GOOGLE HOME MINI SEBAGAI SISTEM PENGONTROL PERANGKAT ELEKTRONIK BERBASIS *VOICE RECOGNITION*

Google Home Mini as an Electronic Device Control System Based on Voice Recognition

Thoriq Dharmawan [1], Suci Aulia, S.T, M.T. [2], Dadan Nur Ramadan, S.Pd., M.T[3]
Prodi D3 Teknologi Telekomunikasi, Universitas Telkom
[1] thoriqdharmawan378@gmail.com, [2]sucia@tass.telkomuniversity.ac.id, [3]dadan.nr@gmail.com

***Abstract*— Voice recognition is a process of capturing words spoken through a microphone and then turning them into digital data. The application of voice recognition can be found on Google Home Mini. With this technology, it was tested the implementation of a system with voice recognition method using Google Home Mini combined with an IoT platform so that it can control electronic equipment through voice commands.**

***Intisari*— *Voice recognition* merupakan sebuah proses menangkap kata - kata yang diucapkan melalui *microphone* lalu mengubahnya menjadi data digital. Penerapan *voice recognition* bisa kita jumpai pada Google Home Mini. Dengan adanya teknologi tersebut, maka diujicobakan pengimplementasian suatu sistem dengan metoda *voice recognition* menggunakan Google Home Mini yang dikombinasikan dengan sebuah *platform* Iot sehingga dapat mengendalikan peralatan elektronik melalui perintah suara.**

Kata Kunci*— *Voice Recognition, Audio Processing, Google Home Mini

I. PENDAHULUAN

Seiring perkembangan zaman, manusia semakin menuntut pola hidup yang lebih praktis dan lebih efisien atas pekerjaan rumah tangga. Pola hidup seperti ini sangat membantu bagi orang penyandang disabilitas, yang mana membutuhkan pengendalian peralatan elektronik tanpa harus beranjak dari tempat mereka untuk menekan tombol untuk mengendalikan sesuatu.

Voice recognition merupakan sebuah proses menangkap kata - kata yang diucapkan melalui microphone ataupun telepon lalu mengubahnya menjadi data digital [13][18-19]. Penerapan voice recognition bisa kita jumpai pada Google Home Mini. Google Home Mini merupakan wireless speaker cerdas dengan bantuan Google Assistant guna memperoleh pembaruan informasi, cuaca, atau memberikan perintah [11]. Sebelum Google Home Mini mengenali apa yang diperintahkan oleh pengguna, sistem akan mengenali konsonan dan vokal dari pengguna, data ini kemudian dipakai untuk menebak perintah yang dimaksud. Dengan adanya teknologi tersebut, maka diujicobakan pengimplementasian suatu sistem dengan metoda voice recognition menggunakan Google Home Mini yang dikombinasikan dengan sebuah platform Internet of Thing (IoT) yang bersifat open-source sehingga dapat mengendalikan peralatan elektronik melalui perintah suara. Telah ada penelitian yang menggunakan voice recognition menjadi pengendalian peralatan elektronik, tetapi penelitian tersebut memepunyai kekurangan yaitu jarak perangkat masukan untuk voice recognition harus dekat dengan sistem yang digunakan, dan terbatasnya variasi kata yang diucapkan.

II. DASAR TEORI

2.1 Suara

Suara merupakan semua getaran yang merambat melalui sebuah medium. Medium yang dapat dilalui oleh suara adalah udara, air, dan benda padat. Syarat terjadinya suara yaitu adanya sumber suara, media yang menghantarkan, dan penerima. Setiap suara mempunyai frequensi, yaitu jumlah getaran dalam satu detik yang mempunyai satuan Hz [19].

Frequensi suara berbeda – beda, pada dasarnya manusia hanya dapat mendengar suara dengan frequensi antara 20 Hz – 20.000 Hz. Masusia menghasilkan suara dari pita suara yang bergetar, dari getaran tersebut merambat melalui udara dan diterima oleh gendang telinga yang terdapat didalam telinga, sehingga suara dapat didengar.

Selain frequensi terdapat juga amplitudo suara yang membuat keras lemahnya suara, amplitudo ini dapat diketahui dengan dilakukan pengukuran dengan satuan tekanan/intensitas suara dalam decibel (dB). Manusia dapat mendengar suara mulai dari batas 0dB [8][10].

2.2 *Voice Recognition*

Voice recognition merupakan suatu teknik pengenalan suara yang memungkinkan komputer menangkap frequensi suara manusia. Tujuan utama sistem ini adalah untuk mengenali kata – kata yang diucpkan dengan cara mengekstraksi, mengkarakterisasi, dan akhirnya mengenali sinyal [2][7][12-14].

Pada voice recognition ada dua fase yaitu identifikasi pembicara dan verifikasi pembicara. Identifikasi pembicara adalah tugas menentukan siapa yang berbicara dari sekumpulan suara atau pembicara yang dikenal, kemudian jika pembicara terdapat sebagai identitas dan suara tertentu maka digunakan untuk memverifikasi, klaim ini disebut verifikasi [18][20][23].

2.3 Google Home Mini

Google Home Mini merupakan *wireless speaker* cerdas dengan bantuan Google Assistant guna memperoleh pembaruan informasi, cuaca, atau memberikan perintah ke Google Assistant. Sehingga dapat menjembatani komunikasi antara manusia dan Google Assistant.

Gambar 2. 1 Google Home Mini

Google Home Mini ini mempuyai spesifikasi [11] sebagai berikut :

- Diameter : 98 mm
- Tinggi : 42 mm
- 802.11b/g/n/ac (2,4GHz/5GHz) Wi-Fi
- *Far-Field Voice Recognition Support*
- *Power* 5V; 1,8 A

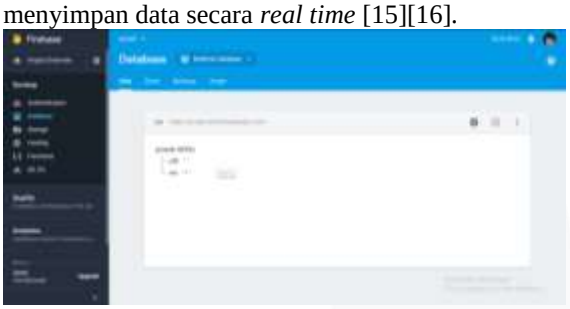
2.4 Action on Google

Action on Google adalah *platform online* yang digunakan untuk mempermudah hubungan lintas platform dengan menggunakan Google Assistant dari *quick commands* hingga *full conversations* [1].

Gambar 2. 2 Tampilan Utama Action on Google

2.5 Firebase

Firebase adalah *platform online* yang memungkinkan membantu pengembangan aplikasi dengan kualitas tinggi [7-8]. *Platform* ini dapat menyimpan data untuk memasukan, memperbaharui, ataupun menghapus data ke dalam bagian *database*. *Platform* ini digunakan sebagai *database* untuk menyimpan data secara *real time* [15][16].



Gambar 2. 3 Tampilan *Database* Firebase

2.6 Arduino IDE (*Integrated Developtment Enviroenment*)

Arduino IDE adalah perangkat lunak *open-source* yang digunakan untuk menyusun, mengedit, dan mengunggah *script code* ke dalam papan Arduino sehingga dapat mempermudah penulisan *script code* yang akan diunggah ke papan Arduino [3][5].

Gambar 2. 4 Tampilan ArduinoIDE

2.7 Firebase CLI (*Command Line Interface*)

Firebase CLI merupakan modul yang terdapat dalam *tool* NPM (Node Package Manager) yang memungkinkan kita mengelola *project* Firebase sehingga pengerjaan *project* menjadi lebih mudah[6][9].

2.8 NodeMCU

NodeMCU merupakan platform IoT yang dibekali dengan *chip* ESP8266, dengan adanya chip ini NodeMCU dapat terhubung ke jaringan internet[4][22][24].

Gambar 2. 5 NodeMCU

Penggunaan NodeMCU dibutuhkan untuk membaca nilai yang terdapat pada *database* Firebase. NodeMCU Mempunyai spesifikasi sebagai berikut :

Tabel 1. Spesifikasi NodeMCU

GPIO	13 Pin
Tegangan Masukan	3.3 volt – 5 volt
Ukuran Papan	57 mm x 30 mm
Mikrokontroller	ESP8266
USB Port	Micro USB

2.9 Module MT3608 *Step Up Votage*

MT3608 merupakan module yang berfungsi untuk meningkatkan tegangan yang masuk menjadi lebih besar, pada module ini terdapat potensio yang dapat diputar untuk mengatur tegangan keluaran, jika pengaturan tegangan pada potensio terlalu kecil maka nilai tegangan keluaran sama dengan tegangan masukan.

