

SISTEM KENDALI DAN *MONITORING* PENGGUNAAN PERALATAN LISTRIK DI RUMAH MENGGUNAKAN RASPBERRY PI DAN *WEB SERVICE*

^[1]Seibu Tomasua, ^[2]Dedi Triyanto, ^[3]Irma Nirmala

^{[1][2]} ^[3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak
Telp./Fax.: (0561) 577963

e-mail:

^[1]seibutomasua@student.untan.ac.id, ^[2]dedi.triyanto@siskom.untan.ac.id,

^[3]irma.nirmala@siskom.untan.ac.id

Abstrak

Sistem kendali peralatan listrik rumah secara konvensional saat ini tidak memungkinkan pengguna yang berada jauh dari rumah untuk dapat melakukan kendali dan monitoring peralatan listriknya. Untuk menjawab permasalahan tersebut dalam penelitian ini dibuat sistem yang mampu melakukan kendali dan monitoring peralatan listrik walaupun ketika pengguna berada jauh dari rumah. Perangkat keras yang digunakan adalah Raspberry Pi, Arduino, Sensor SCT-013, dan Relay Module. Perangkat lunak yang digunakan adalah Laravel framework sebagai kerangka dalam pembuatan Web Service dan website antarmuka. Hasil penelitian yang telah dilakukan adalah sebuah sistem kendali on/off dan monitoring jarak jauh untuk peralatan listrik rumah menggunakan Raspberry Pi dan Web Service. Sistem ini memiliki satu masukan listrik 220Volt serta enam jalur keluaran listrik 220Volt yang dua jalur diantaranya mampu melakukan pengukuran pemakaian daya listrik. Selain itu sistem ini menggunakan sistem manajemen account pengguna sehingga memungkinkan pengendalian peralatan listrik di banyak rumah. Hasil pengujian untuk kecepatan sistem membuktikan bahwa sistem ini dapat bekerja dengan rata-rata waktu tunda kurang dari 1 detik.

Kata kunci: Raspberry Pi, Web Service, Arduino, sensor SCT-013.

1. PENDAHULUAN

Kebutuhan masyarakat Indonesia akan energi listrik terus meningkat. Namun peningkatan kebutuhan energi listrik yang signifikan tersebut ternyata belum diimbangi dengan peningkatan kapasitas energi listrik yang ada di Indonesia. Hingga tahun 2015 rasio elektrifikasi Indonesia baru mencapai 84% yang artinya 16% masyarakat Indonesia masih belum bisa menikmati energi listrik [1]. Hal tersebut diperparah dengan pola hidup masyarakat Indonesia yang masih tergolong boros dalam menggunakan energi listrik [2]. Penghematan energi listrik perlu dilakukan agar dapat mengimbangi permasalahan tersebut. Beberapa cara menghemat energi listrik di rumah diantaranya dengan mematikan peralatan-peralatan listrik jika sedang tidak digunakan dan mengurangi penggunaan peralatan listrik dengan daya listrik yang besar. Permasalahan yang sering timbul dalam hal ini adalah

seringkali pengguna listrik lupa untuk mematikan peralatan listriknya saat sedang berada jauh dari rumah, hal tersebut mengakibatkan pemborosan listrik dan jika dibiarkan dapat menyebabkan hal-hal yang tidak diinginkan seperti kerusakan peralatan listrik dan kebakaran. Selain itu pengguna peralatan listrik tidak dapat mengetahui besarnya konsumsi daya listrik yang sedang digunakan masing-masing peralatan listrik sehingga dapat mempersulit usaha penghematan energi listrik yang dilakukannya.

Salah satu solusi untuk permasalahan tentang kendali peralatan listrik di rumah pernah disampaikan oleh Fanny Andreas pada tugas akhirnya yaitu "Rancang Bangun Sistem Kendali Dan Pemonitoran Lampu Rumah Dengan Smartphone Android Berbasis SMS Gateway Dan Mikrokontroler ATMEGA16". Sistem kendali yang digunakan pada perangkat ini yaitu layanan Short Message Service (SMS) yang dimanfaatkan untuk pengiriman data. Namun,

penggunaan sms pada sistem yang dirancang Fany Andreas memiliki rata-rata waktu tunda yang cukup besar yaitu lebih dari 10 detik [3]. Waktu tunda yang cukup besar tersebut tentu kurang bersahabat bagi masyarakat modern yang cenderung mengutamakan kecepatan dalam setiap aktivitasnya. Berdasarkan masalah-masalah tersebut, dibutuhkan sistem yang memungkinkan pengguna dapat melakukan kendali serta *monitoring* pada peralatan listrik. Sistem kendali dan *monitoring* peralatan listrik di rumah menggunakan Raspberry Pi dan *Web Service* dapat menjadi solusi untuk kebutuhan tersebut. Raspberry Pi merupakan mini komputer yang memiliki *clock speed* sebesar 700MHz [4]. *Clock speed* yang besar ini membuat Raspberry Pi dapat bekerja lebih cepat jika dibandingkan dengan mikrokontroler pada umumnya seperti Arduino Uno yang hanya memiliki *clock speed* 16MHz [5]. Penggunaan jaringan internet sebagai jalur komunikasi data memungkinkan sistem ini dapat bekerja dengan cepat sejalan dengan semakin meningkatnya kecepatan internet di Indonesia yang saat ini sudah mencapai 3,9Mbps [6].

