

Rancang Bangun *Digital Home Assistant* dengan Perintah Suara Menggunakan Raspberry Pi dan *Smartphone*

(*Design and Built Digital Home Assistant with Voice Commands Using Raspberry Pi and Smartphone*)

Isni Fachri Rizal, I Wayan Agus Arimbawa, Royana Afwani

Program Studi teknik Informatika, Fakultas Teknik, Universitas Mataram

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: isnifachririzal@gmail.com, arimbawa@unram.ac.id, royana@unram.ac.id

Abstract—Digital home assistant is one part of smart home technology. Smart home is one of technology research that focuses on home automation, so the interaction between human and home becomes easier and more practical. One of the interactions that can be done is using voice commands. Currently digital home assistant technology can be implemented with a variety of concepts and various tools. One of the tools that can be used is smart phone. In order for home electronics devices to communicate with smart phones, use a micro controller that is attached to the home electronics. Some micro controllers that can be used are Raspberry Pi and Arduino. This research built a digital home assistant with voice commands using Raspberry Pi and smart phone. The built-in digital home assistant is used for lamp, TV and AC control. The Google Speech API is used on smart phones for the process of translation of voice commands provided by users of the system. The stages of this research are: system requirements, designing system architecture and merging of raspberry pi devices, software design (application control, web service, and mobile applications), database design, hardware development, software development, database development, and then testing also evaluation. The results obtained for the control of lights, TV and air conditioning with voice commands using smart phones and Raspberry Pi is a system can run with 90% success and the average time of command execution reaches is 5.32 seconds.

Key words: *Smart Home, Smart Phone, Raspberry Pi, Android, Arduino, Micro Controller.*

I. PENDAHULUAN

Smart home merupakan salah satu riset teknologi yang berfokus untuk *home automation*[1] sehingga interaksi antara manusia dengan rumah menjadi lebih mudah, praktis, dan dapat membantu pekerjaan rumah menjadi lebih cepat. Salah satu dari teknologi yang dapat digunakan adalah *smartphone*. *Smartphone* merupakan suatu alat komunikasi yang memiliki berbagai kelebihan dibandingkan alat komunikasi yang lain, beberapa

diantaranya adalah memiliki kemampuan seperti komputer dan bersifat *Mobile*. Salah satu *tool* yang dapat kita gunakan untuk menerima perintah suara adalah *Google Speech API*. Dengan *Google Speech API* kita dapat membuat aplikasi yang dapat memahami apa yang dikatakan oleh manusia, sehingga interaksi antara *digital home assistant* dengan pengguna dapat dilakukan menggunakan suara. Namun untuk membangun suatu *digital home assistant*, peralatan elektronika di rumah harus dapat berinteraksi dengan *smartphone*. Untuk itu dibutuhkanlah *microcontroller* yang akan dihubungkan dengan peralatan elektronika di rumah agar peralatan tersebut dapat berinteraksi dengan *smartphone*.

Microcontroller merupakan suatu alat yang dapat bekerja seperti komputer dan dapat digunakan untuk mengontrol peralatan elektronika. Beberapa contoh dari *microcontroller* adalah Raspberry Pi, Arduino dan Wemos. Pada *microcontroller* instruksi yang diberikan diubah ke dalam bentuk sinyal dan dikirimkan ke modul-modul elektronika yang akan digunakan untuk mengendalikan peralatan elektronika. Sehingga dengan adanya *Microcontroller* pengendalian peralatan elektronika dapat dilakukan dengan berbagai cara.

Sebelumnya sudah ada penelitian tentang mengontrol lampu menggunakan *smartphone* android sebagai *input* dari *user*. *input* tersebut kemudian dikirimkan ke Arduino melalui jaringan *bluetooth*. Dari percobaan tersebut, hasil positif yang didapatkan adalah efisiensi waktu dan tenaga, praktis untuk digunakan dan disimpan, serta satu aplikasi dapat diterapkan pada beberapa *smartphone* android, asalkan android di bawah versi 4.4 (KitKat)[2].

Kemudian penelitian selanjutnya tentang mengendalikan lampu menggunakan aplikasi *mobile* dimana *input* yang diterima oleh aplikasi *mobile* langsung dikirimkan ke Arduino menggunakan jaringan WiFi. Pada percobaan tersebut sistem dapat berjalan dengan jarak maksimal 2 meter antara alat kontrol dengan arduino[3].

Aryanta, melakukan kontrol lampu melalui sebuah *website*. Percobaan tersebut menggunakan 6 buah lampu yang dihubungkan dengan Arduino dan *ethernet shield*. Pada percobaan tersebut didapatkan kesimpulan bahwa

sistem tersebut dapat diimplementasikan dengan persentase keberhasilan mencapai 86,112%[4].

