

Sistem Monitoring Faktor Daya Berbasis Internet of Things dan Android

Internet of Things-based Power Factor Monitoring System using Android

Lutfi Hakim¹, Sepyan Purnama Kristanto², Subono³, Faris Bagas Dinan⁴

^{1,2,3}Teknik Informatika, Politeknik Negeri Banyuwangi

E-mail: ¹lutfi@poliwangi.ac.id

Abstrak

Faktor daya memiliki peran penting yang jika diabaikan dapat berpotensi konsumsi daya listrik menjadi lebih besar. Faktor yang mempengaruhi perbandingan daya aktif dan daya semu berasal dari seberapa besar daya reaktif yang dihasilkan. Semakin besar daya reaktif yang dihasilkan maka semakin besar selisih antara daya aktif dan daya semu sehingga menyebabkan faktor daya semakin rendah. Dalam implementasinya, instalasi jaringan listrik memiliki jaringan yang sangat kompleks. Jika hal tersebut kurang dikelola dengan baik, maka dapat mengakibatkan konsumsi daya listrik yang kurang efektif dan efisien. Oleh karena itu, perlu dilakukan monitoring agar konsumsi factor daya tetap stabil. Pada penelitian ini dikembangkan sistem monitoring factor daya berbasis internet of things dengan menggunakan sensor PZEM sebagai pembaca arus dan tegangan serta Arduino Mega sebagai pengontrol. Hasil pembacaan sensor ditampilkan pada aplikasi android yang dapat dikontrol dan direkapitulasi hasil monitoring arus listrik, tegangan, daya dan factor dayanya. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat berjalan dengan baik. Hasil pengujian pada perangkat keras menunjukkan perbedaan nilai $\cos\phi$ yang tidak terlalu signifikan pada pengujian dengan beban setrika dan Generator. Sedangkan hasil pengujian pada aplikasi android menunjukkan performa yang baik dengan hasil pengujian menunjukkan semua fitur yang dikembangkan dinyatakan valid.

Kata kunci: sistem monitoring, $\cos\phi$, factor daya, daya reaktif, daya semu.

Abstract

The power factor has an important role. If it is ignored, it can potentially increase the consumption of electric power. The factor that affects the active and apparent powers come from how much reactive power is generated. The greater the reactive power generated, the greater the difference between the active and the apparent powers, causing the power factor to be lower. In its implementation, electrical installations have a very complex network. If this is not managed properly, it can result in less effective and efficient electricity consumption. Therefore, it is necessary to monitor so that power factor consumption remains stable. In this study, an internet of things-based power factor monitoring system was developed using the PZEM sensor as a current and voltage reader and also Arduino Mega as a controller. The results of sensor readings are displayed on an android application that can be controlled and recapitulated the results of monitoring electric current, voltage, power and power factor. The results showed that the developed system could run well. The test results on the hardware show that the difference in $\cos\phi$ values is not too significant in the test with iron and generator loads. While the test results on the android application show good performance with the test results showing all the features developed are declared valid.

Keywords: monitoring system, $\cos\phi$, power factor, reactive power, apparent power.

1. PENDAHULUAN

Sistem jaringan listrik yang terpasang di Indonesia merupakan jaringan listrik tiga fasa yang disalurkan oleh produsen listrik, dalam hal ini PLN, ke konsumen listrik yakni rumah tangga, industri, bisnis, sekolah maupun perkantoran. Penggunaan listrik tiga fasa dan dalam jumlah yang besar pada umumnya digunakan pada bidang usaha, industri maupun pabrik untuk melayani beban-beban seperti motor listrik, transformator, lampu TL dan peralatan listrik lainnya. Beban-beban tersebut bersifat induktif [1]. Bertambahnya beban yang bersifat induktif membutuhkan daya reaktif yang sangat besar sehingga daya listrik yang tersalurkan menjadi lebih besar. Penyaluran daya listrik yang besar ini membuat arus yang mengalir pada saluran ini menjadi bertambah dan menyebabkan factor daya menjadi rendah. Energi listrik yang dihasilkan akan disalurkan ke pusat-pusat beban melalui jaringan transmisi dan distribusi [2]. Pada listrik tiga fasa, semakin banyak beban yang bersifat induktif maka akan mempengaruhi konsumsi daya listrik yang semakin bertambah dan menyebabkan factor daya menurun.

Faktor daya atau *cosphi* merupakan perbandingan antara daya aktif (watt) dengan daya semu/total (VA), atau *cosinus* sudut antara daya aktif dan daya semua. Faktor daya memiliki peran penting yang jika diabaikan akan mengakibatkan konsumsi daya listrik menjadi lebih besar. Daya reaktif yang tinggi akan meningkatkan sudut *cosinus* [3]. Sebagai hasilnya, factor daya akan menjadi lebih rendah. Faktor daya lebih kecil (sama dengan satu) menggambarkan sudut *phase* antara daya aktif dan semu. Faktor daya yang rendah akan merugikan karena mengakibatkan arus beban tinggi. Factor yang mempengaruhi perbandingan daya aktif dan daya semu berasal dari seberapa besar daya reaktif yang dihasilkan [4]. Daya reaktif berasal dari peralatan listrik yang bersifat induksi seperti motor listrik dan ballast pada lampu TL. Semakin besar daya reaktif yang dihasilkan maka semakin besar selisih antara daya aktif dan daya semu sehingga menyebabkan factor daya semakin rendah.

