

repository.ub.ac.id

**IMPLEMENTASI STRUKTUR DATA *WEB SERVICE* SEBAGAI  
METODE KOMUNIKASI PERANGKAT/SENSOR PADA SISTEM  
RUMAH CERDAS**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
M. Kholil Gibran  
NIM: 115060900111025



**SISTEM KOMPUTER  
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
TAHUN  
2015**

## PENGESAHAN

IMPLEMENTASI STRUKTUR DATA *WEB SERVICE* SEBAGAI METODE KOMUNIKASI  
PERANGKAT/SENSOR PADA SISTEM RUMAH CERDAS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

M. Kholil Gibran

NIM: 115060900111025

Skripsi ini telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizgika Akbar, S.T, M.Eng

NIP: 19820809 201212 1 004

Adharul Muttaqin, S.T, M.T

NIP: 19760121 200501 1 001

Mengetahui

Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.

NIP: 19670801 199203 1 001

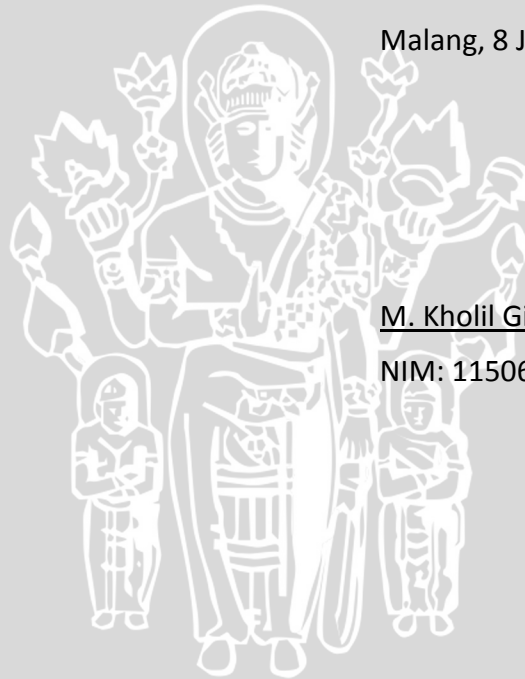


## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 8 Januari 2016



M. Kholil Gibran

NIM: 115060900111025

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul **“IMPLEMENTASI STURKTUR DATA WEB SERVICE SEBAGAI METODE KOMUNIKASI PERANGKAT/SENSOR PADA SISTEM RUMAH CERDAS”** ini dapat terselesaikan. Skripsi ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar Sarjana Pendidikan Strata Satu pada Program Studi Sistem Komputer Fakultas Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Penghargaan dan terima kasih yang setulus-tulusnya kepada Ayahanda tercinta Drs. Moch. Shochib dan Ibunda yang kusayangi Dra. Siti Ulya serta kakak tersayang Dyah Zahrotul Millah, S.si, serta keponakan yang kusayangi Farda Nafisatus Safiyah Az-zahra semoga Allah SWT melimpahkan Rahmat, Kesehatan, Karunia dan keberkahan di dunia dan di akhirat atas budi baik yang telah diberikan kepada penulis.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Prof. Dr. Ir. Mohammad Bisri, MS selaku Rektor Universitas Brawijaya
2. Bapak Ir Sutrisno, M.T. selaku Dekan Fakultas Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
3. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku pembimbing I yang telah banyak membantu dalam pengerjaan skripsi ini.
4. Adharul Muttaqin, S.T, M.T selaku pembimbing II yang telah banyak membantu dalam pengerjaan skripsi ini.
5. Rizky Romadhoni, S.Kom, Januar Rizky Pratama, Muh. Abduh Mujib selaku tim sukses Ploper.
6. Loki Sudiro, Rahadian Sayogo, Sunu Diaz W, Adi Santoso, Bagus P, Yuni P, Anake D, Diny M, Rezky J, M. Afief, Arifin, Rizky P, Labib, M. Agil, Ryan A dan seluruh teman-teman Sistem Komputer 2011 yang telah menemani dan memberikan motivasi selama perkuliahan.

Malang, 8 Januari 2016

penulis

M. Kholil Gibran

## ABSTRAK

Rumah Cerdas merupakan sebuah lingkungan tempat tinggal yang dilengkapi dengan komputasi dan komunikasi namun dapat terintegrasi dengan penghuninya. Pada sistem Rumah Cerdas berkembang berbagai peralatan dengan vendor yang berbeda-beda, akan tetapi keinginan konsumen untuk merealisasinya sangat sedikit sehingga pengembangan aplikasinya pun juga sedikit (Reynold, 2010). Kesulitan komunikasi antar berbagai perangkat atau aplikasi pada sistem Rumah Cerdas dapat menggunakan metode *Web Service*. *Web Service* dapat diakses oleh aplikasi yang berjalan pada platform yang berbeda. *Web Service* adalah sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan software aplikasi lain dengan menggunakan pesan berbasis XML melalui protokol internet

Pada sistem ini digunakan simulasi kontrol peralatan listrik dan monitoring sensor yang relevant dengan peralatan listrik yang dikontrol. Metode komunikasi arduino dengan raspberry pi menggunakan HTTP POST, dimana input dari perangkat/sensor dimasukan ke database kemudian diolah dalam bentuk file JSON, data keluaran dari sensor dikirimkan ke Client melalui *Web Service*. JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman *JavaScript*.

Dengan adanya penelitian ini diharapkan dapat memudahkan *client* mendapatkan informasi nilai sensor secara *real time*. Data yang ditampilkan merupakan 10 data terakhir yang dibaca sensor sehingga data client selalu mengetahui kondisi suhu.

Kata kunci: *HTTP Post*, *JSON (JavaScript Object Notation)*, *Arduino Uno*, *Rumah Cerdas*, sensor LM35.

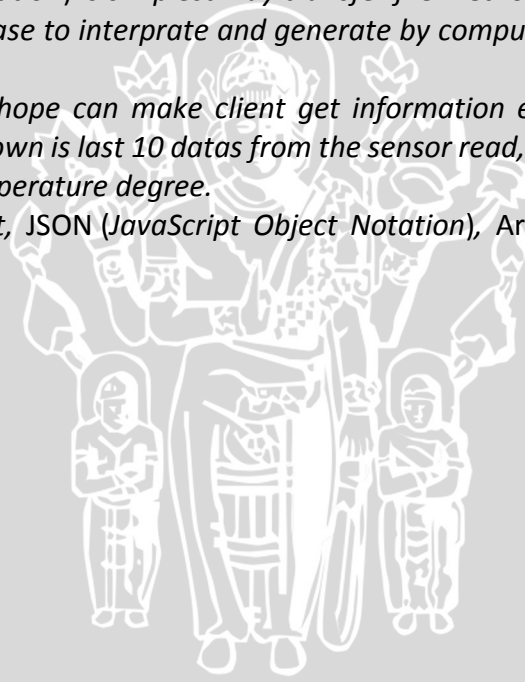
## ABSTRACT

Smart home is a place to live which full of computation and communication, but could be integrated with the owner. The system of smart home development is grown rapidly, but still few developer interested to build (Reynold, 2010). The smart home is highest trend in this century. Because the system has web service who make the owner easy to control and monitoring his house. Web service is an application identified by URI and has simple interface that can be interpreted and understand by XML, also support direct interaction among applications base on internet protocol.

The system was build as simulation electric device control and sensor monitoring. Communication method of Arduino with raspberry pi are using HTTP POST, where the input from sensor delivered into database then processed to create JSON file and the output is sent to client through web service. JSON (JavaScript Object Notation) is simplest way transfer file method, because easy to read and write, also ease to interprate and generate by computer. The script was build base Javascript.

This research was hope can make client get information easier from sensor realtime. Data was shown is last 10 datas from the sensor read, so make the client always knows the temperature degree.

Kata kunci: HTTP Post, JSON (JavaScript Object Notation), Arduino Uno, Smart Home , sensor LM35.



## DAFTAR ISI

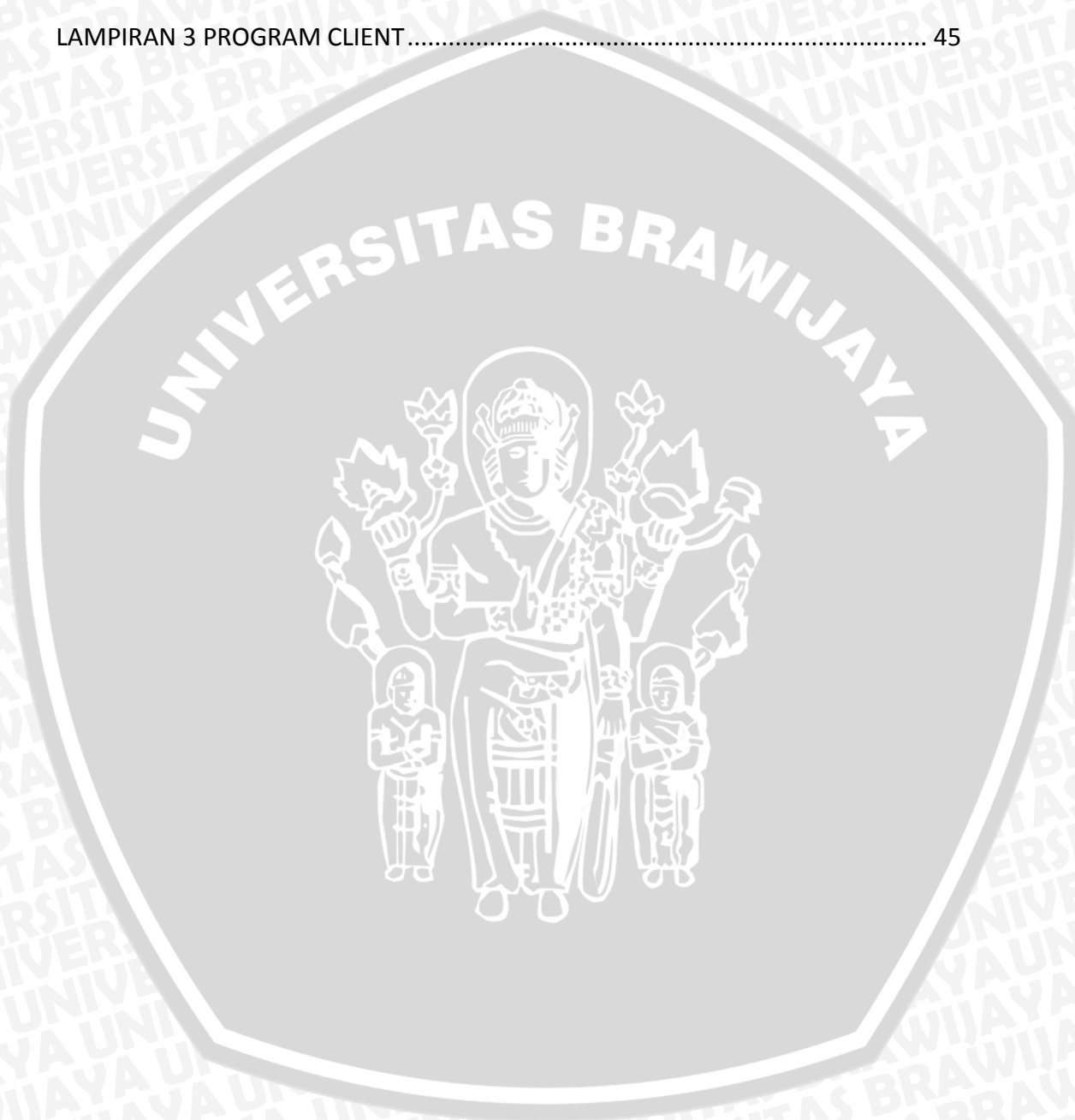
PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i> .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN .....	xii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	2
1.6 Sistematika pembahasan.....	2
<b>BAB 2 Kajian pustaka dan dasar teori .....</b>	<b>4</b>
2.1 Kajian pustaka .....	4
2.2 Dasar Teori.....	4
2.2.1 <i>Web Service</i> .....	4
2.2.2 Mini PC Raspberry Pi .....	6
2.2.3 Sensor LM35.....	7
2.2.4 Arduino UNO .....	8
2.2.5 HTTP POST.....	8
2.2.6 Python .....	8
2.2.7 Ethernet Shield.....	9
<b>BAB 3 METODOLOGI penelitian .....</b>	<b>11</b>
3.1 Perumusan Masalah .....	11
3.2 Penentuan Tujuan Penelitian.....	12

3.3 Studi Literatur .....	12
3.4 Penetapan Kebutuhan .....	12
3.4.1 Kebutuhan Fungsional.....	12
3.4.2 Kebutuhan Non Fungsional.....	12
3.5 Perancangan Sistem.....	12
3.6 Implementasi Sistem .....	13
3.7 Pengujian dan Analisis .....	13
3.8 Kesimpulan.....	14
<b>BAB 4 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>15</b>
4.1 Perancangan Sistem.....	15
4.1.1 Diagram alir keseluruhan sistem.....	15
4.1.2 Analisis perancangan sistem .....	16
4.2 Perancangan perangkat pembaca dan pengirim data sensor .....	16
4.2.1 Diagram alir pembacaan data sensor LM35 .....	17
4.2.2 Diagram alir pengiriman data dari arduino ke Server.....	18
4.3 Perancangan perangkat penerimaan dan pengiriman data.....	19
4.3.1 Diagram alir penerimaan data sensor dari arduino.....	19
4.3.2 Diagram alir pengiriman data sensor ke client .....	20
4.4 Perancangan struktur data sistem.....	20
4.5 Implementasi Sistem .....	21
4.5.1 Bagian pembaca dan pengirim data sensor.....	21
4.5.2 Bagian Web Server .....	23
4.5.3 Bagian data client.....	30
<b>BAB 5 PENGUJIAN .....</b>	<b>33</b>
5.1 Pengujian sistem .....	33
5.1.1 Pengujian Sensor.....	33
5.1.2 Pengujian Arduino.....	34
5.1.3 Pengujian Raspberry Pi .....	35
5.1.4 Pengujian Client .....	36
5.2 Analisis .....	37
<b>BAB 6 Penutup .....</b>	<b>38</b>
6.1 Kesimpulan.....	38