Latief, melakukan pengendalian lampu dengan menggunakan perintah suara melalui *smartphone* dan Raspberry Pi. Pada percobaan tersebut didapatkan kesimpulan bahwa perintah suara berhasil digunakan untuk mengendalikan perangkat elektronik rumah tangga dengan baik tanpa adanya kendala[5].

Berdasarkan tinjauan pustaka di atas, maka hasil dari pustaka tersebut akan dikombinasikan untuk membangun *digital home assistant* menggunakan Raspberry Pi dan *Smartphone* dengan perintah suara sebagai masukan.

II. METODE PERANCANGAN

A. Digital Home Assistant

Internet of thing merupakan sebuah konsep dan paradigma yang menganggap semua hal atau benda dapat dihubungkan melalui kabel ataupun *wireless* dan saling berinteraksi serta bekerja dengan peralatan lain untuk mencapai suatu tujuan yang diinginkan. Dalam hal ini penelitian dan pengembangan dilakukan untuk menciptakan dunia yang sangat cerdas. Sebuah dunia dimana sesuatu yang nyata, *digital* dan *virtual* dikumpulkan untuk menciptakan suatu lingkungan yang lebih cerdas[6].

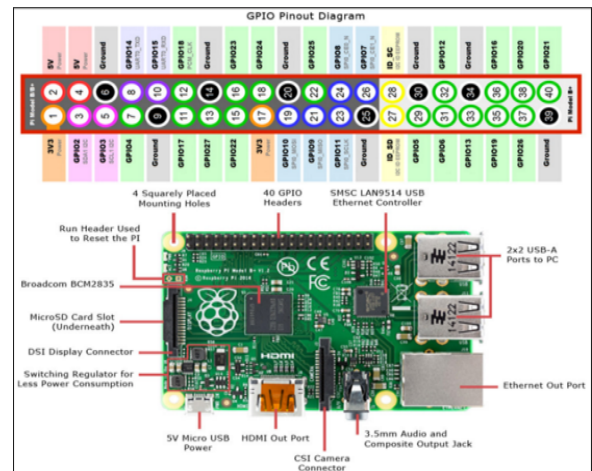
Smart home dapat diterapkan di dalam rumah atau bangunan yang didiami oleh manusia. *Smart home* adalah kombinasi dari teknologi informasi dan pengolahan informasi dengan komputer. Tujuan utama dari implementasi *Smart Home* adalah terpenuhinya konsep keamanan, kenyamanan dan penghematan di dalam rumah. Implementasi *Smart Home* sendiri dapat dilakukan dengan konsep *Internet of Things* dengan memanfaatkan sensor, aktuator, komunikasi dan pemrosesan komputer.

B. Raspberry

Raspberry Pi adalah peralatan yang umum digunakan dalam membangun sistem *Internet of Things*. Raspberry Pi adalah pemroses mikro yang berukuran kompak, minim pemakaian daya, walaupun memiliki kemampuan komputasi yang rendah. Raspberry Pi cocok digunakan sebagai pemroses lokal seperti *agregator*, *preprocessor*, dan *gateway*. Gambar 1 merupakan Gambar dari Raspberry Pi 3 model B[7].

Raspberry Pi juga dilengkapi sebuah slot penyimpanan mikro SD untuk menyimpan berkas-berkas yang diperlukan seperti sistem operasi, aplikasi dan berkas-berkas lainnya. Untuk mendukung keluaran audio visual, Raspberry Pi menyediakan port HDMI yang umum digunakan untuk perangkat elektronik. Sumber daya Raspberry Pi diberikan melalui konektor mikro USB dengan tegangan 5VDC. Raspberry Pi juga dilengkapi konektor video komposit RCA untuk keluaran video serta *jack* stereo 3,5 mm untuk keluaran audionya. Raspberry Pi memiliki 40 kaki-kaki GPIO yang dapat digunakan untuk menghubungkan perangkat *peripheral* tingkat yang lebih rendah dan *expansion boards* lainnya[7].

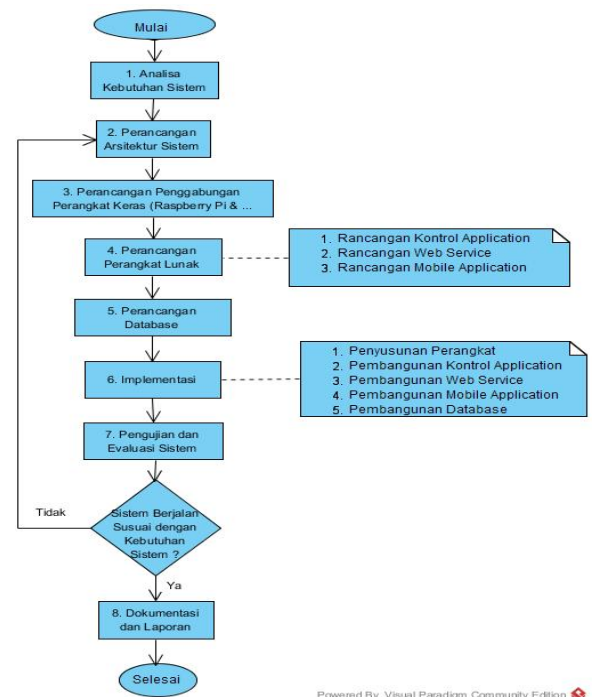
Gambar 1 merupakan Gambar dari Raspberry Pi 3 model B[7].



