

## **SISTEM KENDALI LAMPU DAN STEKER TERINTEGRASI MENGUNAKAN MIKROKONTROLER BERBASIS WEB SERVICE**

**Heldi Ardian<sup>1</sup>, Dedi Triyanto<sup>2</sup>, Tedy Rismawan<sup>3</sup>**

<sup>1,2,3</sup>Jurusan Sistem Komputer; Fakultas MIPA Universitas Tanjungpura Pontianak, Jalan Prof  
Dr. H. Hadari Nawawi Pontianak  
Telp/fax.: (0561) 577963  
e-mail: <sup>1</sup>heldionardi@gmail.com, <sup>2</sup>dedi.triyanto@siskom.untan.ac.id,  
<sup>3</sup>tedyrismawan@siskom.untan.ac.id

### **ABSTRAK**

*Meningkatnya penggunaan energi listrik oleh masyarakat dipengaruhi oleh peningkatan penggunaan perangkat elektronik pada rumah tangga. Penggunaan listrik yang berlebihan akan menghabiskan energi minyak dunia serta menyebabkan meningkatkan jumlah tagihan listrik. Untuk mengatasi permasalahan tersebut maka dilakukan penelitian sistem kendali lampu dan steker terintegrasi menggunakan mikrokontroler berbasis web service. Penelitian ini menggunakan Arduino Yun sebagai kendali utama sistem, Node MCU, steker, lampu, dan aplikasi antarmuka. Arduino Yun berfungsi server kendali yang menyediakan jalur komunikasi serial dan memproses sinyal masukkan dari Node MCU melalui aplikasi antarmuka dan memberikan feedback berupa aksi pada perangkat lampu dan steker. Pada Arduino diterapkan inisialisasi program dan konfigurasi perangkat keras yang menghubungkan perangkat ke server menggunakan Node MCU, kemudian menginstruksi sebuah kondisi sehingga menghasilkan keluaran. Hasil penelitian berupa sistem kendali lampu dan steker terintegrasi yang menerapkan arsitektur REST pada web service dan memonitor kondisi perangkat menggunakan aplikasi antarmuka berbasis web dengan kondisi request transmisi data pada server tidak melebihi 120 data/detik. Tingkat kuat atau lemahnya sinyal server tergantung pada hambatan jarak  $\leq 20$  meter dan ketebalan tembok. Setiap perangkat diatur pada kondisi menyala atau mati, relay akan merespon instruksi yang diberikan oleh Node MCU jika diatur melalui aplikasi antarmuka.*

**Kata Kunci :** Lampu, Steker, Arduino Yun, Node MCU, Aplikasi Antarmuka.

### **1. PENDAHULUAN**

Semakin meningkatnya penggunaan energi listrik oleh masyarakat dipengaruhi oleh meningkatnya penggunaan teknologi baru seperti *smartphone* dan komputer, disamping penggunaan perangkat elektronik pada rumah tangga. listrik telah menjadi sumber energi utama dalam setiap kegiatan di rumah tangga maupun industri.

Pengelolaan sumber daya listrik yang berlebih dipengaruhi oleh banyaknya alat elektronik dan unit lampu yang menyala dan kurangnya waktu pengelolaan dalam mengatur kapasitas penggunaan alat tersebut. Rangkaian eletronika digunakan untuk membantu kehidupan masyarakat seperti

halnya lampu otomatis yang tidak memerlukan saklar konvensional, sehingga teknologi akan menggeser peranan saklar konvensional menjadi saklar otomatis yang dapat di atur berdasarkan waktu dan kondisi tertentu

Pada penelitian sebelumnya membahas mengenai efisiensi listrik menggunakan sensor cahaya sehingga terbentuk suatu sistem penerangan otomatis yang digunakan untuk mengendalikan lampu berdasarkan pengukuran intensitas cahaya menggunakan sensor, dimana sensor yang dikonversi akan diubah menjadi nilai ADC[1]. Selanjutnya terdapat penelitian yang membahas mengenai efisiensi listrik menggunakan metode penjadwalan untuk mengendalikan alat listrik

berbasis waktu. Pada penelitian tersebut masih dalam bentuk simulasi *prototype* menggunakan kabel[2] dan tidak dapat diimplementasikan secara langsung pada rumah, karena harus merubah arsitektur rumah dari awal pembangunan.

