

**SMART HOME SYSTEM MEMANFAATKAN INFRASTRUKTUR WEB
SERVICE DENGAN KONTROL BERBASIS ANDROID**

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

MUHAMMAD ADITYA DARMAWAN

NIM. 105060800111088

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2014

LEMBAR PERSETUJUAN

***SMART HOME SYSTEM MEMANFAATKAN INFRASTRUKTUR WEB
SERVICE DENGAN KONTROL BERBASIS ANDROID***

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

MUHAMMAD ADITYA DARMAWAN

NIM. 105060800111088

Skripsi ini telah diperiksa dan disetujui oleh dosen pembimbing pada tanggal

5 Juni 2014

Dosen Pembimbing I,

Dosen Pembimbing II,

Sabriansyah Rizqika Akbar, ST., M.Eng

NIK. 820809 06 1 1 0084

Issa Arwani, SKom., MSc.

NIK. 830922 06 1 1 0074

LEMBAR PENGESAHAN

***SMART HOME SYSTEM MEMANFAATKAN INFRASTRUKTUR WEB
SERVICE DENGAN KONTROL BERBASIS ANDROID***

SKRIPSI

Disusun Oleh :

MUHAMMAD ADITYA DARMAWAN

NIM. 105060800111088

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 27 Juni 2014

Penguji I,

Penguji II,

Wijaya Kurniawan, ST., MT
NIK. 820125 16 1 1 0418

Agung Setia Budi, ST., MT

Penguji III,

Hurriyatul Fitriyah, ST., M.Sc

Mengetahui,
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, MT
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam 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 yaitu UU No. 20 Tahun 2003, Pasal 25 Ayat (2) dan Pasal 70.

Undang-Undang No. 20 Tahun 2003 Pasal 25 Ayat (2) yang berisi “Lulusan perguruan tinggi yang karya ilmiahnya digunakan untuk memperoleh gelar akademik, profesi, atau vokasi terbukti merupakan jiplakan dicabut gelarnya” dan Pasal 70 yang berisi “Lulusan yang karya ilmiah yang digunakannya untuk mendapatkan gelar akademik, profesi, atau vokasi sebagaimana dimaksud dalam Pasal 25 Ayat (2) terbukti merupakan jiplakan dipidana dengan pidana penjara paling lama dua tahun dan/atau pidana denda paling banyak Rp. 200.000.000,00 (dua ratus juta rupiah)”.

Malang, 5 Juni 2014

Mahasiswa,

Muhammad Aditya Darmawan

NIM. 105060800111088

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas limpahan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan proposal skripsi dengan judul “*Smart Home System* Memanfaatkan Infrastruktur *Web Service* dengan Kontrol berbasis Android”.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan proposal skripsi, diantaranya:

1. Allah SWT atas rahmat dan hidayah yang telah diberikan.
2. Rasulullah Muhammad SAW, semoga shalawat serta salam selalu tercurah kepada beliau.
3. Kedua orang tua penulis, Ayahanda Herry Pratama Sriwidjayanto, SE dan Ibunda Ir. Susi Adiningsih yang selalu tidak lepas dari do’a dan harapan untuk terselesaikannya skripsi ini dan terus memberikan dorongan moral, material dan kasih sayangnya tiada akhir.
4. Kakak penulis Hersynanda Karyadi Utama serta Adik penulis Sabrina Aisha Alyani yang senantiasa memberi do’a dan motivasi. Tak lupa juga penulis berterima kasih untuk segenap keluarga besar penulis.
5. Bapak Sabriansyah R.A, ST.,M.Eng. selaku dosen pembimbing I yang telah memberikan ilmu dan saran untuk laporan skripsi ini.
6. Bapak Issa Arwani, S.Kom.,M.Sc selaku dosen pembimbing II yang juga memberikan ilmu dan saran untuk laporan skripsi ini.
7. Teman-teman yang telah membantu memberi kritik dan saran.

Akhir kata, penulis menyadari bahwa laporan skripsi ini masih jauh dari kata sempurna, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan laporan skripsi ini. Penulis berharap laporan skripsi ini dapat bermanfaat khususnya bagi diri penulis sendiri dan bagi semua pihak. Amin.

Malang, Juni 2014

Penulis

ABSTRAK

Muhammad Aditya Darmawan, Jurusan Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya Malang, Juni 2014, *Smart Home System Memanfaatkan Infrastruktur Web Service dengan Kontrol Berbasis Android*.

Dosen Pembimbing 1 : Sabriansyah Rizqika Akbar, ST.,M.Eng

Dosen Pembimbing 2 : Issa Arwani, S.Kom.,M.Sc.

Smart home system merupakan suatu sistem yang dapat digunakan untuk memonitor dan mengendalikan berbagai perangkat elektronik dari jarak jauh dengan menggunakan *remote control* atau perangkat bergerak. *Web service* diperlukan pada *smart home system* sebagai sarana untuk menjalin komunikasi antara perangkat bergerak dengan *board* mikrokontroler. Penelitian ini mengimplementasikan *smart home system* menggunakan perangkat bergerak pada rangkaian komponen fisik yang terdiri dari 4 buah LED beserta saklarnya yang akan terhubung dengan *board* mikrokontroler. *Board* mikrokontroler akan terhubung ke sebuah komputer yang berfungsi sebagai server yang menyediakan layanan *web service*. Tahap pengujian dari penelitian ini adalah dengan menguji fungsionalitas dan validasi dari akses *web service* dan sistem secara keseluruhan. Hasil yang diperoleh adalah *web service* dapat diimplementasikan sebagai salah satu penghubung antara perangkat bergerak dengan *board* mikrokontroler yang berbeda bahasa pemrograman untuk menjalin komunikasi data.

Kata kunci : *smart home system*, *web service*, mikrokontroler, perangkat bergerak

ABSTRACT

Muhammad Aditya Darmawan, *Majors of Informatics, Information Technology and Computer Science Program, Brawijaya University Malang, June 2014, Smart Home System Utilizing Infrastructure Web Service with Android Based Control.*

Supervisor 1 : Sabriansyah Rizqika Akbar, ST.,M.Eng

Supervisor 2 : Issa Arwani, S.Kom.,M.Sc.

Smart home system is a system that can be used to monitor and control various electronic devices remotely using the remote control or with a mobile devices. Web service is required in the smart home system because can be used to establish communication between mobile devices and microcontroller board. This research has been implemented a smart home system using mobile device on the physical components of the circuit consist of 4 LEDs with the switch that connected to the microcontroller board. Microcontroller board was connected to a computer as a server that provides the web service. The testing phase of this research to test the functionality and validation of access the web service and the overall system. The result obtained that web service can be implemented to connect mobile devices and microcontroller board with different programming languages to establish data communication.

Keywords : *smart home system, web service, microcontroller, mobile devices*



DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Sistematika Penulisan.....	3
BAB II.....	4
TINJAUAN PUSTAKA	4
2.1 Kajian Pustaka	4
2.2 <i>Smart Home</i>	5
2.3 Mikrokontroler	5
2.3.1 Arduino	6
2.3.2 Arduino Uno	8
2.3.3 <i>Ethernet shield</i>	10
2.4 Android.....	11
2.5 <i>Web Service</i>	11

2.5.1	SOAP <i>Web Service</i>	12
2.5.2	REST <i>Web Service</i>	13
2.6	JSON	14
BAB III		16
METODOLOGI PENELITIAN.....		16
3.1	Studi Literatur.....	17
3.2	Analisis Kebutuhan	17
3.3	Perancangan.....	17
3.4	Implementasi	18
3.5	Pengujian	19
3.6	Pengambilan Kesimpulan dan Saran	19
BAB IV		20
ANALISIS DAN PERANCANGAN.....		20
4.1	Analisis Kebutuhan	20
4.1.1	Gambaran Umum Sistem	20
4.1.2	Kebutuhan Antar Muka.....	22
4.1.3	Analisa Perangkat Keras dan Perangkat Lunak	22
4.2	Perancangan.....	23
4.2.1	Perancangan Komponen Fisik.....	23
4.2.2	Perancangan Program pada Web Service	24
4.2.3	Perancangan Program pada Arduino.....	25
4.2.4	Perancangan pada Android	27
4.2.5	Perancangan Basis Data	29
BAB V.....		31
IMPLEMENTASI DAN PENGUJIAN SISTEM		31
5.1	Implementasi Sistem	31



5.1.1	Lingkungan Implementasi Sistem.....	31
5.1.2	Batasan – Batasan Implementasi.....	32
5.1.3	Implementasi Komponen Fisik	32
5.1.4	Implementasi Program pada <i>Web Service</i>	34
5.1.5	Implementasi Program pada Arduino	38
5.1.6	Implementasi pada Android	45
5.1.7	Implementasi Basis Data.....	54
5.2	Pengujian	55
5.2.1	Pengujian Akses <i>Web Service</i>	55
5.2.2	Pengujian Sistem Keseluruhan.....	57
5.3	Analisa.....	61
5.3.1	Analisa Pengujian Akses <i>Web Service</i>	61
5.3.2	Analisa Pengujian Sistem Keseluruhan	61
BAB VI.....		62
PENUTUP.....		62
6.1	Kesimpulan.....	62
6.2	Saran	63
DAFTAR PUSTAKA		64

DAFTAR GAMBAR

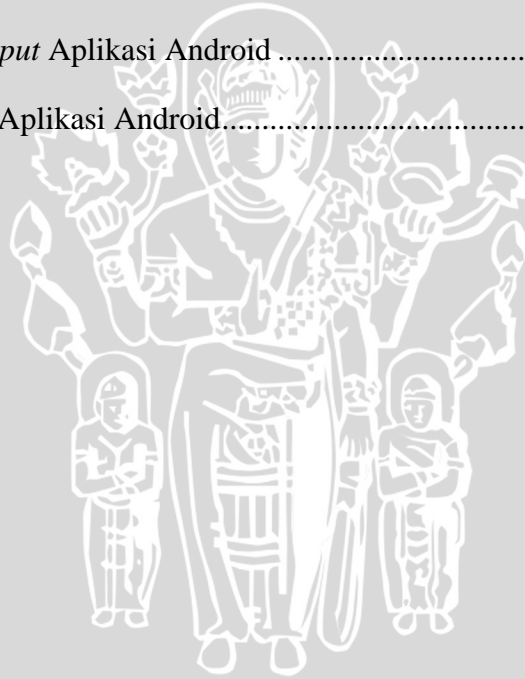
Gambar 2.1 Arduino.....	6
Gambar 2.2 Tampilan IDE Arduino.....	7
Gambar 2.3 Bagian-Bagian Papan Arduino Uno.....	8
Gambar 2.4 <i>Ethernet Shield</i>	10
Gambar 2.5 Aplikasi Penunjang untuk Membangun Aplikasi Android	11
Gambar 2.6 Kerangka pesan SOAP.....	13
Gambar 2.7 Struktur Object JSON	14
Gambar 2.8 Struktur Array JSON.....	15
Gambar 2.9 Contoh JSON.....	15
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	16
Gambar 4.1 Alur Kerja Sistem.....	21
Gambar 4.2 Rangkaian Komponen Fisik	23
Gambar 4.3 Diagram Alir Program pada <i>Web Service</i>	24
Gambar 4.4 Diagram Alir Program pada Arduino.....	25
Gambar 4.5 Diagram Alir Proses Menerima <i>Request</i> Data pada Arduino	26
Gambar 4.6 Diagram Alir Proses Mengirimkan <i>Request</i> Data pada Arduino	27
Gambar 4.7 Diagram Alir Program pada Android.....	28
Gambar 4.8 Rancangan Antar Muka Tampilan Utama Aplikasi	29
Gambar 4.9 Entity Relational Diagram.....	30
Gambar 5.1 Hasil Implementasi Rangkaian Komponen Fisik.....	34
Gambar 5.2 <i>Web Service</i> Menerima Data dari Arduino	36
Gambar 5.3 <i>Web Service</i> Mengirimkan Data ke Arduino.....	38
Gambar 5.4 Konfigurasi Jaringan Arduino	39
Gambar 5.5 Fungsi Menerima <i>Request</i> Data pada Arduino	41

Gambar 5.6 Fungsi Mengirimkan <i>Request</i> Data Pada Arduino.....	44
Gambar 5.7 Fungsi Mengambil Semua Data di <i>Web Service</i> pada Android	47
Gambar 5.8 <i>Syntax</i> Mengambil Data Berdasarkan Pin dan Status pada Android	49
Gambar 5.9 <i>Class JSON Parser</i> pada Android.....	51
Gambar 5.10 Tampilan <i>Flash Screen</i> Aplikasi Android.....	51
Gambar 5.11 Tampilan Awal Memasukkan IP Aplikasi Android.....	52
Gambar 5.12 Tampilan Menu Utama Aplikasi Android.....	52
Gambar 5.13 Tampilan Menu <i>Information About</i> Aplikasi Android.....	53
Gambar 5.14 Tampilan Menu <i>Exit</i> Aplikasi Android.....	53
Gambar 5.15 Implementasi basis data	54
Gambar 5.16 Topologi Jaringan <i>Ad Hoc</i> untuk Pengujian	55



DAFTAR TABEL

Tabel 2.1 Fungsi <i>Input & Output</i>	7
Tabel 2.2 Spesifikasi Arduino Uno	8
Tabel 4.1 Tabel Kebutuhan Fungsional	22
Tabel 4.2 Rancangan tabel untuk menyimpan data	30
Tabel 5.1 Rancangan tabel untuk menyimpan data	54
Tabel 5.2 Pengujian <i>Web Service</i>	56
Tabel 5.3 Pengujian <i>Input</i> Saklar Manual	57
Tabel 5.4 Pembuktian <i>Input</i> Saklar Manual	58
Tabel 5.5 Pengujian <i>Input</i> Aplikasi Android	59
Tabel 5.6 Pembuktian Aplikasi Android	60



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi memunculkan sebuah sistem rumah pintar atau disebut dengan istilah *smart home system*. Teknologi ini dibutuhkan untuk dapat memudahkan pekerjaan manusia agar menjadi lebih cepat, efisien dan efektif. *Smart home system* merupakan suatu sistem yang dapat digunakan untuk memonitor dan mengendalikan berbagai perangkat elektronik rumah tangga dari jarak jauh dengan menggunakan pengendali jarak jauh, misalnya seperti *remote control* atau perangkat bergerak [SMA-13]. Menurut hasil survei untuk pengguna perangkat bergerak di Indonesia, menunjukkan bahwa pengguna mulai beralih ke perangkat ponsel pintar (*smartphone*) berjenis android. Perangkat bergerak ini memiliki jumlah pengguna tertinggi dibandingkan perangkat bergerak dengan jenis sistem operasi lain. Sesuai dengan laporan dari lembaga riset berskala internasional yang berpusat di Jerman yang berkompeten dalam mengkaji pasar ritel dan teknologi yaitu GfK (*Growth from Knowledge*) yang dirilis pada Mei 2013, *smartphone* berbasis android ini telah mendominasi pasar Indonesia sebesar 51 persen, dan terungkap pula sebanyak 4,5 juta *smartphone* yang berhasil terjual di dalam negeri selama bulan Januari-Maret 2013, sebanyak 2,28 juta di antaranya menjalankan OS Android [YUS-13].

Sarana untuk menghubungkan antara perangkat bergerak dengan *board* mikrokontroler pada rangkaian *smart home system* yaitu diperlukan sebuah layanan *web service*. Meskipun berbeda *platform* dan juga bahasa pemrograman, dengan menggunakan *web service* inilah, komunikasi data akan tetap terbentuk. Oleh karena itu pada penelitian ini akan membuat rancangan rangkaian *smart home system* yang memanfaatkan infrastruktur *web service* dengan kontrol menggunakan perangkat bergerak. Model arsitektur *web service* sendiri terdiri dari dua macam yaitu SOAP dan REST [ADA-11]. *Web service* SOAP merupakan model arsitektur berbasis XML dimana dalam melakukan pertukaran pesan berisikan elemen-elemen berikut yaitu *envelope element*,

header element, *body element*, dan *fault element* [JEE-11]. Sedangkan untuk *web service* jenis REST jauh lebih sederhana karena pada intinya REST berbasiskan *resource* yang artinya setiap URL itu unik yang merepresentasikan suatu objek [AJI-13]. Pada penelitian ini, *web service* yang dibuat termasuk dalam kategori jenis REST.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka dapat diambil beberapa rumusan masalah pada penelitian ini yaitu :

1. Bagaimana cara merancang dan mengimplementasikan *web service* untuk *smart home system* ?
2. Bagaimana cara merancang dan mengimplementasikan program pada *board* mikrokontroler untuk *smart home system* ?
3. Bagaimana cara merancang dan mengimplementasikan aplikasi perangkat bergerak untuk *smart home system* ?

1.3 Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal:

1. Membahas *smart home system* untuk rangkaian sistem monitoring dan kontrol lampu.
2. Sistem operasi perangkat bergerak berjenis android.
3. *Board* mikrokontroler yang digunakan adalah jenis Arduino Uno dan *ethernet shield* yang berbasis chip WIZNET W5100.

