

# Implementasi Struktur Data *Web service* Untuk Interoperabilitas Perangkat Pada Sistem Rumah Cerdas

Ulfa Muflika S, Sabriansyah Riqika Akbar, S.T, M.Eng., Adharul Muttaqin, S.T., M.T.  
Program Studi Teknik Komputer  
Fakultas Ilmu Komputer  
Universitas Brawijaya  
Malang, Indonesia  
ulfasefi@gmail.com

Abstrak– Suatu sistem smarthome merupakan gabungan dari entitas-entitas perangkat rumah yang serba cerdas yang masing-masing dapat melakukan komunikasi dengan yang lainnya. Metode komunikasi berbasis IP (Internet Protocol) banyak digunakan namun perlu dilakukan penambahan mekanisme untuk keberhasilan proses pertukaran data antar perangkat (node). Skripsi ini mengangkat topik implementasi struktur data web service untuk interoperabilitas perangkat rumah cerdas yang ditujukan menciptakan sistem otonom pada perangkat rumah cerdas. Implementasi web service sendiri bertujuan agar setiap perangkat rumah cerdas dapat saling bertukar data tanpa melibatkan end user.

Struktur data yang digunakan dalam penelitian ini adalah format data JSON, seperti yang telah diketahui JSON memiliki struktur data yang simple dan mudah dimengerti. Menggunakan arsitektur RESTful untuk komunikasi webservicenya dan pada implementasinya perangkat yang digunakan adalah Arduino Uno sebagai mikrokontroler dan modul esp8266 untuk komunikasi antar perangkat dan antar server. Dilakukan dua pengujian yang berbeda, yang pertama komunikasi antar node arduino dengan server PC yang menyimpan data dalam database dan yang kedua komunikasi antar perangkat arduino yang masing-masing berperan sebagai client-server.

Kemampuan interoperabilitas data diuji dengan melakukan dua jenis pengujian. Pengujian pertama dilakukan untuk mengetahui kemampuan interoperabilitas sistem web service dengan server PC, sedangkan pengujian kedua dilakukan untuk mengetahui keberhasilan pertukaran data antar node. Keberhasilan pengujian pertama ditandai dengan berhasilnya penyimpanan data yang dikirim node ke dalam sebuah database, sedangkan pada pengujian yang kedua ditunjukkan dengan keberhasilan perubahan data yang dilakukan antar node dengan memanfaatkan format JSON untuk pertukaran data.

Kata kunci —*web service*, struktur JSON, Arduino Uno, esp8266, library ArduinoJson

## I. LATAR BELAKANG

Smart home merupakan rumah cerdas dengan berbagai kecanggihan teknologi di dalamnya, sistem rumah dengan otomatisasi pada semua perangkat elektroniknya dengan tujuan memberikan kenyamanan, kemudahan, efisiensi energy, dan keamanan bagi penghuni rumah. Hal tersebut menunjukkan bahwa kecanggihan teknologi tidak hanya berdampak dan berkembang sekedar pada perangkat elektronik, namun perkembangan teknologi saat ini menunjukkan bahwa rumahpun dapat memberi peran khusus terhadap perkembangan teknologi yang semakin tidak dapat ditebak, sehingga manusia dapat merasakan segala kecanggihan hunian pada rumah cerdasnya.

Komponen utama dalam membangun rumah cerdas yaitu dengan jaringan cerdas, untuk mencapai kemudahan dalam pengoperasian perangkat-perangkat pada rumah cerdas dan menerapkan sistem otomatisasi hanya dapat dilakukan dalam jaringan cerdas. Metode komunikasi yang diterapkan saat ini adalah metode berbasis IP, pada kontrol jaringan menggunakan Internet Protocol (IP) tidak memiliki kemampuan untuk berkomunikasi dan beroperasi secara otonom karena komunikasi berbasis IP hanya mengatur pengenalan dan pengiriman data secara umum. Pada penelitian ini dikembangkan komunikasi

smarthome dengan pendekatan berorientasi layanan dan berbasis infrastruktur dengan membangun web service untuk memudahkan interoperation. Kalau sebelumnya sistem komunikasi smarthome hanya melibatkan komunikasi device dengan end user-nya, maka dengan web service diteliti teknologi untuk interaksi antar devicenya.

Web service dianggap menjadi solusi yang menjanjikan untuk menghubungkan semua komponen pada suatu sistem, kontribusi dari layanan yang diberikan oleh rumah-rumah pintar untuk utilitas proses bisnis, penyedia layanan energi dan operator sistemnya merupakan bagian dari penelitian komisi Proyek SmartHouse/SmartGrid yang sedang berlangsung di Eropa (Weidlich et al., 2009). World Wide Web Consortium (W3C) menjelaskan bahwa web service sebagai sistem perangkat lunak yang dirancang untuk mendukung interoperable interaksi mesin ke mesin di atas sebuah jaringan. Arsitektur web service mempunyai karakteristik yang dapat dilihat sebagai hal yang menguntungkan untuk membangun sebuah sistem terbuka. Setiap komponen yang berbeda dapat bervariasi dalam hal desain dan berjalan pada platform yang terpisah. Web service memungkinkan banyak vendor untuk berinteraksi satu sama lain berdasarkan standar high-level, dapat mendukung banyak aplikasi yang

berbeda, dan juga merupakan komponen layanan yang bertindak sebagai entitas otonom. Sehingga layanan yang diberikan oleh rumah-rumah pintar dapat menciptakan otonomi sistem dan bahkan perangkat otonomi (Warmer, 2009).