Gambar 1. Raspberry Pi 3 Model B

C. Alur Jalannya Penelitian

Alur terstruktur dari penelitian dapat dilihat seperti pada Gambar 2.



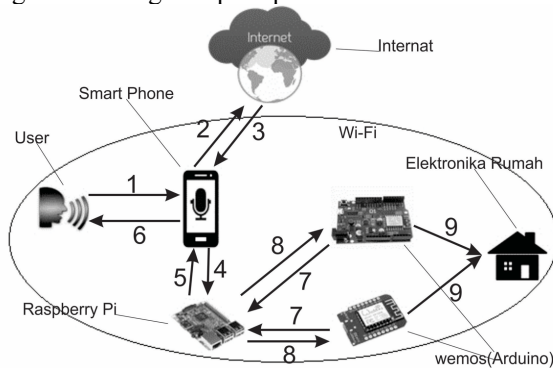
Gambar 2. Alur jalannya penelitian

Gambar 2 menunjukkan bahwa untuk menyelesaikan penelitian ini, dilakukan dengan beberapa tahap yaitu dimulai dengan analisa kebutuhan sistem, perancangan arsitektur sistem dan penggabungan perangkat raspberry pi. Kemudian dilanjutkan dengan perancangan database dan perancangan perangkat lunak yang terdiri dari kontrol aplikasi, *web service*, dan aplikasi *mobile*. Setelah itu dilakukan penyusunan dan pembangunan perangkat keras,

kemudian dilanjutkan dengan pembangunan kontrol *application*, pembangunan *web service*, pembangunan *mobile application*, pembangunan *database*, dan baru kemudian setelah itu dilakukan pengujian dan evaluasi sistem.

D. Rancangan Arsitektur Sistem, Perangkat Keras dan Perangkat Lunak

Gambaran dari arsitektur sistem *digital home assistant* yang akan dibangun seperti pada Gambar 3.



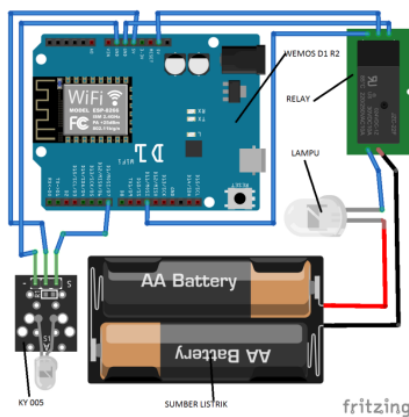
Gambar 3. Arsitektur sistem digital home assistant menggunakan Raspberry Pi dan smartphone

Pada Gambar 3 tersebut sistem saling terhubung dengan menggunakan jaringan WiFi. Jaringan Internet digunakan untuk mengubah suara ke dalam bentuk teks menggunakan bantuan *Google Speech API*, kemudian teks perintah tersebut dikirimkan ke Raspberry Pi dan disimpan pada *database*, kemudian Arduino akan mengambil perintah pada Raspberry Pi untuk memberikan sinyal ke elektronika di rumah.

Rancangan perangkat keras merupakan tahap penyusunan *microcontroller* dengan modul-modul elektronika yang akan dipasang pada objek dari sistem. Gambaran untuk rancangan penyusunan perangkat keras dapat dilihat pada Gambar 4, Gambar 5, dan Gambar 6.

D.1. Untuk lampu dan TV

Rancangan perangkat keras untuk lampu dan TV disajikan pada Gambar 4 yang terdiri dari Wemos D1R2, satu buah modul *relay*, dan satu buah modul KY-005 infra merah transmitter. *Relay* berfungsi sebagai pemutus dan

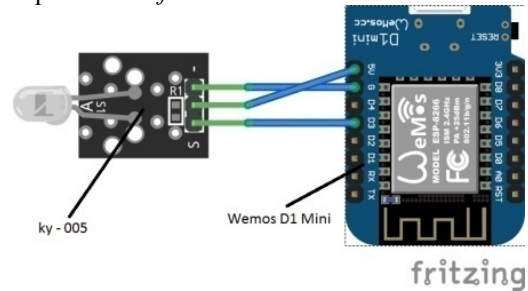


Gambar 4. Arsitektur perangkat keras untuk lampu dan TV

penyambung hubungan listrik arus kuat yang dikendalikan oleh logika pengendalian pada *microcontroller*. Modul infra merah berfungsi sebagai pengirim sinyal infra merah untuk mengendalikan peralatan elektronik.