6.2 Saran .....	38
DAFTAR PUSTAKA.....	39
LAMPIRAN 1 PROGRAM ARDUINO .....	40
LAMPIRAN 2 PROGRAM SERVER.....	42
LAMPIRAN 3 PROGRAM CLIENT.....	45



## DAFTAR TABEL

Tabel 1 Hasil pembacaan sensor LM35.....	33
Tabel 2 Hasil konversi dalam °C .....	34



## DAFTAR GAMBAR

Gambar 2. 1 Perbedaan SOAP dan REST .....	5
Gambar 2. 2 Bentuk objek pada JSON .....	6
Gambar 2. 3 Bentuk array pada JSON .....	6
Gambar 2. 4 Mini pc raspberry pi .....	7
Gambar 2. 5 Sensor LM35 .....	7
Gambar 2. 6 Arduino UNO .....	8
Gambar 2. 7 Arduino Ethernet LAN .....	10
Gambar 3. 1 Alur Pelaksanaan.....	11
Gambar 3. 2 Diagram Blok .....	13
Gambar 4. 1 Diagram blok sistem.....	15
Gambar 4. 2 Simulasi perangkat arduino.....	17
Gambar 4. 3 Diagram alir pembacaan data sensor.....	18
Gambar 4. 4 Diagram alir pengiriman paket data ke server.....	19
Gambar 4. 5 Diagram alir penerimaan data sensor.....	20
Gambar 4. 6 Diagram alir raspberry pi.....	20
Gambar 4. 7 Diagram alir mekanisme pengiriman data.....	21
Gambar 4. 8 Rangkaian Arduino dan LM35 .....	22
Gambar 4. 9 Pembacaan sensor menggunakan avometer.....	23
Gambar 4. 10 Implementasi raspberry pi.....	25
Gambar 4. 11 Implementasi keseluruhan sistem .....	25
Gambar 4. 12 Flowchart main program.....	26
Gambar 4. 13 Flowchart get_tasks .....	27
Gambar 4. 14 Flowchart get_task.....	28
Gambar 4. 15 Flowchart api_message .....	29
Gambar 4. 16 Diagram implementasi client .....	30
Gambar 4. 17 Flowchart tugas client .....	31
Gambar 5. 1 Pengiriman data.....	34
Gambar 5. 2 Pengujian pernerima data.....	35
Gambar 5. 3 Pengujian pengiirman data .....	36
Gambar 5. 4 Pengujian client .....	37

## DAFTAR LAMPIRAN

LAMPIRAN 1 PROGRAM ARDUINO .....	40
LAMPIRAN 2 PROGRAM RASPBERRY PI .....	42
LAMPIRAN 3 PROGRAM CLIENT .....	45



## BAB 1 PENDAHULUAN

Bab ini menjelaskan latar belakang dari penelitian, rumusan masalah yang harus diselesaikan, batasan masalah yang ada pada penelitian, tujuan dari penelitian, manfaat dari penelitian yang dilakukan, dan sistematika penulisan dari penelitian.

### 1.1 Latar belakang

Rumah Cerdas merupakan sebuah lingkungan tempat tinggal yang penuh dengan komputasi dan komunikasi namun dapat terintegrasi dengan penghuninya. Pada sistem Rumah Cerdas berkembang berbagai peralatan dengan vendor yang berbeda-beda, akan tetapi keinginan konsumen untuk merealisasinya sangat sedikit sehingga pengembangan aplikasinya pun juga sedikit (Reynold, 2010). Kesulitan komunikasi antar berbagai perangkat atau aplikasi pada sistem Rumah Cerdas dapat menggunakan metode *Web Service*. *Web Service* dapat diakses oleh aplikasi yang berjalan pada platform yang berbeda.

*Web Service* adalah sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan software aplikasi lain dengan menggunakan pesan berbasis XML melalui protokol internet. *Web Service* tidak terpengaruh oleh platform, *Web Service* akan menyediakan metode-metode yang dapat diakses oleh jaringan.. (www.w3.org, 2001).

*Web Service* digunakan dalam sistem ini karena bersifat interoperability atau dengan kata lain *Web Service* bisa diakses oleh aplikasi yang berjalan pada platform yang berbeda-beda. Dan berberapa vendor luar negeri mulai berkolaborasi satu sama lain dengan konsep *Web Service*, diantaranya IBM, Microsoft, SUN, ORACLE. Dalam dunia hardware *Web Service* dapat digunakan diberbagai macam alat antara lain Raspberry Pi dan Arduino.

Pada sistem ini data sensor LM35 dikirimkan kepada server, kemudian server mengolah paket data dari sensor suhu LM35 dan mengirimkan data tersebut jika ada permintaan dari client.

## 1.2 Rumusan masalah

1. Bagaimana implementasi struktur data pada embedded sistem menggunakan format JSON?
2. Bagaimana mekanisme komunikasi antar embedded sistem dengan menggunakan format JSON?
3. Bagaimana pengujian mekanisme format data komunikasi JSON dengan perangkat AC dan DC?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah untuk merancang suatu sistem yang dapat mengirimkan data yang relevant dengan permintaan client dan memberikan kemudahan kepada pengguna untuk megakses data pada server dengan menggunakan *Web Service*.

## 1.4 Manfaat

Adapun manfaat dari sistem komunikasi *Web Service* adalah sebagai berikut:

1. Memberikan kemudahan client dalam mendapatkan nilai sensor yang relevant dengan alat yang dimonitor.
2. Mengetahui kinerja dari komunikasi perangkat yang berbeda.
3. Dapat diimplementasikan pada rumah-Rumah Cerdas dengan menggunakan teknologi modern.

## 1.5 Batasan masalah

Agar skripsi ini mendapatkan fokus pada bagian yang akan dikerjakan, maka dibuat batasan masalah sebagai berikut.

1. Sistem diterapkan pada lingkup jaringan lokal untuk menjaga agar noise kecil sehingga data yang dikirim lebih akurat dan untuk mempermudah analisis.
2. LM35 berfungsi mewakili sensor lainnya dalam penerapan pengiriman data menggunakan metode web service.
3. Output sistem di representasikan menggunakan perangkat AC dan DC.

## 1.6 Sistematika pembahasan

Penulisan tugas akhir ini terdiri dari beberapa bab, yang dijelaskan sebagai berikut:

### **BAB I : PENDAHULUAN**

Bab ini berisi latar belakang masalah, tujuan, dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, dan sistematika penyusunan tugas akhir.

**BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Bab ini membahas tentang dasar teori yang terkait dengan topik penulisan skripsi yang diangkat dan menjadi acuan dasar dalam pembuatan sistem pengklasifikasian teks ini.

**BAB III : METODOLOGI PENELITIAN**

Pada bab ini dijelaskan metode-metode yang digunakan dalam menyelesaikan masalah, dan perancangan sistem yang dibuat.

**BAB IV : PERANCANGAN DAN IMPLEMENTASI**

Bab ini membahas perancangan dari sistem yang dibuat dan implementasi dari desain sistem disertai dengan potongan *source code* yang penting dalam sistem tersebut dan membahas uji coba dari sistem yang dibuat dengan melihat *output* yang dihasilkan oleh sistem.

**BAB V : ANALISIS DAN PEMBAHASAN**

Bab ini berisi hasil uji coba sistem apakah dapat diterima dengan baik oleh user dan analisis hasil pengujian apakah sistem yang dibuat berhasil.

**BAB VI : KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari hasil ujicoba yang dilakukan serta saran untuk pengembangan sistem selanjutnya.



## BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan tentang tinjauan pustaka dan dasar teori. Tinjauan pustaka menjelaskan tentang penelitian-penelitian yang telah dilakukan sebelumnya. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang dilakukan.

### 2.1 Kajian pustaka

Pada jurnal *Consuming Web Services on Mobile Platform* (Corbazon, 2010) dijelaskan *Web Service* adalah teknologi yang dapat berjalan di berbagai macam platform. Dan dalam jurnal tersebut juga menjelaskan bagaimana data dikirimkan ke Perangkat Mobile menggunakan SOAP protokol melalui Gateway. Client yang menerima data dari *Web Service* mengalami pengolahan XML yang berat sehingga untuk mengatasinya digunakan gateway sebagai server dari perangkat mobile yang menjadi server dari client tersebut.

Pada jurnal *A Sensor-based, Web Service Enable, Emergency Medical Response System* (Hashmi, 2010) dijelaskan penggunaan *Web Service* pada alat medis. *Web Service* disini digunakan untuk mengirimkan data dari sensor Alat Medis. Pada sistem yang dibuat penulis sensor harus memberikan informasi yang realtime, dan *Web Service* dapat memenuhi kebutuhan tersebut karena *Web Service* dapat mengirimkan data secara real time dan efisien.

Dari kedua penelitian tersebut diterapkan sistem pengiriman data sensor dari note menuju ke server. Agar data dapat dibaca diberbagai macam platform maka digunakan metode *Web Service*. Sistem ini dapat mengakses data dari sensor yang relevant dengan perangkat listrik yang dikontrol.

### 2.2 Dasar Teori

Dasar teori pada penelitian Implementasi struktur data *Web Service* sebagai metode komunikasi perangkat/sensor pada Sistem Rumah Cerdas terdiri dari *Web Service* Flask, JSON, Mini Raspberry Pi, Sensor LM35. Dijabarkan pada sub bab di bawah ini.

#### 2.2.1 *Web Service*

*Web Service* merupakan *software* yang dapat digunakan untuk mengolah berbagai macam format data seperti XML dan JSON. *Web Service* memiliki *interface* dan dapat digunakan untuk komunikasi antar berbagai macam *software* karena bersifat interoperability. Beberapa karakteristik dari *Web Service* adalah:

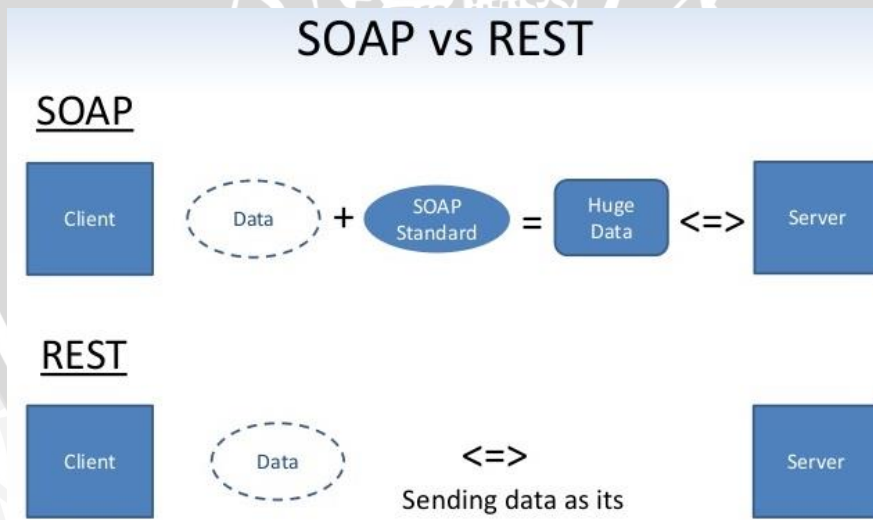
1. Message-based
2. Standards-based
3. Programming language independent
4. Platform-neutral



Terdapat berbagai macam kata kunci pada *Web Service* diantaranya SOAP dan REST. SOAP (*Simple Object Acces* ) adalah metode komunikasi antara client dan server dengan menggunakan format xml, SOAP tidak mendukung format lain selain XML. SOAP mendukung *WS-Security* untuk menambah keamanan dalam pemakaiannya. *WS-Security* mendukung integritas dan privasi data. Pengiriman data menggunakan SOAP envelope yang menjadikannya lebih berat dari REST (*Representational State Transfer*). SOAP dirancang untuk menangani lingkungan komputasi terdistribusi. (www.w3.org, 2001).

REST (*Representational State Transfer*) adalah metode komunikasi yang diterapkan dalam pengembangan layanan berbasis web. REST banyak dijalankan dalam bentuk HTTP (*Hypertext Transfer Protocol*). REST memiliki ciri yang khusus pada interaksi antara client dengan server yang difasilitasi oleh sejumlah tipe operasional dan *Universal Resource Identifiers* (URIs). Arsitektur REST yang memiliki beban komunikasi ringan antara client dan server membuat arsitektur ini menjadi populer. Layanan berbasis web yang menggunakan arsitektur REST dinamakan RESTFUL APIs (*Application Programming Interfaces*) atau REST APIs. (ekajogja, 2010)

Perbedaan antara SOAP dan REST terletak pada format pengiriman struktur data. SOAP memerlukan SOAP standart dalam mengirimkan data sedangkan REST hanya menggunakan data yang akan dikirim. REST mendukung format JSON ataupun XML, sedangkan SOAP hanya mendukung format XML. Besarnya data yang dikirim dalam *Web Service* SOAP lebih besar daripada REST. Gambar 2.1 menunjukkan perbedaan SOAP dan REST dalam pengiriman data.