2. PERANGKAT KENDALI UTAMA DAN WEB SERVICE

2.1. Perangkat Kendali Utama

2.1.1. Raspberry Pi

Raspberry Pi adalah sebuah komputer kecil sebesar kartu kredit yang memiliki prosesor, RAM dan *hardware port* seperti yang dimiliki komputer pada umumnya [7]. Raspberry Pi memiliki spesifikasi layaknya komputer pada umumnya namun dengan ukuran yang minimalis.



Gambar 1. Raspberry Pi

2.1.2. Arduino Nano

Arduino Nano adalah seri Arduino yang memiliki bentuk yang relatif kecil jika dibandingkan dengan bentuk arduino pada umumnya. Spesifikasi yang tertanam di

Arduino yaitu ADC(*Analog to Digital Converter*), komunikasi serial, PWM(*Pulse With Modulation*) dan GPIO(*General Purpose Input Output*) [8].



Gambar 2. Arduino Nano

2.1.3. Sensor SCT-013

Sensor SCT-013 mampu melakukan pengukuran arus listrik hingga 100A dan dapat melakukan pengukuran maksimal pada tegangan 1000V. SCT-013 dapat dioperasikan dengan Arduino sehingga hasil pengukuran sensor ini dapat diterima dan diolah oleh Arduino [9].



Gambar 3. Sensor SCT-013

2.1.4. Relay Module

Relay Module merupakan kumpulan dari beberapa *Relay* yang dirangkai pada sebuah papan PCB(*printed circuit board*). *Relay* yang digunakan pada penelitian ini bekerja pada tegangan 5Volt DC (*Direct Current*) dan dapat menyambungkan atau memutuskan aliran listrik 220Volt AC (*Alternating Current*) [10]. *Relay* dioperasikan dengan memberi tegangan 5Volt atau 0Volt pada kaki pengendalinya.



Gambar 4. Relay module

2.2. Web Service

Web Service sesungguhnya merupakan suatu perangkat lunak/aplikasi yang menyediakan layanan (*service*) bagi aplikasi-aplikasi lainnya (aplikasi klien) dan mengikuti aturan yang selama ini didefinisikan oleh *Web Service Inter-*

operability Organization (WSIO) [11]. Layanan yang disediakan *Web Service* dapat mengintegrasikan baik itu perangkat keras ataupun perangkat lunak tanpa memperdulikan bahasa pemrograman apa yang digunakan pada sisi client.

2.2.1. RESTfull

RESTfull merupakan sebutan untuk *Web Service* yang menerapkan arsitektur REST [12]. Pada dasarnya RESTfull menggunakan operasional sederhana yang dimiliki oleh protocol HTTP dan digunakan sebagai parameter untuk *Uniform Resource Locator* (URL) yang dikirim client application ke server *Web Service*.

2.2.2. JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca, dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer [13]. JSON juga berfungsi sebagai sarana *Web Service* untuk mengirim data ke aplikasi client.

2.2.3. Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*Model View Controller*) [14]. Laravel merupakan framework yang menggunakan RESTfull dalam pengoperasiannya.

2.2.4. Basis data MySQL

Basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi [15]. Basis data MySQL dikelola dengan bahasa *Structured Query Language* (SQL).

2.2.5. Komunikasi Serial

Komunikasi serial adalah salah satu metode komunikasi data dimana proses pengiriman data dilakukan satu per satu sehingga diperlukan penghantar kirim data dan terima data [16]. Sepasang Perangkat yang akan dihubungkan menggunakan komunikasi serial harus memiliki konfigurasi yang sama. Salah satu konfigurasi yang paling penting pada komunikasi serial UART adalah *Baud rate*. *Baud rate* merupakan parameter yang menentukan seberapa cepat data dikirim melalui komunikasi serial.

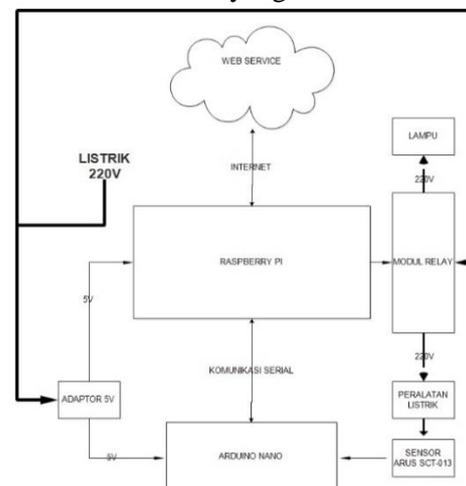
3. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode rancang bangun, diawali dengan menganalisa permasalahan yang diteliti, merancang sistem yang sesuai dengan masalah, dan membangun sistem tersebut. Analisa permasalahan akan dilakukan sebagai acuan untuk menentukan rancangan sistem seperti apa yang sesuai. Setelah Perancangan selesai maka proses pembuatan sistem dapat segera dilakukan. Sistem yang telah selesai dibuat kemudian akan melalui tahap kalibrasi untuk memastikan data yang di proses oleh sistem benar-benar valid. Setelah sistem selesai dibangun maka akan dilakukan tahap pengujian. Setelah semua tahap pengujian dapat dilalui dengan hasil yang sesuai dengan tujuan penelitian maka sistem dianggap siap diimplementasikan.