Berdasarkan kondisi tersebut, maka dibutuhkan suatu sistem yang dapat mengatur arus listrik yang dialirkan melalui perantara steker dan lampu secara otomatis pada waktu dan kondisi tertentu dan membuat perangkat yang dapat diimplementasikan secara langsung pada masyarakat. Salah satu perangkat yang dapat digunakan adalah menggunakan jaringan tanpa kabel (*wireless*) untuk mengendalikan arus listrik dengan memanfaatkan relay sehingga tidak mengubah arsitektur pembangunan rumah.

Alat yang dibuat berupa sistem mikrokontroler (pengendali) berbasis tanpa kabel (*wireless*) yang dapat mengatur hubungan arus listrik yang dialirkan pada steker dan mengkondisikan lampu menyala atau padam secara otomatis pada waktu dan kondisi tertentu. Lampu dan steker dikendalikan berdasarkan perintah waktu pemberian aksi menggunakan *interface* oleh pengguna berupa aplikasi berbasis *web service* yang terpasang pada Arduino Yun. Dalam hal ini OpenWrt-Yun menggunakan arsitektur *REST (Representational State Transfer)* untuk mengakses arduino melalui *URL (Uniform Resource Locator)* dengan menggunakan bahasa pemrograman *PHP (Hypertext Preprocessor)*. *REST API (Application Programming Interface)* pada arduino dapat diakses dengan *password* yang telah dimiliki oleh pengguna dan dapat diubah dengan cara mengkonfigurasi pada Arduino Yun.

## 2. LANDASAN TEORI

### 2.1. Arduino Yun

Arduino adalah *prototype platform* yang bersifat *open-source* yang terdiri dari perangkat keras dan perangkat lunak. Pembuatan Arduino pertama kali dimulai pada tahun 2005 oleh sebuah perusahaan komputer Olivetti di Ivrea, Italia. Tujuan proyek ini awalnya untuk membuat peralatan *control* interaktif dan modul pembelajaran bagi siswa yang lebih murah dibandingkan

dengan *prototype* yang lain. Arduino yang berbasis *open source* melibatkan tim pengembang. Massimo Banzi dan David Cuartielles sebagai pendiri Arduino yang pada awalnya mereka memberi nama proyek tersebut dengan sebutan arduin dari Ivrea, tetapi seiring berkembangnya zaman nama proyek tersebut diubah dalam versi bahasa inggris yang sekarang dikenal dengan sebutan Arduino.

Arduino dikembangkan dari thesis Hernando Barragan didesain interaksi *institute* Ivrea. Arduino dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor dan aktuator lainnya. Mikrokontroler pada *board* Arduino diprogram dengan menggunakan bahasa pemrograman Arduino (*based on wiring*) dan IDE (*Integrated Development Environment*) Arduino (*based on processing*). Proyek Arduino dapat berjalan sendiri atau juga dapat berkomunikasi dengan *software* yang berjalan pada Komputer[3].

Pada penelitian ini digunakan Arduino Yun yang merupakan papan mikrokontroler ATmega32u4 dan Atheros AR9331. Prosesor yang dimiliki oleh Atheros dapat mendukung pendistribusian data menggunakan Linux yang dikenal OpenWrt-Yun. Mikrokontroler ini memiliki fitur *Ethernet* dan *Wifi*, *USB port*, *micro-SD card slot*, *20 pin I/O* dimana *7 pin* digunakan sebagai keluaran *PWM* dan *12 pin* lainnya digunakan sebagai masukan data analog, *crystal* osilator, dukungan koneksi melalui *USB*, *ICSP header*, dan *3 buah tombol reset*. Bentuk fisik Arduino Yun dapat dilihat pada gambar 1.



