

Pengaturan *Fan Speed* dan Suhu Air Conditioner Melalui Ucapan Dengan Layanan *Google Assistant API*

Michael Albertus¹,
Universitas Kristen Maranatha Bandung
Email: michael.albertus@yahoo.com

Muliady¹
Universitas Kristen Maranatha Bandung
Email: muliady@eng.maranatha.edu

ABSTRACT: Air conditioner control using speech recognition is made to help people with disabilities that unable to operate remote control physically but have verbal abilities. Verbal control allows disabled people adjust temperatures and fan speed. Speech received by Respeaker 2-mics Pi HAT module, converted into wav, processed by Google Assistant API with Natural Language Processing algorithm by categorizing words into their types, subject, predicate, object, and description to facilitate Google Assistant API. Words matched with command in command text database Raspberry Pi 3 to enable local commands modulates signal in form of space-coded signal on GPIO, transmitted through infrared transmitter to control the Air Conditioner. Infrared database obtained by receiving infrared signal through infrared receiver that have been coded by LIRC into pulse space, calling function is created, compared in command text database. The infrared light distance from the infrared transmitter can be sent to air conditioner up to 600 cm with β NPN 2N2222A transistor worth 257, Resistor base value is 1500 Ohm, and Resistor collector value is 6.2 Ohm. Speech to text experiments with background sound intensity 35-40 dB, respondent's sound intensity 50-70dB, and the respondent's distance to the microphone 40-50 cm. System can recognize respondent's speeches with success rate above 50%. The word "High" in fan speed speech experiments cannot be detected by the system, so it is necessary to add other word to be recognized. The system can receive Google Translate speech and only got one failure.

Keywords: Speech recognition, Google Assistant API, Voice command, LIRC

ABSTRAK: Perangkat elektronik air conditioner dengan pengendalian pengenalan ucapan untuk membantu penyandang tunadaksa yang tidak mampu mengendalikan remote secara fisik tetapi memiliki kemampuan verbal. Pengendalian dengan cara verbal memungkinkan penyandang tunadaksa untuk mengubah suhu dan mengatur fan speed air conditioner. Ucapan diterima modul Respeaker 2-mics Pi HAT dikonversi menjadi format wav kemudian diolah oleh Google Assistant API dengan algoritma Natural Language Processing yaitu mengategorikan kata menjadi jenisnya, subjek, predikat, objek, dan keterangannya untuk mempermudah pencarian pada kamus Google Assistant API. Kata tersebut dibandingkan dengan perintah ucapan pada commandtextdatabaseRaspberry Pi 3 yang mengaktifkan local command dan memodulasi sinyal space-coded signal pada GPIO, ditransmisikan melalui infrared transmitter untuk mengatur air conditioner. Infrareddatabase diperoleh melalui penerimaan cahaya infrared dan dikodekan menjadi pulse space oleh software LIRC menjadi fungsi pemanggilan, dipasangkan dengan perintah ucapan commandtextdatabase. Jarak cahaya infrared dari infrared transmitter dapat dikirimkan ke air conditioner hingga sejauh 600 cm dengan β transistor NPN 2N2222A bernilai 257, nilai Resistor base sebesar 1500 Ohm, dan nilai Resistor collector sebesar 6,2 Ohm. Uji cobaspeech to text dengan kondisi intensitas background sound 35-40 dB, intensitas suara responden 50-70 dB, dan jarak responden ke microphone 40-50 cm. Sistem yang direalisasi mampu mengenali ucapan yang diberikan responden dengan keberhasilan di atas 50%. Ucapan "High" pada pengujian ucapan fan speed tidak dapat dideteksi oleh sistem, oleh karena itu perlu ditambahkan ucapan suhu agar ucapan dikenal. Sistem mampu menerima ucapan Google Translate dan hanya mendapatkan satu kali kegagalan deteksi ucapan.

Kata kunci: Pengenalan ucapan, Google Assistant API, Perintah suara, LIRC

PENDAHULUAN

Pengenalan ucapan digunakan secara luas untuk membantu penyandang tuna daksa seperti voice user interface dalam beraktivitas, misalkan pengendalian kursi roda untuk berjalan maju, mundur, dan belok dan pengendalian saklar lampu untuk menyalakan dan mematikan lampu. Pengendalian menggunakan pengenalan ucapan juga dapat diimplementasikan ke dalam perangkat elektronik sebagai pengganti tombol-tombol dengan perintah melalui ucapan. Kelebihan pengendalian dengan pengenalan ucapan adalah pada jarak pengoperasiannya. Ini berbeda dengan tombol yang harus ditekan atau disentuh, pengenalan ucapan tidak perlu dikendalikan secara kontak fisik.

¹ Universitas Kristen Maranatha Bandung

Perangkat elektronik *air conditioner* dengan pengendalian menggunakan pengenalan ucapan dibuat untuk membantu penyandang tuna daksa yang tidak mampu untuk mengendalikan *remote* secara fisik baik secara sentuhan maupun tekanan, tetapi masih memiliki kemampuan verbal. Pengendalian *remote* dengan cara verbal dapat memungkinkan penyandang tunadaksa untuk menaikkan atau menurunkan suhu dan mengatur *fan speed*. Oleh karena itu, dengan mengaplikasikan pengenalan ucapan dapat memudahkan dan memberikan kebebasan kepada penyandang tuna daksa untuk mengatur dan mengendalikan *air conditioner* secara verbal.