Gambar 2. 6 MT3608 *Step Up*

Karena pada sistem yang dibuat menggunakan Module Relay yang bekerja pada nilai tegangan 5 *volt*, maka komponen ini diperkulan, untuk meningkatkan nilai tegangan 3 *volt* yang dikeluarkan oleh NodeMCU menjadi 5 *volt*. MT3608 *Step Up Voltage* mempunyai spesifikasi sebagai berikut :

Tabel 2 Spesifikasi MT3608

Input Voltage	2 – 24 volt
Output Voltage	Up to 28 volt
Efficiency	97% efficiency
Adjustable output voltage	-

2.10 Module Relay 5 *volt* 2 *Channel*

Module Relay 5 *volt* 2 *channel* merupakan module yang digunakan untuk memutus dan menymbungkan rangkaian listrik, module ini bekerja pada nilai tegangan 5 *volt* dan mempunyai kapasitas 2 buah relay dalam satu modulenya[6].

Gambar 2. 7 Module Relay 2 channel

Module ini terdapat Pin yang digunakan untuk

memberikan catuan daya dan juga terdapat Pin yang digunakan untuk memberikan tegangan pemicu kondisi. Module ini mempunyai dua kondisi yaitu *Normally Open* (NO) dan *Normally Close* (NC), kondisi NO merupakan kondisi Relay terjadi *open circuit* ketika tidak diberi tegangan pemicu, sedangkan kondisi NC merupakan kondisi Relay *close circuit* ketika tidak diberi tegangan pemicu. Module Relay mempunyai spesifikasi sebagai berikut :

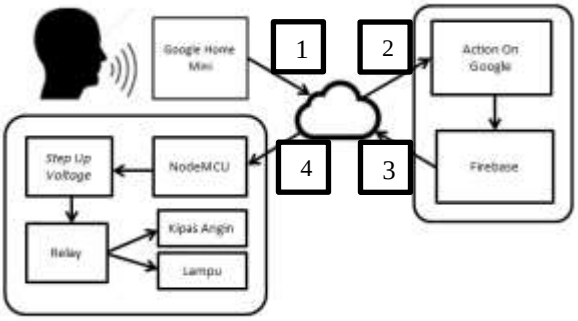
Tabel 3. Spesifikasi *Module* Relay

Tipe	SPDT
Tegangan Kerja	5 volt
AC Voltage Max	250 volt
AC Current Max	10 ampere

III.PERANCANGAN SISTEM VOICE RECOGNITION

Sistem yang dibuat pada Proyek Akhir ini adalah sebuah sistem yang dapat mengendalikan peralatan elektronika seperti lampu, kipas angin, dan perangkat lainnya dengan memanfaatkan Google Home Mini (GHM) sebagai media *voice recognition* yang ditambahkan dengan NodeMCU sebagai platform IoT yang digunakan.

Berikut merupakan blok diagram dari sistem *voice recognition* menggunakan GHM untuk pengendalian perangkat elektronik secara keseluruhan



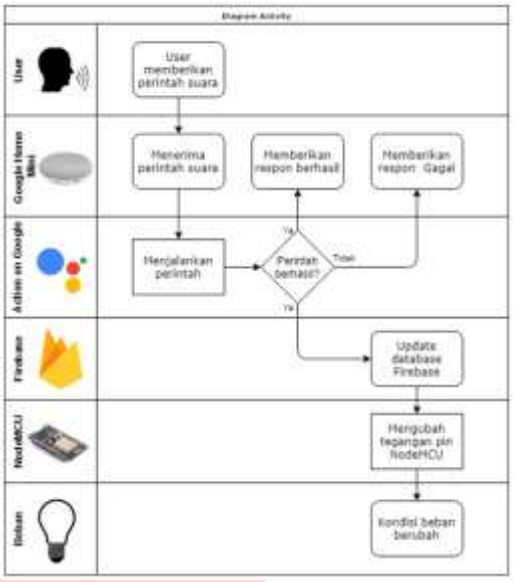
Gambar 3. 1 *Block* Diagram *Voice Recognition*

Pada **Gambar 3.1** manusia memberikan perintah suara ke GHM yang mana suara tersebut akan diolah menjadi sebuah data, kemudian data tersebut dikirimkan ke Action On Google melalui jaringan *Wi-Fi*, setalah itu data diteruskan ke *database* Firebase sehingga dapat merubah nilai data yang terdapat pada *database*, perubahan nilai tersebut akan mempengaruhi pembacaan NodeMCU Firebase sehingga mempengaruhi perubahan nilai tegangan pada pin – pin NodeMCU.

Perubahan nilai tegangan pada pin NodeMCU dapat berubah sesuai kode program yang telah ditambahkan ke papan NodeMCU, dari perubahan nilai tegangan itu akan memicu kerja dari Relay, sehingga Relay dapat memutus atau menyambungkan rangkaian listrik yang telah terpasang.

3.1 *Diagran Activity* Sistem *Voice Recognition* Menggunakan GHM

Berikut merupakan *Diagran Activity* sistem *voice recognition* menggunakan GHM untuk mengendalikan perangkat elektronik :



Gambar 3. 2 *Diagram Activity* Sistem

3.2 Konfigurasi Sistem

Berikut merupakan proses konfigurasi untuk sistem *voice recognition* menggunakan GHM dibagian *software* :

3.2.1 Menghubungkan Firebase dengan Action on Google

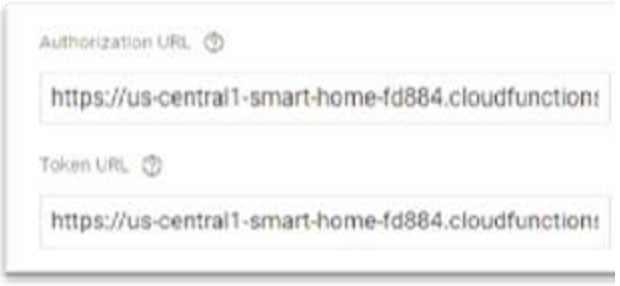
Untuk menghubungkan Firebase dengan Action on Google diperlukan *project id* Firebase yang siap digunakan, pembuatan project id dapat dilakukan melalui *command promopt* dengan melakukan instalasi Firebase CLI.

Setelah melakukan *deployment* melalui *command prompt* maka didapatkan *project id* pada *project* Firebase yang tersedia, untuk menghubungkan Firebase dengan Action on Google kita perlu masuk ke tampilan utama pada Action on Google, lalu disisi kiri pilih baris <Build> lalu <Actions>

Pada halaman <Actions> masukan *project id* yang tadi telah didapatkan dengan menuliskan https://us-central1-<project-id-Firebase>.cloudfunctions.net/smarthome setelah itu klik <Done>.

Gambar 3. 3 *Create Actions*

Setelah itu pergi ke <Advance Options> lalu <Account Linking> Pada halaman ini pilih bagian <Client Information> masukan link https://us-central1-<projectid>.cloudfunctions.net/fakeauth untuk Authorization URL dan masukan https://us-central1-<project-id >.cloudfunctions.net/faketoken untuk Token URL yang akan digunakan dalam kode utama nanti.

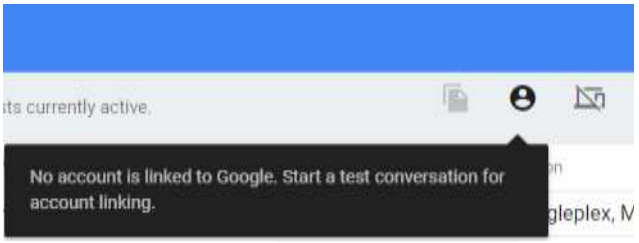


Gambar 3. 4 *Account Linking*

Dengan demikian Action on Google telah terhubung dengan Firebase.