4. PERANCANGAN SISTEM

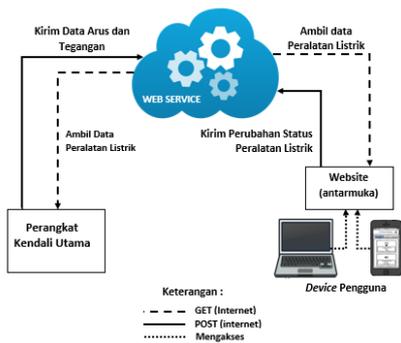
4.1. Perancangan Sistem Perangkat Kendali Utama

Sistem terdiri dari dua bagian utama yaitu perangkat kendali utama yang ada di rumah dan *Web Service*. Perangkat kendali utama bertugas mengendalikan dan memantau aliran listrik yang ada di rumah.



Gambar 5. Rancangan Perangkat Kendali Utama

Bagian sistem berikutnya adalah *Web Service* yang merupakan penghubung antara pengguna dengan perangkat kendali utama.

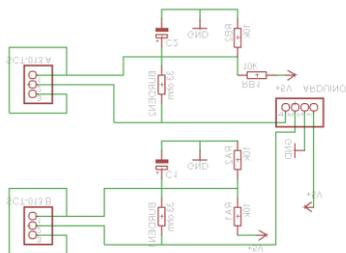


Gambar 6. Gambaran kerja *Web Service*

4.2. Perancangan Perangkat Keras

4.2.1. Sensor SCT-013

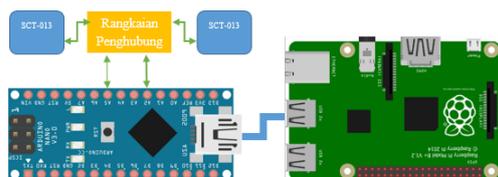
Sensor arus yang digunakan pada penelitian ini adalah sensor SCT-013 yang merupakan sensor arus listrik yang memiliki ke-mampuan pengukuran hingga 10A. Sensor ini menggunakan rangkaian penghubung untuk dapat terkoneksi dengan Arduino.



Gambar 7. Skematik Rangkaian Penghubung Sensor SCT-013

4.2.2. Arduino Nano

Arduino Nano pada sistem ini berperan sebagai perantara antara sensor dengan Raspberry Pi. *Port* Analog to digital Converter(ADC) pada Arduino Nano akan terhubung pada rangkaian penghubung sensor, sedangkan *port* USB (*Universal serial bus*) pada Arduino Nano akan dihubungkan dengan *port* USB pada Raspberry Pi agar dapat mengirim data menggunakan Komunikasi serial.



Gambar 8. Hubungan Arduino Nano dengan Raspberry Pi dan Sensor

4.2.3. Raspberry Pi

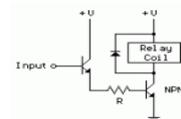
Raspberry Pi bertugas menerima data daya listrik dari arduino, melakukan komunikasi dengan *Web Service* dan mengendalikan kondisi pada *Relay* module. Raspberry Pi dan Arduino Nano dihubungkan dengan USB, dan Setiap *port* pada *relay* module akan terhubung dengan GPIO pada Raspberry Pi.



Gambar 9. Diagram Blok Hubungan Raspberry Pi Dengan Perangkat lain

4.2.4. Relay Module

Relay Module yang digunakan pada penelitian ini memiliki 6 *channel*, setiap *channel* memiliki satu buah *relay*. *Relay* tersebut memiliki pin yang akan terhubung dengan GPIO Raspberry Pi. Skematik dari untuk sebuah *relay* pada *Relay Module* dapat dilihat pada Gambar 10.



Gambar 10. Skematik Sebuah *Relay* pada *Relay Module*

4.3. Perancangan Perangkat Lunak

4.3.1. Perancangan Kode Program Arduino

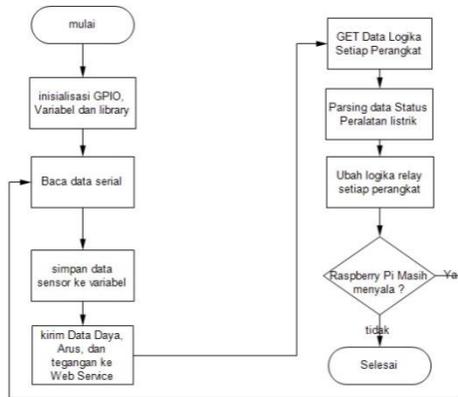
Flowchart perancangan kode program pada Arduino dapat dilihat pada Gambar 11.



Gambar 11. *Flowchart* Kode Program Arduino

4.3.2. Perancangan Kode Program Raspberry Pi

Flowchart kode program pada Raspberry Pi dapat dilihat pada gambar 12.



Gambar 12 *Flowchart* Kode Program Raspberry Pi

4.3.3. Perancangan Web Service

Flowchart dari *function* ini dapat dilihat pada gambar 13.