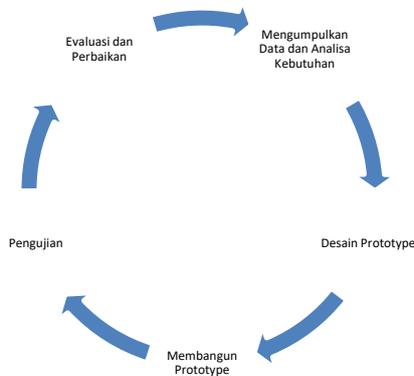
Instalasi listrik di sebuah perusahaan, institusi dan sejenisnya pasti memiliki jaringan listrik yang sangat kompleks. Jika kurang dikelola dengan baik, maka akan beresiko mengakibatkan kerugian dan konsumsi daya listrik yang kurang efisien. Oleh karena itu, perlu dilakukan monitoring secara berkala untuk mengurangi potensi kerugian dan konsumsi daya listrik yang berlebih. Untuk mengatasi hal ini, diperlukan sebuah sistem berbasis *internet of things* yang dikendalikan melalui android untuk memonitoring kondisi factor daya listrik. Sistem monitoring berbasis *internet of things* pada dasarnya telah banyak diimplementasikan di berbagai keperluan seperti monitoring detak jantung [5]; pada bidang peternakan untuk memonitoring suhu, kelembapan, dan berat kandang lebah madu [6], [7]; serta dalam bidang pertanian untuk memonitoring fase pertumbuhan padi varietas unggul yang diterapkan di UD. Mitra Tani Srono [8]. Dalam konteks sistem monitoring factor daya listrik, beberapa peneliti telah melakukan pengembangan sistem monitoring berbasis *Internet of Things* (IoT), baik untuk keperluan rumah tangga [9] maupun untuk keperluan pengembangan prototype dalam skala terbatas [10], [11]. Pada penelitian yang dilakukan oleh Hidayat et al, penerapan sistem dilakukan pada jaringan listrik satu fase [10], sedangkan penelitian yang dilakukan oleh Ma'ruf et al. pengembangannya masih dalam tahap ujicoba prototype yang sederhana [11]. Berdasarkan penelitian yang telah dilakukan didapatkan bahwa pengembangan sistem monitoring factor daya telah dilakukan oleh beberapa peneliti, hanya saja masih dalam skala yang terbatas dan menggunakan jaringan listrik satu fasa.

Berdasarkan konsep dan permasalahan yang dipaparkan, pada penelitian ini dilakukan pengembangan sistem yang dapat memonitoring factor daya dari jaringan listrik disebuah institusi. Studi kasus pada penelitian ini dilakukan di SMK Negeri 1 Glagah Banyuwangi yang memiliki daya total yang didistribusikan sebesar 250 KVA yang mana daya ini didistribusikan pada masing-masing jurusan. Pengembangan sistem ini diimplementasikan pada jaringan listrik tiga fasa dengan menggunakan sensor PZEM-004T untuk merekam kondisi tegangan dan arus untuk mendapatkan nilai factor daya serta menggunakan Arduino Mega 2560 untuk mengontrol sensor dan perangkat pendukung lainnya. Selain itu, untuk penyimpanan data pada penelitian ini juga digunakan platform *Thingspeak* dan *firebase* dan juga dikembangkan aplikasi berbasis

Android untuk melakukan kontrol dan monitoring kondisi factor daya, dan juga parameter lainnya seperti arus, tegangan dan daya pada masing-masing saluran distribusi jaringan listrik. Untuk aplikasi android dikembangkan dengan menggunakan *ionic framework* dari *AngularJS*. Dengan adanya sistem ini, diharapkan teknisi maupun orang terkait dapat melakukan monitoring secara realtime melalui platform android meliputi arus, tegangan, dan factor daya sehingga bisa diketahui data aliran listrik di masing-masing jaringan dan dapat dianalisa ketika terjadi permasalahan seperti ketidakseimbangan beban ataupun penurunan factor daya.

2. METODE PENELITIAN

Pengembangan sistem pada penelitian ini menggunakan model pengembangan *prototyping*. Metode *prototyping* merupakan salah satu metodologi pengembangan sistem menggunakan pendekatan untuk membuat rancangan dengan cepat dan bertahap sehingga dapat segera dievaluasi oleh calon pengguna [12]. Model pengembangan *prototyping* merupakan metodologi pengembangan alat monitoring faktor daya listrik yang terintegrasi dengan aplikasi android. Berikut merupakan metodologi model pengembangan *prototyping* yang ditunjukkan pada gambar 1 berikut:



Gambar 1. Model pengembangan Prototyping

2.1 Mengumpulkan Data dan Analisa Kebutuhan

Pada tahapan ini merupakan proses pengumpulan data dan Analisa kebutuhan terkait sistem yang dikembangkan yaitu sistem monitoring factor daya listrik menggunakan mikrokontroller dan terhubung dengan aplikasi android. Tempat studi kasus dan pengumpulan data penelitian dilakukan di SMK Negeri 1 Glagah. Sedangkan Analisa kebutuhan sistem meliputi Analisa kebutuhan perangkat keras, analisa kebutuhan perangkat lunak, dan analisa kebutuhan informasi.

2.2 Desain Prototype

Pada tahapan desain *prototype* dilakukan perancangan perangkat keras dan perangkat lunak yang dikembangkan yang meliputi desain rangkaian elektronika, alur program dan algoritma program, pembuatan *use case diagram*, dan pembuatan *Mockup*.

2.3 Membangun Prototype

Tahapan ini dilakukan setelah tahapan desain *prototype* yang di terapkan ke dalam sistem yang sebenarnya. Desain perangkat keras diimplementasikan dalam alat monitoring daya listrik yang sesuai seperti pada desain. Pembuatan *prototype* dilakukan dengan memprogram sensor PZEM 100A agar bisa memonitor arus, tegangan, daya dan *cosphi*. Kode program yang digunakan untuk menjalankan sensor PZEM 100A pada perangkat Arduino menggunakan kode program dari salah satu repository di Github [13]. Selanjutnya, setelah pengembangan perangkat keras telah berjalan dengan baik, maka akan dilakukan pengaturan agar data yang dibaca oleh

sensor dapat disimpan dan ditampilkan pada pengguna. Data hasil pembacaan sensor dikirimkan ke *web server* yaitu *thingspeak*. Setelah data terkirim ke *web server* selanjutnya data bisa ditampilkan oleh aplikasi Android menggunakan API key yang telah disediakan oleh *thingspeak*. Untuk pembuatan aplikasi android dikembangkan dengan menggunakan *ionic framework* dari *AngularJS*. Untuk *database* pada menu *loginnya* menggunakan *firebase authentication* untuk menyimpan hasil data sensor yang diambil dari *thingspeak* dan *library* yang digunakan untuk menampilkan grafik pada aplikasi menggunakan *Highcharts.js*.

2.4 Pengujian

Melakukan pengujian alat untuk mengetahui kekurangan dan kelemahan pada sistem rangkaian yang kemudian dilakukan pengkajian ulang terhadap alat menjadi lebih baik dan sempurna. Evaluasi dilakukan juga untuk mengetahui seberapa akurat sistem yang telah dibangun. Pada sistem yang dibuat juga diuji berdasarkan keakuratan sensor yang terdiri dari: a) Sensor arus akan diuji keakuratannya menggunakan tang ampere, b) Sensor tegangan akan diuji keakuratannya menggunakan *volt meter*, c) Untuk menghitung hasil nilai keakuratan antara nilai hasil alat ukur sebenarnya dengan nilai hasil dari sensor maka digunakan rumus sebagai berikut:

$$\frac{\text{Nilai hasil dari alat ukur} - \text{nilai hasil dari sensor}}{\text{Nilai hasil dari alat ukur}} \times 100\% \quad (1)$$

Untuk pengujian aplikasi android dilakukan dengan menggunakan metode *black-box testing*. Metode pengujian ini merupakan salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum) [14].