1.4 Tujuan

Tujuan dari penyusunan penelitian ini yaitu untuk merancang dan merealisasikan pemanfaatan infrastruktur *web service* pada rangkaian *smart home system* untuk sistem monitoring dan kontrol lampu dengan kontrol berbasis android.

1.5 Manfaat

Penulisan penelitian ini diharapkan mempunyai manfaat yang baik dan berguna bagi pembaca dan penulis. Adapun beberapa manfaat yang diharapkan yaitu :

1. Dapat mengimplementasikan pemanfaatan infrastruktur *web service* pada rangkaian *smart home system* untuk mengontrol dan memonitoring keadaan lampu dengan menggunakan aplikasi yang terinstall di perangkat bergerak.
2. Dapat mengetahui kegunaan dari *web service* pada rangkaian *smart home system*, yaitu untuk menjalin komunikasi data antara perangkat bergerak dengan *board* mikrokontroler yang berbeda *platform* dan juga bahasa pemrograman.

1.6 Sistematika Penulisan

Sistematika penulisan laporan ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II Tinjauan Pustaka

Menguraikan tentang kajian pustaka dan dasar teori yang digunakan.

BAB III Metodologi Penelitian

Menjelaskan metode yang digunakan dalam pengerjaan laporan.

BAB IV Analisis dan Perancangan

Membahas analisis kebutuhan dan perancangan yang sesuai dengan teori yang ada.

BAB V Implementasi dan Pengujian Sistem

Membahas tentang implementasi dan hasil pengujian dari sistem.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian sistem yang dikembangkan dalam penelitian ini serta saran - saran untuk pengembangan lebih lanjut.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi tinjauan pustaka yang meliputi kajian pustaka dan dasar teori yang diperlukan untuk penelitian. Kajian pustaka adalah membahas penelitian yang telah ada dan yang diusulkan. Dasar teori adalah membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan.

Kajian pustaka pada penelitian ini adalah berasal dari sebuah jurnal internasional yang berjudul "*Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone*". Teori dasar yang akan dibahas pada bab ini yaitu konsep dasar *Smart Home*, Mikrokontroler, Android, *Web Service*, dan JSON.

2.1 Kajian Pustaka

Jurnal internasional yang berjudul "*Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone*" oleh Rajeev Piyare pada tahun 2013, berisikan tentang sistem kontrol dan monitoring terhadap berbagai peralatan listrik menggunakan android [PIY-13]. Pada jurnal ini mengkaji tentang penerapan *web service* yang melakukan komunikasi antara android dengan arduino.

Sedangkan pada penelitian ini akan membahas tentang *smart home system* yang memanfaatkan infrastruktur *web service* dengan kontrol berbasis android yang di implementasikan untuk mengontrol dan memonitoring kondisi nyala lampu. Terdapat perbedaan yang dilaksanakan oleh penulis dengan jurnal yaitu perbedaan pertama adalah dalam pengimplementasiannya dengan menggunakan *web service* yang terletak pada server tersendiri dan menggunakan mysql sebagai *database*. Perbedaan kedua adalah pada penelitian ini diharapkan sistem dapat mengetahui kondisi keadaan lampu ketika diberikan *inputan* secara manual dengan menggunakan saklar.

2.2 Smart Home

Smart home merupakan sistem yang diterapkan di rumah, apartemen atau pada gedung dengan bantuan komputer yang bermanfaat untuk memberikan segala kenyamanan, keselamatan, keamanan dan penghematan energi yang berlangsung secara otomatis dan terprogram. Dapat digunakan untuk mengendalikan segala peralatan yang ada di rumah. Sedangkan pada tahun 2003 menurut *UK Department of Trade and Industry* (DTI) mengutarakan definisi dari *smart home system* adalah sebuah hunian yang dilengkapi dengan jaringan komunikasi yang menghubungkan dengan peralatan dan layanan elektrik, yang memungkinkan untuk dikendalikan, dipantau ataupun diakses dari jarak jauh [SMA-13].

Saat ini, seiring dengan pesatnya perkembangan teknologi yang berjalan dinamis, memberikan efek yang sangat signifikan. Contohnya saja, dengan kemajuan teknologi dapat memudahkan pekerjaan manusia menjadi lebih efisien, efektif, cepat dan murah. *Smart home* merupakan salah satu teknologi yang bagus, karena dapat melakukan pekerjaan apapun tanpa harus merepotkan sang pemilik rumah. Konsep yang ingin ditampilkan dari sistem *smart home* ini adalah bagaimana produk teknologi canggih dapat diaplikasikan untuk membantu penghuni rumah untuk melakukan pekerjaan rumah di manapun mereka berada. Ciri khas yang ingin ditunjukkan dengan diciptakannya *smart home* ini adalah bisa memberikan kenyamanan dan keamanan penghuninya. *Smart home* dapat digunakan dengan mudah, karena biasanya *smart home* dikendalikan dengan remote control. Bahkan dengan kemajuan teknologi saat ini, *smart home* dapat dikendalikan melalui perangkat *mobile* yang dimiliki oleh para pengguna sistem ini.

2.3 Mikrokontroler

Mikrokontroler adalah otak dari suatu sistem elektronika seperti halnya mikroprosesor sebagai otak komputer dimana berfungsi sebagai pengendali atau pengontrol [WAR-06]. Namun mikrokontroler memiliki nilai tambah karena didalamnya sudah terdapat memori dan sistem *input* dan *output* dalam satu kemasan IC. Chip IC (*Integrated Circuit*) ini bisa diprogram dengan

menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Terdapat berbagai macam produsen mikrokontroler antara lain seperti *ATMEL*, *Microchip*, *Maxim*, *Renesas*, dan *philips*. Penelitian ini menggunakan Arduino sebagai mikrokontrolernya. Sebuah chip mikrokontroler yang digunakan pada arduino adalah berjenis *AVR* dari perusahaan *ATMEL*.

2.3.1 Arduino

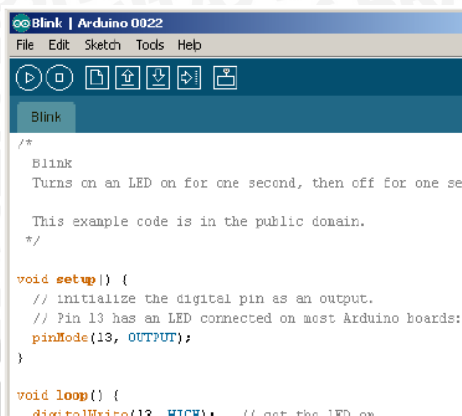
Arduino dikatakan sebagai sebuah *platform* dari *physical computing* yang bersifat *open source* baik secara *hardware* maupun *software* [BAN-08]. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi ia adalah kombinasi dari hardware, bahasa pemrograman dan *Integrated Development Environment (IDE)* yang canggih [DJU-11]. Gambar 2.1 berikut ini merupakan contoh dari Arduino.



Gambar 2.1 Arduino

Sumber : [DJU-11]

IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam *memory microcontroller*. Bahasa pemrograman yang digunakan merupakan penggabungan antara bahasa C++ dan Java. Software Arduino ini dapat di *install* di berbagai sistem operasi seperti *linux*, *Mac OS*, maupun *Windows*. Gambar 2.2 berikut ini merupakan contoh dari tampilan *IDE* Arduino dengan sebuah *listing* program atau biasa disebut *sketch* yang sedang diedit.



Gambar 2.2 Tampilan IDE Arduino

Sumber : [DJU-11]

Struktur perintah pada bahasa pemrograman Arduino secara garis besar terdiri dari 2 (dua) bagian yaitu :

- *Void setup* berisi perintah yang akan dieksekusi hanya satu kali sejak Arduino dihidupkan.
- *Void loop* berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan.

Terdapat juga beberapa fungsi *input* dan *output* yang dimiliki bahasa pemrograman Arduino yang dijelaskan sesuai dengan tabel 2.1 berikut.

Tabel 2.1 Fungsi *Input & Output*

Fungsi	Syntax	Parameter	Return
<i>digitalRead</i>	<i>digitalRead(pin)</i>	Nomor pin	<i>HIGH / LOW</i>
<i>analogRead</i>	<i>analogRead(pin)</i>	Nomor pin	0 – 1023
<i>digitalWrite</i>	<i>digitalWrite(pin,value)</i>	- Nomor pin - Nilai (<i>HIGH / LOW</i>)	-
<i>analogWrite</i>	<i>analogWrite(pin,value)</i>	- Nomor pin - <i>Duty Cycle</i> (0-255)	-
<i>pinMode</i>	<i>pinMode(pin,mode)</i>	- Nomor pin - <i>Mode</i> (<i>INPUT / OUTPUT</i>)	-



Penelitian ini menggunakan jenis Arduino Uno sebagai mikrokontroler nya.

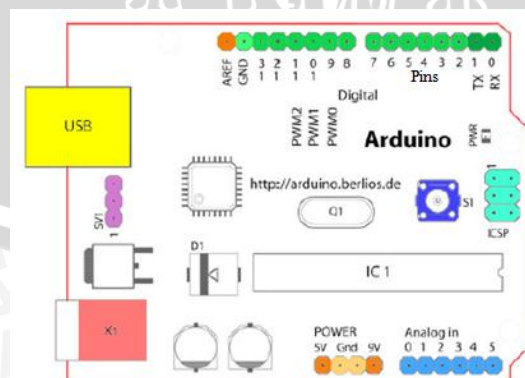
2.3.2 Arduino Uno

Arduino terdiri dari beberapa jenis yang memiliki fitur dan fungsi yang berbeda antara satu dengan yang lainnya. Arduino Uno adalah salah satu jenis dari arduino yang ada. Pada tabel 2.2 merupakan spesifikasi lengkap dari Arduino Uno:

Tabel 2.2 Spesifikasi Arduino Uno

<i>Microcontroller</i>	<i>Atmega328</i>
<i>Operating Voltage</i>	5V
<i>Input Voltage (disarankan)</i>	7-12V
<i>Input Voltage (batas)</i>	6-20V
<i>Digital I/O</i>	Pin 14 (6 memberikan <i>output</i> PWM)
<i>Analog Input</i>	Pin 6
<i>DC Current per I/O</i>	Pin 40 mA
<i>DC Current for 3.3v</i>	Pin 50 Ma
<i>Flash Memory 32 KB (Atmega328)</i>	0,5 KB digunakan oleh <i>bootloader</i>
<i>SRAM</i>	2 KB (Atmega328)
<i>EEPROM</i>	1 KB (Atmega328)
<i>Clock Speed</i>	16 MHz

Dibawah ini akan dijelaskan bagian-bagian dari papan Arduino Uno yang digambarkan pada gambar 2.3 berikut ini .



Gambar 2.3 Bagian-Bagian Papan Arduino Uno

Sumber : [DJU-11]

- 14 pin *input/output digital* (0-13)

Berfungsi sebagai *input* atau *output*, dapat diatur oleh program. Khusus untuk 6 buah pin 3, 5, 6, 9, 10 dan 11, dapat juga berfungsi sebagai pin analog *output* dimana tegangan *output*-nya dapat diatur. Nilai sebuah pin *output* analog dapat diprogram antara 0 – 255, dimana hal itu mewakili nilai tegangan 0 – 5V.
- USB

Berfungsi untuk :

 - Memuat program dari komputer ke dalam papan
 - Komunikasi serial antara papan dan komputer
 - Memberi daya listrik kepada papan
- Sambungan SV1

Sambungan atau *jumper* untuk memilih sumber daya papan, apakah dari sumber eksternal atau menggunakan USB. Sambungan ini tidak diperlukan lagi pada papan Arduino versi terakhir karena pemilihan sumber daya eksternal atau USB dilakukan secara otomatis.
- Q1 - Kristal (*quartz crystal oscillator*)

Jika microcontroller dianggap sebagai sebuah otak, maka kristal adalah jantung-nya karena komponen ini menghasilkan detak-detak yang dikirim kepada microcontroller agar melakukan sebuah operasi untuk setiap detak-nya. Kristal ini dipilih yang berdetak 16 juta kali per detik (16MHz).
- Tombol *Reset S1*

Untuk me-reset papan sehingga program akan mulai lagi dari awal. Perhatikan bahwa tombol reset ini bukan untuk menghapus program atau mengosongkan *microcontroller*.
- *In-Circuit Serial Programming* (ICSP)

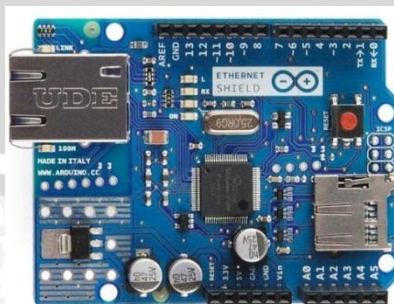
Port ICSP memungkinkan pengguna untuk memprogram microcontroller secara langsung, tanpa melalui bootloader. Umumnya pengguna Arduino tidak melakukan ini sehingga ICSP tidak terlalu dipakai walaupun disediakan.
- IC1 - Mikrokontroler Atmega

Komponen utama dari papan Arduino, di dalamnya terdapat CPU, ROM dan RAM.

- X1 - sumber daya eksternal
Jika hendak disuplai dengan sumber daya eksternal, papan Arduino dapat diberikan tegangan DC antara 9-12V.
- 6 pin *input* analog (0-5)
Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin *input* antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

2.3.3 *Ethernet shield*

Ethernet Shield digunakan untuk menghubungkan Arduino ke jaringan komputer. *Ethernet shield* berbasiskan *chip* ethernet Wiznet W5100. *Ethernet library* digunakan dalam menulis program agar arduino *board* dapat terhubung ke jaringan dengan menggunakan Arduino *ethernet shield*. Pada *ethernet shield* terdapat sebuah *slot* micro-SD, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan. *Onboard* menggunakan *SD library*. Arduino *board* berkomunikasi dengan W5100 dan *SD card* menggunakan bus SPI (*Serial Peripheral Interface*). Komunikasi ini diatur oleh library SPI.h dan Ethernet.h. Bus SPI menggunakan pin digital 11,12, dan 13 pada Arduino Uno. Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih *SD card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input/output* umum ketika kita menggunakan *ethernet shield*. Dalam penelitian ini dibutuhkan sebuah *ethernet shield* yang dihubungkan dengan Arduino sehingga sistem pada perangkat ini dapat terhubung ke jaringan internet. Gambar 2.4 berikut ini merupakan *ethernet shield*.

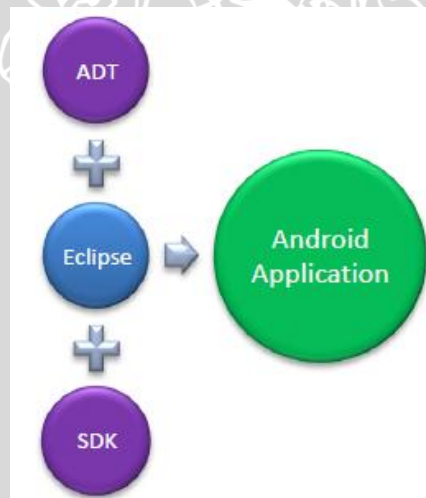


Gambar 2.4 *Ethernet Shield*

Sumber : [DJU-11]

2.4 Android

Android merupakan sistem operasi yang berbasis Linux yang penggunaannya ditujukan untuk perangkat bergerak. Salah satu kelebihan dari Android yang bersifat *open platform* sehingga para *developer* dapat mengembangkan berbagai jenis aplikasi yang dapat berjalan di sistem operasi jenis ini. Untuk membangun sebuah aplikasi android dibutuhkan beberapa aplikasi penunjang yaitu *Eclipse*, *Android Development Tool (ADT)*, dan *Software Development Kit (SDK)*. Eclipse merupakan sebuah editor, secara default editor ini belum bisa dipakai untuk mendevlop android. Agar bisa digunakan untuk membuat aplikasi android maka harus diinstall plugin dulu namanya *ADT (Android Development Tools)*. Setelah terinstal, maka eclipse sudah siap digunakan, hanya saja belum dilengkapi *library* dan emulator. Untuk memenuhinya, maka perlu diinstal *SDK (Software Development Kit)* Android. Sehingga dapat dijelaskan pada gambar 2.5 berikut ini.



Gambar 2.5 Aplikasi Penunjang untuk Membangun Aplikasi Android

Sumber : [AKB-12]

2.5 Web Service

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki interface yang dideskripsikan dalam format yang dapat dibaca

oleh mesin. Menurut *World Wide Web Consortium* (<http://www.w3.org>) yang merupakan badan yang menciptakan dan mengembangkan standar *web service* mendefinisikan bahwa *web service* merupakan sistem komputer yang saling bertukar XML message dengan sistem lain yang menggunakan HTTP sebagai protokol komunikasinya [WWW-04].

Ide untuk memanfaatkan teknologi *web* mulai terealisasi secara nyata ketika pada tahun 1990-an akhir dan salah satu pendekatan implementasi *web service* yang paling banyak digunakan disebut dengan *Simple Access Object Protocol* (SOAP) yang memanfaatkan *Extensible Markup Language* (XML), yang belum lama muncul saat itu sebagai paket format untuk mekanisme *Remote Procedure Call* (RPC). Beberapa tahun setelah kepopuleran *web service* berbasis SOAP muncul model implementasi baru yang dinamakan *REpresentational State Transfer* (REST) dan karena prinsip tersebut menggunakan model REST maka *web service* yang menerapkannya disebut *RESTful Web Services*. Dari uraian tersebut menyatakan bahwa *web service* dapat dibagi menjadi dua jenis, yaitu SOAP dan REST.

