

Implementasi *Azure Cognitive Service* untuk Aplikasi *Speech Recognition*

Rivardi B. Ruspandi¹⁾, Sherwin Sompie²⁾, Feisy Kambey³⁾

Teknik Elektro Fakultas Teknik, Universitas Sam Ratulangi, Jl. Kampus Bahu-Unsrat Manado, 95115

E-mail: rivardi.ruspandi@gmail.com¹⁾, aldo@unsrat.ac.id²⁾, feisykambey@unsrat.ac.id³⁾

Abstrak – Perkembangan ilmu pengetahuan dan teknologi saat ini semakin pesat, *speech recognition* adalah salah satu teknologi yang dapat dimanfaatkan untuk mempermudah manusia contohnya dalam hal penguncian pintu, dengan *speech recognition* kita dapat melakukan penguncian pintu hanya dengan menggunakan suara tentu dapat lebih mempermudah dan lebih efektif. Tujuan penelitian ini yaitu untuk mengimplementasi *azure cognitive service* pada aplikasi *speech recognition* untuk pengunci pintu. Penelitian ini menggunakan mikrokontroler *Raspberry Pi*, dan untuk *input* suara menggunakan sebuah mikrofon yang kemudian akan diproses dengan menggunakan *API speaker verification* yang membuat sistem dapat memverifikasi pengguna melalui suara kemudian *outputnya servo* sebagai penggerak penguncian pada pintu, yang akan membuka pengunci pintu jika *input* suaranya berhasil diverifikasi oleh sistem. Berdasarkan hasil pengujian dan analisis yang telah dilakukan, hasil yang didapatkan cukup baik pada semua parameter – parameter yang diuji. Implementasi *azure cognitive service* pada aplikasi *speech recognition* untuk pengunci pintu berjalan dengan baik ditandai dengan terbukanya pengunci pintu dalam hal ini motor *servo* berdasarkan input suara yang sesuai. Hasil dari penelitian telah berjalan sesuai dengan yang diharapkan.

Kata kunci – *API; Azure Cognitive Service; Raspberry Pi; Speaker Verification; Speech Recognition.*

Abstract – The development of science and technology is now rapidly increasing, word recognition is one of the technologies that can be used to connect humans in terms of door locking, with the *speech recognition* by using voices of some specific users, door locking can be easier and more effective. The purpose of this study is to implement *Azure Cognitive Services* in *speech recognition* applications for door locking. This study uses a *Raspberry Pi* microcontroller, and a microphone for sound input which will be processed using a verification speaker API that makes the system produce user sound profile and then output it as a door lock driver, which will unlock the door if the sound input is successfully verified by the system. Based on the results of tests and analyzes that have been carried out, the results are good on all the parameter parameters. Implementation of *Azure Cognitive Services* in *speech recognition* applications for door locking properly with the entrance, in this case, a *servo motor* based on the appropriate sound input. The results of the research have proceeded as expected.

Keywords - *API; Azure Cognitive Service; Raspberry Pi; Speaker Verification; Speech Recognition.*

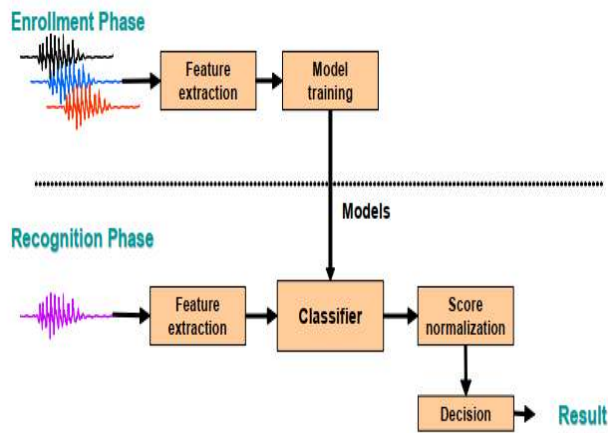
I. PENDAHULUAN

Berkembangnya teknologi dan pengetahuan pada saat ini semakin banyak memberikan kemudahan dalam hidup manusia. Dimana segala sesuatu banyak diterapkan ilmu pengetahuan dan teknologi dengan mesin ataupun alat elektronik, sehingga pekerjaan manusia dapat dipermudah tanpa harus membuang tenaga dan mempersingkat waktu. Seiring dengan perkembangan teknologi dan pengetahuan, salah satu teknologi yang dapat dimanfaatkan untuk mempermudah kita sebagai pengguna adalah *speech recognition*. *Speech Recognition* merupakan suatu proses identifikasi suara berdasarkan kata yang diucapkan dengan melalui konversi sebuah sinyal suara.

Speech recognition sendiri dapat dimanfaatkan dalam berbagai hal, salah satunya yaitu dalam penguncian pintu, dengan memanfaatkan *speech recognition* kita dapat melakukan penguncian pintu hanya dengan menggunakan perintah suara tentunya itu dapat lebih mempermudah dan lebih efektif dibanding penguncian pintu menggunakan kunci yang masih banyak digunakan sampai saat ini, itulah kenapa *speech recognition* sekarang ini perlu untuk diterapkan karena teknologinya sudah berkembang dibanding dengan menggunakan yang konvensional.

Microsoft Azure adalah sebuah platform yang menyediakan komputasi awan dan infrastruktur yang disediakan oleh *Microsoft*. *Microsoft Azure* bisa digunakan untuk membuat, mengembangkan dan menjalankan suatu aplikasi. *Azure cognitive service* merupakan salah satu layanan dari *Microsoft Azure* yang dapat digunakan untuk membantu pembuatan dalam aplikasi.

Berdasarkan latar belakang di atas, pada penelitian ini penulis ingin menggunakan *speech recognition* dalam hal keamanan yaitu untuk pengunci suatu pintu, dan memanfaatkan layanan *azure cognitive service* dari *Microsoft* untuk diterapkan. Diharapkan dengan memanfaatkan penerapan layanan ini dapat membuat keamanan dalam penguncian pintu lebih baik lagi karena terdapat juga fitur *speaker verification* yang ada pada *azure cognitive service* sehingga akan membuat keamanannya semakin meningkat lagi dan bukan hanya itu saja tetapi diharapkan bisa lebih efektif dan praktis dibandingkan dengan menggunakan pengunci pintu yang konvensional yang umumnya masih banyak digunakan sekarang.

Gambar 1. Skema *Speaker Verification*

A. *Speech Recognition*

Speech recognition suatu proses untuk mengenali seseorang dengan mengenali ucapan dari orang tersebut. *Speech recognition* merupakan teknik dimana perangkat akan mengenali masukan berupa suara dan setelah itu perangkat melakukan respon yang sesuai dengan masukan suara tersebut.^[1]

Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan cara digitalisasi kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat. Kata-kata yang diucapkan diubah bentuknya menjadi sinyal digital dengan cara mengubah gelombang suara menjadi sekumpulan angka yang kemudian disesuaikan dengan kode-kode tertentu untuk mengidentifikasi kata-kata tersebut.