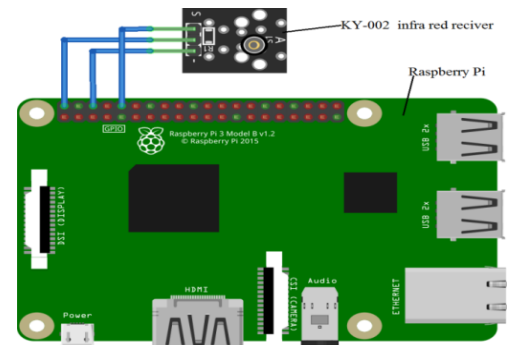
D.2. Untuk AC

Rancangan perangkat keras untuk AC disajikan pada Gambar 5 yang terdiri dari Raspberry Pi dan satu buah modul penerima infra merah KY-002.



Gambar 5. Arsitektur perangkat keras untuk AC

D.3. Untuk merekam sinyal

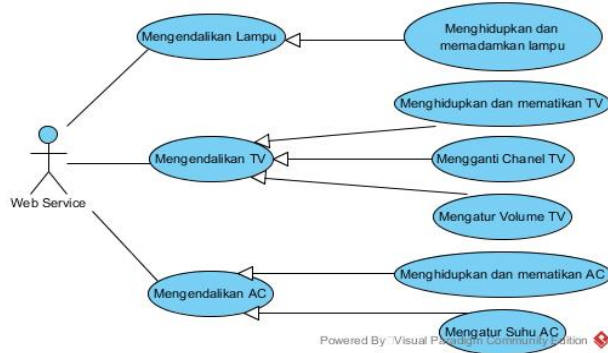


Gambar 6 Arsitektur perangkat keras untuk merekam sinyal infra merah

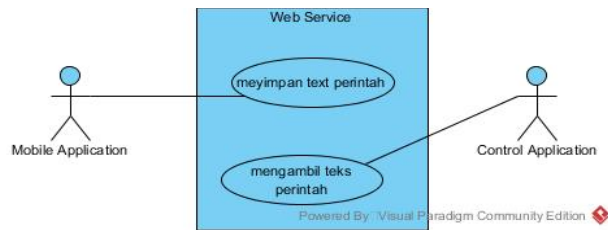
Gambar 5 perangkat keras untuk perekam sinyal terdiri dari Raspberry Pi dan satu buah modul KY-002 infra merah *receiver*.

D.4. Perangkat lunak

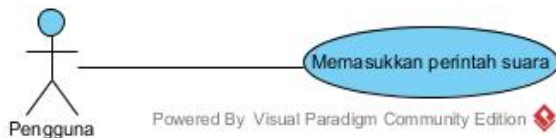
Untuk perangkat lunaknya, sistem *digital home assistant* yang dibangun terdiri dari tiga buah sub sistem yaitu *Control Application*, *Web Service* dan *Mobile Application*. *Control Application* merupakan aplikasi yang akan dipasang pada Wemos (Arduino) dan digunakan untuk mengendalikan alat-alat elektronika di rumah sesuai dengan instruksi yang diberikan oleh pengguna. Gambar 7, 8 dan 9 merupakan diagram *usecase* yang menunjukkan fitur dari *control application*, *web service* dan *mobile application*. Sementara itu *Web Service* merupakan suatu aplikasi yang akan dipasang pada Raspberry Pi dan digunakan sebagai pusat data dari *system digital home assistant*. Aplikasi terakhir yang perlu disiapkan adalah *Mobile Application* dimana merupakan aplikasi yang dipasang pada smart phone dan digunakan untuk menerima perintah suara dari pengguna. *Mobile Application* akan dibangun dengan menggunakan bahasa pemrograman Java.



Gambar 7. Diagram usecase control application



Gambar 8. Diagram usecase web service



Gambar 9. Diagram usecase diagram mobile application

Untuk Gambaran *prototype interface* dari *mobile application* yang akan dibuat dapat dilihat pada Gambar 10 dimana terdapat satu buah tombol bergambar mikrofon, tombol tersebut digunakan untuk memasukkan perintah suara ke dalam *Mobile Application*.