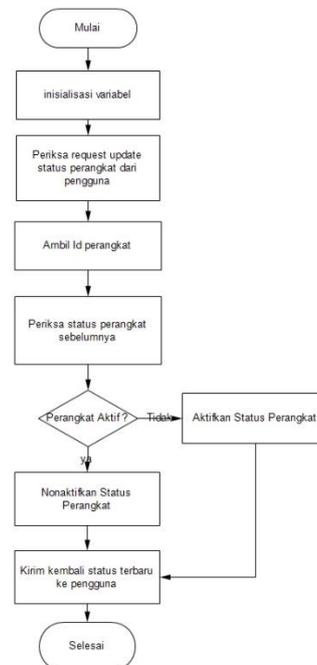
Gambar 13 *Flowchart* *Function* status peralatan listrik

Function berikutnya bertugas melayani perubahan data listrik dari perangkat kendali utama. Data yang dapat diubah oleh perangkat kendali utama yaitu data arus dan tegangan. *Flowchart* dari *function* ini dapat dilihat pada Gambar 14.



Gambar 14 *Flowchart* *Function* update data peralatan listrik

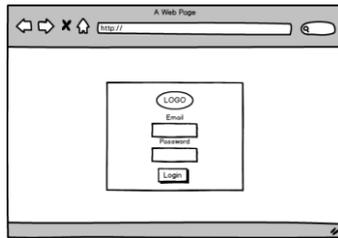
Function utama yang terakhir bertugas melayani perubahan status dari peralatan listrik dari pengguna. *Flowchart* dari *function* ini dapat dilihat pada Gambar 15.



Gambar 15 *Flowchart* update status peralatan listrik

4.3.4. Perancangan Antarmuka

Sistem dapat diakses jika telah melalui proses *login* pada halaman *login*.

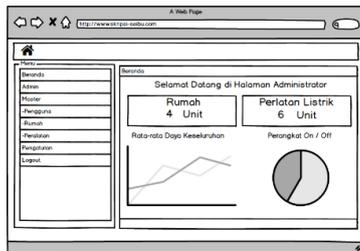


Gambar 16 Mockup halaman login

Antarmuka sistem ini terdiri dari dua sisi yaitu:

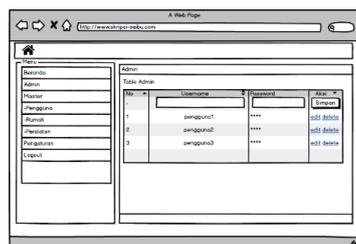
a. Administrator

Sisi Administrator dibuat agar administrator dapat mengelola sistem *Web Service* dengan mudah tanpa harus membuka sistem basis data. Sisi administrator terdiri dari beberapa menu, diantaranya menu beranda, menu admin, menu master dan menu pengaturan.



Gambar 17 Mockup beranda administrator

Halaman beranda berisi rangkuman data-data serta aktifitas pada *Web Service*. Pada bagian bawah menu beranda terdapat menu admin, yang merupakan menu yang berisi halaman data pengelola sistem.

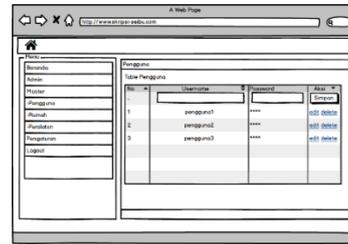


Gambar 18 Mockup menu admin

Menu admin berisi tabel yang memuat data dari semua administrator untuk semua pengelola sistem. Pada bagian bawah menu admin terdapat menu Master yang berisi tiga submenu yaitu pengguna, rumah dan peralatan.

Sub menu pengguna memiliki fungsi yang tidak jauh berbeda dari menu admin. Perbedaan dari menu pengguna ini adalah

pada data yang disimpan. Data yang disimpan adalah data untuk keperluan login pengguna.



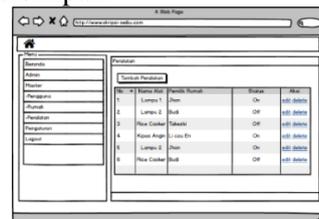
Gambar 19 Mockup menu pengguna

Sub menu rumah berisi tabel yang memuat data semua rumah yang menggunakan perangkat kendali dan *monitoring*. Pada tabel ini disimpan kode rumah, pemilik rumah dan jumlah peralatan.



Gambar 20 Mockup menu rumah

Sub menu kedua dari menu master adalah sub menu peralatan. Pada sub menu ini semua data peralatan yang ada disetiap rumah disimpan.



Gambar 21 Mockup sub menu peralatan

Menu terakhir administrator adalah menu *logout*, menu ini merupakan menu yang digunakan untuk keluar dari sistem. Jika seorang admin telah memilih menu *logout* maka untuk kembali masuk ke sistem admin harus melakukan *login* kembali.

b. Pengguna

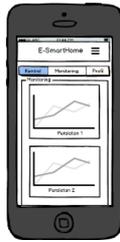
Sisi pengguna memiliki tiga halaman utama yaitu halaman kendali, *monitoring*, dan profil. Halaman kendali berisi tombol pengontrol power untuk setiap perangkat

listrik dan dilengkapi dengan nama serta data daya listrik yang digunakan pada masing-masing perangkat.