Hasil dari identifikasi kata yang diucapkan dapat ditampilkan dalam bentuk tulisan atau dapat dibaca oleh perangkat teknologi sebagai sebuah perintah untuk melakukan suatu pekerjaan. Misalnya sebuah sistem komputer dapat menerjemahkan kata-kata yang diucapkan dari pengguna menjadi sebuah perintah yang kemudian oleh sistem akan dijalankan. *Speech recognition* memiliki empat tahapan umum dalam prosesnya yaitu:

- 1) Tahap penerimaan masukan
- 2) Tahap ekstraksi
- 3) Tahap perbandingan
- 4) Tahap validasi identitas pengguna

Secara umum, *speech recognition* memproses sinyal suara yang masuk dan menyimpannya dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan dianalisa dengan membandingkannya dengan template suara pada database sistem. Sebelumnya, data suara masukan dipilah-pilah dan diproses satu per satu berdasarkan urutannya.

Dalam penelitian ini *speech recognition* digunakan untuk menjalankan perintah penguncian pintu melalui sebuah mikrofon usb yang kemudian dihubungkan dengan *raspberry pi* dan sebuah *micro servo* sebagai motor penggerak untuk pengunci pintu.

Gambar 2. *Raspberry Pi 3 Model B*

B. *Azure Cognitive Service*

Azure cognitive service adalah *APIs*, *SDKs*, dan layanan-layanan yang tersedia untuk membantu para pengembang yang tidak memiliki keahlian dan pengetahuan secara mendalam tentang *AI* atau data dalam membuat aplikasi cerdas.^[2]

Microsoft cognitive service bisa memperluas pembelajaran tentang *application programming interface* sehingga semakin berkembang dan memungkinkan pengembang bisa menambah fitur-fitur cerdas seperti pendeteksi emosi dan video, pengenalan wajah dan penglihatan kedalam aplikasi. Tujuan dari *azure cognitive service* adalah untuk meningkatkan pengalaman komputasi dan produktivitas dengan dibantu oleh sistem yang semakin cerdas. Saat ini *azure cognitive service* dapat dikelompokkan dalam 5 kategori yaitu *vision*, *knowledge*, *language*, *speech*, dan *search* dimana masing-masing dari kategori yang terdapat pada *azure cognitive service* memiliki fungsinya masing-masing. Pada penelitian ini memanfaatkan *azure cognitive service* pada layanan *speech*.

Application Programming Interface (API) adalah sebuah teknologi untuk memfasilitasi pertukaran informasi atau data pada perangkat lunak.^[3] *API* dapat digambarkan sebagai pembawa pesan yang menerima permintaan pengguna dan memberikan respon sesuai dengan permintaan tersebut. *Azure cognitive service API* adalah sebuah *framework* yang dikembangkan oleh *Microsoft* untuk digunakan sesuai dengan kebutuhan yang diperlukan oleh pengembang. Dalam *azure cognitive service* terdapat lima kategori *API* yang bisa digunakan pengembang sesuai kebutuhan yakni *vision*, *knowledge*, *language*, *speech* dan *search*. Dengan tujuan untuk mempercepat proses *development*.

Pada gambar 1 merupakan skema dari *speaker verification* dimana terdapat dua tahap yaitu *enrollment* atau *training* dan *recognition* atau *testing*. Pada tahap *enrollment*, sinyal suara yang masuk akan melewati proses ekstraksi kemudian ekstraksi dari sampel suara tersebut akan dibuatkan sebuah model yang merepresentasikan karakteristik khusus dari sinyal suara itu.

Pada tahap *recognition* suara yang masuk juga akan melewati proses ekstraksi setelah itu ke proses *classifier* dimana pada proses *classifier* suara yang masuk akan dibandingkan dengan model suara saat *training* pada tahap *enrollment* setelahnya proses perbandingan akan menghasilkan *score* lalu *score* tersebut akan dinormalisasi untuk menentukan *score* pada proses *decision*. Sistem akan membandingkan *score* yang dinormalisasi tadi untuk menentukan suara *testing* diterima atau ditolak.

C. Python

Python merupakan salah satu bahasa pemrograman yang mulai populer digunakan dalam pengembangan web, GUI desktop, maupun pengembangan perangkat lunak. Python memiliki sifat *open-source* dan pemrograman ini tidak menggunakan *compiler*, python dapat digunakan diberbagai sistem operasi diantaranya *linux*, *windows*, *machintosh*, dll.

Python menjadi bahasa resmi yang terintegrasi dalam Raspberry Pi. Kata “Pi” pada Raspberry Pi merupakan slang yang merujuk pada “Python”. Oleh karenanya, tepat dikatakan bahwa Python adalah bahasa natural Raspberry Pi.^[4] Pada penelitian ini penulis menggunakan pemrograman python dalam pengembangan sistem.

D. Aplikasi

Menurut Kamus Komputer Eksekutif aplikasi adalah program yang membentuk bagian tugas pemakai dan ditulis oleh pemakai. Hal ini berbeda dengan program tujuan umum yang digunakan untuk mengelola operasi seluruh *system*.^[5]

Menurut Jogiyanto aplikasi adalah penggunaan dalam suatu komputer, instruksi atau pernyataan yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.^[6]

Menurut Kamus Besar Bahasa Indonesia aplikasi adalah pengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu yang diterapkan dari suatu perancangan sistem.^[7]

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CEO_N) GPIO08	24
25	Ground	(SPI_CEO_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Gambar 3. Raspberry Pi 3 GPIO

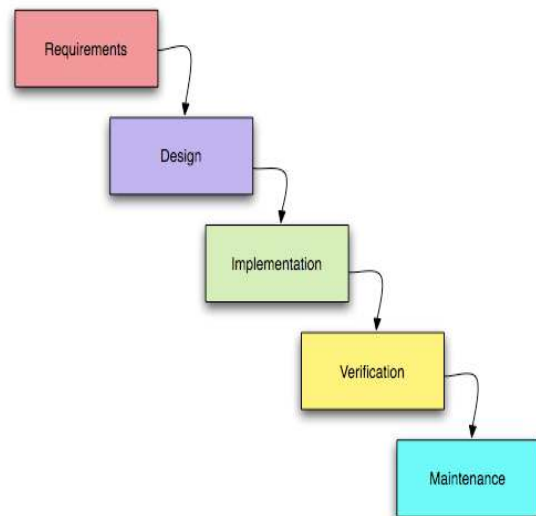
E. Raspberry Pi

Raspberry Pi adalah sebuah komputer mini (*Single Board Computer*) yang dikembangkan oleh *Raspberry Pi Foundation*, *Raspberry Pi* menggunakan sistem operasi berbasis *Linux*.^[8] *Raspberry* memiliki ukuran kira-kira hanya sebesar dompet dimana *raspi* memiliki *input* dan *output digital port*. *Raspberry Pi* juga bisa digunakan untuk pengontrolan lebih dari satu *device*, baik jarak dekat ataupun jarak jauh. Untuk pengontrolan unit *device* yang akan dikontrol, *Raspberry Pi* bisa menggunakan bahasa pemrograman namun yang paling banyak digunakan adalah *python* sebagai bahasa pemrogramannya. Bentuk dari *Raspberry Pi 3 Model B* dapat dilihat pada gambar 2. Berikut ini adalah arsitektur dari sebuah *Raspberry Pi 3 model B* yang digunakan pada penelitian ini.

