

**PROTOTYPE SISTEM KONTROL DAN MONITORING  
PERGERAKAN WIPER, STARTER, DAN KACA JENDELA  
PADA MOBIL LISTRIK MENGGUNAKAN PERINTAH  
SUARA DENGAN SENSOR BLUETOOTH DAN WIFI  
BERBASIS IOT**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan  
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Fajri                      NIRM: 0031908

Rizkia Meilani        NIRM: 0031924

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2022**

**LEMBAR PENGESAHAN**

**JUDUL PROYEK AKHIR**

**PROTOTYPE SISTEM KONTROL DAN MONITORING PERGERAKAN  
WIPER, STARTER, DAN KACA JENDELA PADA MOBIL LISTRIK  
MENGUNAKAN PERINTAH SUARA DENGAN SENSOR BLUETOOTH  
DAN WIFI BERBASIS IOT**

Oleh:

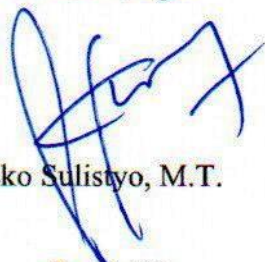
Fajri /0031908

Rizkia Meilani/0031924

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan  
Diploma III Politeknik Manufaktur Negeri Bangka Belitung

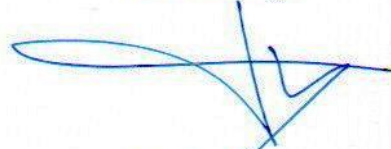
Menyetujui,

Pembimbing 1



Eko Sulistyono, M.T.

Pembimbing 2



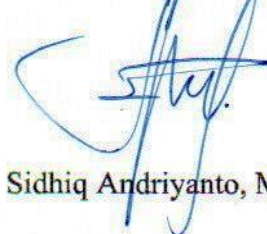
Surojo, M.T.

Penguji 1



Indra Dwisaputra, M.T.

Penguji 2



Sidhiq Andriyanto, M.Kom

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1: Fajri NIRM: 0031908

Nama Mahasiswa 2: Rizkia Meilani NIRM: 0031924

Dengan Judul :PROTOTYPE SISTEM KONTROL DAN MONITORING PERGERAKAN WIPER, STARTER, DAN KACA JENDELA PADA MOBIL LISTRIK MENGGUNAKAN PERINTAH SUARA DENGAN SENSOR BLUETOOTH DAN WIFI BERBASIS IOT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 12 Agustus 2022

Nama Mahasiswa

Tanda Tangan

1. Fajri



2. Rizkia Meilani



## ABSTRAK

*Pengontrolan peralatan menggunakan perintah suara masih belum banyak diterapkan pada mobil listrik. Umumnya peralatan mobil listrik masih banyak dilakukan pengontrolan secara elektrik. Untuk pengoptimalan pengontrolan peralatan tersebut perlu dilakukan pengembangan pengontrolan melalui perintah suara via smartphone. Oleh karena itu, dibutuhkan sebuah sistem kontrol mobil listrik menggunakan perintah suara dan monitoring pada peralatan mobil listrik seperti starter, wiper, dan kaca jendela melalui aplikasi SKMML pada smartphone. Metodologi yang dilakukan dalam penelitian ini yaitu pengubahan perintah suara melalui speech recognizer yang terdapat pada aplikasi SKMML yang sudah dibuat, pengujian respon peralatan mobil listrik ketika diberikan perintah suara berdasarkan kode perintah yang telah ditentukan, serta memantau perubahan data sensor tegangan pada masing-masing peralatan. Dari hasil pengujian ketika peralatan mobil listrik starter, wiper, dan kaca jendela diberikan perintah suara mempunyai persentase keberhasilan sebesar 100% untuk kondisi tanpa kebisingan dan persentase keberhasilan 96% untuk kondisi dengan kebisingan. Durasi respon terhadap perintah suara tersebut selama 4 detik. Berdasarkan hasil pengujian yang telah dilakukan, alat ini dapat diterapkan untuk pengontrolan peralatan mobil listrik menggunakan perintah suara.*

*Kata kunci: perintah suara, mobil listrik, aplikasi SKMML*

## **ABSTRACT**

*Controlling equipment using voice commands is still not widely applied to electric cars. Generally, electric car equipment is still mostly controlled electrically. To optimize the control of this equipment, it is necessary to develop control via voice commands via smartphones. Therefore, we need an electric car control system using voice commands and monitoring of electric car equipment such as starters, wipers, and window panes through the SKMML application on a smartphone. The methodology used in this research involves changing voice commands through the speech recognizer contained in the SKMML application that has been developed, testing the response of electric car equipment when given a voice command based on a predetermined command code, and monitoring changes in voltage sensor data on each piece of equipment. From the test results, when the electric car starter, wiper, and window glass are given voice commands, the percentage of success is 100% for the condition without noise, and the percentage of success is 96% for the condition with noise. The duration of the response to the voice command is 4 seconds. Based on the results of the tests that have been carried out, this tool can be applied to control electric car equipment using voice commands.*

*Keyword: voice command, electric car, SKMML application*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan.

Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 3 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orangtua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Eko Sulistyono, M.T. selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung sekaligus dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Surojo, M.T. selaku dosen pembimbing 2 dalam proyek akhir ini.

5. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Ocsirendi, M.T. selaku Ketua Program Studi DIII Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir.
9. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 31 Juli 2022

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
BAB II LANDASAN TEORI.....	4
2.1 Pengertian Mobil Listrik.....	4
2.2 Studi Literatur .....	4
2.3 Speech Recognition (Pengenalan Kata).....	5
2.4 Arduino Mega 2560.....	6
2.5 Modul Bluetooth HC -05 .....	7
2.6 NodeMCU ESP8266 .....	8
2.7 Motor DC.....	9
2.8 Driver Motor L298N .....	10
2.9 Sensor Tegangan DC.....	11



2.10 Aki (Accumulator).....	12
2.11 Arduino IDE.....	12
2.12 MIT App Inventor .....	13
2.13 <i>Firebase</i> .....	13
2.14 IoT .....	14
<b>BAB II METODE PELAKSANAAN.....</b>	<b>16</b>
3.1 Pengumpulan dan Pengolahan Data.....	18
3.2 Perancangan <i>Software</i> dan <i>Hardware</i> .....	19
3.2.1 <i>Desain Software</i> .....	19
3.2.2 <i>Desain Hardware</i> .....	19
3.3 Pembuatan <i>Software</i> dan <i>Hardware</i> .....	20
3.3.1 <i>Pembuatan Software</i> .....	20
3.3.2 <i>Pembuatan Hardware</i> .....	21
3.4 Pengujian <i>Hardware</i> dan <i>Software</i> .....	21
3.4.1 <i>Pengujian Hardware</i> .....	22
3.4.2 <i>Pengujian Software</i> .....	22
3.4.3 <i>Pengujian Keseluruhan Sistem</i> .....	22
3.5 Analisis Data .....	22
3.5 Pembuatan Makalah Proyek Akhir.....	23
<b>BAB IV PEMBAHASAN.....</b>	<b>24</b>
4.1 Deskripsi Alat.....	24
4.2 Perancangan dan pembuatan <i>Software</i> .....	25
4.2.1 Perancangan dan Pembuatan Aplikasi.....	25
4.2.2 Perancangan dan Pembuatan Perintah Suara.....	34
4.2.3 Perancangan dan Pembuatan <i>Database</i> .....	36

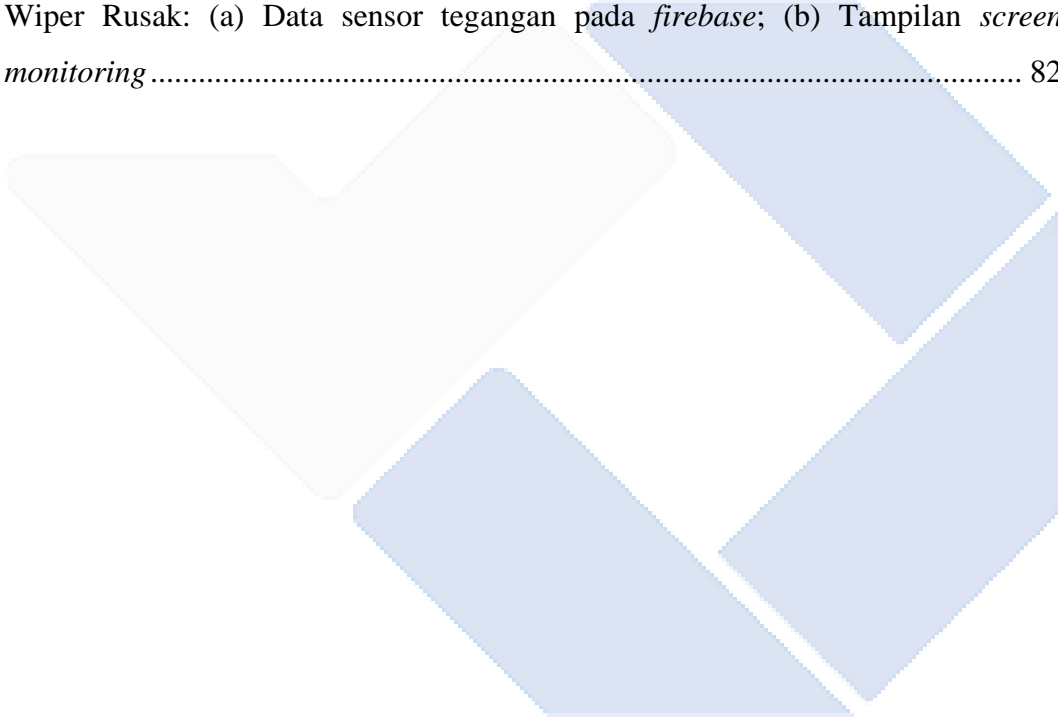
4.2.4 Komunikasi Serial pada <i>Platform</i> MIT App Inventor.....	36
4.2.5 Koneksi Modul Bluetooth HC-05 dengan MIT App Inventor .....	37
4.2.6 Menghubungkan <i>Database</i> dengan MIT App Inventor.....	40
4.3 Perancangan dan Pembuatan <i>Hardware</i> .....	41
4.3.1 Merancang <i>Hardware</i> secara Mekanik.....	41
4.3.2 Pembuatan <i>Hardware</i> secara Mekanik.....	44
4.3.3 Merancang dan Membuat <i>Hardware</i> Elektrik.....	46
4.4 Pengujian <i>Hardware</i> Elektrik .....	47
4.4.1 Pengujian Pengiriman Perintah Suara Melalui Modul Bluetooth .....	47
4.4.2 Pengujian Sensor Tegangan.....	55
4.4.3 Pengujian Komunikasi Serial NodeMCU ESP8266.....	59
4.5 Pengujian Keseluruhan Alat .....	63
4.5.1 Pengujian Pengaruh Jarak Sumber Suara ke Smartphone .....	63
4.5.2 Pengujian Pengontrolan <i>Starter</i> dengan Perintah Suara.....	66
4.5.3 Pengujian Pengontrolan Pergerakan Naik Turun Kaca Jendela dengan Perintah Suara.....	71
4.5.4 Pengujian Pengontrolan Pergerakan <i>Wiper</i> Menggunakan Perintah Suara. ....	75
4.5.5 Pengujian Monitoring Data sensor Tegangan Pada Aplikasi SKMML .....	80
<b>BAB V PENUTUP</b> .....	<b>84</b>
5.1 Kesimpulan .....	84
5.2 Saran .....	84

## DAFTAR GAMBAR

Gambar 2. 1 Arduino Mega 2560 .....	6
Gambar 2. 2 Modul Bluetooth HC-05 .....	8
Gambar 2. 3 NodeMCU ESP8266 .....	9
Gambar 2. 4 Motor DC Gear Box .....	10
Gambar 2. 5 Driver L298N .....	10
Gambar 2. 6 Sensor Tegangan DC .....	11
Gambar 2. 7 Aki 12V .....	12
Gambar 2. 8 Tampilan <i>Software</i> Arduino .....	12
Gambar 2. 9 Tampilan <i>Software</i> MIT App Inventor .....	13
Gambar 2. 10 Tampilan <i>Software</i> Firebase .....	14
Gambar 2. 11 Logo IoT [22] .....	14
Gambar 4. 1 Blok Diagram Cara Kerja Alat .....	24
Gambar 4. 2 Desain <i>Screen</i> Beranda Aplikasi SKMML .....	26
Gambar 4. 3 Tampilan <i>Screen</i> Beranda pada <i>Smartphone</i> User .....	27
Gambar 4. 4 Desain <i>Screen</i> Portal Login Aplikasi SKMML .....	28
Gambar 4. 5 Tampilan <i>Screen</i> Portal Login pada <i>Smartphone</i> .....	29
Gambar 4. 6 Desain <i>Screen</i> <i>Voice Control</i> Aplikasi SKMML .....	30
Gambar 4. 7 Tampilan <i>Screen</i> <i>Voice Control</i> pada <i>Smartphone</i> .....	31
Gambar 4. 8 Desain <i>Screen</i> <i>Monitoring</i> Perangkat Aplikasi SKMML .....	32
Gambar 4. 9 Tampilan <i>Screen</i> <i>Monitoring</i> Perangkat Pada <i>Smartphone</i> .....	33
Gambar 4. 10 Tampilan Rancangan <i>Interface</i> Aplikasi .....	34
Gambar 4. 11 Cara Menambahkan <i>Speech Recognizer</i> .....	35
Gambar 4. 12 Program Pengubahan Perintah Suara Menjadi Teks .....	35
Gambar 4. 13 Pembuatan <i>Database</i> .....	36
Gambar 4. 14 Cara Menambahkan Koneksi Bluetooth .....	38
Gambar 4. 15 Program Blok Mengubungkan Koneksi Bluetooth .....	38
Gambar 4. 16 Program Blok Pengiriman Data Perintah Suara Melalui Bluetooth .....	39

Gambar 4. 17 Tampilan Menu Experimental pada <i>Software</i> MIT App Inventor .	40
Gambar 4. 18 Program Blok Pemanggilan Data Sensor Tegangan .....	41
Gambar 4. 19 Desain Konstruksi <i>Prototype</i> Mobil Listrik .....	42
Gambar 4. 20 Desain Box Kontrol.....	42
Gambar 4. 21 Desain <i>Prototype</i> Tampak Depan .....	43
Gambar 4. 22 Desain <i>Prototype</i> Tampak Samping.....	43
Gambar 4. 23 Desain <i>Prototype</i> Tampak Atas.....	44
Gambar 4. 24 <i>Prototype</i> Mobil Listrik Tampak Depan .....	45
Gambar 4. 25 <i>Prototype</i> Mobil Listrik Tampak Samping .....	45
Gambar 4. 26 <i>Prototype</i> Mobil Listrik Tampak Atas .....	45
Gambar 4. 27 Wiring Diagram Sistem Kontrol dan <i>Monitoring</i> .....	46
Gambar 4. 28 Hasil Akhir Pemasangan Pengkabelan.....	47
Gambar 4. 29 Skema Rangkaian Pengujian Pengiriman Perintah Suara .....	48
Gambar 4. 30 Rangkaian Pengujian Pengiriman Perintah Suara.....	49
Gambar 4. 31 Blok Diagram Prosedur Pengujian Pengiriman Perintah Suara .....	51
Gambar 4. 32 Pengujian Pada Area Dengan Halangan.....	52
Gambar 4. 33 Pengujian Pengiriman Perintah Suara Pada Area Tanpa Halangan	53
Gambar 4. 34 Skema Rangkaian Pengujian Sensor Tegangan .....	56
Gambar 4. 35 Rangkaian Pengujian Sensor Tegangan .....	56
Gambar 4. 36 Prosedur Pengujian Sensor Tegangan.....	57
Gambar 4. 37 Proses Pengujian Sensor Tegangan.....	58
Gambar 4. 38 Skema Rangkaian Komunikasi Serial .....	60
Gambar 4. 39 Rangkaian Pengujian Komunikasi Serial .....	61
Gambar 4. 40 Prosedur Pengujian Komunikasi Serial.....	62
Gambar 4. 41 Proses Pengujian Komunikasi Serial.....	62
Gambar 4. 42 Hasil Pengujian Pada Serial Monitor .....	62
Gambar 4. 43 Tampilan <i>Screen Voice Control</i> : (a) tampilan saat kondisi starter on; (b) tampilan saat kondisi starter off .....	67
Gambar 4. 44 Tampilan <i>Voice Control</i> : (a) tampilan saat kondisi kaca jendela terbuka; (b) tampilan saat kondisi kaca jendela tertutup.....	71

Gambar 4. 45 Tampilan <i>Voice Control</i> : (a) tampilan saat kondisi wiper on; (b) tampilan saat kondisi wiper off .....	76
Gambar 4. 46 Hasil Pengujian <i>Monitoring</i> Sensor Tegangan: (a) Tampilan data sensor pada <i>firebase</i> ; (b) Tampilan data sensor pada <i>screen monitoring</i> .....	80
Gambar 4. 47 Hasil Pengujian <i>Monitoring</i> Sensor Tegangan Ketika Starter Rusak: (a) Data sensor tegangan pada <i>firebase</i> ; (b) Tampilan <i>screen monitoring</i> .....	81
Gambar 4. 48 Hasil Pengujian <i>Monitoring</i> Data Sensor Tegangan Ketika Window Rusak: (a) Data sensor tegangan pada <i>firebase</i> ; (b) Tampilan <i>screen monitoring</i>	81
Gambar 4. 49 Hasil Pengujian <i>Monitoring</i> Data Sensor Tegangan Ketika Kondisi Wiper Rusak: (a) Data sensor tegangan pada <i>firebase</i> ; (b) Tampilan <i>screen monitoring</i> .....	82



## DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino Mega 2560 .....	6
Tabel 2. 2 Spesifikasi Modul Bluetooth HC-05 .....	8
Tabel 2. 3 Spesifikasi NodeMCU ESP8266 .....	9
Tabel 2. 4 Spesifikasi Driver L298N .....	10
Tabel 2. 5 Spesifikasi Sensor Tegangan .....	11
Tabel 4. 1 Hasil Pengujian Pengiriman Perintah Suara Pada Area dengan Halangan .....	52
Tabel 4. 2 Hasil Pengujian Pengiriman Perintah Suara Pada Area Tanpa Halangan .....	53
Tabel 4. 3 Hasil Pengujian Sensor Tegangan .....	58
Tabel 4. 4 Hasil Pengaruh Jarak Sumber Suara ke <i>Smartphone</i> .....	65
Tabel 4. 5 Hasil Pengujian Perintah Suara Pengontrolan <i>Starter</i> dengan Volume Suara 70 dB .....	69
Tabel 4. 6 Hasil Pengujian Perintah Suara Pengontrolan <i>Starter</i> dengan Volume Suara 75 dB .....	69
Tabel 4. 7 Hasil Pengujian Perintah Suara Pengontrolan <i>Starter</i> dengan Volume Suara 80 dB .....	70
Tabel 4. 8 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 70 dB .....	73
Tabel 4. 9 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 75 dB .....	74
Tabel 4. 10 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 80 dB .....	74
Tabel 4. 11 Hasil Pengujian Perintah Suara Pengontrolan <i>Wiper</i> dengan Volume Suara 70 dB .....	78
Tabel 4. 12 Hasil Pengujian Perintah Suara Pengontrolan <i>Wiper</i> dengan Volume Suara 75 dB .....	78

Tabel 4. 13 Hasil Pengujian Perintah Suara Pengontrolan *Wiper* dengan Volume  
Suara 80 dB ..... 79  
Tabel 4. 14 Hasil Pengujian Sistem Monitoring Data Sensor Tegangan..... 82



## **DAFTAR LAMPIRAN**

LAMPIRAN 1: RIWAYAT HIDUP PERORANGAN

LAMPIRAN 2: PROGRAM ARDUINO

LAMPIRAN 3: PROGRAM NODEMCU ESP8266





# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Mobil listrik adalah teknologi transportasi yang sedang dikembangkan pada era modern saat ini [1]. Hal ini dikarenakan ketersediaan bahan bakar fosil yang semakin minim dan diprediksikan beberapa tahun kedepan habis serta harga yang semakin melonjak. Selain itu, penggunaan bahan bakar ini berdampak negatif terhadap lingkungan, dimana emisi gas yang dihasilkan dari penggunaan bahan bakar fosil menyebabkan polusi yang tinggi sehingga berkontribusi terhadap pencemaran lingkungan yang akan berdampak pada pemanasan global [2].