Gambar 22 Mockup Halaman Kendali

Halaman *monitoring* berisi grafik berjalan yang menggambarkan perubahan daya listrik untuk masing-masing perangkat secara realtime. Grafik perubahan daya listrik ini dibuat menggunakan Canvas JS.



Gambar 23 Halaman Monitoring

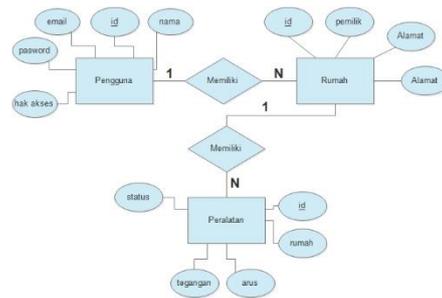
Halaman yang terakhir adalah halaman profil pengguna yang berisi informasi tentang data pemilik rumah dan nama sistem ini.



Gambar 24 Halaman Profil

4.3.5. Perancangan Basis data

Sistem basis data akan dibangun menggunakan basis data MySQL dan akan tersimpan pada *Web Service*. Secara umum rancangan basis data dapat dilihat pada *Entity Relationship Diagram (ERD)* yang ada di gambar 4.22.



Gambar 25 ERD pada Basis Data Sistem

5. HASIL DAN PEMBAHASAN

5.1. Implementasi Perangkat Kendali Utama

Perangkat kendali utama ini berfungsi menentukan menyala atau tidaknya peralatan listrik yang ada di rumah berdasarkan data yang ada di *Web Service*. Selain itu sistem ini juga bertanggung jawab mengirimkan informasi arus dan tegangan pada perangkat-perangkat listrik tertentu yang terpasang sistem sensor pada jalur listriknya. Bentuk dari perangkat kendali utama dapat dilihat pada Gambar 26.



Gambar 26 Perangkat Kendali Utama

5.2. Implementasi Web Server dan Aplikasi Antarmuka

5.2.1. Web Service

Web Service merupakan bagian yang bertugas untuk memberikan tanggapan jika terdapat permintaan dari perangkat kendali utama maupun dari aplikasi antarmuka admin dan pengguna. Tanggapan yang diberikan berdasarkan format-format URL tertentu yang disebut *route*.

```
Route::post('api/updateDaya', 'ToolController@updateDaya');
Route::post('api/tool/switch', 'ToolController@switchTool'); //user
Route::get('api/alltool/{id_home}', 'ToolController@allTool');
```

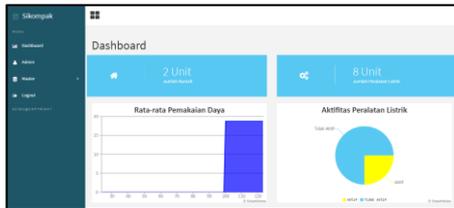
Route pertama berfungsi untuk melakukan perubahan data daya listrik di basis data. *Route* Kedua berfungsi untuk melakukan pergantian status peralatan listrik

oleh aplikasi antarmuka. Sedangkan *route* ketiga berfungsi untuk memanggil semua data peralatan listrik pada sebuah rumah.

5.2.2. Aplikasi Antarmuka

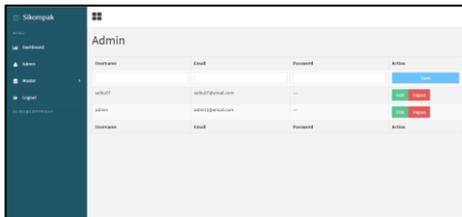
Antarmuka yang dibuat memiliki dua sisi yaitu administrator dan pengguna. Berikut merupakan tampilan dari masing-masing halaman pada halaman administrator.

- Halaman Beranda



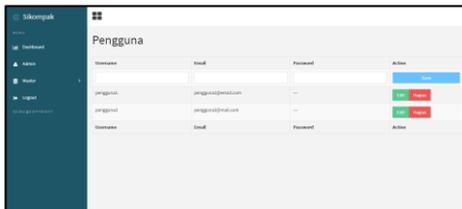
Gambar 27. Halaman Beranda

- Halaman Admin



Gambar 28. Halaman Manajemen Admin

- Halaman pengguna



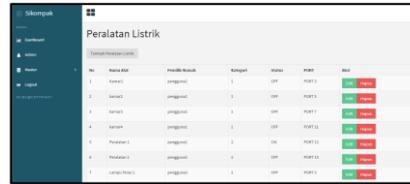
Gambar 29. Halaman Manajemen Pengguna

- Halaman Manajemen Rumah



Gambar 30. Halaman Manajemen Rumah

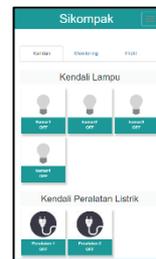
- Halaman manajemen Peralatan listrik



Gambar 31. Halaman Manajemen Peralatan Listrik

Selain bagian untuk admin pada sistem ini terdapat bagian khusus untuk pengguna. Pada bagian ini pengguna peralatan listrik dapat melakukan kendali dan *monitoring* peralatan listrik yang ada dirumahnya.