2.5 Evaluasi dan Perbaikan

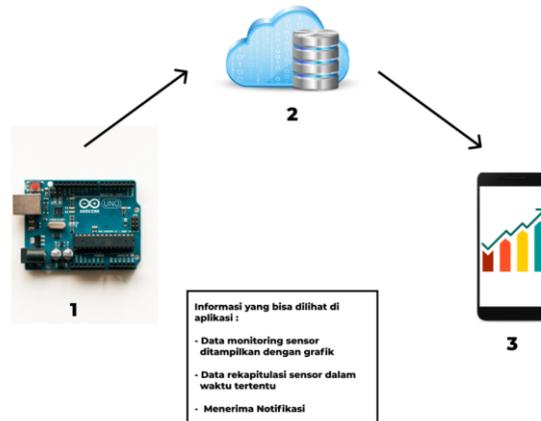
Tahapan ini dilakukan setelah melakukan pengujian dan telah diketahui kekurangan *prototype* tersebut sehingga bisa dilakukan perbaikan. Ada 2 hal yang dapat dilakukan perbaikan, yaitu perbaikan secara fisik dengan cara memperbaiki pengkabelan alat atau penambahan sensor. Sedangkan yang kedua adalah perbaikan program dari alat tersebut. Perbaikan ini dilakukan agar fungsionalitas alat tersebut semakin baik dan meminimalisir *trouble* pada saat pengoperasian di lapangan.

3. HASIL DAN PEMBAHASAN

Sistem monitoring factor daya dibuat untuk memberikan kemudahan teknisi dalam mengontrol arus, tegangan, daya dan factor daya (*cos phi*) agar selalu dalam kondisi yang seimbang pada tiap fasanya. Pada sistem ini dirancang sistem hardware untuk dapat membaca kondisi arus, tegangan, dan factor daya yang selanjutnya data dikirimkan ke *firebase* untuk disimpan dan ditampilkan melalui aplikasi android. Komponen perangkat keras yang dibutuhkan antara lain: Modul Arduino Mega 2560, Power Supply 12 V, Sensor PZEM-004T-100A (V3.0), Relay 4 Channel 10A, Sensor SCT013 100A: 50mA, Kipas 12V DC, Stop Kontak dan LCD 12C 20 x 4. Sedangkan untuk aplikasi android dibutuhkan *firebase* dalam melakukan penyimpanan data.

Sistem monitoring ini digambarkan menjadi dua bagian, yakni bagian perangkat keras dan aplikasi android. Secara garis besar, gambaran sistem yang diusulkan ditunjukkan pada gambar 2. Arduino akan menerima data yang dikirim oleh sensor PZEM 100A V3 yang meliputi arus, tegangan, daya dan factor daya. Setelah menerima data dari sensor, maka data tersebut dikirim oleh Arduino ke *webserver*. Di dalam *web server*, data-data tersebut disimpan. Protokol yang digunakan untuk mengirimkan data sensor ke *webserver* menggunakan protokol HTTP. Protokol ini bertugas untuk mengirimkan permintaan (*request*) dan memberikan respon dari *request URL* yang diberikan. Setelah data sensor tersimpan di dalam *web server*, maka aplikasi akan mengambil data tersebut. Untuk bisa mengambil data tersebut, pada *webserver* sudah memiliki API URL. API (*Application Programming Interface*) berfungsi sebagai alamat target bagi aplikasi untuk mengambil data sensor. Setelah data diterima oleh aplikasi,

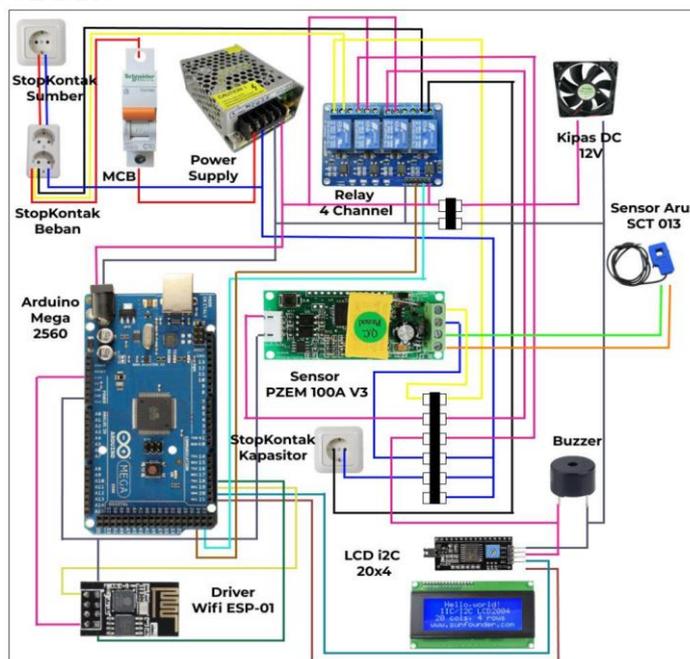
selanjutnya data akan ditampilkan. Dalam fitur monitoring sensor data akan ditampilkan melalui grafik, untuk fitur rekapitulasi data, data akan ditampilkan dengan format csv sesuai waktu yang ditentukan sebelumnya.