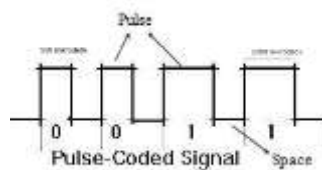
KAJIAN PUSTAKA

Transmisi Cahaya *Infrared*

Transmisi cahaya *infrared* dapat dibangkitkan menggunakan rangkaian komparator yaitu dengan membandingkan gelombang tegangan gigi gergaji sebagai masukan dengan tegangan referensi. Saat tegangan gigi gergaji bernilai lebih besar dari tegangan referensi maka akan menghasilkan keluaran komparator yang bernilai *high* atau *pulse*, sedangkan jika tegangan gigi gergaji bernilai lebih kecil dari tegangan referensi maka akan menghasilkan keluaran komparator yang bernilai *low* atau *space*. Ada tiga jenis pengkodean transmisi *infrared* yang digunakan dan semuanya berdasarkan *pulse* dan *space* yaitu *Pulse-Width Coded Signal*, *Space-Coded-Signal*, dan *Shift-Coded Signal* [1].

1. *Pulse-Width- Coded Signal*

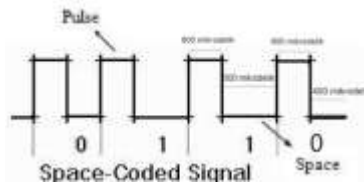
Pengkodean *Pulse-Width-Coded Signal* lebar pulsa merupakan kode informasinya. Logika '0' diperoleh saat panjang *pulse* kurang dari sama dengan 550 mikrodetik tetapi jika panjang *pulse* memiliki rentang 551 sampai 2200 mikrodetik maka dinyatakan dalam logika '1' untuk *header* yaitu saat rentang *pulse* di atas 2200 mikrodetik, dan *stop pulse* saat tidak terdapat *pulse*[1]. Contoh penggunaan *Pulse-Width-Coded Signal* adalah pada protokol SONY. Ilustrasi sinyal dapat dilihat pada Gambar 1.



■Gambar 1 *Pulse Width Coded Signals*

2. *Space-Coded Signal*

Pengkodean *Space-Coded Signal* atau *pulse distance coded signal* didasarkan pada ukuran panjang *space*. Logika '0' diperoleh saat panjang *space* kurang dari sama dengan 400 mikrodetik sedangkan jika ukuran panjang *space* 401 sampai 1200 mikrodetik maka dinyatakan sebagai logika '1', untuk *header* yaitu ketika panjang dari *pulse* sekitar 4000 mikro detik dan *space* sepanjang 1201 sampai 1600 mikrodetik, sedangkan untuk *stop pulse* yaitu ketika panjang *pulse* 400 mikrodetik dan *space* panjang memiliki panjang tak hingga[1]. Contoh penggunaan *space-coded signal* adalah pada protokol NEC Ilustrasi sinyal dapat dilihat pada Gambar 2.

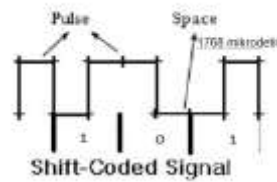


■Gambar 2 *Space Coded Signal*

3. *Shift-Coded Signal*

Pengkodean *shift-coded signal* atau *Bi-phase Modulation* ditentukan pada urutan *pulse* dan *space*. Berbeda dengan pengkodean yang lain pada pengkodean ini selalu diawali dengan 2 bit start yaitu 90 milidetik *pulse* dan 90 milidetik *space*. Logika '1' diawali dengan panjang *space* sekitar 884 mikrodetik dan *pulse* sekitar 884 mikrodetik, sedangkan untuk logika '0' diawali dengan panjang

pulse sekitar 884 mikrodetik dan *space* sekitar 884 mikrodetik [1]. Contoh dari penggunaan *shift-coded signal* ada pada protokol RC-5 PHILLIPS. Ilustrasi sinyal dapat dilihat pada Gambar 3.

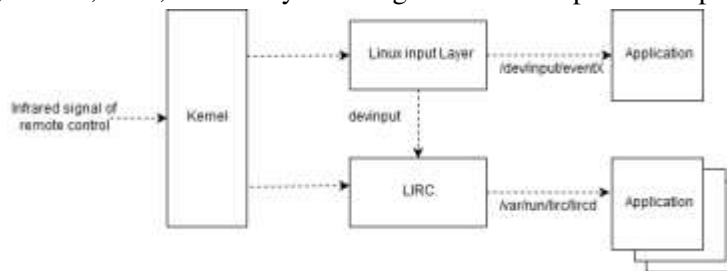


■ Gambar 3 Shift-Coded Signal

Linux Infrared Remote Control

Linux Infrared Remote Control (LIRC) adalah *software* yang digunakan untuk menerima, mendekode, dan mengirimkan sinyal *infrared*. LIRC mengonversi durasi *on* dan *off* dari cahaya *infrared* menjadi bentuk *pulse-space* dalam satuan mikrodetik. Bentuk *pulse-space* dicatat pada *kernel Linux* yang selanjutnya dapat digunakan untuk melakukan aktivitas pengendalian alat elektronik dengan memberikan perintah pada *terminal Linux* [2].

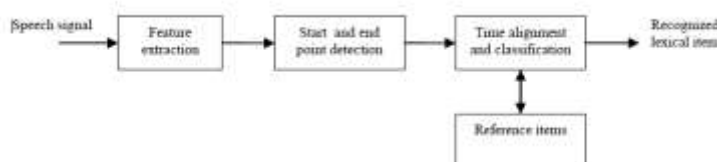
Pada konfigurasi LIRC, aplikasi akan mendapatkan data dari *Linux input layer (/dev/input)* yaitu masuk yang diterima dari perangkat *peripheral* seperti USB maupun GPIO atau LIRC (*/var/run/lirc/lircd*). LIRC juga dapat memperoleh masukan dari *Linux input layer (/dev/input)*. Perbedaan antara penggunaan LIRC maupun hanya dengan *Linux input layer* adalah LIRC dapat mengirimkan data ke lebih dari satu aplikasi sedangkan *Linux input layer* hanya dapat mengirim ke satu aplikasi. Penggunaan data LIRC membutuhkan *application support* yang digunakan pada *Linux* seperti *mythtv, kodi* dan *vlc* [2]. LIRC digunakan untuk memodulasi sinyal *infrared* dengan berbagai macam format seperti NEC, SONY, RC5, dan lainnya. Konfigurasi LIRC dapat dilihat pada Gambar 4.