Gambar 1. Bentuk Fisik Arduino Yun

## 2.2 Node MCU (Multipoint Control Unit) Version 3.0

Node MCU diciptakan setelah ESP8266 muncul dipasaran yang di produksi pada saat sistem *espressif* mulai berkembang. ESP8266 merupakan *Wireless Fidelity* (Wi-Fi) yang terintegrasi dengan *core* Tensilica Xtensa LX106 dan banyak diterapkan dalam aplikasi IoT (*Internet of Things*)

Penerapan penggunaan *NoceMCU-firmware* pertama kali dimulai pada saat Hong yang berkomitmen untuk percobaan *file* pertama untuk GitHub. Setelah dua bulan berjalan, proyek diperluas untuk menyertakan sebuah *platform open-Hardware* oleh Huang R. Arduino.cc mulai mengembangkan MCU *board* baru yang berdasarkan pada prosesor non-AVR seperti ARM/SAM MCU dan digunakan dalam arduino. Bentuk fisik Node MCU v3 dapat dilihat pada gambar 2.



Gambar 2. Bentuk fisik Node MCU v3

## 2.3 Web Service

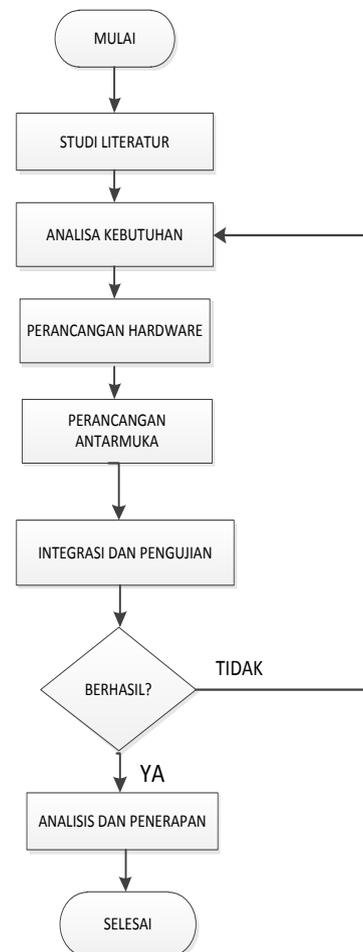
Hubungan antara *platform* yang berbeda antara *client* dan *server* membutuhkan sebuah standar aplikasi yang disebut *web service*. *Web service* adalah komponen aplikasi yang menyediakan akses data dan operasi tanpa harus berhubungan secara langsung dengan *database* pada sistem. Dalam dunia internet, *web service* merupakan *interface* yang dapat diakses melalui jaringan untuk memanggil fungsi aplikasi yang dibangun menggunakan standar teknologi internet[4].

*REST* (*Representation State Transfer*) atau yang biasa disebut *RESTful API* merupakan gaya arsitektur antara hubungan

*client* dan *server* menggunakan protokol komunikasi *stateless* seperti *HTTP* dalam pembangunan *web service*. Selain *REST* terdapat juga tipe *SOAP* (*Simple Object Access Protocol*) dalam *web service*. Berbeda dengan *SOAP*, *REST* lebih menitikberatkan pada representasi *resource* (data)[5].

## 3. METODE PENELITIAN

Metodologi yang digunakan pada penelitian ini meliputi studi literatur, metode pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian, pembahasan, kesimpulan. Diagram alir penelitian dapat dilihat pada gambar 3.

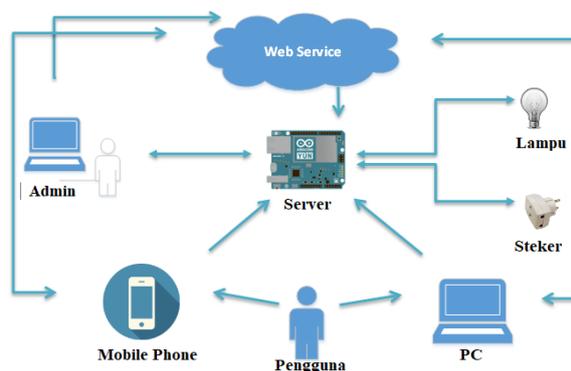


Gambar 3. Diagram Alir Penelitian

## 4. PERANCANGAN SISTEM

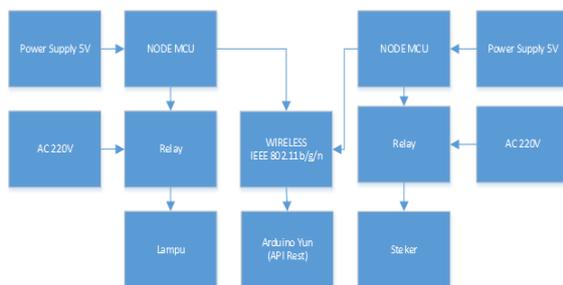
### 4.1 Gambaran Umum Sistem

Perancangan sistem dijelaskan pada Gambar 4.