Gambar 10. Prototype pengguna interface aplikasi mobile

III. HASIL DAN PEMBAHASAN

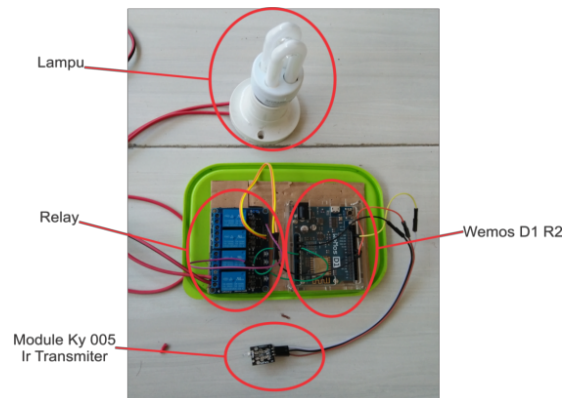
Gambaran dari alur kerja sistem *digital home assistant* yang dibangun pada penelitian ini dapat dijelaskan dengan tahapan sebagai berikut :

- a) Pengguna memasukkan perintah suara ke dalam *Mobile Application*.

- b) *Mobile Application* mengubah suara yang diterima dari pengguna ke dalam bentuk teks dengan menggunakan *Google Speech API*.
- c) Teks perintah pada *mobile application* dikirimkan ke *web service*.
- d) *Web service* menerima dan menyimpan teks perintah dari *mobile application* ke dalam *database*.
- e) Teks perintah yang diterima *web service* disimpan pada *databases*.
- f) *Web service* memberikan *feed back* ke *mobile application* bahwa perintah sedang diproses.
- g) *Mobile application* meneruskan *feed back* dari *web service* ke pengguna.
- h) *Control application* meminta teks perintah yang akan dikerjakan pada *web service*
- i) *Web service* mengambil teks perintah pada *database* untuk diberikan ke *control application*.
- j) Pengambilan perintah pada *database*.
- k) *Web service* memberikan teks perintah ke *control application*.
- l) *Control application* memberikan sinyal ke elektronika sesuai dengan perintah yang diterima dari *web service*.

A. Realisasi Perangkat Keras

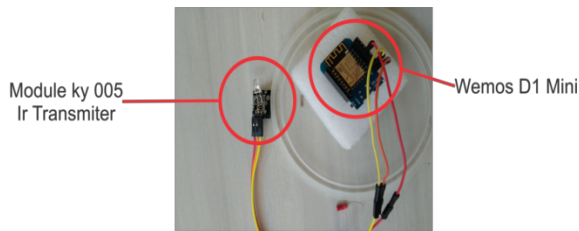
Realisasi perangkat keras dari sistem dapat dilihat pada Gambar 11 dan Gambar 12.



Gambar 11. Realisasi perangkat dari Gambar 3

Gambar 11 merupakan realisasi dari rancangan arsitektur perangkat pada Gambar 4 yang terdiri dari wemos D1 R2, *relay*, dan modul ky 005 *infra red transmitter*. Fungsi dari masing-masing perangkat tersebut adalah :

1. Wemos digunakan untuk mengambil data pada *web service* yang kemudian akan dilanjutkan ke *relay* atau modul ky 005.
2. Modul ky 005 digunakan untuk mengirimkan sinyal infra merah untuk mengendalikan Tv.
3. *Relay* digunakan untuk mengendalikan lampu.



Gambar 12. Realisasi perangkat dari Gambar 4

Gambar 12 merupakan realisasi dari rancangan arsitektur perangkat pada Gambar 5 yang terdiri dari wemos D1 mini, dan modul ky 005 *infra red transmitter*. Fungsi dari masing-masing perangkat tersebut adalah :

1. Wemos digunakan untuk mengambil data pada *web service* yang kemudian akan dilanjutkan ke modul ky 005.
2. Modul ky 005 digunakan untuk mengirimkan sinyal infra merah untuk mengendalikan AC.

B. Realisasi Perangkat Lunak

Perangkat lunak yang dibuat terbagi menjadi tiga bagian yaitu pembangunan *control application*, *web service*, dan aplikasi bergerak. Untuk pembuatan *control application* bahasa yang digunakan adalah bahasa C, dan IDE yang digunakan adalah Arduino IDE.

Untuk membangun *control application* agar dapat terkoneksi ke jaringan Wi-Fi dan dapat mengakses *web service*, dibutuhkan suatu *library* tambahan. *Library* yang ditambahkan dapat dilihat pada *Source code* di bawah ini :

```
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <IRremoteESP8266.h>
```

Fungsi dari *library - library* pada *Source code* di atas adalah sebagai berikut :

- a. ESP8266WiFiMulti.h : Digunakan untuk mengakses Wi-Fi.
- b. ESP8266HTTPClient.h : Digunakan untuk mengakses *web service*.
- c. IRremoteESP8266.h : Digunakan untuk mengirimkan sinyal infra merah

Source code untuk menghubungkan Arduino dengan Jaringan Wi-Fi dapat dilihat pada *Source code* di bawah ini :

```
wm.addAP("Internet of Things","@IoT2017");
wm.run()
```

Source code di atas merupakan *Source code* untuk mendaftarkan nama dan *password* jaringan Wi-Fi ke Arduino, yang kemudian menghubungkan Arduino ke jaringan Wi-Fi yang telah didaftarkan.