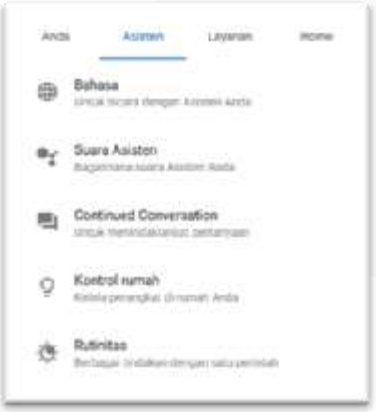
3.2.2 Penautan Akun Google

Agar Google Assistant nantinya dapat mengenali perintah pengembangan Home Control yang terdapat pada Action on Google, diperlukan penautan dengan akun Google pada *smartphone*, akun Google pada *smartphone* harus sama dengan akun Google yang digunakan sebelumnya. untuk melihat akun Google apakah sudah terhubung atau belum perlu masuk ke menu <Test> lalu <Simulator> pada Action on Google. Berikut merupakan contoh akun Google yang belum terhubung dengan Action on Google :



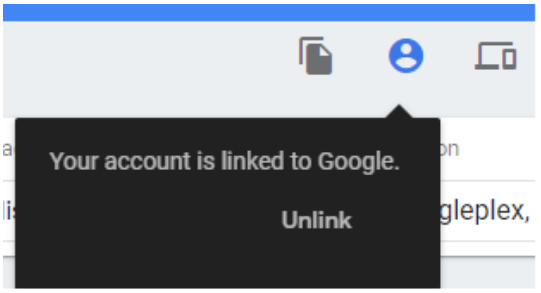
Gambar 3. 5 Belum Terhubung

Dengan melakukan penautan ini maka Google Assistant dapat mengerti perintah pengembangan Home Control yang terdapat pada Action on Google. Penutan ini dilakukan dengan cara masuk ke Google Assistant Setting pada *smartphone*. Masuk ke <Google Assistant> lalu pilih <Setelan> kemudian pilih <Kontrol Rumah>



Gambar 3. 6 Setelan Google Assistant

Setelah itu klik *icon* (+) pada sisi kanan bawah layar *smartphone*. Setelah itu akan muncul pilihan *<[test] my test app>* kemudian klik. Setelah dipilih akun Google Assistant di *smartphone* telah terhubung, dengan demikian Google Assistant dapat mengenali perintah yang masuk ke pengembangan Home Control pada Action on Google. Tanda bahwa Assistant Google telah terhubung dengan Action on Google yaitu terdapat tanda seperti pada dibawah ini pada menu <Simulator> yang terdapat pada Action on Google.



Gambar 3. 7 Sudah Terhubung

Jika ada perangkat yang ditambahkan atau diubah ke dalam sistem *voice recognition* menggunakan GHM perlu memutus dan menyambungkan kembali Google Assistant dengan Action on Google dengan langkah yang sama seperti ini.

3.2.3 Pembangunan Kode Perintah

Dalam pembangunan kode perintah ini diperlukan *Text Editor* untuk mempermudahkan penulisan kode. Penulisan kode ini berada pada direktori yang telah terinstal modul Firebase CLI dengan nama *index.js*. Pada file tersebut tulislah kode – kode perintah pembangunan sistem ini. Berikut merupakan kode yang digunakan untuk memanggil *library* yang diperlukan :

```
1.  Masukan Library
2.
3.  functions <= require('firebase-functions');
4.  smarthome <= require('actions-on-google');
5.  util <= require('util');
6.  admin <= require('firebase-admin');
```

Kode tersebut digunakan untuk memanggil *library* yang diperlukan dalam kode utama dalam file ini. Kemudian kode dibawah ini digunakan untuk menginisialisasi kedalam *database* yang terdapat di Firebase :

```
1. inisialisasi Database
2.
3. firebaseRef <= firebase database
```

Kode dibawah ini digunakan untuk penautan Action on Google dengan Firebase

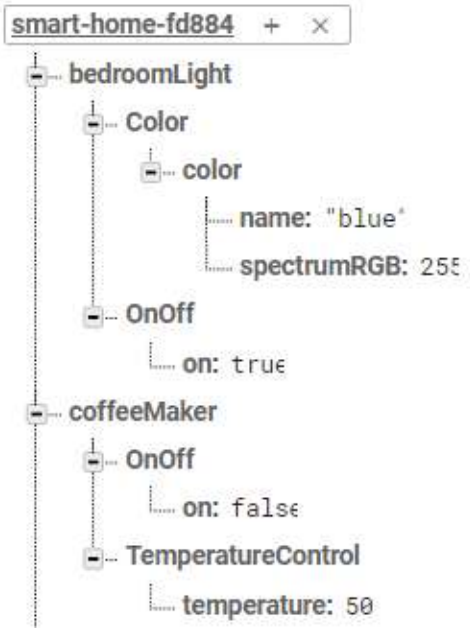
```
1. Penautan dengan Action on Google
2.
3. Fungsi HTTPS = require('actions-on-google')
4.
5. Eksprot Autorization KE fungsi HTTPS {
6.                    Respon => Util (format= CODE+STATUS)
7.                    }
8. Eksprot Token KE fungsi HTTPS {
9.                    Respon => Util (format= CODE+STATUS)
10.                   }
```

Untuk menambahkan dan memperbaharui perangkat yang digunakan dalam *voice recognition* dapat menggunakan kode dibawah ini :

```
1. Sync PERANGKAT
2.
3. smarthome = require('actions-on-google')
4.
5. smarthome.onSync (body) => (devices)
6. Devices = [id, type, name, traits, attributes]
```

Kode diatas untuk menambahkan perangkat pada sistem, penambahan perangkat menggunakan *array* yang ditulis deperti pada contoh *code* dengan menggunakan *<id>* untuk penamaan *database*, *<type>* untuk menentukan tipe perangkat yang ditambahkan, *<traits>* untuk mengatur perintah perangkat, dan *<name>* untuk penyebutan, *<attributes>* untuk memasukan spesifikasi perintah

Dengan menggunakan kode tersebut maka dapat didapatkan struktur *database* Firebase sebagai berikut :

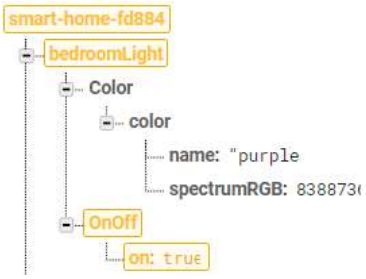


Gambar 3. 8 Struktur *Database* 1

Database yang didapatkan mempunyai nilai bilangan *Boolean*, untuk memperbaharui *database* ketika mendapatkan perintah dari GHM, dapat dituliskan menggunakan kode dibawah ini :

```
1. Excecute PERANGKAT
2.
3. smarthome = require('actions-on-google')
4.
5. smarthome.onExecute(body) => {payload[commands[id,
   status, states]]}
6.
7.   for (input DARI body.inputs) {
8.     for (const command DARI input.payload.commands)
   {
9.       for (const device DARI command.devices) {
10.         const deviceId = device.id;
11.         Update commands KE database
12.         for (const execution DARI command.execution)
    {
13.           const execCommand = execution.command;
14.           const {params} = execution;
15.           switch (execCommand) {
16.             case 'action.devices.commands.OnOff':
17.                 Update Database pada ON/OFF
18.             case 'action.devices.commands.ColorAbsol
   ute':
19.                 Update Database pada perubahan warna
20.             }
21.         }
22.       }
23.     }
24.   }
25. });
```

Kode diatas menggunakan *device commands* yang digunakan untuk memperbaharui *database* pada *child* <Color> dan <OnOff> didalam struktur *database*, pada contoh kode diatas database yang diperbaharui terdapat dalam struktur *database* sebagai berikut :



Gambar 3. 9 Struktur *Database* 2

3.2.4 Penulisan Kode ArduinoIDE

Pada penulisan kode berikut ini diperlukan *software* ArduinoIDE yang telah terpasang *board* NodeMCU untuk menjalankannya. Penambahan *board* tersebut dengan cara klik <Tools> lalu pilih <Board:> kemudian pilih <NodeMCU 1.0 (ESP-12E Module)>. Diperlukan juga tambahan *library* ESP8266 untuk dapat terhubung ke jaringan *Wi-Fi* dan *library* FirebaseArduino untuk dapat membaca *database* pada Firebase. Penambahan *library* dengan cara klik <Sketch> kemudian pilih <Include Library> dan pilih <Add .ZIP Library> untuk menambahkan *library* berbentuk file .Zip.