- Halaman Kendali



Gambar 32. Halaman Kendali

- Halaman *Monitoring*



Gambar 33. Halaman *Monitoring*

- Halaman Profil



Gambar 34. Halaman Profil

5.3. Pengujian dan Analisa

5.3.1. Pengujian Perangkat Keras

Pengujian perangkat keras diawali dengan menguji kemampuan *power supply* pada perangkat kendali utama. Pada awal pembuatan sistem perangkat kendali utama ditenagai oleh satu buah adaptor 5V dengan

kuat arus 1A namun sistem me-ngalami kekurangan daya listrik. Per-masalahan tersebut dapat diatasi dengan menambah *power supply* menjadi dua unit. Pengujian sensor dilakukan dengan men-cocokkan hasil pengukuran alat ukur arus standar dengan hasil pengukuran sensor arus SCT-013. Pengukuran dilakukan dengan menambah beban pada perangkat berupa peralatan listrik dengan daya besar.

Tabel 1. Pengukuran Arus Listrik

| No | Hasil Pengukuran | | error |
|-----------------|------------------|-----------------------|---------|
| | SCT-013 (A) | Alat Ukur Standar (A) | |
| 1 | 1.78 | 1.76 | 1.12 % |
| 2 | 1.61 | 1.61 | 0.00 % |
| 3 | 0.94 | 0.74 | 21.28 % |
| 4 | 1.92 | 1.82 | 5.21 % |
| 5 | 1.82 | 1.81 | 0.55 % |
| 6 | 1.17 | 0.94 | 19.66 % |
| 7 | 2.07 | 2.01 | 2.90 % |
| 8 | 2.06 | 2.01 | 2.43 % |
| 9 | 1.11 | 1.08 | 2.70 % |
| 10 | 1.23 | 1.21 | 1.63 % |
| 11 | 2.24 | 2.25 | 0.45 % |
| 12 | 2.26 | 2.26 | 0.00 % |
| 13 | 2.17 | 2.16 | 0.46 % |
| 14 | 2.12 | 2.12 | 0.00 % |
| 15 | 0.8 | 0.77 | 3.75 % |
| Rata-rata error | | | 4.08 % |

5.3.2. Pengujian Perangkat Lunak

Tahap pengujian perangkat lunak diawali dengan *whitebox testing* dan kemudian dilanjutkan dengan *blackbox testing*. *Whitebox testing* yang pertama dilakukan dengan menguji *loop* setiap program yang terdapat pada sistem. Pengujian dilakukan pada *loop* program Arduino dan Raspberry Pi dan didapati masalah pada komunikasi data antar kedua perangkat. Data yang dikirim oleh Arduino menumpuk sehingga tidak dapat dibaca sempurna oleh Raspberry Pi. Hal ini dikarenakan beban kerja Raspberry Pi yang lebih besar dibandingkan dengan beban kerja Arduino. Permasalahan tersebut dapat diatasi dengan memberikan waktu tunda sebesar 100ms pada perulangan Arduino. Pengujian perangkat lunak selanjutnya yang menemui masalah adalah saat menguji sistem *monitoring* daya listrik pada antarmuka *website*. Saat pengguna membuka menu menu *monitoring*, *Web Service* bekerja dengan lambat dan kinerja *Web Service* menjadi semakin berat saat menu *monitoring* dibuka. Untuk mengatasi masalah tersebut maka dilakukan penyederhanaan alur program.

Setelah *whitebox testing* selesai dilakukan maka tahap pengujian perangkat lunak di lanjutkan dengan *blackbox testing*. *Blackbox testing* pertama dilakukan pada antarmuka *website* khususnya sisi admin dan sisi pengguna. Pengujian dilakukan dengan menguji autentifikasi pada form *login*. Sistem autentifikasi berjalan dengan baik selama dilakukan pengujian.

5.3.3. Pengujian Waktu Trasmisi

Pengujian waktu transmisi Tahap pertama adalah tahap pengguna mengubah status peralatan listrik melalui tombol. Waktu yang dibutuhkan sistem dalam merespon perintah tersebut diuji sebanyak 15 kali.

Tabel 2. Respon *Web Service* terhadap Aplikasi Pengguna

| No | Waktu Respon (ms) |
|-----------|-------------------|
| 1 | 172 |
| 2 | 379 |
| 3 | 169 |
| 4 | 132 |
| 5 | 201 |
| 6 | 363 |
| 7 | 163 |
| 8 | 250 |
| 9 | 140 |
| 10 | 179 |
| 11 | 236 |
| 12 | 198 |
| 13 | 189 |
| 14 | 322 |
| 15 | 287 |
| Rata-rata | 225.33 |

Berdasarkan percobaan yang dilakukan pada Tabel 5.2 sistem memberi umpan balik kepada aplikasi antarmuka dengan waktu rata-rata 255,33 ms. Waktu tersebut merupakan waktu yang dibutuhkan sistem dari mulai menerima *request*, melakukan perubahan basis data, sampai memberi umpan balik ke aplikasi antarmuka. Berikutnya Raspberry Pi secara terus menerus akan melakukan pembacaan data yang ada di basis data dengan melakukan *request* POST pada *Web Service* dengan menyertakan id rumah. Setelah menerima *request* POST dari Raspberry Pi, *Web Service* akan memberi umpan balik berupa

data peralatan listrik yang memiliki id tersebut berupa array JSON.