■ Gambar 4 Konfigurasi LIRC

Sistematika Pengenalan Ucapan

Sistem pengenalan ucapan (*speech recognition*) merupakan teknik yang diaplikasikan dalam suatu sistem komputer untuk menerima masukan berupa suara lisan yang kemudian diubah menjadi bentuk teks digital dengan teknik *speech to text* [3]. Alur sistem yang digunakan untuk sistem pengenalan ucapan ditampilkan pada Gambar 5.



■ Gambar 5 Alur Sistem Pengenalan Ucapan

1. Feature Extraction

Langkah pertama yang dilakukan dalam pengenalan ucapan adalah *feature extraction* (ekstrasi ciri) yaitu pengambilan ciri dari suatu sinyal yaitu frekuensi, energi, dan fonem ucapan, informasi yang nantinya nilai yang diperoleh akan dianalisis untuk proses selanjutnya. Setiap informasi memiliki ciri yang berbeda atau unik. Prinsip kerja ekstrasi ciri adalah dengan mengonversi sinyal suara ke dalam beberapa parameter, ada sebagian informasi yang tidak berguna yang dibuang tanpa

menghilangkan arti sesungguhnya dari sinyal suara tersebut seperti *noise* dan distorsi. Hasil keluaran dari ekstraksi ciri ini yang akan digunakan menjadi masukan pada proses selanjutnya[3].

2. *Start and end point detection*

Langkah kedua adalah memisahkan bagian ucapan dengan bukan ucapan dengan memeriksa energi setiap sinyal, *Zero-crossings* dan karakteristik lainnya. *Zero-crossings* merupakan kondisi saat sinyal suara berubah dari positif ke negatif atau sebaliknya. Saat energi dan *zero-crossings* berada pada tingkat tertentu sinyal yang memiliki informasi berupa ucapan maupun non-ucapan dipisahkan, hanya bagian sinyal suara yang memiliki informasi ucapan yang diteruskan ke proses selanjutnya sedangkan sinyal suara bukan ucapan dibuang[3].

3. *Time alignment*

Langkah ketiga adalah mengompensasi perbedaan kecepatan dan durasi ucapan dengan memampatkan atau memanjangkan sumbu waktu (*time warping*). Pada proses ini sinyal-sinyal suara ucapan yang memiliki kecepatan yang berbeda diselaraskan periodenya hal ini perlu dilakukan untuk menghindari kesalahan translasi seperti pengucapan kata ‘bukan’ dan ‘akan’ memiliki kecepatan ucap yang berbeda namun memiliki energi pada tingkatan yang hampir sama hal ini tentunya akan menghasilkan kesalahan pada proses selanjutnya[9].

4. *Classification*

Langkah terakhir adalah *classification* (klasifikasi) yaitu sinyal suara ucapan yang telah diproses pada langkah sebelumnya kemudian diklasifikasi dan dicocokkan dengan referensi yang dimiliki, pada proses ini sinyal suara ucapan dikonversi menjadi bentuk simbolik berupa *alphanumeric* yang secara umum disebut sebagai *text* [3].

Pengenalan ucapan pada umumnya menggunakan dua algoritma utama yaitu *Dynamic Time Warping* dan *Artificial Neural Networks* [4].

1. *Dynamic Time Warping*

Dynamic Time Warping (DTW) adalah sebuah algoritma yang menghitung proyeksi pemampatan dan peregangan optimal antara dua deretan waktu. Algoritma ini menghitung kedua nilai jalur pembengkokan antara dua deretan dan jarak di antara mereka. Algoritma ini dimulai dengan perhitungan jarak lokal antara elemen dari dua urutan menggunakan berbagai jenis jarak. Metode yang paling sering digunakan untuk perhitungan jarak adalah jarak absolut antara nilai-nilai dari dua elemen (jarak euclidean). Dua ucapan dari kata yang sama oleh pengguna yang sama dapat memiliki waktu yang berbeda. *Dynamic Time Warping* merupakan metode yang efisien untuk memecahkan masalah keselarasan waktu. Algoritma DTW ditujukan untuk menyelaraskan dua deret vector dengan membelokkan sumbu waktu berulang-ulang sampai kecocokan optimal antara dua deret ditemukan [4].

2. *Artificial Neural Network*

Artificial Neural Network adalah algoritma yang didasarkan oleh cara kerja system syaraf biologis, seperti otak untuk memproses informasi. *Artificial Neural Network* seperti makhluk hidup, belajar melalui contoh-contoh dan kumpulan pengalaman. *Artificial Neural Network* dapat dikonfigurasi untuk aplikasi khusus seperti pengenalan pola atau klasifikasi data, melalui proses belajar. Sama seperti sistem biologis, pembelajaran melibatkan pengaturan koneksi sinapsis yang terdapat di antara neuron-neuron. Perhitungan nilai *thresholding* ini bermacam-macam tergantung dari sistem yang dibuat oleh pengguna [4]. Algoritma yang biasa digunakan untuk pengenalan ucapan adalah algoritma *Natural Language Processing* (NLP) yaitu dengan menganalisis kata yang terucap sesuai dengan tatabahasa manusia. Pada algoritma NLP, kata-kata yang diucapkan oleh pengguna dibagi menjadi jenisnya masing-masing yaitu subjek, predikat, objek, dan keterangan. Kata-kata yang telah dibagi menjadi jenisnya tersebut kemudian dibandingkan dengan *database* yang dimiliki. Jika kata yang dimaksud tidak ada pada *database* maka akan dicari kata yang memiliki kemiripan dari kata yang dimaksud [5].

