

BAB III

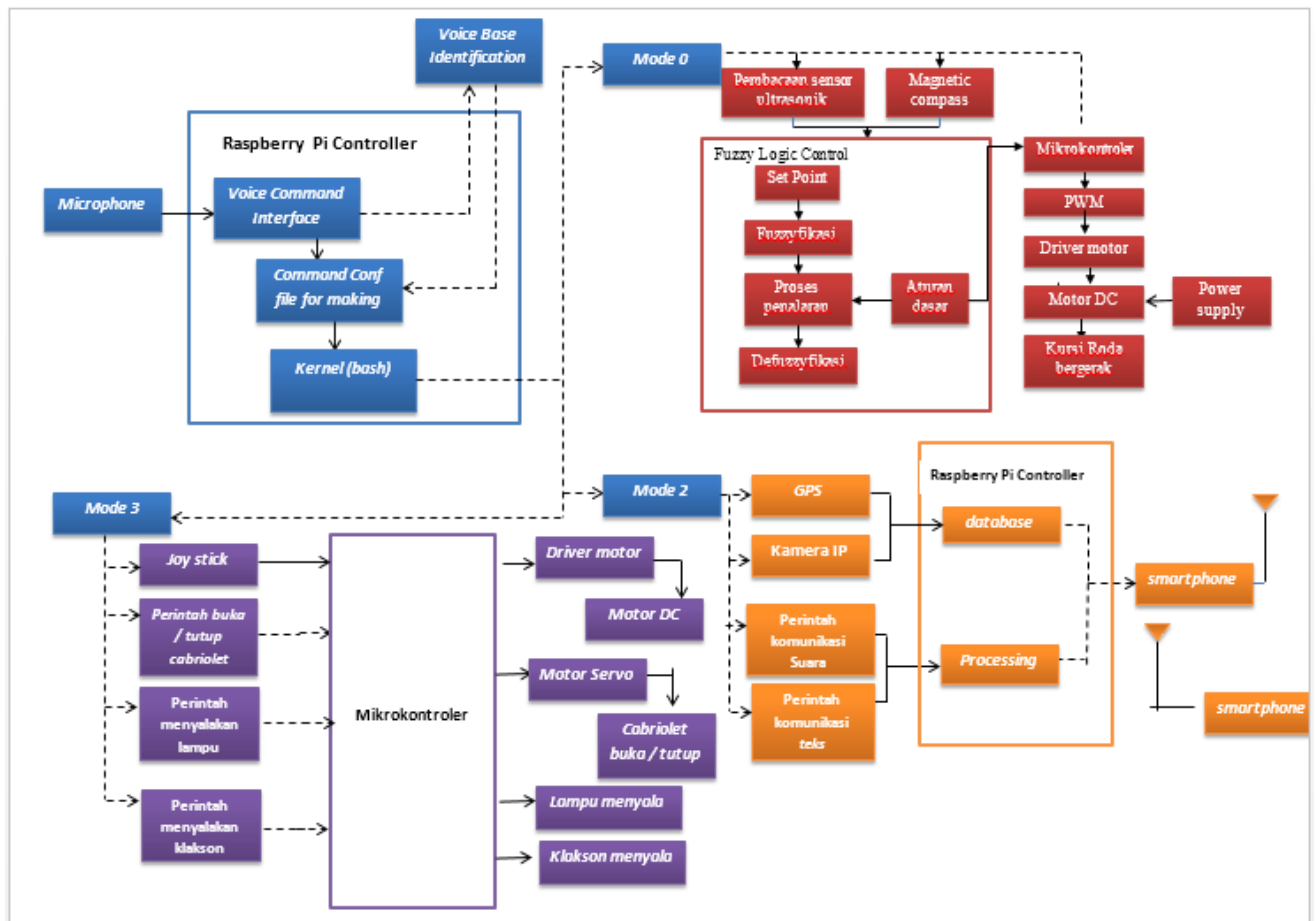
METODE DAN PROSES PENYELESAIAN

3.1 Perancangan

3.1.1 Perancangan Blok Diagram

Perancangan blok diagram sistem keseluruhan terdapat pada gambar 1, dimana blok diagram tersebut terbagi kedalam 4 sub bagian. Yaitu sub sistem kendali suara, sub sistem kendali dengan joystick, sub sistem kendali motor dc dan sub sistem untuk monitoring objek bergerak. Namun pada pembahasan disini akan lebih berfokus pada bagian sistem kendali suaranya.