Setelah file *library* berhasil ditambahkan, kemudian tulis kode program untuk memanggil *library* tersebut kedalam program. Penulisan kode ini bertujuan untuk menghubungkan *board* NodeMCU dengan *Wi-Fi* dan melakukan pembacaan *database* Firebase secara *reltime*. Penulisan kode pada ArduinoIDE sebagai berikut :

```
1. Masukan Library <ESP8266HTTPClient.h>
2. Masukan Library <ESP8266WiFi.h>
3. Masukan Library <FirebaseArduino.h>
```

Kode tersebut merupaka *library* yang akan digunakan pada kode ini. setelah itu masukan kode untuk pendefinisian dan pendeklarasian :

```
1. Deklasrasi dan Inisialisasi
2.
3. DEFINISIKAN FIREBASE_HOST
4. DEFINISIKAN FIREBASE_AUTH
5. DEFINISIKAN WIFI_SSID
6. DEFINISIKAN WIFI_PASSWORD
7.
8. DEFINISIKAN ON HIGH
9. DEFINISIKAN OFF LOW
```

Kode tersebut digunakan untuk pendefinisian dan pendeklarasian Relay yang akan digunakan. Setelah itu masukan kode untuk menghubungkan ke jaringan *Wi-Fi*

```
1. Menghubungkan dengan WI-Fi
2.
3. WiFi.begin(DEFINISI FIREBASE_HOST dan FIREBASE_AUTH);
4. while (Status WI-Fi != Conected) {
5.   Serial.print(".");
6. }
7. Tampilkan (IP Address);
8.
9. Firebase.begin(DEFINISI FIREBASE_HOST dan
   FIREBASE_AUTH);
10.  if (Gagal menghubungkan ke Firebase) {
11.    Tampilkan pesan error
12. }
```

Kode tersebut digunakan untuk menghubungkan dengan jaringan *Wi-Fi* yang tersedia dilokasi *board* NodeMCU. Lalu masukan kode dibawah :

```
1. pinMode(Nama Pin KE OUTPUT);
```

Kode tersebut digunakan untuk penjelasan pada pin Relay digunakan sebagai *output*. Kemudian masukan kode berikut :

```
1. Menampilkan nilai database
2. Tampilkan(Referensi Database);
```

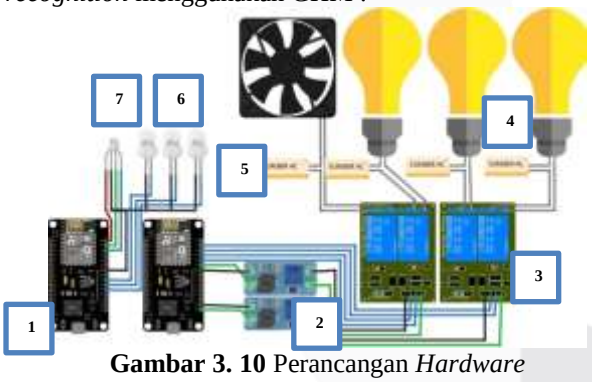
Kode tersebut digukanan untuk pembacaan *database* Firebase yang kemudian ditampilkan ke *serial monitor* yang terdapat pada ArduinoIDE, Pembacaan nilai data pada *serial monitor* dapat digunkan untuk pemantauan data saat melakukan pengujian. Data yang dibaca merupakan bilangan *boolean*. Setelah itu masukan kode dibawah ini :

```
1. Mengubah Kondisi Pin
2.
3.  if (Referensi Database == TRUE){
4.   SET (Nama PIN KE OFF)
5. }
6.  if (Referensi Database == FALSE){
7.   SET (Nama Pin KE ON)
8. }
```

Kode tersebut digunakan untuk algoritma pembacaan pada perubahan struktur *database* Firebase. Ketika Setelah semua kode ditulis lalu *upload* kode tersebut ke dalam *board* NodeMCU yang terpasang dengan komputer sesuai dengan *port* yang digunakan. Dan tunggu hingga kode berhasil di *upload*.

3.3 Perancangan Hardware

Berikut merupakan perancangan *hardware* pada sistem *voice recognition* menggunakan GHM :



Gambar 3. 10 Perancangan *Hardware*

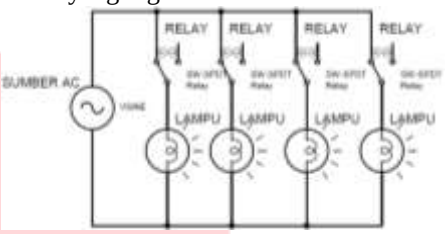
Pada **Gambar 3. 22** menunjukan alur perancangan sistem dengan keterangan sebagai berikut :

1. NodeMCU
2. MT3608 *Step Up Voltage*
3. Module Relay 5 *volt* 2 *Channel*
4. Beban Catuan
5. Sumber Catuan AC 220 *volt*
6. LED (*Light Emitting Diode*)
7. RGB LED (*Red-Green-Blue Light Emitting Diode*)

Perancangan *hardware* pada sistem ini menggunakan NodeMCU sebagai platform IoT yang digunakan dengan memanfaatkan ESP8266 untuk dapat terhubung dengan jaringan internet, yang nantinya pada NodeMCU terdapat kode program yang digunakan untuk membaca *database* Firebase. Dengan kemampuan pembacaan *database* tersebut, NodeMCU dapat mengendalikan keluaran melalui pin NodeMCU. Hasil dari keluaran tegangan yang dikendalikan dari pin NodeMCU dapat memicu kerja Relay yang telah terhubung, sehingga Relay dapat aktif dan tidak aktif.

Terdapat dua kondisi pada Relay yaitu *normally open* dan *normally close*, pada *normally open* yaitu kondisi Relay akan *open circuit* ketika tidak diberi tegangan catu, dan *normally close* yaitu kondisi Relay akan *close circuit* ketika tidak diberi tegangan catu. Dari kondisi *open circuit* dan *close circuit* inilah yang menyebabkan Relay dapat memutus dan menyambung rangkaian listrik yang terpasang pada Relay tersebut.

Dalam pembutan sistem ini diperlukan rangkaian listrik yang siap digunakan, yaitu kabel listrik yang telah terpasang dengan steker dan telah terhubung dengan fitting lampu. Agar lampu dapat bekerja dengan baik yaitu ketika lampu satu mati tidak mempengaruhi lampu yang lain maka pemasangan harus secara *paralel* antara Lampu satu dan lainnya, berikut merupakan rangkaian *paralel* yang digunakan dalam sistem ini :



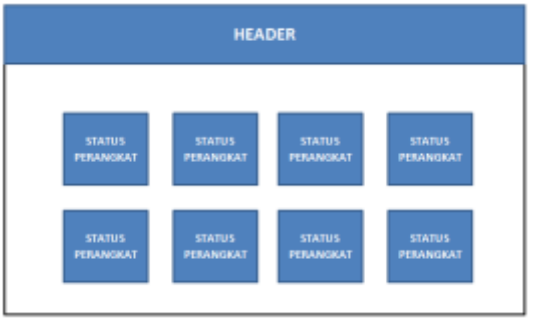
Gambar 3. 12 Rangkaian Paralel

Kemudian, proses perancangan *hardware* dimulai dari menghubungkan MT3608 dengan sumber tegangan keluaran dari NodeMCU kemudian putar potensio yang ada pada MT3608 hingga keluaran tegangan bernilai 5 *volt*, setelah itu hubungkan dengan tegangan catuan masukan pada Relay, setelah tegangan catu terhubung maka hubungkan Pin Relay dengan pin NodeMCU yang telah diatur sesuai kode dengan kode program. Setelah itu hubungkan rangkaian listrik dengan Relay, pada sistem kali ini rangkaian listrik pada beban (lampu) dipasangkan dengan Relay pada kondisi *normally close*.

Kemudian siapkan NodeMCU dan hubungkan Pin GPIO NodeMCU dengan Pin In pada Relay menggunakan kabel *jumper*. Setelah semua selesai kemudian pasangkan Lampu pada fitting lampu.

3.4 Perancangan Web *Monitoring*

Web *monitoring* digunakan untuk memantau kondisi alat yang terhubung dengan sistem, berikut merupakan desain tampilan Web yang digunkan untuk memantau perangkat :



Gambar 3. 13 *Desain Web*

Pada **Gambar 3.24** merupakan desain dari tampilan web, terdapat bagian <Status Perangkat> yang digunakan untuk melihat kondisi perangkat. Kondisi perangkat yang dipanatau meliputi keadaan menyala/mati, temperatur, kecepatan kipas, tingkat kecerahan, dan warna lampu.