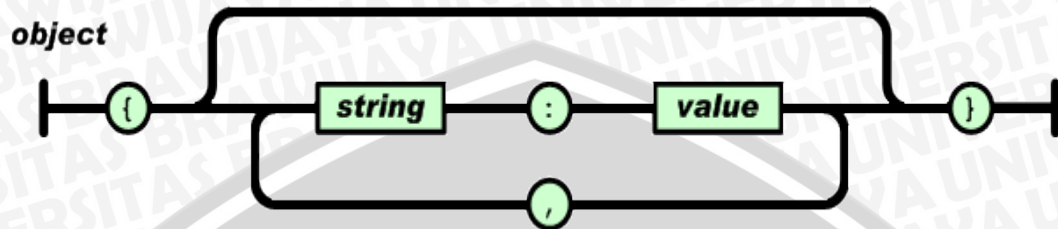


Gambar 2. 1 Perbedaan SOAP dan REST

JSON (*JavaScript Object Notation*) merupakan format pertukaran data yang mudah dibaca dan diprogram, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format JSON dikembangkan berdasarkan bagian dari bahasa pemrograman. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum. (Pengenalan JSON, 2013)



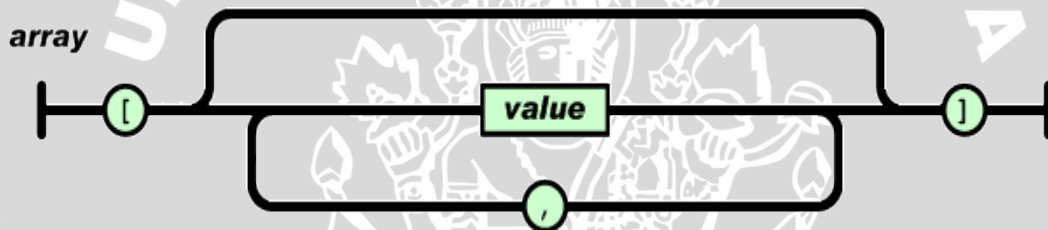
Bentuk utama dari JSON antara lain Object dan Array. Gambar object dan array dilustrasikan pada gambar 2.2 dan gambar 2.3. Objek adalah pasangan nama atau nilai yang tidak terurutkan. Objek dimulai dengan '{' (kurung kurawal buka) dan diakhiri dengan '}' (kurung kurawal tutup), setiap nama diikuti dengan ':' (titik dua). Objek pada JSON ditunjukkan pada gambar 2.2.



**Gambar 2. 2 Bentuk objek pada JSON**

(Pengenalan JSON, 2013)

Array adalah kumpulan nilai yang terurutkan. Array dimulai dengan '[' (kurung kotak buka) dan diakhiri dengan ']' (kurung kotak tutup). Setiap nilai dipisahkan oleh ',' (koma). Array pada JSON ditunjukkan pada gambar 2.3.



**Gambar 2. 3 Bentuk array pada JSON**

(Pengenalan JSON, 2013).

### 2.2.2 Mini PC Raspberry Pi

Raspberry Pi adalah salah satu Mini PC yang dikembangkan pertama kali oleh *Raspberry Pi Foundation* di UK. Raspberry Pi memiliki *System On Chip Broadcom BCM2835* dengan prosesor ARM1176JZF-S, VideoCore IV GPU, dengan RAM 256MB hingga 512MB. Selain itu untuk *booting* dan *storage* menggunakan SD Card. Raspberry Pi memiliki spesifikasi sebagai berikut:

- SoC Broadcom BCM2835 (CPU, GPU, DSP, and SDRAM)
- CPU: 700 MHz ARM1176JZF-S core (ARM11 family)
- GPU: Broadcom VideoCore IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC high-profile decoder
- Memory (SDRAM): 512 Megabytes (MiB)
- Video outputs: Composite RCA, HDMI (High-Definition Multimedia Interface)
- Audio outputs: 3.5 mm jack, HDMI
- Onboard storage: SD, MMC, SDIO card slot

- 10/100 Ethernet RJ45 onboard network
- Storage via SD/ MMC/ SDIO card slot.

Raspberry pi digunakan karena memiliki spesifikasi yang sesuai dengan kebutuhan sistem yang dibahas dalam penelitian ini. *Mini PC* Raspberry Pi ditunjukkan pada Gambar 2.4.



**Gambar 2. 4** Mini pc raspberry pi

(Raspberry.org, 2015)

### 2.2.3 Sensor LM35

Sensor Suhu LM35 adalah salah satu jenis sensor yang merubah besaran suhu ke besaran listrik dalam bentuk tegangan. LM35 memiliki 3 buah pin kaki, pin1 untuk INPUT tegangan positif (+), pin2 OUTPUT, pin3 INPUT tegangan negatif/GND (-). Dapat beroperasi pada tegangan 4 volt sampai 30 volt. Setiap perubahan suhu 1°C mengakibatkan perubahan tegangan sebesar 10 mV.

Sensor suhu LM35 digunakan untuk mendapatkan data suhu ruangan, sensor mengumpulkan data sebelum akhirnya dikirim melalui metode HTTP POST ke Server. LM35 digunakan untuk menghidupkan dan menyalakan Kipas dalam penelitian ini. Simulasi pengontrolan kipas dilakukan terhadapnya. Sensor LM35 ditunjukkan pada gambar 2.5.



**Gambar 2. 5** Sensor LM35

### 2.2.4 Arduino UNO

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada mikrokontroler ATmega328. Arduino UNO memiliki 14 pin digital *input* dan *output*, 6 dari 14 pin tersebut dapat digunakan sebagai *output* PWM. 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah *port* USB, sebuah *power jack* DC 12 volt, sebuah ICSP *header*, dan sebuah tombol *reset*. (Arduino, 2015). Bahasa pemrograman yang digunakan oleh Arduino berbasis pada bahasa pemrograman C/C++. Arduino Uno ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Arduino UNO

(Arduino, 2015)

### 2.2.5 HTTP POST

HTTP (Hypertext Transfer Protocol) digunakan untuk komunikasi antara client dan server. HTTP POST merupakan metode yang bisa digunakan apabila terdapat banyak data yang akan dikirimkan. Data akan dimasukkan dalam bentuk format data yang baru dan kemudian dikirimkan ke sumber yang akan dituju. Ada beberapa bentuk format data didalam HTTP POST antara lain: message, plain dan HTML. (CloudMailin, 2015)

### 2.2.6 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python di klaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python mendukung multi paradigma pemrograman, utamanya, namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang

tersedia pada python adalah sebagai bahasa pemrograman yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. (Python Software, 2001)

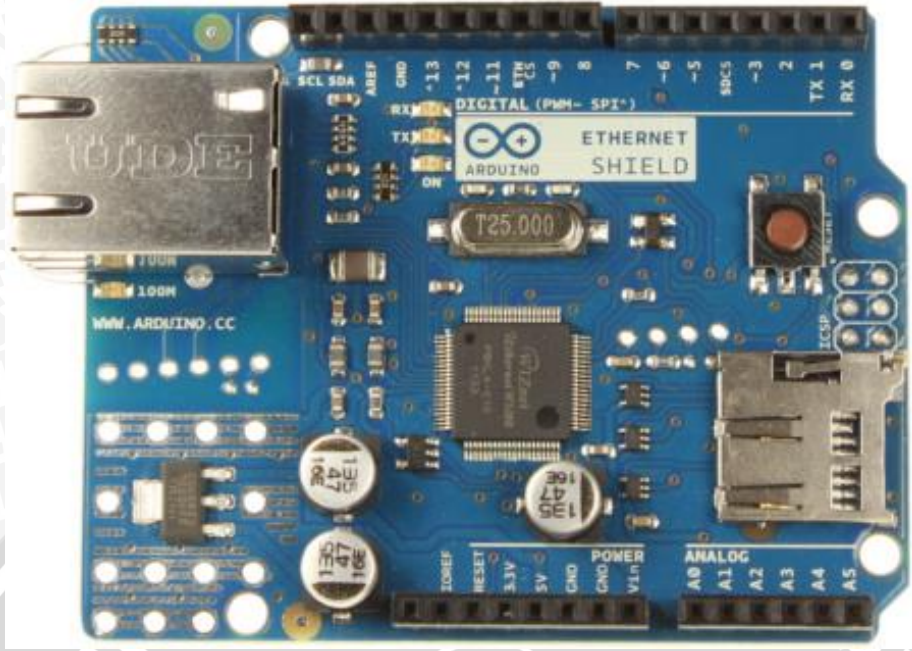
Terdapat beberapa lingkungan yang dapat dihuni Python. Berikut adalah beberapa lingkungan sistem operasi yang bisa dihuni Python:

- Win 9x/NT/2000
- Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, dan lain - lain)
- Macintosh (Intel, PPC, 68K)
- OS/2
- DOS
- PalmOS
- Nokia mobile phones
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS
- QNX
- VxWorks
- Psion
- Python juga dirancang ulang di Java dan .NET Virtual Machine

### 2.2.7 Ethernet Shield

Ethernet Shield adalah salah satu jenis modul yang berfungsi menyambungkan Arduino board dengan Internet melalui kabel RJ45.

Ethernet Shield digunakan untuk menghubungkan arduino dengan perangkat lain menggunakan kabel LAN. Dalam penelitian ini arduino digunakan mengirimkan data ke raspberry pi melalui LAN. Ethernet shield ditunjukkan pada Gambar 2.7.

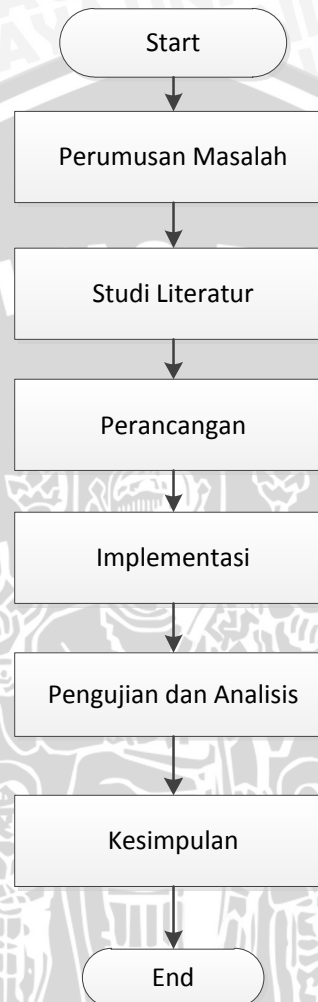


**Gambar 2. 7** Arduino Ethernet LAN  
(electrical4u, 2012)



## BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan langkah-langkah yang dilakukan dalam pelaksanaan penelitian. Metode penelitian yang digunakan pada skripsi ini ditunjukkan pada diagram alir pada Gambar 3.1.



Gambar 3. 1 Alur Pelaksanaan

### 3.1 Perumusan Masalah

Perumusan masalah merupakan tahap menentukan masalah yang dibahas dalam penelitian. Masalah yang terbentuk meliputi:

1. Bagaimana format pengiriman data dari Arduino menuju ke server
2. Bagaimana penerapan *Web Service* dalam sistem
3. Bagaimana fungsional keseluruhan sistem yang dibuat

### 3.2 Penentuan Tujuan Penelitian

Dalam penelitian ini ditetapkan beberapa tujuan untuk memfokuskan permasalahan dengan hasil akhir berupa laporan. Adapun tujuan dari penelitian ini adalah yang sudah dijelaskan pada bab 1. Hasil dari penelitian ini diharapkan dapat menjadi dasar pembentukan project Rumah Cerdas.

### 3.3 Studi Literatur

Pada tahap ini dilakukan kajian dan pembahasan mengenai dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, artikel, jurnal, *e-book*, dan dokumentasi *project*. Teori-teori pendukung tersebut meliputi :

1. *Web Service* FLASK
2. JSON
3. HTTP POST
4. Mini PC Raspberry Pi
5. Sensor LM35

### 3.4 Penetapan Kebutuhan

Penetapan kebutuhan dalam pelaksanaan penelitian ini meliputi 2 (dua) hal yaitu kebutuhan fungsional dan kebutuhan non fungsional.

#### 3.4.1 Kebutuhan Fungsional

Kebutuhan fungsional yang ingin dicapai dalam pembuatan sistem ini adalah sebagai berikut:

1. Sistem dapat membaca data yang diperoleh oleh sensor.
2. Sistem dapat mengirimkan data kepada server.
3. Sistem dapat menjalankan perangkat AC dan perangkat DC.
4. Sistem dapat memberikan informasi kepada user tentang data sensor .