2.5.1 SOAP Web Service

SOAP (*Simple Object Access Protocol*) merupakan protokol komunikasi berbasis XML yang memperbolehkan aplikasi saling bertukar informasi melalui HTTP. SOAP merupakan format untuk mengirimkan *message* melalui Internet, bersifat *platform independent*, *language independent*, dan merupakan standar W3C. Sebuah pesan SOAP adalah sebuah dokumen XML yang berisi elemen-elemen berikut [JEE-11] :

1. *Envelope element* yang mengidentifikasi dokumen XML sebagai sebuah pesan SOAP.
2. Elemen *header* yang berisi informasi *header*. (bersifat opsional)
3. Elemen *body* yang berisi panggilan dan merespon informasi.
4. *Fault element* yang berisi pesan kesalahan yang terjadi pada waktu proses. (bersifat opsional)

Gambar 2.6 berikut merupakan contoh kerangka pesan SOAP.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-
encoding">
<soap:Header>
...
</soap:Header>
    <soap:Body">
        ...
        <soap:Fault>
            ...
        </soap:Fault>
    </soap:Body>
</soap:Envelope>
```

Gambar 2.6 Kerangka pesan SOAP

Sumber : [JEE-11]

2.5.2 REST Web Service

REST (*Representational State Transfer*) merupakan sebuah model arsitektur yang menghubungkan jalur komunikasi antar mesin/aplikasi melalui protokol HTTP yang berbasis *synchronous request/response*. Proses transfer data dimulai ketika *client* mengirimkan sebuah *request message* yang mencakup HTTP method yang akan dipanggil, lokasi *resource* dalam format URI, serta pilihan format pesan. Kemudian server memproses permintaan dari *client* dan meresponse balik dengan mengirimkan data yang sesuai dengan permintaan *client* dalam format apapun. Namun sekarang ini format yang banyak dikenali oleh *client* adalah JSON.

Berikut ini adalah beberapa ciri dari REST [AJI-13]:

- Penggunaan HTTP *method* yang tepat untuk memanipulasi *resource*.

- *Request-URI* tidak mengandung *query string* dan hanya terdiri dari *element resource* yang dipisahkan dengan karakter “ / ”. Berikut ini contoh pengaksesan *resource* melalui URI :

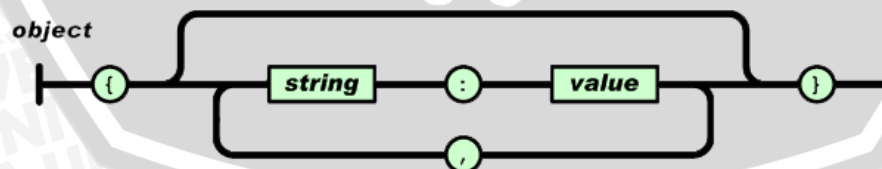
```
https://example.com/customers?lastName=Ghifar&zipcode=40293
```

Jadi pada *REST* tidak ada peraturan *envelope element, header, body, etc* seperti yang ada di *SOAP*, sehingga *REST* dapat dikatakan lebih sederhana dan lebih mudah untuk dikembangkan. Yang harus dipahami dalam konsep *REST* yaitu pada dasarnya setiap URL itu adalah unik yang merupakan representasi dari objek. Memperoleh konten-konten objek tersebut dengan menggunakan *HTTP method* yang sesuai, yaitu dengan menggunakan fungsi *GET, POST, PUT* atau *DELETE*.

2.6 JSON

JSON (Java Script Object Notation) adalah format pertukaran data yang ringan, berbasis teks, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *JSON* pada umumnya digunakan sebagai pertukaran data pada internet. Berkas *JSON* berbasis teks sehingga mudah dikenali oleh berbagai macam bahasa pemrograman sehingga sangat ideal untuk digunakan dalam pertukaran data antar aplikasi yang berbeda bahasa pemrogramannya. Ada 2 macam struktur *JSON* yaitu sebagai *object* pada gambar 2.7 dan sebagai *array* pada gambar 2.8.

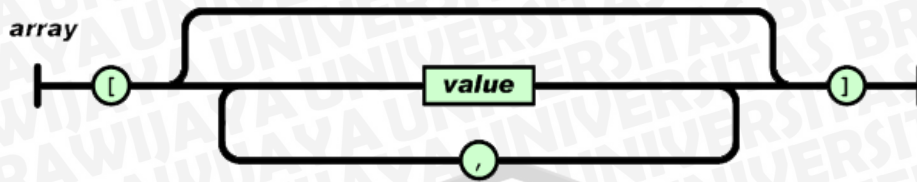
a. Object



Gambar 2.7 Struktur Object JSON

Sumber : [JSO-13]

b. Array

**Gambar 2.8** Struktur Array JSON

Sumber : [JSO-13]

Contoh penggunaan JSON adalah sebagai berikut :