Gambar 3. 11 Desain Status Perangkat

3.5 Analisis Kebutuhan *Software*

Dalam perancangan sistem ini dibutuhkan beberapa *software* yang yang diperlukan antara lain :

1. Firebase CLI

   Firebase CLI merupakan modul yang terdapat dalam *tool* NPM (*Node Package Manager*) yang memungkinkan kita mengelola mengelola *project* Firebase sehingga pengerjaan *project* menjadi lebih mudah.
2. Text Editor

   Text Editor merupakan software yang digunakan untuk mempermudah penulisan *script code.*
3. ArduinoIDE

   Arduino IDE adalah perangkat lunak *open-source* yang digunakan untuk menyusun, mengedit, dan mengunggah *script code* ke dalam papan Arduino sehingga dapat mempermudah penulisan *script code* yang akan diunggah ke papan Arduino. Pada sistem ini diperlukan tambahan *board* NodeMCU, *library* ESP8266, dan *library* FirebaseArduino.
4. Bootstrap

   Bootstrap merupakan *framework front-end open-source* yang dapat digunkan untuk mempermudah melakukan desain pada tampilan Web.

3.6 Skenario pengujian

Skenario pengujian dilakukan dengan cara menempatkan GHM pada suatu ruangan dan melakukan pengujian terhadap *noise*, dan terhadap *recognition*. Pengujian dilakukan untuk menilai performansi berdasarkan presentase keberhasilan pengujian dan *delay* waktu proses.

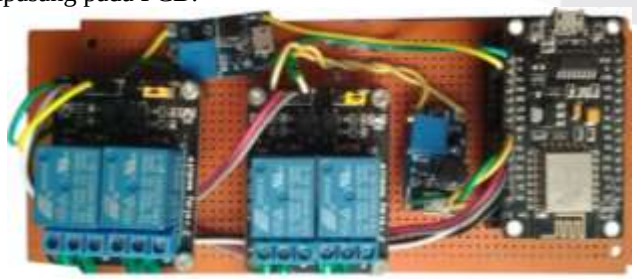
Pengujian terhadap jarak dilakukan dengan pempatan jarak GHM dan sumber suara dengan variasi jarak yang berbeda, pengujian terhadap *noise* dilakukan dengan melakukan pengujian memberikan perintah suara ke GHM dengan memberikan *noise* suara disekitar GHM, dan pengujian terhadap *recognition* dilakukan dengan memberikan perintah suara ke GHM dengan suara orang lain.

IV. PENGUJIAN DAN ANALISA SISTEM *VOICE RECOGNITION*

Pembuatan alat dilakukan berdasarkan perancangan yang telah dibuat sebelumnya. pembuatan alat dengan cara menghubungkan komponen dan *module* yang digunakan dengan kabel jumper, dan ditempatkan diatas PCB (*Printed Circuit Board*). Dengan ditempatkannya pada PCB komponen dapat saling dihubungkan.

4.1 Rangkaian Utama

Berikut merupakan gambar rangkaian utama sistem yang dipasang pada PCB:



Gambar 4. 1 Rangkaian Utama

Pada **Gambar 4.1** menggunakan Relay 2 *Channel* untuk mengendalikan perangkat elektronika dan NodeMCU yang digunakan untuk menghubungkan perangkat dengan jaringan *Wi-Fi*, agar nilai tegangan Relay terpenuhi maka digunakanlah MT3608 *Step Up* sebagai module yang digunakan untuk meningkatkan nilai tegangan, peningkatan nilai tegangan masukan berasal dari NodeMCU yang kemudian diteruskan ke Relay, sebelum menggunakan *module* MT3608 perlu adanya kalibrasi yang dilakukan dengan cara memutar potensio yang terdapat pada sisi module sambil mengukur nilai tegangan keluar sesuai nilai yang diinginkan, dengan demikian nilai tegangan pada Relay dapat terpenuhi.

Lampu dan kipas angin yang digunakan bekerja pada tegangan AC (*Alternaring Current*) 220 *volt*. Sedangkan untuk perangkat tambahan lainnya menggunakan LED (*Light Emitting Diode*).

4.2 Rangkaian Sistem Keseluruhan

Rangkaian secara keseluruhan merupakan tampilan dari keseluruhan sistem yang dibuat. Sistem ini terdiri dari rangkaian utama dan beban catuan seperti lampu, kipas angin dan perangkat elektronik lainnya. Berikut merupakan tampilan sistem secara keseluruhan :

Gambar 4. 2 Keseluruhan Sistem

Pada **Gambar 4.2** sistem ditempatkan pada sebuah papan yang telah dibentuk, sistem ini terdiri dari rangkaian utama sebagai pusat kendali perangkat dan juga menggunakan beban catuan seperti lampu dan *stopkontak*, yang mana *stopkontak* tersebut dapat dipasangkan dengan kipas angin, sehingga dapat mengendalikan kipas angin yang terhubung. Selain itu ada perangkat pengganti menggunakan LED sebagai pengganti perangkat lain.

Agar sistem dapat bekerja dengan baik yaitu ketika pengguna hanya ingin mengendalikan salah satu perangkat saja dan tidak ingin ada perangkat lain yang terganggu pemasangan beban catuan harus secara *paralel*, pemasangan *paralel* tersebut dapat dilihat pada G**ambar 3.11**.

Dalam pengujian ini dilakukan pengujian terhadap beberapa jenis pengujian yaitu pengujian terhadap intensitas suara, *noise*, *voice recognition* mesin, *voice recognition* manusia, fungsi, dan jaringan

4.3 Pengujian Terhadap Jarak Sumber Suara dengan Perangkat GHM

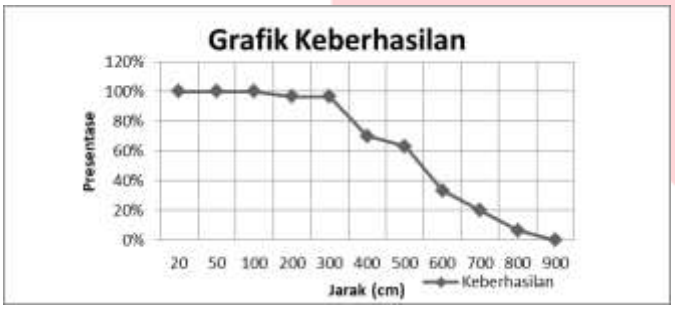
Pengujian ini dilakukan dengan meletakan GHM pada suatu tempat dan memberikan perintah suara berdasarkan jarak yang berbeda, pengujian ini dilakukan sebanyak masing masing 30 kali percobaan dengan jarak yang bervariasi. Berikut merupakan tabel pengujian terhadap jarak :

Tabel 4 Pengujian Terhadap Jarak

No	Jarak	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	20 cm	100 %	4.00
2	50 cm	100 %	3.77
3	100 cm	100 %	4.13

4	200 cm	96.6 %	4.24
5	300 cm	96.6 %	3.85
6	400 cm	70.0 %	4.28
7	500 cm	63.3 %	3.91
8	600 cm	33.3 %	3.73
9	700 cm	20.0 %	3.91
10	800 cm	6.6 %	4.14
11	900 cm	0.0 %	-

Dari **Tabel 4** menunjukan hasil pengujian terhadap jarak, tingkat akurasi sistem akan menurun ketika telah melewati jarak 100 cm, dan perangkat tidak dapat menangkap perintah suara ketika telah mencapai jarak 900 cm. Perincian hasil pengujian terdapat pada **Lampiran A**.



Gambar 4. 3 Grafik Pengujian Terhadap Jarak

4.4 Pengujian Terhadap Intensitas Suara

Pengujian ini dilakukan dengan meletakan GHM pada suatu tempat dan memberikan perintah suara ke GHM dengan mengukur nilai intensitas suara (dB) yang terdapat pada sekitar GHM. Intensitas suara yang diberikan bervariasi dari 40dB hingga 70dB.

