, ruangan);
    }
}
void bacaPesan(){
    if (millis() - bot_lasttime > BOT_MTBS)
    {
        int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        while (numNewMessages)
        {
            Serial.println("got response");
            handleNewMessages(numNewMessages);
            numNewMessages = bot.getUpdates(bot.last_message_received + 1);
        }
        bot_lasttime = millis();
    }
}

```