**Gambar 4.** Gambaran Umum Sistem

Diagram blok sistem ditampilkan pada gambar 5.



**Gambar 5.** Diagram Blok Sistem

## 5. IMPLEMENTASI PERANCANGAN DAN PENGUJIAN SISTEM

### 5.1 Implementasi Lampu dan Steker

Penerapan penggunaan Node MCU sebagai perantara penghubung perangkat pada server Arduino Yun dan Relay sebagai kendali hubungan arus listrik pada lampu dan steker dapat dilihat pada gambar 6 dan gambar 7.



**Gambar 6.** Perangkat Lampu



**Gambar 7.** Perangkat Steker

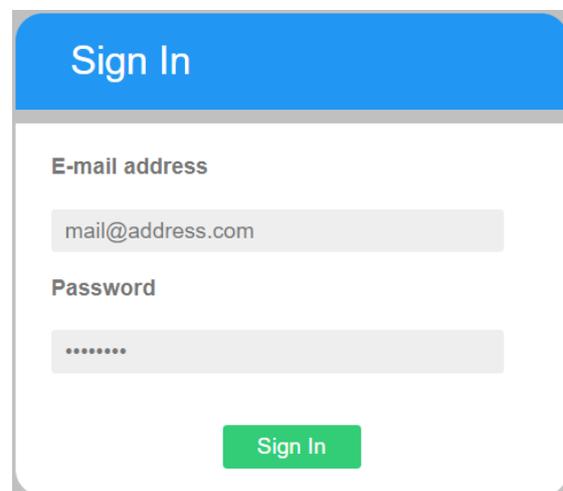
Penerapan penggunaan Arduino Yun sebagai server dapat dilihat pada gambar 8.



**Gambar 8.** Arduino Yun (Server)

## 4.1 Implementasi Antarmuka

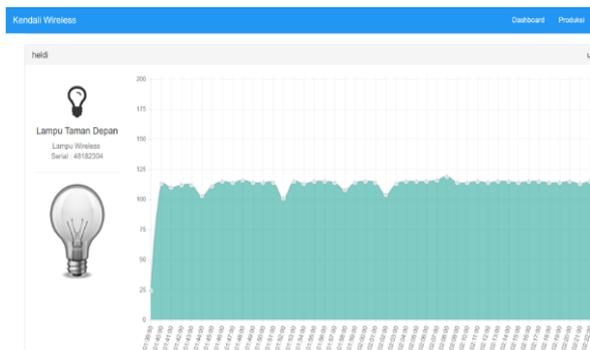
### 1. Halaman Login



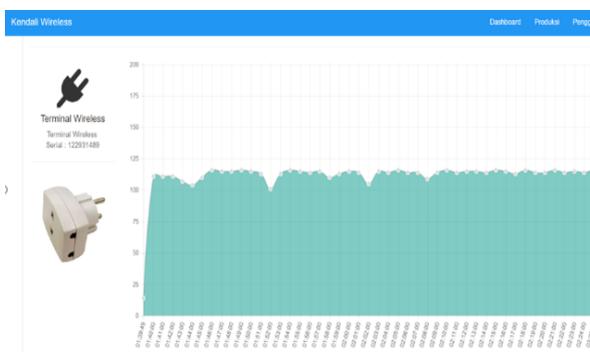
**Gambar 9.** Halaman Login

Proses pertama dengan mengakses alamat 192.168.240.1/app pada localhost menggunakan aplikasi web browser kemudian ditampilkan halaman login yang terdiri atas kolom login dengan memasukkan *username* dan *password* untuk mengakses halaman selanjutnya. Dilakukan pengujian untuk *login level admin*.