Source code untuk mengambil data pada *web service* dapat dilihat pada *Source code* di bawah ini :

```
http.begin("http://172.22.99.222/Web-Service-Digital-Home-Assistant/public/index.php/api/simpan/2");
data = http.GET();
```

Fungsi dari *Source code* di atas adalah sebagai berikut :

- a. `http.begin` : digunakan untuk mendaftarkan URL yang akan diakses.
- b. `http.GET` : digunakan untuk mengakses URL dengan *method* GET.

Source code untuk mengendalikan Lampu dan mengirimkan sinyal infra merah dapat dilihat pada *Source code* di bawah ini :

```
digitalWrite(D7, HIGH); //Lampu
irsend.sendRaw(tv_power,
sizeof(tv_power)/sizeof(int), 38); //IR
```

Sementara untuk pembuatan *web service*, bahasa yang digunakan adalah bahasa PHP dengan menggunakan *framework* Laravel. *Source code* *web service* yang digunakan untuk menerima data dari *mobile application* dapat dilihat pada *Source code* di bawah ini :

```
public function store(Request $request)
{
    $cek =
    Aktivitas::where("perintah","=", $request->kata)->first();
    if($cek){
        $dt = new ListModel;
        $dt->id_alat = $cek->id_alat;
        $dt->kata = $cek->sinyal;
        $dt->save();
        $data = "perintah $request->kata sedang diproses";
    }else{
        $data = "perintah $request->kata belum terdaftar"
    }
    return $data;
}
```

Source code *web service* yang digunakan untuk mengirimkan data dari *web service* ke Arduino dapat dilihat pada *Source code* di bawah ini :

```
public function show($id)
{
    $data =
    ListModel::where('id_alat','=', $id)->first();
    if($data){
        $perintah = $data->kata;
        $q = 'DELETE FROM List where id_alat = ? LIMIT 1';
        DB::delete($q, [$id]);
    }else{
        $perintah = '';
    }
    return $perintah;
}
```

Pada *Source code* di atas, data yang dikirimkan akan diurutkan berdasarkan data yang diterima pertama kali oleh *web service*.

Sementara itu dalam pembuatan *mobile application* bahasa yang digunakan adalah bahasa Java, dan IDE yang digunakan adalah Android Studio. Pada aplikasi bergerak yang dibuat, diimplementasikan teknologi Google Speech API, contohnya pada *Source code* dibawah ini :

```

super.onActivityResult(requestCode,
resultCode, data);
StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll(
).build();
StrictMode.setThreadPolicy(policy);

switch (requestCode) {
case RESULT_Speech: {
if (resultCode == RESULT_OK && data != null)
{
final ArrayList<String> text =
data.getStringArrayListExtra(RecognizerInten
t.EXTRA_RESULTS);
System.out.println(text.get(0));
try {
k.start();
notif = k.send(text.get(0));
k.stop();
} catch (IOException e) {
notif = "Server Tidak Di Temukan";
e.printStackTrace();
}
}
break;
}
}
}
    
```

Pada *Source code* di atas setelah proses pengenalan kata dengan menggunakan *Google Speech API* selesai, hasilnya akan dikirimkan ke *web service* dengan menggunakan kelas *Koneksi*. Kemudian *feed back* dari *web service* akan diubah ke dalam bentuk *speech* untuk diberitahukan ke pengguna. *Source code text to speech* pada *mobile application* dapat dilihat pada *Source code* di bawah ini :

```

if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {
feedback.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null, null);
}
    
```

C. Hasil Survey dan Pengujian

Untuk menentukan kata-kata yang digunakan di dalam sistem, telah dilakukan *survey* terhadap 30 mahasiswa Teknik Informatika Universitas Mataram. Hasil dari *survey* tersebut didapatkan tiga buah *table* yang kemudian dimasukkan ke dalam *database* dan digunakan sebagai perintah untuk melakukan pengendalian lampu, TV dan AC. Tabel I merupakan salah satu *table* yang dihasilkan dari *survey* untuk perintah yang dilakukan pada AC.

Berdasarkan *survey* yang disajikan pada Tabel I, kemudian dilakukan pengujian *Google Speech API* untuk menguji pengenalan kata oleh 5 orang pengguna sistem terhadap daftar kata yang digunakan. Hasil dari pengujian kesesuaian perintah dengan hasil *Google Speech API* terhadap 5 orang pengguna tersebut dapat dilihat pada Tabel II.