Mobil listrik merupakan energi alternatif pengganti bensin yang telah dikembangkan sebagai bentuk konservasi energi dan perlindungan lingkungan. Kendaraan listrik banyak diproduksi oleh pabrik-pabrik mobil ternama di luar Indonesia. Penggunaan mobil listrik dianggap lebih hemat dari sisi biaya dibandingkan menggunakan kendaraan konvensional karena tidak mencemari udara dan desain mesinnya lebih sederhana. [3].

Pada umumnya, mobil listrik bekerja dengan dua sistem dasar, khususnya sistem mekanis dan sistem elektronik. Sistem mekanik ialah sistem yang tersambung pada pengereman di bagian depan dan belakang, sistem gas, casis bawah serta sistem pemandu (*sterring*). Selain itu pada sistem elektronik berkaitan dengan rangka yang terhubung dengan sensor monitoring dan motor listrik [3].

Pembuatan sistem kontrol dan *monitoring* pada mobil listrik telah dilakukan pada beberapa penelitian sebelumnya yaitu Asep Saefullah dan Dhida Restu Giri Madya (2014) yang membuat prototipe pengontrol pintu dan kaca jendela mobil melalui *smartphone* dengan memanfaatkan Arduino Mega 2560 sebagai mikrokontroler untuk modifikasi *prototype*. Feby Pratama (2019) yang membuat sebuah sistem pengamanan yang mengatur *powerstart* atau *stop* mobil listrik menggunakan *smartphone* dengan sensor Bluetooth.

Mengingat beberapa penelitian yang telah selesai sebelumnya, sistem kontrol yang dibuat hanya menggunakan menggunakan sensor bluetooth serta tidak terdapat *monitoring* dalam pengaplikasiannya. Oleh karena itu pada proyek akhir dibuat sistem kontrol dan *monitoring* pada mobil listrik dengan menggabungkan kontrol pergerakan naik turun kaca jendela mobil dan pergerakan *wiper* serta *starter* mobil listrik dengan perintah suara melalui *smartphone* yang dapat diakses menggunakan koneksi bluetooth. Kemudian, ditambahkan sistem *monitoring* untuk memantau kondisi baik atau tidak dari *wiper*, *starter*, dan kaca jendela pada mobil listrik melalui *smartphone*. Melalui ide ini diharapkan pengguna mobil listrik dapat lebih mudah dalam mengontrol dan memantau peralatan-peralatan mobil listrik. Mengenai pelaksanaan dari pengembangan teknologi tersebut, maka pada proyek akhir akan dibuat “*Prototype Sistem Kontrol dan Monitoring Pergerakan Wiper, Starter dan Kaca Jendela Pada Mobil Listrik Menggunakan Perintah Suara Berbasis IoT*”. Dengan adanya alat ini diharapkan dapat digunakan dalam penggunaan mobil listrik sehingga memberikan kemudahan bagi pengemudi.

## **1.2 Rumusan Masalah**

Pengembangan sistem kontrol dan *monitoring* mobil listrik sangat diperlukan terkait dengan aspek kemudahan dalam penggunaan dan peningkatan keamanan. Adapun rumusan masalah pada pembuatan proyek akhir ini sebagai berikut.

1. Bagaimana cara mendesain dan membuat *prototype* mobil listrik?
2. Bagaimana cara menggunakan Arduino Mega 2560 untuk membuat system kontrol dan *monitoring* pergerakan *wiper*, *starter*, dan kaca jendela pada mobil listrik menggunakan perintah suara?
3. Bagaimana cara mendesain *interface* aplikasi sistem kontrol dan *monitoring* menggunakan MIT App Inventor?
4. Bagaimana mentransfer data perintah suara melalui Bluetooth untuk mengontrol pergerakan *wiper*, *starter*, dan kaca jendela melalui *smartphone*?

5. Bagaimana cara mengirim data melalui NodeMCU ESP8266 untuk menampilkan *update* data sensor tegangan untuk memantau kondisi dari *wiper*, *starter*, dan kaca jendela pada mobil listrik?

### 1.3 Tujuan

Tujuan dalam pembuatan proyek akhir ini adalah sebagai berikut.

1. Perancangan dan pembuatan *prototype* mobil listrik.
2. Pembuatan sistem kontrol dan *monitoring* dengan Arduino Mega 2560 untuk mengontrol pergerakan *wiper*, *starter*, dan kaca jendela pada saat menggunakan mobil listrik.
3. Menciptakan aplikasi untuk *smartphone* yang berguna dalam pengontrolan dan *monitoring* mobil listrik menggunakan perintah suara.
4. Mentransfer data perintah suara secara nirkabel untuk mengontrol pergerakan *wiper*, *starter* dan kaca jendela melalui *smartphone*.
5. Mengirim data secara nirkabel dan menampilkan data sensor melalui *smartphone*.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengertian Mobil Listrik**

Mobil listrik ialah salah satu teknologi transportasi yang sedang dikembangkan pada era modern saat ini. Sebuah kendaraan listrik terdiri dari berbagai komponen utama, seperti motor listrik, baterai (*accu*), *charger* (pengisi daya), *speed control* (pengatur kecepatan) dan sistem manajemen energi (*Energi Management System* atau EMS). Penggerak utama dari sebuah mobil listrik adalah motor listrik. Prinsip kerja dari mobil listrik yaitu energi listrik diubah menjadi energi mekanik oleh motor listrik. Motor listrik mengambil energi dari baterai yang bertindak sebagai penyimpan energi listrik. Ketika energi listrik yang tersimpan habis, maka baterai dapat diisi ulang [4].

#### **2.2 Studi Literatur**

Penelitian tentang *prototype* sistem kendali dan *monitoring* pada mobil listrik ini telah banyak dilakukan oleh beberapa perguruan tinggi, khususnya untuk keperluan proyek akhir ini. *Prototype* sistem kontrol dan *monitoring* mobil listrik ini bertujuan untuk melakukan kontrol melalui perintah suara kepada peralatan pada mobil listrik dan melakukan pemantauan secara *real time* terkait kondisi dari masing-masing peralatan pada mobil listrik. Studi sebelumnya telah membahas sistem kontrol dan *monitoring* pada mobil listrik di berbagai bidang studi. Semua penelitian yang telah dilakukan memiliki beberapa persamaan dan perbedaan, baik dari sisi mikrokontroler yang digunakan, tujuan penelitian, dan komponen lainnya.

Terdapat beberapa penelitian yang berkaitan dengan sistem kontrol pergerakan turun naik kaca jendela dan sistem kontrol *starter* pada mobil listrik, diantaranya penelitian Asep Saefullah dan Dhida Restu Giri Madya (2014) yang membuat *prototype* pengontrol pintu dan jendela mobil dengan *smartphone* berbasis ATMEGA328P yang menggunakan Arduino Mega 2560 sebagai *platform* untuk

desain dan pengembangan *prototype*. Feby Pratama (2019) membuat sistem kontrol yang menggunakan *smartphone* dengan Bluetooth untuk memulai dan menghentikan daya pada mobil listrik.

Berdasarkan penelitian yang telah dilakukan sebelumnya, maka *prototype* sistem kontrol dan *monitoring* pada mobil listrik yang akan dibuat pada proyek akhir ini menggabungkan kontrol pergerakan naik turun kaca jendela mobil dan pergerakan *wiper* serta *starter* mobil listrik dengan perintah suara melalui *smartphone*. Pada alat ini juga dilakukan beberapa pengembangan yaitu dengan menambah koneksi dalam mengakses sistem kontrol dan *monitoring* dengan menggunakan bluetooth. Kemudian, ditambahkan sensor tegangan yang berfungsi sebagai sistem *monitoring* untuk memantau kondisi baik atau tidak dari *wiper*, *starter*, dan jendela pada mobil listrik melalui *smartphone*. Sistem kontrol dan *monitoring* ini akan berfungsi sesuai dengan pemrograman yang sudah dirancang.

### **2.3 Speech Recognition (Pengenalan Kata)**

*Speech Recognition* atau pengenalan kata adalah perkembangan teknologi untuk menerima dan memahami suatu perintah suara yang diucapkan dengan mendigitalkan dan mencocokkan sinyal digital dengan pola tertentu yang tersimpan di perangkat [5]. Perintah suara melewati tahap konversi ke sinyal digital dengan mengubah gelombang suara menjadi angka dan kemudian menafsirkan kata-kata tersebut agar sesuai kode tertentu. Hasil pengenalan bahasan lisan (perintah suara) dibaca oleh mikrokontroler sebagai sebuah perintah untuk melakukan sesuatu, seperti menyalakan lampu secara otomatis dengan perintah suara [6].

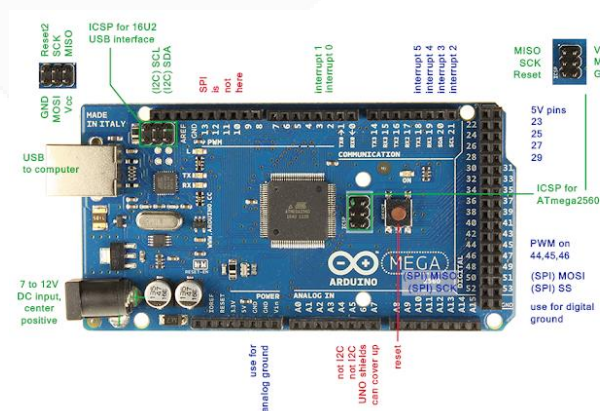
Prinsip kerja dari teknologi *speech recognition* terdiri dari empat fase sebagai berikut: fase penerimaan masukan, selama fase ini perintah suara diterima dari media perantara. Setelah perintah suara diterima akan memasuki fase ekstraksi yaitu perintah suara yang telah diterima akan disimpan diikuti dengan pembuatan basis data dari perintah suara yang telah diterima. Fase selanjutnya yaitu fase perbandingan dimana data baru dibandingkan dengan data kebahasaan yang sudah terdapat pada pola. Fase terakhir dari prinsip kerja *speech recognition* adalah fase

verifikasi dimana pada fase ini dilakukan menerjemahkan suara tersebut menjadi suatu perintah [6].

## 2.4 Arduino Mega 2560

Arduino merupakan papan mikrokontroler open source yang terdiri dari perangkat lunak dan perangkat keras. Mikrokontroler ialah chip yang digunakan dalam perangkat elektronik digital yang memiliki input dan output serta dikendalikan oleh sebuah program yang dirancang khusus. Mikrokontroler terdiri dari prosesor, memori, input dan output serta ADC (*Analog to Converters*) yang melakukan pengubahan data analog menjadi data digital yang dapat terbaca oleh mikrokontroler [7].

Arduino Mega 2560 ialah papan Arduino yang dilengkapi dengan IC mikrokontroler ATmega 2560. Arduino jenis ini memiliki 54 pin input dan pin output (I/O). Fitur tambahan dari Arduino Mega 2560 dilengkapi dengan kristal 16 Mhz [8].



Gambar 2. 1 Arduino Mega 2560 [8]

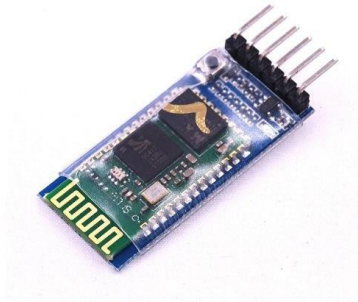
Tabel 2. 1 Spesifikasi Arduino Mega 2560 [8]

No	Spesifikasi	Nilai
1.	Mikrokontroler	ATmega 2560
2.	Tegangan Operasi	5 V

3.	Tegangan Input (rekomendasi)	7-12 V
4.	Tegangan Input (limit)	6-20 V
5.	Pin Digital I/O	54 (15 sebagai output PWM)
6.	Pin Analog Input	16
7.	Arus DC per Pin I/O	20 mA
8.	Arus DC untuk Pin 3.3 V	50 mA
9.	Memori <i>Flash</i>	256 KB dengan 8 KB digunakan sebagai bootloader
10.	SRAM	8 KB
11.	EEPROM	4 KB
12.	<i>Clock Speed</i>	16 MHz
13.	LED_BUILTIN	101.52 mm
14.	Panjang	101.52 mm
15.	Lebar	53.3 mm
16.	Berat	37 g

## 2.5 Modul Bluetooth HC -05

Bluetooth adalah perangkat komunikasi nirkabel yang mampu mentransmisikan layanan komunikasi data. Bluetooth beroperasi di pita frekuensi 2,4 GHz ISM (*Industrial, Scientific and Medial*) bebas lisensi. HC-05 digunakan sebagai penerima dan pemancar dengan komunikasi serial SPP (*Serial Port Protocol*) dengan frekuensi operasi 2.4 GHz. Mode konfigurasi dari modul bluetooth HC-05 terdiri dari 2 mode, yang pertama adalah mode AT dan mode kedua mode komunikasi [9].



Gambar 2. 2 Modul Bluetooth HC-05 [10]

Tabel 2. 2 Spesifikasi Modul Bluetooth HC-05 [10]

No	Spesifikasi	Nilai
1.	Tegangan kerja	4-6 V (rekomendasi 5 V)
2.	Arus Kerja	50 mA
3.	Sensitivitas	-84dBm (0.1% BER)
4.	Daya emisi	4 dBm
5.	Suhu	-20°C - 75°C
6.	Dimensi	15.2 x 35.7 x 5.6
7.	Baudrate default	9600

## 2.6 NodeMCU ESP8266

NodeMCU ESP8266 ialah sebuah papan *platform Internet of Thing* (IoT) untuk keluarga ESP8266. NodeMCU ESP8266 memiliki kemampuan WiFi dan *firmware open source* [11]. Spesifikasi dari board ini didasarkan pada Wifi *Single-on-Chips* (SoC) serial ESP8266 dengan USB terintegrasi ke TTL dan menggunakan IEEE 802.11b/g/n sebagai global nirkabel. NodeMCU ESP8266 memiliki konektor micKo USB yang digunakan sebagai transfer atau tegangan input untuk data pemrograman, serta tombol *reset* dan *flash* [12].





Gambar 2. 3 NodeMCU ESP8266 [12]

Tabel 2. 3 Spesifikasi NodeMCU ESP8266 [12]

No.	Spesifikasi	Keterangan
1.	Mikrokontroler	ESP8266
2.	Ukuran	57mm x 30mm
3.	Pin GPIO	13
4.	Pin ADC	1
5.	Pin PWM	10
6.	Wifi	IEE 802.11b/g/n
7.	Flash Memory	4 MB
8.	Frekuensi	2.4 GHz-22.5 GHz
9.	USB Port	Muikro USB

## 2.7 Motor DC

Motor DC ialah motor arus searah yang berfungsi untuk merubah energi listrik menjadi sebuah energi kinetik. Rotasi motor listrik dapat searah jarum jam atau berlawanan arah jarum jam, tergantung pada polaritas listrik yang disuplai pada motor tersebut [13].

Ada dua bagian utama pada konstruksi motor DC yaitu rotor dan stator. Rotor terdiri dari kumparan jangkar yang berputar. Stator terdiri dari kumparan rangka dan kumparan medan yang tidak berputar [14].

Prinsip kerja dari sebuah motor DC berkaitan dengan fenomena electromagnet untuk setiap putarannya. Ketika kumparan mendapatkan arus listrik menyebabkan

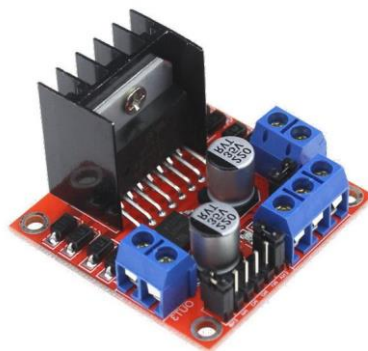
permukaan yang kumparan utara akan bergerak menuju magnet kutub selatan kemudian kumparan selatan bergerak ke utara magnet. Gerakan ini menyebabkan saling tarik-menarik dan menghentikan gerakan kumparan [14].