## 2. Halaman Admin

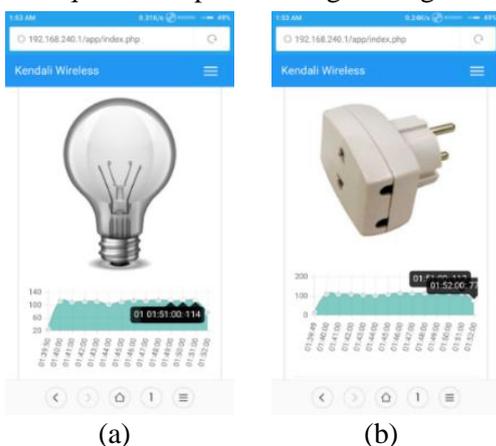


**Gambar 10.** Tampilan (a) Data Lampu  
(Pengguna : user@gmail.com)

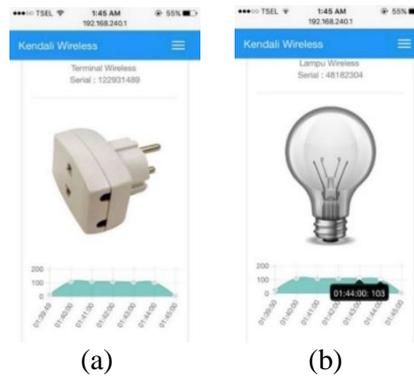


**Gambar 11.** Tampilan (b) Data Steker  
(Pengguna : user@gmail.com)

Pada gambar 10 dan gambar 11 ditampilkan hasil setelah dilakukan proses login menggunakan *username* dan *password* level *admin*, dan aplikasi antarmuka menampilkan hasil daftar perangkat dan grafik *request data* pada masing-masing alat.



**Gambar 12.** (a) dan (b) Tampilan Halaman Login Admin pada web browser jenis Android



**Gambar 13.** (a) dan (b) Tampilan Halaman Login Admin pada Google Chrome web browser jenis iOS

## 3. Halaman Produksi Alat (*Admin*) Implementasi dari proses produksi alat oleh *admin* dapat dilihat pada gambar 14.

Tambah Produk

Nama Produk

Jenis Produk

Tanggal Produksi

Font Awesome Icon >>> <http://fontawesome.io/icons/>

Kode Serial

Status  
 Aktif  Tidak Aktif
 Gambar  
 No file chosen

**Gambar 14.** Form tambah produk oleh Admin

Dilakukan pengujian dengan mendaftarkan 2 buah perangkat dengan jenis “lampu wireless” untuk perngakat jenis lampu dan “terminal wireless” untuk perangkat jenis steker. Informasi data pada *form* halaman diisi dengan data Nama Produk, Jenis Produk, Tanggal Produksi dan Icon (\**Optional*). Setelah data diisi, didapat kode serial untuk setiap perangkat dan didapat dari proses generate kode serial dari instruksi yang dapat dilihat pada Kode program 1.

### Kode Program 1. Instruksi Kode Serial

```
<?php
include "konfigurasi.php";
$array['serial'] = (strtotime(date("Y-m-d h:i:sa")) - 1466000000) * rand(1,99);
echo json_encode($array);

?>
```

Didapat kode serial untuk perangkat jenis lampu dengan kode 48182304 dan kode serial untuk perangkat jenis steker dengan kode 122931489.

## 4.2 Pengujian Sistem

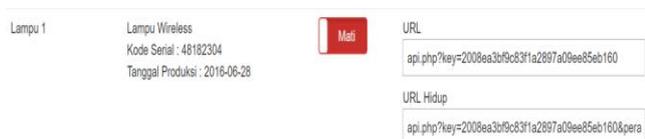
Pengujian dilakukan dengan memasukkan instruksi “Hidup” dan “Mati” untuk setiap perangkat lampu dan steker pada aplikasi antarmuka.

### 1. Lampu

Dilakukan pengujian kondisi lampu “Mati” pada lampu



Gambar 18 Kondisi Lampu Hidup



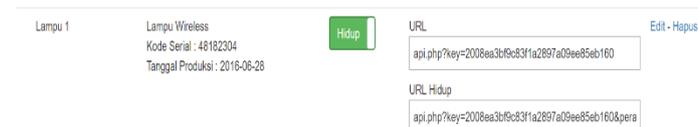
Gambar 15. Instruksi Lampu Mati

Hasil pengujian didapat kondisi Lampu tidak menyala. Dapat dilihat pada gambar 16.



Gambar 16. Kondisi Lampu Mati

Dilakukan pengujian kondisi lampu “Hidup” pada perangkat jenis lampu.