- 1) *Broadcom BCM2837*
- 2) *HDMI Output*
- 3) *CSI connector camera*
- 4) *Ethernet Output*
- 5) *USB 2.0*
- 6) *Audio Output*
- 7) *DSI Display connector*
- 8) *SD card slot*
- 9) *Micro USB power*
- 10) *GPIO Header*

GPIO (General Purpose Input/Output) header pada *Raspberry Pi 3* terdapat 40 pin *GPIO* seperti pada gambar 3 dimana beberapa pin tersebut memiliki fungsinya masing-masing.

- 1) Sumber tegangan: 3.3v DC, 5v DC 4 pin warna merah.
- 2) *General purpose digital inputs/outputs* 17 pin warna hijau.
- 3) Pin *GND* atau *Ground* 8 pin warna hitam.
- 4) *I²C* 2 pin warna biru
- 5) *EEPROM* 2 pin warna kuning.
- 6) *SPI* 5 pin warna ungu.
- 7) *UART*: 2 pin warna jingga.



Gambar 4. Metode Waterfall

TABEL I. BAHAN DAN ALAT PENELITIAN

No.	Langkah-langkah Aktivitas Riset	Alat dan Bahan yang Digunakan
1.	Analisis kebutuhan	Laptop
2.	Perencanaan	1) Laptop 2) Alat tulis menulis
3.	Perancangan <i>software</i>	1) Laptop 2) <i>Raspberry Pi 3 Model B</i> 3) <i>Etcher</i> 4) <i>Raspbian Stretch</i> 5) <i>Python</i> 6) <i>Advanced IP Scanner</i> 7) <i>VNC Connect</i> 8) <i>ClickCharts Diagram & Flowchart Software</i> 9) <i>Audacity</i> 10) <i>Azure cognitive service API</i> 11) Koneksi internet
4.	Perancangan <i>hardware</i>	1) Laptop 2) <i>Raspberry Pi 3 Model B</i> 3) <i>Micro servo SG-90</i> 4) Mikrofon <i>USB</i> 5) <i>Fritzing</i>
5.	Pengujian sistem	1) Laptop 2) <i>Azure cognitive service API</i> 3) Koneksi internet 4) <i>SpeedTest</i> Internet 5) <i>Python</i> 6) <i>Sound Meter</i> 7) <i>Raspberry Pi 3 Model B</i> 8) <i>Powerbank</i> 9) <i>Micro SD Card</i> 10) <i>Breadboard</i> 11) Kabel <i>jumper</i> 12) Solder 13) Lampu <i>LED</i> merah dan hijau 14) <i>Push button</i> 15) <i>Resistor</i> 16) <i>Micro servo SG-90</i> 17) Mikrofon <i>USB</i> 18) Maket rumah

F. *Waterfall*

Metode *Waterfall* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.^[9]

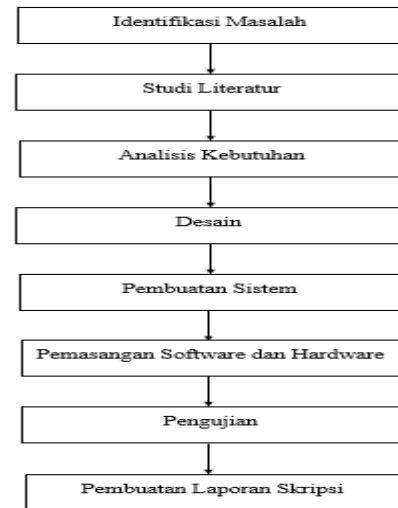
Tahapan-tahapan pada *Waterfall* adalah sebagai berikut:

1) *Requirements*

Pada tahap ini, dilakukan penganalisaan dan pengumpulan kebutuhan sistem, pengembang sistem diperlukan suatu komunikasi yang bertujuan untuk memahami aplikasi yang diharapkan dan batasannya.

2) *Design*

Tahap ini dilakukan sebelum melakukan pengkodean. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana hasilnya.



Gambar 5. Kerangka Berpikir

3) *Implementation*

Pada tahap ini dilakukan pemrograman. Pengkodean merupakan proses menterjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman.

4) *Verification*

Pada tahap ini dilakukan proses pengujian pada program perangkat lunak, baik Pengujian *software* maupun *hardware* untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.

5) *Maintenance*

Ini merupakan tahap terakhir dalam metode *waterfall*. *Software* yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya.

G. *Mikrofon*

Mikrofon adalah alat untuk mengubah gelombang bunyi ke dalam isyarat listrik untuk penyiaran atau perekaman bunyi, seperti pidato, musik; alat pengeras suara.^[10]

Mikrofon merupakan sebuah alat yang dapat mengubah atau mengkonversikan gelombang suara ke sinyal audio, mikrofon yang digunakan pada penelitian ini yaitu mikrofon dengan sambungan *USB* sebagai alat bantu untuk menangkap gelombang suara yang akan dipasang pada *raspberry pi* sehingga nantinya sistem dapat mengenali suara pengguna yang masuk dari mikrofon.

II. METODOLOGI PENELITIAN

A. *Kerangka Berpikir*

Kerangka penelitian yang akan dibuat dalam penelitian ini terdapat pada gambar 5.

B. *Tempat dan Waktu Penelitian*

Penelitian ini dilaksanakan di Universitas Sam Ratulangi Fakultas Teknik Manado pada bulan Mei 2018 sampai Oktober 2018.

C. Bahan dan Alat Penelitian

Metode Bahan dan alat penelitian yang digunakan dalam penelitian ini terdapat pada tabel I.