Gambar 2. Gambaran Sistem yang Diusulkan

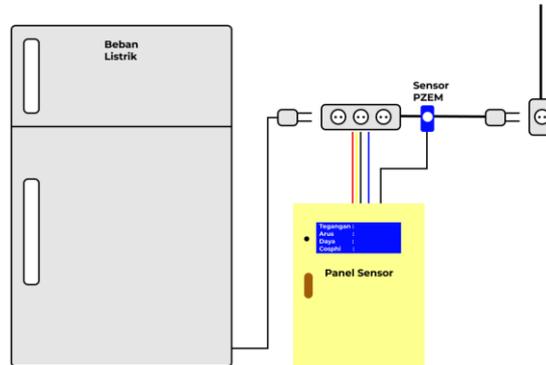
3.1. Hasil Rancangan Perangkat Keras

Rangkaian komponen untuk memonitoring factor daya listrik terdapat beberapa komponen penting didalamnya. Arduino Mega 2560 berfungsi untuk mengontrol sensor PZEM untuk mendeteksi arus, tegangan, daya dan factor daya. Selain itu juga bertugas untuk mengirim data ke LCD untuk ditampilkan dan menghidupkan *buzzer* melalui relay. Wifi ESP-01 terhubung dengan Arduino melalui pin 18 dan 19 dimana Wifi ESO-01 berfungsi untuk menangkap sinyal wifi yang dipancarkan melalui akses point dan kemudia data-data tersebut dikirimkan ke *thingspeak*. Relay terhubung pada 5VDC pada Arduino melalui channel 3. Relay pada channel 3 ini berfungsi untuk menghidupkan *buzzer* ketika nilai tegangan dan factor daya tidak stabil. Sensor PZEM mini terhubung pada 5VDC pada VCC, GND pada GND PZEM, RX pada D11 arduino dan TX pada D12. LCD 12C 20 x 4 terhubung dengan Arduino melalui pin SDA ke D20 dan pin SCL D21.



Gambar 3. Rangkaian Perangkat Keras dari Sistem Monitoring Faktor Daya Listrik

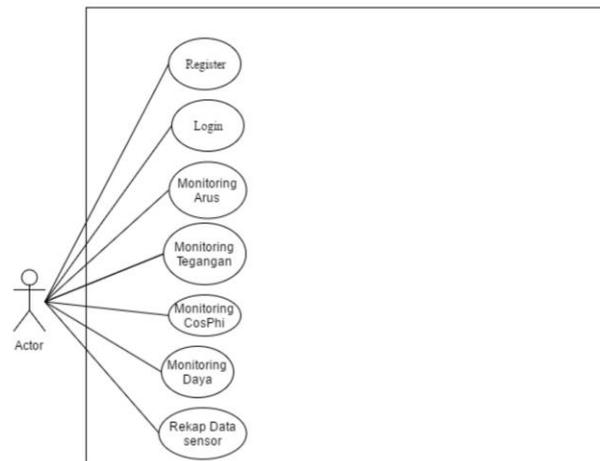
Rangkaian perangkat keras tersebut dikemas ke dalam sebuah box agar dapat melindungi komponen-komponen yang sudah dirakit. Pada gambar di bawah merupakan gambaran sistem hardware yang dibuat. Bagaimana ketika beban listrik disambungkan ke stop kontak yang sudah terpasang sensor PZEM, maka sensor tersebut bekerja untuk mendeteksi nilai tegangan, arus, daya dan *cos phi* dan menampilkannya ke sebuah LCD dan juga dikirimkan ke thingspeak untuk bisa dikelola oleh aplikasi android. Ketika beban disambungkan dan factor daya yang terdeteksi di bawah normal, maka akan menghidupkan kapasitor bank. Kapasitor bank ini berfungsi untuk mengembalikan nilai factor daya yang tidak normal. Salah satu penyebab factor daya menjadi tidak normal adalah karena beban yang dijalankan merupakan beban induktif. Beban ini memiliki daya reaktif yang besar, sehingga membuat factor daya menjadi turun.



Gambar 4. Desain Box sistem perangkat keras

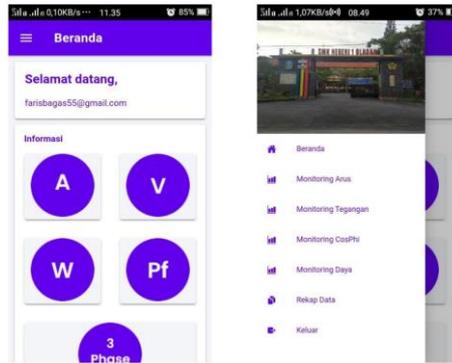
3.2. Hasil Rancangan Aplikasi Android

Aplikasi android pada penelitian ini digunakan untuk menampilkan data yang didapatkan dari sensor. Untuk desain aplikasi android hanya mempunyai satu role user yakni user teknisi. User teknisi bertugas untuk memonitoring factor daya listrik dan melakukan perbaikan ketika terjadi masalah.

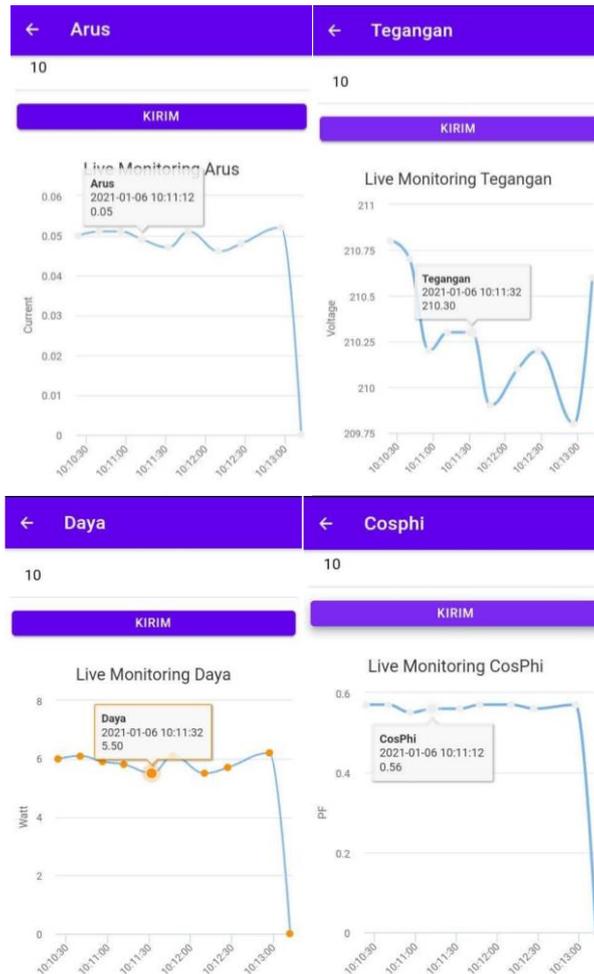


Gambar 5. Use Case Diagram

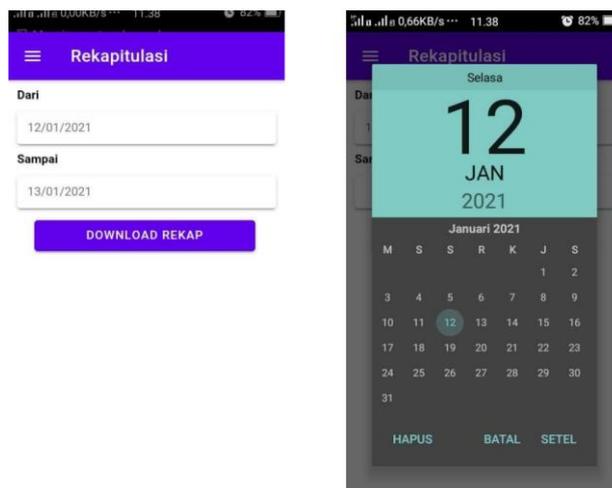
Fitur dari aplikasi android terdiri dari fitur beranda, monitoring arus, tegangan, faktor daya, dan daya, serta fitur rekapitulasi data. Untuk tampilan beranda ditunjukkan pada gambar berikut:



Gambar 6. Tampilan Beranda dan Side Menu



Gambar 7. Tampilan Monitoring Arus, Tegangan, Daya dan Faktor Daya (Cosphi)

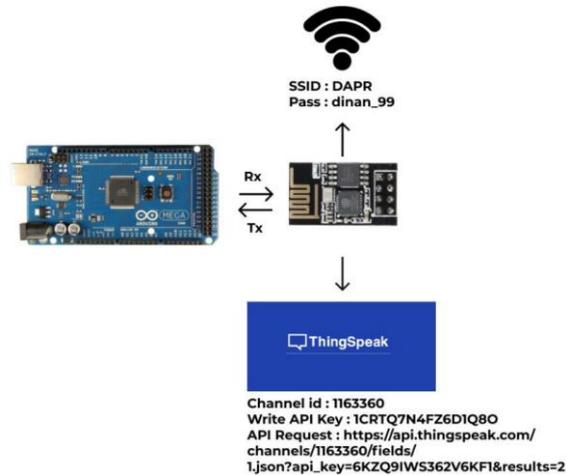


Gambar 8. Tampilan Rekapitulasi Data

Fitur utama dari aplikasi android ini adalah bisa memonitor arus, tegangan, factor daya dan daya. Pada tampilan monitoring, dibuat grafik untuk mengetahui kondisi keempat parameter tersebut pada saat beban menyala. Untuk bisa menampilkan grafik, pada penelitian ini menggunakan *library* dari *highchart.js* dan bisa berjalan secara *realtime*. Tampilan monitoring dari keempat parameter tersebut ditunjukkan pada gambar 7 dan 8. Selanjutnya, data keempat parameter tersebut dapat juga direkap berdasarkan interval waktu yang diinginkan seperti yang ditunjukkan pada gambar di atas. Menu rekapitulasi ini berfungsi untuk mengambil data keseluruhan meliputi data arus, tegangan, daya, factor daya, tanggal data dibuat dan jumlah data. Pengguna dapat memilih waktu kapan data tersebut diambil dengan memilih tanggal dari dan sampai pada menu tersebut.

3.3. Komunikasi Hardware dengan Thingspeak

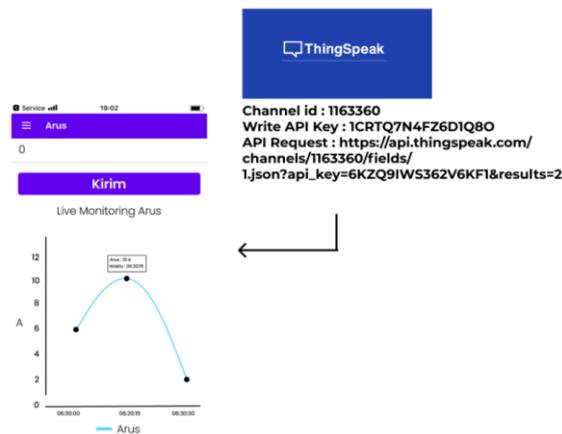
Pada gambar 9 menunjukkan komunikasi yang digunakan antara Wifi ESP-01 dengan *thingspeak*. Komunikasi yang digunakan adalah menggunakan protocol HTTP. Pada modul Wifi ESP-01 berfungsi untuk menghubungkan antara *board Arduino Mega 2560* dengan internet agar daya yang ditangkap oleh sensor dapat dikirim ke *thingspeak*. Pada modul *wifi ESP-01* sudah diatur untuk konfigurasi SSID dan password agar ketika SSID tersebut menyala, maka secara otomatis modul tersebut bisa terhubung dengan internet. Untuk bisa menghubungkan modul tersebut dengan *board Arduino mega 2560* menggunakan komunikasi serial. Komunikasi ini berfungsi untuk menghubungkan *board Arduino* dengan perangkat lainnya. Untuk dapat melakukan komunikasi ini, digunakan port Bernama RX dan TX. Untuk aturan pemasangan port pada *board Arduino* dengan modul *Wifi ESP-01* adalah port RX pada modul *Wifi ESP-01* maupun sebaiknya, sehingga dengan adanya komunikasi tersebut data yang diterima oleh *board Arduino mega* dapat dikirimkan ke *thingspeak*. Ketika data dikirimkan ke *thingspeak*, juga perlu dikonfigurasi *channel id* berapa yang akan menerima data sensor tersebut. Sehingga pada awal pembuatan *channel thingspeak*, disediakan *channel id* yang telah dikonfigurasi pada *Wifi ESP-01* agar data sensor tersebut dapat ditampilkan pada *channel* tersebut. Kemudian tipe data yang dikirim oleh *board Arduino* ke *thingspeak* adalah tipe string.



Gambar 9. Alur Komunikasi Perangkat Keras dengan thingspeak

3.4. Komunikasi Thingspeak dengan Android

Pada gambar 10 menggambarkan komunikasi yang digunakan oleh *thingspeak* dengan android. Ketika data sensor telah diterima oleh *thingspeak*, selanjutnya data tersebut akan ditampilkan ke android. Komunikasi yang digunakan untuk mengambil data dari *thingspeak* ke android adalah menggunakan protocol HTTP. Pada *thingspeak* juga sudah disediakan *API request*. *API request* berfungsi sebagai alamat bagi android untuk mengambil data tersebut. Karena pada *API request* tersebut sudah dicantumkan *channel id* dan *fields* berapa yang akan diambil data sensornya. Didalam android juga perlu di *install HttpClient* agar permintaan data ke *thingspeak* dapat dilakukan.