Gambar 17. Instruksi Lampu Hidup

Hasil pengujian didapat kondisi Lampu menyala. Dapat dilihat pada gambar 18.

### 2. Steker

Dilakukan pengujian kondisi steker “Mati”.



Gambar 19. Instruksi Steker Mati

Hasil pengujian didapat kondisi steker mati. Dapat dilihat pada gambar 20.



Gambar 20. Kondisi Steker Mati

Dilakukan pengujian kondisi steker “Hidup”



Gambar 21. Instruksi Steker Hidup

Hasil pengujian ditampilkan pada gambar 22.



Gambar 22. Kondisi Steker Hidup

Pada gambar 22 menampilkan hasil pengujian dan didapat kondisi steker hidup dengan paramater lampu indikator pada charger menyala.

### 3. Pengujian Request Data

Di uji lama waktu perangkat di aktifkan selama rentang waktu 20 menit dan 60 menit dan menghasilkan transmisi data yang berbeda setiap satuan menit. Didapat hasil transmisi data pada database sistem tidak melewati batas 110 data. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Pengujian dalam satuan per 5 menit

Waktu (m)	Jenis perangkat	
	Lampu	Steker
	Transmisi Request Data/5 menit	Transmisi Request Data/5 menit
0.00 – 5.00	110.16 data	114.1 data
5.01 – 10.00	114.6 data	114.6 data
10.01 – 15.00	111.6 data	115 data
15.01 - 20.00	113.2 data	115.4 data

### 4. Pengujian Respon Relay

Diuji perhitungan celah waktu jeda dalam menyalakan dan mematikan perangkat yang telah terhubung. Dapat dilihat pada Tabel 2, hasil pengujian ditampilkan dalam 5 kali percobaan untuk setiap perangkat.

Tabel 2. Pengujian Respon Relay per satuan detik

Percobaan	Steker (s)		Lampu(s)	
	Hidup(ON)	Mati(off)	Hidup(ON)	Mati(off)
1	1s	1s	1s	1s
2	1s	1s	2s	1s
3	1s	1s	1s	1s
4	1s	1s	1s	1s
5	1s	1s	1s	1s

Didapat hasil uji untuk setiap respon rata-rata melakukan aksi pada detik pertama untuk setiap perangkat dan disimpulkan

bahwa relay 100% merespon instruksi yang diberikan, namun terdapat jeda 1detik pada pengujian kedua pada perangkat jenis lampu dikarenakan besar tegangan yang diterima oleh lampu tidak stabil serta besarnya penggunaan *cache* pada *phpminiadmin* mengakibatkan respon aplikasi *web* menjadi cukup lama.

### 5. Pengujian Jarak Jangkauan Kendali Wireless (Node MCU) ke Server

Dilakukan pengujian pada jarak tertentu untuk mendapatkan hasil seberapa jauh jarak jangkauan sinyal wireless pada Node MCU dalam menghubungkan perangkat pada *server* Arduino Yun. Paramater jangkauan sinyal dikondisikan dalam kondisi sinyal kuat, sinyal sedang, dan sinyal lemah. Didapat hasil pengujian pada Tabel 3 dimana hasil uji jarak jangkauan sinyal kuat didapat pada jarak 1 – 11 meter, jarak jangkauan sinyal sedang didapat pada jarak 12 – 16 meter, jarak jangkauan sinyal lemah didapat pada jarak 17 – 20 meter dan jarak jangkauan sinyal tidak ada pada jarak 21 meter dan seterusnya.

Tabel 3. Hasil Uji Jarak Jangkauan Sinyal per satuan meter

No.	Jarak (meter)	Respon Sinyal (Status)
1	1 meter	√ ( Kuat )
2	2 meter	√ ( Kuat )
3	3 meter	√ ( Kuat )
4	4 meter	√ ( Kuat )
5	5 meter	√ ( Kuat )
6	6 meter	√ ( Kuat )
7	7 meter	√ ( Kuat )
8	8 meter	√ ( Kuat )
9	9 meter	√ ( Kuat )
10	10 meter	√ ( Kuat )
11	11 meter	√ ( Kuat )
12	12 meter	√ ( Sedang )
13	13 meter	√ ( Sedang )
14	14 meter	√ ( Sedang )
15	15 meter	√ ( Sedang )
16	16 meter	√ ( Sedang )
17	17 meter	√ ( Lemah )
18	18 meter	√ ( Lemah )
19	19 meter	√ ( Lemah )
20	20 meter	√ ( Lemah )
21	21 meter	-
22	22 meter	-