Google Assistant API

Google Assistant API adalah layanan asisten *virtual* berbasis kecerdasan buatan yang dibuat oleh Google untuk membantu pengguna dalam beraktivitas sehari-hari. Google Assistant menggunakan *natural language* sebagai *user interface* untuk menjawab pertanyaan, membuat rekomendasi, dan melakukan perintah dengan mengajukan permintaan ke layanan *web*. Google Assistant API sangat bergantung dengan akses internet dalam penggunaannya. Google Assistant memiliki layanan berupa Google Cloud Speech to Text API untuk mengubah ucapan yang diterima serta melakukan eksekusi perintah yang diminta dengan mengubahnya ke bentuk *text* digital [6].

Google Cloud Speech to Text API merupakan layanan yang disediakan oleh Google untuk mengubah ucapan menjadi bentuk *text* digital dengan menggunakan *speech to text*. Kata yang telah ditemukan pada kamus kemudian ditampilkan dalam bentuk *text* pada *user interface* pengguna [7]. Layanan yang digunakan oleh Google Cloud Speech to Text API dibagi menjadi tiga yaitu *Synchronous speech Recognition*, *Asynchronous Speech Recognition*, dan *Streaming Speech Recognition*.

1. Synchronous Speech Recognition

Data *audio* yang dikirim ke *speech to text API*, kemudian dilakukan proses pengenalan pada data tersebut. Permintaan pada *synchronous recognition* dibatasi pada data *audio* dengan durasi maksimum 1 menit. Contoh penggunaan pada layanan ini adalah *speech command* (perintah suara) [7].

2. Asynchronous Speech Recognition

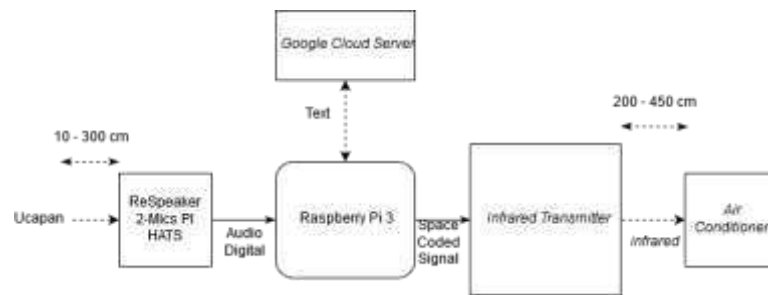
Data *audio* yang dikirim ke *speech to text API* yang memiliki durasi lebih dari 1 menit dilakukan proses pengenalan secara keseluruhan. Proses yang digunakan pada layanan ini membutuhkan waktu lebih lama daripada layanan lainnya. Contoh penggunaan pada layanan ini adalah penerjemahan pidato dan music [7].

3. Streaming Speech Recognition

Metode pengenalan data *audio* yang dirancang untuk keperluan waktu nyata seperti merekam suara yang bersifat aktif dari *microphone*. *Streaming recognition* memberikan hasil sementara saat *audio* sedang direkam, hasil *audio* yang terjadi sebelumnya ditampilkan ketika pengguna masih berbicara. Contoh penggunaan pada layanan ini adalah *live speech* (pidato langsung) [7].

PERANCANGAN DAN REALISASI

Perancangan dan Realisasi Sistem Pengaturan *Air Conditioner* dengan Pengenalan Ucapan Sistem yang dibangun dalam Tugas Akhir ini adalah sistem pengaturan *air conditioner* berbasis *Raspberry Pi 3* dengan menggunakan ucapan sebagai masukan. Modul *Respeaker 2-mics Pi HAT* digunakan untuk menerima ucapan serta mengubahnya menjadi bentuk audio digital dalam format wav. *Raspberry Pi 3* digunakan untuk mengolah masukan audio digital dengan menggunakan layanan *speech to text* Google Assistant API pada *Google Cloud Server* melalui koneksi internet dan dibandingkan pada *command text database* pada *Raspberry Pi 3*, kemudian digunakan untuk modulasi *space coded signal* ke *infrared transmitter* dengan menggunakan *software LIRC*. *Infrared transmitter* mengirimkan cahaya *infrared* termodulasi dengan menggunakan IR LED untuk mengatur suhu dan *fan speed* *air conditioner*. Diagram blok sistem dapat dilihat pada Gambar 6.



■ **Gambar 6** Diagram Blok Sistem Pengendali Air Conditioner dengan Pengenalan Ucapan

Modul *Respeaker 2-mics Pi HAT* menggunakan seluruh GPIO yaitu GPIO 1 sampai 40 yang dimiliki oleh *Raspberry Pi 3 model B* agar dapat digunakan sebagai *microphone*. Rangkaian *infrared transmitter* dihubungkan dengan dua GPIO yang terdapat pada modul *Respeaker 2-mics Pi HAT* yang merupakan GPIO 18 sebagai keluaran dan pin *ground Raspberry Pi 3 model B*. Rangkaian *Infrared Transmitter* menggunakan komponen transistor, IR LED, resistor, *voltage regulator 5V*, dan baterai 9V. Baterai 9V digunakan untuk memberikan *supply* pada rangkaian *infrared transmitter* karena modul *Respeaker 2-mics Pi HAT* tidak memiliki pin untuk memberikan *supply* dan *voltage regulator 5V* digunakan untuk membatasi tegangan menjadi 5V agar rangkaian *infrared transmitter* tidak mendapatkan kelebihan tegangan. Realisasi sistem yang dirancang dapat dilihat pada Gambar 7.



■ **Gambar 7** Realisasi Sistem Pengaturan Air Conditioner dengan Pengenalan Ucapan yang Dirancang

Konfigurasi Modul *Respeaker 2-mics Pi HAT*

Konfigurasi yang dilakukan pada modul *Respeaker 2-mics Pi HAT* adalah instalasi *voice card* dan registrasi *sub device*. Instalasi *voice card* dilakukan agar modul *Respeaker 2-mics Pi HAT* dapat digunakan untuk menerima ucapan dan mengonversinya menjadi bentuk audio digital dalam format wav. Pin pada modul *Respeaker 2-mics Pi HAT* dihubungkan dengan pin yang dimiliki pada *Raspberry Pi 3* agar dapat melakukan instalasi *voice card*. Instalasi *voice card* memerlukan koneksi internet dan dilakukan pada *terminal Raspberry Pi*.