#### 3.4.2 Kebutuhan Non Fungsional

Kebutuhan non-fungsional yang ingin dicapai dalam pembuatan sistem ini adalah sebagai berikut:

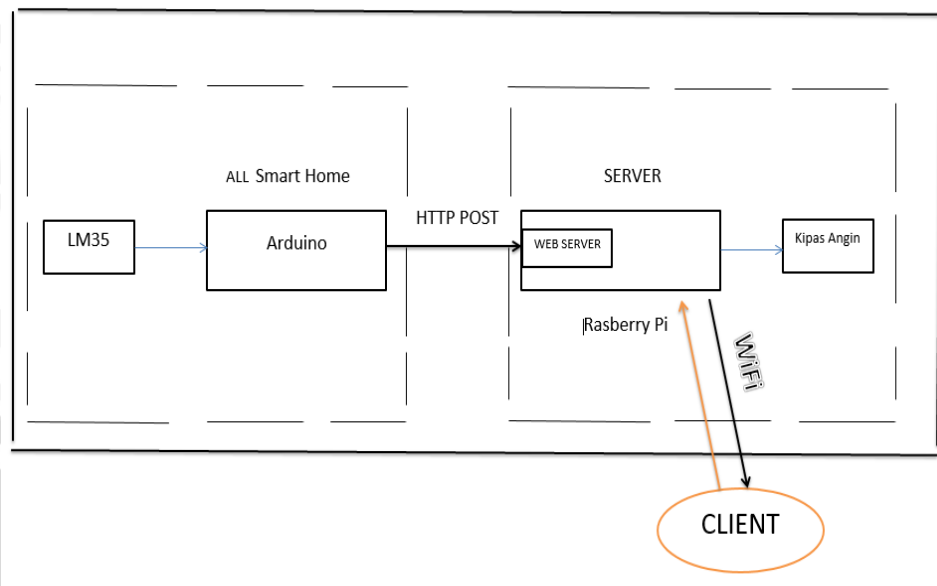
1. Jika listrik padam, sistem dapat bekerja saat listrik sudah menyala.
2. Sistem memberikan data yang relevant dengan perangkat yang dikontrol.

### 3.5 Perancangan Sistem

Skema sistem yang dirancang ditunjukkan pada gambar 3.2 terdapat 3 (tiga) bagian dalam pembentuk sistem yaitu, bagian arduino, raspberry pi dan client. Arduino mengolah data dari sensor LM35 kemudian dikirimkan ke raspberry pi



melalui metode HTTP POST. Raspberry mengolah paket data dari arduino selanjutnya dikirimkan ke client jika ada permintaan data.



**Gambar 3. 2 Diagram Blok**

### 3.6 Implementasi Sistem

Implementasi sistem mengacu pada perancangan sistem dan diagram blok seperti yg terdapat pada Gambar 3.2. implementasi sistem dibagi menjadi 3 (tiga) bagian. Bagian arduino, bagian raspberry pi, dan bagian client. Secara garis besar terdapat beberapa langkah-langkah implementasi yaitu:

1. Arduino mengambil besaran suhu dari sensor LM35 dan dijadikan dalam bentuk digital.
2. Arduino mengirim paket data sensor ke raspberry pi menggunakan metode HTTP POST context:type : text/plain.
3. Raspberry pi mengolah kiriman paket data sensor dari arduino dan diubah dalam bentuk JSON.
4. Client meminta data ke raspberry pi menggunakan metode HTTP GET.
5. Raspberry pi mengirimkan paket data ke client dalam format JSON.
6. Client melakukan parsing data terhadap paket data yang diterima dari Raspberry pi.

### 3.7 Pengujian dan Analisis

Pengujian sistem pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi dari keutuhan yang melandasinya. Pengujian yang dilakukan meliputi:

1. Pembacaan data sensor LM35 oleh arduino.
2. Pengiriman data sensor LM35 dari arduino ke raspberry pi menggunakan metode HTTP POST

3. Data sensor LM35 dari arduino dirubah menjadi format JSON oleh raspberry pi.
4. Client meminta data sensor LM35 kepada raspberry pi menggunakan metode HTTP GET.

### 3.8 Kesimpulan

Kesimpulan didapatkan setelah melakukan perancangan, implementasi, pengujian dan analisis sistem. Kesimpulan diambil berdasarkan hasil pengujian dan analisis sistem yang telah dibuat. Isi dari kesimpulan diharapkan dapat menjadi acuan pada penelitian lain untuk mengembangkan penelitian yang berhubungan dengan penelitian ini.



## BAB 4 PERANCANGAN DAN IMPLEMENTASI

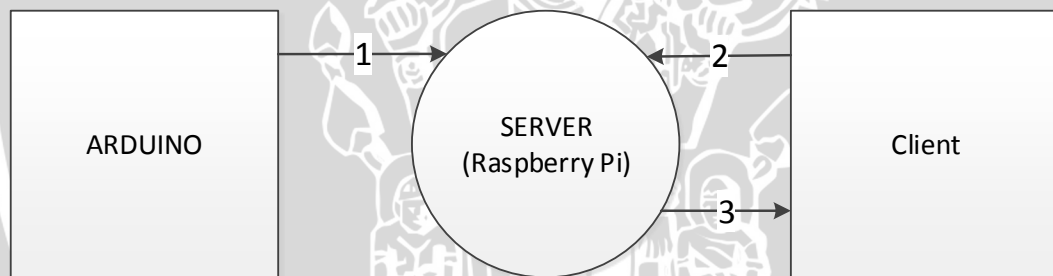
Bab ini menjelaskan perancangan sistem yang dibuat dan kemudian implementasinya. Bab ini akan memudahkan proses pengujian dan analisis karena didasarkan pada perancangan dan implementasi.

### 4.1 Perancangan Sistem

Tahap perancangan berisi tentang analisa perangkat keras dan perangkat lunak yang digunakan untuk membuat sistem, ada beberapa tahap di antaranya sebagai berikut: diagram alir keseluruhan sistem, analisa perancangan sistem, perancangan perangkat arduino, perancangan perangkat raspberry pi.

#### 4.1.1 Diagram alir keseluruhan sistem

Diagram alir keseluruhan menggambarkan sistem alat yang dibuat dibagi menjadi 3 (tiga) bagian blok atau kelompok sistem berupa Input, Proses, dan Output. Seperti pada gambar 4.1 input yang digunakan berupa sensor LM35, pemrosesan data dan pengiriman data ke client dilakukan oleh raspberry pi, client melakukan permintaan data kepada raspberry pi.



Gambar 4. 1 Diagram blok sistem

Pada gambar 4.1 terdapat diagram blok sistem yang menjelaskan sistem dimulai dari pembacaan data sensor suhu LM35 oleh arduino kemudian paket data dari sensor suhu LM35 dikirimkan ke raspberry pi menggunakan metode *Web Service*, setelah paket data diterima oleh raspberry pi, data sensor suhu LM35 diolah dan dikirim dengan menggunakan format data JSON. Pengiriman data sensor suhu LM35 dilakukan secara terus menerus kedalam raspberry pi, dimana data dari sensor akan di rata-rata setiap 10 (detik) terakhir. Setelah client meminta data sensor LM35 maka program client akan menampilkan data rata-rata sensor setiap 10 (detik) terakhir.

#### 4.1.2 Analisis perancangan sistem

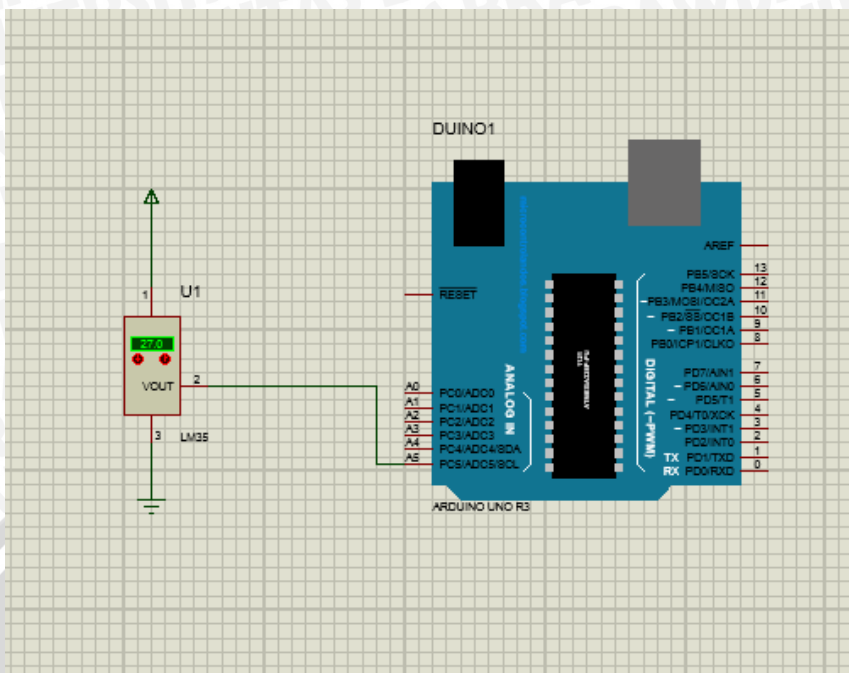
Perangkat keras dalam sistem ini memiliki sistematika atau infrastruktur penunjang yang saling berkaitan satu dengan yang lain. Perancangan sistem digunakan sebagai media komunikasi antara perangkat satu dengan yang lain. Perancangan dimulai dengan pembacaan sensor, pengiriman data oleh arduino hingga menampilkan sebuah data kepada client.

Perancangan *hardware* dilakukan dengan menganalisa semua kebutuhan sistem. Pada sistem digunakan arduino sebagai media pengirim data dari sensor ke raspberry pi. Data dari sensor LM35 sendiri berupa data analog, sehingga harus diubah menjadi data digital agar dapat dibaca oleh server. Modul pengiriman data sensor LM35 menggunakan protokol HTTP POST melalui Ethernet LAN. Server (Raspberry Pi) dikontrol menggunakan WiFi adapter dikarenakan port LAN pada Raspberry Pi digunakan untuk menghubungkan antara Arduino UNO dengan Server (Raspberry Pi). Penerapan dilakukan berdasarkan kebutuhan masing-masing dari sebuah *hardware*.

Program pada server raspberry pi berfungsi mengirim dengan format data JSON pada client, setelah client meminta data pada raspberry pi, maka program menampilkan data yang telah diminta sebelumnya. Data sensor berubah terus menerus dikarenakan arduino mengirimkan data secara terus-menerus. Data sensor diambil rata-rata dari 10 data terakhir.

#### 4.2 Perancangan perangkat pembaca dan pengirim data sensor

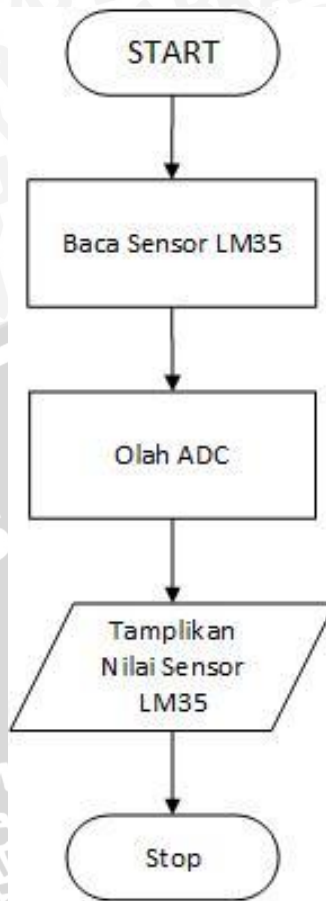
Perangkat arduino terhubung dengan sensor LM35 sehingga sensor LM35 dapat mengakuisisi nilai suhu pada ruangan. Setelah nilai suhu didapatkan, Arduino mengirim data suhu ke server melalui HTTP POST. Sensor LM35 dihubungkan dengan pin A5 arduino, VCC, dan ground, perancangan perangkat arduino dan sensor LM35 ditunjukkan pada gambar 4.2.



Gambar 4. 2 Simulasi perangkat arduino

#### 4.2.1 Diagram alir pembacaan data sensor LM35

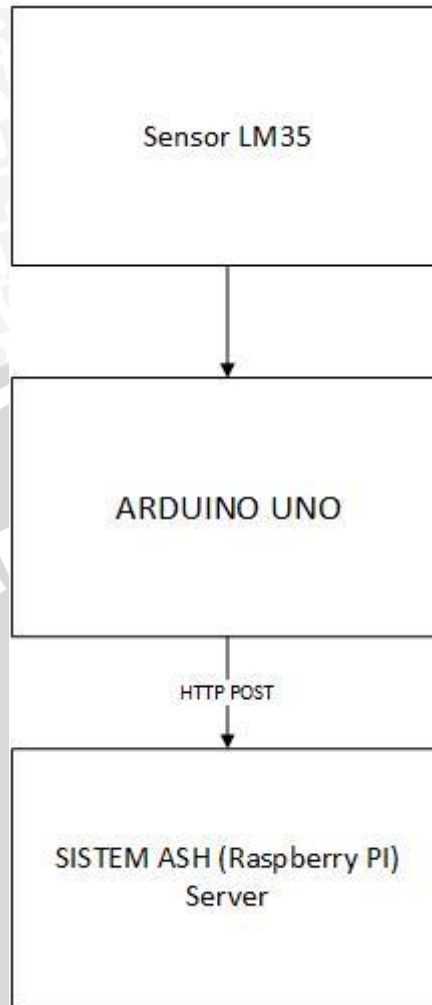
Diagram alir sensor menjelaskan tentang alur dari data sensor suhu yang digunakan pada sistem yang akan dibuat. Diagram alir ini menggambarkan pembacaan data oleh sensor LM35 (Analog) diubah menjadi data digital oleh Arduino. Data sensor LM35 (Digital) dikirimkan menuju server (Raspberry Pi). Arduino mengirimkan data dari sensor suhu LM35 setiap detik. Diagram alir pembacaan data sensor LM35 ditunjukkan pada gambar 4.3.



Gambar 4. 3 Diagram alir pembacaan data sensor

#### 4.2.2 Diagram alir pengiriman data dari arduino ke Server

Diagram alir sensor ini menggambarkan diagram alur dari sensor LM35 yang digunakan pada sistem yang akan dibuat. Pada diagram alir sensor LM35 sistem kerjanya dimulai dari start kemudian data dari sensor akan dimasukkan kedalam pin adc yang terdapat pada Arduino untuk didapatkan nilai digitalnya karena sensor LM35 adalah sensor analog. Setelah nilai suhu sudah didapatkan maka akan dikirimkan dengan metode HTTP POST. Gambar 4.4 diagram alir pengiriman data dari arduino ke server.