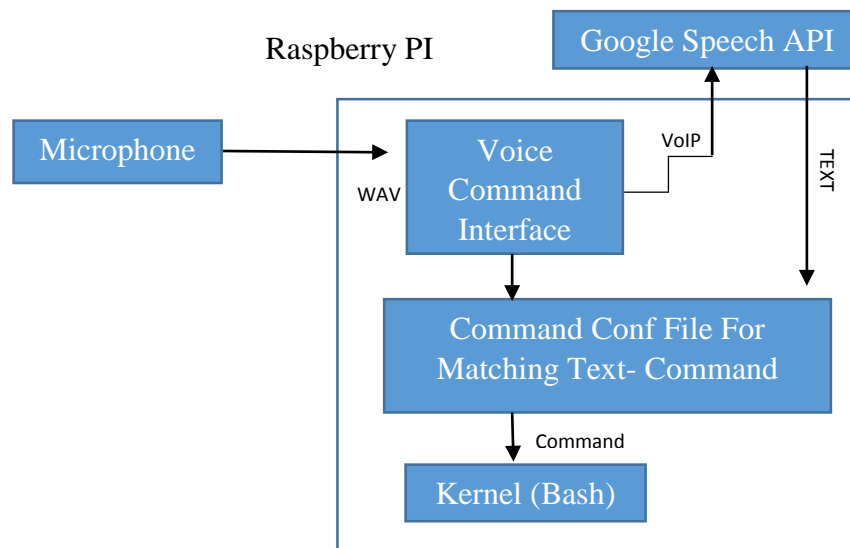
- **Blok Diagram Sistem Keseluruhan**



Gambar 1. Blok Diagram Sistem Keseluruhan

- **Blok Diagram Sistem Kendali Suara**

Untuk perancangan sistem kendali suara terdapat pada gambar 2. Dimana blok diagram tersebut terdiri dari bagian input, proses pengolahan suara dan juga output. Pada bagian input sistem dibutuhkan input berupa suara sehingga digunakan *microphone* untuk menangkap sinyal suara dari pembicara yang dijadikan sebagai input dan bentuk input suara tersebut disimpan dalam format .wav. Microphone dapat terhubung langsung dengan mikroprosesor raspberry pi. Pada bagian mikroprosesor digunakan raspberry pi untuk pemrosesan sinyal suaranya. Pemrosesan sinyal suara terbagi kedalam 2 bagian yang pertama yaitu pemrosesan sinyal untuk sistem *speaker recognition* dan yang kedua pemrosesan sinyal untuk sistem kendali dengan *voice control* atau *voice command* dengan memanfaatkan fitur Google Speech Api pada raspberry pi.



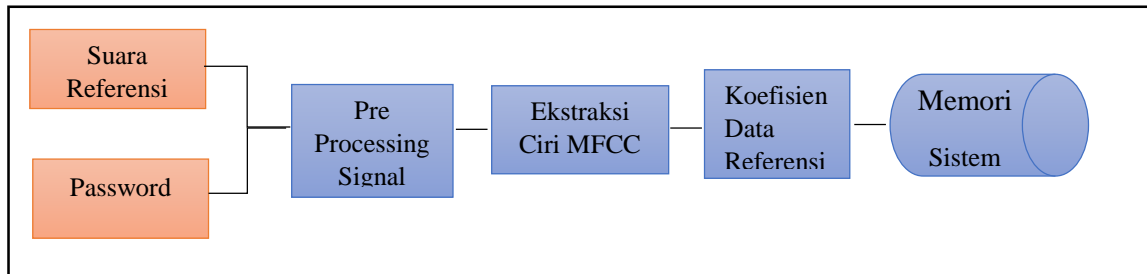
Gambar 2. Blok Diagram Sistem Kendali Suara

1. Speaker Recognition

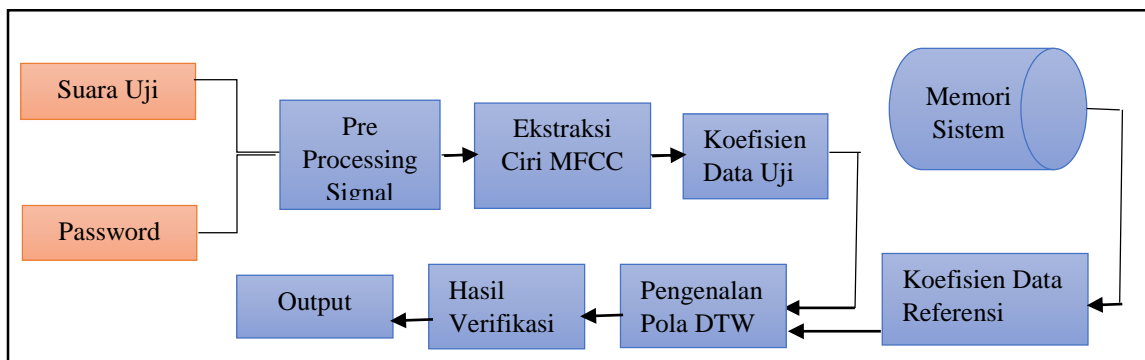
Untuk bagian sistem *speaker recognition* input diambil dari suara yang ditangkap *microphone*. Suara uji diambil dari input microphone yang terhubung ke salah satu port raspberry pi dalam bentuk .wav dan suara tersebut akan diproses oleh mikroprosesor. Suara referensi telah disimpan pada sistem penyimpanan mikroprosesor yang nanti nya akan dibandingkan dengan suara input dari *microphone*. Pada sistem terdapat 2 fungsi yang disediakan yaitu fungsi untuk penambahan suara baru dan fungsi pengujian suara.

Berikut adalah gambaran dari proses kedua fungsi sistem tersebut.