Registrasi *sub device* dilakukan agar modul *Respeaker 2-mics Pi HAT* dapat digunakan oleh *Raspberry Pi*. Registrasi *sub device* dilakukan dengan mengubah file *.asoundrc* pada *directory /home/pi/.asoundrc* yaitu mengubah *voicecarddefault* yang dimiliki oleh *Raspberry Pi* menjadi *voice card seed* yang merupakan *voice card* modul *Respeaker 2-mics Pi HAT* dan mengubah *hardware default* menjadi *hardware seed* agar *microphone* yang dimiliki modul *Respeaker 2-mics Pi HAT* dapat digunakan.

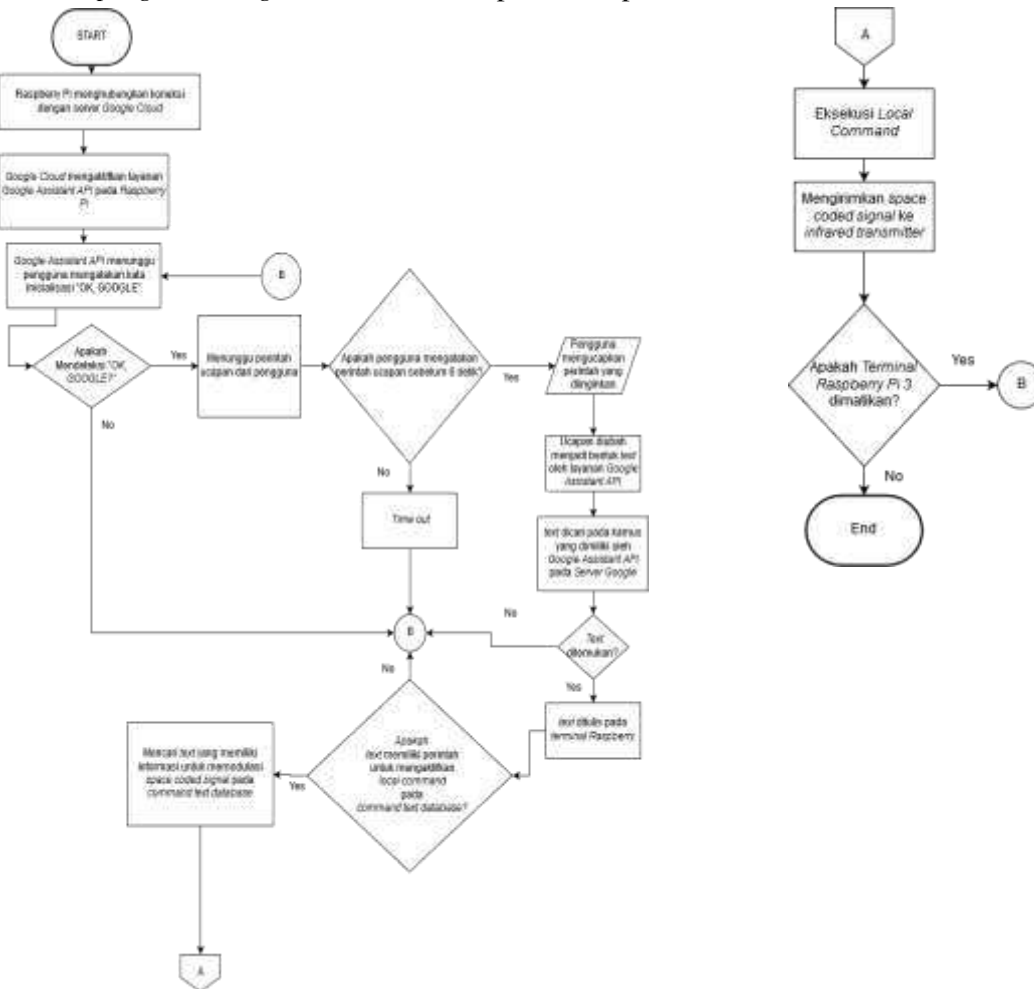
Konfigurasi *Google Assistant API*

Konfigurasi *Google Assistant API* meliputi registrasi *Google Assistant API*, mengunduh dan menyalin file *client_id.JSON*, dan membuat *command text database*. Registrasi *Google Assistant API* dilakukan agar layanan yang diberikan oleh *Google Cloud Platform* yaitu *speech to text* dapat

digunakan. Registrasi *Google Assistant API* membutuhkan koneksi internet dan dilakukan pada website <https://console.cloud.google.com/> yang meliputi penamaan *project*, aktivasi *Google Assistant API*, membuat kredensial, dan membuat *Authorisation client ID* untuk pengaturan layanan. Pada implementasinya, perintah ucapan yang digunakan memiliki dua frasa, frasa pertama adalah besaran suhu dan frasa kedua adalah *fan speed*.

Flowchart Program Google Assistant API

Google Assistant API membutuhkan internet agar dapat digunakan, langkah pertama yang dilakukan untuk menggunakan *Google Assistant API* adalah menghubungkannya dengan *server Google Cloud* melalui *terminal Raspberry Pi*. *Text* yang ditulis pada *terminal Raspberry Pi* kemudian dicari pada *command text database*, jika *text* tersebut memiliki perintah untuk mengaktifkan *local command* maka langkah selanjutnya adalah mencari *text* yang memiliki informasi untuk memodulasi *space coded signal* pada *command text database* dan *local command* dieksekusi untuk mengirimkan *space coded signal* ke *infrared transmitter*, sedangkan sebaliknya jika tidak memiliki perintah untuk mengaktifkan *local command* maka akan kembali ke langkah pengguna untuk mengucapkan “OK, GOOGLE”. *Flowchart program Google Assistant API* dapat dilihat pada Gambar 8.