Gambar 4. 4 Diagram alir pengiriman paket data ke server

### 4.3 Perancangan perangkat penerimaan dan pengiriman data

Raspberry pi merupakan server dari keseluruhan sistem, dalam raspberry pi terdapat program untuk menerima data suhu dari arduino dan merubah paket data tersebut dalam format JSON. Penerimaan data sensor oleh server, dan data sensor diubah dalam bentuk JSON kedua pekerjaan itu dilakukan oleh *Web Service*.

#### 4.3.1 Diagram alir penerimaan data sensor dari arduino

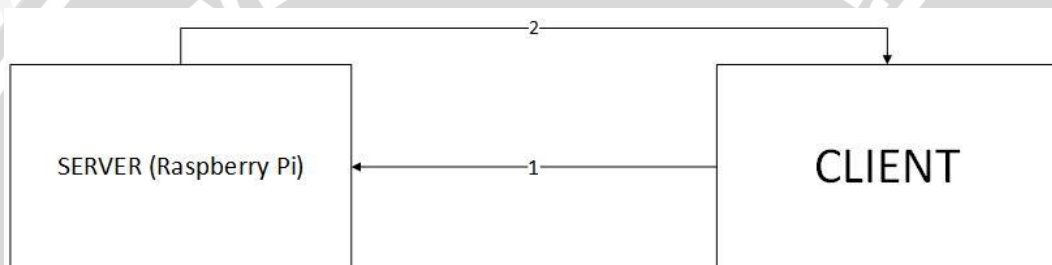
Data sensor suhu LM35 dikirimkan ke server melalui kabel ethernet LAN, data sensor suhu LM35 dikirimkan menggunakan metode HTTP POST dengan context:type : text/plain. Server atau raspberry pi hanya menerima data dengan context:type : text/plain, jika ada paket data dikirimkan dari arduino selain dengan ketentuan dari server maka data ditolak. Raspberry pi hanya akan menampung 10 (sepuluh) data dari sensor suhu LM35. Proses penerimaan paket data ditunjukkan pada gambar 4.5.



Gambar 4. 5 Diagram alir penerimaan data sensor

### 4.3.2 Diagram alir pengiriman data sensor ke client

Paket data akan dikirimkan setelah mendapat permintaan dari client melalui metode HTTP GET kemudian pengiriman paket data dari server ke client menggunakan format JSON, client bertugas melakukan parsing data dari paket data yang diterima dari raspberry pi atau server.



Gambar 4. 6 Diagram alir raspberry pi

Dijelaskan pada gambar 4.6 bahwa client meminta data sensor LM35 pada Server melalui protokol HTTP GET (1), dan server memberikan data dalam format JSON kepada client (2).

### 4.4 Perancangan struktur data sistem

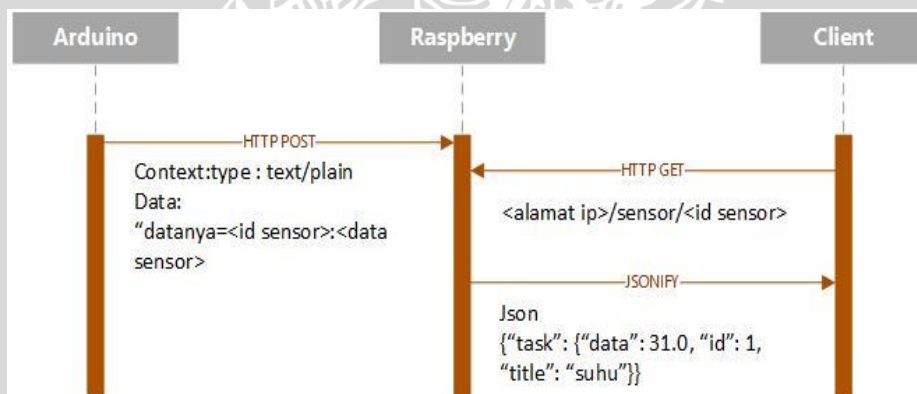
Arduino membaca pin 5 (lima) yang merupakan output dari sensor LM35, kemudian data analog dikonversi oleh pin adc sehingga didapatkan angka digital. Kemudian angka digital akan dikalikan dengan 9.31 untuk mendapatkan nilai dalam derajat celsius (°C). Nilai 9.31 didapatkan dari pembagian 10/1.07242. karena 10mV tegangan yang didapatkan sama dengan 1 derajat celsius. 1.0742 merupakan pembagian hasil maksimum dari adc(analog digital converter) 1.1V dan 1024 merupakan prosentasi dari jarak.

Format data post yang digunakan adalah context:type :text/plain dan dapat ditunjukkan dengan "datanya=<nomer sensor>, <data sensor>". Sensor LM35 merupakan sensor ke 1 hasil ini ditempatkan pada variabel <nomer sensor>. Data dari sensor LM35 ditempatkan di variabel <data sensor>, misalnya hasil data sensor LM35 31 derajat celsius maka format pengirimannya "datanya=<1>,<31>".



Dalam raspberry pi terdapat program *Web Service*. *Web Service* memiliki 3(tiga) service yaitu: ID ke 0 (nol), ID ke 1 (satu), dan ID ke 2 (dua), hanya digunakan ID ke 1 (satu) untuk sensor LM35. Raspberry pi menerima data dalam bentuk context:type : text/plain. Raspberry pi akan menerima format data text/plain, jika data sudah diterima maka server akan menampilkan pada jendela putty dalam format "datanya=1:31". Terjadi pemisahan data, yaitu pemisahan "datanya" dan "1:31", raspberry pi menerima data "1:31" untuk dimasukan ke database. hanya ada (10) sepuluh data dari sensor LM35, jika terdapat data ke 11 (sebelas) maka data pertama akan dihapus diganti dengan data ke 11 (sebelas), begitu pula seterusnya. 10 data tersebut dirata-rata untuk mengontrol perangkat ac ataupun dc. Kemudian data rata-rata sensor diubah dalam format JSON, yaitu "data":31.0, "id":1, "tittle":"suhu".

Client meminta data ke server dengan menggunakan metode HTTP GET dengan alamat `http://192.168.0.66:808/sensor/1`, 192.168.0.808 merupakan alamat ip pada raspberry pi, sensor merupakan variabel dari sensor itu sendiri dan 1 (satu) merupakan ID untuk sensor LM35. Setelah client meminta pada alamat ip yang benar maka raspberry pi merespon dan mengirimkan data dalam format JSON. Client melakukan parsing data sehingga dapat dilakukan pembacaan dengan mudah. Format JSON "data":31.0, "id":1, "tittle":"suhu" dilakukan parsing terhadapnya sehingga client akan membaca "sensor:suhu, data: 31.0". Diagram alir perancangan struktur data sistem ditunjukkan pada gambar 4.7.



Gambar 4. 7 Diagram alir mekanisme pengiriman data

## 4.5 Implementasi Sistem

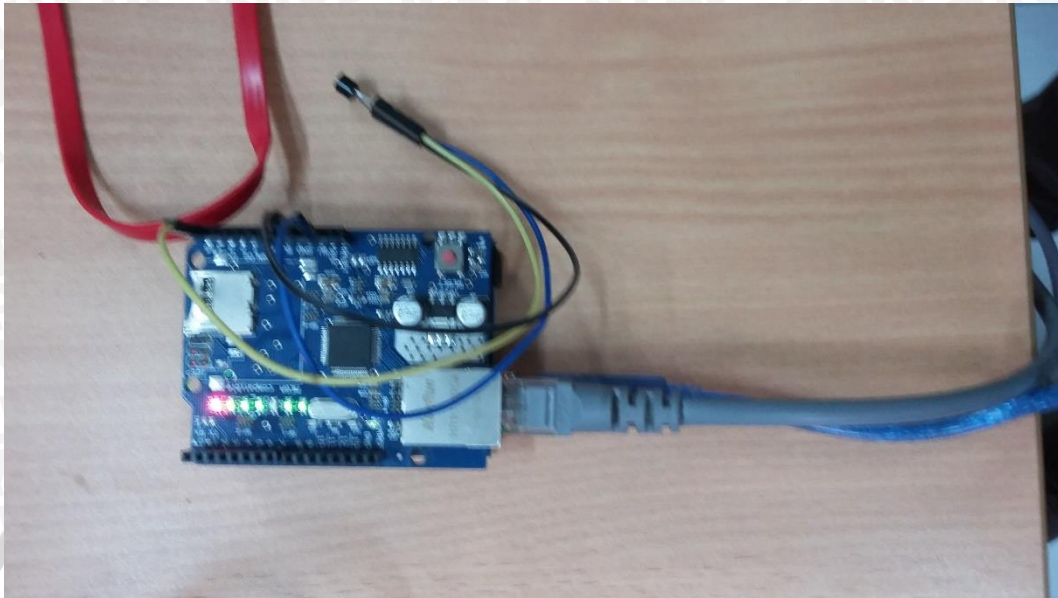
Implementasi sistem mencakup implementasi bagian pembaca dan pengirim data sensor, implementasi bagian server, dan implementasi data client.

### 4.5.1 Bagian pembaca dan pengirim data sensor

Pada perangkat arduino dilakukan pengiriman data sensor LM35 menggunakan Protokol HTTP POST melalui Ethernet LAN. Pengiriman data melalui ethernet LAN menuju kepada Server (Raspberry Pi).

Pembacaan suhu dalam ruangan menggunakan sensor suhu LM35. Data analog dari sensor LM35 diubah menjadi digital oleh pin adc pada arduino. Kemudian

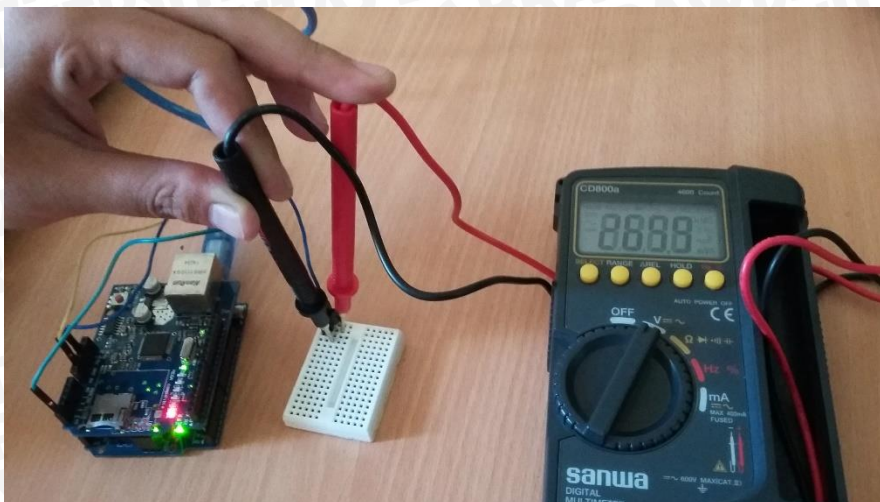
data dikirimkan kepada raspberry pi. Rangkaian Arduino dan sensor LM35 ditunjukkan pada Gambar 4.8.



**Gambar 4. 8 Rangkaian Arduino dan LM35**

Pembacaan sensor suhu dilakukan dalam 2 tahap, pertama dilakukan dengan pembacaan sensor menggunakan program pada arduino dan kedua dilakukan pembacaan sensor LM35 menggunakan alat avometer.

Penggunaan alat avometer dilakukan dengan menghubungkan sensor LM35 dengan alat avometer, pin minus (-) pada avometer diletakkan di kaki ground pada sensor LM35, dan pin plus (+) pada avometer diletakkan pada kaki out pada sensor LM35 seperti ditunjukkan pada gambar 4.9.



Gambar 4. 9 Pembacaan sensor menggunakan avometer

#### 4.5.2 Bagian Web Server

Implementasi pada server merupakan bagian terpenting dalam sistem ini, dikarenakan pada bagian ini *Web Service* bisa dijalankan. Penulis menggunakan Flask sebagai *Web Service*. Karena tujuan dari *Web Service* tersendiri agar bisa berjalan di berbagai macam platform. Server berfungsi menerima data dari arduino dan mengirimkan data kepada Client menggunakan format data JSON.

```
tasks = [  
  {  
    'id': 0,  
    'sensor': "xxx",  
    'data': "35"  
  },  
  {  
    'id': 1,  
    'title': "suhu",  
    'data': "30"  
  },  
  {  
    'id': 2,  
    'title': "xxx",  
    'data': "20"  
  }  
]
```

```

@app.route('/', methods=['GET'])
def get_tasks():
    return jsonify({'tasks': tasks})
    return jsonify({'tasks': [make_public_task(task) for task in
tasks]})

@app.route('/sensor/<int:task_id>', methods=['GET'])
def get_task(task_id):
    task = [task for task in tasks if task['id'] == task_id]

task[0]['data']=sum(sensor[int(task[0]['id'])])/float(len(sensor[i
nt(task[0]['id'])]))
    if len(task) == 0:
        abort(404)
    return jsonify({'task': task[0]})

@app.route('/messages', methods = ['POST'])
def api_message():

    if request.headers['Content-Type'] == 'text/plain':
        data = request.data
        new_data = data.split('=')
        new_data_1 = new_data[1].split(':')
        if len(sensor[int(new_data_1[0])])<10:
            sensor[int(new_data_1[0])].append(int(new_data_1[1]))
        else:
            sensor[int(new_data_1[0])].pop(0)
            sensor[int(new_data_1[0])].append(int(new_data_1[1]))

```

Source code di atas menjelaskan program yang ditulis pada server. Pada server terdapat *Web Service* yang memiliki 3 (tiga) utama. 3 (tiga) fungsi tersebut antara lain penerimaan data dari arduino, pengiriman data kepada Client dan pengubahan data kedalam bentuk JSON. selain terdapat 3 (tiga) fungsi *Web Service* di dalam server terdapat database sederhana berupa 3 (tiga) service yang dimiliki oleh server.