Bukan berarti menciptakan sistem perangkat otonom, end user tidak berperan dalam operasi sistem rumah cerdas, end user tetap menjadi kontrol utama dalam pengoperasian sistem tetapi antar perangkat yang terdapat pada rumah cerdas tetap bertanggung jawab atas tugasnya masing-masing, terkait keamanan dan pertukaran informasinya.

Web service dibangun untuk mendukung interoperabilitas perangkat pada sistem rumah cerdas dan diimplementasikan pada setiap perangkat dimana data informasi disimpan dalam suatu format tertentu dan fungsinya dapat dipinjam oleh aplikasi lain sehingga setiap perangkat yang terdapat pada rumah cerdas dapat saling berkolaborasi. Pembangunan web service ini juga dapat dimanfaatkan untuk memantau penggunaan energi dan memberikan informasi saran penggunaan/konsumsi energi perangkat rumah cerdas pada end user agar end user dapat mengorganisir penggunaan energi pada rumahnya secara lebih efektif jika dikembangkan lebih lanjut.

## II. RUMUSAN MASALAH

Berdasarkan latar belakang yang sudah diuraikan diatas, maka dapat dirumuskan suatu masalah sebagai berikut:

1. Bagaimana rancangan struktur data *web sevice* untuk interoperabilitas perangkat pada sistem rumah cerdas?
2. Bagaimana implementasi *web sevice* untuk interoperabilitas antar perangkat pada rumah cerdas?
3. Bagaimana pengujian *web sevice* dalam interoperabilitas perangkat rumah cerdas?

## III. TUJUAN

Tujuan dalam penulisan dan pelaksanaan penelitian ini terurai dalam tujuan umum dan khusus yaitu:

Tujuan umum:

Mengimplementasikan struktur data *web sevice* untuk interoperabilitas perangkat pada sistem rumah cerdas.

Tujuan khusus:

1. Merancang struktur data *web sevice* untuk interoperabilitas perangkat.
2. Mengimplementasikan *web sevice* untuk interoperabilitas antar perangkat pada sistem rumah cerdas.
3. Menguji sistem *web sevice* untuk interoperabilitas perangkat pada rumah cerdas.

## IV. BATASAN MASALAH

Adapun batasan masalah dalam perancangan struktur data *web sevice* untuk interoperabilitas sistem rumah cerdas yaitu:

1. Struktur data yang dipertukarkan pada sistem *web sevice* dalam format JSON.
2. Ruang lingkup penelitian ini diimplementasikan pada perangkat rumah cerdas (informasi lampu).
3. Komunikasi untuk bertukar data dilakukan dengan jaringan wifi.
4. Menggunakan board mikrokontroler Arduino Uno dan modul komunikasi Wi-Fi ESP8266.

## V. ANALISA KEBUTUHAN

Pada sub bab ini dijelaskan segala kebutuhan dalam perancangan sistem agar dapat berjalan sesuai dengan fungsionalitasnya. Penjelasan tentang kebutuhan antar muka sistem, kebutuhan fungsional dan non-fungsional dibahas disini.

### • Kebutuhan Antarmuka Eksternal

Pada sub bab ini dijelaskan tentang kebutuhan antarmuka eksternal yang meliputi kebutuhan antarmuka pengguna, kebutuhan perangkat keras dan perangkat lunak.

### • Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

1. Arduino Uno sebagai mikrokontroler yang mampu mengolah data dan operasi pertukarannya.
2. ESP8266 adalah modul komunikasi wifi yang mampu menjalin komunikasi dengan server.
3. Laptop/PC digunakan sebagai server yang mengolah request client dan menyimpan data pada database.
4. Jumper digunakan untuk menghubungkan antar arduino dengan modul wifi ESP8266.
5. Kabel USB digunakan untuk menyambungkan Arduino dengan laptop/PC.

### • Kebutuhan Perangkat Lunak

1. Arduino IDE sebagai editor program yang dikirim pada Arduino Uno dan menampilkan operasinya pada serial monitor.
2. XAMPP sebagai penyedia layanan server lokal yang menghubungkan server dengan database MySQL.
3. JetBrains PhpStorm sebagai editor program server dalam bahasa pemrograman php.
4. PuTTY sebagai serial monitor yang yang berkomunikasi dengan node arduino lainnya dan memunculkan hasil pembacaan arduino.
5. Windows 7 merupakan sistem operasi yang digunakan PC untuk menjalankan program Arduino IDE, XAMPP, JetBrains PhpStorm, dan PuTTY.

## VI. PERANCANGAN SISTEM

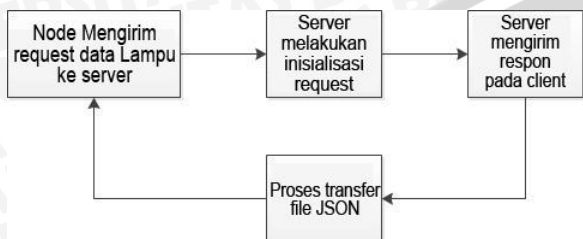
Pada perancangan sistem ini dibuat gambaran rangkaian sistem secara umum dan alur kerja sistem. Berikut gambaran perangkat komunikasi *web sevice* antar node dapat dilihat pada Gambar VI.1 berikut.





Gambar VI.1 Komunikasi Perangkat *Web service*

Sedangkan gambaran alur komunikasi sistem *web service* antar node dapat dilihat pada Gambar VI.2 seperti berikut.