■ **Gambar 8** *Flowchart Program Google Assistant API*

Konfigurasi LIRC

Konfigurasi LIRC meliputi instalasi *software LIRC*, demodulasi cahaya *infrared* menjadi bentuk *pulse-space*, *pulse-space database*, dan pemeriksaan *pulse-space code*. Cahaya *infrared* tidak dapat langsung diterima oleh komponen *IR receiver TSOP 1738* pada *Raspberry Pi*, namun perlu menggunakan bantuan dari *software LIRC*. Instalasi *software LIRC* membutuhkan koneksi internet dan

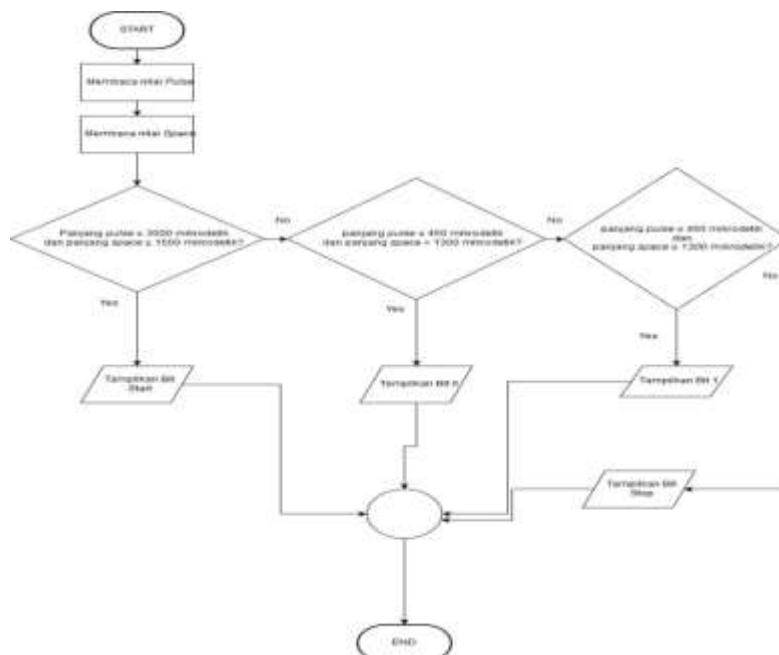
dilakukan pada *terminal Raspberry Pi*. Software LIRC kemudian dikonfigurasi untuk mengaktifkan GPIO sebagai masukan cahaya *infrared* dan keluaran *pulse-space code* ke *infrared transmitter*, karena GPIO yang tersisa pada *expansion board* adalah GPIO 23 dan GPIO 18, maka kedua GPIO tersebut digunakan sebagai masukan dan keluaran.

Semua cahaya *infrared* pada *remote control* disalin dan disimpan bentuk *pulse-space*-nya agar dapat digunakan, bentuk *pulse-space* disimpan pada *pulse-spacedatabase*. *Pulse-space database* memiliki isi yaitu bentuk *pulse-space* dari cahaya *infraredremote control* yang disalin dan dipasangkan dengan fungsinya masing-masing. Fungsi-fungsi dari setiap bentuk *pulse-space* dari cahaya *infrared* dapat digunakan dengan memberikan perintah IR SEND_ONCE pada *terminal Raspberry Pi* yang kemudian akan mengirimkan *pulse space code* pada keluaran GPIO untuk mentransmisikan cahaya *infrared* termodulasi pada *infrared transmitter*. Perintah untuk mengirimkan *pulse space* dipasangkan dengan perintah ucapan pada *command text database* yang terdapat pada *Google Assistant API*. Berdasarkan pengamatan yang dilakukan, bentuk sinyal yang dihasilkan pada keluaran GPIO adalah modulasi *space-coded signal* karena adanya perbedaan jarak antar *pulse* yang dapat dilihat pada Gambar 9.



■ Gambar 9 Sinyal Keluaran GPIO 18

Keluaran GPIO 18 yang berupa modulasi *space coded signal* memudahkan untuk memeriksa *pulse space* yang diterima dari cahaya *infrared* memiliki kesalahan baik pernah diambil ataupun ketidaklengkapan pengambilan cahaya *infrared*. Pemeriksaan dilakukan dengan cara mengubahnya menjadi bentuk *binary*. Bit *start* diperoleh jika nilai *pulse* lebih dari 3000 mikrodetik dan *space* lebih dari 1500 mikrodetik, bit 0 diperoleh jika panjang *pulse* kurang dari sama dengan 450 mikrodetik dan panjang *space* sama dengan 1300 mikrodetik, bit 1 diperoleh jika panjang *pulse* kurang dari sama dengan 450 mikrodetik dan panjang *space* lebih dari sama dengan 1300 mikrodetik, dan bit *stop* diperoleh jika panjang *pulse* kurang dari sama dengan 450 mikrodetik dan panjang *space* lebih dari 76 milidetik. *Flowchart* program untuk mengubah bentuk *pulse-space* menjadi *binary* dapat dilihat pada Gambar 10.



■ Gambar 10 Flowchart Program Mengubah *Pulse-Space* ke *Binary*

Berdasarkan pengamatan yang dilakukan pada perancangan ini bit yang dianggap penting adalah bit 45 hingga bit 48 untuk suhu dan bit 68 hingga 71 untuk *fan speed*. Dapat dilihat keterangan bit 45 hingga bit 48 pada Tabel 1 dan keterangan bit 68 hingga 71 pada Tabel 2.