Dari hasil pengujian jarak jangkauan sinyal *server* yang dapat terhubung pada *wireless* perangkat smarphone jenis iphone 5s, iphone 6, iphone 7s dan xiaomi redminote 4 menunjukkan rata-rata kondisi sinyal yang sama pada status sinyal kuat diperoleh pada jarak 1-11 meter dengan acuan pada sedikitnya balok dinding atau tembok rumah yang tidak terlalu banyak. Rata-rata kondisi sinyal status sedang diperoleh pada jarak 12-16 meter dengan acuan tembok rumah yang cukup banyak. Rata-rata sinyal status lemah diperoleh pada jarak 17-20 meter dengan kondisi tembok rumah yang cukup banyak dan kondisi sinyal server tidak terdeteksi pada jarak > 20 meter dikarenakan faktor ketebalan dan banyaknya tembok rumah

## 6. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil percobaan pembuatan sistem kendali lampu dan steker terintegrasi menggunakan mikrokontroler berbasis *web service*, didapatkan hasil sebagai berikut:

1. Node MCU menjadi perantara dalam menghubungkan komunikasi antara perangkat dan *server*. Pengolahan perintah untuk membuka jalur tegangan pada relay direspon dalam 2 kondisi, dalam pengolahan perintah hidup/*on* menginstruksikan relay untuk membuka jalur tegangan dimana keadaan relay dalam posisi *normally open* sehingga perangkat teraliri listrik. Pengolahan instruksi mati/*off* menginstruksikan relay untuk menutup jalur tegangan.
2. Pemanfaatan arsitektur *RESTful* pada *web service* digunakan sebagai jalur komunikasi dan pengolahan perintah secara langsung ke *server* tujuan dengan mengakses API yang tersedia untuk setiap perangkat yang terdaftar.
3. Terdapat perbandingan nilai waktu respon pada perangkat jenis lampu dalam merespon keadaan hidup/*on* selama 1 detik dari batas perbandingan respon selama  $\leq 1$  detik dikarenakan perbedaan tegangan yang diterima oleh relay sehingga coil memberikan *feedback* pada *contact* yang

pergerakannya tergantung pada ada tidaknya arus listrik pada coil menjadi terhambat.

4. Jarak jangkauan sinyal Yun *server* pada setiap perangkat smarphone disebabkan oleh beberapa faktor berupa jarak akses smarphone dan banyaknya tembok yang menghalangi kuat tidaknya sinyal server.

### 5.2 Saran

Berdasarkan penelitian yang telah dilakukan mengenai sistem kendali lampu dan steker terintegrasi menggunakan mikrokontroler berbasis *web service*.

Disarankan untuk penelitian lebih lanjut yaitu :

- Implementasi penggunaan Arduino Yun sebagai *server* dapat dikombinasikan dengan mikrokontroler jenis lain untuk memudahkan kendali sistem dengan kapasitas dan hak akses yang lebih banyak dan cepat.
- Penerapan perangkat dapat ditambah dengan jenis perangkat lain seperti televisi, AC dan lain-lain.

## 6. DAFTAR PUSTAKA

- [1] Kurniawan, Eddi. 2013. Sistem Penerangan Rumah Otomatis dengan Sensor Cahaya Berbasis Mikrokontroler. Vol. 1 No. 2. Hal 1-10.
- [2] Ridwanda, H. 2014. Sistem Kendali Alat Listrik berbasis Waktu dengan ATMEGA8535, Vol 02. Hal 1-10.
- [3] <https://arduino.cc> diakses pada tanggal 4 februari 2016.
- [4] Rahman, A. M. (2013). Perancangan dan Implementasi *RESTful Web Service* untuk Game Sosial Food Merchant Saga pada Perangkat Android. *POMITS*, Vol. 2, No. 1.
- [5] Richardson, L., & Ruby, S. (2007). *RESTful Web Service*. Sebastopol,CA: O'Reilly Media, Inc.