```
{
  "pelatih": {
    "nama": "adi saka",
    "kota": "Langit",
    "murid": [
      {
        "nama": "sila gading",
        "umur": 6
      },
      {
        "nama": "putri tunggal",
        "umur": 8
      },
      {
        "nama": "anka satu",
        "umur": 7
      }
    ]
  }
}
```

Gambar 2.9 Contoh JSON

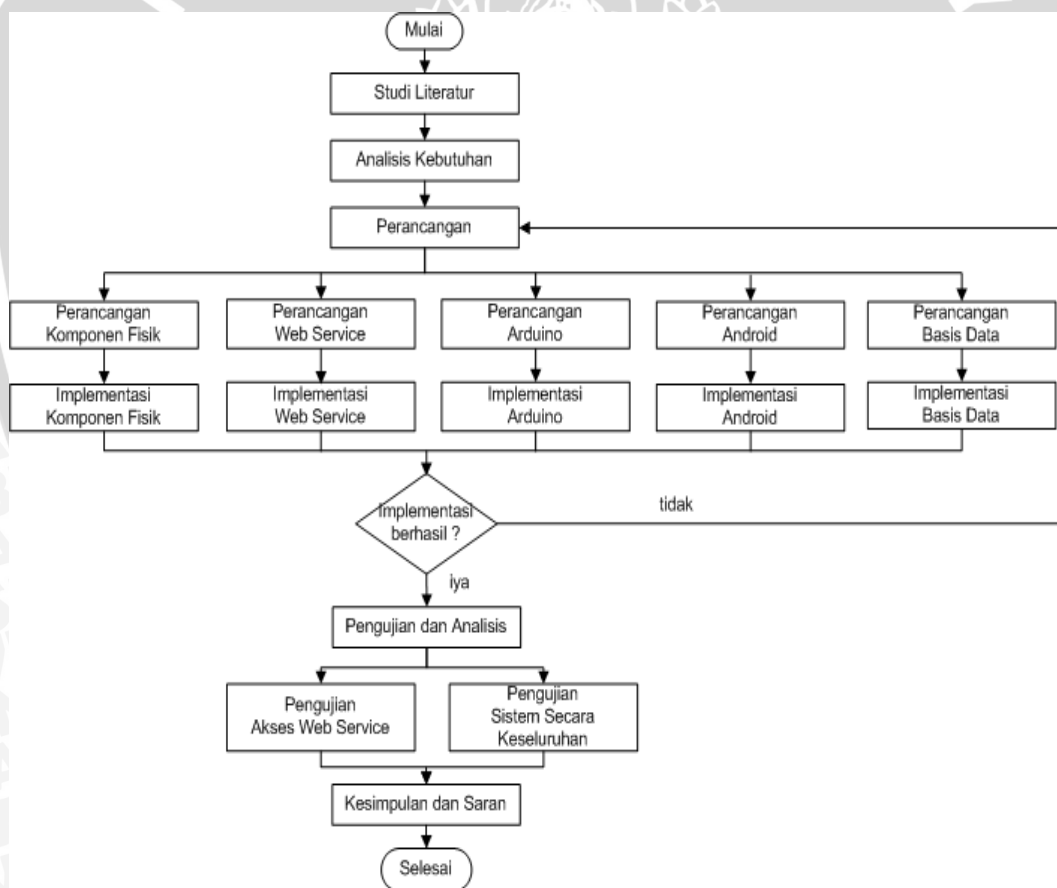
Sumber : [BAY-12]

Dari kode gambar 2.9 di atas, dapat dipahami bahwa data pelatih memiliki properti nama, kota, dan murid. Data pelatih ini merupakan *object*. Hal ini dikarenakan pada penggunaan karakter { }. Kemudian untuk murid, berisikan *array object*, yang dapat ditunjukkan pada penggunaan karakter.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan pada penelitian ini adalah jenis metode *waterfall*. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Pada bab ini akan dijelaskan urutan langkah-langkah yang akan dilakukan dalam pelaksanaan penelitian, yaitu studi literatur, analisa kebutuhan, perancangan, implementasi, pengujian, analisis, serta penutup yang berisikan kesimpulan dan saran. Kesimpulan dan saran disertakan sebagai catatan atas rangkaian sistem ini dan kemungkinan arah pengembangan selanjutnya. Gambar 3.1 dibawah ini merupakan diagram alir dari tahapan penelitian yang dijabarkan secara umum.



Gambar 3.1 Diagram Alir Tahapan Penelitian

Sumber : [Perancangan]

3.1 Studi Literatur

Studi literatur digunakan untuk mempelajari berbagai referensi yang diambil dari buku, jurnal, laporan skripsi, maupun artikel. Referensi yang diperlukan untuk menunjang penulisan laporan ini adalah menjelaskan dasar teori yang digunakan untuk menunjang penulisan laporan. Teori – teori pendukung tersebut yaitu :

1. Kajian pustaka
2. *Smart home*
3. *Microcontroller*
4. Android
5. *Web Service*
6. JSON

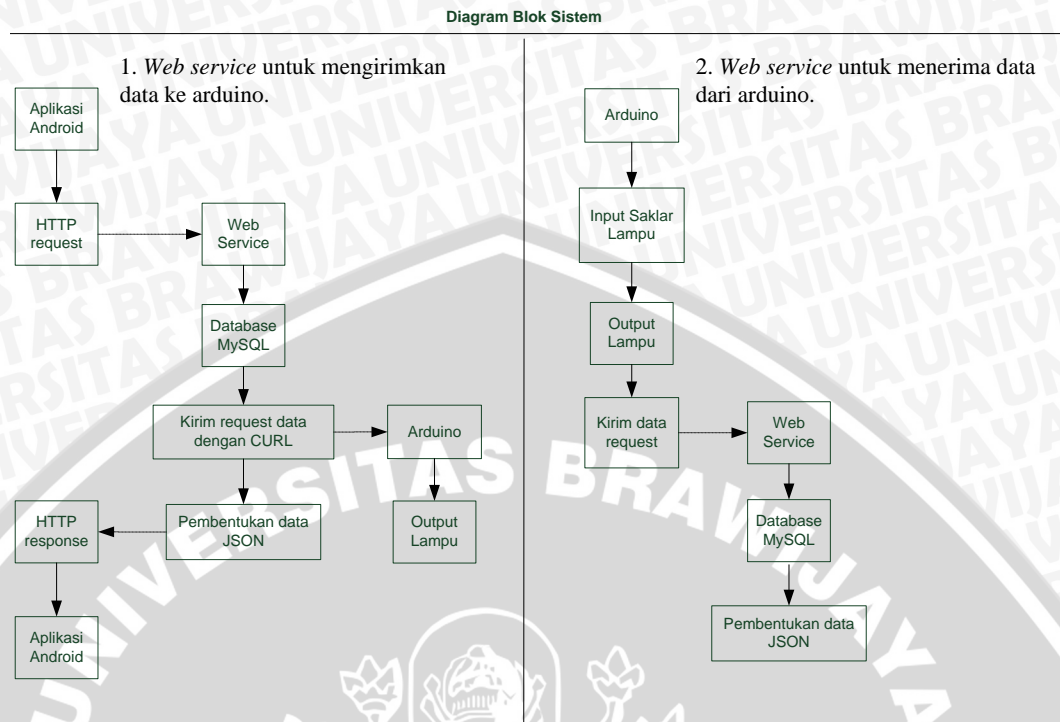
3.2 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan apa saja yang terlibat di dalamnya. Pada analisis kebutuhan ini akan membahas tentang gambaran umum tentang sistem yang akan dibuat, kebutuhan fungsional sistem, analisa perangkat keras, dan analisa perangkat lunak.

3.3 Perancangan

Setelah dilakukan analisis kebutuhan, maka langkah selanjutnya yaitu ke tahap perancangan. Untuk memudahkan pemahaman terhadap perancangan sistem secara keseluruhan maka dapat dijelaskan dalam bentuk diagram blok sistem. Diagram blok sistem pada penelitian ini akan menjelaskan dua prinsip kerja dari *web service* saat menjalin komunikasi data antara android dengan arduino yang berbeda *platform* dan juga bahasa pemrograman, yang akan dijelaskan pada gambar 3.2 sebagai berikut :

1. *Web service* untuk mengirimkan data ke arduino.
2. *Web service* untuk menerima data dari arduino.



Gambar 3.2 Diagram Blok Sistem

Sumber : [Perancangan]

Dari diagram blok tersebut akan dipaparkan lebih dalam lagi dari sistem yang akan digambarkan dalam bentuk flowchart untuk mengetahui tahapan aktivitas proses dari sistem ini. Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Pada tahapan ini akan terbagi dalam 5 tahapan perancangan yaitu perancangan komponen fisik sistem, perancangan *web service*, perancangan program arduino, perancangan aplikasi android dan perancangan basis data.

3.4 Implementasi

Implementasi mengacu kepada tindakan lanjut dari tahap perancangan yang telah dibahas sebelumnya. Pada bagian ini terdapat 5 tahapan implementasi yaitu implementasi komponen fisik, implementasi *web service*, implementasi pemrograman arduino, implementasi aplikasi android dan implementasi basis data. Implementasi *web service* dilakukan dengan menggunakan bahasa pemrograman PHP. Kemudian melakukan implementasi pemrograman arduino

yang mengacu pada perancangan sebelumnya. Tahap selanjutnya adalah akan dilakukan implementasi pada aplikasi android. Dari tahapan implementasi ini akan dibahas penjelasan alur kerja program berdasarkan perancangan yang telah dilakukan.

3.5 Pengujian

Pengujian pada penelitian ini dilakukan agar dapat menunjukkan bahwa sistem telah mampu bekerja sesuai dengan spesifikasi kebutuhan yang melandasinya. Tahap pengujian penelitian ini dengan menguji fungsionalitas dan validasi dari akses *web service* dan sistem secara keseluruhan. Untuk pengujian sistem secara keseluruhan yaitu terdiri dari dua skenario pengujian yaitu skenario pertama ketika memberikan *input* saklar secara manual dan skenario kedua ketika pemberian *input* dari aplikasi android. Pada tahapan ini dijelaskan langkah-langkah pengujian dari sistem yang telah dibuat dan membahas hasil pengujian tersebut. Setelah tahap pengujian, dilakukan analisis untuk mengetahui hasil dari pengujian sistem sehingga dapat didapatkan kesimpulan dari rancangan rangkaian sistem yang telah dibuat.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan sistem lebih lanjut.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis Kebutuhan

Proses analisis kebutuhan ini diawali dengan penjabaran tentang gambaran umum sistem yang akan dibuat. Kemudian akan dilakukan mendeskripsikan analisa kebutuhan fungsional, analisa perangkat keras, dan analisa perangkat lunak yang dibutuhkan.

4.1.1 Gambaran Umum Sistem

Secara garis besar *smart home system* yang memanfaatkan infrastruktur *web service* untuk monitoring dan kontrol lampu berbasis android yang akan dibuat dalam penelitian ini, terbagi menjadi 3 komponen utama pembentuk sistem yaitu aplikasi android, *web service*, dan mikrokontroler arduino.

➤ Aplikasi android

Program aplikasi yang nantinya akan terinstall di *device* android ini digunakan oleh *user* sebagai alat untuk memonitoring dan mengontrol kondisi lampu dari jarak jauh.

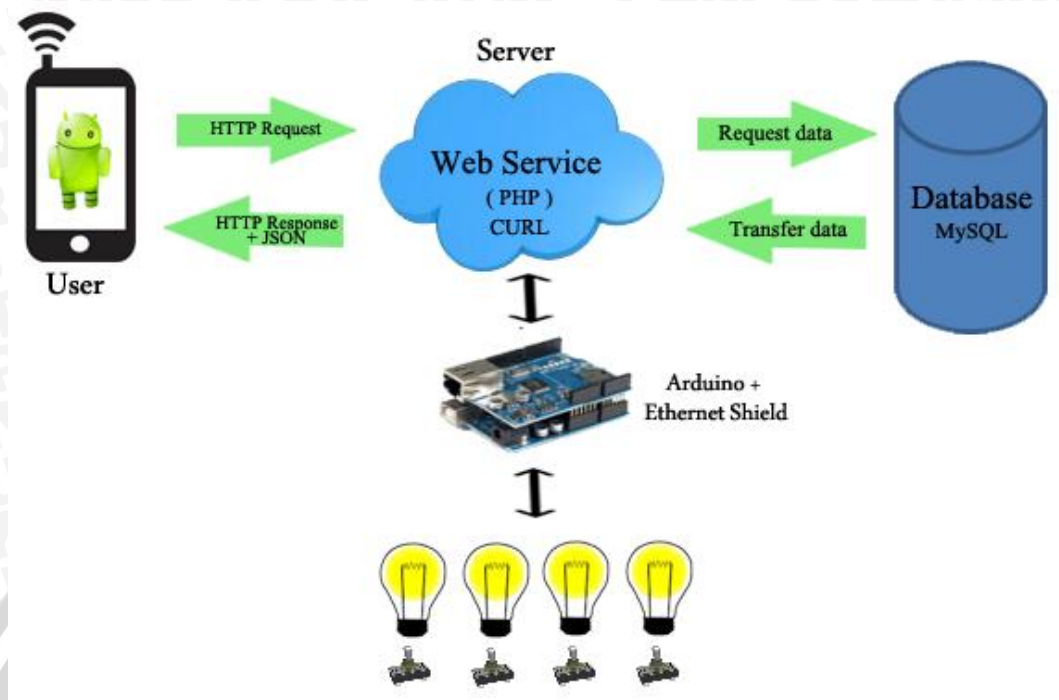
➤ *Web service*

Web service berupa program PHP, berguna sebagai sarana untuk menghubungkan antara android dengan arduino.

➤ Mikrokontroler Arduino

Mikrokontroler arduino akan memproses operasi *input/output* (*digitalRead* dan *digitalWrite*) yaitu terhadap saklar lampu ketika ditekan dan juga kondisi nyala lampu.

Gambaran umum dari sistem akan dijabarkan melalui gambar 4.1 alur kerja sistem berikut ini.



Gambar 4.1 Alur Kerja Sistem

Sumber : [Perancangan]

Alur kerja sistem dapat dideskripsikan sebagai berikut :

1. User akan mencoba membentuk koneksi dengan mikrokontroler dengan mengirimkan pesan *request* ke server.
2. Server akan menerima *request* tadi dengan menggunakan *web service*. Dari *request* yang diterima *web service* akan melakukan *query* ke *database* serta dengan menggunakan library *curl*, data URL akan dikirimkan ke mikrokontroler arduino dengan berbasis metode HTTP GET.
3. Mikrokontroler arduino akan menerima *request* dan akan memproses operasi *digitalWrite* untuk mengendalikan kondisi lampu. Sedangkan untuk operasi *digitalRead* yaitu ketika mendapat *input* dari saklar , kemudian arduino akan mengirimkan *request* kembali ke server.
4. Server akan menerima *request* dari operasi *input* tersebut dan akan melakukan *query* ke *database*. Kemudian server akan mengirimkan pesan *response* dan data JSON ke *user* sesuai dengan isi *database* yang tersimpan.

4.1.2 Kebutuhan Antar Muka

Daftar kebutuhan antar muka perangkat lunak yang harus dipenuhi dapat diperlihatkan oleh tabel 4.1 berikut.

Tabel 4.1 Tabel Kebutuhan Fungsional

No.	Deskripsi Kebutuhan
1.	Aplikasi android harus bisa menampilkan kondisi keadaan lampu dalam bentuk on untuk kondisi menyala dan off untuk kondisi mati.
2.	Aplikasi android harus bisa mengendalikan kondisi lampu sesuai dengan keinginan <i>user</i> menggunakan tombol on dan off
3.	<i>Web service</i> harus bisa menghubungkan aplikasi android dengan mikrokontroler arduino.
4.	Mikrokontroler arduino harus bisa melakukan operasi digital <i>read</i> dan <i>write</i> untuk menyalakan dan mematikan lampu.

4.1.3 Analisa Perangkat Keras dan Perangkat Lunak

Kebutuhan perangkat keras yang akan digunakan dalam pelaksanaan penelitian ini meliputi :

1. Satu unit komputer sebagai server yang menyimpan *web service*.
2. Satu unit hp android.
3. Satu unit arduino sebagai mikrokontroler dan satu unit *ethernet shield*.
4. Komponen elektronik yang meliputi LED, resistor, kabel, dan saklar.

Sedangkan, kebutuhan perangkat lunak yang akan dibutuhkan dalam pelaksanaan penelitian ini meliputi :

1. *Integrated Development Environment (IDE)* yang digunakan untuk membangun aplikasi android adalah Eclipse JUNO, serta telah terinstall *Android Development Tools (ADT)* dan *Android Software Development Kit (SDK)*.
2. Bahasa pemrograman yang digunakan untuk membuat aplikasi android adalah *Java Development Kit (JDK) 1.6* dan *Java Runtime Environment (JRE)*.
3. *Integrated Development Environment* yang digunakan pada untuk pemrograman *microcontroller* adalah *IDE Arduino*

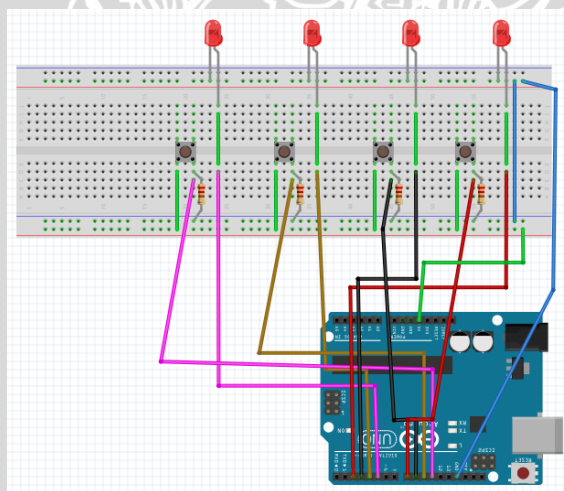
4. XAMPP yang digunakan sebagai aplikasi penyedia layanan *web server*, *database*, dan menggunakan bahasa pemrograman PHP, dijelaskan sebagai berikut :
 - a. *Web server* yang digunakan adalah apache.
 - b. *Database* yang digunakan adalah MySQL.
 - c. Menggunakan pemrograman PHP untuk membuat *web service*.

4.2 Perancangan

Pada tahap perancangan akan terbagi menjadi 4 tahap diantaranya adalah perancangan komponen fisik, perancangan *web service*, perancangan program arduino, perancangan aplikasi android, dan perancangan basis data.

4.2.1 Perancangan Komponen Fisik

Sistem yang akan dirancang pada penelitian ini, akan di implementasikan pada rangkaian komponen fisik yang sesuai dengan gambar di bawah ini :

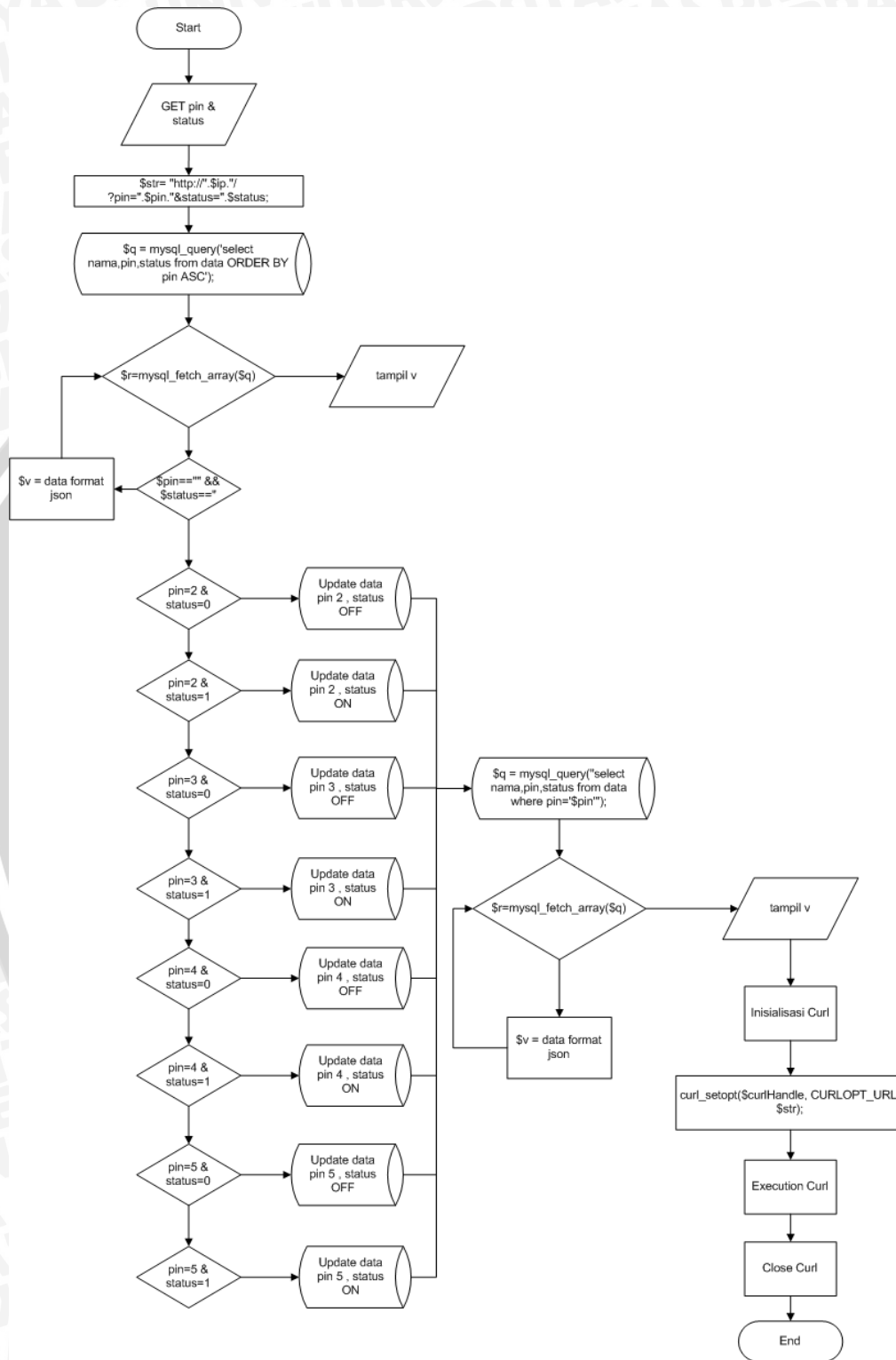


Gambar 4.2 Rangkaian Komponen Fisik

Sumber : [Perancangan]

Gambar 4.2 rangkaian fisik diatas terdiri dari 4 buah LED dengan masing-masing saklar. Kemudian akan terhubung ke masing-masing pin yang ada di *board* arduino beserta *ethernet shield* nya sesuai dengan rancangan diatas.