Adapun spesifikasi alat dan bahan penelitian yang digunakan adalah:

- 1) Laptop yang digunakan adalah Laptop *ASUS N46VZ* dengan spesifikasi:
 - a. *Processor Intel Core i7 3630QM*
 - b. *RAM 8 GB*
 - c. *Sistem Operasi Windows 10 64 bit*
- 2) *Raspberry Pi* yang digunakan adalah *Raspberry Pi 3 Model B* dengan spesifikasi:
 - a. *Quad Core 1.2GHz Broadcom BCM2837 64bit CPU*
 - b. *1GB RAM*
 - c. *100 Base Ethernet*
 - d. *40-pin extended GPIO*
 - e. *4 USB 2 ports*
 - f. *HDMI*
 - g. *CSI camera port for connecting a Raspberry Pi camera*
 - h. *DSI display port for connecting a Raspberry Pi touchscreen display*
 - i. *Micro SD port for loading your operating system and storing data*
 - j. *Micro USB power source up to 2.5A*
- 3) Mikrofon *USB* digunakan sebagai alat untuk menginput suara dari *user* dengan spesifikasi:
- 4) *Etcher Portable* versi 1.4.4 untuk menginstal sistem operasi *Raspbian* pada *Raspberry Pi*
- 5) *Python* versi 2.7 sebagai bahasa pemrograman yang digunakan dalam pembuatan sistem.
- 6) *Advanced IP Scanner* versi 2.5.3646 digunakan untuk mengetahui *IP* dari *Raspberry Pi* yang terhubung ke internet lalu kemudian *IP* tersebut akan dimasukkan ke *VNC viewer* pada laptop sebagai pengganti monitor.
- 7) *Audacity* versi 2.1.0 adalah *software* yang membantu untuk merekam suara dan dapat menyimpan format suara sesuai yang diinginkan.
- 8) *VNC Connect* versi 6.3.1 adalah *software* untuk *remote control* yang memungkinkan untuk mengontrol *Raspberry Pi* pada laptop melalui koneksi *network*.
- 9) *Fritzing* versi 0.9.3b adalah salah satu *software* yang digunakan untuk membantu dalam merancang dan mendesain berbagai macam peralatan elektronika yang akan digunakan.

D. Metode Pengembangan Aplikasi

Dalam penelitian ini, aplikasi yang akan dibangun melalui tahap-tahap model *Waterfall*. *Waterfall* terdiri dari beberapa tahap, yaitu *requirements*, *design*, *implementation*, *verification*, dan *maintenance*.

1) Requirements

Pada tahap ini, peneliti akan melakukan analisa tentang kebutuhan sistem yang nantinya akan dibangun sesuai dengan yang diminta.

2) Design

Pada tahap ini peneliti akan membuat perancangan tentang sistem yang akan dibangun untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana hasilnya.

3) Implementation

Pada tahap ini peneliti akan memulai proses pembuatan sistem. Pertama peneliti akan membuat akun *Microsoft Azure* untuk mendapatkan *API* dari *azure cognitive service*. Setelah itu peneliti akan membuat aplikasi *speech recognition* untuk penguncian pintu. Setelah itu peneliti akan mengimplementasikan *API azure cognitive service* sebelumnya dengan *speech recognition* untuk dapat melakukan penguncian pintu.

4) Verification

Setelah aplikasi tersebut selesai dibuat di *Raspberry Pi*, peneliti kemudian akan memasang alat-alat pendukung yang telah direncanakan sebelumnya beserta aplikasi dalam sebuah maket rumah. Setelah semua terpasang peneliti akan melakukan pengujian.

5) Maintenance

Pada tahap ini peneliti akan memperbaiki kesalahan-kesalahan sistem yang tidak ditemukan pada tahap sebelumnya.

III. HASIL DAN PEMBAHASAN

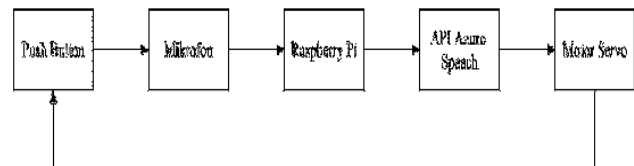
A. Blok Diagram

Blok diagram sistem yang akan dibangun dapat dilihat pada gambar 6. Pada gambar 6 dapat dilihat cara kerja dari sistem *speech recognition* untuk pengunci pintu. *Push button* berfungsi sebagai tombol untuk mengaktifkan mikrofon untuk menangkap suara dari user, mikrofon menerima suara dan mengkonversikannya ke sinyal digital kemudian diteruskan dan diproses oleh *raspberry pi*.

B. Perancangan Hardware

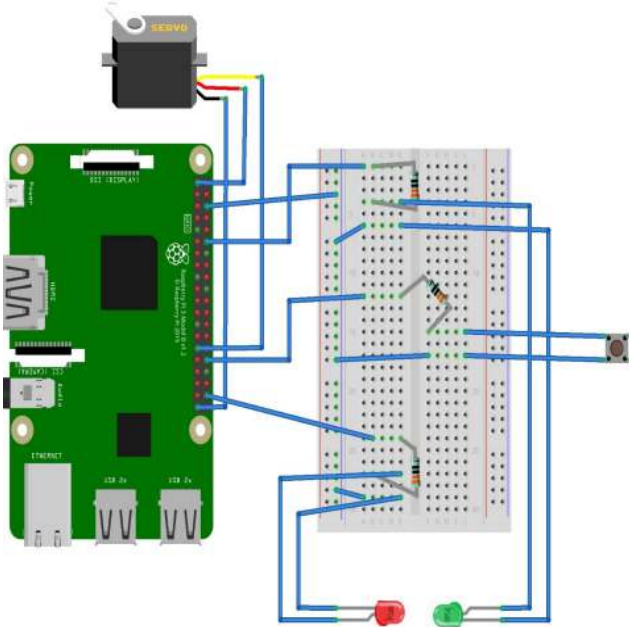
Berikut merupakan rancangan hardware pada penelitian dengan menggunakan skema *fritzing* agar dapat mempermudah proses pembuatan dalam penelitian ini.

Pada penelitian ini perancangan hardware pada gambar 7, menggunakan beberapa alat elektronika seperti *raspberry pi* 3 model B sebagai mini komputer yang digunakan sebagai alat untuk menjalankan aplikasi pada penelitian ini, kemudian menggunakan *servo SG-90* dimana untuk positif dihubungkan pada *power 3.3V* di *raspberry pi* dan untuk signal dihubungkan pada *GPIO 5*, *servo* digunakan sebagai alat pengunci pintu. Pada gambar juga terdapat sebuah *breadboard* yang digunakan sebagai papan rangkaian untuk mencoba rangkaian elektronik pada penelitian ini.