Perancangan dan Realisasi Rangkaian *Infrared Transmitter*

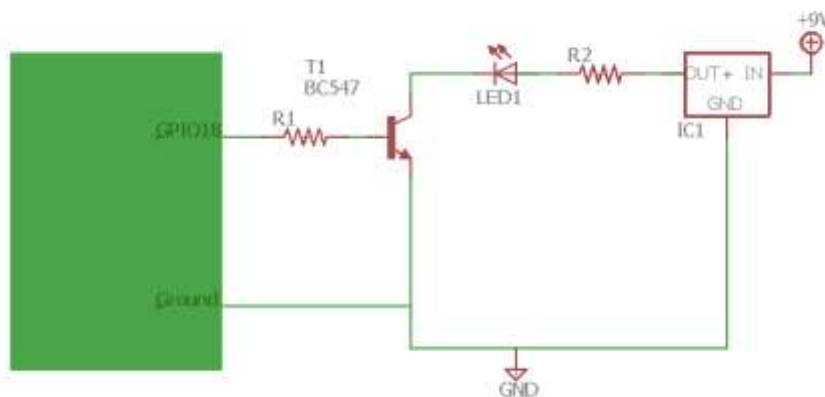
Rangkaian *infrared transmitter* menggunakan komponen IR LED untuk mengirimkan cahaya *infrared* ke *air conditioner*, komponen IR LED dapat bekerja pada arus *forward* 100 mA dengan tegangan *forward* 1,85 V. komponen IR LED tidak langsung dihubungkan dengan GPIO 18 untuk menerima modulasi *space coded signal* karena tegangan keluaran sebesar 3,6 V dapat merusak komponen dan arusnya yang sangat kecil yaitu antara 0,8 sampai 1 mikroampere, oleh karena itu perlu menggunakan transistor untuk menguatkan dari GPIO agar IR LED dapat bekerja. Skema rangkaian dapat dilihat pada Gambar 11.

■ **Tabel 1** Bentuk *Binary* dari Suhu

Bit 45-48	Keterangan (°C)
0000	16
1000	17
1100	18
0010	19
1010	20
0110	21
1110	22
0001	23
1001	24
0101	25
1101	26
0011	27
1011	28
0111	29
1111	30

■ **Tabel 2** Bentuk *Binary* dari *Fan Speed*

Bit 68-71	Keterangan
0101	<i>Auto Speed</i>
1000	<i>Low Speed</i>
1010	<i>Medium Speed</i>
1110	<i>High Speed</i>



■ **Gambar 11** Skema Rangkaian *Infrared Transmitter*

Nilai β dari transistor NPN 2N2222A yang diukur adalah 257, untuk memperoleh nilai arus *collector* sebesar 500 mA maka nilai arus *base* adalah sebesar 1,95 mA. Resistor *base* digunakan agar

memperoleh arus *base* sebesar 1,95 mA, tegangan GPIO18 yang dihasilkan sebesar 3,6 V dan tegangan antara kaki *base* dengan kaki *emitter* sebesar 0,7 V maka diperoleh resistor *base* sebesar 1487 ohm, namun pada realisasinya nilai resistor *base* yang digunakan adalah 1500 ohm.

Resistor *Collector* digunakan agar komponen IR LED tidak melebihi tegangan kerjanya akibat pemberian tegangan dari V_{CC} sebesar 5 V yang dapat mengakibatkan rusaknya komponen, komponen IR LED bekerja pada tegangan 1,8 V dan arus kerja 500 mA, maka nilai resistor *collector* adalah sebesar 6,4 ohm, namun dalam realisasinya nilai komponen yang digunakan adalah 6,2 ohm. IR LED pada rangkaian *transmitter infrared* mengirimkan cahaya *infrared* termodulasi untuk mengatur suhu dan *fan speed* pada *air conditioner*.

HASIL DAN PEMBAHASAN

Pengujian Jarak *Infrared Transmitter* Terhadap *Air Conditioner*

Pengujian ini bertujuan untuk menguji jarak cahaya *infrared* dari *infrared transmitter* yang dapat dikirimkan ke *air conditioner* sehingga perintah yang diberikan dapat dijalankan. Perintah yang diberikan diketik pada *terminal Raspberry Pi*. Pengujian dilakukan pada jarak 155 cm, 217 cm, 400 cm, 513 cm, dan 600 cm. Hasil pengujian dapat dilihat pada Tabel 3.

■ **Tabel 3** Hasil pengujian jarak *infrared transmitter* terhadap *Air Conditioner*

Jarak (cm)	Hasil
155	Berhasil diterima oleh <i>Air Conditioner</i>
217	Berhasil diterima oleh <i>Air Conditioner</i>
400	Berhasil diterima oleh <i>Air Conditioner</i>
513	Berhasil diterima oleh <i>Air Conditioner</i>
600	Berhasil diterima oleh <i>Air Conditioner</i>

Pengujian *Speech to Text Google Assistant API* terhadap Umur dan Pekerjaan Responden

Pengujian ini bertujuan untuk menguji keberhasilan *Speech to Text Google Assistant API* jika umur dan pekerjaan responden yang berbeda-beda. Pengujian dilakukan dengan mengucapkan suhu 16°C hingga 30°C dan *fan speed Auto, Low, Medium, dan High*. Pengujian keberhasilan *Google Assistant API* mendeteksi ucapan setiap responden dengan keterangan ‘1’ untuk berhasil dan ‘0’ untuk gagal. Karakter responden dapat dilihat pada Tabel 4, keberhasilan pengujian dapat dilihat pada Tabel 5 untuk pengucapan suhu dan Tabel 6 untuk pengucapan *fan speed*.

Rata-Rata Persentase Keberhasilan Responden dalam ucapan Suhu dan *fanspeed* dapat dilihat pada Tabel 7 dan Tabel 8. Rata-rata persentase keseluruhan pengujian ucapan responden dapat dilihat pada Tabel 9 untuk pengujian ucapan suhu, dan Tabel 10 untuk pengujian ucapan *fan speed*. Pada pengujian ucapan *fan speed*, pengucapan “*High*” dideteksi sebagai “*Hi*” oleh *Google Assistant API*, oleh karena itu diuji kembali dengan menambahkan ucapan suhu sebelum “*High*” agar ucapan “*High*” dapat dideteksi. Pengujian ini dilakukan oleh responden ke dua dan dapat dilihat pada Tabel 11. Pengujian terakhir dilakukan dengan menggunakan sumber suara *Google Translate* melalui *smartphone* dengan ucapan yang diacak. Pengujian dapat dilihat pada Tabel 12.