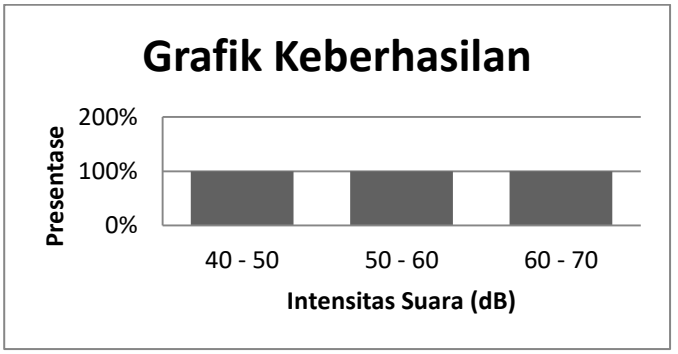
Gambar 4. 4 Skema Pengujian Intensitas Suara

Pengujian dilakukan sebanyak 30 kali setiap pembagian intensitas. Berikut merupakan tabel penguijan terhadap intensitas suara :

Tabel 5 Pengujian Terhadap Intensitas Suara yang dikeluarkan

No	Intensitas Suara (dB)	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	40 – 50	100%	3.97
2	50 – 60	100%	4.20
3	60 – 70	100%	4.00

Dari **Tabel 5** menunjukan hasil pengujian terhadap intensitas suara dengan rentang 40dB hingga 70dB didapatkan hasil dengan tingkat keberhasilan 100% dengan delay waktu rata – rata selama 4.05 detik. Perincian hasil percobaan dapat dilihat pada halaman Lampiran.



Gambar 4. 5 Grafik pengujian terhadap intensitas suara

Pengujian terhadap intensitas suara tidak dapat dilakukan dibawah ±40dB dikarenakan tidak adanya ruangan kedap suara yang memadai, maka kondisi alat ukur yang digunakan membaca kebisingan paling kecil berada pada ±40dB, dan pengujian tidak dapat dilakukan diatas ±70dB dikarenakan pada saat pengujian tidak dapat membuat suara diatas ±70dB.

Apliaksi yang digunakan untuk mengukur tingkat intensitas suara bernama "Sound Meter" yang dapat diunduh melalui Play Store (pengguna Android).

Gambar 4. 6 Sound Meter

4.5 Pengujian Terhadap *Noise*

Pengujian ini dilakukan dengan meletakan GHM pada suatu tempat yang terdapat *noise* di sekitar perangkat GHM, *noise* yang digunakan berupa suara musik dengan tingkat intensitas antara 40dB – 70dB, dan memberikan perintah suara pada intensitas antara 40dB-50dB.

Gambar 4. 7 Skema Pengujian Terhadap *Noise*

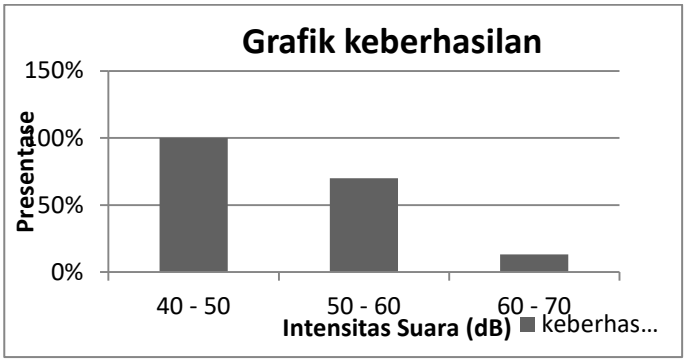
Pengujian ini dilakukan sebanyak 30 kali dari tiap pembagian intensitasnya. Berikut merupakan tabel pengujian *noise* :

Tabel 6 Pengujian Terhadap *Noise*

No	Noise (dB)	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	40 – 50	100%	3.87
2	50 – 60	70.0%	4.14
3	60 - 70	13.3%	5.24

Dari **Tabel 6** menunjukan hasil pengujian terhadap *noise*, tingkat akurasi sistem mendapatkan presentase keberhasilan percobaan sebesar 100% pada intensitas suara *noise* 40dB – 50dB, 70% pada intensitas suara *noise* 50dB – 60dB, dan 13.3% pada intensitas suara *noise* 60dB – 70dB dengan delay waktu 4.41 detik dari total percobaan sebanyak 30 kali pada tiap

pembagian intensitasnya. Perincian hasil percobaan dapat dilihat pada halaman Lampiran.



Gambar 4. 8 Grafik Pengujian Terhadap *Noise*

4.6 Pengujian Terhadap *Voice Recognition* Mesin

Pengujuan ini dilakukan dengan memberikan perintah suara dari kata yang diucapkan oleh mesin *text-to-speech*, dengan cara menuliskan perintah masukan yang kemudian dibacakan perintah oleh mesin, penggunaan *text-to-speech* yang digunakan dalam pengujian ini memanfaatkan Google Translate.

Dengan memanfaatkan Google Translate sebagai media *text-to-speech* bisa didapatkan 2 jenis pembacaan yaitu pembacaan normal dan pembacaan pengejaan (lambat), dengan demikian dapat dilakukan pengujian dengan 2 intonasi kecepatan yang berbeda dengan pelafalan yang jelas. *Text-to-speech* juga digunakan untuk mengurangi pelafalan salah yang berasal dari manusia. Pengujian dilakukan sebanyak 30 kali tiap masing masing jenisnya.

Gambar 4. 9 Skema Pengujian *Voice Recognition* dengan mesin

Pengujian pertama dilakukan dengan menguji setiap fungsi penggunaan perintah dengan intonasi normal yang terdapat pada *text-to-speech* Google Translate. Berikut merupakan data pengujian terhadap intonasi yang normal :

Tabel 7 Pengujian Intonasi Normal

No	Intonasi Normal pada perintah	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	*Turn the device on or off*	100%	4.13
2	*Set the device's temperature*	0.0%	-
3	*Adjust the LED color*	100%	3.94
4	*Adjust the fan speed*	100%	4.17
5	*Adjust the LED brightness level*	100%	4.28

Dari **Tabel 7** didapatkan hasil pengujian dengan memanfaatkan *text-to-speech* Google Translate dengan intonasi normal, dari data menunjukan tingkat keberhasilan 100% dengan rata – rata delay selama 4.38 detik kecuali pada perintah pengaturan temperatur tidak dapat dikerjakan menggunakan intoasi normal pada *text-to-speech* Google Translate.

Pengujian kedua dilakukan dengan menguji setiap fungsi penggunaan perintah dengan intonasi ejaan (lambat) yang terdapat pada *text-to-speech* Google Translate. Berikut merupakan pengujian terhadap intonasi ejaan :

Tabel 8 Pengujian Intonasi Ejaan

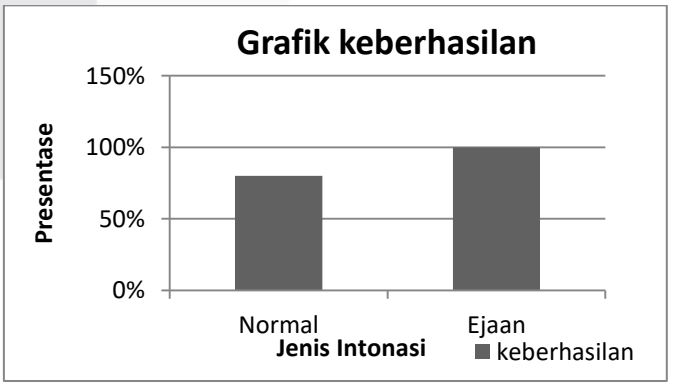
No	Intonasi Ejaan pada Perintah	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	Turn the device on or off	100%	4.13
2	Set the device's temperature	100%	4.43
3	Adjust the LED color	100%	4.35
4	Adjust the fan speed	100%	4.32
5	Adjust the LED brightness level	100%	4.15

Dari **Tabel 8** didapatkan hasil pengujian dengan memanfaatkan *text-to-speech* Google Translate dengan intonasi ejaan, dari data menunjukan hasil presentase keberhasilan sebesar 100% dengan rata – rata delay selama 4.28 detik. Dengan demikian didapatkan hasil perbandingan antara intonasi normal dan intonasi ejaan menggunakan bahasa mesin

Tabel 9 Perbandingan Intonasi

No	Intonasi Ejaan pada Perintah	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	Turn the device on or off	100%	4.13
2	Set the device's temperature	100%	4.43

Dari **Tabel 9** didapatkan perbandingan hasil pengujian terhadap 2 jenis intonasi yang berbeda melalui pengucapan mesin, dari data tersebut didapatkan hasil presentase sebesar 80% untuk pengujian dengan intonasi normal dan mendapatkan hasil presentase sebesar 100% untuk pengujian dengan intonasi ejaan dengan total delay rata – rata selama 4.33 detik



Gambar 4. 10 Grafik Perbandingan Intonasi Normal dan Ejaan

4.7 Pengujian Terhadap *Voice Recognition* Manusia

Pengujuan ini dilakukan dengan memberikan perintah suara ke perangkat GHM dengan suara orang yang berbeda. Perincian pengujian dilakukan pada suara dengan rentang usia 5 – 12 tahun, 12 – 18 tahun, dan diatas 18 tahun.