Gambar 2. 4 Motor DC *Gear Box* [14]

## 2.8 Driver Motor L298N

Driver motor L298N merupakan modul motor DC yang digunakan untuk mengatur kecepatan dan arah putaran motor DC. Driver Motor L298N menggunakan IC L298N dengan rangkaian *H-Bridge* yang dapat mengontrol beban induktif kumparan. Rangkaian IC L298N juga menyertakan *Transistor Transistor Logic* (TTL) dengan gerbang NAND untuk membantu mengontrol arah putaran motor [15].



Gambar 2. 5 Driver L298N [15]

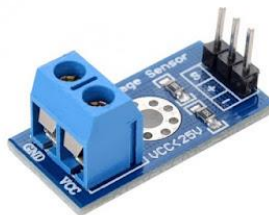
Tabel 2. 4 Spesifikasi Driver L298N [15]

No	Spesifikasi	Keterangan
1.	Tegangan Masukan	3.2 – 40 VDC
2.	Driver	L298N Dual H Bridge DC Motor Driver

3.	Tegangan Operasional	5 VDC
4.	Arus maksimal untuk keluaran per Output	2 A
5.	Arus masukan	0 – 36 mA
6.	Daya maksimal	25 W
7.	Dimensi	43 x 43 x 26 mm

### 2.9 Sensor Tegangan DC

Sensor Tegangan DC adalah sebuah modul rangkaian pembagi tegangan. Sensor ini memiliki batasan pengukuran tegangan sebesar 25V [16]. Prinsip kerja sensor tegangan didasarkan pada prinsip menekan resistansi dan mengurangi tegangan input menjadi seperlima dari tegangan aslinya [17].



Gambar 2. 6 Sensor Tegangan DC [17]

Tabel 2. 5 Spesifikasi Sensor Tegangan [17]

No.	Spesifikasi	Nilai
1.	Tegangan Input	0 – 25 VDC
2.	Tegangan Deteksi	0.02445 – 25 VDC
3.	Ketelitian Pengukuran	0.00489 V
4.	Dimensi	25 x 13 mm

## 2.10 Aki (Accumulator)

Aki merupakan alat yang berguna untuk menyimpan energi listrik berupa energi kimia dapat menjadi sumber energi listrik. Keluaran yang dihasilkan dari aki ialah tegangan DC. Ketika dibutuhkan, aki akan menyalurkan daya ke masing-masing komponen pada sistem kelistrikan kendaraan. Aki yang digunakan pada proyek akhir ini adalah aki 12V [18].



Gambar 2. 7 Aki 12V

## 2.11 Arduino IDE

Arduino IDE (*Integrated Development Environment*) ialah sebuah *software* yang berfungsi untuk membuat sebuah program melalui Arduino yang memiliki berbagai fitur diantaranya teks editor, membuat program dan memvalidasi sebuah kode program serta upload ke board Arduino. Bahasa pemrograman Arduino (*sketch*) menggunakan Bahasa pemrograman yang mirip dengan C dan telah dimodifikasi dari bahasa aslinya, sehingga memudahkan pengguna dalam membuat program [19].



Gambar 2. 8 Tampilan *Software* Arduino [19]

## 2.12 MIT App Inventor

MIT App Inventor adalah *platform* yang dikelola oleh Massachusetts Institute of Technology yang berfungsi untuk menciptakan aplikasi android dalam bentuk visual. Melalui MIT App Inventor para pengguna dapat berkreasi membuat aplikasi sederhana hingga aplikasi yang lebih kompleks.

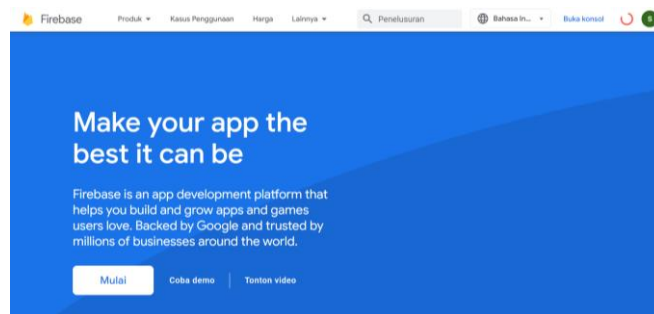
Ada dua bagian penting yang harus diketahui pada saat ingin membuat aplikasi menggunakan MIT App Inventor yaitu desain dan *block code*. Pada bagian desain berfungsi untuk membuat tampilan antarmuka dari aplikasi kemudian pada bagian *block code* berfungsi untuk membuat kode-kode logika untuk menjalankan tampilan antarmuka dari aplikasi yang dibuat. MIT App Inventor juga dilengkapi dengan fitur emulator yang dapat menjalankan aplikasi yang dibuat secara *real time* dan mengekspornya menjadi file (apk) [20].



Gambar 2. 9 Tampilan Software MIT App Inventor [20]

## 2.13 Firebase

*Firebase* adalah salah satu platform untuk menyimpan data yang disediakan oleh google berbasis *Backend as Service* (BaaS). Tampilan *firebase* dapat dilihat pada Gambar 2.10 berikut.



Gambar 2. 10 Tampilan *Software Firebase*

Terdapat beberapa fitur yang tersedia pada firebase antara lain *Firestore Real Time Database*, *Analytics*, *Cloud Messaging and Notification*, *Authentication*, *Remote Config*, dan *Crash Reporting*. Berdasarkan dari beberapa menu yang telah disebutkan, menu yang sering digunakan adalah *Firestore Real Time Database*. *Firestore Real Time Database* ialah menu berfungsi menyimpan serta mengeksekusi data dengan format JSON secara real time serta menyinkronkannya dengan aplikasi yang terkoneksi [21].

## 2.14 IoT



Gambar 2. 11 Logo IoT [22]

*Internet of Things* atau biasa disebut dengan IoT adalah sistem komunikasi yang mengirimkan data melalui jaringan internet untuk terhubung ke sistem. Terdapat lima unsur utama dari internet diantaranya *Artificial Intelligence*, konektivitas, perangkat berukuran kecil, sensor, dan *active engagement* [23]. Prinsip kerja dari IoT adalah memanfaatkan perintah-perintah dari pemrograman yang telah dibuat yang selanjutnya mengakibatkan terjadinya interaksi sistem secara otomatis dengan sistem kerja yang luas. Cara kerja dari konsep IoT berlandaskan tiga bagian

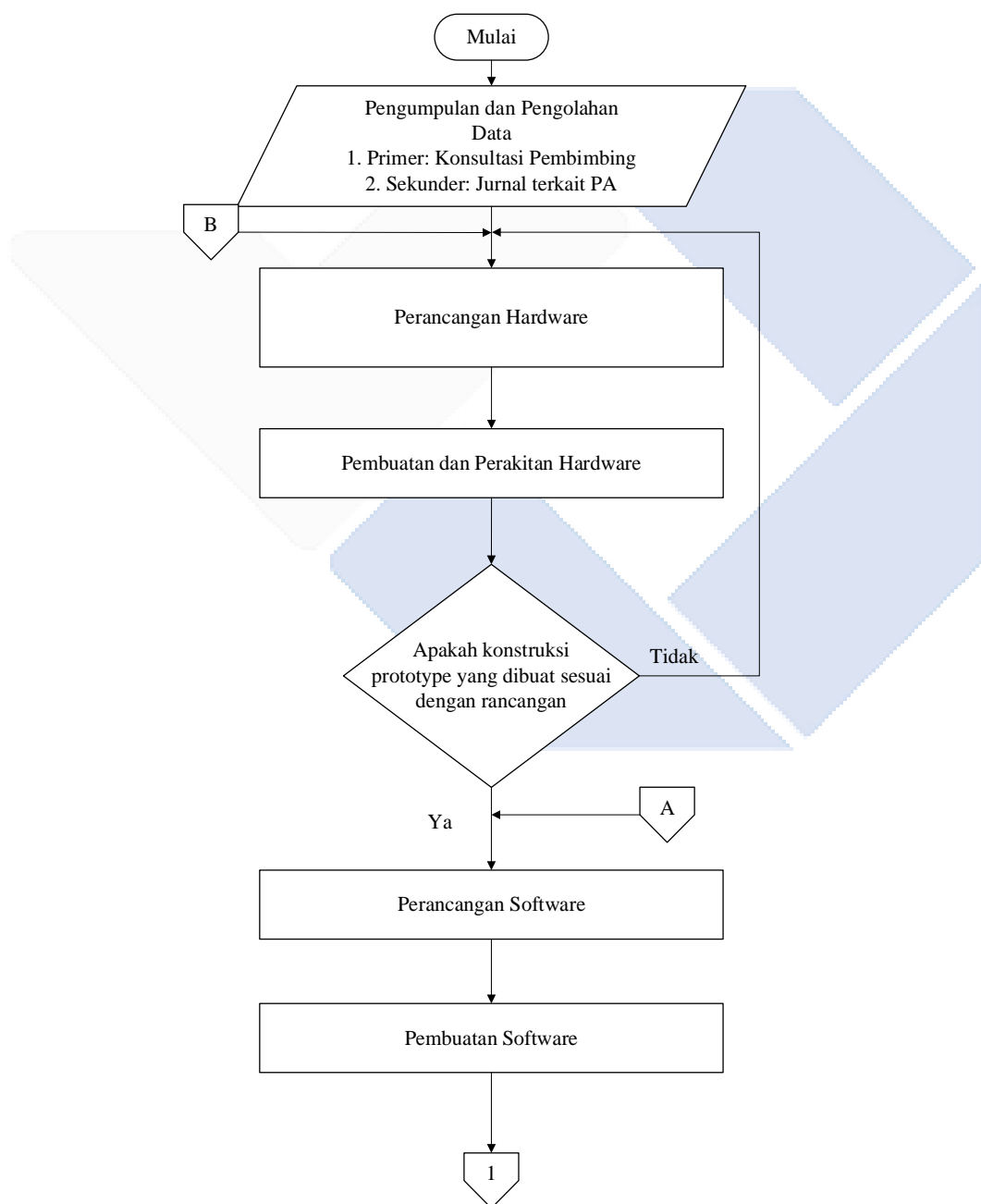
yaitu produk fisik dari modul IoT, perangkat yang terhubung ke internet seperti modem dan router berkecepatan tinggi serta bagian terakhir adalah pusat data yang digunakan sebagai database [22].



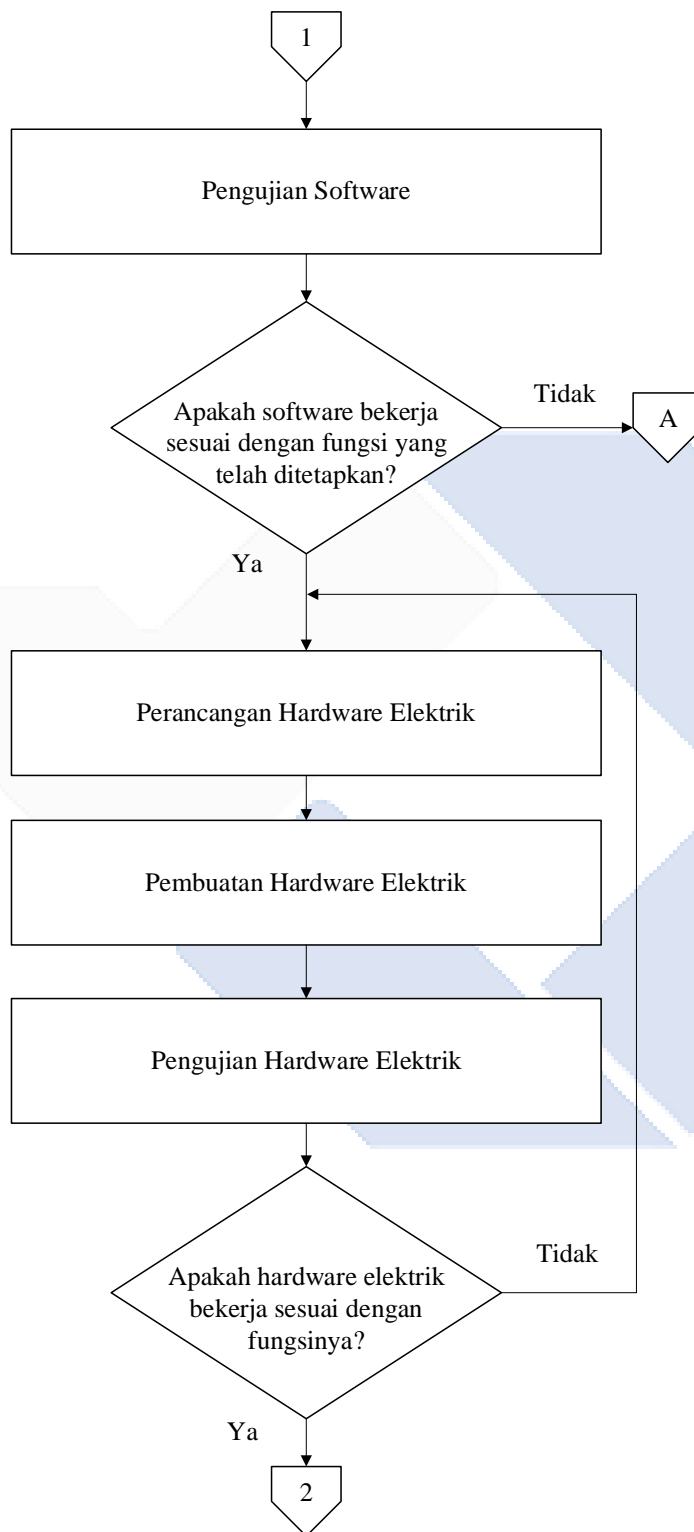
### BAB III

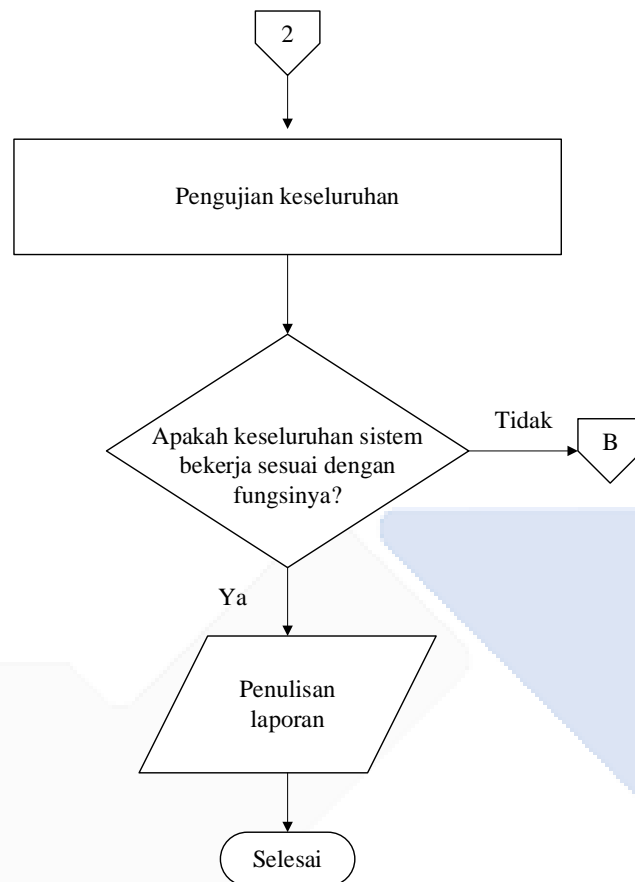
#### METODE PELAKSANAAN

Ada beberapa tahap dalam menyelesaikan proyek akhir ini untuk memudahkan pengerjaan proyek akhir. Tahapan pengerjaan proyek akhir dapat dilihat pada diagram alir dibawah ini.









Gambar 3. 1 Diagram Alir Metode Pelaksanaan Proyek Akhir

### 3.1 Pengumpulan dan Pengolahan Data

Tahap ini, data-data dikumpulkan melalui pencarian kajian literatur yang berhubungan dengan judul proyek akhir ini. Tinjauan pustaka ini dilakukan untuk mengerjakan proyek akhir dan memperoleh informasi yang dapat digunakan sebagai acuan makalah proyek akhir. Dengan penelusuran literatur ini, dapat mengetahui sketsa mengenai sistem kontrol dengan menggunakan perintah suara pada sebuah mobil listrik serta sistem *monitoring* dari masing-masing peralatan mobil listrik.

Metode pengumpulan data yang digunakan pada tahapan ini terdiri dari dua aspek yakni pengumpulan data langsung (primer) serta pengumpulan data tidak langsung (sekunder). Pengumpulan data langsung didapatkan melalui konsultasi dengan dosen pembimbing. Kemudian pengumpulan data secara tidak langsung

didapatkan melalui pencarian jurnal-jurnal baik nasional maupun internasional yang berhubungan dengan judul proyek akhir yang dibuat.

Setelah tahapan pengumpulan data selesai, tahapan selanjutnya memproses data yang telah diperoleh, memilahnya, dan menggunakannya sebagai referensi untuk mengembangkan ide-ide baru untuk membuat proyek akhir.

### **3.2 Perancangan Software dan Hardware**

Tahapan perancangan *software* dilakukan untuk mendesain pengiriman perintah suara dan monitoring serta tampilan aplikasi android. Sementara itu perancangan *hardware* berguna untuk mendesain bentuk dari miniatur mobil listrik dan peletakan motor *wiper*, motor *window* dan motor *starter*. Kemudian perancangan sistem kontrol untuk mengendalikan masing-masing peralatan mobil listrik dengan menggunakan perintah suara.

#### **3.2.1 Desain Software**

Pada tahapan desain *software* ini beberapa perencanaan sebagai berikut:

1. Pemrograman Arduino yang digunakan mengolah data hasil perintah suara yang dikirimkan oleh modul Bluetooth, menampilkan perubahan data dari sensor tegangan, dan komunikasi serial untuk mengirimkan data ke NodeMCU ESP8266.
2. Merancang tampilan aplikasi pada smartphone yang digunakan untuk mengirimkan perintah suara yang digunakan untuk mengontrol peralatan mobil listrik (*wiper*, *window* dan *starter*) dan menampilkan perubahan data dari sensor tegangan untuk *monitoring* kondisi dari masing-masing peralatan mobil listrik (*wiper*, *window* dan *starter*).
3. Pemrograman NodeMCU ESP8266 untuk pengiriman data sensor ke *firebase*.