#### 4.5.2.1 Implementasi rangkaian raspberry pi

Pada implementasi raspberry pi dibuat rangkaian penghubung server dan arduino menggunakan ethernet LAN. Pada raspberry pi terdapat modul WiFi adapter untuk menghubungkan raspberry pi dengan laptop. Rangkaian raspberry pi ditunjukkan pada gambar 4.10.



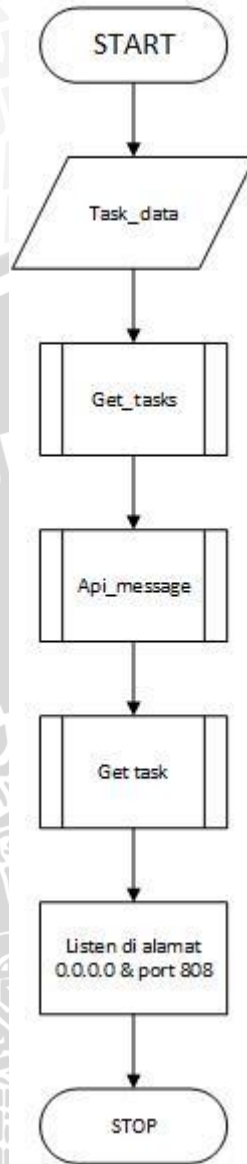
**Gambar 4. 10 Implementasi raspberry pi**

Setelah rangkaian server menggunakan raspberry pi terbentuk selanjutnya dihubungkan dengan arduino menggunakan ethernet LAN. Rangkaian raspberry pi dan arduino ditunjukkan pada gambar 4.11.



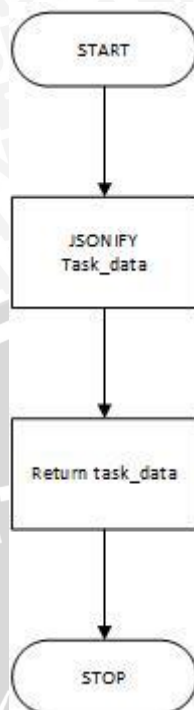
**Gambar 4. 11 Implementasi keseluruhan sistem**

#### 4.5.2.2 Implementasi program server



**Gambar 4. 12 Flowchart main program**

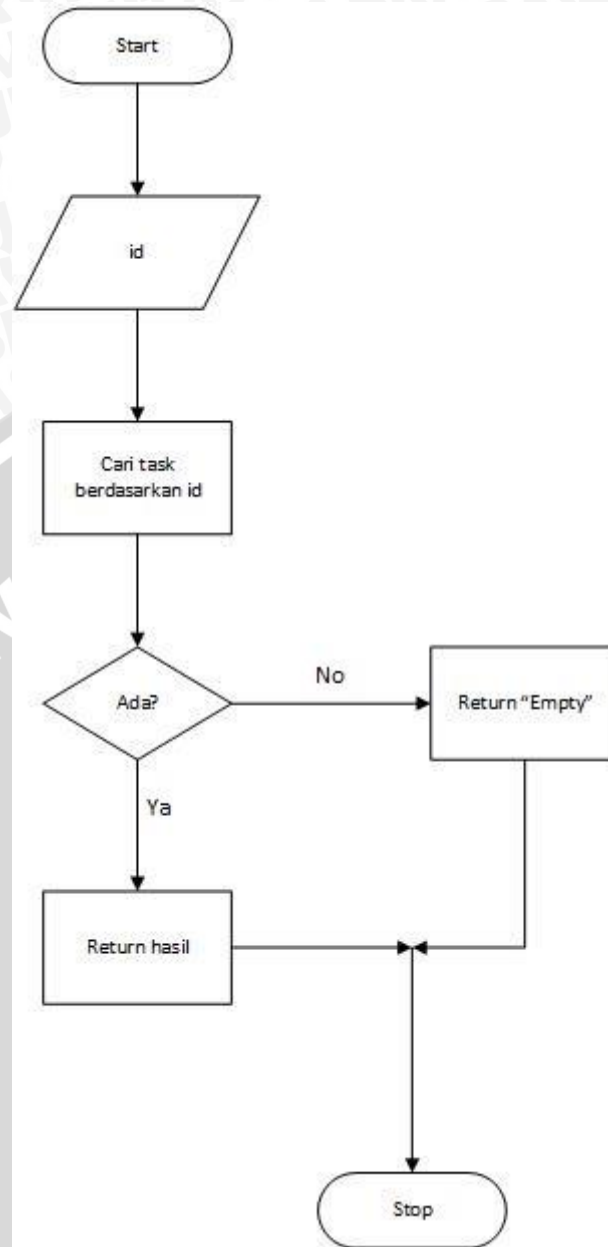
Flowchart main program dimulai dari start kemudian program melakukan inisialisasi variable `task_data`, kemudian server melakukan inisialisasi pada 3(tiga) fungsi *Web Service* flask yaitu: `get_task`, `api_message`, dan `get_task`.



**Gambar 4. 13 Flowchart get\_tasks**

Dari gambar 4.13 dapat dilihat bahwa service `get_tasks` berfungsi untuk memberikan informasi kepada client tentang service apa saja yang dimiliki Server dalam bentuk JSON. Flowchart dimulai dengan melakukan `JSONIFY` (merubah ke bentuk JSON) pada variable `task_data`, setelah selesai melakukan proses `JSONIFY` maka data dalam bentuk JSON dikembalikan ke variable `task_data`.

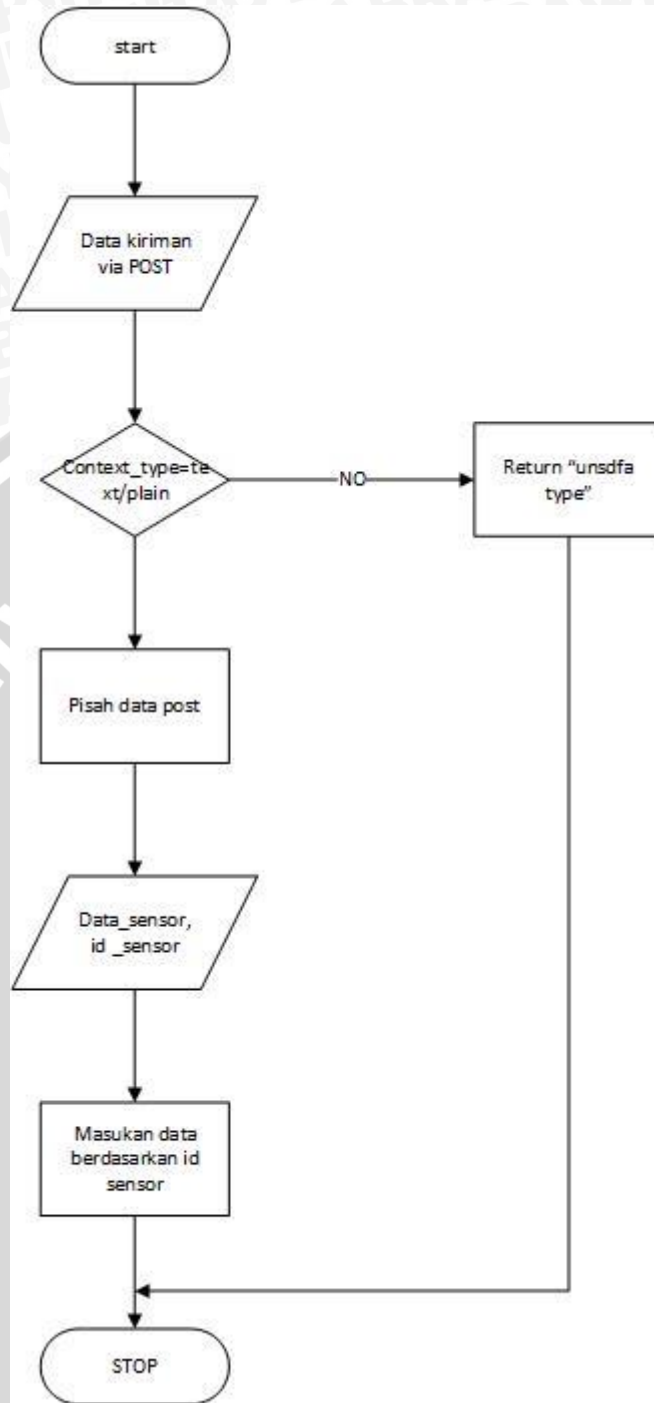
Jika client mengakses url root `"192.168.0.66:808"` maka client dapat melihat sensor yang dapat diakses. Server memiliki 3 (tiga) ID yaitu ID ke-0 (Nol), ID ke-1 (satu), dan ID ke-2 (dua). Penulis hanya menggunakan 1 (satu) sensor LM35 dengan ID ke-1(satu)



Gambar 4. 14 Flowchart get\_task

Pada gambar 4.14 dijelaskan client mengakses HTTP GET dari alamat url "192.168.0.66:808/sensor/<id\_sensor>" untuk mengakses 1 (satu) sensor, jika client memasukan nilai <id\_sensor> dengan benar maka server akan memberikan data yang diminta oleh client. Misalnya: server memiliki 3 (tiga) ID, ID ke-0(nol), ID ke-1(satu), dan ID ke-2(dua), jika client mengakses <id\_sensor> ke-0, ke-1, ke-2 maka server akan memberikan informasi dari data suhu yang diminta oleh client. Fungsi ini juga melakukan rata-rata terhadap 10 (sepuluh) data sensor terakhir.



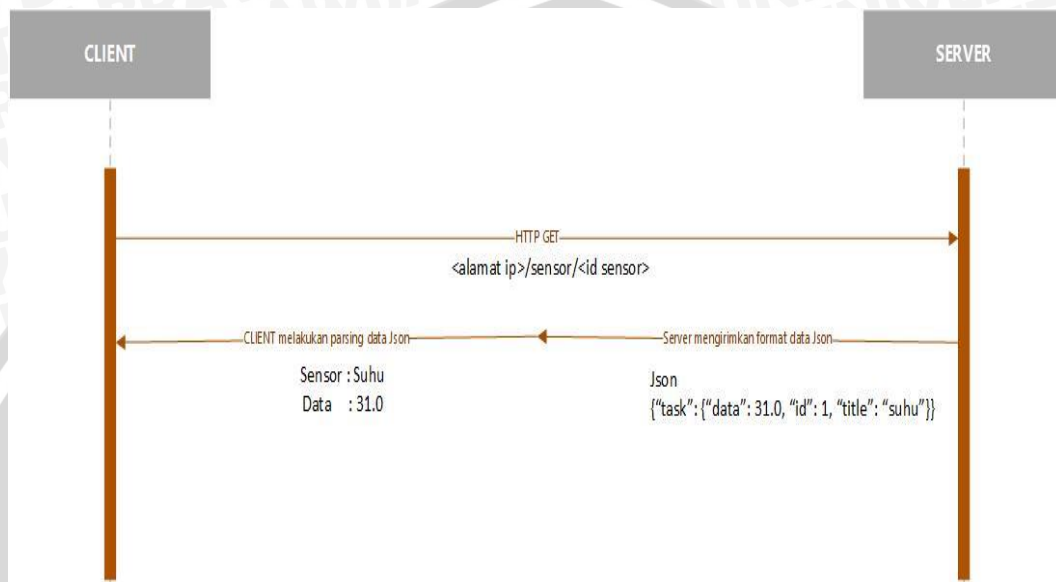


**Gambar 4. 15 Flowchart api\_message**

Dari gambar 4.12 dijelaskan api\_message merupakan fungsi dari Web Service . Pada flowchart api\_message gambar 4.15 data dikirimkan menggunakan metode HTTP POST dengan context:type = text/plain jadi jika ada data kiriman dari arduino dengan context header selain context:type = text/plain maka data akan di tolak. Format data HTTP POST berupa data text yaitu dengan format "datanya=<nosensor>:<datasensor>. Setelah data diterima oleh server data akan dipisah dalam format <data\_sensor><id\_sensor>.

### 4.5.3 Bagian data client

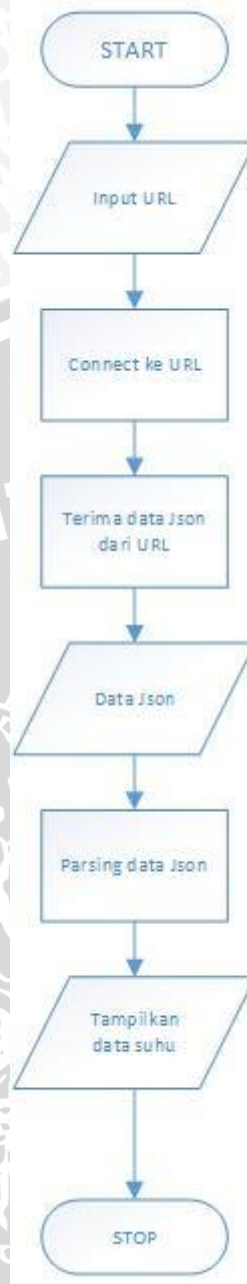
Client dalam sistem ini bertujuan untuk meminta data dari server, kode python digunakan untuk melakukan permintaan data ke raspberry pi. Client menggunakan metode HTTP GET untuk melakukan permintaan data, setelah data dikirim dari raspberry pi dengan format JSON client melakukan parsing data sehingga didapatkan hasil yang diinginkan yang ditunjukkan pada gambar 4.16.