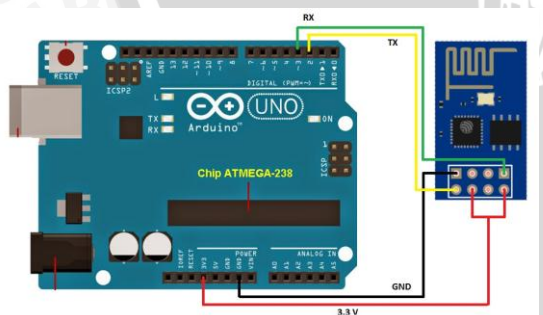


Gambar VI.2 Alur Komunikasi Sistem *Web service*

Deskripsi alur kerja sistem dijelaskan sebagai berikut:

1. Node mengirim request (berupa status lampu) ke mikrokontroler.
2. Mikrokontroler meneruskan data ke server *web service*.
3. Server mengolah request dengan *web service* dan meneruskan ke client.
4. Proses transfer file JSON ke client.

Rangkaian node client terdiri atas Arduino Uno sebagai mikrokontroler yang terhubung ke komputer server dan ESP8266 sebagai modul komunikasi via WIFI. Seperti ditunjukkan pada Gambar VI.3 dan sambungan pin ditunjukkan pada Table VI.1 berikut.



Gambar VI.3 Rangkaian node Arduino Uno dengan ESP8266

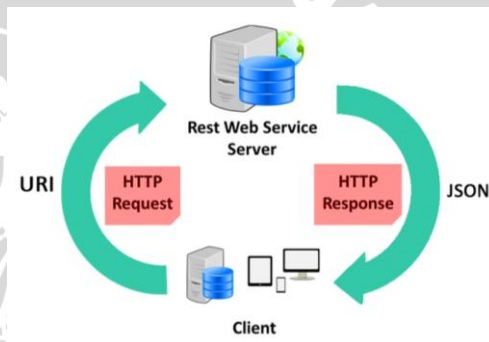
Table VI.1 Pin sambungan Arduino Uno dengan ESP8266

Arduino Uno	ESP8266	Warna
D2	TX	Kuning
D3	RX	Hijau

3V3	VCC	Merah
3V3	CH_PD	Merah
GND	GND	Hitam

a. Perancangan Metode Komunikasi dan Arsitektur Restful pada Sistem

Web service adalah metode komunikasi yang dipilih pada penelitian ini karena sesuai dengan kebutuhan pengguna dimana komunikasi dilakukan antar mesin (antar arduino dengan PC, arduino dengan arduino). Web service berjalan diatas protocol web (http) dengan mekanisme request-response dan memberikan response data berupa struktur data yang dapat ditentukan, dalam penelitian ini dipilih struktur dan format data JSON dengan arsitektur style REST. Berikut gambaran umum komunikasi Restful web service ditunjukkan pada Gambar VI.4.



Gambar VI.4 Gambaran umum komunikasi REST web service

Server mengidentifikasi request client melalui URI (Uniform Resource Identifier) sehingga dapat menuju target resource untuk berinteraksi. URI mengidentifikasi resource dan menyediakan ruang pengalamatan global untuk layanan dan discovery. Resource dimanipulasi menggunakan empat method yang telah ditetapkan, membaca, memperbaharui, dan menghapus operasi. PUT memperbaharui sumber daya baru, yang kemudian dapat dihapus menggunakan DELETE. GET mengambil keadaan sumber daya saat ini dan POST mengirim keadaan baru ke resource. Interaksi ke resource bersifat stateless (pesan permintaan mandiri) dan resource diakses dalam format JSON.

b. Perancangan Web Service pada sisi PC

Server web service pada PC ditulis dalam bahasa pemrograman PHP dengan memanfaatkan database MySQL sebagai penyimpan datanya. Program server pada PC menghubungkan setiap request dengan database MySQL. Pada penelitian ini database yang digunakan diberi nama rest dengan tabel data lampu seperti.



c. Perancangan Server Web Service pada Arduino

Server web service pada arduino ditulis menggunakan bahasa C/C++ dengan memanfaatkan library arduinoJson untuk encode data dalam format JSON. Pada server arduino perangkat mikrokontroler terhubung secara langsung dengan perangkat simulasinya (dalam penelitian ini menggunakan led) sehingga perangkat dapat dikontrol secara langsung dengan memanfaatkan pertukaran data Json.

d. Perancangan Struktur Data Web Service

Struktur data JSON relatif lebih terstruktur dan mudah dalam penulisannya. Pada penelitian ini bentuk penulisan struktur data JSON yang digunakan adalah berbentuk Objek. Objek terbentuk dari sepasang nama/nilai, dimulai dengan '{' kurung kurawal buka dan diakhiri '}' kurung kurawal tutup. Setiap nama diikuti dengan ':' titik dua dan setiap pasangan nama/nilai dipisah oleh ',' koma.