Gambar 10. Komunikasi Thingspeak dengan Android

3.5. Mendapatkan Data dari Thingspeak

Proses yang pertama yang harus dilakukan adalah dengan melakukan *import* terhadap modul *HttpClient*. Modul ini berfungsi untuk membuat sambungan, lalu mengirimkan permintaan dokumen ke *webservice*. Setelah melakukan *import* terhadap *module*, selanjutnya menginisialisasi *HttpClient* pada *constructor* dan selanjutnya menjalankan *HttpClient* pada fungsi kirim() di bawah. Pada fungsi tersebut, menjalankan pengambilan data dari *thingspeak*. Di dalam *thingspeak*, suda disediakan API (*Application Programming Interface*) yang berfungsi untuk menghubungkan *thingspeak* dengan aplikasi yang dibuat. Setelah di inisialisasi, maka *http* akan memanggil *variable* data untuk mengirimkan *response*, sehingga data pada *thingspeak* dapat dipanggil melalui method *get*. Setelah itu *HttpClient* menunggu respon dari server tersebut yang selanjutnya *webservice* merespon permintaan dengan kode status data yang

dilanjutkan dengan menutup sambungan ketika telah selesai memproses permintaan.

```

kirim() {
  this.loadingService.present({
    duration: 2000
  });
  const data =
  'https://api.thingspeak.com/channels/1163360/fields/1.json?api_key=6KZQ9IWS362V6KF1&results='+this.jumlah_result;
  this.http.get(data).subscribe(res => {
    this.url = res;
  });
  console.log(this.url);
}

```



Gambar 11. Hasil Pengambilan Data Arus

3.6. Memecah Data dari Thingspeak

Proses yang pertama kali dilakukan ketika data akan dipecah adalah dengan memberikan perubahan pada API URLnya. Pada source code di bawah merupakan proses untuk memecah data. Setelah data tersebut dipecah melalui API URL, selanjutnya data tersebut dipecah Kembali pada grafik. Pada grafik yang memiliki 2 koordinat, yaitu x dan y. pada koordinat x, data yang ditampilkan adalah data waktu yang diambil dari *method date*, sedangkan koordinat y data yang ditampilkan adalah data sensor yang diambil dari *thingspeak*.

```

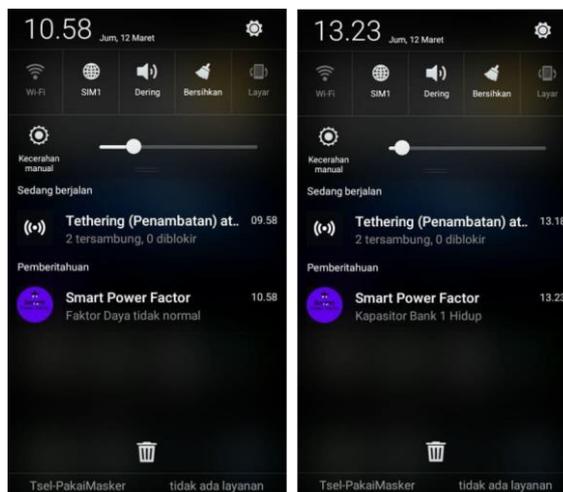
const data = 'https://api.thingspeak.com/channels/1163360/fields/1.json?api_key=6KZQ9IWS362V6KF1&results='+this.jumlah_result;
  this.http.get(data).subscribe(res => {
    this.url = res;
  });
  console.log(this.url);
}

async chartOnLoad() {
  if(this.url != undefined) {
    this.chart.series[0].setData(this.url.feeds.map(feed => {
      var x = (new Date(feed.created_at)).getTime();
      var y = parseFloat(feed.field1);
      return {
        x: x,
        y: y,
        name: "Point",
        color: "#E9E9E9"
      }
    })), true);
  }
}

```

3.7. Ujicoba Notifikasi Faktor Daya Tidak Normal dan Kapasitor Bank Menyala

Pada gambar berikut menunjukkan bahwa notifikasi akan tampil pada aplikasi pengguna apabila factor daya yang termonitoring pada beban kurang dari 0.8. Pada percobaan notifikasi tersebut menggunakan kulkas sebagai ujicoba. Ketika kulkas menyala, factor daya termonitoring sebesar 0.75. Pada gambar 12 sebelah kanan menunjukkan bahwa notifikasi tersebut akan muncul apabila *kapasitor bank* telah hidup untuk memperbaiki factor daya, sehingga pengguna mengetahui kapasitor bank telah hidup agar tidak mendatangi pada panel.



Gambar 12. Notifikasi Faktor Daya tidak normal dan Kapasitor Bank Hidup

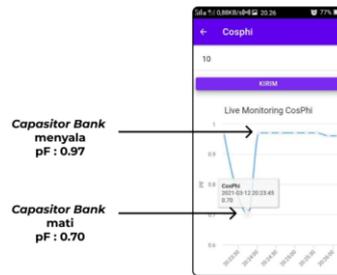
Telihat hasil pengukuran sensor di bawah bahwa sebelum kapasitor bank hidup, arus yang termonitor sensor adalah 0.10, sedangkan setelah kapasitor bank hidup terjadi penurunan arus menjadi 0.09. Hal tersebut berlaku juga pada factor daya. Terlihat sebelumnya factor daya yang termonitor sebesar 0.75 dan setelah kapasitor bank hidup terjadi peningkatan nilai factor daya mendekati normal menjadi 0.93.



Gambar 13. Hasil Pengukuran Sensor ketika Kapasitor Bank mati dan hidup

3.8. Pengaruh Capasitor Bank terhadap Faktor Daya Kulkas

Capasitor bank berpengaruh terhadap perbaikan factor daya, karena komponen ini merupakan beban yang bersifat kapasitor yang dapat menyerap daya aktif dan mengeluarkan daya reaktif, sehingga beban ini dapat memperbaiki factor daya. Pada gambar di bawah menunjukkan pengujian menggunakan beban kulkas. Pada saat capasitor bank belum menyala, factor daya yang terbaca oleh sensor sebesar 0.70, sehingga grafik mengalami penurunan seperti yang ditampilkan pada gambar di bawah. Sedangkan ketika *capasitor bank* menyala, maka factor daya pada kulkas pun diperbaiki, dari yang sebelumnya 0.70 menjadi 0.97.