**Gambar 4. 16 Diagram implementasi client**

Raspberry pi akan mengirimkan data jika client meminta data pada alamat ip raspberry yang benar. Client juga harus memberikan informasi pada server data yang akan diakses. Jika kedua kriteria tersebut dilakukan dengan benar maka raspberry pi akan mengirimkan data yang diminta oleh client.

Client diprogram dalam kode python, dalam sistem ini client bertugas meminta data pada server dengan menggunakan metode HTTP GET, dan client bertugas melakukan parsing data terhadap data JSON yang diterima dari server. Design client ditunjukkan pada gambar 4.17



**Gambar 4. 17** Flowchart tugas client

Client memasukan alamat url '192.168.0.66:808/sensor/1' untuk mendapatkan data dari server. Kemudian server mengirim data JSON ke client (yang ditunjukkan pada source code dibawah) dan client akan melakukan parsing data sehingga didapatkan data yang diinginkan (sensor: suhu, data: 31.0).

```
{  
  "task": {  
    "data": 31.0,  
    "id":1,  
    "tittle": "suhu"  
  }  
}
```



## BAB 5 PENGUJIAN

Pada bab ini dilakukan proses pengujian dan analisis dari sistem yang telah dibuat. Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui bahwa semua kebutuhan fungsional maupun non-fungsional yang dirancang sebelumnya telah terpenuhi dan berjalan dengan sesuai yang diharapkan. Proses pengujian terdapat empat tahap pengujian yaitu: pengujian sensor, pengujian pada arduino, pengujian pada server, dan pengujian terhadap client.

Pengujian dalam penelitian ini dilakukan untuk mengetahui apakah semua sub-sistem dapat bekerja sesuai dengan yang diinginkan. Pengujian fungsional merupakan pengujian yang dilakukan untuk mengetahui kebutuhan fungsional dari sistem dapat terpenuhi setelah pengujian semua sub-sistem berhasil.

Analisis dilakukan untuk mendapatkan kesimpulan dari hasil pengujian sistem *Web Service* yang terintegrasi. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan pada setiap tahapan pengujian.

### 5.1 Pengujian sistem

Berdasarkan perancangan maka dilakukan 4 (empat) tahap pengujian yaitu: pengujian pada sensor, pengujian pada arduino, pengujian pada raspberry pi dan pengujian terhadap client

#### 5.1.1 Pengujian Sensor

##### 5.1.1.1 Pengujian sensor LM35

Pengujian Sensor LM35 dilakukan dengan mengambil data dari data pertama hingga kelima. Dan berikut adalah hasil pembacaan data keluaran dari sensor LM35 dengan menggunakan alat avometer yang ditunjukkan pada tabel 1.

**Tabel 1 Hasil pembacaan sensor LM35**

Data ke-	Hasil Avometer (dalam mV)
1	306.8
2	668.7
3	567.5
4	300
5	290

Dalam pengujian sensor suhu dilakukan pengambilan data sensor lima (5) kali secara random yang menunjukkan keluaran dari avometer, setiap hasil 10 mV pada

avometer bernilai 1°C, dari data tersebut hasil keluaran dari sensor LM35 ditunjukkan pada tabel 2.

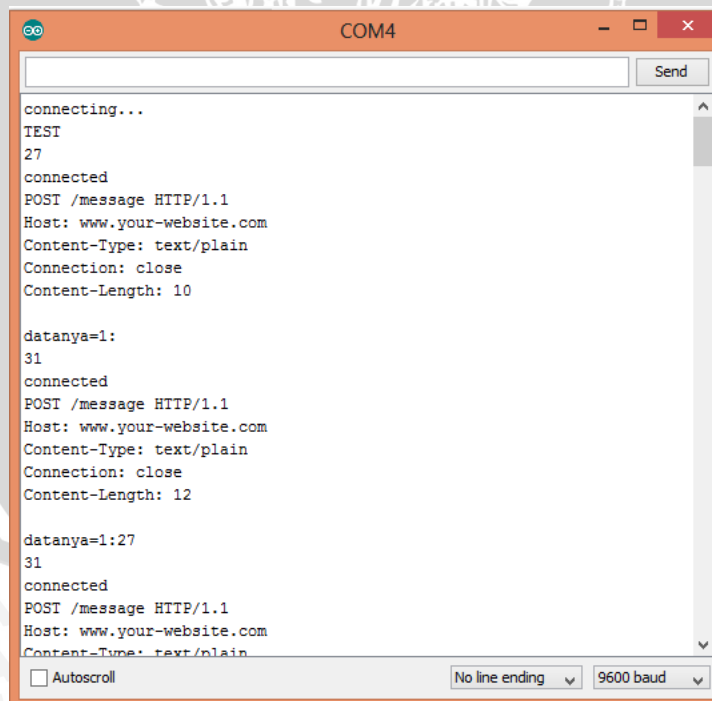
**Tabel 2 Hasil konversi dalam °C**

Data ke	Hasil konversi ke °C
1	30.68
2	66.87
3	56.75
4	30.0
5	29.0

### 5.1.2 Pengujian Arduino

Pengujian pengiriman data dari arduino dilakukan dengan pembacaan sensor LM35 berupa data Analog dirubah menjadi data Digital di dalam source code yang terdapat pada Arduino.

Pengiriman data dimulai dari alamat IP arduino (192.168.66.67) menuju alamat IP Ethernet Raspberry Pi (192.168.66.66) melalui Protokol HTTP POST. Pada gambar 5.1 dijelaskan bahwa pengiriman data dilakukan oleh alamat IP 192.168.66.67 yaitu Arduino menggunakan protokol HTTP POST.

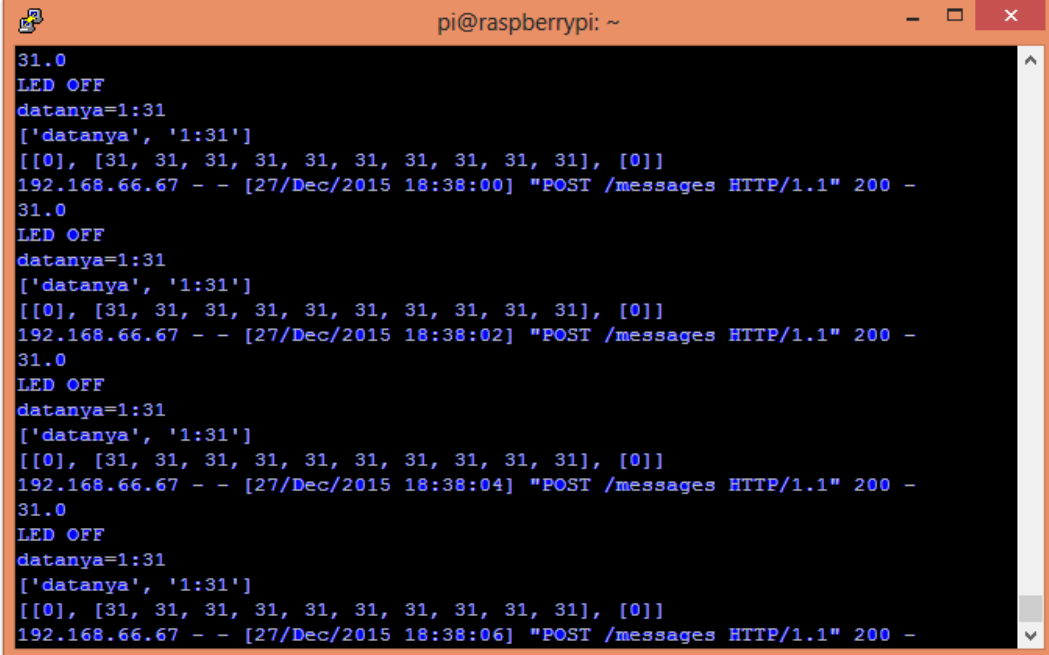


**Gambar 5. 1 Pengiriman data**

### 5.1.3 Pengujian Raspberry Pi

#### 5.1.3.1 Pengujian penerimaan data

Pengujian penerimaan data dilakukan setelah data dari sensor suhu LM35, diterima oleh Server Raspberry Pi. Data diterima per baris dalam bentuk angka. Dijelaskan pada gambar 5.2 data sensor diterima dari alamat IP 192.168.66.67 yaitu alamat dari Arduino.



```
pi@raspberrypi: ~  
31.0  
LED OFF  
datanya=1:31  
['datanya', '1:31']  
[[0], [31, 31, 31, 31, 31, 31, 31, 31, 31, 31], [0]]  
192.168.66.67 - - [27/Dec/2015 18:38:00] "POST /messages HTTP/1.1" 200 -  
31.0  
LED OFF  
datanya=1:31  
['datanya', '1:31']  
[[0], [31, 31, 31, 31, 31, 31, 31, 31, 31, 31], [0]]  
192.168.66.67 - - [27/Dec/2015 18:38:02] "POST /messages HTTP/1.1" 200 -  
31.0  
LED OFF  
datanya=1:31  
['datanya', '1:31']  
[[0], [31, 31, 31, 31, 31, 31, 31, 31, 31, 31], [0]]  
192.168.66.67 - - [27/Dec/2015 18:38:04] "POST /messages HTTP/1.1" 200 -  
31.0  
LED OFF  
datanya=1:31  
['datanya', '1:31']  
[[0], [31, 31, 31, 31, 31, 31, 31, 31, 31, 31], [0]]  
192.168.66.67 - - [27/Dec/2015 18:38:06] "POST /messages HTTP/1.1" 200 -
```

Gambar 5. 2 Pengujian penerimaan data

#### 5.1.3.2 Pengujian pengiriman data

Pengiriman data dari server ke Client menggunakan format JSON, dalam hal ini client meminta data pada sensor 1 yaitu sensor suhu. Pengujian pengiriman data menggunakan web browser seperti ditunjukkan pada gambar 5.3.

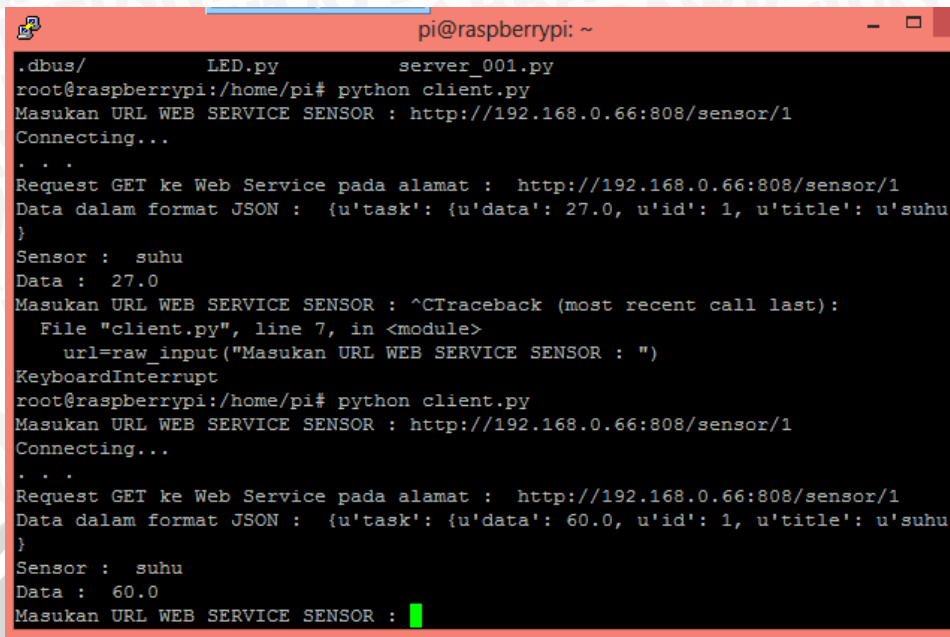


Gambar 5. 3 Pengujian pengiriman data

#### 5.1.4 Pengujian Client

Pemrograman Client menggunakan bahasa pemrograman python, setelah pengiriman data suhu dari sensor LM35, maka server dapat memberikan data jika client melakukan permintaan data sensor. Client meminta data dengan memasukan alamat `192.168.0.66:808/sensor/1`. `192.168.0.66` merupakan alamat IP dari Server pada socket Wifi, dan `/sensor/1` merupakan alamat dari data suhu LM35. Seperti ditunjukkan pada gambar 5.4





```
pi@raspberrypi: ~
.dbus/          LED.py          server_001.py
root@raspberrypi:/home/pi# python client.py
Masukan URL WEB SERVICE SENSOR : http://192.168.0.66:808/sensor/1
Connecting...
. . .
Request GET ke Web Service pada alamat : http://192.168.0.66:808/sensor/1
Data dalam format JSON : {"task": {"data": 27.0, "id": 1, "title": "suhu"
}}
Sensor : suhu
Data : 27.0
Masukan URL WEB SERVICE SENSOR : ^CTraceback (most recent call last):
  File "client.py", line 7, in <module>
    url=raw_input("Masukan URL WEB SERVICE SENSOR : ")
KeyboardInterrupt
root@raspberrypi:/home/pi# python client.py
Masukan URL WEB SERVICE SENSOR : http://192.168.0.66:808/sensor/1
Connecting...
. . .
Request GET ke Web Service pada alamat : http://192.168.0.66:808/sensor/1
Data dalam format JSON : {"task": {"data": 60.0, "id": 1, "title": "suhu"
}}
Sensor : suhu
Data : 60.0
Masukan URL WEB SERVICE SENSOR : █
```

**Gambar 5. 4 Pengujian client**

Data 60.0 pada gambar 60.0 merupakan rata-rata dari 10(sepuluh) data terakhir pada server raspberry pi.