TABEL I. PERINTAH PADA AC

Fungsi	Perintah	Jumlah
Menghidupkan AC	Hidupkan AC	5 Orang
	AC Hidup	14 Orang
	Nyalakan AC	2 Orang
	AC Nyala	9 Orang
Mematikan AC	Matikan AC	7 Orang
	AC Mati	23 Orang
Menaikkan Suhu AC	Naikkan Suhu AC	7 Orang
	Tambah Suhu	3 Orang
	AC Naik	7 Orang
	Dingin	1 Orang
	Naikkan Suhu	4 Orang
Menurunkan Suhu AC	Suhu Naik	8 Orang
	Turunkan Suhu AC	7 Orang
	Turunkan Suhu	3 Orang
	AC Turun	7 Orang
	Panas	1 Orang
	Suhu Turun	4 Orang
	Turun Suhu	8 Orang

TABEL II. PENGUJIAN KESESUAIAN KONVERSI PERINTAH UNTUK AC

Fungsi	Perintah	Peserta				
		1	2	3	4	5
Menghidupkan AC	Hidupkan AC	√	√	√	√	√
	AC Hidup	√	√	√	√	√
	Nyalakan AC	√	√	√	√	√
	AC Nyala	√	√	√	√	√
Mematikan AC	Matikan AC	√	√	√	√	x
	AC Mati	√	√	√	√	√
Menaikkan Suhu AC	Naikkan Suhu AC	√	x	x	√	√
	Tambah Suhu	√	√	√	x	√
	AC Naik	√	√	√	√	√
	Dingin	√	√	√	√	√
	Naikkan Suhu	√	√	√	√	√
Menurunkan Suhu AC	Suhu Naik	√	√	√	√	√
	Turunkan Suhu AC	√	√	√	√	√
	Turunkan Suhu	√	√	√	√	x
	AC Turun	√	√	x	√	√
	Panas	x	√	√	√	√
	Suhu Turun	√	√	√	√	√
	Turun Suhu	√	√	√	x	√

Dari hasil pengujian terhadap konversi perintah suara untuk lampu, TV, dan AC didapatkan hasil bahwa akurasi *Google Speech API* dalam mengkonversi perintah suara untuk lampu, TV dan AC mencapai 94,36%.

Sementara Hasil dari pengujian waktu konversi perintah suara ke teks dengan *Google Speech API* terhadap 5 orang pengguna dapat dilihat pada Tabel III. Dari hasil pengujian pada Tabel III yaitu pengujian waktu konversi perintah untuk AC dan ditambah dengan pengujian untuk lampu dan TV didapatkan hasil bahwa rata-rata waktu yang dibutuhkan pada saat proses konversi suara yaitu 4.00 detik.

TABEL III. PENGUJIAN WAKTU KONVERSI PERINTAH UNTUK AC

Fungsi	Perintah	Peserta				
		1*	2*	3*	4*	5*
Menghidupkan AC	Hidupkan AC	3.83	4.21	4.61	3.43	3.64
	AC Hidup	3.9	3.66	3.93	4.11	4.13
	Nyalakan AC	4.7	4.32	3.56	3.55	3.69
	AC Nyala	4.71	4.32	3.56	3.55	4.05
Mematikan AC	Matikan AC	4.44	4.01	4.01	3.57	3.77
	AC Mati	4.64	3.64	4.05	3.75	4.22
Menaikkan Suhu AC	Naikkan Suhu AC	4.71	3.54	3.55	3.53	4.21
	Tambah Suhu	3.63	4.51	4.61	3.43	3.64
	AC Naik	4.05	3.99	4	3.43	3.76
	Dingin	4.61	3.54	4.55	3.59	3.63
	Naikkan Suhu	4.71	4.32	3.56	3.55	4.05
	Suhu Naik	4.55	4.27	3.92	3.67	4.28
Menurunkan Suhu AC	Turunkan Suhu AC	3.56	4.01	4.12	3.59	4.01
	Turunkan Suhu	3.9	4.51	4.61	3.43	3.64
	AC Turun	4.61	3.54	4.68	3.59	4.01
	Panas	3.95	4.55	3.77	4.24	3.91
	Suhu Turun	4.64	3.64	4.05	3.75	4.22
	Turun Suhu	3.56	4.01	4.12	3.59	4.01

*satuhan detik

Selain itu pengujian yang dilakukan yaitu :

- Pengujian web *service* untuk memperlihatkan server dalam menerima dan mengembalikan data, di mana pengujian dilakukan menggunakan aplikasi Postman.
- Perekaman Sinyal *Remote* TV dan AC. Pada tahap ini telah dilakukan proses perekaman sinyal infra merah yang dikirimkan melalui *remote* TV dan AC. Pada tahap ini Raspberry Pi dihubungkan dengan modul Ky-002 *infra* merah *receiver* untuk perekaman sinyal. Proses perekaman dilakukan dengan cara menekan tombol pada *remote* secara satu per satu dengan mengarahkan *remote* ke modul Ky-002, kemudian hasil dari rekaman tersebut disimpan dalam bentuk *file*
- Pengujian Arduino (Wemos). Pada pengujian Arduino dilakukan pengujian terhadap kesesuaian sinyal yang dikirimkan dari Arduino dengan sinyal yang dikirimkan dari *remote* TV dan AC. Pada pengujian ini digunakan Raspberry Pi sebagai penerima sinyal dari masing-masing alat. Kemudian *row* dari masing-masing sinyal yang diterima oleh Raspberry Pi akan ditampilkan pada layar untuk dilakukannya pengecekan kesesuaian sinyal antara sinyal yang dikirimkan Arduino dan sinyal yang dikirimkan *remote* TV dan AC. Hasil dari pengujian tersebut didapatkan bahwa kesesuaian sinyal yang dikirimkan tidak mengalami kesalahan
- Pengujian Fungsional Sistem. Pada pengujian fungsional sistem dilakukan pengujian terhadap fungsional sistem yang terdiri dari pengendalian Lampu, TV dan AC. Pengujian dilakukan dengan dua