Gambar 14. Grafik Faktor Daya

3.9. Menampilkan Data dalam Bentuk Grafik dan Membuat Rekapitulasi Data

Data yang telah didapatkan dari proses pembacaan sensor, selanjutnya dikirimkan ke *thingspeak* dan ditampilkan pada aplikasi android. Pada penelitian ini, data yang direkam ditampilkan dalam bentuk grafik dengan menggunakan library *highcharts*. Proses pertama yang harus dilakukan dalam menampilkan data dalam bentuk grafik adalah dengan melakukan instalasi library *highchart* melalui NPM dan selanjutnya diimport dan dijalankan. Ketika library sudah ditambahkan pada halaman *ts*, maka selanjutnya adalah membuat fungsi untuk mendapatkan data melalui *api key thingspeak*, menambahkan fungsi untuk menampuk ke grafik *highcharts* yang sudah didapatkan dari *api key thingspeak*, dan membuat fungsi untuk menjalankan grafik ketika halaman tersebut telah ditampilkan ke pengguna. Ketika data ditampilkan, maka secara tidak langsung data tersimpan ke dalam database. Pada aplikasi ini, juga dilengkapi fitur rekapitulasi data yang digunakan untuk melakukan monitoring secara berkala dan melihat trend data dari parameter yang direkam untuk keperluan yang lebih penting seperti perawatan, pemadaman berkala dan sebagainya. Proses yang harus dilakukan dalam melakukan rekapitulasi data adalah dengan membuat halaman *service* untuk memanggil API dari *thingspeak*. Proses ini dilakukan dengan menggunakan CMD (*Command Line Prompt*) dan meng-import *HttpClient*. Setelah itu dilanjutkan dengan membuat variable bernama *apiUrl* menggunakan tipe data *const* yang berisi *api url* dari *thingspeak* dan dilanjutkan dengan membuat *params start* dan *end*. Setelah membuat halaman rekapitulasi data, selanjutnya adalah melakukan import module *json2csv*, *HttpClient*, *apiService*, *File*. *Json2csv* merupakan sebuah module yang berfungsi untuk merubah file bertipe *json* menjadi *csv* yang mana file tersebut nantinya akan ditampilkan menggunakan format *csv*. *HttpClient* merupakan sebuah module yang berfungsi untuk mengirimkan http request dan menerima respon dari alamat yang diminta dalam hal ini adalah *thingspeak*. *apiService* merupakan halaman *service* yang dibuat sebelumnya, berisi *url api* target dan *params* untuk menentukan tanggal data mulai diminta dan terakhir diminta. Sedangkan *file* merupakan sebuah module yang berfungsi untuk mengirimkan file dari *thingspeak* menuju internal seluler pengguna. Jika file tersebut sudah tersimpan pada penyimpanan internal seluler pengguna, maka file tersebut bisa dibuka dengan memasang aplikasi *csv viewer*. Aplikasi ini merupakan aplikasi yang dapat membuka file dengan format *csv*.

3.10. Hasil Pengujian Sistem

Pada tahapan pengujian ini dilakukan pada perangkat keras dan aplikasi android yang telah dikembangkan. Pada pengujian perangkat keras, dilakukan pada pengujian nilai tegangan dan pengujian pada nilai beban. Pada pengujian nilai tegangan menunjukkan bahwa nilai parameter yang dibaca oleh sensor dapat dibaca dan ditampilkan dengan baik pada layar LCD dan pada aplikasi android. Selain itu, juga dilakukan pengujian menggunakan beberapa beban, seperti pengujian pada kulkas, mesin gerindra, dan setrika. Hasil pengujian yang dilakukan sensor PZEM dengan menggunakan kulkas sebagai beban. Pada kulkas tersebut tertulis daya yang digunakan maksimum sebesar 50 W. Hasil pengujian didapatkan perbedaan nilai *cosphi* antara yang direkam oleh sensor (ditampilkan pada layar LCD/android) dan hasil perhitungan manual. Hasil yang didapatkan menunjukkan tegangan yang dihasilkan sebesar 191.10 V, arus

sebesar 0.10 Ampere, Daya Aktif sebesar 39.466 Watt, daya semu sebesar 19.11 VA dan nilai Cospfi sebesar 0.75. Sedangkan pada perhitungan manual didapatkan nilai cosphi atau factor daya sebesar 0.999. Pengujian selanjutnya dilakukan dengan beban mesin gerinda. Pada mesin gerindra tertulis daya maksimum sebesar 570 W. Hasil pengujian menunjukkan bahwa tegangan yang didapatkan sebesar 184.50 Volt, Arus 1.00 Ampere, Daya semua sebesar 184.50 VA, daya aktif 304.70 W dan nilai cospfi sebesar 0.94. Sedangkan untuk perhitungan manual didapatkan nilai cospfi sebesar 0.999. Dan untuk pengujian yang terakhir dengan menggunakan setrika sebagai beban. Pada setrika tersebut tertulis daya yang digunakan maksimum sebesar 300 W. Hasil pengujian didapatkan tegangan yang dihasilkan sebesar 197.50 V, Arus sebesar 1.11 Ampere, Daya semu sebesar 219.22 VA, Daya Aktif sebesar 300W dan cospfi sebesar 0.99. Sedangkan untuk perhitungan manual nilai cospfi sebesar 0.999.



Gambar 15. Hasil Pengujian: (a) Dengan Bebas Kulkas, (b) Dengan beban Gerinda, dan (c) Dengan Beban Setrika