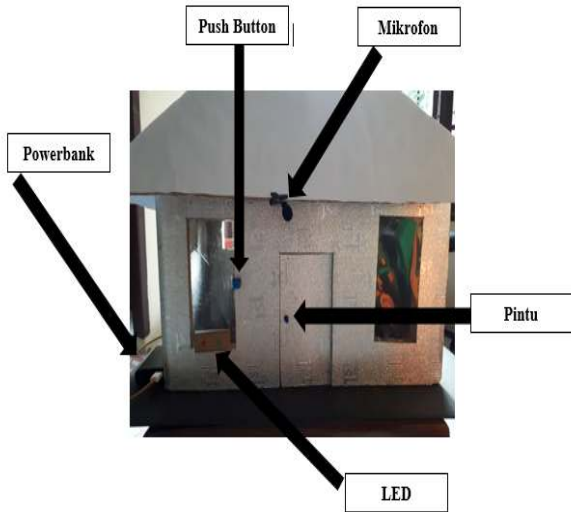


Gambar 6. Blok Diagram

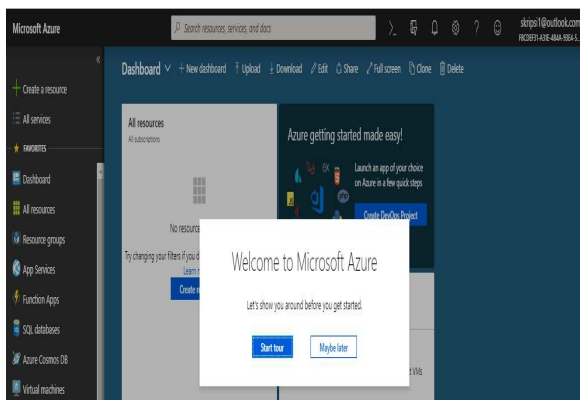
Hello skripsi1@outlook.com ([Log out](#))



Gambar 7. Rangkaian *Hardware* Menggunakan *Fritzing*



Gambar 8. Maket Rumah Tampak Depan



Gambar 9. *Dashboard Microsoft Azure*

Use speech to identify and verify individual speakers
20,000 transactions per month, 20 per minute.

[Quick-start Guide >](#)

Endpoint: <https://westus.api.cognitive.microsoft.com/spid/v1.0>

Key 1: 6f147fe50c0d401bbdedb42df7260698

Key 2: 3c68249541f44c2c8596579e38f03320

Gambar 10. *API Azure Cognitive Service*

```
File Edit Tabs Help
pi@raspberrypi:~/azure/verification $ python3 CreateProfile.py 6f147fe50c0d401bbdedb42df7260698
Profile ID = c25b0f51-49ae-48c4-8040-0a798523b79b
pi@raspberrypi:~/azure/verification $
```

Gambar 11. *Create User Profile*

```
File Edit Tabs Help
pi@raspberrypi:~/azure/verification $ python3 PrintAllProfiles.py 6f147fe50c0d401bbdedb42df7260698
Profile ID, Locale, Enrollments Count, Remaining Enrollments Count, Created Date Time, Last Action Date
Time, Enrollment Status
c25b0f51-49ae-48c4-8040-0a798523b79b, en-us, 0, 3, 2018-10-02T06:06:43.538Z, 2018-10-02T06:06:43.538Z,
Enrolling
pi@raspberrypi:~/azure/verification $
```

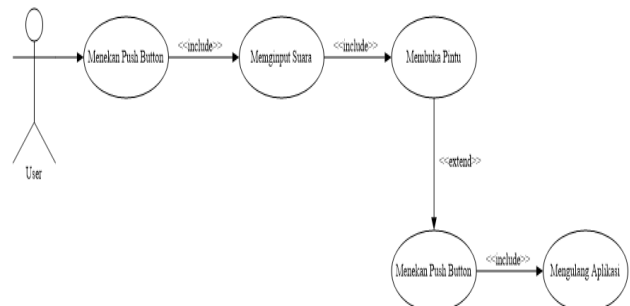
Gambar 12. *Print User Profile*

```
File Edit Tabs Help
pi@raspberrypi:~/azure/verification $ python3 EnrollProfile.py 6f147fe50c0d401bbdedb42df7260698 c25b0f51-49ae-48c4-8040-0a798523b79b /home/pi/azure/train/ivan1.wav
Enrollments Completed = 1
Remaining Enrollments = 2
Enrollment Status = Enrolling
Enrollment Phrase = my voice is my passport verify me
pi@raspberrypi:~/azure/verification $ python3 EnrollProfile.py 6f147fe50c0d401bbdedb42df7260698 c25b0f51-49ae-48c4-8040-0a798523b79b /home/pi/azure/train/ivan2.wav
Enrollments Completed = 2
Remaining Enrollments = 1
Enrollment Status = Enrolling
Enrollment Phrase = my voice is my passport verify me
pi@raspberrypi:~/azure/verification $ python3 EnrollProfile.py 6f147fe50c0d401bbdedb42df7260698 c25b0f51-49ae-48c4-8040-0a798523b79b /home/pi/azure/train/ivan3.wav
Enrollments Completed = 3
Remaining Enrollments = 0
Enrollment Status = Enrolled
Enrollment Phrase = my voice is my passport verify me
pi@raspberrypi:~/azure/verification $
```

Gambar 13. *Enroll User Profile*

```
File Edit Tabs Help
pi@raspberrypi:~/azure/verification $ python3 VerifyFile.py 6f147fe50c0d401bbdedb42df7260698 /home/pi/azure/test/ivantest.wav
Verification Result = Accept
Confidence = Normal
pi@raspberrypi:~/azure/verification $
```

Gambar 14. *Test User Profile*



Gambar 15. *Use case Diagram Aplikasi*

C. Perancangan Software

Use case diagram merupakan pemodelan suatu informasi yang akan dibuat. Kegunaannya adalah untuk mengetahui fungsi dan cara kerja dari sistem. *Use case* diagram pada aplikasi *speech recognition* yang bisa dilihat pada gambar 15.

Activity diagram speech recognition dimulai dari user akan menekan *push button* lalu menginput suara aplikasi akan menerima input suara dari *user* dan mengirimkan data suara tersebut ke server *azure* dengan *API* yang kemudian di server *azure* suara *user* akan dianalisa dengan kecocokkan suara di server lalu hasil analisa akan dikirim kembali.

Setelah menerima hasilnya jika tidak cocok maka *user* bisa mengulang kembali proses *menginput* suara, tapi jika cocok maka pengunci pintu akan terbuka dan *user* bisa membuka pintu tersebut.

Tahap perancangan *software* selanjutnya membuat akun *azure* terlebih dahulu pada *website azure.microsoft.com*, setelah akun berhasil dibuat akan muncul tampilan seperti pada gambar 9, langkah selanjutnya adalah mengambil *API* dari *azure cognitive service* yang ada pada *website azure*. Pada gambar 10 menunjukkan bahwa peneliti berhasil mendapatkan *API* dari *azure cognitive service*.