4.2.2 Perancangan Program pada Web Service



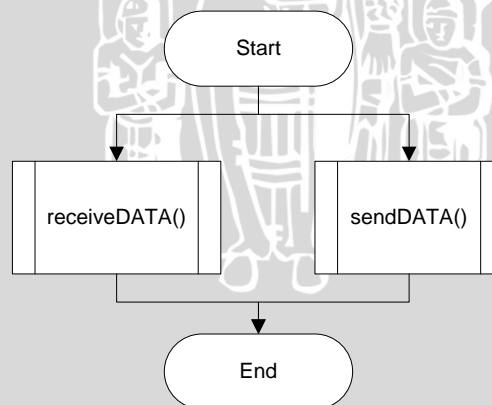
Gambar 4.3 Diagram Alir Program pada Web Service

Sumber : [Perancangan]

Web service harus dapat menjalankan tugas utamanya untuk menjalin komunikasi antara android dengan arduino. Oleh karena itu *web service* harus dapat menjalankan fungsi untuk menerima data dari arduino dan mengirimkan data ke arduino. Kemudian aplikasi android akan mengambil data dari *web service* ini dengan format data JSON.

Pada gambar 4.3 proses diagram alir dari algoritma program pada *web service* diatas dapat dilihat bahwa *web service* menggunakan metode GET untuk nilai pin dan status dari lampu LED. Apabila mendapatkan *request* data GET `pin=""&status=""` program akan menampilkan semua data yang ada di dalam *database* ke dalam format data JSON. Kemudian jika mendapatkan *request* data dari nilai masing-masing pin dan status dari lampu misalkan saja mendapatkan *request* GET `pin=2&status=1` program akan melakukan *query update* data pada *database* dan menampilkan data dengan nilai itu saja yang ada dalam *database* ke dalam format data JSON serta akan mengirimkan *request* data tersebut ke arduino dengan menggunakan *library curl* sehingga akan terbentuk komunikasi dengan arduino.

4.2.3 Perancangan Program pada Arduino

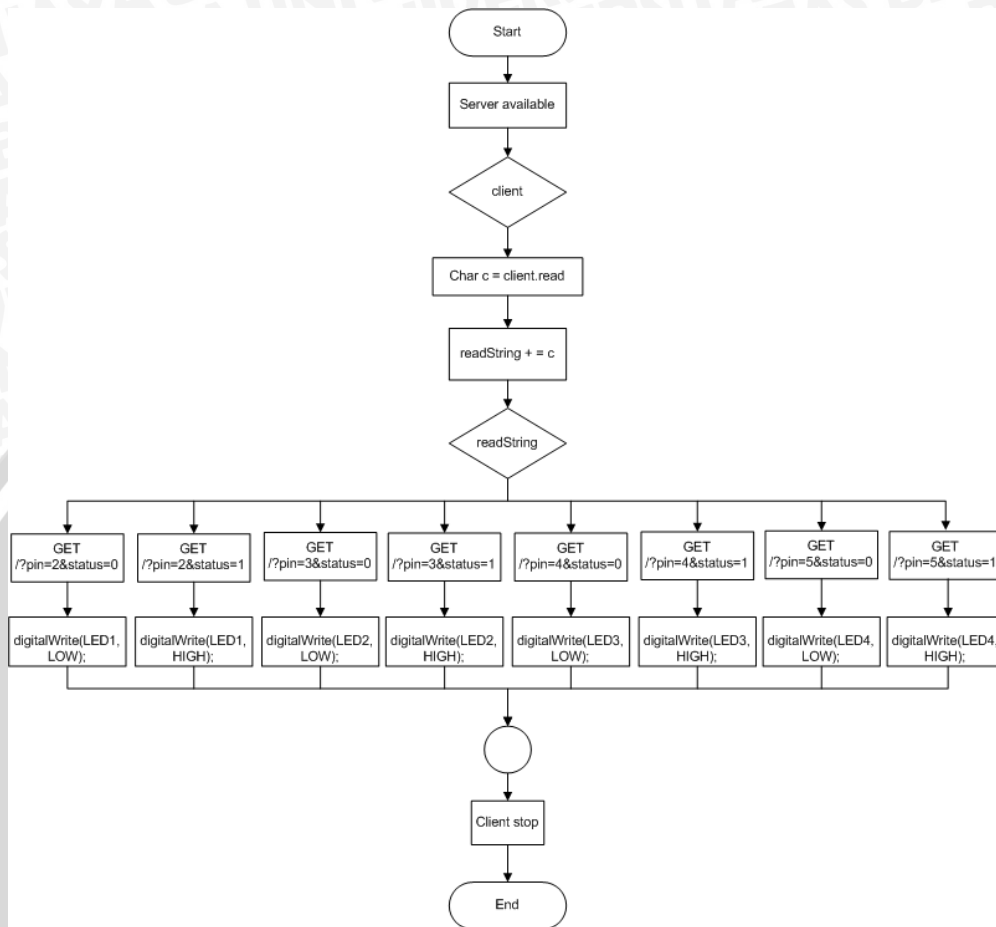


Gambar 4.4 Diagram Alir Program pada Arduino

Sumber : [Perancangan]

Pada gambar 4.4 menunjukkan perancangan program pada Arduino terdiri dari dua fungsi yaitu :

1. Menerima *request* data dari *web service* untuk menampilkan keadaan lampu yang diinginkan oleh user sesuai dengan *button* yang ada di aplikasi android.

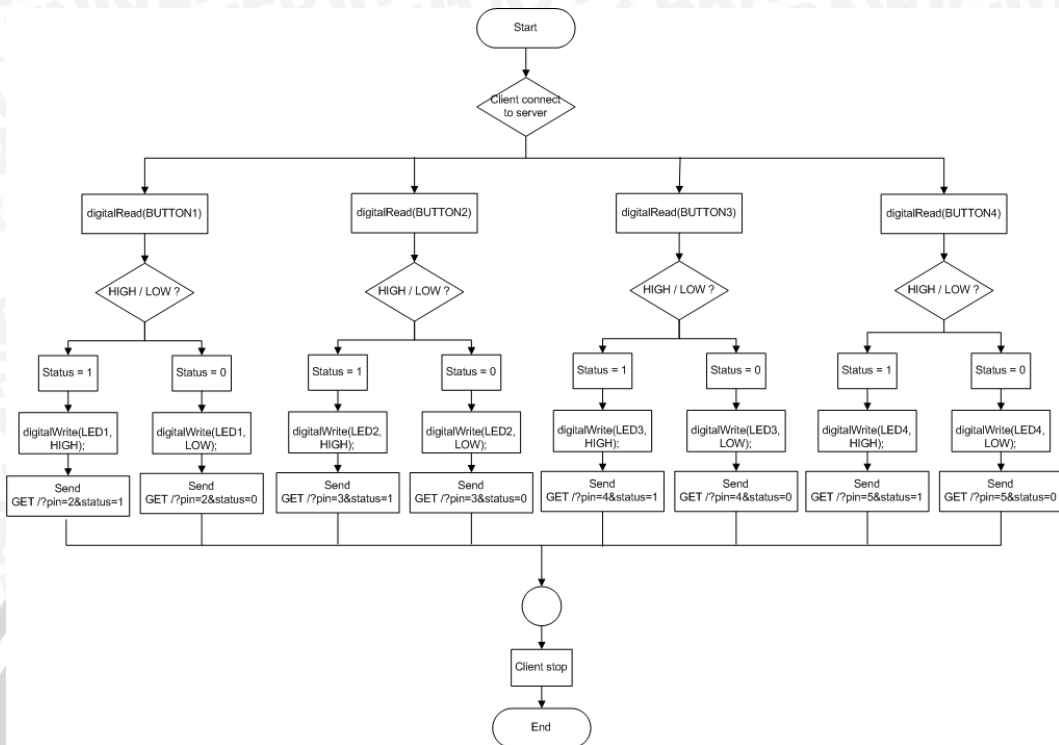


Gambar 4.5 Diagram Alir Proses Menerima *Request* Data pada Arduino

Sumber : [Perancangan]

Pada gambar 4.5 proses diagram alir dari algoritma program pada arduino diatas dapat dilihat bahwa apabila ada *request* data dari *web service*, kemudian akan dibaca per karakter dan disimpan ke dalam variabel *readString*. Kemudian lampu yang akan ditampilkan sesuai dengan nilai pin dan status yang diperoleh dengan menggunakan metode GET.

2. Mengirimkan *request* data ke *web service* saat perangkat ini mendapatkan *input* dari saklar untuk menjalankan lampu secara manual.



Gambar 4.6 Diagram Alir Proses Mengirimkan *Request* Data pada Arduino

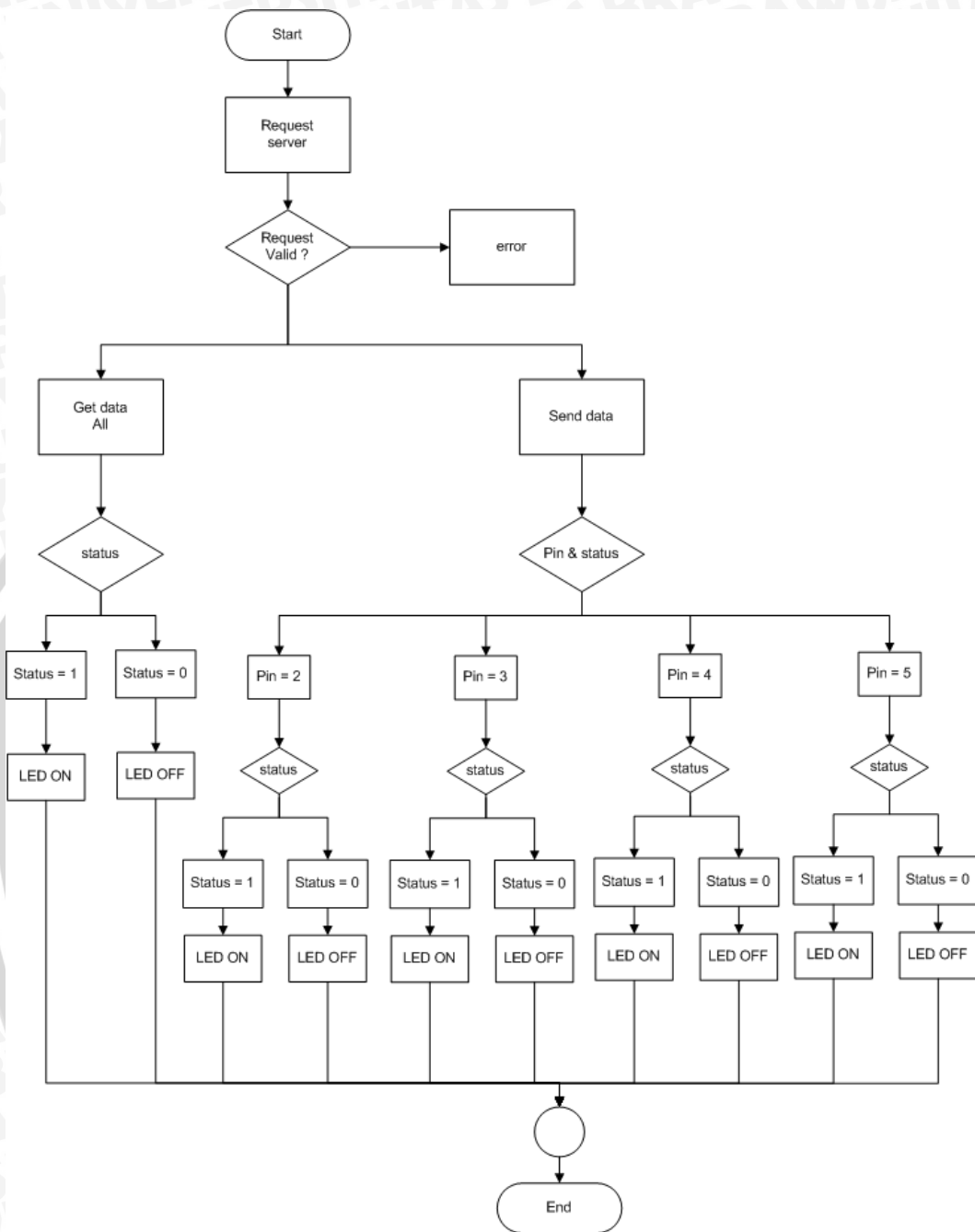
Sumber : [Perancangan]

Pada gambar 4.6 proses diagram alir dari algoritma program pada arduino diatas dapat dilihat bahwa sistem akan mencoba melakukan koneksi ke server. Akan menjalankan operasi *digitalRead* ketika saklar pada komponen fisik ditekan. Bila kondisi on maka operasi *digitalRead* akan mengembalikan nilai *high* sehingga akan menyalakan lampu serta akan mengirimkan *request* data ke *web service* sesuai dengan pin dan statusnya adalah 1. Kemudian jika kondisinya off maka operasi *digitalRead* akan mengembalikan nilai *low* sehingga akan mematikan lampu serta akan mengirimkan *request* data ke *web service* sesuai dengan pin dan statusnya adalah 0.

4.2.4 Perancangan pada Android

4.2.4.1 Perancangan Program

Dibawah ini akan digambarkan diagram alir dari perancangan program di android.



Gambar 4.7 Diagram Alir Program pada Android

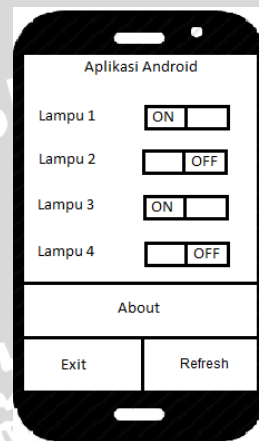
Sumber : [Perancangan]

Pada gambar 4.7 proses diagram alir dari algoritma program pada android diatas dapat dilihat bahwa sistem akan melakukan *request* ke server dengan alamat url yang sudah ditentukan. Ketika telah terhubung maka ada dua kemungkinan yang dapat dilakukan yaitu mengambil semua data JSON yang ada

di *web service* dan mengirimkan data serta mengambil data berdasarkan nilai pin dan status.

4.2.4.2 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan awal yang akan digunakan dalam implementasi antarmuka pengguna pada aplikasi android. Perancangan tampilan utama aplikasi ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Rancangan Antar Muka Tampilan Utama Aplikasi

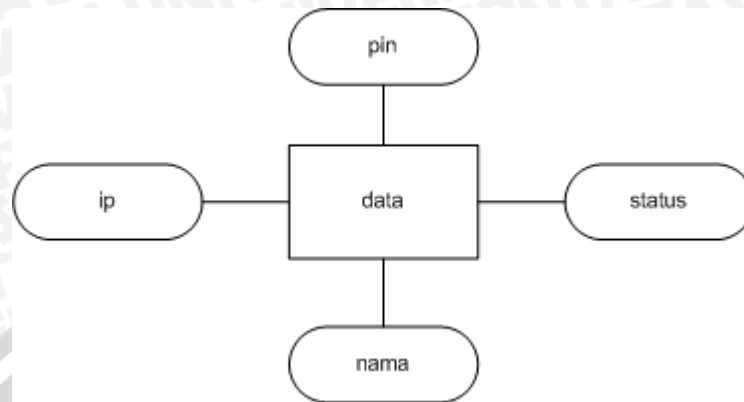
Sumber : [Perancangan]

Setelah terjalin koneksi ke *web service*, aplikasi akan menuju ke tampilan utama yang akan menampilkan kondisi keadaan lampu, sehingga pengguna dapat memonitoring keadaan lampu dan pengguna dapat mengendalikan kondisi lampu dengan tombol on / off yang telah tersedia.

4.2.5 Perancangan Basis Data

Basis data berfungsi sebagai tempat menyimpan data. Pada penelitian ini perancangan basis data direpresentasikan dalam bentuk *Entity Relationship Diagram* (ERD). ERD menunjukkan hubungan yang terjadi diantara objek (entitas) yang terlibat dalam suatu *database*. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan

yang nyata. Pada perancangan basis data sistem ini terdapat satu buah tabel yaitu tabel data. ERD sistem ini dapat dilihat dalam Gambar 4.9.



Gambar 4.9 Entity Relational Diagram

Sumber : [Perancangan]

Pada tabel 4.2 merupakan struktur tabel serta keterangan masing-masing *table* dan *field* yang ada pada *database*.

Tabel Data	
Kolom	Tipe Data
Ip	Varchar
Pin	Int
Status	Int
Nama	Varchar

Tabel 4.2 Rancangan tabel untuk menyimpan data

Sumber : [Perancangan]

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas mengenai tahapan implementasi dan pengujian sistem berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan aplikasi.

5.1 Implementasi Sistem

Pembahasan terdiri dari lingkungan implementasi sistem, implementasi komponen fisik, implementasi program pada *web service*, implementasi program pada arduino, implementasi program pada aplikasi android dan implementasi basis data.

5.1.1 Lingkungan Implementasi Sistem

Implementasi program pada *web service* dibuat dengan menggunakan bahasa skrip PHP, program yang berjalan pada arduino dibuat dengan menggunakan *Arduino Programming Language (APL)* yang berbasis pada bahasa pemrograman C++, dan implementasi pada aplikasi android dengan menggunakan *software eclipse*.

1. Perangkat Keras

a. Komputer

Spesifikasi :

- *Processor* : Intel Core 2 Duo T5750 – 2.00 GHz
- *Memory* : 2560 MB

b. Arduino

- *Model* : Arduino UNO R3
- *Mikrokontroler* : Atmega328
- *Clock Speed* : 16MHz
- *Memory* : 2KB SRAM
- *Digital I/O Pin* : 14 *Digital I/O Pin* dan memiliki 6 Pin yang dapat digunakan sebagai *output PWM*
- *Analog Input* : 6 *Analog Input Pin*

- *Ethernet shield* : WIZNET W5100
2. Perangkat Lunak
 - a. Server
 - Sistem Operasi : Windows 7 Ultimate
 - *Web Server* : Apache 2.2.14
 - PHP : PHP 5.3.1
 - *Database* : MySQL
 - b. *Software* pembuat aplikasi android : Eclipse
 - c. *Software* pembuat perancangan komponen fisik : Fritzing
 - d. *Software* pembuat diagram model perancangan : Microsoft Visio
 - e. *Software* alat pendukung basis data : XAMPP

5.1.2 Batasan – Batasan Implementasi

Batasan - batasan dalam mengimplementasikan sistem *smart home system* ini adalah sebagai berikut :

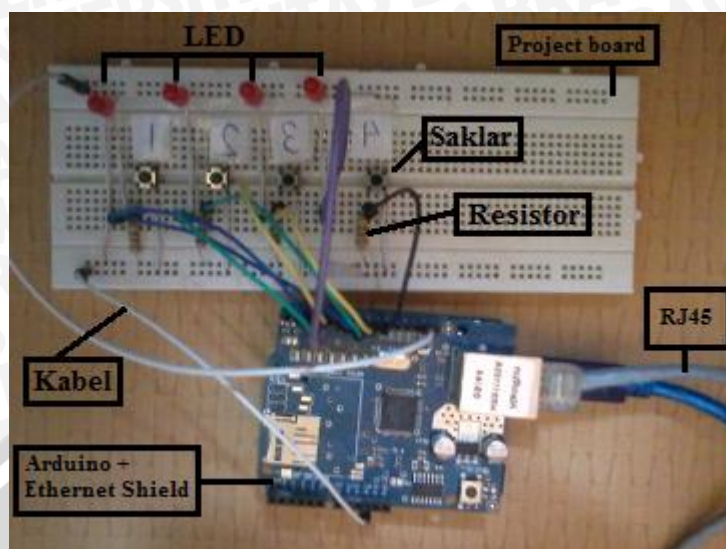
1. Pengendalian monitoring dan kontrol yang dilakukan di aplikasi android berupa tombol on/off.
2. Komponen elektronik yang digunakan untuk mensimulasikan lampu yaitu berupa LED.
3. Sistem ini di simulasikan pada topologi jaringan *ad hoc* dimana sebuah komputer berfungsi sebagai server yang akan menyimpan *web service* beserta *database* nya.
4. Penelitian ini belum memperhatikan aspek keamanan sistem.

5.1.3 Implementasi Komponen Fisik

Berdasarkan perancangan komponen fisik yang ditunjukkan pada bab analisis dan perancangan, maka proses implementasi dari komponen fisik tersebut dapat dijelaskan pada langkah-langkah berikut ini.

1. Alat dan bahan
 - a. Menyiapkan *board* Arduino Uno.
 - b. Menyiapkan *ethernet shield*.

- c. Menyiapkan lampu 4 buah LED warna merah.
 - d. Menyiapkan 4 buah saklar.
 - e. Menyiapkan 4 buah resistor.
 - f. Menyiapkan kabel *jumper* secukupnya.
 - g. Menyiapkan *project board*.
 - h. Menyiapkan kabel *ethernet* RJ45.
2. Langkah-langkah penyusunan
- a. Pasang *board* Arduino dengan *ethernet shield* yang menggunakan pin *digital* 0,1,10,11,12,13.
 - b. Rangkai semua LED, saklar, dan resistor pada *project board* sesuai dengan gambar 4.2 yang ada pada bab analisis dan perancangan.
 - c. Pin digital 2 pada Arduino akan terhubung dengan pin anode (+) pada LED 1. Katode (-) akan terhubung dengan pin GND pada Arduino.
 - d. Pin digital 3 pada Arduino akan terhubung dengan pin anode (+) pada LED 2. Katode (-) akan terhubung dengan pin GND pada Arduino.
 - e. Pin digital 4 pada Arduino akan terhubung dengan pin anode (+) pada LED 3. Katode (-) akan terhubung dengan pin GND pada Arduino.
 - f. Pin digital 5 pada Arduino akan terhubung dengan pin anode (+) pada LED 4. Katode (-) akan terhubung dengan pin GND pada Arduino.
 - g. Saklar untuk LED 1 akan terhubung dengan resistor dengan pin digital 6 , pin 5v, dan pin GND.
 - h. Saklar untuk LED 2 akan terhubung dengan resistor dengan pin digital 7 , pin 5v, dan pin GND.
 - i. Saklar untuk LED 3 akan terhubung dengan resistor dengan pin digital 8 , pin 5v, dan pin GND.
 - j. Saklar untuk LED 4 akan terhubung dengan resistor dengan pin digital 9 , pin 5v, dan pin GND.
 - k. Hasil rangkaian komponen fisik setelah dirangkai akan ditunjukkan pada gambar 5.1.



Gambar 5.1 Hasil Implementasi Rangkaian Komponen Fisik

Sumber : [Implementasi]

5.1.4 Implementasi Program pada Web Service

Pada implementasi *web service* harus dapat menjalankan dua *syntax* program sebagai berikut :

1. Untuk menerima data dari arduino

Web service akan menerima *request* data dari arduino yang menggunakan metode GET dengan nilai parameter data dari pin dan status, kemudian melakukan *query* yang akan disimpan ke dalam *database* dan akan diterjemahkan ke dalam tampilan format JSON. Syntax di bawah ini :