- Fungsi Penambahan Suara



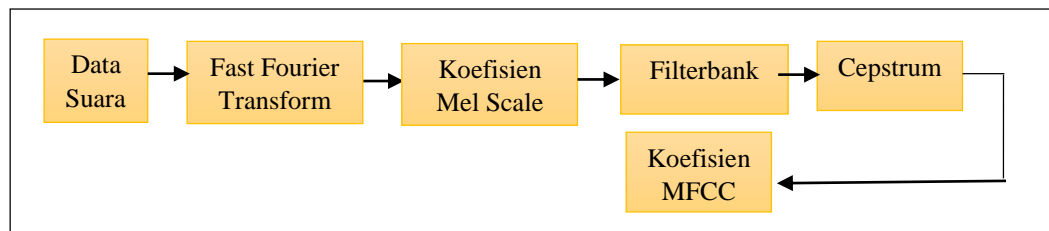
- Fungsi Pengujian Suara



Gambar 3. Proses Pengolahan Signal Pada Kedua Fungsi

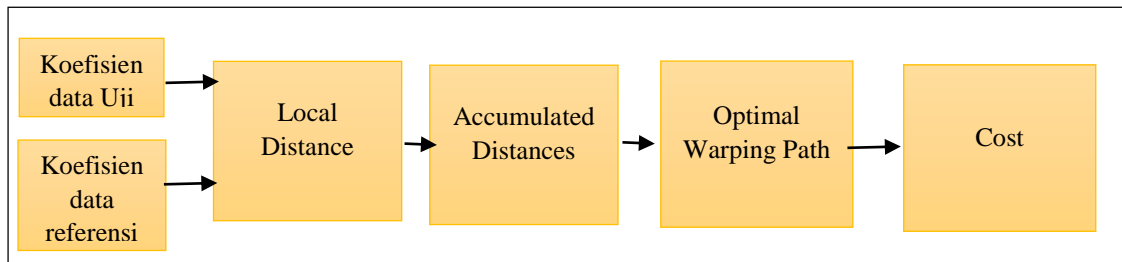
Sebelum melalui proses Ekstraksi ciri, Sinyal suara akan terlebih dahulu melewati proses *pre – processing signal*, yang terdiri dari proses *front – end detection*, *emphasize filter*, *frame blocking* dan *windowing*. Front – end detection merupakan proses pemisahan sinyal *voiced* dan *unvoiced*. Selanjutnya sinyal suara akan masuk ke proses *filtering* yang disebut *pre-emphasize filter*. Dan sinyal akan melalui proses *frame – blocking* setelah melewati ketiga proses sebelumnya untuk dibagi menjadi *frame – frame* kecil yang akan masuk pada proses *windowing*.

- Proses Ekstraksi Ciri MFCC



Gambar 4. Diagram Blok Proses Ekstraksi Ciri MFCC

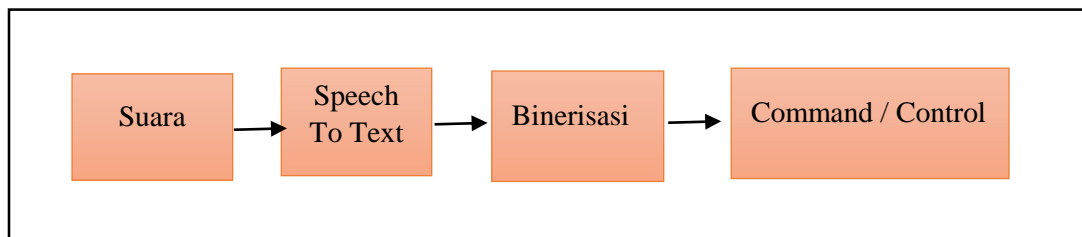
- Proses Pengenalan Pola DTW



Gambar 5. Diagram Blok Proses Pengenalan Pola DTW

Sinyal akan masuk proses ekstraksi ciri MFCC setelah melalui proses *pre-processing signal*, untuk mendapatkan nilai koefisien sebanyak jumlah dari orde yang digunakan. Selanjutnya untuk proses verifikasi hasil dari koefisien MFCC dari data referensi dan data uji akan melalui proses pengenalan pola DTW yaitu metode pengenalan pola yang didasarkan pada pencocokan pola sinyal. Hasil pencocokan ditampilkan pada tampilan *display*, jika terdapat kecocokan antara pembicara dengan password akses dapat diterima dan pembicara dapat menggunakan suaranya untuk mengendalikan kursi roda. Jika tidak terdapat kecocokan maka akses tidak dapat dibuka atau akses ditolak.

2. Sistem Kendali Suara



Gambar 6. Blok Diagram Sistem Kendali Suara

Pada bagian sistem kendali, sistem dapat dikendalikan jika akses pada proses *speaker recognition* telah diverifikasi kecocokannya antara suara pengguna dengan password. Untuk kendali kursi roda telah dibuat 8 bit perintah suara dengan total perintah sebanyak 20 perintah suara yang terbagi ke dalam 3 mode atau 3 sub sistem yang terdapat pada blok diagram sistem keseluruhan. Setiap perintah suara yang ditangkap oleh *microphone* akan ditampilkan dalam bentuk text. Setiap kata atau pengucapan yang tepat dan telah dibuat dalam bentuk text akan dilakukan binerisasi atau pengubahan hasil kata dalam bentuk text dari proses *speech-to-text* ke dalam bentuk biner atau digital. Dilakukan

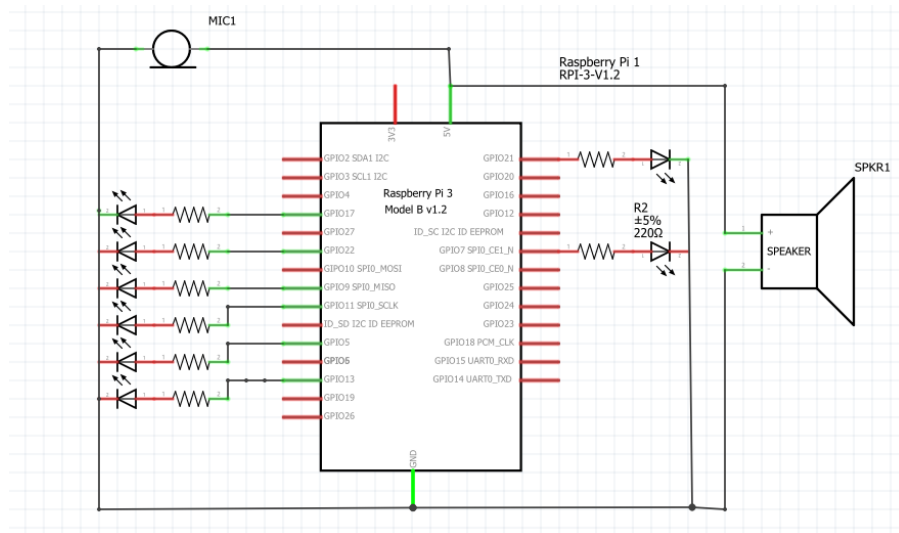
proses binerisasi agar setiap perintah dengan fungsi mengendalikan atau mengontrol dapat dipahami oleh hardware yang hanya mengetahui nilai dalam bentuk biner atau digital yaitu 1 atau 0. Setelah semua perintah suara dibuat atau diterjemahkan dalam bentuk biner, setiap nilai biner dari perintah suara tersebut dijadikan input untuk kendali atau control pada sub sistem yang dibuat.