#### **3.2.2 Desain Hardware**

Ada 2 jenis desain *hardware* yang dibuat pada *prototype* mobil listrik yaitu perangkat mekanik dan perangkat elektrik. Pada tahap ini dilakukan rancang bangun konstruksi dari miniatur mobil listrik dengan *sketch up*. Tahap ini

menentukan ukuran, alat, dan bahan yang digunakan untuk pembuatan *prototype* mobil listrik.

Desain perangkat mekanik yang akan dibuat seperti berikut.

1. Membuat miniatur mobil listrik dengan bahan kayu dan akrilik yang memiliki bagian-bagian diantaranya: kabin dengan ukuran kabin yaitu panjang 14,5 cm, lebar 18 cm dan tinggi 26,5 serta panjang keseluruhan dari miniatur mobil listrik 72cm.
2. Pemasangan motor *wiper*, motor *starter* dan motor *window* pada miniatur mobil listrik.
3. Pembuatan box kontrol sebagai tempat untuk meletakkan mikrokontroler, driver motor, NodeMCU ESP8266, sensor tegangan DC, dan *buck boost converter*.

Kemudian pada tahap desain perangkat elektrik ini merancang *wiring* kontrol menggunakan Arduino Mega 2560 sebagai mikrokontroler untuk mengontrol dan memantau *prototype* mobil listrik.

### **3.3 Pembuatan Software dan Hardware**

Tahapan ini adalah tahap mewujudkan rancangan *software* dan *hardware* yang telah didesain pada tahapan diatas.

#### **3.3.1 Pembuatan Software**

Pada tahap pembuatan *software* dilakukan berdasarkan desain *software* yang telah dirancang sebelumnya. Pembuatan *software* pada proyek akhir ini terdiri atas dua jenis yaitu *software* mikrokontroler dan *software* IoT.

Tahap-tahap membuat *software* untuk mikrokontroler sebagai berikut.

1. Membuat program Arduino Mega 2560 untuk memprogram modul bluetooth HC-05 untuk mengirimkan data perintah suara kemudian pengolahan data perintah suara untuk menggerakkan masing-masing peralatan dari mobil listrik.

2. Membuat programan Arduino Mega 2560 untuk mendapat perubahan data sensor tegangan dan kemudian menjalankan komunikasi serial kepada NodeMCU untuk mengirimkan data sensor.

Tahp-tahap membuat *software* untuk IoT sebagai berikut.

1. Membuat aplikasi menggunakan *platform* MIT App Inventor yang dapat di instal pada *smartphone* pengguna.
2. Membuat blok program untuk menampilkan perubahan data sensor tegangan yang terpasang pada masing-masing peralatan mobil listrik (*starter, wiper, dan kaca jendela*) menggunakan *platform* MIT App Inventor

### 3.3.2 Pembuatan Hardware

Tahap membuat perangkat secara mekanis dibuat dengan membuat bentuk fisik dari miniatur mobil listrik yang mengacu pada desain yang telah dibuat sebelumnya. Pilihan bahan dasar untuk pembuatan diselaraskan dengan desain.

Tahapan membuat perangkat keras secara mekanis seperti dibawah ini.

1. Membuat miniatur mobil truk dengan bahan utama kayu dan PVC.
2. Memotong akrilik bening untuk membuat sebuah box kontrol dengan menggunakan cutter akrilik,
3. Pembuatan box kontrol dengan melekatkan akrilik bening yang sudah dipotong berdasarkan ukuran menggunakan lem akrilik.
4. Memasang motor *wiper*, motor *window*, dan motor *starter* pada miniatur mobil.
5. Meletakkan mikrokontroler (Arduino Mega 2560), NodeMCU ESP8266, Driver Motor L298N, sensor tegangan, *buck boost converter*, dan aki pada box kontrol.

### 3.4 Pengujian Hardware dan Software

Tahapan selanjutnya ialah pengujian pada *hardware* dan *software* pada sistem kontrol dan monitoring dari masing-masing peralatan mobil listrik. Tujuan dilakukannya tahapan ini untuk mengetahui apakah sistem kontrol perintah suara

dan sistem *monitoring* data sensor dapat bekerja sesuai dengan target yang diinginkan.

#### **3.4.1 Pengujian Hardware**

1. Pengujian Arduino Mega 2560 dengan modul Bluetooth HC-05 untuk mengetahui kemampuan pengiriman data perintah suara.
2. Pengujian Arduino Mega 2560 dengan motor *wiper*, motor *starter*, dan motor *window* untuk mengetahui respon masing-masing motor ketika diberikan instruksi perintah suara.
3. Pengujian Arduino Mega 2560 terhadap sensor tegangan sehingga dapat mengetahui pengukuran oleh sensor tegangan kepada tegangan pada input masing-masing motor.

#### **3.4.2 Pengujian Software**

1. Pengujian aplikasi yang telah dibuat dalam mengirimkan perintah suara melalui aplikasi *smartphone*.
2. Pengujian aplikasi *smartphone* yang sudah diciptakan untuk memantau data sensor melalui *smartphone* serta memantau kondisi dari masing-masing motor.

#### **3.4.3 Pengujian Keseluruhan Sistem**

Pada tahap ini sistem secara keseluruhan diuji baik sistem kontrol pergerakan *wiper*, *starter*, dan kaca jendela pada mobil listrik serta sistem *monitoring* data sensor tegangan dari masing-masing peralatan mobil listrik untuk memantau apakah seluruh sistem yang sudah dibuat berfungsi sesuai dengan tujuan. Pengujian dilakukan dengan mengirimkan perintah suara melalui aplikasi kemudian melihat reaksi yang terjadi serta memantau kondisi dari masing-masing peralatan dengan melihat melalui aplikasi.

#### **3.5 Analisis Data**

Tahapan selanjutnya adalah menganalisis data hasil pengujian sistem yang sudah dilakukan menurut berbagai aspek seperti konstruksi *prototype* mobil listrik,

perangkat keras, serta perangkat lunak. Jika masih ditemukan kekurangan atau kinerja belum maksimal akan melakukan perbaikan untuk mencapai hasil terbaik.

### **3.5 Pembuatan Makalah Proyek Akhir**

Tahapan ini merupakan tahapan akhir dari proses pengerjaan proyek akhir. Pembuatan makalah proyek akhir dimaksudkan untuk merangkum semua kegiatan yang berkaitan dengan proyek akhir yang dimulai dari pendahuluan, landasan teori, metode pelaksanaan, pembahasan dan yang terakhir kesimpulan dan saran.

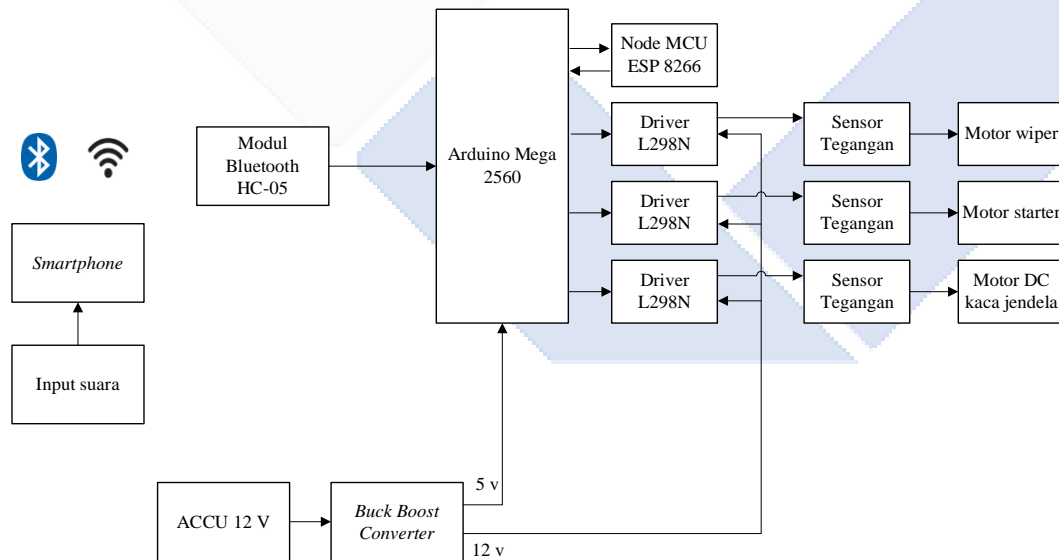


## BAB IV PEMBAHASAN

Dalam bab ini akan dibahas tentang proses pembuatan proyek akhir, dimulai dengan proses merancang dan membuat sistem kontrol serta *monitoring*, merancang dan membuat konstruksi, merancang dan membuat aplikasi untuk *smartphone*, serta pengujian keseluruhan *prototype* mobil listrik sebagai berikut.

### 4.1 Deskripsi Alat

Prototype sistem kontrol dan *monitoring* pada mobil listrik ini dapat melakukan pengontrolan pergerakan *wiper*, *starter*, dan kaca jendela pada mobil listrik menggunakan perintah suara dan *monitoring* sensor tegangan secara *real time* untuk memantau keadaan dari masing-masing peralatan mobil listrik dengan menggunakan *smartphone*. Blok diagram pengoperasian *prototype* ini ditunjukkan pada gambar dibawah ini.



Gambar 4. 1 Blok Diagram Cara Kerja Alat

Sistem pengontrolan *prototype* mobil listrik ini melibatkan Arduino Mega 2560 sebagai mikrokontroler untuk mengeksekusikan data masukan dalam bentuk teks yang dikirimkan oleh modul Bluetooth HC-05 untuk menggerakkan peralatan mobil



listrik serta mengolah data input dari sensor tegangan kemudian ditransfer melalui komunikasi serial ke NodeMCU ESP8266 untuk *monitoring* perubahan data dari sensor tegangan sehingga bisa dilakukan pemantauan secara langsung pada *smartphone*. User akan mengirimkan perintah suara melalui sebuah aplikasi yang bernama SKMML dan user dapat memantau langsung data sensor tegangan yang terpasang pada masing-masing peralatan mobil listrik.

## **4.2 Perancangan dan pembuatan Software**

Pada tahap ini akan membuat perancangan aplikasi untuk *smartphone* yang bernama SKMMML, komunikasi serial pada *platform* MIT App Inventor, perancangan dan pembuatan *database* untuk *monitoring* sensor tegangan, dan koneksi MIT App Inventor dengan *database*.

### **4.2.1 Perancangan dan Pembuatan Aplikasi**

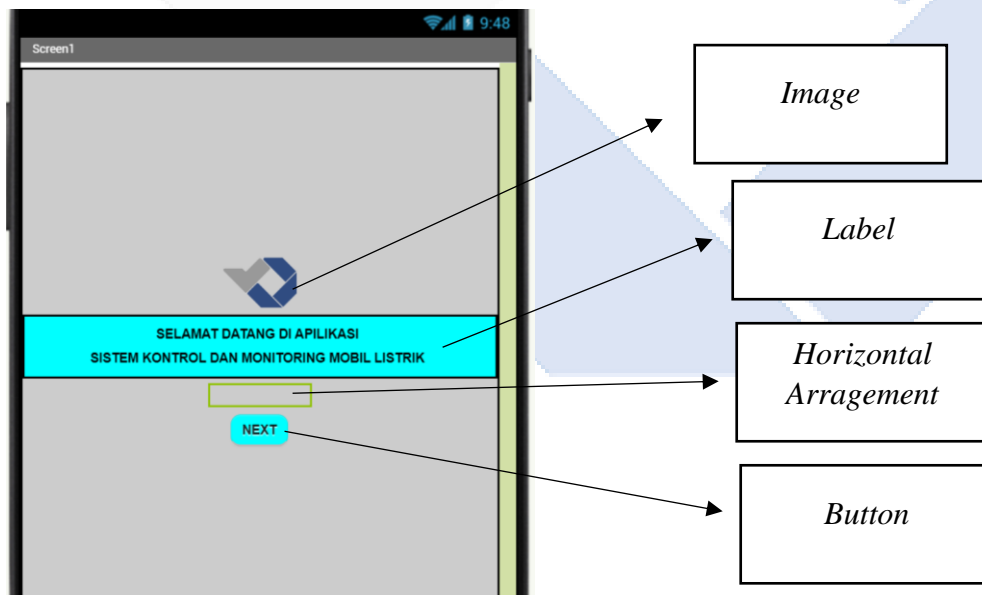
Perancangan dan pembuatan aplikasi digunakan sebagai media mengirimkan perintah suara untuk memberikan perintah menghidupkan atau mematikan peralatan mobil listrik (*wiper*, *starter*, dan kaca jendela) serta memantau perubahan data sensor tegangan yang terpasang pada masing-masing peralatan mobil listrik (*wiper*, *starter*, dan kaca jendela) sehingga dapat mengetahui kondisi dari masing-masing peralatan mobil listrik. Aplikasi yang dibuat ini diberi nama SKMML dengan *interface* aplikasi dirancang menggunakan *platform* MIT App Inventor. Aplikasi ini yang terdiri dari empat *screen* yaitu pertama, *screen* satu ialah tampilan awal ketika membuka aplikasi “SKMML”.

Pada *screen* pertama terdapat tombol yang bertuliskan NEXT untuk menuju ke *screen* berikutnya. Kedua, *screen* kedua ialah *screen* untuk *login* agar bisa masuk ke menu utama kontrol dan *monitoring*. Pada *screen login* ini user diminta untuk mengisi *username* dan *password*. *Username* dan *password* telah ditentukan oleh tim pembuat aplikasi. Setelah mengisikan *username* dan *password* yang sesuai berhasil selanjutnya user akan masuk ke *screen* ketiga yaitu menu kontrol perintah suara. Pada *screen* ini terdapat simbol koneksi bluetooth dimana pengguna harus menghubungkan koneksi Bluetooth aplikasi kepada modul Bluetooth HC-05, setelah terhubung pengguna dapat mengirimkan perintah suara dengan menekan

*icon microphone* yang kemudian akan terhubung dengan *google voice*. Pada pojok bawah *screen* ketiga ini terdapat tombol GO TO MONITOR dimana jika ditekan maka akan masuk ke *screen* keempat yaitu menu *monitoring* data sensor tegangan. Pada *screen* keempat ini terdapat informasi rentang tegangan kerja dari masing-masing peralatan mobil listrik sehingga dapat dilihat dari data sensor yang tampil pada *screen* keempat apakah dalam rentang tegangan optimum. Pada pojok bawah *screen* ketiga ini terdapat tombol GO TO CONTROL sehingga pengguna dapat kembali ke *screen* keempat untuk melakukan kontrol dengan menggunakan perintah suara. Adapun tampilan aplikasi yang buat menggunakan platform MIT App Inventor dapat dilihat sebagai berikut.

#### 1. Beranda

Beranda merupakan tampilan awal ketika membuka aplikasi. Pengguna dapat menekan tombol NEXT untuk melanjutkan ke *screen* berikutnya pada *screen* beranda tersebut. Gambar berikut menunjukkan tampilan beranda di *smartphone* yang didesain dengan platform MIT App Inventor.



Gambar 4. 2 Desain *Screen* Beranda Aplikasi SKMML

Berikut keterangan dari *interface screen* beranda pada aplikasi SKMML:

- *Image*, berfungsi sebagai peletakan gambar yang diinginkan pada *interface*. Pada menu beranda ditambahkan logo polmanbabel sebagai identitas pada aplikasi SKMML.
- *Label*, berfungsi untuk menuliskan sebuah kalimat. Selain itu, label bisa digunakan menampilkan kalimat yang ingin ditampilkan pada *screen* aplikasi seperti pemberian informasi.
- *Horizontal Arrangement*, berfungsi untuk desain tampilan untuk menempatkan komponen secara horizontal dan memberikan ruang di antara komponen untuk kejelasan dan kerapian.
- *Button*, berfungsi untuk pendeteksi ketukan ketika *user* menyentuh tombol. Tombol kan mendeteksi sentuhan tersebut kemudian akan menjalankan perintah sesuai dengan blok program. Pada *screen* beranda ini, terdapat satu buah tombol yang bertuliskan NEXT yang akan merubah tampilan aplikasi menuju *screen* selanjutnya yaitu Portal Login.

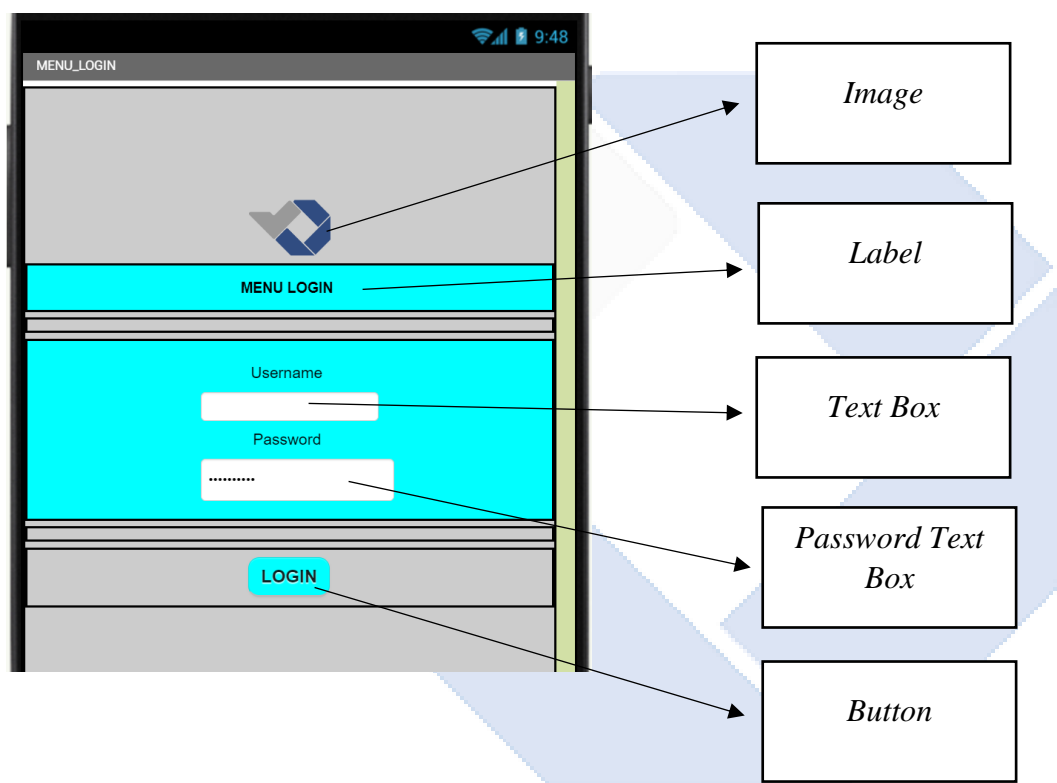
Tata letak menu beranda yang ditampilkan di layar *smartphone* saat aplikasi SKMML dibuka adalah sebagai berikut.