■ **Tabel 4** Karakter Responden

Responden ke	Umur	Pekerjaan
1	48	Dosen
2	22	Mahasiswa
3	20	Mahasiswa

■ **Tabel 5** Hasil Pengujian Ucapan Suhu

Responden	Percobaan	Ucapan Suhu														
		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0
	3	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
	4	1	0	1	1	1	0	1	0	1	1	1	1	1	1	1
2	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	1
	2	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
	3	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

■ **Tabel 6** Hasil Pengujian Ucapan *fanspeed*

Responden	Percobaan	Ucapan <i>Fanspeed</i>			
		<i>Auto</i>	<i>Low</i>	<i>Medium</i>	<i>High</i>
1	1	1	1	1	0
	2	1	1	1	0
	3	1	1	1	0
	4	1	1	1	0
2	1	1	1	0	0
	2	1	1	1	0
	3	1	1	1	0
	4	1	1	1	0
3	1	1	1	1	0
	2	1	1	1	0
	3	1	0	1	0
	4	1	1	1	0

■ **Tabel 7** Rata-Rata Keberhasilan Responden dalam Ucapan Suhu

Responden	Rata- Rata Persentase Keberhasilan (%)															
	Ucapan Suhu															
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	10			10	10		10		10	10	10	10	10			
	0	75	75	0	0	50	0	75	0	0	0	0	0	75	50	
2	10	10		10	10	10	10		10	10	10	10	10			
	0	0	75	0	0	0	0	50	0	0	0	0	0	75	75	
3	10		10	10	10			10	10	10			10	10	10	
	0	75	0	0	0	75	75	0	0	0	75	75	0	0	0	

■ **Tabel 8** Rata-Rata Keberhasilan Responden dalam Ucapan *Fan Speed*

Responden	Rata- Rata Persentase Keberhasilan (%)			
	Ucapan <i>Fan Speed</i>			
	<i>Auto</i>	<i>Low</i>	<i>Medium</i>	<i>High</i>
1	100	100	100	0
2	100	100	75	0
3	100	75	100	0

■ **Tabel 9** Rata-Rata Persentase Keseluruhan Ucapan Responden dalam Pengujian Ucapan Suhu

Responden	Rata-Rata Keseluruhan Ucapan
-----------	------------------------------

	Suhu (%)
1	87
2	92
3	92

■ **Tabel 10** Rata-Rata Persentase Keseluruhan Ucapan Responden dalam Pengujian Ucapan *Fan Speed*

Responden	Rata-Rata Keseluruhan Ucapan <i>Fan Speed</i> (%)
1	75
2	69
3	69

■ **Tabel 11** Pengujian Ucapan *fan speed High* dengan Menambahkan Suhu.

Percobaan Ke	Ucapan			
	16 High	20 High	25 High	30 High
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	1

■ **Tabel 12** Pengujian Ucapan Menggunakan *Google Translate*.

Percobaan Ke	Ucapan										
	16	17	20	Auto	Low	Medium	20 Auto	22 Low	28	Medium	27 High
1	1	1	1	1	1	0	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1

KESIMPULAN

Sistem berhasil direalisasikan menggunakan modul *Respeaker 2-mics Pi HAT*, *Raspberry Pi 3*, *Google Assistant API*, *LIRC*, dan *infrared transmitter* dengan transistor NPN 2N2222A dengan nilai β sebesar 257, nilai resistor *base* sebesar 1500 Ohm, dan resistor *collector* sebesar 6,2 Ohm. Uji coba dengan kondisi intensitas *background sound* 35-40 dB, intensitas suara responden 50-70 dB, dan jarak responden ke *microphone* 40-50 cm. Sistem yang direalisasi mampu mengenali ucapan yang diberikan responden dengan keberhasilan di atas 50%. Ucapan “*High*” pada pengujian ucapan *fan speed* tidak dapat dideteksi oleh sistem, oleh karena itu perlu ditambahkan ucapan suhu agar ucapan “*High*” dapat dikenali. Sistem mampu menerima ucapan yang diberikan oleh *Google Translate* dan hanya mendapatkan satu kali kegagalan deteksi ucapan.

DAFTAR REFERENSI

- [1] Innovative Electronic “INFRARED TRANSCIEVER”, 2014
Internet: http://www.innovativeelectronics.com/innovative_electronics/manual/Manual%20SPC%20IR%20TRANSCEIVER.pdf. [30 Oktober 2018]
- [2] LIRC “LIRC Configuration Guide”, 2017
Internet : <http://www.lirc.org/html/configuration-guide.html>. [31 Oktober 2018]
- [3] Gyulyustan, H., S. Enkov.. ”*Experimental speech recognidtion system based on Raspberry Pi 3*,”IOSR Journal of Computer Engineering (IOSR-JCE). Volume 19, Issue 3, Ver. II : 2.2017.

- [4] Milivojša, S., S. Ivanović, T. Erić. "Implementation of Voice Control Interface for Smart Home Automation System,". IEEE 7th International Conference on Consumer Electronics – Berlin, ICCE-Berlin. 2017.
- [5] Julia, H., C.D. Manning. "Advances in Natural Language Processing,"SCIENCE. Volume 349, Issue 6245. AAAS. pp 261-265. 2016.
- [6] Surinder, K. Sanchit, S. Utkarsh, J and Arpit, R."Voice Command System Using Raspberry Pi,".Advanced Computational Intelligence: An International Journal (ACIJ). Vol.3, No.3 : pp 227-229. 2016.
- [7] M. Josef,"Voice control of smart home by using Google Cloud Speech-To-Text API," Bachelor Thesis,JAMK University of Applied Sciences, Finland, 2018.