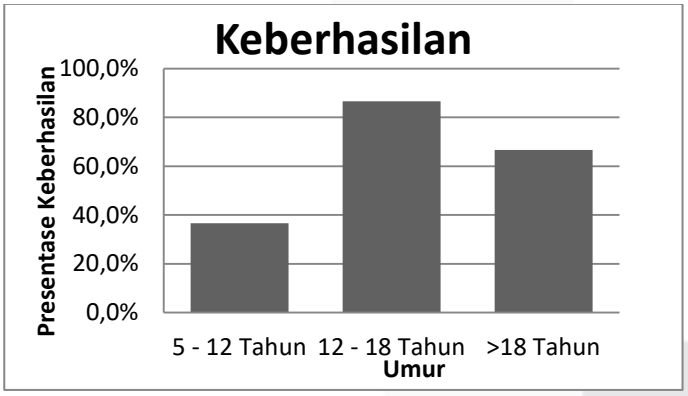
Gambar 4. 5 Skema Pengujian *Voice Recognition* dengan manusia

Pengujian ini dilakukan sebanyak 30 kali pada masing – masing perinciannya. Berikut merupakan tabel pengujian terhadap *recognition* :

Tabel 10. Pengujian Terhadap *Recognition*

No	Recognition umur Manusia (Tahun)	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	5 – 12	36.6%	4,11
2	12 – 18	86,6%	4,14
3	>18	66.6%	3.97

Dari **Tabel 9** menunjukan hasil pengujian dengan mendapatkan presentase keberhasilan pengujian sebesar 36.6% untuk pengujian terhadap rentang usia 5 – 12 Tahun, 86.6% untuk pengujian terhadap rentang usia 12 – 18 Tahun, dan 66.6% untuk usia >18 Tahun dengan didapatkan rata – rata *delay* 4.07 detik dari total pengujian sebanyak 30 kali pada masing masing perinciannya. Perincian hasil percobaan dapat dilihat pada halaman Lampiran.



Gambar 4. 6 Grafik Pengujian *Recognition*

4.8 Pengujian Fungsi Sistem

Pengujuan ini dilakukan dengan mencoba semua perintah untuk mengendalikan perangkat elektronik melalui *voice recognition* yang telah didaftarkan.

Gambar 4. 7 Skema Pengujian Fungsi Sistem

Pengujian ini dilakukan sebanyak 30 kali siap pembagiannya. Berikut merupakan data pengujian terhadap fungsi :

Tabel 5 Pengujian Terhadap Fungsi Sistem

No	Fungsi perangkat	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	*Turn the device on*	100%	3.81
2	*Turn the device off*	100%	3.97
3	*Set the device's temperature*	100%	4.00
4	*Adjust the LED color*	100%	4.04
5	*Adjust the fan speed*	100%	4.06
6	*Adjust the LED brightness level*	100%	4.07

Dari **Tabel 11** menunjukan data pengujian terhadap setiap fungsi perintah yang berbeda, dari pengujian tersebut didapatkan hasil pengujian sebesar 100% degan delay total rata – rata selama 4.02 detik. Perincian hasil percobaan dapat dilihat pada halaman Lampiran.

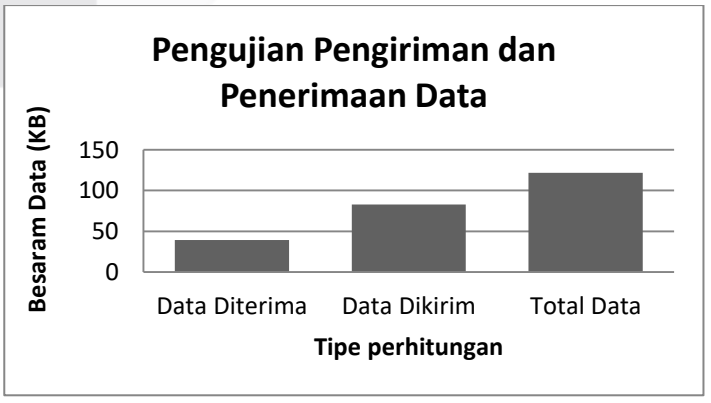
4.9 Pengujian Pengiriman dan Penerimaan Data

Pengujuan ini dilakukan dengan menghitung besar data yang dikirim dan diterima dalam 1 kali pemberian perintah, pengijian dilakukan sebanyak 30 kali percobaan dengan memperhatikan besaran data yang dikirim dan diterima. Perhitungan penggunaan data menggunakan aplikasi bernama "3G Watchdog" yang dapat diunduh melalui aplikasi Play Store (pengguna Android).

Tabel 6 Pengujian Terhadap Pengiriman dan Penerimaan data

No	Pemberian Perintah	Rata – Rata Data Diterima (KB)	Rata Rata Data Dikirim (KB)	Rata – Rata Total (KB)
1	Perhitungan Data	39.2	82.7	121.9

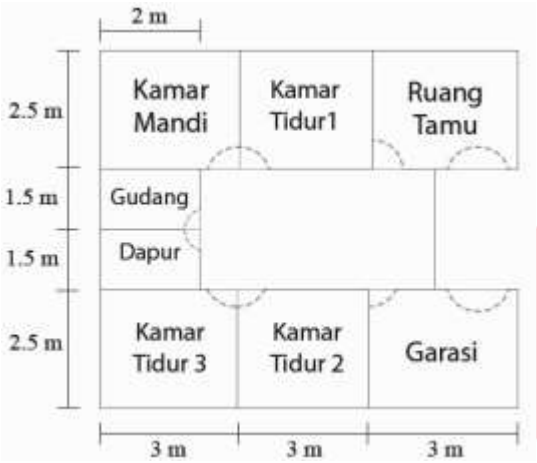
Dari **Tabel 12** menunjukan hasil pengujian pengiriman dan penerimaan data dengan dilakukan sebanyak 30 kali percobaan dan didapatkan hasil rata – rata besar data diterima untuk 1 perintah adalah 39.2KB dan rata – rata besar data yang dikirimkan sebesar 82.7KB dengan rata – rata total sebesar 121.9KB. Perincian hasil perngujian dapat dilihat pada halaman lampiran.



Gambar 4. 8 Grafik Pengiriman dan Penerimaan data

4.10 Pengujian di Ruangan Berbeda

Pengujuan ini dilakukan dengan meletakan perangkat GHM pada sebuah ruangan dan memberikan perintah suara dari ruangan lain yang berdekatan, terdapat 8 ruangan yang berbeda, pengujian dilakukan ke semua ruangan yang ada. Berikut merupakan denah rumah yang digunakan untuk pengujian :



Gambar 4. 9 Denah Rumah Pengujian

Penempatan perangkat GHM pada masing - masing ruangan terdapat pada **Lampiran B**, pengujian ini dilakukan sebanyak 20 kali dari masing – masing ruangan. Berikut merupakan hasil pengujian dari setiap ruangan.

Tabel 7 Pengujian Terhadap Ruangan Berbeda

No	Nama Ruangan	Presentase Keberhasilan	Rata – Rata Waktu Proses (Detik)
1	Kamar Tidur 1	0%	0
2	Ruang Tamu	0%	0
3	Garasi	0%	0
4	Kamar Tidur 2	0%	0
5	Kamar Tidur 3	0%	0
6	Dapur	5%	4.30
7	Gudang	5%	4.03
8	Kamar Mandi	0%	0

Dari **Tabel 13** menunjukan hasil pengujian terhadap ruangan yang berbeda, dari pengujian tersebut didapatkan hasil dengan presentase keberhasilan sebesar 0% pada semua ruangan kecuali dapur dan gudang mendapat presentase 5%. Perincian hasil pengujian terdapat pada **Lampiran A**.