Gambar 4. 3 Tampilan *Screen* Beranda pada *Smartphone* User

## 2. Portal Login

Portal *login* adalah tampilan yang berfungsi untuk mengakses *screen* kontrol dan *monitoring* Pada *screen* portal *login*, *user* harus mengisi *username* dan *password* sebagai proteksi keamanan dalam penggunaan aplikasi SKMML. Pada *screen* login tidak ditambahkan menu pendaftaran *username* dan *password*, hal ini bertujuan agar tidak semua pengguna dapat mendapatkan akun untuk mengakses aplikasi ini karena aplikasi ini dirancang khusus untuk setiap satu pengguna satu akun.



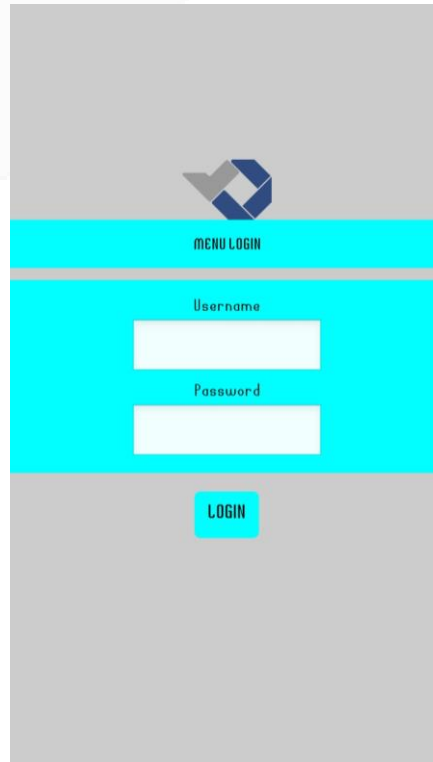
Gambar 4. 4 Desain *Screen* Portal *Login* Aplikasi SKMML

Berikut keterangan dari *interface* menu Login pada aplikasi SKMML:

- *Image*, berfungsi untuk meletakkan gambar yang diinginkan pada *interface*. Pada menu login ditambahkan logo polmanbabel sebagai identitas pada aplikasi SKMML.

- *Label*, berfungsi untuk menuliskan sebuah kalimat. Selain itu, label bisa digunakan menampilkan kalimat yang ingin ditampilkan pada *screen* aplikasi seperti pemberian informasi.
- *Text Box*, berfungsi untuk menuliskan *username*.
- *Password Text Box*, berfungsi untuk mengisikan *password* dan ketika menggunakan mengisikan *password* kalimat yang telah ditulis tidak dapat dibaca sehingga terjamin keamanannya.
- *Button*, berfungsi untuk pendeteksi ketukan ketika *user* menyentuh tombol. Tombol kan mendeteksi sentuhan tersebut kemudian akan menjalankan perintah sesuai dengan blok program. Pada menu ini terdapat satu buah *button* yang bertuliskan LOGIN untuk melanjutkan ke halaman sistem kontrol mobil listrik.

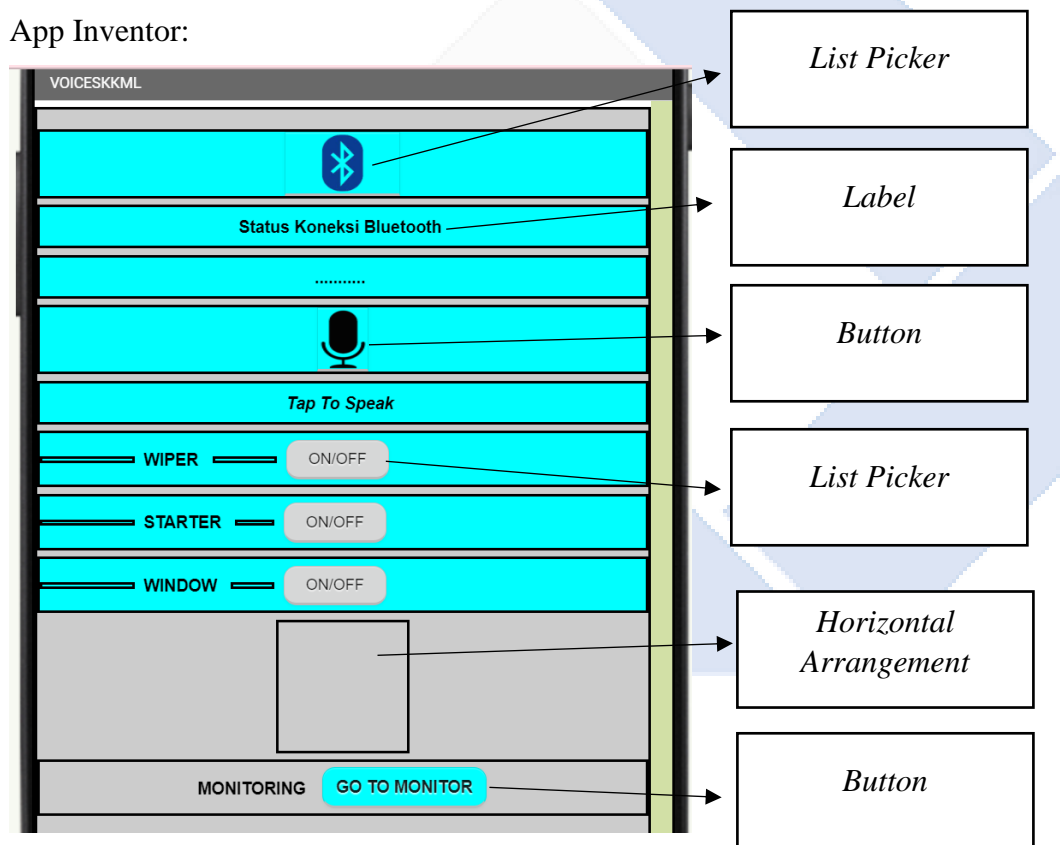
Tata letak *screen* portal login yang ditampilkan di layar *smartphone* pada aplikasi SKMML adalah sebagai berikut.



Gambar 4. 5 Tampilan *Screen Portal Login* pada *Smartphone*

### 3. Voice Control

*Voice control* berisi tampilan untuk mengontrol pergerakan dari masing-masing peralatan mobil listrik seperti *starter*, *wiper*, dan pergerakan kaca jendela dengan memberikan instruksi berupa perintah suara. Pada *screen voice control* terdapat simbol Bluetooth untuk menghubungkan koneksi dengan modul Bluetooth HC-05, selain itu pada menu ini terdapat *icon microphone* untuk memberikan perintah suara untuk mengontrol pergerakan dari masing-masing peralatan mobil listrik. Pada *screen* ini juga terdapat informasi keterangan dari masing-masing peralatan mobil listrik apakah dalam kondisi ON atau OFF maupun TERBUKA atau TERTUTUP. Berikut tampilan Menu Utama Kontrol yang dibuat menggunakan aplikasi MIT App Inventor:



Gambar 4. 6 Desain *Screen Voice Control* Aplikasi SKMML

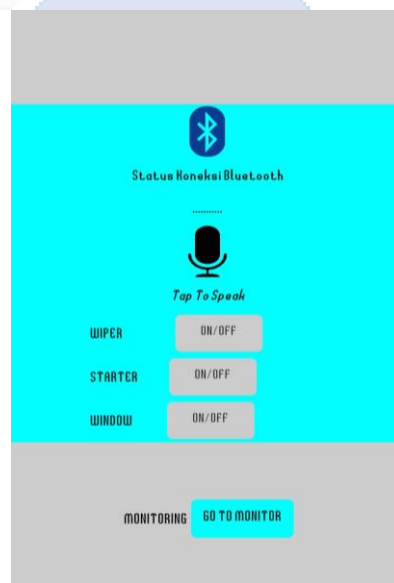
Berikut keterangan dari *interface screen voice control* pada aplikasi SKMML:

- *List Picker*, berfungsi menampilkan daftar yang dapat dipilih oleh user ketika pengguna menekan daftar pilihan. Pada aplikasi SKMML *list picker*

digunakan untuk memilih koneksi Bluetooth untuk menghubungkan ke Modul Bluetooth HC-05, serta *list picker* juga digunakan untuk *update* status dari masing-masing peralatan apakah dalam kondisi ON atau OFF maupun dalam kondisi TERBUKA atau TERTUTUP.

- *Label*, berfungsi untuk menuliskan sebuah kalimat. Selain itu, label bisa digunakan menampilkan kalimat yang ingin ditampilkan pada *screen* aplikasi seperti pemberian informasi.
- *Button*, berfungsi untuk pendeteksi ketukan ketika *user* menyentuh tombol. Tombol kan mendeteksi sentuhan tersebut kemudian akan menjalankan perintah sesuai dengan blok program. Pada *screen* ini terdapat dua buah tombol yaitu *icon microphone* dan GO TO MONITORING. Button microphone digunakan untuk merekam perintah suara sedangkan *button* GO TO MONITORING digunakan untuk *screen* selanjutnya.
- *Horizontal Arrangement*, berfungsi untuk desain tampilan untuk menempatkan komponen secara horizontal dan memberikan ruang di antara komponen untuk kejelasan dan kerapian.

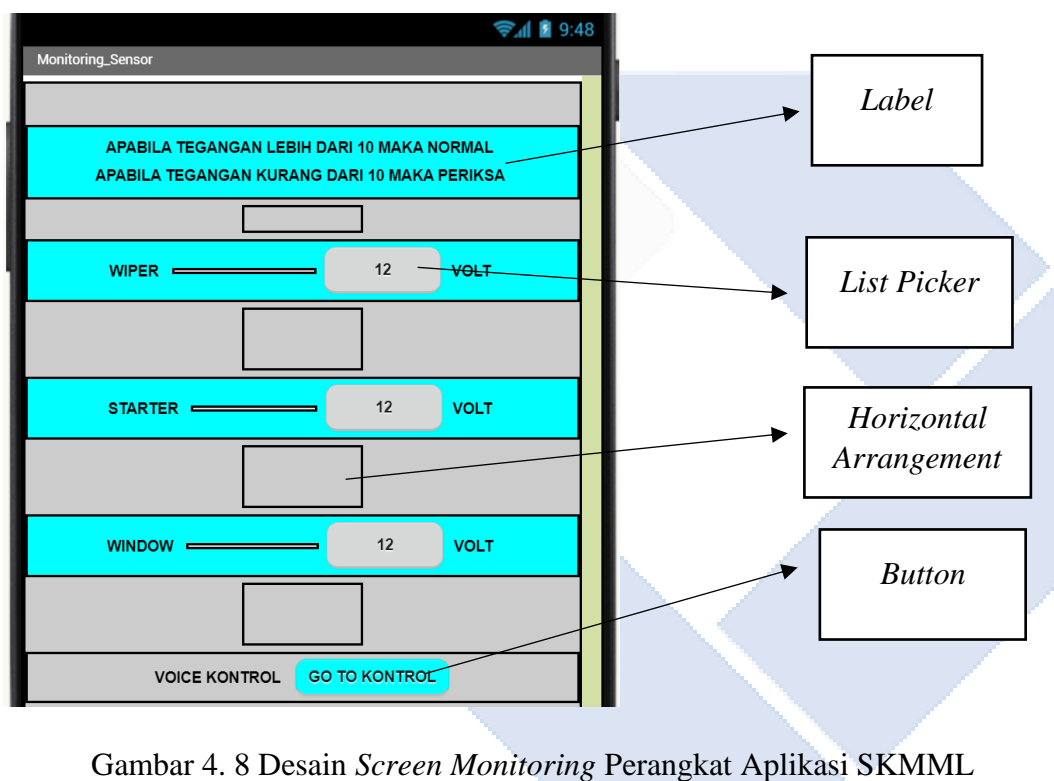
Tata letak *screen voice control* yang ditampilkan aplikasi SKMML di layar *smartphone* adalah sebagai berikut.



Gambar 4. 7 Tampilan *Screen Voice Control* pada *Smartphone*

#### 4. *Monitoring* Perangkat

*Monitoring* perangkat berisi tampilan yang berhubungan dengan observasi data sensor tegangan. *Screen monitoring* berisi informasi mengenai data-data dari sensor tegangan yang terpasang pada masing-masing motor DC, sehingga memungkinkan user untuk melihat *update* data dari sensor tegangan. Pada menu ini memuat informasi batas tegangan optimum untuk tegangan operasi masing-masing motor DC. Berikut ini adalah tampilan Menu *monitoring* yang dibuat melalui *platform* MIT App Inventor:



Gambar 4. 8 Desain *Screen Monitoring* Perangkat Aplikasi SKMML

Berikut keterangan dari *interface screen monitoring* perangkat pada aplikasi SKMML:

- *Label*, berfungsi untuk menuliskan sebuah kalimat. Selain itu, *label* bisa digunakan menampilkan kalimat yang ingin ditampilkan pada *screen* aplikasi seperti pemberian informasi.
- *List Picker*, berfungsi menampilkan daftar yang dapat dipilih oleh user ketika pengguna menekan daftar pilihan. Pada aplikasi SKMML *list picker*



digunakan untuk menampilkan informasi mengenai data sensor tegangan untuk memantau kondisi dari masing-masing motor DC.

- *Horizontal Arrangement*, berfungsi untuk desain tampilan untuk menempatkan komponen secara horizontal dan memberikan ruang di antara komponen untuk kejelasan dan kerapian.
- *Button*, berfungsi untuk pendeteksi ketukan ketika user menyentuh tombol. Tombol kan mendeteksi sentuhan tersebut kemudian akan menjalankan perintah sesuai dengan blok program Pada menu ini terdapat satu buah *button* yang betuliskan GO TO CONTROL, dimana *button* ini berfungsi memberikan perintah untuk menuju ke halaman kontrol mobil listrik.

Tata letak *screen monitoring* perangkat yang ditampilkan di layar smartphone pada aplikasi SKMML adalah sebagai berikut.

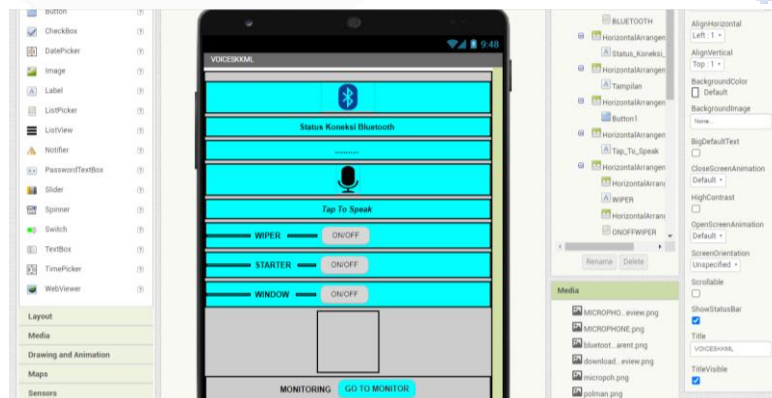


Gambar 4. 9 Tampilan *Scéen Monitoring* Perangkat Pada *Smartphone*

#### 4.2.2 Perancangan dan Pembuatan Perintah Suara

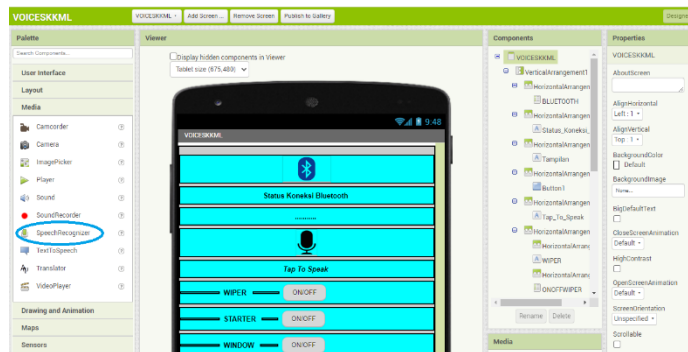
*Speech recognizer* adalah media penerimaan dan pemahan suatu perintah suara yang diucapkan dengan mendigitalkan kata-kata dan pencocokkan sinyal digital dengan acuan tertentu yang telah diatur. Pada proyek akhir ini *platform* yang digunakan untuk memfasilitasi *speech recognizer* sudah tersedia pada MIT App Inventor. Perancangan pengolahan perintah suara dibuat pada MIT App Inventor dengan memanfaatkan fitur *speech recognizer*, setelah user memberikan perintah suara maka akan dilakukan perbandingan dengan data perintah suara berupa teks yang telah diatur pada blok program aplikasi. *Speech recognizer* yang terdapat pada MIT App Inventor langsung terhubung kepada *google voice* sebagai fasilitator penerimaan perintah suara. Cara menambahkan media *speech recognizer* dalam MIT App Inventor sebagai berikut.

1. Merancang *interface* aplikasi kontrol suara yang akan dibuat pada menu *designer* yang terdapat pada MIT App Inventor. Berikut salah satu contoh desain tampilan *interface* aplikasi kontrol suara.



Gambar 4. 10 Tampilan Rancangan *Interface* Aplikasi

2. Menambahkan media *speech recognizer* pada menu Media yang terdapat pada menu MIT App Inventor. Berikut cara penambahan media *speech recognizer* pada MIT App Inventor dapat dilihat pada Gambar 4.11 di bawah ini.