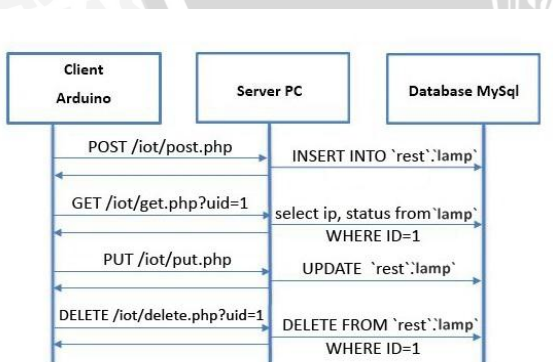
```

1 {
2   "id": "20",
3   "ip": "192.138.43.87",
4   "status": "off"
5 }
    
```

Gambar VI.5 Perancangan Struktur Data JSON

Pada Gambar VI.5 diatas menunjukkan struktur data JSON berbentuk objek dimulai dengan kurung kurawal buka dan diakhiri kurung kurawal tutup. Objek terbentuk dari sepasang nama/nilai, pada gambar diatas nama "id" diikuti tanda ':' dan pasangannya dengan nilai "20". Pada setiap pasangan nama/nilai dipisah menggunakan tanda ',' koma.

e. Alur Komunikasi Arduino dengan Server PC



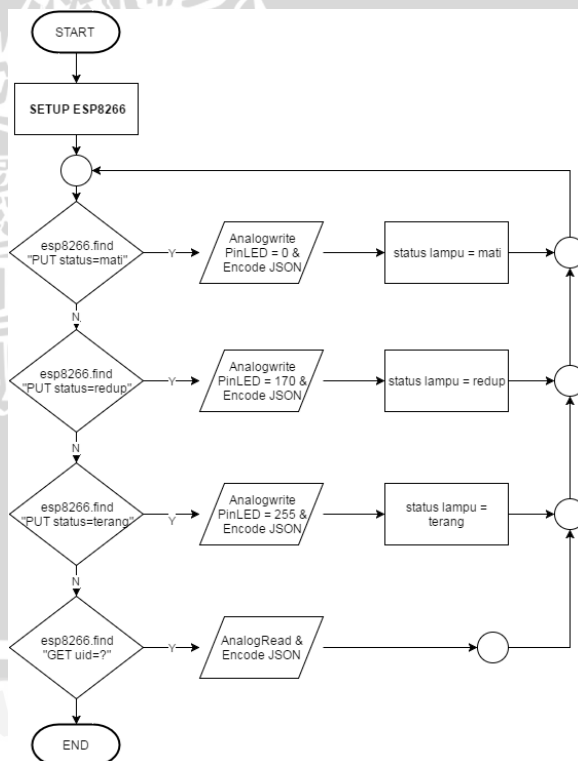
Gambar VI.6 Alur Komunikasi Arduino dengan Server PC

Pada Gambar VI.6 diatas menunjukkan alur komunikasi antar client arduino dengan server PC.

Arduino mengirim request dengan URI kepada server, server PC (web service) mengidentifikasi URI yang diterima dengan method web service. Request yang diterima server diidentifikasi sesuai method, jika yang diterima adalah GET, maka server mencari data pada database 'rest' sesuai dengan request ID. Method POST maka server memasukkan data baru yang dikirim client pada database 'rest'. Method PUT, server melakukan update data sesuai request id dan terakhir method DELETE, server melakukan delete data pada ID yang dituju dalam database 'rest'.

f. Alur Komunikasi Arduino dengan Arduino

Komunikasi antar arduino dilakukan dengan mempertukarkan data dalam format JSON melalui jaringan internet menggunakan modul komunikasi wifi ESP8266. Node arduino harus mampu berperan sebagai server dan client agar dapat saling berkomunikasi, keduanya harus terhubung pada jaringan yang sama. Node arduino yang berperan sebagai server menyimpan data led dalam variabel yang kemudian di encode dalam format JSON, ketika node tersebut menerima request maka node server harus dapat membaca request URI dari node client kemudian memberikan respon berupa data yang telah di encode tersebut dalam format JSON. Berikut gambaran flowchart pada sisi server arduino ditunjukkan pada Gambar VI.7.



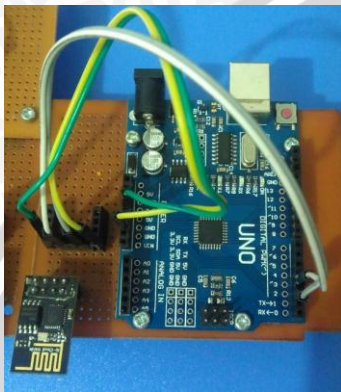
Gambar VI.7 Flowchart Algoritma Web Service pada Arduino



## VII. IMPLEMENTASI SISTEM

### a. Implementasi Rangkaian Node

Implementasi rangkaian setiap node yang berperan sebagai device yang saling berkomunikasi yaitu menggunakan Arduino Uno sebagai mikrokontroler yang mampu berperan sebagai client juga server dan berkomunikasi melalui jaringan wifi dengan modul komunikasi wifi esp8266. Server yang menyediakan database sebagai penyimpan data permanent terdapat pada PC/laptop yang juga berperan untuk menampilkan hasil *traffic* komunikasi perangkat. Berikut Implementasi rangkaian node ditunjukkan pada Gambar VII.1.

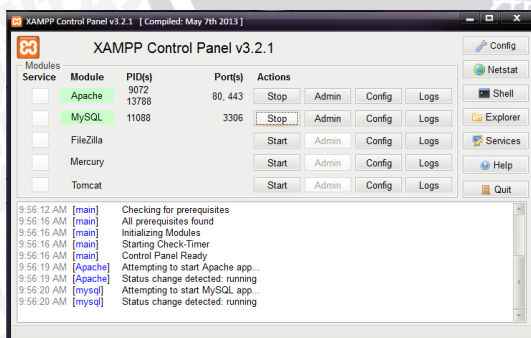


Gambar VII.1 Implementasi Rangkaian Node Arduino Uno dengan ESP8266

Pada implementasi rangkaian node Arduino Uno dengan ESP8266 dilakukan berdasarkan pada perancangan rangkaian node yang telah dijelaskan pada bab sebelumnya.