Tabel 3 Respon *Web Service* Terhadap Raspberry Pi

| No | Waktu Respon (ms) |
|------------------|-------------------|
| 1 | 144 |
| 2 | 187 |
| 3 | 152 |
| 4 | 166 |
| 5 | 224 |
| 6 | 139 |
| 7 | 137 |
| 8 | 231 |
| 9 | 328 |
| 10 | 161 |
| 11 | 320 |
| 12 | 145 |
| 13 | 155 |
| 14 | 294 |
| 15 | 189 |
| Rata-rata | 198.13 |

Beberapa pengujian kecepatan yang telah dilakukan sebelumnya hanya berfokus pada waktu transmisi data pada sisi perangkat lunak dan belum termasuk waktu kerja perangkat keras. Untuk menguji waktu kerja perangkat keras, perlu dilakukan pengujian secara keseluruhan yaitu dengan menghitung lama waktu yang dibutuhkan sistem mulai dari user menekan tombol sampai peralatan listrik menyala. Pengujian kali ini menggunakan lampu pijar sebagai indikator perubahan status peralatan listrik dan *stop-watch* untuk menghitung waktu respon.

Tabel 4. Respon Sistem Secara Keseluruhan

| No | Kecepatan (s) |
|------------------|---------------|
| 1 | 0.7 |
| 2 | 1.1 |
| 3 | 0.9 |
| 4 | 0.5 |
| 5 | 0.8 |
| 6 | 0.8 |
| 7 | 0.5 |
| 8 | 1.7 |
| 9 | 0.9 |
| 10 | 1.5 |
| 11 | 1.3 |
| 12 | 0.4 |
| 13 | 0.5 |
| 14 | 0.4 |
| 15 | 0.2 |
| Rata-rata | 0.813 |

Berdasarkan beberapa pengujian yang telah dilakukan maka dapat dihitung lama waktu yang dibutuhkan sistem perangkat keras untuk melakukan pengolahan data status dari *Web Service* sampai mengubah logika *relay* yaitu sebesar 359,54 ms. Lamanya waktu tersebut didapatkan dengan mengurangi waktu kerja kerja keseluruhan dengan waktu kerja *Web Service*.

5.3.4. Hasil Pengujian Keseluruhan

Setelah dilakukan pengujian maka dapat dilakukan analisis terhadap hasil pengujian yang telah dilakukan. Analisis dari pengujian ini dilakukan untuk mengetahui sejauh mana sistem bekerja pada saat digunakan. Analisis dari hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4. Hasil Analisis

| No | Pengujian | Proses | Hasil yang diharapkan | Hasil Pengujian |
|----|--|--|--|-----------------|
| 1 | Kemampuan <i>Power supply</i> | Pengamatan kinerja sistem terhadap penggunaan <i>power supply</i> yang ada. | Sistem perangkat keras dapat bekerja tanpa kekurangan tegangan listrik. | Berhasil |
| 2 | Pengukuran Arus Listrik | Pengukuran dibandingkan dengan alat ukur <i>Clamp Meter</i> . | Hasil pengukuran sensor mendekati atau sama dengan hasil pengukuran alat ukur standar. | Berhasil |
| 3 | Pengiriman data listrik dari Arduino ke Raspberry Pi | Dilakukan pengiriman arus dan tegangan peralatan listrik dari Arduino ke Raspberry Pi. | Data hasil pengukuran sensor tampil pada terminal Raspberry Pi. | Berhasil |

| | | | | |
|---|--|---|---|----------|
| 4 | Pengiriman daya Listrik dari Raspberry Pi ke <i>Web Service</i> | Dilakukan pengiriman data daya listrik hasil pengukuran sensor dari Raspberry Pi ke <i>Web Service</i> . | Data Listrik pada basis data <i>Web Service</i> dapat berubah sesuai hasil pengukuran sensor. | Berhasil |
| 5 | Pengiriman data status peralatan listrik dari <i>Web Service</i> ke Raspberry Pi | Dilakukan <i>request</i> data status peralatan listrik oleh Raspberry Pi ke <i>Web Service</i> . | <i>Web Service</i> memberi umpan balik berupa data status peralatan listrik kepada Raspberry Pi dalam bentuk <i>array JSON</i> . | Berhasil |
| 6 | Autentifikasi <i>login</i> pada aplikasi antarmuka | Dilakukan upaya masuk ke sistem dengan data-data pengguna yang ada pada basis data. | Sistem menerima permintaan masuk jika data pengguna yang dimasukkan benar dan sistem menolak permintaan masuk jika data pengguna yang dimasukkan salah. | Berhasil |
| 7 | Perubahan staus peralatan listrik pada antarmuka pengguna | Mengamati perubahan status peralatan listrik jika tombol-tombol peralatan listrik ditekan. | Status peralatan listrik pada basis data berubah disertai perubahan Indikator pada tombol peralatan listrik. | Berhasil |
| 8 | Antarmuka menampilkan data dalam bentuk grafik | Dilakukan pengambilan data grafik dari basis data | Perubahan data grafik mengikuti perubahan nilai daya listrik di basis data. | Berhasil |
| 9 | Waktu kerja sistem pengontrolan | Dilakukan pengukuran waktu kerja sistem dari mulai tombol peralatan listrik ditekan pengguna sampai terjadi perubahan status (aktif/tidak aktif) pada perangkat kendali utama | waktu rata-rata kerja sistem maksimal 1 detik | Berhasil |

6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan yaitu.