Selain melakukan pengujian pada perangkat keras yang dibuat, juga dilakukan pengujian pada perangkat android yang telah dikembangkan. Pengujian menggunakan metode *Black Box testing* yang menguji seluruh tampilan dari aplikasi android, mulai dari halaman daftar, halaman login, halaman utama (dashboard), halaman tampilan arus, tegangan, daya dan cospfi serta halaman lainnya. Hasil menunjukkan bahwa hasil pengujiannya 100% valid dan tidak ada error apapun.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan yang telah dipaparkan pada bagian sebelumnya didapatkan kesimpulan bahwa sistem monitoring berbasis android dan IoT dapat memonitor factor daya listrik dengan baik. Tidak hanya factor daya listrik, sistem ini juga dapat melakukan monitoring terhadap arus, tegangan dan daya listrik. Karena untuk melakukan monitoring terhadap factor daya, harus diketahui terlebih dahulu besar arus dan tegangan yang mengalir terhadap beban. Perangkat ini juga dilengkapi dengan sensor yang bekerja sesuai fungsinya dan terhubung dengan internet dengan menggunakan modul Wifi ESP-01, Arduino Mega, sensor PZEM dan sebagainya. Data sensor yang terbaca, dikirimkan ke *Websserver Thingspeak* menggunakan protocol HTTP dan dapat ditampilkan ke aplikasi android. Data yang ditampilkan menggunakan grafik dan dapat mencetak file monitoring pada waktu tertentu. Hasil pengujian perangkat keras dilakukan dengan beban kulkas, gerindra dan setrika. Hasil pengujian didapatkan bahwa adanya perbedaan nilai factor daya yang signifikan pada saat pengujian dengan menggunakan beban kulkas antara hasil perekaman dan perhitungan manual. Pada pembacaan sensor menunjukkan nilai factor daya sebesar 0.75 dan berdasarkan perhitungan manual menunjukkan nilai sebesar 0.999. Namun pada pengujian dengan beban gerindra dan setrika tidak menunjukkan perbedaan yang signifikan. Selain melakukan pengujian pada perangkat keras, juga dilakukan pengujian pada aplikasi android. Hasil yang didapatkan menunjukkan semua fitur yang dikembangkan dapat berjalan dengan baik dan tanpa ada error apapun. Sistem ini belum dikatakan sempurna dan terdapat beberapa kekuarangan. Perlu adanya

pengembangan lebih lanjut agar menjadi lebih efisien. Untuk pengembangan lebih lanjut, disarankan untuk menggantikan sumber listrik PLN menjadi genset, karena genset memiliki kapasitas daya yang terbatas, sehingga ketika daya yang digunakan hampir melebihi kapasitas maksimal, maka akan terjadi penurunan tegangan yang berakibat mudah rusaknya alat-alat elektronik. Selain itu, untuk aplikasi android juga perlu perbaikan dengan menambahkan fitur notifikasi. Fitur ini diharapkan berfungsi untuk memberitahukan kepada pengguna ketika kapasitor bank yang seharusnya memperbaiki factor daya yang turun, tetapi belum berfungsi dengan baik sehingga factor daya tetap turun. Selain itu juga perlu ditambahkan progress unduh file rekapitulasi pada status bar, sehingga pengguna bisa mengetahui apakah file tersebut sudah selesai diunduh atau masih dalam proses.

DAFTAR PUSTAKA

- [1] Syufrijal and R. Monantun, *Jaringan Distribusi Tenaga Listrik*. Jakarta: Kementerian Pendidikan Dasar Menengah dan kebudayaan Republik Indonesia, 2014.
- [2] F. E. Nasution and J. M. T, “Pengaruh Faktor Daya Pelanggan Industri Terhadap Rugi – Rugi Pada Jaringan Sisi Sekunder Transformator Distribusi PT. PLN (Persero) Area Serpong,” in *Seminar Nasional Teknologi Fakultas Teknik Universitas Krisnadwipayana*, 2019, pp. 148–163.
- [3] J. Mahseredjian, *Power System Harmonics and Passive Filter Designs*. Canada: IEEE Press, 2015.
- [4] F. R. Sebayang and A. R. Hasibuan, “Analisis Perbaikan Faktor Daya Beban Resistif, Induktif, Kapasitif Generator Sinkron 3 Fasa Menggunakan Metode Pottier,” *Singuda Ensikom*, vol. 3, no. 2, pp. 76–81, 2013, [Online]. Available: https://jurnal.usu.ac.id/index.php/singuda_ensikom/article/view/2304.
- [5] R. Hariri, L. Hakim, and R. F. Lestari, “Sistem Monitoring Detak Jantung Menggunakan Sensor AD8232 Berbasis Internet of Things,” *J. Telekomun. dan Komput.*, vol. 9, no. 3, pp. 164–172, 2019, doi: 10.22441/incomtech.v9i3.7075.
- [6] A. Hidayat *et al.*, “Monitoring Suhu, Kelembapan, Dan Berat Kandang Lebah Madu Terintegrasi Berbasis Android,” *Pros. Semin. Nas. Terap. Ris. Inov.*, vol. 6, no. 1, pp. 1032–1039, 2020.
- [7] A. Hidayat, Subono, V. A. Wardhany, L. Hakim, A. Sarirayndra, and A. S. Nugroho, “Sistem Pengelolaan Peternak dan Kandang Lebah Madu Terintegrasi Berbasis Web,” *Semin. Nas. Terap. Ris. Inov. Ke-6*, vol. 6, no. 1, pp. 1024–1031, 2020.
- [8] D. Yusuf *et al.*, “Implementasi Layanan Monitoring Fase Pertumbuhan Padi Varietas Unggul Berbasis Web dan Android sebagai Sarana Peningkatan Mutu Layanan di UD Mitra Tani,” in *Seminar Nasional Terapan Riset Inovatif (SENTRINOV) ke-7*, 2021, vol. 7, no. 3, pp. 385–392.
- [9] G. M. Berasa, R. Himawan, and F. Atabiq, “Sistem Pemantauan Perbaikan Faktor Daya Listrik Rumah Tangga Berbasis IoT,” *J. Appl. Sci. Electr. Eng. Comput. Technol.*, vol. 1, no. 3, pp. 30–35, 2020, doi: 10.30871/aseect.v1i3.2359.
- [10] H. N. Isnianto and E. Puspitaningrum, “Monitoring Tegangan, Arus, Dan Daya Secara Real Time untuk Perbaikan Faktor Daya Secara Otomatis pada Jaringan Listrik Satu Fase Berbasis Arduino,” *J. Nas. Teknol. Terap.*, vol. 2, no. 1, p. 129, 2018, doi: 10.22146/jntt.39205.
- [11] A. Ma’ruf, R. Purnama, and K. E. Susilo, “Rancang Bangun Alat Monitoring Tegangan, Arus, Daya dan Faktor Daya Berbasis IoT,” *J. Sist. Komput. dan Kecerdasan Buatan*, vol. 5, no. September, pp. 81–86, 2021.
- [12] R. S. Pressman, *Software Engineering: A Practitioner’s Approach, Seventh Edition*, 7th Editio., vol. 9781118592. New York: The McGraw-Hill Companies, 2010.
- [13] Mandulaj and Vortigont, “Arduino library for the Updated PZEM-004T v3.0 Power and Energy meter.” <https://github.com/mandulaj/PZEM-004T-v30> (accessed Apr. 28, 2022).
- [14] A. Roman, “Black-Box Testing Techniques,” in *A Study Guide to the ISTQB® Foundation Level 2018 Syllabus*, Springer, Cham, 2018, pp. 25–55.