### b. Implementasi Web Service pada PC/Laptop

Implementasi web service pada PC/laptop ditulis menggunakan bahasa pemrograman server side, PHP: Hypertext Preprocessor dengan arsitektur REST dan memanfaatkan XAMPP sebagai penyedia server lokal yang menghubungkan dengan database MySQL.



Gambar VII.2 Implementasi XAMPP Sebagai Penyedia Server Lokal

Pada Gambar VII.2 diatas menunjukkan XAMPP sedang menjalankan module Apache sebagai layanan

server lokal dan module MySql sebagai penghubung program menuju database MySql.

Database MySql dapat diakses melalui browser dengan mengetikkan URL localhost/phpmyadmin kemudian muncul web page database MySql. Implementasi program server ditulis menggunakan editor program PhpStorm dengan ekstensi .php.

### c. Implementasi Web Service Pada Perangkat Arduino

Implementasi web service pada arduino ditentukan dengan menjadikan salah satu node arduino berperan sebagai server yang dapat melayani request client node arduino lainnya dan memberikan respon berupa data JSON. Implementasi setting sever pada node arduino menggunakan modul esp8266 dengan setting command "AT+CIPSERVER=1, 80", parameter 1 yang berarti membuat modul esp8266 menjadi server dan dapat menerima request dari client, parameter 80 adalah port yang digunakan sebagai komunikasinya. Berikut implementasi program server pada arduino ditunjukkan pada Gambar VII.3.



Gambar VIII.2 Hasil Pengujian Request Post

1. Pengujian Komunikasi Web Service Antar Server PC dan Arduino (server-client)

```

void postdata(){
  ip = "192.168.43.66";
  String data = "ip="+ ip +"&status=mati";
  String cmd = "POST /lot/post.php/ HTTP/1.1\r\n";
  cmd += "Host: 192.168.43.65\r\n";
  cmd += "Content-Length: ";
  cmd += data.length();
  cmd += "\r\n";
  cmd += "Content-Type: application/x-www-form-urlencoded\r\n\r\n";
  cmd += data;

  String dataput = "ip="+ ip +"&status=mati";
  esp8266.print( "AT+CIPSEND=0," );
  esp8266.println( cmd.length() );
  espdo(2000);

  esp8266.print(cmd);
}
    
```

Gambar VIII.1 Program Post Client

Gambar VIII.1 diatas adalah program post yang dikirim melalui Arduino dengan ip yang telah didefinisikan sebelumnya dan status lampu dalam keadaan mati

```

AT+CWMODE=1
OK
AT+CWJAP="seviI","sevistring"
WIFI DISCONNECT
WIFI CONNECTEDAT+CIPMUX=1
OK
AT+CIPSTART="TCP","192.168.43.65",80
0,CONNECT
OK
AT+CIPSEND=0,150
OK
>
Recv 150 bytes

SEND OK

+IPD,0,236:HTTP/1.1 200 OK
Date: Wed, 19 Oct 2016 00:08:45 GMT
Server: Apache/2.4.9 (Win32) OpenSSL/1.0.1g PHP/5.5.11
X-Powered-By: PHP/5.5.11
Content-Length: 46
Content-Type: application/json

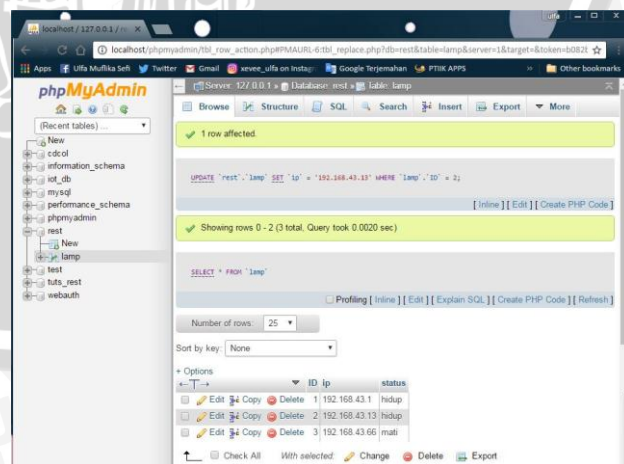
{"status":1,"msg":"Done added data at id '3'"}0,CLOSED
    
```

Bedasarkan pengujian request POST yang telah dikirim, Gambar VIII.2 diatas menunjukkan hasil dari setting komunikasi module wifi esp8266. AT+CWMODE=1 setting untuk mode station dan memberikan respon OK, AT+CWJAP setting untuk menghubungkan dengan hotspot yang tersedia memberikan respon WIFI CONNECTED yang berarti telah terhubung dengan jaringan wifi. AT+CIPMUX=1 untuk setting multiple connection memberikan respon OK, AT+CIPSTART memulai koneksi kepada ip server dan memberikan respon 0, CONNECT yang berarti komunikasi dengan id 0 telah terhubung dengan ip tujuan. Barulah pada command AT+CIPSEND=0, 150 request URI telah dikirim pada server dengan mode 0 (single connection) dan 150 adalah length data yang dikirim dalam satuan byte, lalu memberikan respon Recv 150 bytes SEND OK yang berarti data telah diterima dan status terkirim.