Berikut adalah tabel untuk perintah suara dalam bentuk biner.

DAFTAR PERINTAH SUARA									
No	PERINTAH	MODE		BINER					
0	Berhenti	0	0	0	0	0	0	0	0
1	Maju Perlahan	0	0	0	0	0	0	0	1
2	Maju Normal	0	0	0	0	0	0	1	0
3	Maju Cepat	0	0	0	0	0	0	1	1
4	Mundur Perlahan	0	0	0	0	0	1	0	0
5	Mundur Normal	0	0	0	0	0	1	0	1
6	Belok Kanan	0	0	0	0	0	1	1	0
7	Belok Kiri	0	0	0	0	0	1	1	1
8	Lampu Nyala	0	1	0	0	1	0	0	0
9	Lampu Mati	0	1	0	0	1	0	0	1
10	Klakson	0	1	0	0	1	0	1	0
11	Tudung Buka	0	1	0	0	1	1	0	0
12	Tudung Tutup	0	1	0	0	1	1	0	1
13	Bagi Lokasi	1	0	0	0	1	1	1	0
14	Ambil Foto	1	0	0	0	1	1	1	1
15	Ambil Video	1	0	0	1	0	0	0	0
16	Telepon Keluarga	1	0	0	1	0	0	0	1
17	Telepon Polisi	1	0	0	1	0	0	1	0
18	Telepon Ambulan	1	0	0	1	0	0	1	1
19	SMS Keluarga	1	0	0	1	0	1	0	0

Tabel 1. Perintah Suara Untuk Kendali Sistem

3.1.2 Perancangan Skema Alat



Gambar 7. Skema Rangkaian Sistem

Sistem terdiri dari 3 bagian yaitu, input, prosesor dan juga output. Untuk bagian input digunakan usb microphone yang berfungsi sebagai penangkap sinyal suara dari pembicara. Pada bagian prosesor digunakan raspberry pi yang digunakan untuk pemrosesan sinyal suara dari microphone. Kemudian untuk output terdapat dua output yaitu monitor dan LED, untuk monitor digunakan sebagai penampil hasil verifikasi speaker recognition berupa aplikasi yang dibuat dengan GUI. Sedangkan LED digunakan sebagai indikator perintah suara yang sudah diubah kedalam bentuk biner.

3.1.3 Perancangan Algoritma

3.1.3.1 Algoritma *Pre – Processing Signal*

Bagian perancangan *pre – processing signal* merupakan bagian pemrosesan awal dari sinyal suara yang digunakan baik sebagai data uji ataupun data referensi yang terdapat pada sistem. Format rekaman suara yang digunakan harus dalam bentuk format .wav. Tujuan dari *pre-processing signal* yaitu untuk memisahkan antara sinyal *voiced* dan *unvoiced*. Terdapat tahapan yang harus dilalui agar sinyal *voiced* dan *unvoiced* bisa dipisahkan yaitu, dengan melalui tahap *front-end detection*, *filter emphasis*, dan *frame blocking*. Setelah sinyal *voiced* dan *unvoiced* terpisahkan sinyal akan terbagi ke dalam frame – frame kecil.

Kemudian data dari suara referensi atau suara data uji diubah ke dalam bentuk array, dimana *array* tersebut akan mempresentasikan nilai dari fungsi *amplitude* sinyal suara terhadap waktu. Lalu data *array* dikonversi dari bentuk bilangan *float* ke dalam bentuk bilangan tipe *double*. Sinyal akan melalui proses *front end detection* untuk disisakan sinyal *voiced*nya saja. Dimana untuk menyisakan sinyal *voiced* tersebut sinyal *unvoiced* akan dipotong pada bagian awal dan akhir sinyal. Untuk pemotongan sinyal dilakukan dengan menentukan nilai *threshold* sebesar ± 0.1 , *threshold* tersebut digunakan untuk mencari titik pertama, dimana amplitudo sinyal lebih dari 0.1 atau kurang dari -0.1. nilai dari titik pertama tersebut akan dijadikan titik potong untuk memotong sinyal *unvoiced* dan hanya akan menyisakan sinyal *voiced*-nya saja.

Setelah sinyal melalui proses *front-end detection* selanjutnya sinyal akan melalui proses filter emphasis. Sinyal hasil dari filter akan dipisahkan berdasarkan suku kata “MA” dan “JU”. Hal tersebut dilakukan agar proses dari pengenalan pola sinyal dibandingkan sesuai dengan suku kata yang diucapkan. Untuk pemisahan suku kata digunakan metode dengan membagi panjang data sinyal menjadi dua bagian dengan diasumsikan bahwa penyebutan kata “MA” dan “JU” itu sama. Setelah suku kata tersebut dipisahkan, kemudian sinyal akan melalui proses *resampling* mengikuti sinyal yang lebih pendek. Kemudian masing – masing dari sinyal akan diframing menjadi *frame – frame* kecil dengan pembagian waktu framing sebesar 25ms, kemudian masing masing dari frame tersebut melalui proses *windowing* dengan metode *hamming window*. Akan didapat 4 data array dari hasil akhir proses ini, dimana data array tersebut terdiri dari kumpulan frame sinyal suara “MA” dan “JU” dari suara data uji dan suara dari data referensi.