Tahap pembuatan profil *user* terdiri dari 4 tahapan. Pada gambar 11 merupakan tahap dalam membuat profil *user* dengan menggunakan *API* dari *azure cognitive service* yang nantinya profil tersebut akan digunakan pada tahap selanjutnya.

Pada gambar 12 merupakan tahap *print* profil *user* untuk melihat profil *ID* yang telah dibuat dan *enrollment* status nantinya.

Pada gambar 13 merupakan proses *enroll* dari suara *user*, untuk *enroll* suara *user* harus sesuai dengan format dari *azure cognitive service* yaitu *container* berjenis *WAV*, *encoding PCM* dengan *rate 16K*, *sample format 16-bit* dan menggunakan *channel mono* dan yang terakhir menggunakan salah satu frasa yang telah disediakan *azure*, dimana pada penelitian ini peneliti menggunakan salah satu frasa yaitu *my voice is my passport verify me*.

Agar bisa memasukkan rekaman suara sesuai dengan format yang diminta peneliti menggunakan *software Audacity*. Setelah selesai memasukkan suara *user* maka status *enroll* akan menjadi *enrolled*. Pada gambar 14 merupakan pengujian suara yang dicoba untuk mengetahui apakah sistem sudah dapat mengenali suara dari *user* dan hasilnya sistem sudah dapat berjalan sesuai dengan yang diharapkan.

Selanjutnya tahap pembuatan aplikasi *speech recognition* untuk pengunci pintu Pada tahap ini, peneliti memulai untuk membuat aplikasi *speech recognition* untuk pengunci pintu dengan memanfaatkan *API azure cognitive service* dan menggunakan bahasa pemrograman *python* pada *raspberry pi*.

Gambar 17 merupakan *source code* untuk menjalankan aplikasi *speech recognition* untuk pengunci pintu secara keseluruhan diantaranya terdapat pengaturan

lampu *LED*, *push button* dan motor *servo* yang digunakan dalam aplikasi.

```

main.py x audio.py x
1 import audio as a
2 import RPi.GPIO as GPIO
3 import os
4 import time
5
6 GPIO.setmode(GPIO.BCM)
7 GPIO.setwarnings(False)
8
9 GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_UP) #push button
10 GPIO.setup(18, GPIO.OUT) # led green
11 GPIO.setup(20, GPIO.OUT) # led red
12 GPIO.setup(5, GPIO.OUT) # servo SG90
13
14 p = GPIO.PWM(5, 50) # GPIO 5 for PWM 50Hzp = None
15
16 def nyala_hijau():
17     GPIO.output(18, GPIO.HIGH)
18     time.sleep(0.5)
19
20 def nyala_merah():
21     GPIO.output(20, GPIO.HIGH)
22     time.sleep(0.5)
23
24 def nyala_semua():
25     GPIO.output(18, GPIO.HIGH)
26     GPIO.output(20, GPIO.HIGH)
27     time.sleep(0.5)
28
29 def mati_hijau():
30     GPIO.output(18, GPIO.LOW)
31     time.sleep(0.5)
32
33 def mati_merah():
34     GPIO.output(20, GPIO.LOW)
35     time.sleep(0.5)
36
37 def mati_semua():
38     GPIO.output(18, GPIO.LOW)
39     GPIO.output(20, GPIO.LOW)
40     time.sleep(0.5)
41
42 def initialize():
43     nyala_semua()
44     mati_semua()
45     nyala_semua()
46     mati_semua()
47     nyala_semua()
48     mati_semua()
49
50 p.start(2.3) # Initialization
51
52 print("silahkan tekan tombol biru")
53
54
55 initialize()
56
57 status_selesai = 0
58
59 while True:
60     new_input = GPIO.input(12)
61
62     if (new_input == 0):
63         print("SILAHKAN UCAPKAN PASSWORD")
64         nyala_semua()
65         p.ChangeDutyCycle(2.3)
66
67         os.system("arecord -D plughw:1 -d 8 -f S16 -r 16000 voice_record.wav")
68
69         print("MENGANALISA SUARA....")
70
71         mati_semua()
72
73         if(a.voiceVerify()):
74             print("ANALISA SUARA --- BERHASIL ---")
75             nyala_hijau()
76             p.ChangeDutyCycle(6.3)
77         else:
78             print("ANALISA SUARA --- GAGAL ---")
79             nyala_merah()
80
81         status_selesai = 1
82
83     if (status_selesai == 1):
84         print("Tekan tombol untuk mulai lagi")
85         while True:
86             reboot_button = GPIO.input(12)
87             if (reboot_button == new_input):
88                 initialize()
89                 old_input = GPIO.input(12)
90                 status_selesai = 0
91
92         break

```

Gambar 17. Source Code main.py

TABEL II. PENGUJIAN LAMPU *LED*

No	Komponen Pengujian	Hasil yang Diharapkan	Status
1.	Lampu <i>LED</i> warna merah	Lampu menyala selama 5 detik kemudian padam	Berhasil
2.	Lampu <i>LED</i> warna hijau	Lampu menyala selama 5 detik kemudian padam	Berhasil

TABEL III. PENGUJIAN MOTOR *SERVO*

No	Komponen Pengujian	Hasil yang Diharapkan	Keterangan
1.	Motor Servo SG-90	Motor servo bergerak dan kembali ke posisi awal	Berhasil

TABEL IV. PENGUJIAN KESELURUHAN SISTEM

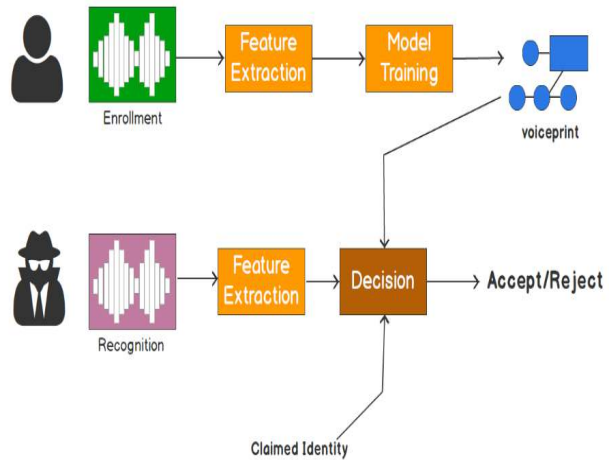
No	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan	Keterangan
1.	<i>Push Button</i>	Menekan <i>push button</i>	Melanjutkan ke proses berikutnya	Berhasil
2.	Mikrofon	Merekam suara	Menerima <i>input</i> suara <i>user</i>	Berhasil
3.	Lampu <i>LED</i> Merah	<i>LED</i> menyala dan padam	Lampu <i>LED</i> menyala dan padam sesuai dengan perancangan	Berhasil
4.	Lampu <i>LED</i> Hijau	<i>LED</i> menyala dan padam	Lampu <i>LED</i> menyala dan padam sesuai dengan perancangan	Berhasil
5.	Motor <i>Servo</i>	<i>Servo</i> bergerak sebagai pengunci pintu	Motor <i>servo</i> sebagai pengunci pintu bergerak/terbuka jika suara <i>user</i> sesuai	Berhasil