```

1. $status = $_GET['status'];
2. $pin = $_GET['pin'];
3.
4. $q = mysql_query('select nama,pin,status from data ORDER BY
5. pin ASC');
6. $num_row = mysql_num_rows($q);
7. $jml_data=1;
8.
9. $v = '{"data" : [';
10. while($r=mysql_fetch_array($q))
11. {
12. if($pin==" " && $status=="")

```

```
13.     {
14.         if($jml_data==$num_row){
15.             $v .= '{"nama" : "'. $r['nama'].'"', "pin" : "'. $r['pin'].'"',
16.             "status" : "'. $r['status'].'"' };
17.         }else{
18.             $v .= '{"nama" : "'. $r['nama'].'"', "pin" : "'. $r['pin'].'"',
19.             "status" : "'. $r['status'].'"' },';
20.         }
21.         $jml_data++;
22.     }
23. else{
24.
25.     if($pin==2 && $status == 0){
26.         $query = mysql_query("update data set nama='Lampu Mati',
27.         status='$status' where pin='$pin'");
28.     }
29.     elseif ($pin==2 && $status == 1){
30.         $query = mysql_query("update data set nama='Lampu Nyala',
31.         status='$status' where pin='$pin'");
32.     }
33.     elseif ($pin==3 && $status == 0){
34.         $query = mysql_query("update data set nama='Lampu Mati',
35.         status='$status' where pin='$pin'");
36.     }
37.     elseif ($pin==3 && $status == 1){
38.         $query = mysql_query("update data set nama='Lampu Nyala',
39.         status='$status' where pin='$pin'");
40.     }
41.     elseif ($pin==4 && $status == 0){
42.         $query = mysql_query("update data set nama='Lampu Mati',
43.         status='$status' where pin='$pin'");
44.     }
45.     elseif ($pin==4 && $status == 1){
46.         $query = mysql_query("update data set nama='Lampu Nyala',
47.         status='$status' where pin='$pin'");
48.     }
49.     elseif ($pin==5 && $status == 0){
50.         $query = mysql_query("update data set nama='Lampu Mati',
```



```

51. status='$status' where pin='$pin');
52. }
53. elseif ($pin==5 && $status == 1){
54. $query = mysql_query("update data set nama='Lampu Nyalala',
55. status='$status' where pin='$pin');
56. }
57.
58.
59.
60. $q = mysql_query("select nama,pin,status from data where
61. pin='$pin');
62.
63. while($r=mysql_fetch_array($q)){
64. $v .= '{"nama" : "'. $r['nama'].'", "pin" : "'. $pin.'",
65. "status" : "'. $status.'"}';
66.
67. }
68.
69. }
70.
71. }
72. $v .= ' ] }';
73.
74. echo $v;

```

Gambar 5.2 Web Service Menerima Data dari Arduino

Sumber : [Implementasi]

Penjelasan syntax penerimaan data dari arduino pada gambar 5.2 adalah :

- 1) Pada baris 1-2 inialisasi metode GET untuk nilai pin dan status dari masing-masing LED.
- 2) Pada baris 4-7 inialisasi variabel yang dibutuhkan untuk query, dan variabel untuk membantu menampilkan data format JSON.
- 3) Pada baris 12-22 untuk menampilkan data JSON semua data pin dan status yang tersimpan di dalam *database*.
- 4) Pada baris 25-27 jika mendapatkan *request* GET data pin=2 dan status=0, akan mengupdate data di *database* untuk LED 1 lampu dalam keadaan mati,

kemudian akan ditampilkan dalam format JSON untuk pin 2 dengan status 0 pada baris 60-70.

- 5) Pada baris 29-31 jika mendapatkan *request* GET data pin=2 dan status=1, akan mengupdate data di *database* untuk LED 1 lampu dalam keadaan menyala, kemudian akan ditampilkan dalam format JSON untuk pin 2 dengan status 1 pada baris 60-71.
- 6) Pada baris 33-35 jika mendapatkan *request* GET data pin=3 dan status=0, akan mengupdate data di *database* untuk LED 2 lampu dalam keadaan mati, kemudian akan ditampilkan dalam format JSON untuk pin 3 dengan status 0 pada baris 60-71.
- 7) Pada baris 37-39 jika mendapatkan *request* GET data pin=3 dan status=1, akan mengupdate data di *database* untuk LED 2 lampu dalam keadaan menyala, kemudian akan ditampilkan dalam format JSON untuk pin 3 dengan status 1 pada baris 60-71.
- 8) Pada baris 41-43 jika mendapatkan *request* GET data pin=4 dan status=0, akan mengupdate data di *database* untuk LED 3 lampu dalam keadaan mati, kemudian akan ditampilkan dalam format JSON untuk pin 4 dengan status 0 pada baris 60-71.
- 9) Pada baris 45-47 jika mendapatkan *request* GET data pin=4 dan status=1, akan mengupdate data di *database* untuk LED 3 lampu dalam keadaan menyala, kemudian akan ditampilkan dalam format JSON untuk pin 4 dengan status 1 pada baris 60-71.
- 10) Pada baris 49-51 jika mendapatkan *request* GET data pin=5 dan status=0, akan mengupdate data di *database* untuk LED 4 lampu dalam keadaan mati, kemudian akan ditampilkan dalam format JSON untuk pin 5 dengan status 0 pada baris 60-71.
- 11) Pada baris 53-55 jika mendapatkan *request* GET data pin=5 dan status=1, akan mengupdate data di *database* untuk LED 4 lampu dalam keadaan menyala, kemudian akan ditampilkan dalam format JSON untuk pin 5 dengan status 1 pada baris 60-71.
- 12) Pada baris 74 mencetak data JSON.

2. Untuk mengirimkan data ke arduino.

Web service akan menerima *request* data dari android yang menggunakan metode GET dengan nilai parameter data dari pin dan status, kemudian akan melakukan query yang akan disimpan ke dalam *database* dan akan diterjemahkan ke dalam tampilan format JSON. Kemudian pada proses pengiriman data ke arduino, penulis memanfaatkan *library* curl yaitu dengan menggunakan syntax sebagai berikut

1.	<code>\$str= "http://".\$ip."/?pin=".\$pin."&status=".\$status;</code>
2.	<code>\$curlHandle = curl_init();</code>
3.	<code>curl_setopt(\$curlHandle, CURLOPT_URL, \$str);</code>
4.	<code>curl_exec(\$curlHandle);</code>
5.	<code>curl_close(\$curlHandle);</code>

Gambar 5.3 *Web Service* Mengirimkan Data ke Arduino

Sumber : [Implementasi]

Penjelasan syntax pengiriman data ke arduino dengan menggunakan bantuan *library* curl pada gambar 5.3 adalah :

- 1) Pada baris 1 merupakan inisialisasi variabel \$str yang berupa nilai IP, pin dan status yang nantinya akan dikirimkan.
- 2) Pada baris 2 inisialisasi *library* curl.
- 3) Pada baris 3-4 mensetting *options library* curl dengan akan mengirimkan data ke alamat ip 192.168.1.5 (ip arduino) dan isi datanya berada pada variabel \$str yaitu berupa nilai pin dan status
- 4) Pada baris 5 eksekusi *library* curl.
- 5) Pada baris 6 akan menutup *library* curl.

5.1.5 Implementasi Program pada Arduino

Tahap pertama pada implementasi program pada arduino yaitu diperlukan konfigurasi konektifitas jaringan.

1.	<code>Byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // mac</code>
2.	<code>byte ip[] = { 192, 168, 1, 5 }; // ip in lan</code>
3.	<code>byte gateway[] = { 192, 168, 1, 1 }; // ip gateway</code>

4.	byte subnet[] = { 255, 255, 255, 0 }; //subnet mask
5.	byte myserver[] = { 192, 168, 1, 2 }; //ip server

Gambar 5.4 Konfigurasi Jaringan Arduino

Sumber : [Implementasi]

Penjelasan konfigurasi konektivitas jaringan di arduino pada gambar 5.4 adalah :

- 1) Pada baris 1 *mac address* arduino yaitu 0xDE 0xAD 0xBE 0xEF 0xFE 0xED.
- 2) Pada baris 2 *ip address* arduino yaitu 192.168.1.5
- 3) Pada baris 3 *ip gateway* yaitu 192.168.1.1
- 4) Pada baris 4 *ip subnet* yaitu 255.255.255.0
- 5) Pada baris 5 *ip server* yang menyimpan *web service* yaitu 192.168.1.2

Kemudian program pada arduino harus dapat menjalankan dua fungsi sebagai berikut :

1. Menerima *request* data

Ketika koneksi telah tersedia *board* arduino akan membaca *request* data berdasarkan parameter pin dan status yang dikirimkan oleh *web service* yang menggunakan metode HTTP GET.

```

1. void receiveDATA() {
2.     EthernetClient client = server.available();
3.     if (client) {
4.         while (client.connected()) {
5.             if (client.available()) {
6.                 char c = client.read();
7.
8.                 //membaca karakter HTTP request
9.                 if (readString.length() < 100) {
10.
11.                     //simpan karakter ke dalam string
12.                     readString = readString + c;
13.                 }
14.
15.                 //jika HTTP request berakhir
16.                 if(c=='\n'){
17.                     client.println("HTTP/1.1 200 OK");

```

```
18. client.println("Content-Type:text/html");
19. client.println();
20.
21. //PIN=2 untuk LED 1
22. if(readString.startsWith("GET /?pin=2&status=1")){
23.     digitalWrite(LED1,HIGH);
24.     statel = 1;
25. }
26.
27. if(readString.startsWith("GET /?pin=2&status=0")){
28.     digitalWrite(LED1,LOW);
29.     statel = 0;
30. }
31.
32. //PIN=3 untuk LED 2
33. if(readString.startsWith("GET /?pin=3&status=1")){
34.     digitalWrite(LED2,HIGH);
35.     state2 = 1;
36. }
37.
38. if(readString.startsWith("GET /?pin=3&status=0")){
39.     digitalWrite(LED2,LOW);
40.     state2 = 0;
41. }
42.
43. //PIN=4 untuk LED 3
44. if(readString.startsWith("GET /?pin=4&status=1")){
45.     digitalWrite(LED3,HIGH);
46.     state3 = 1;
47. }
48.
49. if(readString.startsWith("GET /?pin=4&status=0")){
50.     digitalWrite(LED3,LOW);
51.     state3 = 0;
52. }
53.
54. //PIN=5 untuk LED 4
55. if(readString.startsWith("GET /?pin=5&status=1")){
```

```
56.         digitalWrite(LED4,HIGH);
57.         state4 = 1;
58.     }
59.
60.         if(readString.startsWith("GET /?pin=5&status=0")){
61.             digitalWrite(LED4,LOW);
62.             state4 = 0;
63.         }
64.         readString="";
65.         client.stop();
66.     }
67. }
68. }
69. }
70. }
```

Gambar 5.5 Fungsi Menerima *Request* Data pada Arduino

Sumber : [Implementasi]

Penjelasan proses fungsi menerima *request* data pada gambar 5.5 adalah :

- 1) Pada baris 2 membuat koneksi agar tersedia.
- 2) Pada baris 3-7 membaca per-karakter dan disimpan pada variabel `readString` dengan panjang karakter kurang dari 100.
- 3) Pada baris 16-19 jika proses request yang dikirimkan telah berakhir.
- 4) Pada baris 22-25 dapat menjalankan kondisi yaitu jika `readString = /?pin=2&status=1` maka LED1 akan menyala.
- 5) Pada baris 27-30 dapat menjalankan kondisi yaitu jika `readString = /?pin=2&status=0` maka LED1 akan mati.
- 6) Pada baris 33-36 dapat menjalankan kondisi yaitu jika `readString = /?pin=3&status=1` maka LED2 akan menyala.
- 7) Pada baris 38-41 dapat menjalankan kondisi yaitu jika `readString = /?pin=3&status=0` maka LED2 akan mati.
- 8) Pada baris 44-47 dapat menjalankan kondisi yaitu jika `readString = /?pin=4&status=1` maka LED3 akan menyala.

- 9) Pada baris 49-52 dapat menjalankan kondisi yaitu jika `readString = /?pin=4&status=0` maka LED3 akan mati.
- 10) Pada baris 55-58 dapat menjalankan kondisi yaitu jika `readString = /?pin=5&status=1` maka LED4 akan menyala.
- 11) Pada baris 60-63 dapat menjalankan kondisi yaitu jika `readString = /?pin=5&status=0` maka LED4 akan mati.

2. Mengirimkan *request* data

Ketika telah terbentuk koneksi ke server, *board* arduino akan mengirimkan *request* data menggunakan metode HTTP GET berdasarkan *input* dari parameter pin dan status yang diterima oleh *board* arduino.

```
1. void sendData() //fungsi untuk mengirimkan GET request data
2. {
3.     //LED1
4.     if (client.connect(myserver,80)) {
5.         val1 = digitalRead(BUTTON1);
6.         // mengecek perubahan value untuk LED A
7.         if ((val1 == HIGH) && (old_val1 == LOW)){
8.             state1 = 1 - state1;
9.         }
10.        old_val1 = val1;
11.        if (state1 == 1) {
12.            digitalWrite(LED1, HIGH); // nyalakan LED
13.            client.println("GET
14. /skripsi/receiveData.php?pin=2&status=1");
15.        }
16.        else {
17.            digitalWrite(LED1, LOW);
18.            client.println("GET
19. /skripsi/receiveData.php?pin=2&status=0");
20.        }
21.        client.stop();
22.    }
23.
24.    //LED2
25.    if (client.connect(myserver,80)) {
```

```
26.
27.     val2 = digitalRead(BUTTON2);
28.     // mengecek perubahan value untuk LED A
29.     if ((val2 == HIGH) && (old_val2 == LOW)){
30.         state2 = 1 - state2;
31.     }
32.     old_val2 = val2;
33.     if (state2 == 1) {
34.         digitalWrite(LED2, HIGH); // nyalakan LED
35.         client.println("GET
36. /skripsi/receiveData.php?pin=3&status=1");
37.     }
38.     else {
39.         digitalWrite(LED2, LOW);
40.         client.println("GET
41. /skripsi/receiveData.php?pin=3&status=0");
42.     }
43.     client.stop();
44. }
45.
46. //LED3
47. if (client.connect(myserver,80)) {
48.
49.     val3 = digitalRead(BUTTON3);
50.     // mengecek perubahan value untuk LED A
51.     if ((val3 == HIGH) && (old_val3 == LOW)){
52.         state3 = 1 - state3;
53.     }
54.     old_val3 = val3;
55.     if (state3 == 1) {
56.         digitalWrite(LED3, HIGH); // nyalakan LED
57.         client.println("GET
58. /skripsi/receiveData.php?pin=4&status=1");
59.     }
60.     else {
61.         digitalWrite(LED3, LOW);
62.         client.println("GET
63. /skripsi/receiveData.php?pin=4&status=0");
```



```
64.     }
65.     client.stop();
66. }
67.
68. //LED4
69. if (client.connect(myserver,80)) {
70.
71.     val4 = digitalRead(BUTTON4);
72.     // mengecek perubahan value untuk LED A
73.     if ((val4 == HIGH) && (old_val4 == LOW)){
74.         state4 = 1 - state4;
75.     }
76.     old_val4 = val4;
77.     if (state4 == 1) {
78.         digitalWrite(LED4, HIGH); // nyalakan LED
79.         client.println("GET
80. /skripsi/receiveData.php?pin=5&status=1");
81.     }
82.     else {
83.         digitalWrite(LED4, LOW);
84.         client.println("GET
85. /skripsi/receiveData.php?pin=5&status=0");
86.     }
87.     client.stop();
88. }
89. }
90. }
```

Gambar 5.6 Fungsi Mengirimkan *Request* Data Pada Arduino

Sumber : [Implementasi]

Penjelasan proses fungsi mengirimkan *request* data pada gambar 5.6 adalah :

- 1) Pada baris 4 akan memeriksa koneksi ke server yang menyimpan layanan *web service*.
- 2) Pada baris 5 membaca nilai *input* dari LED1 yaitu pada pin = 2 untuk saklar 1 dan menyimpannya ke dalam variabel val 1.

- 3) Pada baris 7-10 memeriksa perubahan *value* untuk LED 1 , jika *input* bernilai HIGH maupun bernilai LOW, kemudian akan disimpan ke dalam sebuah variabel.
- 4) Pada baris 11-15 akan memeriksa status jika bernilai = 1 akan menyalakan LED1 serta mengirimkan *request* data ke server dengan menggunakan metode GET /skripsi/receiveData.php?pin=2&status=1.
- 5) Pada baris 16 – 20 jika status bernilai = 0 maka lampu LED1 akan mati serta mengirimkan *request* data ke server dengan menggunakan metode GET /skripsi/receiveData.php?pin=2&status=0.
- 6) Pada baris 21 proses untuk LED1 akan berhenti dikerjakan.
- 7) Pada baris 24 – 87 , proses nya sama dengan point 2 hingga 6, yang membedakan adalah nomor pinnya saja yaitu untuk LED2 pin = 3, LED3 pin = 4, dan LED4 pin = 5.

5.1.6 Implementasi pada Android

5.1.6.1 Implementasi Program

1. Fungsi mengambil semua data di *web service*.

Untuk mengambil semua data dari pin 2 hingga pin 5 dengan format data JSON. Serta akan menampilkan status pada setiap pin yang ditampilkan dalam bentuk ON / OFF pada semua *button* di aplikasi.