Gambar 4. 11 Cara Menambahkan *Speech Recognizer*

Komponen *speech recognizer* terletak di menu media di sisi kiri tampilan MIT App Inventor. Langkah selanjutnya ialah pembuatan blok program yang berfungsi merubah perintah suara menjadi sebuah teks serta melakukan perbandingan perintah suara yang masuk dengan data yang telah diatur sebelumnya sebagai berikut.

```

when Button1.Click
do call SpeechRecognizer1.GetText

when SpeechRecognizer1.AfterGettingText
result partial
do set Tampilan.Text to SpeechRecognizer1.Result
if compare texts "Close application" = Tampilan.Text
then close application
else if compare texts "wiper on" = Tampilan.Text
then call TextToSpeech1.Speak
message wiper menyala
set ONOFFWIPER.Text to ON
set ONOFFWIPER.BackgroundColor to green
else if compare texts "wiper off" = Tampilan.Text
then call TextToSpeech1.Speak
message wiper mati
set ONOFFWIPER.Text to OFF
set ONOFFWIPER.BackgroundColor to red
else if compare texts "starter on" = Tampilan.Text
then call TextToSpeech1.Speak
message Mesin menyala
set ONOFFSTARTER.Text to ON
set ONOFFSTARTER.BackgroundColor to green
else if compare texts "starter of" = Tampilan.Text
then call TextToSpeech1.Speak
message Mesin mati
set ONOFFSTARTER.Text to OFF
set ONOFFSTARTER.BackgroundColor to red

```

Gambar 4. 12 Program Pengubahan Perintah Suara Menjadi Teks

Program blok di atas berfungsi untuk merubah perintah suara menjadi sebuah teks dengan memanggil *speech recognizer* untuk mendapatkan teks. Setelah perintah suara yang telah diubah menjadi sebuah teks akan dilakukan perbandingan dengan teks yang telah ditentukan menjadi sebuah perintah. Jika perintah suara yang didapatkan sama dengan kode perintah yang telah ditentukan maka aplikasi akan memberikan respon jawaban dari perintah tersebut

#### 4.2.3 Perancangan dan Pembuatan Database

Basis data atau *database* ialah kumpulan informasi atau data tersimpan pada suatu sistem yang dapat diakses untuk berbagai keperluan. Dalam proyek akhir ini, *firebase* digunakan sebagai sebuah *platform* wadah untuk membuat dan menyimpan database. Berikut adalah data base pada *firebase* untuk keperluan *monitoring* sensor tegangan yang terpasang pada masing-masing peralatan mobil listrik:



Gambar 4. 13 Pembuatan *Database*

#### 4.2.4 Komunikasi Serial pada Platform MIT App Inventor

*Interface* ialah representasi tampilan asli dari aplikasi SKMML yang dirancang untuk memberikan kemudahan bagi pengguna (*user*) untuk menggunakan aplikasi. *Interface* aplikasi SKMML yang dibangun di MIT App Inventor mengirimkan perintah suara dan memantau perubahan data sensor untuk setiap perangkat mobil listrik seperti *starter*, *wiper*, dan kaca jendela melalui aplikasi SKMML yang telah terinstal pada *smartphone* pengguna. Agar aplikasi SKMML dapat berfungsi sesuai dengan desain yang telah dibuat maka perlu menjalin komunikasi serial antara

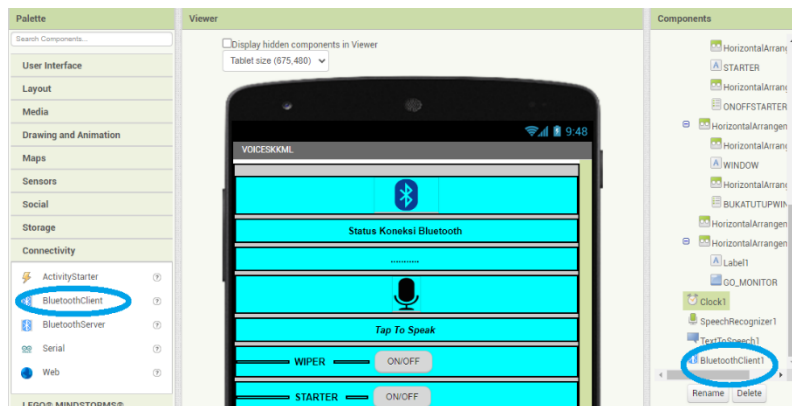
Arduino Mega 2560 terhadap aplikasi SKMML. Terdapat tiga komunikasi serial yang digunakan dalam transfer data pada aplikasi SKMML yaitu:

1. Modul Bluetooth HC-05, digunakan untuk mentransfer data perintah suara yang telah diubah kedalam bentuk teks oleh *speech recognizer* dari aplikasi SKMML ke Arduino Mega 2560. Ketika transfer data selesai kemudian data yang diterima akan langsung diolah sesuai dengan program yang telah dibuat pada Arduino Mega 2560.
2. Arduino Mega 2560, digunakan untuk mikrokontroler membaca data sensor tegangan kemudian mengirimkan data dari Arduino Mega 2560 kepada NodeMCU. Setelah data diterima oleh NodeMCU maka data dikirimkan ke *platform firebase*.
3. Firebase, digunakan sebagai media penyimpanan data sensor yang telah dikirimkan oleh NodeMCU. Blok program yang telah dibuat pada aplikasi SKMML akan memanggil data yang tersimpan pada *firebase* kemudian dapat menampilkan data sensor pada menu utama *monitoring* yang terdapat pada aplikasi SKMML.

#### **4.2.5 Koneksi Modul Bluetooth HC-05 dengan MIT App Inventor**

Pengiriman perintah suara yang telah diolah oleh *speech recognizer* pada aplikasi MIT App Inventor memanfaatkan sambungan nirkabel Modul Bluetooth HC-05. Untuk mengirimkan data perintah suara yang terdapat pada aplikasi MIT App Inventor diperlukan sebuah program cara untuk menghubungkan koneksi antara Modul Bluetooth HC-05 dengan aplikasi MIT App Inventor. Kemudian program pengambilan data perintah suara yang sudah dikonversi menjadi sebuah teks yang nantinya akan dikirimkan melalui Modul Bluetooth HC-05 sebagai media perantara transfer data ke Arduino Mega 2560. Berikut cara menghubungkan koneksi dari MIT App Inventor ke Modul Bluetooth HC-05.

1. Menambahkan Bluetooth *Client* pada interface aplikasi pada bagian *designer* dengan memilih menu *connectivity* seperti yang ditampilkan pada Gambar 4.14 dibawah ini.



Gambar 4. 14 Cara Menambahkan Koneksi Bluetooth

2. Membuat blok program pada *platform* MIT App Inventor untuk menghubungkan Modul Bluetooth HC-05 terhadap MIT App Inventor. Berikut blok pemrograman menghubungkan koneksi dari *platform* MIT App Inventor ke Modul Bluetooth HC-05.

```

when VOICESKML . BackPressed
do
  call BluetoothClient1 . Disconnect
  close application

when BLUETOOTH . BeforePicking
do
  set BLUETOOTH . Elements to BluetoothClient1 . AddressesAndNames

when BLUETOOTH . AfterPicking
do
  if
    call BluetoothClient1 . Connect
    address BLUETOOTH . Selection
  then
    set BLUETOOTH . Elements to BluetoothClient1 . AddressesAndNames
    if
      BluetoothClient1 . IsConnected
    then
      set Status_Koneksi_Bluetooth . Text to "Terhubung"
      set Status_Koneksi_Bluetooth . TextColor to black
    else
      set Status_Koneksi_Bluetooth . Text to "Tidak Terhubung"
      set Status_Koneksi_Bluetooth . TextColor to red
  end
end

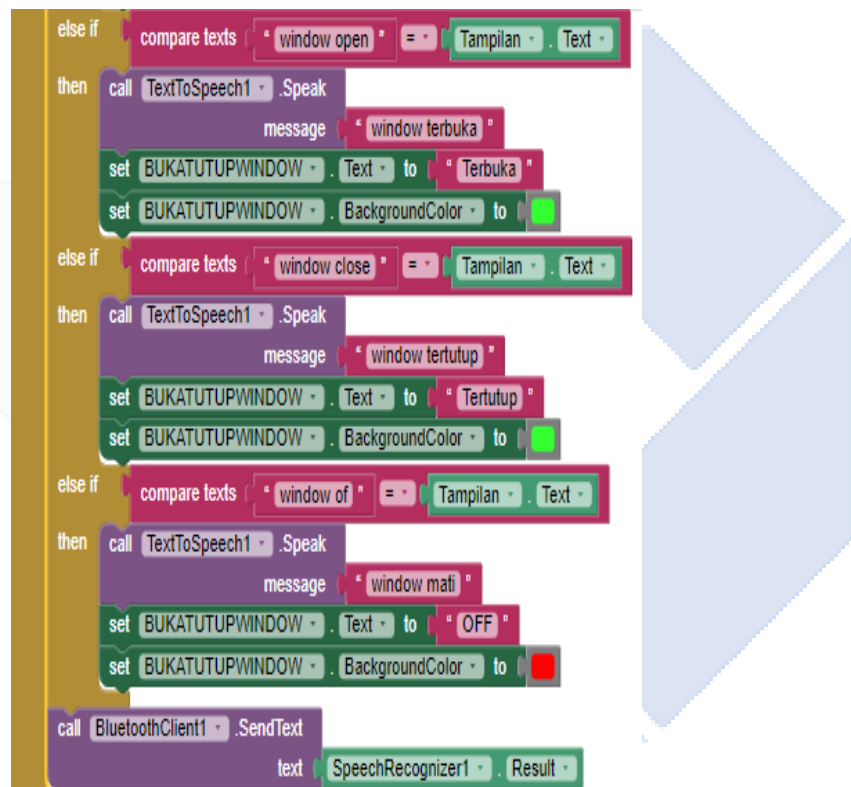
```

Gambar 4. 15 Program Blok Menghubungkan Koneksi Bluetooth

Program untuk menghubungkan koneksi dari aplikasi MIT App Inventor ke Modul Bluetooth HC-05. Pada program ini dibuat pemanggilan untuk menghubungkan konektifitas serta menampilkan status koneksi Bluetooth pada

interface aplikasi apakah konektifitas dari Bluetooth terhubung atau tidak terhubung.

Selain program pemanggilan untuk menghubungkan konektifitas antara Modul Bluetooth HC-05 ke platform MIT App Inventor, dibuat juga program yang berfungsi untuk mengirimkan data perintah suara yang telah dikonversikan dalam bentuk teks dari aplikasi MIT App Inventor ke Arduino Mega 2560 melalui perantara Modul Bluetooth HC-05. Salah satu contoh pemrograman untuk transfer data perintah suara dalam bentuk teks seperti dibawah ini.



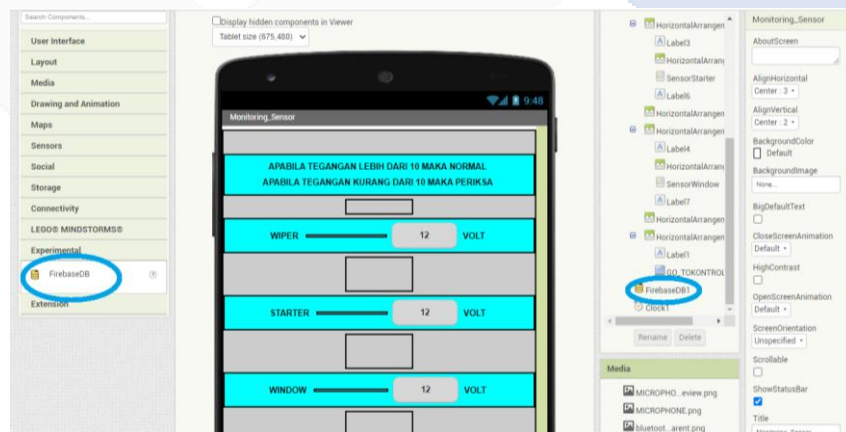
Gambar 4. 16 Program Blok Pengiriman Data Perintah Suara Melalui Bluetooth

Program yang digunakan pengiriman data perintah suara melalui MIT App Inventor menuju Arduino Mega 2560. Ketika *user* atau pengguna sudah mengirimkan perintah suara, maka perintah suara yang masuk akan diolah menjadi teks dan dilakukan perbandingan dengan teks perintah yang sudah ditetapkan. Jika perintah tersebut sama maka perintah suara dalam bentuk teks tersebut akan dikirimkan melalui oleh Bluetooth *Client*.

#### 4.2.6 Menghubungkan Database dengan MIT App Inventor

Data tersimpan di *firebase* akan ditampilkan pada menu *monitoring* yang terdapat pada aplikasi SKMML. Untuk memproses *database* dan terkoneksi ke MIT App Inventor maka memerlukan program yang memanggil data disimpan di *firebase* untuk MIT App Inventor. Cara menghubungkan dari MIT App Inventor ke *firebase* seperti dibawah ini.

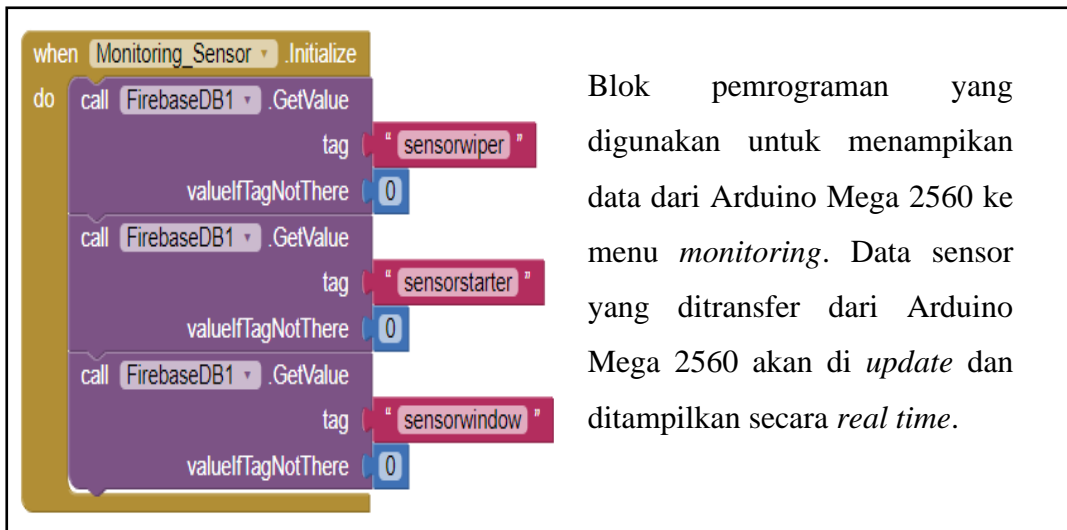
1. Pembuatan *database* pada *platform firebase* yang ditunjukkan pada Gambar 4.13.
2. Pada menu blok MIT App Inventor, buat blok pemrograman untuk menghubungkan *firebase* dengan MIT App Inventor. Berikut ini adalah program pemanggilan kembali data sensor dari *firebase* secara terprogram dan menampilkan data tersebut di tampilan menu utama monitoring.



Gambar 4. 17 Tampilan Menu *Experimental* pada *Software MIT App Inventor*

Untuk menghubungkan aplikasi MIT App Inventor ke *firebase*, perlu menambahkan komponen *FirebaseDB* ke menu *designer*. Komponen ini tertletak di menu *eksperimental* di sisi kiri menu menu MIT App Inventor. Pencarian komponen *firebase* dapat dilihat pada gambar diatas. Setelah menambahkan komponen *FirebaseDB*, langkah selanjutnya adalah memprogram di menu *block* MIT App Inventor.





Blok pemrograman yang digunakan untuk menampilkan data dari Arduino Mega 2560 ke menu *monitoring*. Data sensor yang ditransfer dari Arduino Mega 2560 akan di *update* dan ditampilkan secara *real time*.