Tahap perancangan software yang terakhir adalah *speaker verification API*. *Azure cognitive service API* adalah sebuah *framework* yang dikembangkan sendiri oleh *Microsoft* untuk digunakan bagi para pengembang *software* dalam pembuatan suatu aplikasi. Dalam *azure cognitive service* terdapat lima kategori *API* yang bisa digunakan pengembang sesuai kebutuhan yakni *vision, knowledge, language, speech* dan *search*. Dengan tujuan untuk mempercepat proses *development*.

Pada penelitian ini, peneliti memanfaatkan *azure cognitive service* khususnya pada layanan *speech* dimana salah satu layanan pada kategori *speech* terdapat *speaker verification API* yang diimplementasikan pada aplikasi *speech recognition* ini. *Speaker verification* pada *azure* merupakan suatu sistem yang dapat memverifikasi *user*.

TABEL V. PENGUJIAN RESPON SISTEM

Percobaan Ke-	Waktu Respon	Status Verifikasi Suara
1	00:00:05:59	Berhasil
2	00:00:05:10	Berhasil
3	00:00:06:70	Berhasil
4	00:00:06:94	Berhasil
5	00:00:05:66	Berhasil
Rata – rata waktu respon:		00:00:05:99



Gambar 18. *Speaker Verification*

```

pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7b843ae577915 /home/pi/azure/test/Alfi.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7b843ae577915 /home/pi/azure/test/Claudia.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7b843ae577915 /home/pi/azure/test/Ribka.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
    
```

Gambar 19. Proses Verifikasi Suara *User*

Gambar 18 menunjukkan proses *speaker verification* sehingga sistem dapat memverifikasi suara dari *user*, terdapat dua tahap yaitu *enrollment* atau *training* dan *recognition* atau *testing*. Pada tahap *enrollment*, sinyal suara yang masuk akan melewati proses ekstraksi kemudian ekstraksi dari sampel suara tersebut akan dibuatkan sebuah model yang akan merepresentasikan karakteristik khusus dari sinyal suara tersebut sehingga dapat dibedakan dengan suara lain. Dalam penelitian ini, pada tahap *enrollment* peneliti memasukkan sampel suara sebanyak tiga kali agar sistem dapat melakukan *training* dari suara tersebut.

TABEL VI. PENGUJIAN JARAK KE MIKROFON

No.	Jarak ke Mikrofon	Status Verifikasi Suara
1.	10 cm	Berhasil
2.	20 cm	Berhasil
3.	30 cm	Berhasil
4.	40 cm	Berhasil
5.	50 cm	Berhasil
6.	60 cm	Berhasil
7.	70 cm	Gagal
8.	80 cm	Gagal

TABEL VII. PENGUJIAN TERHADAP KEBISINGAN

No.	Tingkat Kebisingan	Sound Meter App		Status Verifikasi Suara
		Abc Apps	Smart Tools co.	
1.	Sunyi	42 dB	47 dB	Berhasil
2.	Normal	54 dB	59 dB	Berhasil
3.	Sedikit Berisik	64 dB	61 dB	Berhasil
4.	Berisik	69 dB	67 dB	Gagal
5.	Sangat Berisik	75 dB	72 dB	Gagal

D. Pengujian Sistem

Pada tahap ini peneliti akan melakukan pengujian terhadap sistem dan cara kerja sistem, apakah sistem telah berjalan sesuai dengan perancangan sebelumnya. Pengujian dilakukan dengan cara mengevaluasi kerja sistem yang telah dibuat. Terdapat tujuh pengujian yang dilakukan pada penelitian ini yaitu seperti pada tabel berikut:

Tabel II menunjukkan pengujian yang bertujuan untuk mengetahui apakah lampu *LED* yang digunakan peneliti pada penelitian ini berfungsi atau tidak.

Tabel III menunjukkan pengujian yang bertujuan untuk mengetahui apakah motor *servo SG-90* yang digunakan peneliti pada penelitian ini berfungsi atau tidak.

Tabel IV menunjukkan pengujian yang bertujuan untuk mengetahui apakah kerja sistem secara keseluruhan berjalan sesuai dengan yang direncanakan pada perancangan sebelumnya.

Tabel V menunjukkan pengujian yang bertujuan untuk mengetahui seberapa cepat respon dari sistem untuk dapat mengenali suara dari *user*.

Tabel VI menunjukkan pengujian yang bertujuan untuk mengetahui berapa jauh jarak ideal agar sistem dapat menerima suara *user*.

TABEL VIII. PENGUJIAN TERHADAP *USER* LAIN

No.	Nama	Status yang Diharapkan	Keterangan
1.	Reynaldi	Ditolak	Berhasil
2.	Farrel	Ditolak	Berhasil
3.	Darrel	Ditolak	Berhasil
4.	Shirley	Ditolak	Berhasil
5.	Doris	Ditolak	Berhasil
6.	Oktaviano	Ditolak	Berhasil
7.	Alfi	Ditolak	Berhasil
8.	Willy	Ditolak	Berhasil
9.	Chyntia	Ditolak	Berhasil
10.	Ribka	Ditolak	Berhasil
11.	Claudia	Ditolak	Berhasil
12.	Victorin	Ditolak	Berhasil
13.	Axl	Ditolak	Berhasil
14.	Cluivert	Ditolak	Berhasil
15.	Jonathan	Ditolak	Berhasil

Tabel VII menunjukkan pengujian yang bertujuan untuk mengetahui tingkat keberhasilan sistem dalam menangkap suara dan memverifikasi suara pada tingkat kebisingan tertentu.

Dan yang terakhir tabel VIII menunjukkan pengujian yang bertujuan untuk mengetahui apakah sistem sudah berjalan sesuai dengan yang diinginkan. Sistem yang diinginkan oleh peneliti yaitu hanya suara dari peneliti yang dapat digunakan untuk membuka pengunci pintu.