3.1.3.2 Algoritma Ekstraksi Ciri MFCC

Pada proses ini sinyal diubah kedalam bentuk domain frekuensi melalui FFT (*Fast Fourier Transform*). Sinyal akan melalui filterbanks dengan menghitung nilai *mel scale* terlebih dahulu untuk mendapatkan koefisien filter. Untuk bagian ekstraksi ciri dengan MFCC ini digunakan pemrosesan sinyal dengan menggunakan library program yang tersedia pada *python library*.

3.1.3.2 Algoritma Pengenalan Pola DTW

Pada proses ini terdapat 2 data sinyal suara yang akan diproses, yaitu sinyal suara data uji dan data referensi. Data uji kemudian akan dibandingkan dengan data referensi yang dituju sesuai dengan password yang terdapat pada sistem. Untuk proses perbandingan dilakukan sebanyak jumlah frame sinyal suara “MA” “JU”. Setelah dilakukan perbandingan akan didapatkan nilai cost dimana nilai cost untuk masing – masing sinyal suara “MA” dan “JU” tersebut merupakan hasil akhir dari proses tersebut.

Terdapat beberapa tahapan proses pada pengenalan pola DTW, untuk proses pertama yang akan dilalui sinyal yaitu menentukan *local distance* antara kedua variable data uji dan data referensi. *Local distance* didapat dengan cara mencari selisih jarak setiap nilai koefisien uji yang dihasilkan dengan koefisien referensi, dimana satu nilai koefisien uji dibandingkan dengan seluruh koefisien referensi untuk dicari selisih nilainya. Proses selanjutnya atau proses kedua yaitu mencari nilai *accumulated cost*, yaitu dengan cara menjumlahkan jarak *local distance* sebelumnya yang telah didapat sesuai dengan titik matriks local distances yang dituangkan ke dalam bentuk matriks yang dinamakan *accumulated cost*.