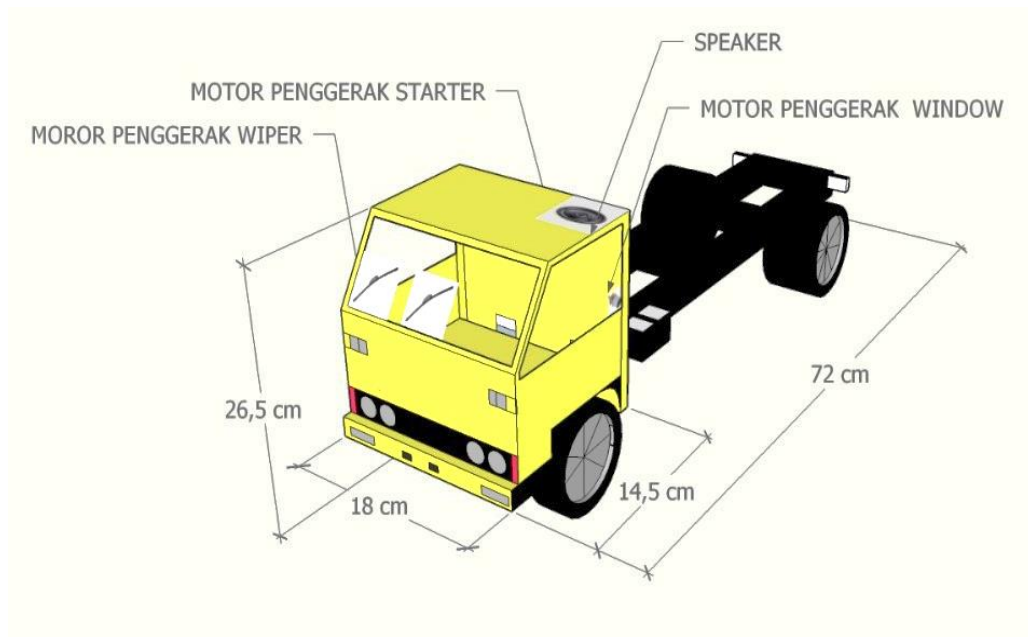
Gambar 4. 18 Program Blok Pemanggilan Data Sensor Tegangan

### 4.3 Perancangan dan Pembuatan Hardware

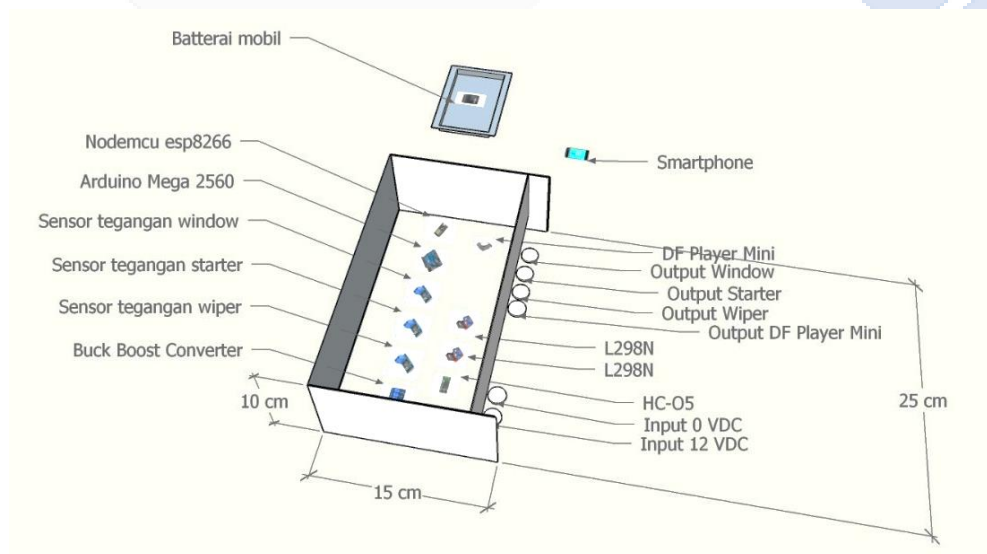
Merancang dan membuat *hardware* terdiri dari dua bagian yaitu bagian elektrik dan mekanik. Tahapan perancangan dan pembuatan *hardware* dari proyek akhir ini sebagai berikut

#### 4.3.1 Merancang Hardware secara Mekanik

Tahap merancang *hardware* mekanik ialah membuat desain konstruksi prototype mobil listrik pada aplikasi *sketch up*. Bahan utama pembuatan miniatur mobil listrik terbuat dari kayu dan PVC yang dibentuk seperti mobil truk. Ukuran panjang, lebar, dan tinggi dari kabin mobil ini yaitu 14,5cm, 18cm, dan 26,5cm serta panjang keseluruhan miniatur mobil listrik yaitu 72cm. Kemudian rancangan sistem kontrol *prototype* mobil listrik akan diletakkan dalam box dengan ukuran panjang, lebar, dan tinggi ialah 25cm, 15cm, dan 10cm. Rancangan konstruksi *prototype* mobil listrik beserta box kontrol dapat dilihat sebagai berikut.



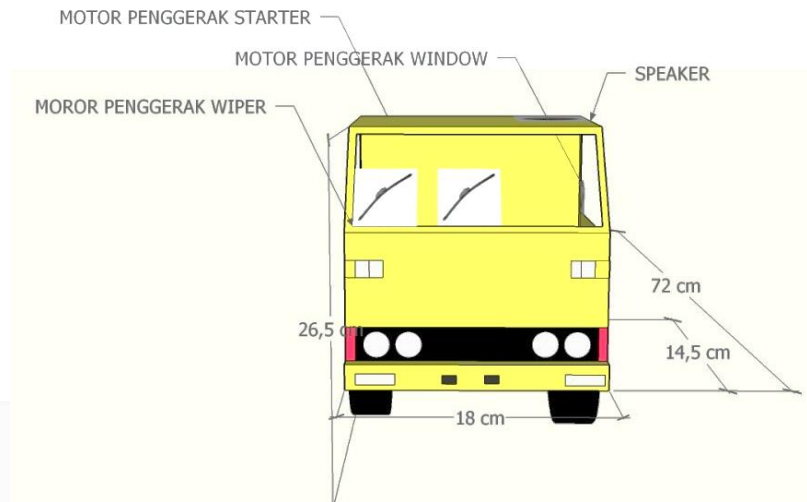
Gambar 4. 19 Desain Konstruksi *Prototype* Mobil Listrik



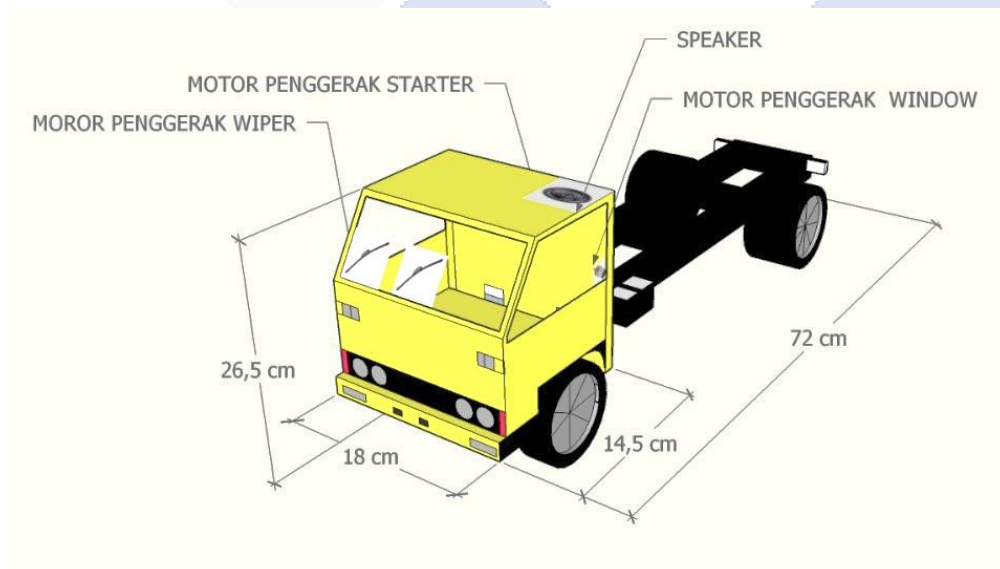
Gambar 4. 20 Desain Box Kontrol

Pada tahap ini dibuat rancangan penempatan posisi dari masing-masing komponen. Motor DC *window* diletakkan pada pintu miniatur mobil yang digunakan untuk menggerakkan kaca jendela mobil. Motor DC penggerak *starter*

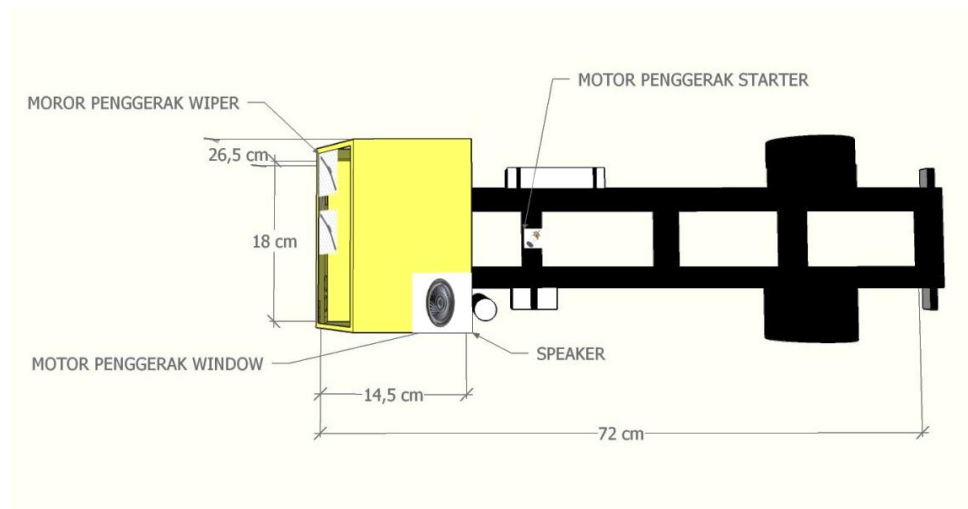
diletakkan pada bagian belakang kabin. Motor DC penggerak *wiper* diletakkan pada bagian dalam kabin depan. Kemudian yang terakhir *speaker* diletakkan pada sisi atas bagian belakang kabin miniatur mobil. Berikut adalah desain dari miniatur mobil listrik tampak depan, samping, dan atas.



Gambar 4. 21 Desain *Prototype* Tampak Depan



Gambar 4. 22 Desain *Prototype* Tampak Samping



Gambar 4. 23 Desain *Prototype* Tampak Atas

#### 4.3.2 Pembuatan Hardware secara Mekanik

Miniatur mobil listrik dibuat menggunakan bahan dasar kayu sebagai kerangka. Kabin dari mobil listrik ini dibuat dari kayu yang telah dipahat kemudian dihubungkan dengan menggunakan paku. Pintu miniatur mobil ini terbuat dari PVC yang telah melalui tahap pemanasan kemudian di potong sesuai dengan pola yang telah dibuat.

Kemudian pada kabin depan dipasang konstruksi penggerak *wiper* dan motor DC. Pada pintu mobil dipasang konstruksi penggerak kaca jendela yang terbuat dari PVC serta pada bagian pintu mobil juga dipasang motor DC. Motor DC sebagai *starter* dipasang pada bagian belakang kabin mobil.

Box kontrol dari *prototype* mobil listrik ini dibuat dengan menempelkan akrilik bening telah diukur dan dipotong membentuk balok menggunakan lem akrilik. Box kontrol ini dipasang dua buah engsel agar bagian atas box dapat dibuka dan ditutup. Pada bagian dalam box kontrol ini diletakkan komponen-komponen diantaranya Arduino Mega 2560, NodeMCU ESP8266, Driver L298N, Buck Boost, Sensor Tegangan, Modul Bluetooth HC-05, dan DF Mini Player. Hasil akhir dari pembuatan *prototype* mobil listrik dan box kontrol adalah sebagai berikut.



Gambar 4. 24 *Prototype* Mobil Listrik Tampak Depan



Gambar 4. 25 *Prototype* Mobil Listrik Tampak Samping

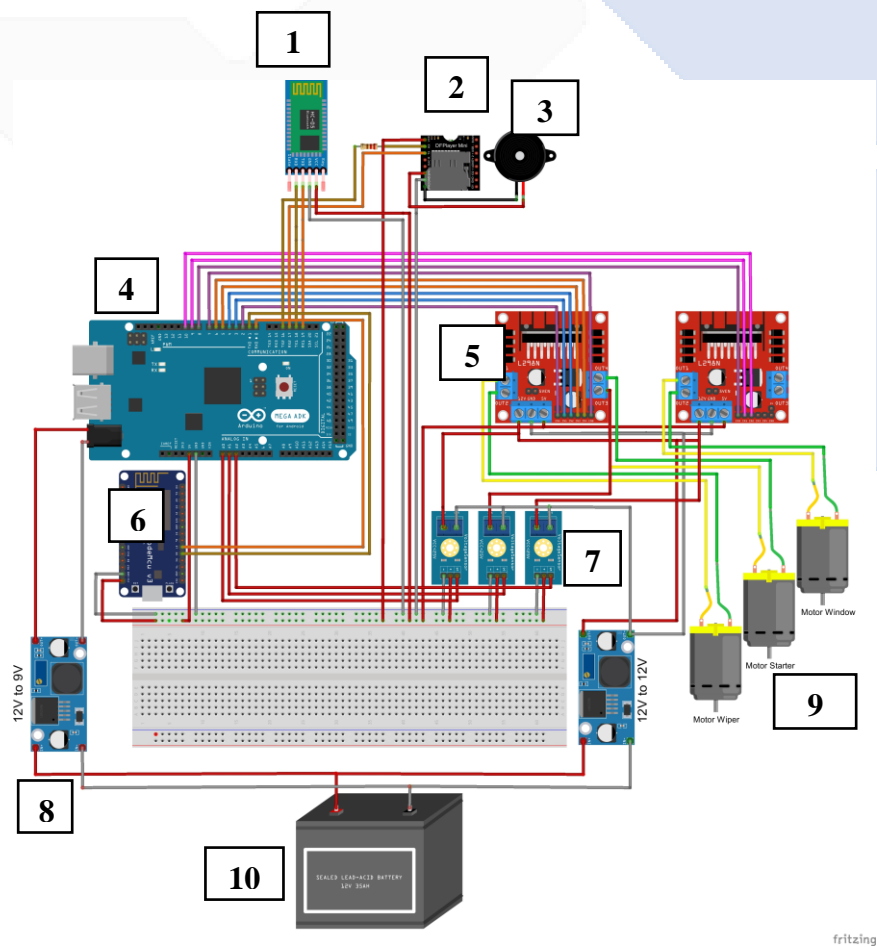


Gambar 4. 26 *Prototype* Mobil Listrik Tampak Atas

### 4.3.3 Merancang dan Membuat Hardware Elektrik

Pada tahap merancang dan membuat *hardware* elektrik merupakan bagian penting dari proyek akhir ini, dimana pada tahap ini dilakukan perancangan sistem kontrol dan *monitoring* peralatan mobil listrik. Setelah perancangan selesai selanjutnya tahapan membuat hasil rancangan *hardware* tersebut. Adapun tahapan dari proses ini sebagai berikut.

1. Merancang *wiring* untuk rangkaian kontrol dan *monitoring* pada *prototype* mobil listrik menggunakan aplikasi Fritzing. Pembuatan rancangan *wiring* ini berfungsi untuk pemasangan pengkabelan sehingga tidak terjadi kesalahan. Adapun skema pengkabelan dari komponen yang digunakan adalah sebagai berikut.

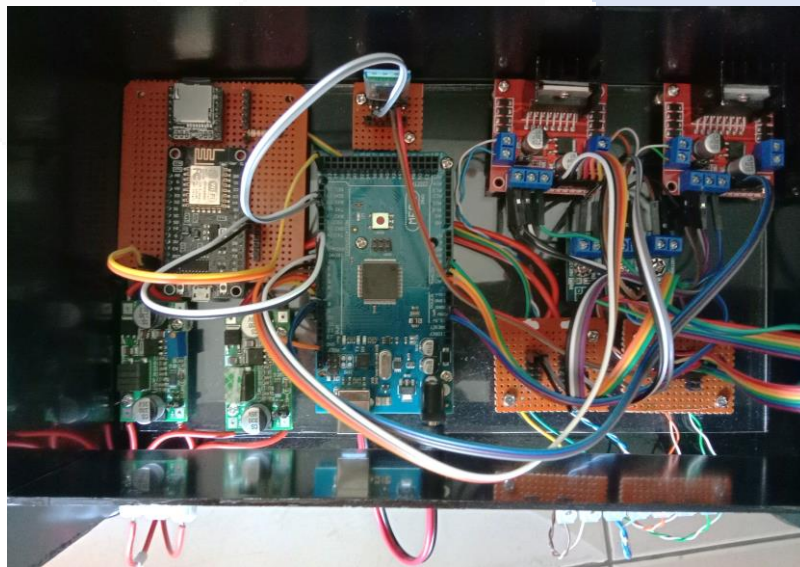


Gambar 4. 27 *Wiring* Diagram Sistem Kontrol dan *Monitoring*

Keterangan:

1. Modul Bluetooth HC-05
2. DF Mini Player
3. Speaker
4. Arduino Mega 2560
5. Driver Motor L289N
6. NodeMCU ESP8266
7. Sensor Tegangan DC
8. Buck Boost Converter
9. Motor DC Gear Box
10. Accu 12V

2. Menghubungkan modul Bluetooth HC-05, DF Mini Player, driver motor L289N, NodeMCU ESP8266, dan sensor tegangan DC pada pin Arduino Mega 2560. Hasil akhir pemasangan *wiring* sistem kontrol dan *monitoring* pada box kontrol dan miniatur mobil adalah sebagai berikut.



Gambar 4. 28 Hasil Akhir Pemasangan Pengkabelan

#### 4.4 Pengujian Hardware Elektrik

Pengujian *hardware* elektrik dimaksudkan untuk mengetahui fungsionalitas dan kesesuaian komponen yang digunakan dalam pembuatan proyek akhir ini.

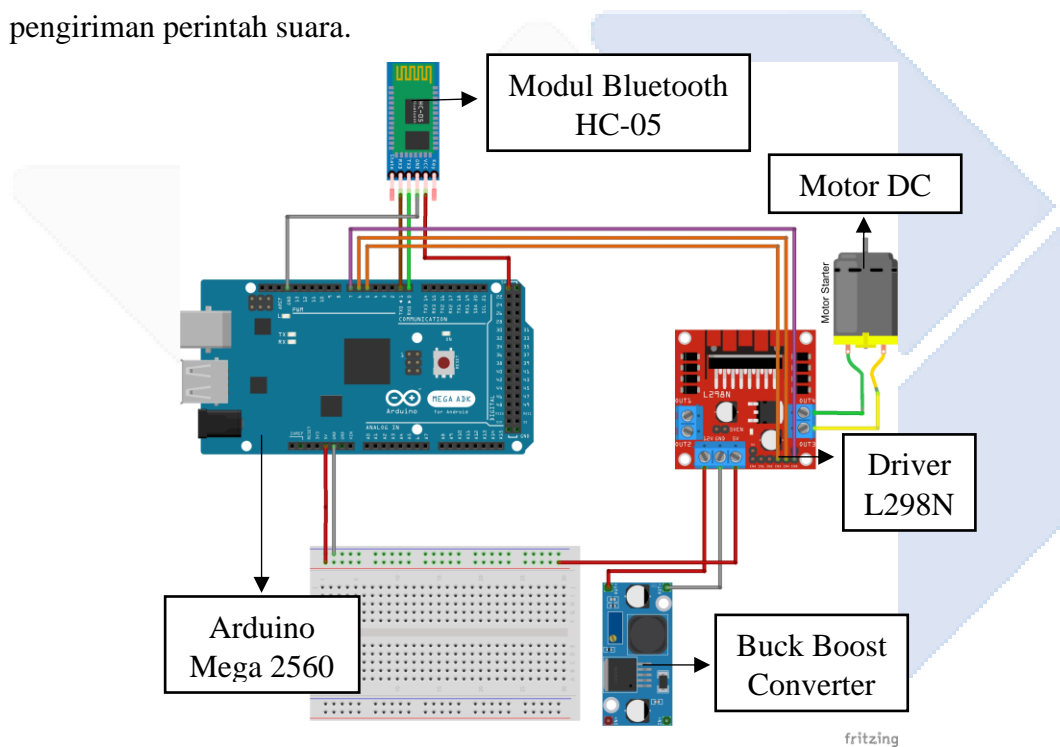
##### 4.4.1 Pengujian Pengiriman Perintah Suara Melalui Modul Bluetooth

Pengujian ini berfungsi adalah mengetahui jarak pendeksian serta kecepatan pengiriman dan penerimaan data perintah suara melalui Modul Bluetooth HC-05 ke

Arduino Mega. Setelah mengetahui kemampuan pengiriman dan penerimaan data perintah suara maka akan lebih mudah untuk menentukan batasan jarak minimum dan maksimum dalam memberikan perintah suara untuk mengontrol pergerakan dari Motor DC pada *prototype* mobil listrik. Adapun pengujian ini terdiri dari tahapan dibawah ini.