Pada pengujian ini terbagi menjadi dua pengujian yaitu yang pertama adalah pengujian mengucapkan kalimat untuk membuka pintu secara langsung pada mikrofon yang disediakan dimana untuk pengujian yang pertama ini peneliti melibatkan lima orang selain peneliti untuk mencoba sistem yang sudah dibuat. Kelima orang tersebut nantinya akan mengucapkan frasa atau kalimat untuk membuka penguncian pintu sesuai dengan frasa yang telah ditentukan sebelumnya yaitu mengucapkan *my voice is my passport verify me*.

Dan untuk pengujian yang kedua adalah pengujian dimana peneliti mengambil sampel suara yang mengucapkan kalimat untuk membuka penguncian pintu yaitu *my voice is my passport verify me* dari sepuluh orang yang berbeda dari sebelumnya. Setelah sepuluh sampel suara didapatkan selanjutnya peneliti melakukan pengujian dari masing-masing sampel suara tersebut untuk mengetahui apakah sampel suara tersebut dapat diterima oleh sistem atau tidak.

```

pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7
b843ae577915 /home/pi/azure/test/Alfi.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7
b843ae577915 /home/pi/azure/test/Claudia.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7
b843ae577915 /home/pi/azure/test/Ribka.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Reject
Confidence = Normal
pi@raspberrypi:~/azure $ python3 verification/VerifyFile.py 8a96f5bdea3c4f53a0b7
b843ae577915 /home/pi/azure/test/Rivardi.wav 1e1057e4-062d-4d70-aa91-b9af68a8240e
Verification Result = Accept
Confidence = Normal
pi@raspberrypi:~/azure $ █

```

Gambar 20. Pengujian Terhadap Sampel Suara

Gambar 20 merupakan cara untuk melakukan pengujian terhadap sepuluh sampel suara lain dari orang-orang yang berbeda dengan tujuan untuk mengetahui apakah sistem telah berjalan sesuai yang diharapkan atau tidak. Berdasarkan kedua pengujian yang telah dilakukan sebelumnya, maka hasil dari pengujian tersebut dapat dilihat pada tabel VIII.

IV. PENUTUP

A. Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

- 1) Implementasi *azure cognitive service* pada aplikasi *speech recognition* berhasil dilakukan.
- 2) Percobaan penggunaan aplikasi *speech recognition* untuk penguncian pada pintu dengan menggunakan *azure cognitive service* membuat keamanan lebih meningkat.
- 3) Berdasarkan hasil pengujian keseluruhan sistem, aplikasi dapat berjalan sesuai dengan yang diharapkan.
- 4) Berdasarkan hasil pengujian terhadap respon sistem, rata-rata waktu yang diperlukan oleh sistem untuk merespon adalah 00:00:05:99.
- 5) Berdasarkan hasil pengujian terhadap jarak mikrofon, jarak yang disarankan yaitu tidak lebih dari 60 cm.
- 6) Berdasarkan hasil pengujian terhadap kebisingan, tingkat kebisingan yang masih bisa diterima oleh sistem adalah dibawah 65 dB.
- 7) Berdasarkan hasil pengujian terhadap user lain, dapat dilihat bahwa 15 sampel suara selain suara dari peneliti yang diinput hasil dari verifikasi suara gagal atau artinya pengunci pada pintu tidak akan terbuka. Sistem berjalan sesuai yang diharapkan.

B. Saran

Tentunya masih ada kekurangan dalam penelitian yang dilakukan ini sehingga terdapat hal-hal yang perlu dikaji kembali agar dapat dikembangkan. Oleh karena itu, beberapa saran berikut dibuat untuk pengembangan lebih lanjut:

- 1) Untuk penelitian berikutnya profil pengguna dapat ditambah menjadi lebih dari satu pengguna.
- 2) Layanan *azure cognitive service* bisa dimanfaatkan oleh peneliti dalam banyak hal bukan hanya untuk *speech recognition*.

KUTIPAN

- [1] Muchtar, Raisha, dkk. 2016. Penerapan Perintah Suara Berbahasa Indonesia untuk Mengoperasikan Perintah Dasar di *Windows*. Universitas Sumatera Utara
- [2] Microsoft. 2018. *What is Cognitive Service?* [online]. Available: <https://docs.microsoft.com/en-us/azure/cognitive-services/welcome>
- [3] Anonim 2016. Understanding Microsoft Cognitive Services. Wintellect.
- [4] Muharram, Mulki Syahputra, dkk. 2015. Implementasi Kunci Pintu Otomatis dengan *RFID* Berbasis *Raspberry Pi* sebagai Sub Sistem dari Kunci Otomatis pada Ruang Dosen Universitas Telkom. E-Proceeding of Engineering. Vol 2, No.2, ISSN 2355-9365.
- [5] Tim Dinastindo. 1993. *Kamus Komputer Eksekutif*. Dinastindo. Jakarta.
- [6] Jogiyanto, H.M. 1999, *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Penerbit Andi. Yogyakarta.
- [7] Ali, Muhammad. 1998. *Kamus Besar Bahasa Indonesia*. Pustaka Amani. Jakarta.
- [8] Darmawan, Rian Suryo dan Addin Suwastono. 2015. Model Sistem Antrian Berbasis *Arduino Uno R3* dan *Raspberry Pi* Menggunakan *Python*. Seminar Nasional XI. ISSN 1978-0176.
- [9] Pressman, Roger. 2012. *Rekayasa Perangkat Lunak 7th ed*. Yogyakarta: ANDI.
- [10] Setiawan Ebta. 2017. *Kamus Besar Bahasa Indonesia (KBBI)* [online]. Available: <https://kbbi.web.id/mikrofon>.
- [11] Anonim. 2006. *Panduan Penulisan Karya Tulis Ilmiah Sarjana (KTIS)*. Universitas Sam Ratulangi Manado fakultas Teknik. Manado.
- [12] Masykur, Fauzan dan Fiqiana Prasetyowati. 2016. Aplikasi Rumah Pintar (*Smart Home*) Pengendali Peralatan Elektronik Rumah Tangga Berbasis *Web*. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*. Vol 3, No.1, hlm. 51-58.

TENTANG PENULIS



Saya bernama Rivardi Benedictus Ruspandi dan merupakan anak kedua dari pasangan Harry Ruspandi dan Doris Tambengie, lahir di Kota Manado pada tanggal 12 Januari 1997. Saya mulai menempuh pendidikan di sekolah dasar SD Frater Don Bosco Manado (2002 – 2008). Kemudian melanjutkan studi tingkat pertama di SMP Frater Don Bosco Manado (2008 – 2011) dan selanjutnya saya menempuh pendidikan tingkat atas di SMA Frater Don Bosco Manado (2011 – 2014). Setelah itu, di tahun 2014 saya melanjutkan pendidikan ke salah satu perguruan tinggi yang berada di Manado yaitu Universitas Sam Ratulangi Manado, dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Elektro Fakultas Teknik.