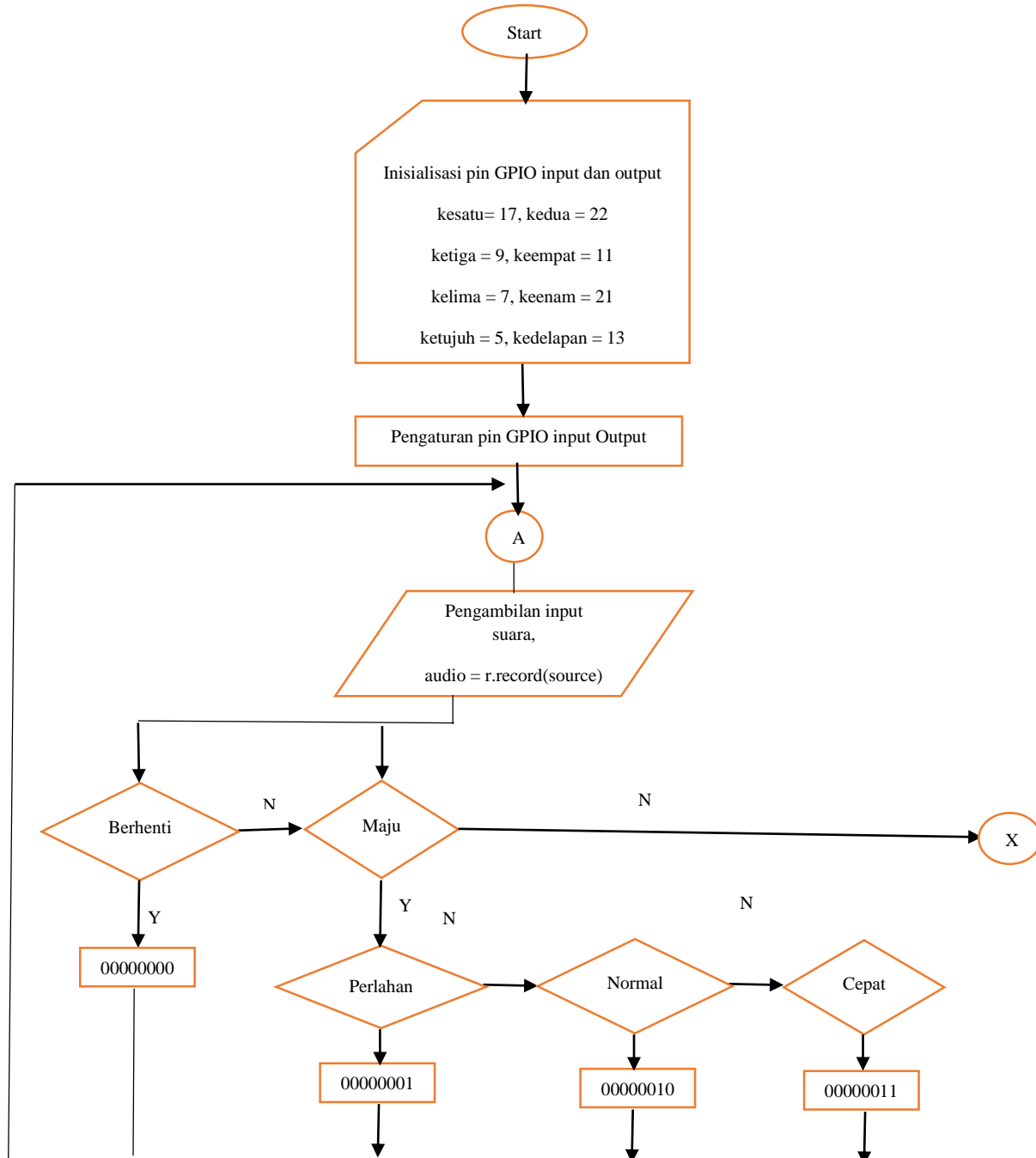
3.1.3.5 Algoritma Penggunaan Aplikasi

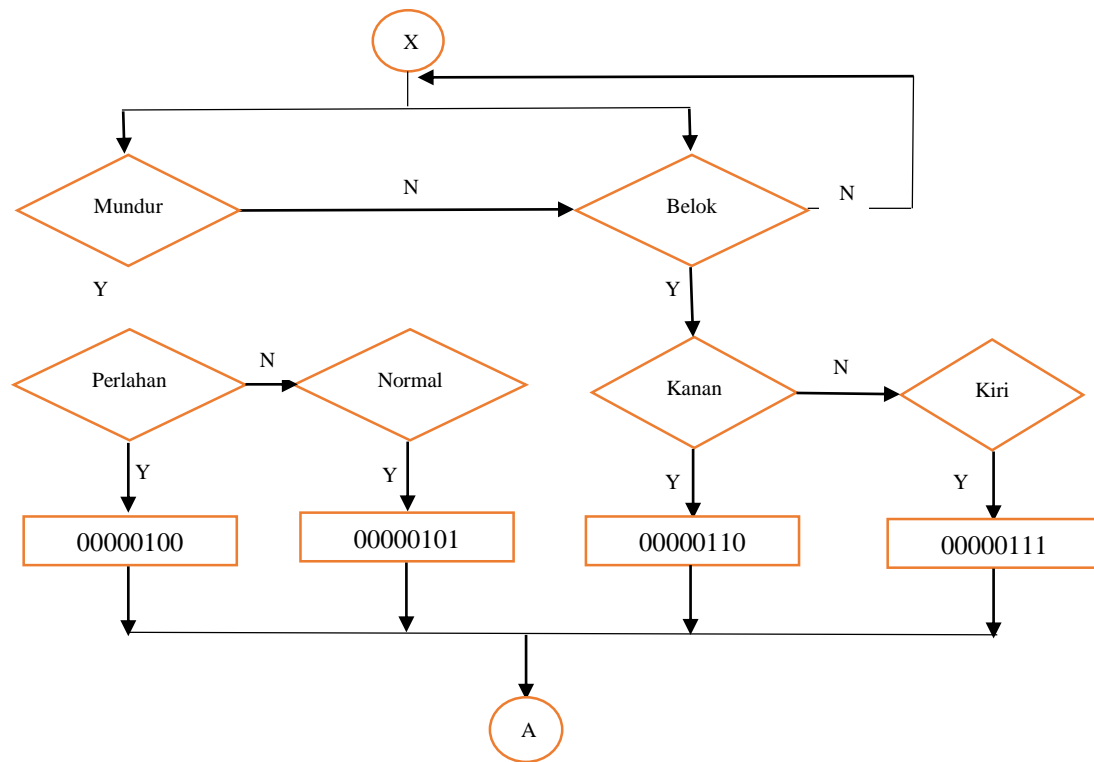
Untuk proses *speaker recognition* dibuat aplikasi dengan menggunakan GUI pada python sebagai penampil menu untuk pengujian atau pengetesan dan juga menu untuk penambahan suara baru jika sistem ingin mengganti settingan pembicara untuk dapat mengendalikan kursi roda dengan suara pembicara baru. Aplikasi *speaker recognition* dirancang agar dapat membatasi suara yang dapat mengendalikan sistem. Jika hasil pengujian suara pembicara terdapat kecocokan dengan settingan sistem yang sudah dibuat maka sistem akan membuka akses agar sistem dapat dikendalikan oleh pembicara tersebut. Dimana pembicara dapat mengendalikan sistem dengan perintah suara yang sudah tersedia dan tersimpan pada sistem yaitu sebanyak 21 perintah untuk dapat mengendalikan sistem yang terbagi kedalam 3 mode berdasarkan sub sistem yang dibuat. Setiap perintah suara yang diucapkan dengan benar oleh pembicara dan dimengerti sistem perintah suara akan ditampilkan pada layar dan akan langsung di eksekusi oleh mikrokontroler. Jika perintah suara tidak dimengerti sistem, sistem akan meminta pengulangan pengucapan untuk perintah suara tersebut.