Respon yang diterima ketika request berhasil dikirim adalah "+IPD,<id>,<len data yang diterima>:<data HTTP respon 200 OK>" respon HTTP dengan status 200, request berhasil dieksekusi. Pada respon line berikutnya terdapat informasi date, server, resource server, dan informasi panjang data yang dikirim dan formatnya. Pada hasil diatas panjang data yang diterima 46 byte dengan format JSON dan pada line paling bawah adalah respon data JSON yang dikirim oleh server dengan status 1 yang berarti data telah berhasil ditambahkan pada id 3 dan esp8266 melakukan close connection.

Berikut hasil data yang telah di post dalam database dilihat melalui browser. Data yang ditambahkan dengan ip 192.168.43.66 dan status mati telah ditambahkan pada id 3 dapat dilihat pada

Gambar VIII.3.



Gambar VIII.3 Hasil Post Data pada Database MySQL



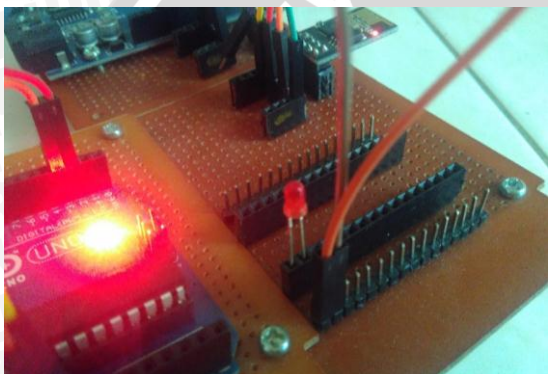


## 2. Pengujian Komunikasi Web Service Antar Perangkat Arduino (server-client)

Pada pengujian komunikasi web service antar perangkat arduino terdapat dua node perangkat arduino yang salah satunya berperan sebagai server dan satunya berperan sebagai client. Method yang dikirim sama dengan program client menuju PC, yaitu dengan URI yang didefinisikan oleh server pada arduino namun hanya terdapat dua method pada pengujian antar perangkat arduino, yaitu method PUT dan GET karena perangkat real tidak mungkin di POST dan DELETE.

### Hasil dan Analisa Pengujian

Berdasarkan pengujian yang telah dilakukan pada kedua perangkat node arduino, berikut hasil komunikasi web service antar keduanya. Perangkat node server yang telah dihubungkan dengan led sebagai indikasi keberhasilan komunikasi ditunjukkan pada Gambar VIII.4 berikut dengan kondisi awal led dalam keadaan mati.



Gambar VIII.4 Perangkat LED pada Server Arduino

Pada pengujian pertama node client mengirim request PUT pada server dengan URI "PUT uid=1&&status=terang" untuk membuat led berubah kedalam kondisi terang, berikut code program yang dikirim oleh client ditunjukkan pada Gambar VIII.5.

```

void putdata() {
  String PUT = "PUT uid=1&&status=terang" ;
  String cmd = PUT + "\r\n\r\n";
  esp8266.print( "AT+CIPSEND=0," );
  esp8266.println( cmd.length() );
  espdo(2000);
  esp8266.print(cmd);
  espdo(2000);
}
    
```

Gambar VIII.5 Program Request PUT Data untuk Server Arduino

Pada Gambar VIII.6 berikut adalah hasil dari pengujian request PUT pada serial monitor server dengan Putty.

```

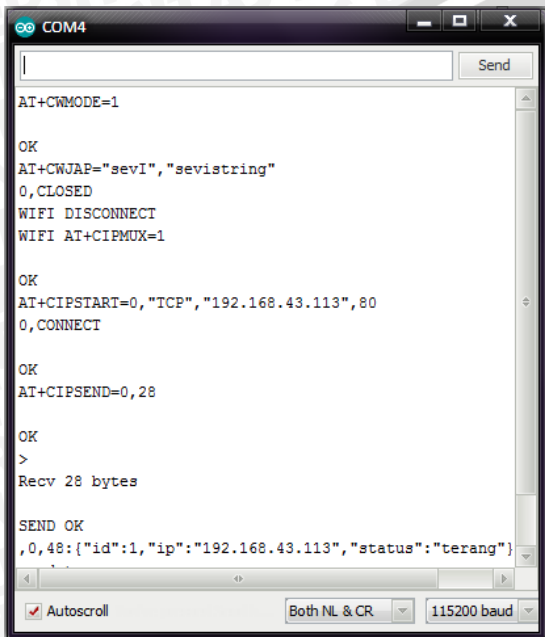
COM13 - Putty
AT+CWMODE=1
OK
AT+CWJAP="sevi","sevistring"
0,CONNECT FAIL
OKFI DISCONNECT
AT+CIPMUX=1
OK
AT+CIFSR
+CIFSR: STAIP,"192.168.43.113"
+CIFSR: STAMAC,"5c:cf:AT+CIPSERVER=1,80
no change
OK
--
--
--
--
--
request PUT diterima
{"id":1,"ip":"192.168.43.113","status":"terang"}
AT+CIPSEND=0,48
OK
>
Recv 48 bytes
SEND OK
lampu dalam kondisi terang
    
```

Gambar VIII.6 Hasil Pengujian Serial Monitor Server

Hasil pengujian diatas menunjukkan setting node sebagai server dengan status OK. Setting AT+CWMODE=1 sebagai station memberi respon status OK, AT+CWJAP untuk connect dengan hotspot dan status WIFI CONNECTED, AT+CIPMUX=1 untuk multiple connection, AT+CIFSR untuk mengetahui ip station dengan hasil ip "192.168.43.113" dan AT+CIPSERVER=1, 80" untuk setting mode server dengan port 80 dan hasilnya "no change" karena telah disetting sebelumnya sebagai server maka status respon nya berupa status "no change"/"tidak diganti".