4.11 Pengujian Terhadap Jaringan

Pengujuan ini dilakukan dengan menghubungkan perangkat GHM dengan jaringan Wi-Fi yang berbeda dengan Google Assistant diperangkat *mobile*.

Tabel 8 Pengujian Jaringan

No	Jenis Jaringan	Terhubung / Tidak Terhubung
1	Jaringan seluler sendiri	Tidak dapat terhubung
2	Menggunakan *Wi-Fi* dari perangkat luar dengan jaringan yang sama	Dapat terhubung
3	Menggunakan *Wi-Fi* dari perangkat luar dengan jaringan yang berbeda	Tidak dapat terhubung

Dari **Tabel 14** menunjukan hasil pengujian terhadap penggunaan jaringan yang berbeda antara perangkat GHM dengan Google Assistant pada perangkat mobile, hasil pengujian menunjukan penggunaan sistem ini hanya bisa digunakan dengan jaringan Wi-Fi dari perangkat luar dengan jaringan yang sama.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan data yang telah diujicobakan pada sistem *voice recognition* menggunakan GHM yang telah dibuat, dapat ditarik kesimpulan sebagai berikut :

1. Jarak mempengaruhi performansi sistem *voice recognition* menggunakan GHM, dari pengujian yang dilakukan sebanyak 30 kali percoban pada masing – masing jarak didapatkan hasil pengujian pada jarak 0 cm – 100 cm mendapatkan tingkat keberhasilan pengujian mencapai 100% dan akan menurun hingga pada jarak 900 cm tingkat keberhasilan pengujian 0%.
2. Pengujian intensitas suara pada rentang 40dB – 70dB mendapatkan presentase keberhasilan sebesar 100% yang dilakukan dengan total 90 kali percobaan dengan rata – rata delay 4.05 detik.
3. *Noise* yang diberikan pada perangkat GHM mempengaruhi performansi sistem dengan tingkat keberhasilan 100% pada 40dB – 50dB dengan rata – rata delay 3.87 detik, dan mendapatkan hasil 70% pada 50 dB – 60dB dengan rata – rata delay 4.14 detik, dan mendapatkan hasil 13.33% pada 60dB – 70dB dengan rata – rata delay 5.24 detik.
4. Berdasarkan data yang telah dilakukan pengujian sebanyak masing – masing 30 kali pada pengujian *voice recognition* terhadap mesin, didapatkan hasil presentase keberhasilan sebanyak 80% dengan delay 4.38 detik untuk intonasi normal dan mendapatkan hasil 100% dengan delay rata – rata 4.28 detik untuk intonasi ejaan.
5. Berdasarkan data yang telah dilakukan pengujian sebanyak masing – masing 30 kali pada pengujian *voice recognition* terhadap manusia, didapatkan hasil presentase keberhasilan sebanyak 36.6% dengan delay 4.11 detik untuk suara umur 9 – 12 Tahun dan mendapatkan hasil 86.6% dengan delay rata – rata 4.14 detik untuk suara 12 – 18 Tahun, dan 66.6% untuk umur >18 Tahun dengan rata – rata delay 3.97 detik.

6. Berdasarkan data pengujian terhadap fungsi sistem yang telah dilakukan didapatkan hasil 100% untuk semua perintah yang digunakan dengan delay rata – rata selama 4.02 detik.
7. Berdasarkan data pengujian terhadap pengiriman dan penerimaan data, dapat diketahui sistem mengirimkan data dengan rata – rata sebesar 82.7KB, dan menerima data dengan rata – rata sebesar 39.2KB, dengan total sebesar 121.9KB.
8. Berdasarkan data pengujian terhadap pengiriman dan penerimaan data, sistem mendapat hasil 0% keberhasilan pada semua ruangan kecuali ruangan Dapur dan Gudang, dengan presentase keberhasilan sebesar 5%.
9. Berdasarkan pengujian pada jaringan, sistem hanya dapat terhubung dengan jaringan Wi-Fi yang sama antara perangkat GHM dan Google Assistant yang terdapat pada perangkat *mobile*.

5.2 Saran

Saran yang diberikan setelah pembuatan sistem *voice recognition* menggunakan GHM untuk pengendalian perangkat elektronik yaitu diperlukan penggunaan sistem yang dapat terhubung melalui jaringan yang berbeda.

DAFTAR PUSTAKA

[1] *Action on Google*, Support, Google. (2019) dari https://developers.google.com/actions/

[2] Anusuya, M. A. and S. K. Katti. 2009. "*Speech Recognition by Machine : A Review*" 6(3):181–205.

[3] *Arduino 1.8.9*, Arduino. (2019). dari https://www.arduino.cc/en/Main/Software

[4] Aziz, Dlnya Abdulahad. 2018. "*Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module*" 9(6):6.

[5] Buest. Baddi. 2017. "*Working, Operation And Types Of Arduino Microcontroller*" 6(6):155–58.

[6] Chetan, D. Vanishree, K. 2019. "*Benefits of CRM Application and Tech Stack*" 7(02):689–90.

[7] Das, Prerana, Kakali Acharjee, Pranab Das, and Vijay Prasad. 2016. "*Voice Recognition System : Speech-To-Text*" 01(2):2395–5562.

[8] Dasgupta, Poorna Banerjee. 2017. "*Detection and Analysis of Human Emotions through Voice and Speech Pattern Processing*" 52(1):1–3.

[9] *Firebase CLI Reference*, Support, Google. (2019) dari https://firebase.google.com/docs/cli

[10] George, Jobin, Anila Cyril, Bino I. Koshy, and Leena Mary. 2013. "*E Xploring S Ound S Ignature For V Ehicl*." 4(2):29–36.

[11] *Google Home Specifications*, Support, Google. (2018). dari https://support.google.com/googlehome/answer/7072284?hl=en&ref_topic=7195843

[12] Jadhav, Ishwar S., V. T. Gaikwad, and Gajanan U. Patil. 2011. *"Human Identification Using Face and Voice Recognition*" 2(3):1248–52.

[13] Kaladharan, N. 2015. "*A Study of Speech Recognition*" 03(9):8030–34.

[14] Kevin, Omyonga and Kasamani Bernard Shibwabo. 2015. "*The Application of Real-Time Voice Recognition to Control Critical Mobile The Application of Real-Time Voice Recognition to Control Critical Mobile Device Operations*" 2(7):174–84.

[15] Khawas, Chunnu. 2018. "*Application of Firebase in Android App Development-A Study*" 179(46):49–53.

[16] Khedkar, Sonam. 2017. "*Real Time Databases for Applications*" 04(06):2078–82.

[17] Mirowski, Piotr and New York. 2014. "*Feature-Rich Continuous Language Models For Speech Recognition Feature-Rich Continuous Language Models For Speech Recognition*" 01(01):3–6.

[18] Nizam, Hairol, Mohd Shah, Mohd Zamzuri, Ab Rashid, and Mohd Fairus Abdollah. 2014. "*Biometric Voice Recognition in Security System*" 07(2):104–12.

[19] Saini, Preeti and Parneet Kaur. 2013. "*Automatic Speech Recognition : A Review*." 4(2):132–36.

[20] Shurti, Joshi., Kumari. Aarti, and Sangaonkar Pooja, Pai. Sainesh. 2017. "*Voice Recognition System*" 03(01):6–9.

[21] Strik, Helmer and Catia Cucchiarini. 1999. "*Modeling Pronunciation Variation for ASR : A Survey of the Literature*" 29:225–46.

[22] Sunitha, S. 2017. "*Distance Measurement Using Ultrasonic Sensor and NodeMCU*" 4(06):1794–97.

[23] Trivedi, Ayushi, Navya Pant, Pinal Shah, Simran Sonik, and Supriya Agrawal. 2018. "*Speech to Text and Text to Speech Recognition Systems-Areview*" 20(2):36–43.

[24] Vanaja, K. Jyostsna, Aala Suresh, S. Srilatha, K. Vijay Kumar, and M. Bharath. 2018. "*IOT Based Agriculture System Using NodeMCU*" 05(03):3025–28.

[25] Yadav, Deepika and B. Tech Student. 2018. "*Controlling Of Relay Using Raspberry Pi Via Internet For Home*" 9(1):1–11.