cara yaitu dengan memberikan perintah secara satu per satu, di mana perintah akan diberikan lagi setelah perintah sebelumnya selesai dilaksanakan. Dan cara kedua yaitu dengan memberikan tiga buah perintah secara berturutan, di mana perintah akan diberikan tanpa menunggu perintah sebelumnya selesai dilaksanakan namun perintah yang dimasukkan dapat dikenali oleh sistem.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pembahasannya dapat ditarik beberapa kesimpulan, yaitu:

- Dengan menggunakan Arsitektur IoT, peralatan elektronika yang ada di rumah dapat dihubungkan dengan Raspberry Pi dan *smart phone*.
- Dengan menggunakan *Google Speech* API, aplikasi *smart phone* dapat mengolah perintah suara menjadi teks sebagai *input* untuk sistem *digital home assistant*.
- Akurasi *Google Speech* API dalam mengkonversi perintah suara untuk lampu, TV dan AC mencapai 94,36% dengan rata-rata waktu mencapai 4 detik.
- Waktu rata-rata yang dibutuhkan sistem untuk mengerjakan satu perintah hingga perintah selesai dikerjakan adalah 5.32 detik.
- Waktu rata-rata yang dibutuhkan sistem untuk mengerjakan tiga perintah hingga seluruh perintah selesai dikerjakan adalah 17.18 detik.
- Proses pengendalian Lampu, TV, dan AC pada sistem *digital home assistant* mencapai keberhasilan 90%.

Mengingat bahwa sistem *digital home assistant* ini menggunakan bantuan *Google Speech* API dalam pengenalan perintah suaranya, sehingga membutuhkan koneksi internet. Agar sistem dapat berjalan tanpa koneksi internet, pada penelitian selanjutnya perlu dibuat sistem pengenalan suara sendiri yang dipasangkan pada Raspberry Pi.

DAFTAR PUSTAKA

- M. Al-Amin, dan S.Z. Aman, "Design of an Intelligent Home Assistant," Dept. of Electronics and Communication Engineering Institute of Science and Technology: Dhaka, Bangladesh, 2016.
- D. Febriansyah, H. Kuswara, dan A. Saputra, "Alat Kendali Lampu Rumah Menggunakan Bluetooth Berbasis Android," *Jurnal Teknologi dan Informatika (Teknomatika)*, vol. 4, no. 1, pp. 273-286, 2014.
- S.A Nugroho, "Aplikasi Mobile Untuk pengendalian Lampu Berbasis Android," Sekolah Tinggi Manajemen Informatika dan Komputer Amikom: Yogyakarta, 2014.
- D. Aryanta, A.M. Jaya, dan D.A. Ramadhan, "Perancangan Dan Implementasi Prototype Kendali Peralatan Listrik Melalui Internet," *Jurnal Teknik Elektro Institut Teknologi Nasional*, vol. 2, no. 2, pp. 75-89, 2014.
- M.A. Latief, "Pengontrolan Alat Elektronika Melalui Media Wi-Fi Berbasis Raspberry Pi," Skripsi Program Studi Teknik

- Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga: Yogyakarta, 2015.
- [6] F.G. Aditya, Hafidudun, dan A.G. Permana, A.G., “Analisa dan Perancangan Prototype Smart Home dengan Sistem Client Server Berbasis Platform Android Melalui Koneksi Wireless,” Skripsi Teknik Telekomunikasi Universitas Telkom: Bandung, 2015.
- [7] R.F. Giant, Darjat, dan Sudjadi, “Perancangan Aplikasi Pemantau dan Pengendali Piranti Elektronik pada Ruangan Berbasis WEB,” Skripsi Jurusan Teknik Elektro Universitas Diponegoro: Semarang, 2015.
- [8] P.A. Budi, E. Haritman, dan M. Rahayu, “Pengontrolan Alat Elektronika Melalui Media Wi-Fi Berbasis Raspberry Pi,” *Jurnal Program Studi Teknik Elektro FPTK UPI*, vol. 13, no. 1, pp. 35-42, 2014.