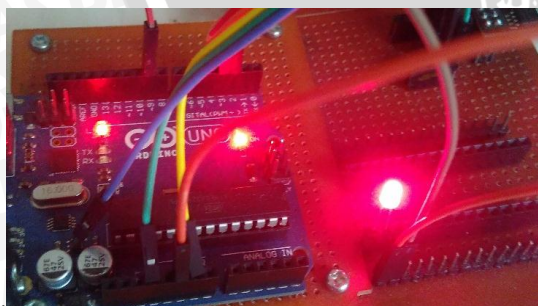
Tanda “-“ adalah perulangan ketika server menunggu request dari client, saat client mengirim request "PUT uid=1&&status=terang" maka server memberi respon seperti yang ditunjukkan pada gambar diatas dan melakukan encode data dalam format JSON yang kemudian dikirim menuju client.

Berikut hasil pengujian pada sisi client yang menerima respon data JSON dari server dengan panjang data 48 byte dan data JSON {"id": 1, "ip": "192.168.43.113", "status": "terang"} ditunjukkan pada Gambar VIII.78.



Gambar VIII.7 Hasil Pengujian PUT data Client

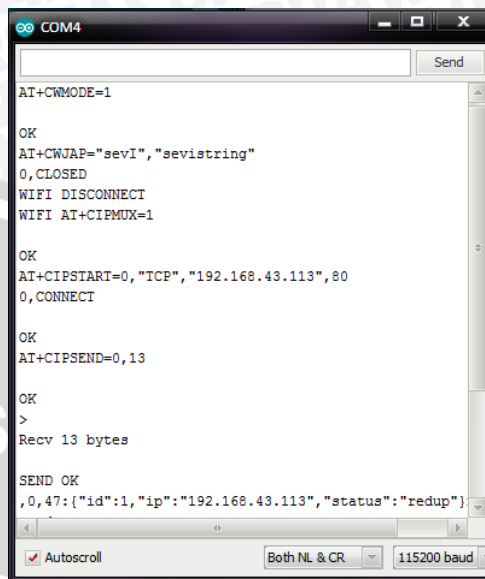
Hasil pengujian pada perangkat led nya sendiri pun dapat berjalan dengan baik. Berikut hasil pada perangkat led yang telah berubah dalam keadaan terang ditunjukkan pada Gambar VIII.8.



Gambar VIII.8 Perangkat LED server dalam keadaan terang

Pengujian berikutnya adalah method get, disini client mengirim request dengan URI "GET uid=1" pada server. Berikut hasil pengujian dari request GET client, setup modul esp8266 nya masih sama, disini panjang request yang dikirim hanya 13 byte. Terlihat pada Gambar VIII.9 dibawah request terkirim dengan baik dan client dapat

menerima kembalian server berupa data JSON dengan baik juga.



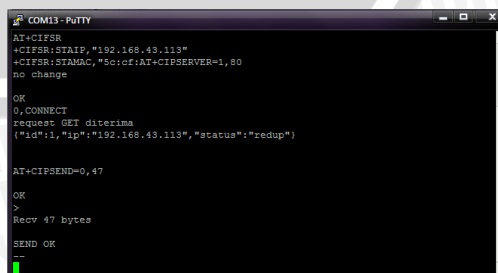
Gambar VIII.9 Hasil Pengujian GET data pada Client

Hasil dari status data diatas didapat berdasarkan pembacaan nilai pwm LED, seperti yang ditunjukkan pada Gambar VIII.10 syntax program server berikut. Jika nilai pwm led = 0, status led dalam kondisi mati, jika nilai pwm 255, status led dalam kondisi terang, dan jika selain nilai tersebut status led dalam kondisi redup.

```
else if(esp8266.find("GET uid=1")){
    Serial.println("request GET diterima");
    if(pinLED == 0){
        val= "mati";
    }else if(pinLED == 255){
        val = "terang";
    }else{
        val = "redup";
    }
}
```

Gambar VIII.10 Program Pembacaan Nilai LED

Pada Gambar VIII.11 dibawah ini adalah hasil pengujian method GET pada serial monitor server, server dapat menerima request dan dapat mengirim data JSON dengan sangat baik. Pada sisi server juga terdapat fungsi encode data JSON, dan disini proses encode berjalan dengan baik.



Gambar VIII.11 Hasil Pengujian GET request Server Arduino



## IX. KESIMPULAN

Berikut kesimpulan yang dapat diambil dari penelitian ini:

1. Perancangan struktur data web service untuk interoperabilitas perangkat pada sistem rumah cerdas dapat berjalan baik dengan mempertukarkan data JSON dan antar perangkat saling memanfaatkan layanan yang dimiliki oleh masing-masing perangkat sehingga interoperabilitas sistem dapat tercapai.
2. Implementasi sistem web service pada server PC/Laptop dengan menggunakan XAMPP sebagai penyedia server lokal dan ditulis dalam bahasa pemrograman PHP sehingga mampu menjalankan 4 method utama web service, yaitu POST, GET, PUT dan DELETE. Client arduino menggunakan modul komunikasi esp8266 yang berperan sebagai station mengirim fungsi request method web service pada server PC/laptop dan menerima data JSON melalui modul esp8266. Server web service pada perangkat arduino menggunakan library ArduinoJson untuk encode data dalam format JSON dan dikirim pada perangkat arduino lainnya yang berperan sebagai client. Pada implementasinya, sistem web service telah mencapai interoperability sistem dimana semua dapat berkomunikasi dan mempertukarkan data yang sama.
3. Kemampuan interoperabilitas data diuji dengan melakukan dua jenis pengujian. Pengujian pertama dilakukan untuk mengetahui kemampuan interoperabilitas system web service dengan server PC, sedangkan pengujian kedua dilakukan untuk mengetahui keberhasilan pertukaran data antar node. Keberhasilan pengujian pertama ditandai dengan berhasilnya penyimpanan data yang dikirim node ke dalam sebuah database, sedangkan pada pengujian yang kedua ditunjukkan dengan keberhasilan perubahan data yang dilakukan antar node dengan memanfaatkan format JSON untuk pertukaran data.

## X. DAFTAR PUSTAKA

- Aful, Interoperability Working Group. Definisi dari Interoperabilitas. [online] Tersedia di: <<http://interoperability-definition.info/id/>> [Diakses 30 Januari 2017]
- Arduino Corporation, 2016. Arduino Nano. [online] Tersedia di: <<https://www.arduino.cc/en/Main/ArduinoBoardNano>> [Diakses 6 Februari 2016]
- Febrika Perpustakaanku, Sejarah Perkembangan Mikrokomputer. [pdf] Tersedia di: <[http://febrika.16mb.com/PERPUSTAKAANKU/ab1-sejarah\\_perkembangan\\_mikrokomputer.pdf](http://febrika.16mb.com/PERPUSTAKAANKU/ab1-sejarah_perkembangan_mikrokomputer.pdf)> [Diakses 7 Februari 2016]
- Gozali, A.A., 2014. MicroComputer, MiniComputer, MidRange, Mainframe. [online] Tersedia di: <<http://andriahmadgozali.ilearning.me/2014/02/19/microcomputer-minicomputer-midrange-mainframe/>> [Diakses 7 Februari 2016]
- Hafiz, A., 2014. interoperabilitas. [electronic print] Tersedia di: <<https://www.scribd.com/doc/204501171/Makalah-Pengertian-Interoperabilitas-docx>> [Diakses 30 Januari 2017]
- Hamblen, J., Using the ESP8266 to add Wi Fi to the mbed LPC1768. [online] Tersedia di: <[https://developer.mbed.org/users/4180\\_1/notebook/using-the-esp8266-with-the-mbed-lpc1768/](https://developer.mbed.org/users/4180_1/notebook/using-the-esp8266-with-the-mbed-lpc1768/)> [Diakses 7 Februari 2016]
- Hendriono, D., 2014. Mengenal Arduino Nano. [online] Tersedia di: <<http://www.hendriono.com/blog/post/mengenal-arduino-nano>> [Diakses 7 Februari 2016]
- Nurds, 2014. ESP8266. [online] Tersedia di: <<https://nurdspace.nl/ESP8266>> [Diakses 6 Februari 2016]
- Oracle, 2013. The Java EE 6 Tutorial. [online] Tersedia di: <<http://docs.oracle.com/javase/6/tutorial/doc/gijqy.html>> [Diakses 6 September 2016]
- Prasetyo, N., Perkembangan Microcontroler AVR. [pdf] Tersedia di: <[http://lecturer.polindra.ac.id/~shumaru/tugas\\_mata\\_kuliah/Aplikasi\\_mikroprocessor/Tugas%20aplikasi%20mikro.pdf](http://lecturer.polindra.ac.id/~shumaru/tugas_mata_kuliah/Aplikasi_mikroprocessor/Tugas%20aplikasi%20mikro.pdf)> [Diakses 8 Februari 2016]
- Putra, R.S., File : UNIKOM\_Rian Satria Putra\_bab ii.pdf. [pdf] Elib Unikom. Tersedia di: <[http://elib.unikom.ac.id/files/disk1/597/jbptunikompp-gdl-riansatria-29820-9-unikom\\_r-i.pdf](http://elib.unikom.ac.id/files/disk1/597/jbptunikompp-gdl-riansatria-29820-9-unikom_r-i.pdf)> [diakses 6 Februari 2016]
- Setiawan, A., 2013. Transiskom Portal Komputer dan teknologi. Mengenal Home Automation [online], <<http://www.transiskom.com/2013/01/mengenal-home-automation.html>> [diakses 6 Februari 2016]
- Setyawan, G.E., 2014. Mikrokomputer: Perkembangan Mikrokomputer. [pdf] Tersedia di: <<http://gembong.lecture.ub.ac.id/files/2014/09/2-perkembangan-mikrokomputer.pdf>> [Diakses 7 Februari 2016]
- Utama, Y., 2006. Pengenalan Web Service. [electronic print] Tersedia di: <<http://www.unsri.ac.id/upload/arsip/Web%20Service%20-%20NuSOAP%20-%20WSDL.doc>> [Diakses 6 Februari 2016]
- Warmer C.J., J.K. Kok, S. Karnouskos, A. Weidlich, D. Nestle, P. Selzam, J. Ringelstein, A. Dimeas and S. Drenkard, 2009. "Web services for integration of smart houses in the smart grid", Grid-Interop Forum.
- Weidlich, A. and S. Karnouskos, 2009. "Integrating Smart Houses with the Smart Grid Through Web Services for Increasing Energy Efficiency". 10th IAEE European Conference.