3.1.4 Perancangan Diagram Alir

3.1.4.1 Diagram Alir Untuk Sistem Kendali Suara

- MODE 1

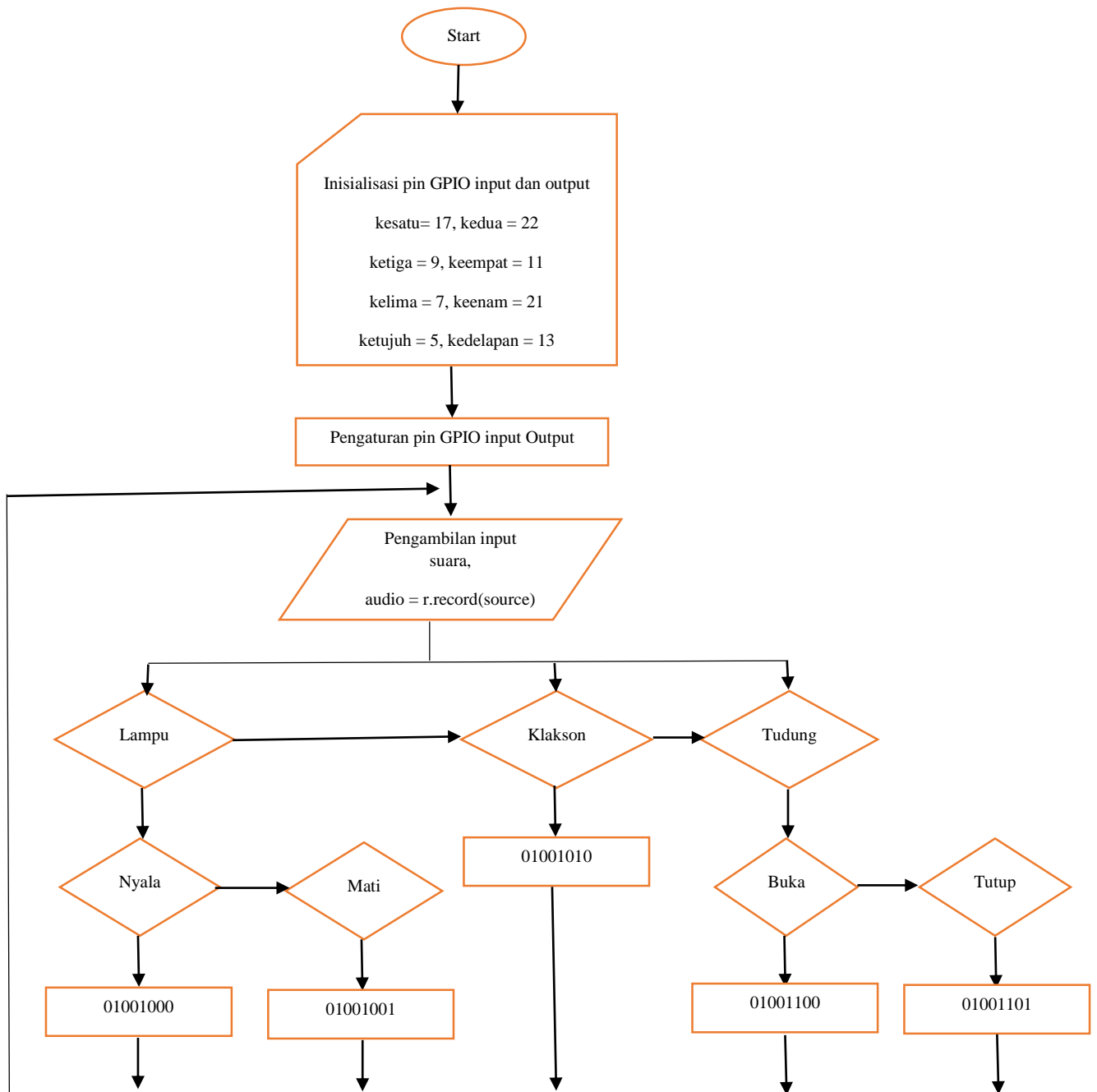




Gambar 8. Diagram Alir Sistem Kendali Suara Mode 1

Gambar 8 diagram alir sistem kendali dengan mode 1 menjelaskan bagaimana alur atau proses kendali sistem. Mulai dari inisialisasi penggunaan pin raspberry yang digunakan, pengambilan input suara dengan microphone sampai dengan proses pemberian instruksi atau perintah dengan menterjemahkan perintah dari bentuk teks kedalam bentuk biner. Dilakukan penterjemahan atau konversi dari teks ke biner agar dapat dimengerti oleh sistem yang hanya mengerti bentuk digital atau nilai 0 dan 1. Sinyal suara yang ditangkap oleh mikrofon akan diproses oleh raspberry untuk dikenali kata apa yang diucapkan oleh pembicara dengan menggunakan fitur `google_speech_api` yang penggunaannya secara online. Jika kata yang diucapkan telah terdeteksi atau dikenali dan terdapat pada sistem, kemudian sistem akan menterjemahkan sinyal suara tersebut ke dalam bentuk teks atau dikenal dengan `speech to text`. Dan hasil `speech to text` ini lah yang kemudian di konversi atau diubah ke dalam bentuk biner. Data biner tersebut akan dikirim ke mikrokontroler sebagai input untuk diolah agar dapat mengendalikan sistem yang akan diaplikasikan untuk sistem kendali pada mode ini.