## 5.2 Analisis

Sesuai data pengujian dari Arduino dan Server (Raspberry pi) menunjukkan bahwa hasil pengiriman dari Arduino ke Server dapat berkerja dengan baik. Arduino mengirimkan data dari sensor LM35 menggunakan protokol HTTP POST menuju ke Server.

Pada Server data dari Arduino dapat diterima oleh server, maksimum data yang diterima oleh Server dibatasi 10 (data). Jika data sudah mecapai batas yang ditentukan terdapat data yang masuk ke server maka data awal akan dihapus sehingga data yang baru dapat dimasukan ke dalam server. Pada sisi client, client meminta data pada server. Server menginformasikan pada client bahwa data dikirim melalui format data JSON, sehingga pada sisi client harus melakukan pemisahan data JSON agar dapat menampilkan data yang diminta. Hasil pengujian dengan *Web Service* menunjukkan bahwa keseluruhan program dapat berfungsi.

## BAB 6 PENUTUP

Bab ini berisi kesimpulan dari hasil perancangan, implementasi, dan analisis pengujian yang telah dilakukan, beserta saran yang dapat meningkatkan dan mengembangkan sistem yang telah ada.

### 6.1 Kesimpulan

Pada bab ini akan diambil kesimpulan dari perancangan, implementasi, dan pengujian yang telah dilakukan. Kesimpulan yang diambil meliputi:

1. Implementasi struktur data pada embedded sistem menggunakan format JSON merupakan gabungan komunikasi antara arduino dan raspberry. Pin adc pada arduino mengubah data analog dari sensor LM35 menjadi data digital, data digital dikirim ke raspberry pi menggunakan metode HTTP POST. Raspberry pi mengubah data dari arduino menjadi format JSON. Client meminta data ke raspberry pi menggunakan metode HTTP POST, kemudian client melakukan parsing data terhadap data kiriman dari raspberry pi.
2. Mekanisme komunikasi antar embedded sistem dengan menggunakan format JSON merupakan sebuah komunikasi antara arduino dengan raspberry pi. data sensor LM35 dari arduino dikirimkan menggunakan metode HTTP POST text/plain, data tersebut dikirimkan dari arduino menggunakan ethernet LAN, paket data dari arduino akan disimpan di raspberry pi. Raspberry pi hanya dapat menerima 10(data) dari arduino, jika ada data baru setelah raspberry pi menerima 10(data) maka data pertama yang masuk ke raspberry pi akan dihapus.
3. Mekanisme format data komunikasi JSON dengan perangkat AC dan DC merupakan output atau keluaran dari data sensor LM35 beroperasionalnya atau tidak tergantung dari keluaran sensor LM35.

### 6.2 Saran

1. Menambah perangkat AC dan DC pada implementasi sistem.
2. Menambah Arduino pada implementasi sistem.

## DAFTAR PUSTAKA

- Arduino, 2015. *Arduino.cc.* [Online]  
Available at: <http://arduino.cc/en/main/arduinoBoardUno>  
[Diakses 22 March 2015].
- CloudMailin, 2015. *incoming email for your web app.* [Online]  
Available at: [http://docs.cloudmailin.com/http\\_post\\_formats/](http://docs.cloudmailin.com/http_post_formats/)  
[Diakses 3 1 2016].
- Corbazon, A., 2010. Faculty of Economic Sciences. *Consuming Web Services on Mobile Platform*, p. 8.
- Datasheet, 2010. *www.alldatasheet.com.* [Online]  
Available at: <http://www.alldatasheet.com/datasheet-pdf/pdf/8866/NSC/LM35.html>  
[Diakses 15 September 2015].
- ekajogja, 2010. <http://ekajogja.com/>. [Online]  
Available at: <http://ekajogja.com/definisi/rest-representational-state-transfer/>  
[Diakses 5 10 2015].
- electrical4u, 2012. *www.electrical4u.com.* [Online]  
Available at: <http://www.electrical4u.com/light-dependent-resistor-ldr-working-principle-of-ldr/>  
[Diakses 15 September 2015].
- Hashmi, M. G. M., 2010. Journal. *A Sensor-based, web Service Enable, emergency Medical Response System*, p. 8.
- <http://www.json.org/>, t.thn. <http://www.json.org/>. [Online]  
Available at: <http://www.json.org/json-id.html>
- Pengenalan JSON, 2013. *Pengenalan JSON.* [Online]  
Available at: <http://www.json.org/json-id.html>  
[Diakses 1 12 2015].
- Python Software, F., 2001. <https://www.python.org/>. [Online]  
Available at: <https://www.python.org/>  
[Diakses 14 12 2015].
- Raspberry.org, 2015. *RASPBERRY PI 1 MODEL B+.* [Online]  
Available at: <https://www.raspberrypi.org/products/model-b-plus/>  
[Diakses 1 9 2015].
- Reynold, F., 2010. *The Ubiquitous Web, UPNP and Smart Homes.* s.l.:s.n.
- [www.w3.org](http://www.w3.org/), 2001. *Web Services Description Language (WSDL).* [Online]  
Available at: [www.w3.org/TR/ws-arch](http://www.w3.org/TR/ws-arch)  
[Diakses 9 September 2015].

## LAMPIRAN 1 PROGRAM ARDUINO

```
#include <Ethernet.h>
#include <SPI.h>
static char outstr[5];
//byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x55, 0x8D};
EthernetClient client;

float tempC=0;
int reading;
int tempPin = 5;

void setup()
{
  Serial.begin(9600);
  analogReference(INTERNAL);
  byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xF6, 0xFF };
  byte ip[] = { 192, 168, 66, 67};
  Ethernet.begin(mac,ip);
  delay(1000);
  Serial.println("connecting...");
}

void loop(){

  //data+=String(tempC,2);
  Serial.println("TEST")
  ;
  while (client.connect("192.168.66.66",808)) {
    reading = analogRead(tempPin);
    delay(100);
    tempC = reading / 9.31;
    String data;
    if (tempC>0){
```

```
data="datanya=1:";
data+=outstr;
dtostrf(tempC,2, 0, outstr);
Serial.println(outstr);

Serial.println("connected");
client.println("POST /messages HTTP/1.1");
client.println("Host: www.your-website.com");
client.println("Content-Type: text/plain");
client.println("Connection: close");
client.print("Content-Length: ");
client.println(data.length());
client.println();
client.print(data);
client.println();

//Prints your post request out for debugging
Serial.println("POST /message HTTP/1.1");
Serial.println("Host: www.your-website.com");
Serial.println("Content-Type: text/plain");
Serial.println("Connection: close");
Serial.print("Content-Length: ");
Serial.println(data.length());
Serial.println();
Serial.print(data);
Serial.println();
client.stop();
delay(2000);
}
}
}
```

## LAMPIRAN 2 PROGRAM SERVER

```
from flask import *
import re

app = Flask(__name__)
sensor=[[0],[0],[0]]

def make_public_task(task):
    new_task = {}
    for field in task:
        if field == 'id':
            new_task['uri'] = url_for('get_task',
task_id=task['id'], _external=True)
        else:
            new_task[field] = task[field]
    return new_task

tasks = [
    {
        'id': 0,
        'sensor': "xxx",
        'data': "35"
    },
    {
        'id': 1,
        'title': "suhu",
        'data': "30"
    },
    {
        'id': 2,
        'title': "xxx",
        'data': "20"
    }
]
```

```
]

@app.route('/', methods=['GET'])
def get_tasks():
    # return jsonify({'tasks': tasks})
    return jsonify({'tasks': [make_public_task(task) for
task in tasks]})

@app.route('/sensor/<int:task_id>', methods=['GET'])
def get_task(task_id):
    task = [task for task in tasks if task['id'] == task_id]

    task[0]['data']=sum(sensor[int(task[0]['id'])])/float(len(se
nsor[int(task[0]['id'])]))
    if len(task) == 0:
        abort(404)
    return jsonify({'task': task[0]})

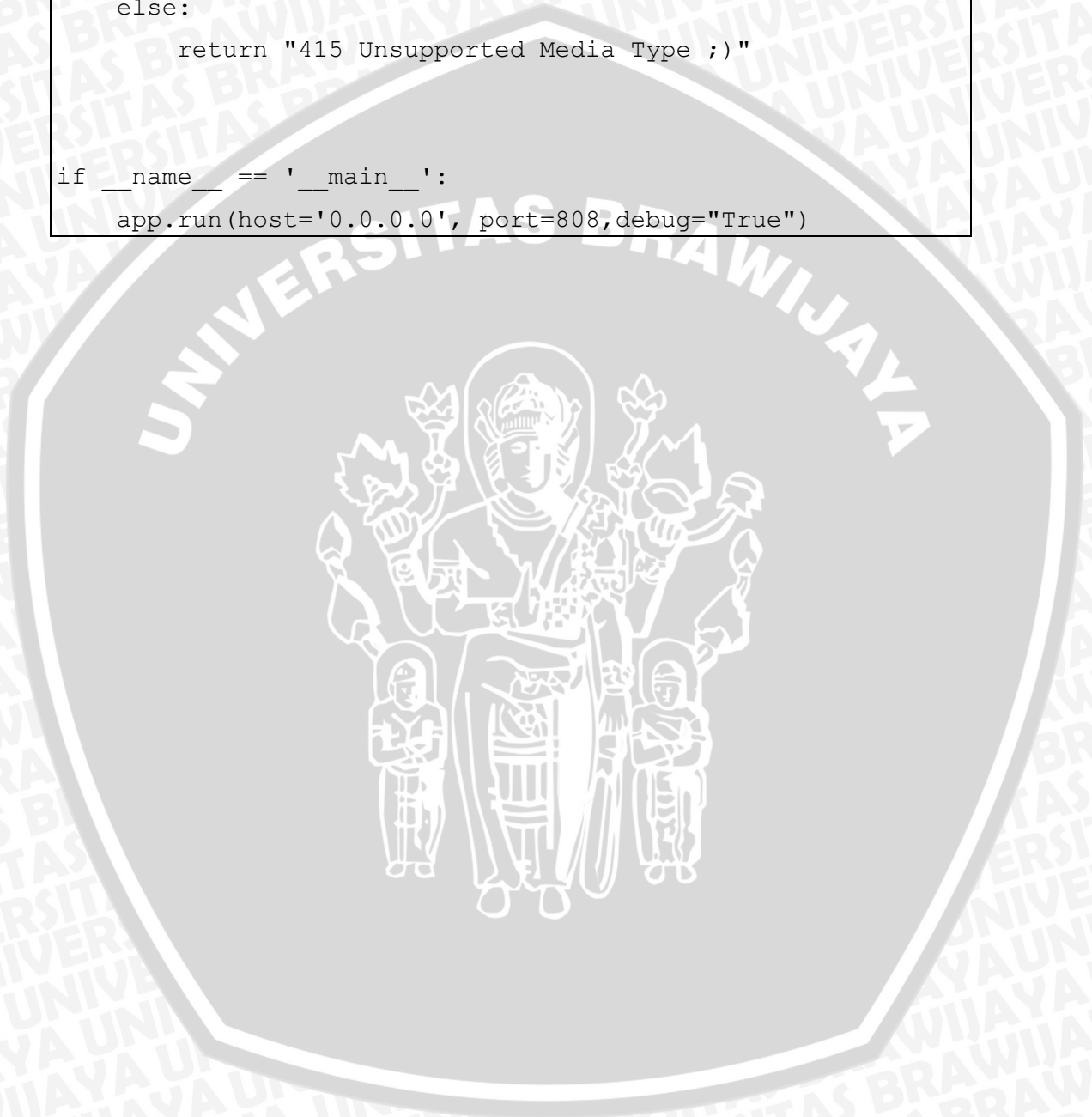
@app.route('/messages', methods = ['POST'])
def api_message():

    if request.headers['Content-Type'] == 'text/plain':
        data = request.data
        new_data = data.split('=')
        new_data_1 = new_data[1].split(':')
        if len(sensor[int(new_data_1[0])])<10:

            sensor[int(new_data_1[0])].append(int(new_data_1[1]))
        else:
            sensor[int(new_data_1[0])].pop(0)

        sensor[int(new_data_1[0])].append(int(new_data_1[1]))
        print data
        print new_data
        print sensor
        return "Text Message: " + request.data
```

```
#         elif request.headers['Content-Type'] ==  
'application/JSON':  
#         print "JSON Message: " + request.JSON['datanya']  
#         return "JSON Message: " + JSON.dumps(request.JSON)  
  
else:  
    return "415 Unsupported Media Type ;)"  
  
if __name__ == '__main__':  
    app.run(host='0.0.0.0', port=808, debug="True")
```





### LAMPIRAN 3 PROGRAM CLIENT

```
import urllib2
import JSON
import pprint
import time

while True:
    url=raw_input("Masukan URL WEB SERVICE SENSOR : ")
    print "Connecting "
    time.sleep(1)
    print ".",
    time.sleep(1)
    print ".",
    time.sleep(1)
    print "."
    req = urllib2.Request(url)
    opener = urllib2.build_opener()
    f = opener.open(req)
    JSON = JSON.loads(f.read())
    print "Request ke Web Service pada alamat : ",url
    print "Data dalam format JSON : ",JSON
    print "Sensor : ",JSON['task']['title']
    print "Data : ",JSON['task']['data']
```