```

1. public void getSemua(){
2.     JSONParser jParser = new JSONParser();
3.     JSONObject json = jParser.AmbilJson(url);
4.
5.     try {
6.         arduino = json.getJSONArray("data");
7.         for (int i = 0; i < arduino.length(); i++) {
8.
9.             JSONObject ar = arduino.getJSONObject(i);
10.
11.             TextView status = (TextView) findViewById(R.id.txtstStatus);
12.             TextView pin = (TextView) findViewById(R.id.txtstPin);
13.
14.
15.             tButton = (ToggleButton) findViewById(R.id.ButtonLampu1);
16.             tButton2 = (ToggleButton) findViewById(R.id.ButtonLampu2);
17.             tButton3 = (ToggleButton) findViewById(R.id.ButtonLampu3);
18.             tButton4 = (ToggleButton) findViewById(R.id.ButtonLampu4);
19.
20.

```

```
21. String status_d = ar.getString("status").toString();
22. String pin_d = ar.getString("pin").toString();
23.
24. status.setText(status_d);
25. pin.setText(pin_d);
26.
27.
28. if (pin.getText().toString().equals("2")){
29.
30.     if (status.getText().toString().equals("1")){
31.         tButton.setChecked(true);
32.     }else{
33.         tButton.setChecked(false);
34.     }
35.
36. }
37.
38. if (pin.getText().toString().equals("3")){
39.
40.     if (status.getText().toString().equals("1")){
41.         tButton2.setChecked(true);
42.     }else{
43.         tButton2.setChecked(false);
44.     }
45.
46. }
47.
48. if (pin.getText().toString().equals("4")){
49.
50.     if (status.getText().toString().equals("1")){
51.         tButton3.setChecked(true);
52.     }else{
53.         tButton3.setChecked(false);
54.     }
55.
56. }
57.
58. if (pin.getText().toString().equals("5")){
59.
60.     if (status.getText().toString().equals("1")){
61.
62.         tButton4.setChecked(true);
63.     }else{
64.         tButton4.setChecked(false);
65.     }
66.
67. }
68.
69.
70.
71. }
72.
73. } catch (JSONException e) {
74.     e.printStackTrace();
75.     // TODO: handle exception
```

76.	}
77.	}
78.	}

Gambar 5.7 Fungsi Mengambil Semua Data di *Web Service* pada Android

Sumber : [Implementasi]

Penjelasan program pada gambar 5.7 adalah :

- 1) Pada baris 2 inialisasi JSONParser.
- 2) Pada baris 3 inialisasi JSONObject.
- 3) Pada baris 6 mengambil array data.
- 4) Pada baris 7 looping data array.
- 5) Pada baris 9 mengambil JSONObject dengan nilai indeks yang terkait.
- 6) Pada baris 11-18 inialisasi variabel yang akan digunakan.
- 7) Pada baris 21-22 simpan status dan pin di variabel string.
- 8) Pada baris 28-34 jika pin bernilai 2 dan status bernilai 1 maka button 1 akan menampilkan on. Apabila pin bernilai 2 dan status bernilai 0 maka button 1 akan menampilkan off.
- 9) Pada baris 38-44 jika pin bernilai 3 dan status bernilai 1 maka button 2 akan menampilkan on. Apabila pin bernilai 3 dan status bernilai 0 maka button 2 akan menampilkan off.
- 10) Pada baris 48-54 jika pin bernilai 4 dan status bernilai 1 maka button 3 akan menampilkan on. Apabila pin bernilai 4 dan status bernilai 0 maka button 3 akan menampilkan off.
- 11) Pada baris 58-65 jika pin bernilai 5 dan status bernilai 1 maka button 4 akan menampilkan on. Apabila pin bernilai 5 dan status bernilai 0 maka button 4 akan menampilkan off.

2. Mengambil data berdasarkan pin dan status.

Untuk mengambil data JSON berdasarkan nilai pin dengan status tertentu ketika *button* tertentu pada aplikasi ditekan. Tampilan *button* bentuknya berupa ON / OFF.

```
1. tButton.setOnCheckedChangeListener(new OnCheckedChangeListener()
2. {
3. @Override
4. public void onCheckedChanged(CompoundButton buttonView, boolean
5. isChecked) {
6.
7.     if (isChecked) {
8.         JSONParser jParser = new JSONParser();
9.         JSONObject json = jParser.AmbilJson(url+"?pin=2&status=1");
10.
11.     } else {
12.         JSONParser jParser = new JSONParser();
13.         JSONObject json = jParser.AmbilJson(url+"?pin=2&status=0");
14.     }
15.
16. }
17. });
18.
19. tButton2.setOnCheckedChangeListener(new OnCheckedChangeListener()
20. {
21. @Override
22. public void onCheckedChanged(CompoundButton buttonView, boolean
23. isChecked) {
24.
25.     if (isChecked) {
26.         JSONParser jParser = new JSONParser();
27.         JSONObject json = jParser.AmbilJson(url+"?pin=3&status=1");
28.
29.     } else {
30.         JSONParser jParser = new JSONParser();
31.         JSONObject json = jParser.AmbilJson(url+"?pin=3&status=0");
32.     }
33.
34. }
35. });
36.
37. tButton3.setOnCheckedChangeListener(new OnCheckedChangeListener()
38. {
39. @Override
40. public void onCheckedChanged(CompoundButton buttonView, boolean
41. isChecked) {
42.
43.     if (isChecked) {
44.         JSONParser jParser = new JSONParser();
45.         JSONObject json = jParser.AmbilJson(url+"?pin=4&status=1");
46.
47.     } else {
48.         JSONParser jParser = new JSONParser();
49.         JSONObject json = jParser.AmbilJson(url+"?pin=4&status=0");
50.     }
51.
52. }
53. });
54.
55.
```

```

56. tButton4.setOnCheckedChangeListener(new OnCheckedChangeListener()
57. {
58. @Override
59. public void onCheckedChanged(CompoundButton buttonView, boolean
60. isChecked) {
61.
62.     if (isChecked) {
63.         JSONParser jParser = new JSONParser();
64.         JSONObject json = jParser.AmbilJson(url+"?pin=5&status=1");
65.
66.     } else {
67.         JSONParser jParser = new JSONParser();
68.         JSONObject json = jParser.AmbilJson(url+"?pin=5&status=0");
69.     }
70.
71. }
72. });
73.

```

Gambar 5.8 Syntax Mengambil Data Berdasarkan Pin dan Status pada Android

Sumber : [Implementasi]

Penjelasan program pada gambar 5.8 adalah :

- 1) Pada baris 1-17 ketika button1 ditekan on akan mengambil data JSON dengan pin=2&status=1, sebaliknya bila off akan mengambil data JSON dengan pin=2&status=0.
- 2) Pada baris 19-35 ketika button2 ditekan on akan mengambil data JSON dengan pin=3&status=1, sebaliknya bila off akan mengambil data JSON dengan pin=3&status=0.
- 3) Pada baris 37-53 ketika button3 ditekan on akan mengambil data JSON dengan pin=4&status=1, sebaliknya bila off akan mengambil data JSON dengan pin=4&status=0.
- 4) Pada baris 56-72 ketika button4 ditekan on akan mengambil data JSON dengan pin=5&status=1, sebaliknya bila off akan mengambil data JSON dengan pin=5&status=0.

3. Class JSON parser

Di bawah ini merupakan *class* untuk mengolah data dari JSON. Mengelola JSON dibutuhkan *httpClient* dan *httpResponse* untuk method yang digunakan dapat menggunakan metode get maupun post, pada *syntax* program ini menggunakan method GET.