1. Sistem kendali dan *monitoring* peralatan listrik rumah dengan menggunakan Raspberry Pi dan *Web Service* berhasil dibuat.
2. Sistem yang telah dibuat pada penelitian ini dapat bekerja secara *realtime (soft realtime)* dengan waktu rata-rata kerja sistem kurang dari 1 detik.
3. Raspberry Pi dan *Web Service* dapat terhubung dengan jaringan internet dan memanfaatkan metode-metode pada arsitektur REST yaitu POST dan GET.
4. Pengukuran arus listrik dilakukan dengan menggunakan Sensor arus SCT-013 yang dikendalikan oleh Arduino menggunakan *library Emon*.
5. Sensor SCT-013 dapat melakukan pengukuran data arus secara akurat.
6. Arduino dan *Web Service* terhubung dengan komunikasi serial (UART) pada *baud rate 9600*.
7. Antarmuka *website* digunakan untuk manajemen data peralatan listrik yang ada di basis data *Web Service*.

6.2. Saran

Berdasarkan penelitian yang telah dilakukan dalam membuat sistem kendali dan *monitoring* peralatan listrik menggunakan Raspberry Pi dan *Web Service* maka di peroleh beberapa ide yang menjadi saran untuk penelitian selanjutnya antara lain.

1. Penggunaan perangkat *wireless* sangat dianjurkan untuk mengendalikan setiap *relay* peralatan listrik agar perangkat kendali utama dapat mengendalikan peralatan listrik yang letaknya jauh tanpa menggunakan kabel.
2. Penggunaan sensor tegangan dianjurkan untuk mendapatkan hasil pengukuran daya listrik yang lebih akurat.
3. Pada penelitian selanjutnya disarankan menggunakan mini komputer yang mampu membaca data *ADC*.

REFERENSI

- [1] Kementerian Komunikasi dan Informasi, "Pembangunan Sektor Energi, Bekal Awal Kekuatan Mendunia," 2016. [Online]. Available: <https://kominfo.go.id/content/detail/6223/pembangunan-sektor-energi-bekal-awal-kekuatan->

- mendunia/0/infografis. [Accessed 28 9 2016].
- [2] BAPENAS, "Masyarakat Indonesia Boros Gunakan Listrik," 2016. [Online]. Available: <http://perpustakaan.bappenas.go.id/1ontar/file?file=digital/blob/F28555/Masyarakat%20Indonesia%20Boros.htm>. [Accessed 16 9 2016].
- [3] F. Andreas, "Rancang Bangun Sistem Kontrol Dan Pemantauan Lampu Rumah Dengan Smartphone Android Berbasis SMS Gateway Dan Mikrokontroler ATMEGA16," *Jurnal Coding Sistem Komputer Untan*, vol. 3, pp. 33-34, 2015.
- [4] Raspberry Pi, "RASPBERRY PI 1 MODEL B+," 2015. [Online]. Available: <https://www.raspberrypi.org/products/model-b-plus/>. [Accessed 8 9 2016].
- [5] Arduino, "Arduino/Genuino UNO," 2016. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Accessed 10 10 2016].
- [6] O. Yusuf, "Kecepatan Internet Indonesia Naik Dua Kali Lipat," 2016. [Online]. Available: <http://tekno.kompas.com/read/2016/04/01/19290007/Kecepatan.Internet.Indonesia.Naik.Dua.Kali.Lipat>. [Accessed 10 10 2016].
- [7] Gus, "What is a Raspberry Pi: All you need to know about the Pi," 2015. [Online]. Available: <https://pimylifeup.com/what-is-raspberry-pi/>. [Accessed 28 9 2016].
- [8] Datasheet, datasheet arduino nano, 2008.
- [9] Datasheet, Datasheet SCT-013, Beijing: Electronic, Beijing YaoHuadechang, 2011.
- [10] Datasheet, Rellay Sngle, Beijing, 2006.
- [11] A. Nugroho and K. Mustofa, "PERBANDINGAN ANTARA "BIG" WEB SERVICE DENGAN RESTFUL," *JURNAL INFORMATIKA VOL. 11 UGM*, pp. 11-12, 2012.
- [12] Feridi, "Mengenal RESTful Web Services," 2016. [Online]. Available: <https://www.codepolitan.com/mengenal-restful-web-services>. [Accessed 4 9 2016].
- [13] json.org, "Pengenalan JSON," 2016. [Online]. Available: www.json.org/json-id.html. [Accessed 4 9 2016].
- [14] IdCloudHost, "Pengertian dan Keunggulan Framework Laravel," 2016. [Online]. Available: <https://idcloudhost.com/pengertian-dan-keunggulan-framework-laravel/>. [Accessed 4 9 2016].
- [15] Hindro, "Pengertian Basis data," 2012. [Online]. Available: <http://www.termasmedia.com/65-pengertian/69-pengertian-basis-data.html>. [Accessed 31 8 2016].
- [16] D. Arinyus, "Komunikasi Data," Yogyakarta, Andi Offset, 2008.