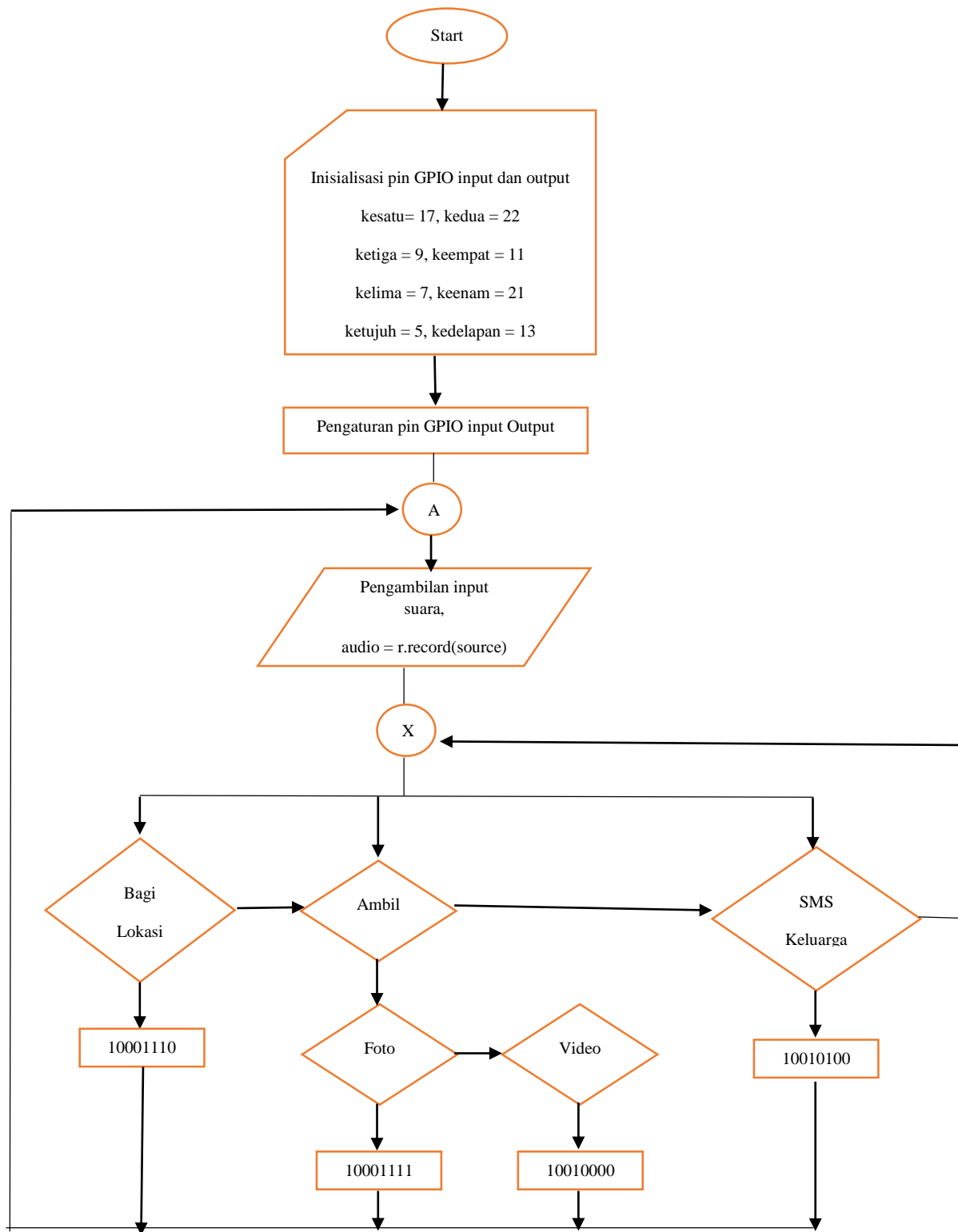
- MODE 2

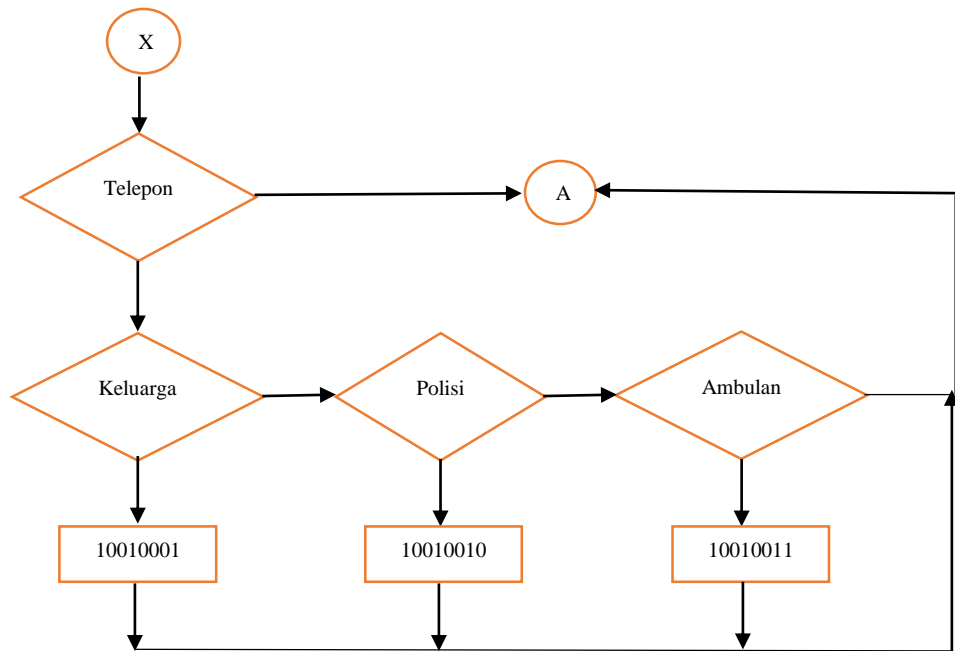


Gambar 9. Diagram Alir Sistem Kendali Suara Mode 2

Diagram alir pada mode 2 merupakan diagram alir untuk fitur tambahan yang akan diaplikasikan pada sebuah kursi roda. Data biner yang didapat dari input suara akan dikirim ke mikrokontroler sebagai input untuk diolah agar dapat mengendalikan sistem yang akan diaplikasikan untuk sistem kendali pada mode ini.

- MODE 3





Gambar 10. Diagram Alir Sistem Kendali Suara Mode 3

Diagram alir pada mode 3 merupakan diagram alir untuk fitur tambahan yang akan diaplikasikan pada sebuah kursi roda. Data biner yang didapat dari input suara akan dikirim ke mikrokontroler sebagai input untuk diolah agar dapat mengendalikan sistem yang akan diaplikasikan untuk sistem kendali pada mode ini.