```
1.  Public class JSONParser {
2.
3.  static InputStream is = null;
4.  static JSONObject jsonObj = null;
5.  static String json = "";
6.
7.  public JSONObject AmbilJson(String url) {
8.
9.  try {
10. DefaultHttpClient httpClient = new DefaultHttpClient();
11. // Prepare a Request object URL
12. HttpGet httpGet = new HttpGet(url);
13. // Execute the request
14. HttpResponse httpResponse = httpClient.execute(httpGet);
15. // Get the response entity
16. HttpEntity httpEntity = httpResponse.getEntity();
17. // After get entity contents and convert it to to string
18. is = httpEntity.getContent();
19.
20. } catch (UnsupportedEncodingException e) {
21. e.printStackTrace();
22. } catch (ClientProtocolException e) {
23. e.printStackTrace();
24. } catch (IOException e) {
25. e.printStackTrace();
26. }
27.
28. // Get the server response
29. try {
30. BufferedReader reader = new BufferedReader(new InputStreamReader(is,
31. "iso-8859-1"), 8);
32. StringBuilder sb = new StringBuilder();
33. String line = null;
34.
35. // Read Server Response
36. while ((line = reader.readLine()) != null) {
37. sb.append(line);
38. }
39. is.close();
40.
41. //Append Server Response To content String
42. json = sb.toString();
43. } catch (Exception e) {
44. Log.e("Buffer Error", "Error converting result " + e.toString());
45. }
46.
47. try {
48. jsonObj = new JSONObject(json);
49. } catch (JSONException e) {
50. Log.e("JSON Parser", "Error parsing data " + e.toString());
51. }
52.
53. return jsonObj;
54.
55. }
```

56.	}
-----	---

Gambar 5.9 *Class JSON Parser* pada Android

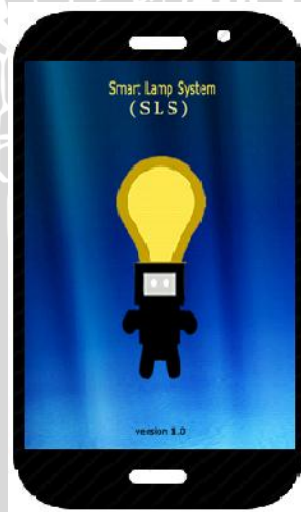
Sumber : [Implementasi]

Penjelasan *syntax* program pada gambar 5.9 adalah :

- 1) Pada baris 10 mempersiapkan *httpClient*.
- 2) Pada baris 12 mempersiapkan sebuah *request* berupa *object* URL.
- 3) Pada baris 13 akan mengeksekusi *request*.
- 4) Pada baris 16 akan mendapatkan entity *response*.
- 5) Pada baris 18 setelah mendapatkan isi dari entity akan di konversi ke dalam bentuk string.
- 6) Pada baris 30-33 mendapatkan *response* dari server.
- 7) Pada baris 36-39 membaca *response* dari server.
- 8) Pada baris 42-51 menambahkan server *response* ke *content* string.

5.1.6.2 Implementasi Antarmuka

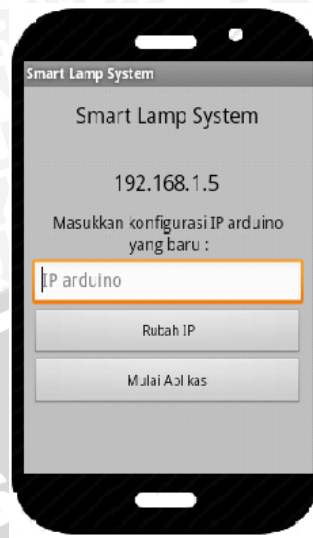
Dibawah ini merupakan gambar dari tampilan antarmuka aplikasi android.



Gambar 5.10 Tampilan *Flash Screen* Aplikasi Android

Sumber : [Implementasi]

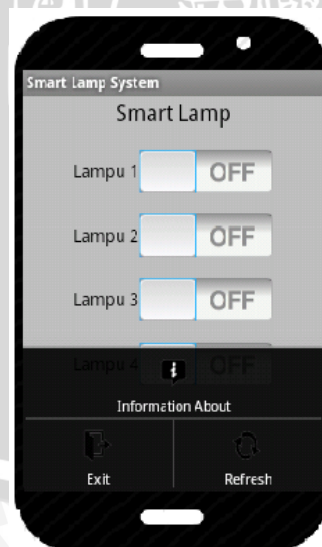
Gambar 5.10 merupakan tampilan awal ketika mengakses aplikasi *smart home system* ini. Nama aplikasi ini adalah *Smart Lamp System* yang berfungsi untuk memonitoring dan mengontrol kondisi lampu.



Gambar 5.11 Tampilan Awal Memasukkan IP Aplikasi Android

Sumber : [Implementasi]

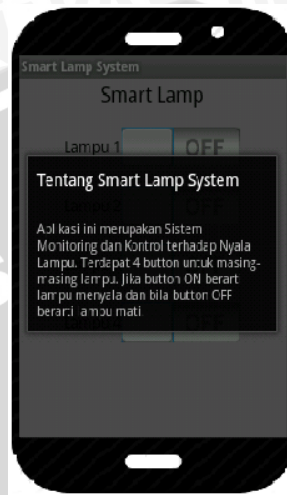
Gambar 5.11 merupakan tampilan awal untuk memasukkan IP dari arduino. Pada menu ini dapat menampilkan konfigurasi IP sebelumnya yang sudah tersimpan pada *database*. Ketika ingin merubah IP sebelumnya yaitu dengan memasukkan IP baru pada *text field* yang telah tersedia, kemudian tekan *button* rubah IP. Setelah konfigurasi IP berhasil dilakukan, kemudian tekan *button* mulai aplikasi.



Gambar 5.12 Tampilan Menu Utama Aplikasi Android

Sumber : [Implementasi]

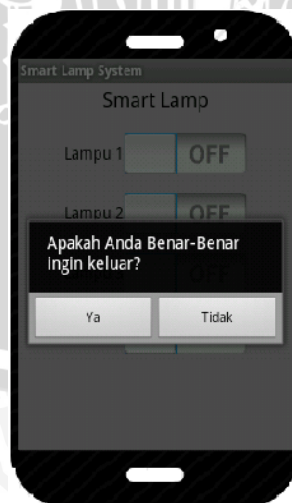
Gambar 5.12 merupakan tampilan menu utama, yang berisikan 4 buah *button* dari masing-masing lampu yang akan di monitoring dan di kontrol keadaannya. Dan juga terdapat 3 menu pilihan tambahan yaitu menu *information about*, *exit* dan *refresh*.



Gambar 5.13 Tampilan Menu *Information About* Aplikasi Android

Sumber : [Implementasi]

Gambar 5.13 merupakan tampilan menu *information about* yang berisikan tentang deskripsi aplikasi *Smart Lamp System*. Deskripsi singkat tersebut adalah sebagai berikut “Aplikasi ini merupakan Sistem Monitoring dan Kontrol terhadap Nyala Lampu. Terdapat 4 button untuk masing-masing lampu. Jika button ON berarti lampu menyala dan bila button OFF berarti lampu mati.”



Gambar 5.14 Tampilan Menu *Exit* Aplikasi Android

Sumber : [Implementasi]

Gambar 5.14 merupakan tampilan menu *exit* untuk keluar dari aplikasi. Terdapat pesan peringatan sebelum pengguna mengakhiri aplikasi ini.

5.1.7 Implementasi Basis Data

Pada sistem *web service* menggunakan basis data sebagai tempat untuk menyimpan data. Tabel 5.1 merupakan penjabaran dari tabel data yang digunakan untuk menyimpan data kondisi keadaan lampu, sedangkan Gambar 5.15 merupakan implementasi dari basis data pada *database* MySQL.

Tabel Data		
Kolom	Tipe Data	Deskripsi
Ip	varchar	Alamat IP yang digunakan oleh arduino
Pin	int	Nomor pin pada arduino
status	int	Status lampu
Nama	varchar	Kondisi keadaan lampu

Tabel 5.1 Rancangan tabel untuk menyimpan data

Sumber : [Implementasi]

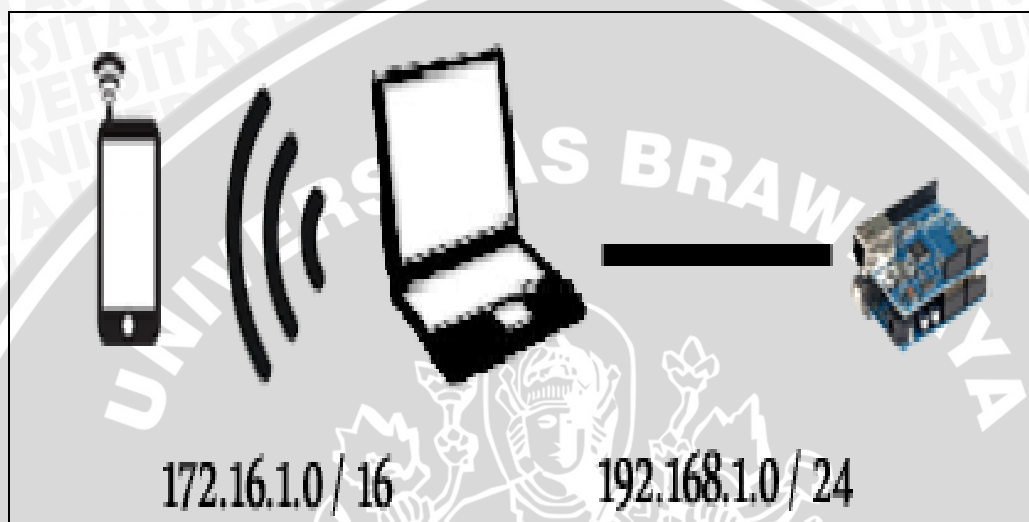
			nama	status	pin	ip
<input type="checkbox"/>			Lampu Nyala	1	2	192.168.1.5
<input type="checkbox"/>			Lampu Nyala	1	3	192.168.1.5
<input type="checkbox"/>			Lampu Mati	0	4	192.168.1.5
<input type="checkbox"/>			Lampu Mati	0	5	192.168.1.5

Gambar 5.15 Implementasi basis data

Sumber : [Implementasi]

5.2 Pengujian

Pengujian dilakukan menggunakan topologi jaringan *Ad Hoc* dengan sebuah hp android sebagai *client* yang berada pada *network address* 172.16.1.0 / 16 sedangkan arduino berada pada *network address* 192.168.1.0 / 24 dapat dilihat pada gambar 5.15 berikut ini.



Gambar 5.16 Topologi Jaringan *Ad Hoc* untuk Pengujian

Sumber : [Pengujian]

5.2.1 Pengujian Akses *Web Service*

Pengujian akses pada *web service* apakah dalam implementasinya *web service* sudah berjalan seperti yang diharapkan yaitu *resource* data yang diinginkan sesuai dengan *request* dari pengguna. Hasil pengujian dapat ditunjukkan pada tabel 5.1 berikut ini.

No.	Pin	Status	HTTP Metode	Isi Database	JSON	Hasil
1.	2	1	GET /?pin=2&status=1	<input type="checkbox"/> Lampu Nyala 2 1	{"data": [{"nama": "Lampu Nyala", "pin": "2", "status": "1"}]}	Valid
	2	0	GET /?pin=2&status=0	<input type="checkbox"/> Lampu Mati 2 0	{"data": [{"nama": "Lampu Mati", "pin": "2", "status": "0"}]}	Valid
2.	3	1	GET /?pin=3&status=1	<input type="checkbox"/> Lampu Nyala 3 1	{"data": [{"nama": "Lampu Nyala", "pin": "3", "status": "1"}]}	Valid
	3	0	GET /?pin=3&status=0	<input type="checkbox"/> Lampu Mati 3 0	{"data": [{"nama": "Lampu Mati", "pin": "3", "status": "0"}]}	Valid
3.	4	1	GET /?pin=4&status=1	<input type="checkbox"/> Lampu Nyala 4 1	{"data": [{"nama": "Lampu Nyala", "pin": "4", "status": "1"}]}	Valid
	4	0	GET /?pin=4&status=0	<input type="checkbox"/> Lampu Mati 4 0	{"data": [{"nama": "Lampu Mati", "pin": "4", "status": "0"}]}	Valid
4.	5	1	GET /?pin=5&status=1	<input type="checkbox"/> Lampu Nyala 5 1	{"data": [{"nama": "Lampu Nyala", "pin": "5", "status": "1"}]}	Valid
	5	0	GET /?pin=5&status=0	<input type="checkbox"/> Lampu Mati 5 0	{"data": [{"nama": "Lampu Mati", "pin": "5", "status": "0"}]}	Valid
5.	ALL	ALL	GET /?pin=&status=	<input type="checkbox"/> Lampu Nyala 2 1 <input type="checkbox"/> Lampu Nyala 3 1 <input type="checkbox"/> Lampu Mati 4 0 <input type="checkbox"/> Lampu Mati 5 0	{"data": [{"nama": "Lampu Nyala", "pin": "2", "status": "1"}, {"nama": "Lampu Nyala", "pin": "3", "status": "1"}, {"nama": "Lampu Mati", "pin": "4", "status": "0"}, {"nama": "Lampu Mati", "pin": "5", "status": "0"}]}	Valid

Tabel 5.2 Pengujian Web Service

Sumber : [Pengujian]

5.2.2 Pengujian Sistem Keseluruhan

Pengujian sistem keseluruhan terdiri dari dua skenario pengujian yaitu pengujian ketika memberikan *input* saklar secara manual dan pengujian melalui pemberian *input* pada aplikasi android.

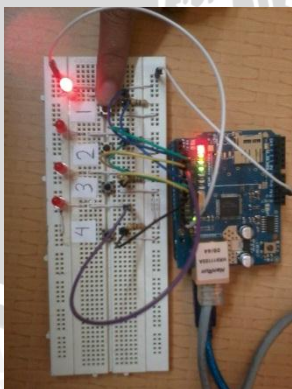
1. Skenario Pertama : pengujian *input* saklar manual

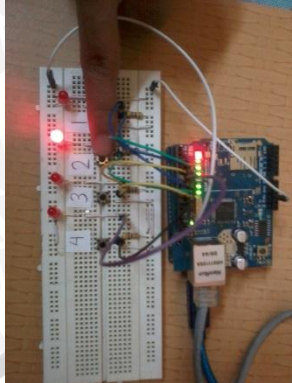
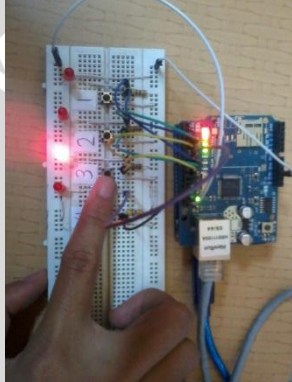
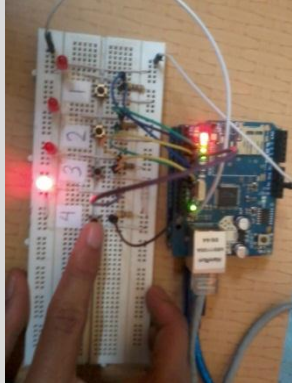
No.	Nama	Kondisi Saklar	Keadaan Lampu	Hasil
1.	Saklar 1	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
2.	Saklar 2	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
3.	Saklar 3	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
4.	Saklar 4	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid

Tabel 5.3 Pengujian *Input* Saklar Manual

Sumber : [Pengujian]

Hasil yang diperlihatkan pada Tabel 5.2 merupakan pengujian ketika saklar pada rangkaian elektronik ditekan, yaitu baik dalam kondisi ON maupun OFF. Sedangkan untuk pembuktiannya dapat dilihat pada tabel 5.3 berikut ini.

No.	Nama	Tampilan Saklar Lampu	Tampilan Aplikasi Android
1.	Saklar 1		<div style="text-align: center;">Smart Lamp</div> <p>Lampu 1 <input checked="" type="checkbox"/> ON</p> <p>Lampu 2 <input type="checkbox"/> OFF</p> <p>Lampu 3 <input type="checkbox"/> OFF</p> <p>Lampu 4 <input type="checkbox"/> OFF</p>

<p>2.</p>	<p>Saklar 2</p>		<p>Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input checked="" type="checkbox"/> ON</p> <p>Lampu 3 <input type="checkbox"/> OFF</p> <p>Lampu 4 <input type="checkbox"/> OFF</p>
<p>3.</p>	<p>Saklar 3</p>		<p>Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input type="checkbox"/> OFF</p> <p>Lampu 3 <input checked="" type="checkbox"/> ON</p> <p>Lampu 4 <input type="checkbox"/> OFF</p>
<p>4.</p>	<p>Saklar 4</p>		<p>Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input type="checkbox"/> OFF</p> <p>Lampu 3 <input type="checkbox"/> OFF</p> <p>Lampu 4 <input checked="" type="checkbox"/> ON</p>

Tabel 5.4 Pembuktian *Input* Saklar Manual

Sumber : [Pengujian]

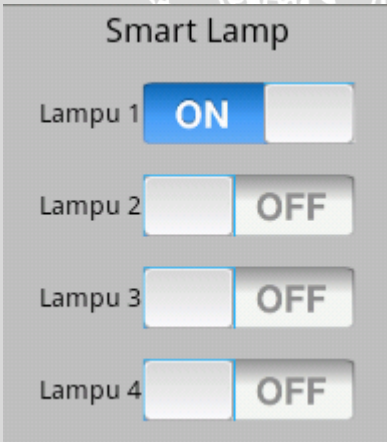
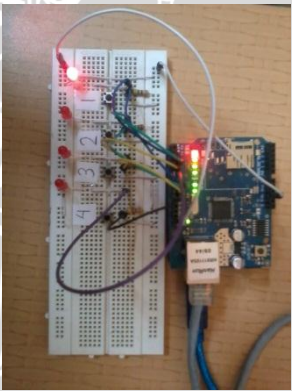
2. Skenario kedua : pengujian *input* aplikasi android

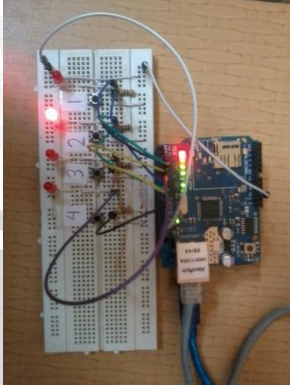
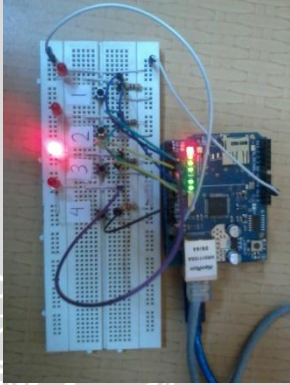
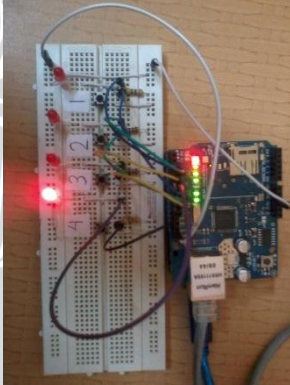
No.	Nama	Kondisi Button	Keadaan Lampu	Hasil
1.	Button 1	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
2.	Button 2	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
3.	Button 3	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid
4.	Button 4	ON	Lampu Nyala	Valid
		OFF	Lampu Mati	Valid

Tabel 5.5 Pengujian *Input* Aplikasi Android

Sumber : [Pengujian]

Hasil yang diperlihatkan pada Tabel 5.4 merupakan pengujian ketika button pada aplikasi android ditekan, yaitu baik dalam kondisi ON maupun OFF. Sedangkan untuk pembuktiannya dapat dilihat pada tabel 5.5 berikut ini.

No.	Nama	Tampilan Aplikasi Android	Tampilan Keadaan Lampu
1.	Button 1		

<p>2.</p>	<p>Button 2</p>	<p style="text-align: center;">Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input checked="" type="checkbox"/> ON</p> <p>Lampu 3 <input type="checkbox"/> OFF</p> <p>Lampu 4 <input type="checkbox"/> OFF</p>	
<p>3.</p>	<p>Button 3</p>	<p style="text-align: center;">Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input type="checkbox"/> OFF</p> <p>Lampu 3 <input checked="" type="checkbox"/> ON</p> <p>Lampu 4 <input type="checkbox"/> OFF</p>	
<p>4.</p>	<p>Button 4</p>	<p style="text-align: center;">Smart Lamp</p> <p>Lampu 1 <input type="checkbox"/> OFF</p> <p>Lampu 2 <input type="checkbox"/> OFF</p> <p>Lampu 3 <input type="checkbox"/> OFF</p> <p>Lampu 4 <input checked="" type="checkbox"/> ON</p>	

Tabel 5.6 Pembuktian Aplikasi Android

Sumber : [Pengujian]

5.3 Analisa

Proses analisa bertujuan untuk mendapatkan kesimpulan dari hasil pengujian aplikasi *smart home system* yang telah dilakukan. Proses analisa yang dilakukan meliputi analisa pengujian akses *web service*, dan pengujian sistem secara keseluruhan. Untuk pengujian sistem secara keseluruhan meliputi pengujian ketika diberikan *input* dari saklar secara manual dan pengujian ketika diberikan *input* dari aplikasi android, yang akan dianalisa hasil akhirnya.

5.3.1 Analisa Pengujian Akses Web Service

Analisa berdasarkan hasil pengujian akses *web service* dapat disimpulkan bahwa implementasi *web service* yang dibentuk mampu menyediakan *resource* data dengan menggunakan format data JSON, yang nantinya data tersebut akan dikirimkan oleh server ke sisi *client* pada aplikasi android. Data JSON yang dikirimkan nantinya sesuai dengan *request* dari *client* pengguna dari aplikasi, dapat dibuktikan dari tabel pengujian tersebut bahwa terbukti valid.

5.3.2 Analisa Pengujian Sistem Keseluruhan

Analisa berdasarkan dua skenario hasil pengujian sistem secara keseluruhan dapat dijelaskan yaitu skenario pertama ketika diberikan *input* dari saklar secara manual menunjukkan bahwa dari keempat buah saklar yang telah dilakukan pengujian baik secara ON maupun OFF telah berjalan sesuai dengan keinginan pengguna, yaitu dapat menghidupkan dan mematikan lampu sesuai dengan saklar dari lampu yang telah ditekan.

Sedangkan untuk pengujian pada skenario kedua yaitu ketika diberikan *input* dari aplikasi android menunjukkan bahwa aplikasi android juga dapat berjalan sesuai dengan apa yang diinginkan oleh pengguna. Pengujian yang telah dilakukan baik secara ON maupun OFF dalam menghidupkan dan mematikan lampu dapat sesuai dengan keinginan. Jadi hasil pengujian sistem secara keseluruhan pada skripsi ini terbukti valid.

BAB VI PENUTUP

6.1 Kesimpulan

Setelah seluruh tahapan perancangan, implementasi dan pengujian sudah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Berdasarkan hasil perancangan dan implementasi *web service* pada *smart home system*, diketahui bahwa dengan menggunakan *web service* komunikasi data antara perangkat bergerak dengan *board* mikrokontroler dapat terbentuk, meskipun berbeda *platform* dan juga bahasa pemrograman, dapat dijabarkan yaitu *web service* dengan menggunakan bahasa pemrograman PHP, perangkat bergerak menggunakan bahasa pemrograman Java, dan *board* mikrokontroler menggunakan bahasa pemrograman C++.
2. Berdasarkan hasil perancangan dan implementasi program *board* mikrokontroler pada *smart home system*, diketahui bahwa *board* mikrokontroler dapat menerima *request* data dari *web service* dan dapat mengirimkan *request* data ke *web service* dengan menggunakan metode HTTP GET.
3. Berdasarkan hasil perancangan dan implementasi aplikasi perangkat bergerak pada *smart home system*, diketahui bahwa aplikasi perangkat bergerak dapat mengambil *resource* data dari *web service* yang menggunakan format data JSON sesuai dengan *request* yang dilakukan.
4. Berdasarkan hasil pengujian, diketahui bahwa sistem berjalan dengan baik yaitu dapat mengontrol dan memonitoring keadaan lampu ketika dioperasikan secara manual dengan menggunakan saklar maupun melalui aplikasi yang telah di *install* pada perangkat bergerak.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian ini antara lain.

1. Penelitian ini belum memperhatikan aspek keamanan sistem, penelitian selanjutnya dapat lebih diperhatikan lagi dalam segi keamanan sistem.
2. Penelitian ini baru sebatas mengontrol dan memonitoring lampu LED melalui aplikasi perangkat bergerak, penelitian selanjutnya dapat diimplementasikan ke peralatan elektronik lain.
3. Perlunya penelitian lebih lanjut terkait dengan *real time monitoring*.

UNIVERSITAS BRAWIJAYA



DAFTAR PUSTAKA

- [SMA-13] Smarthomeenergy. “*What is Smart Home ?*”.
<http://www.smarthomeenergy.co.uk/what-smart-home>
- [YUS-13] Yusuf, Oik. 17 Mei 2013. “Android Jadi Juara di Indonesia”.
<http://tekno.kompas.com/read/2013/05/17/10474329/android.jadi.quot.otjuaraquot.di.indonesia>.
- [ADA-11] Adamczyk Paul *at all* . 2011. “*REST and Web Services: In Theory and in Practice*”. Booz Allen Hamilton Inc.
- [JEE-11] Jeevan, Gnana. 2011. “*SOAP (Simple Object Access Protocol)*”. MAMCE, New Delhi.
- [AJI-13] Aji Sena , Samuel. 2013. “Perancangan dan Pembuatan *Application Programing Interface Server untuk Arduino*”. Fakultas Teknik, Universitas Brawijaya, Malang.
- [WWW-04] *World Wide Web Consortium (W3C)*.2004. “*Web Service Architecture*”. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#introduction>.
- [PIY-13] Piyare, Rajeev. 2013. “*Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone*”. Jurnal : Mokpo National University.
- [BAN-08] Banzi, Massimo. 2008. “*Getting Started with Arduino*”. O’Reilly.
- [DJU-11] Djuandi, Feri.2011. “Pengenalan Arduino”. Copyright tobuku.com.
- [WAR-06] Wardhana, Lingga. 2006. “Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, *Hardware*, dan Aplikasi”. Yogyakarta: Andi Offset.
- [JSO-13] JSON.org. 2013. “Pengenalan JSON”. <https://www.json.org>,
- [AKB-12] Akbarul Huda, Arif. 2012. “24 Jam Pintar Pemrograman Android”. Yogyakarta: PT. Andi.
- [BAY-12] Bayu Purwanto, Arief. 2012. “Memproses Data JSON pada Android”. <http://bayu.freelancer.web.id/2012/02/24/memproses-data-json-pada-android/>