#### 4.4.1.1 Perancangan dan Pembuatan Pengiriman Perintah Suara

Pada tahapan ini perancangan *wiring* diagram dengan menggunakan aplikasi Fritzing untuk menghubungkan modul bluetooth HC-05 dan Driver Motor L298N ke Arduino Mega 2560. Dibawah ini adalah *wiring* diagram dari rangkaian pengiriman perintah suara.



Gambar 4. 29 Skema Rangkaian Pengujian Pengiriman Perintah Suara

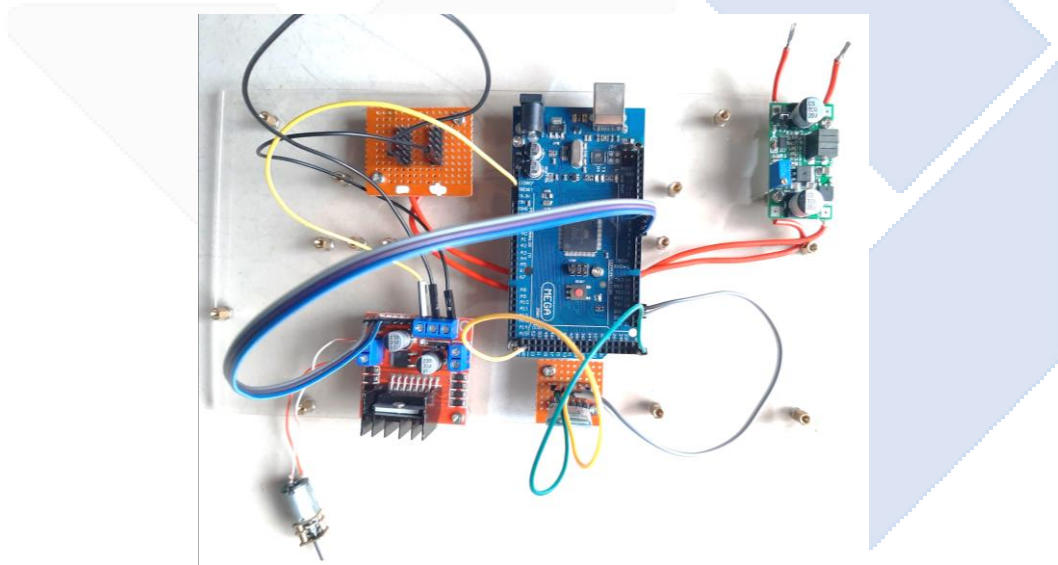
Dilihat dari Gambar, berikut adalah penjelasan inisialisasi pin yang digunakan pada rangkaian tersebut:

- Pin Vcc Modul Bluetooth HC-05 dihubungkan pada pin 5V Arduino Mega 2560



- Pin Gnd Modul Bluetooth HC-05 dihubungkan pada pin Gnd Arduino Mega 2560
- Pin Tx Modul Bluetooth HC-05 dihubungkan pada pin 0 Arduino Mega 2560
- Pin Rx Modul Bluetooth HC-05 dihubungkan pada pin 0 Arduino Mega 2560
- Pin EN Driver L298N dihubungkan pada pin 7 Arduino Mega 2560
- Pin IN1 Driver L298N dihubungkan pada pin 5 Arduino Mega 2560
- Pin IN2 Driver L298N dihubungkan pada pin 6 Arduino Mega 2560
- Pin Out 4 dan Out 5 Driver L298N ke Motor Dc

Hasil akhir dari rangkaian dapat dilihat pada gambar dibawah.



Gambar 4. 30 Rangkaian Pengujian Pengiriman Perintah Suara

Setelah melakukan perangkaian pada masing-masing komponen tahapan selanjutnya pemograman pada Arduino IDE untuk pengiriman perintah suara melalui modul bluetooth terhadap pergerakan motor dc yaitu sebagai berikut.

```
#include <SoftwareSerial.h>
#include <Wire.h>
#define ENB 7 // deklarasi pin ENB
#define IN3 5 // deklarasi pin IN3
```

```

#define IN4 6 // deklarasi pin IN4

String voice;
void setup()
{
  Serial1.begin(9600);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
}

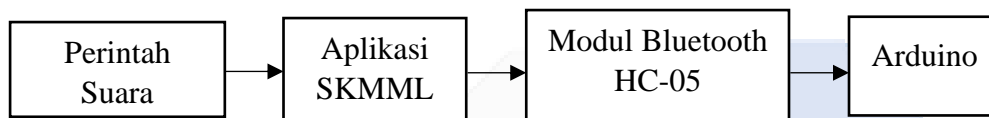
while (Serial1.available()) {
  delay (10);
  char c = Serial1.read ();
  if (c == '#') {
    break;
  }
  voice += c;
}

if (voice.length() > 0) {
  Serial.println(voice);
  if (voice == "starter on" )
  {
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
    delay(2000); // Jeda 2 detik
  }
  else if (voice == "starter of" ) {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 0);
    delay(2000);
  }
  voice = "";}}

```

#### 4.4.1.2 Prosedur Pengujian Pengiriman Perintah Suara

Pengujian ini menggunakan input perintah suara untuk mengontrol pergerakan motor DC *starter* dan dilakukan pada area dengan halangan dan tanpa halangan untuk mengetahui jarak dan kecepatan pengiriman data oleh modul Bluetooth HC-05. Kemudian hasil pengujian dari pengiriman perintah suara melalui modul bluetooth terhadap pergerakan motor dc ini akan dicatat dalam tabel. Adapun blok diagram prosedur pengujian pengiriman perintah suara menggunakan modul Bluetooth sebagai berikut.



Gambar 4. 31 Blok Diagram Prosedur Pengujian Pengiriman Perintah Suara

#### 4.4.1.3 Hasil Pengujian Pengiriman Perintah Suara Melalui Modul Bluetooth Terhadap Pergerakan Motor DC

Pengujian pengiriman perintah suara melalui modul Bluetooth dilakukan dengan dua cara yaitu pengujian pada area dengan halangan dan pada area tanpa halangan.

- Pengujian pada area dengan halangan dilakukan di arena perumahan masyarakat dengan melakukan pengiriman suara sampai pada jarak maksimum pengiriman perintah suara dengan percobaan pengiriman perintah suara setiap 1 meter dan dihitung kecepatan pendeteksian perintah suara. Adapun proses pengujian dapat dilihat pada gambar dibawah ini.



Gambar 4. 32 Pengujian Pada Area Dengan Halangan

Berikut adalah tabel hasil pengujian pengiriman perintah suara melalui modul Bluetooth HC-05 terhadap pergerakan motor DC di area dengan halangan.

Tabel 4. 1 Hasil Pengujian Pengiriman Perintah Suara Pada Area dengan Halangan

Percobaan Ke	Jarak Pengiriman (Dengan Halangan)	Keterangan	Durasi Pengiriman (s)
1	1 m	Terkirim	4
2	2 m	Terkirim	4
3	3 m	Terkirim	4
4	4 m	Terkirim	4
5	5 m	Terkirim	4
6	6 m	Terkirim	4
7	7 m	Terkirim	4
8	8 m	Terkirim	4
9	9 m	Terkirim	4
10	10 m	Terkirim	4
11	11 m	Terkirim	4
12	12 m	Terkirim	4
13	13 m	Terkirim	4

14	14 m	Terkirim	4
15	15 m	Terkirim	4
16	16 m	Terkirim	4

- Pengujian pada area tanpa halangan dilakukan di lapangan olahraga Polmanbabel dengan melakukan pengiriman suara sampai pada jarak maksimum pengiriman perintah suara dengan percobaan pengiriman perintah suara setiap 1 meter dan dihitung kecepatan pendeteksian perintah suara. Adapun proses pengujian pengiriman perintah suara pada area tanpa halangan sebagai berikut.



Gambar 4. 33 Pengujian Pengiriman Perintah Suara Pada Area Tanpa Halangan  
Berikut adalah tabel hasil pengujian pengiriman perintah suara melalui modul Bluetooth HC-05 terhadap pergerakan motor dc di area tanpa halangan.

Tabel 4. 2 Hasil Pengujian Pengiriman Perintah Suara Pada Area Tanpa Halangan

Percobaan Ke	Jarak Pengiriman (Tanpa Halangan)	Keterangan	Durasi Pengiriman (s)
1	1 m	Terkirim	4

2	2 m	Terkirim	4
3	3 m	Terkirim	4
4	4 m	Terkirim	4
5	5 m	Terkirim	4
6	6 m	Terkirim	4
7	7 m	Terkirim	4
8	8 m	Terkirim	4
9	9 m	Terkirim	4
10	10 m	Terkirim	4
11	11 m	Terkirim	4
12	12 m	Terkirim	4
13	13 m	Terkirim	4
14	14 m	Terkirim	4
15	15 m	Terkirim	4
16	16 m	Terkirim	4
17	17 m	Terkirim	4
18	18 m	Terkirim	4
19	19 m	Terkirim	4
20	20 m	Terkirim	4
21	21 m	Terkirim	5
22	22 m	Terkirim	5
23	23 m	Terkirim	5
24	24 m	Terkirim	5
25	25 m	Terkirim	5
26	26 m	Terkirim	5
27	27 m	Terkirim	5
28	28 m	Terkirim	5
29	29 m	Terkirim	5
30	30 m	Terkirim	5
31	31 m	Terkirim	5

32	32 m	Terkirim	5
33	33 m	Terkirim	5
34	34 m	Terkirim	5
35	35 m	Terkirim	5
36	36 m	Terkirim	5
37	37 m	Terkirim	5
38	38 m	Terkirim	5
39	39 m	Terkirim	5
40	40 m	Terkirim	5

Kesimpulan:

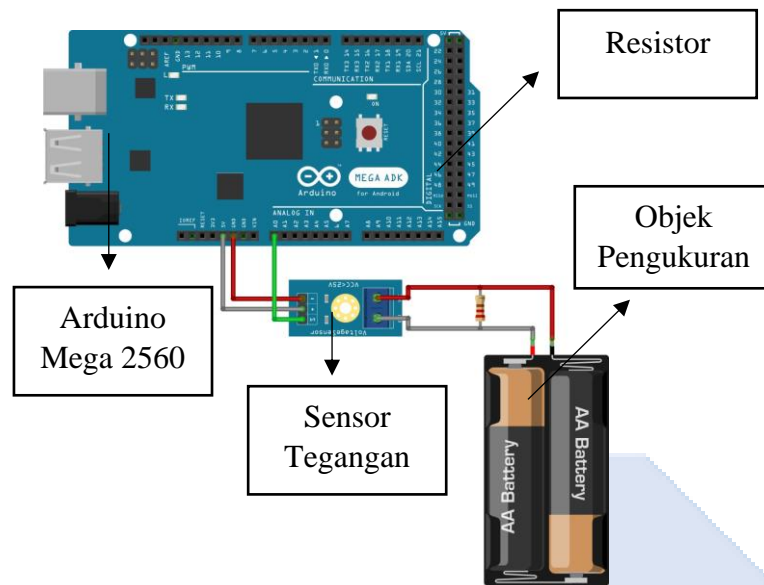
Dari dua tabel hasil pengujian diatas, kami menyimpulkan modul Bluetooth HC-05 mampu berfungsi dengan baik ketika jarak transmisi data di area berhalangan menacapai 16 meter dan jarak transmisi data di area tanpa halangan mencapai 40 meter, karena memiliki jarak transmisi data yang cukup baik dapat digunakan sebagai transmisi komunikasi perintah suara pada pembuatan proyek akhir.

#### **4.4.2 Pengujian Sensor Tegangan**

Tujuan dari pengujian ini adalah untuk mengetahui kemampuan deteksi dari sensor tegangan terhadap tegangan DC. Setelah mengetahui batas kemampuan deteksi dari sensor ini maka akan lebih mudah dalam penentuan penempatan sensor dalam mendeteksi tegangan pada motor DC sehingga dapat berfungsi secara maksimal. Adapun pengujian sensor tegangan ini meliputi tahap-tahap sebagai berikut.

##### **4.4.2.1 Perancangan dan Pembuatan Sensor Tegangan**

Pada tahap ini merancang dan membuat skema dengan menghubungkan pin sensor tegangan ke Arduino Mega 2560. Di bawah ini adalah skema rangkaian sensor tegangan menggunakan Arduino Mega 2560 melalui Fritzing.

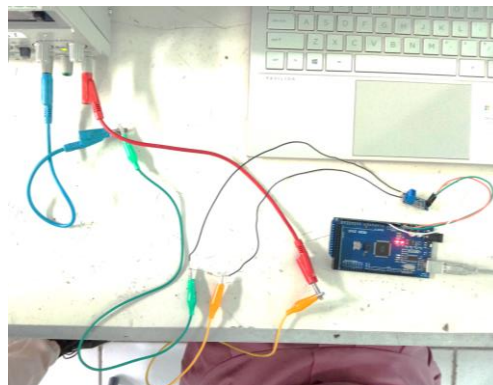


Gambar 4. 34 Skema Rangkaian Pengujian Sensor Tegangan

Dilihat dari Gambar, berikut adalah penjelasan inialisasi pin yang digunakan pada rangkaian tersebut:

- Pin Vcc(sumber) sensor tegangan dihububungkan pada pin 5V Arduino Mega 2560
- Pin A0 sensor tegangan dihubungkan pada pin A0 Arduino Mega 2560
- Pin Gnd(*ground*) sensor tegangan dihubungkan pada pin Gnd Arduino Mega 2560

Hasil akhir dari rangkaian dapat dilihat pada gambar dibawah.



Gambar 4. 35 Rangkaian Pengujian Sensor Tegangan



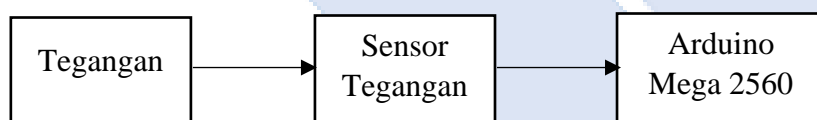
Setelah melakukan perangkaian pada masing-masing komponen tahapan selanjutnya pemograman pada Arduino IDE untuk menguji sensor tegangan sebagai berikut.

```
#include <Wire.h>
float Tegangan;
float BacaTegangan;

void setup()
{
  Serial.begin(9600);
  Serial.println("Tegangan: ");
}
void loop()
{
  BacaTegangan=analogRead(0);
  Tegangan=( (BacaTegangan*0.00489) *5);
  Serial.print(Tegangan);
  Serial.println("V");
  delay(1000);
}
```

#### 4.4.2.2 Prosedur Pengujian Sensor Tegangan

Pengujian sensor tegangan menggunakan input tegangan dari motor DC. Kemudian hasil pengujian dari sensor tegangan dicatat dalam Tabel 4.3. Prinsip kerja sensor tegangan didasarkan pada prinsip menekan resistansi dan mengurangi tegangan input menjadi seperlima dari tegangan aslinya. Berikut adalah blok diagram pengujian sensor tegangan.



Gambar 4. 36 Prosedur Pengujian Sensor Tegangan

#### 4.4.2.3 Hasil Pengujian Sensor Tegangan

Pengujian sensor tegangan ini menggunakan satu buah *power supply* sebagai objek tegangan yang diukur, kemudian dilakukan pengaturan kondisi tegangan yang diukur. Hasil pengukuran dari sensor tegangan dilakukan perbandingan dengan alat ukur multimeter. Berikut proses pengujian sensor tegangan dapat dilihat pada gambar berikut.



Gambar 4. 37 Proses Pengujian Sensor Tegangan

Berikut adalah hasil pengujian sensor tegangan dapat dilihat pada tabel dibawah ini.

Tabel 4. 3 Hasil Pengujian Sensor Tegangan

Percobaan	Sumber Tegangan	Hasil Pengukuran (V)		Persentase Error
		Multimeter	Sensor Tegangan	
1	1 V	0,95	0,96	1%
2	2 V	1,96	1,97	0,50%
3	3 V	2,92	2,93	0,20%
4	4 V	3,95	3,95	0%
5	5 V	4,96	4,97	0,20%
6	6 V	6,01	6	0,16%
7	7 V	6,99	6,99	0%
8	8 V	7,99	7,99	0%
9	9 V	9	9	0%
10	10 V	10,01	10,01	0%
11	11 V	11	11	0%
12	12 V	12,05	12,04	0,10%
Rata-rata Persentase Error				0,22%

Keterangan:

Rumus perhitungan persentase error:

$$\text{Persentase error} = \left| \frac{\text{Alat Ukur} - \text{Sensor Tegangan}}{\text{Alat Ukur}} \right| \times 100\%$$

Alat Ukur = Hasil pengukuran yang terukur pada multimeter

Sensor = Hasil pengukuran yang terbaca pada sensor tegangan

Berikut ini perhitungan persentase *error* secara teori:

Perhitungan persentase error pada tabel pengujian sensor tegangan.

- Persentase *error* percobaan ke-1 =  $\left| \frac{0,95-0,96}{0,95} \right| \times 100\% = 1\%$
- Persentase *error* percobaan ke-5 =  $\left| \frac{4,96-4,97}{4,96} \right| \times 100\% = 0,2\%$
- Persentase *error* percobaan ke-8 =  $\left| \frac{7,99-7,99}{7,99} \right| \times 100\% = 0\%$
- Persentase *error* percobaan ke-12 =  $\left| \frac{12,05-12,04}{12,05} \right| \times 100\% = 0,1\%$

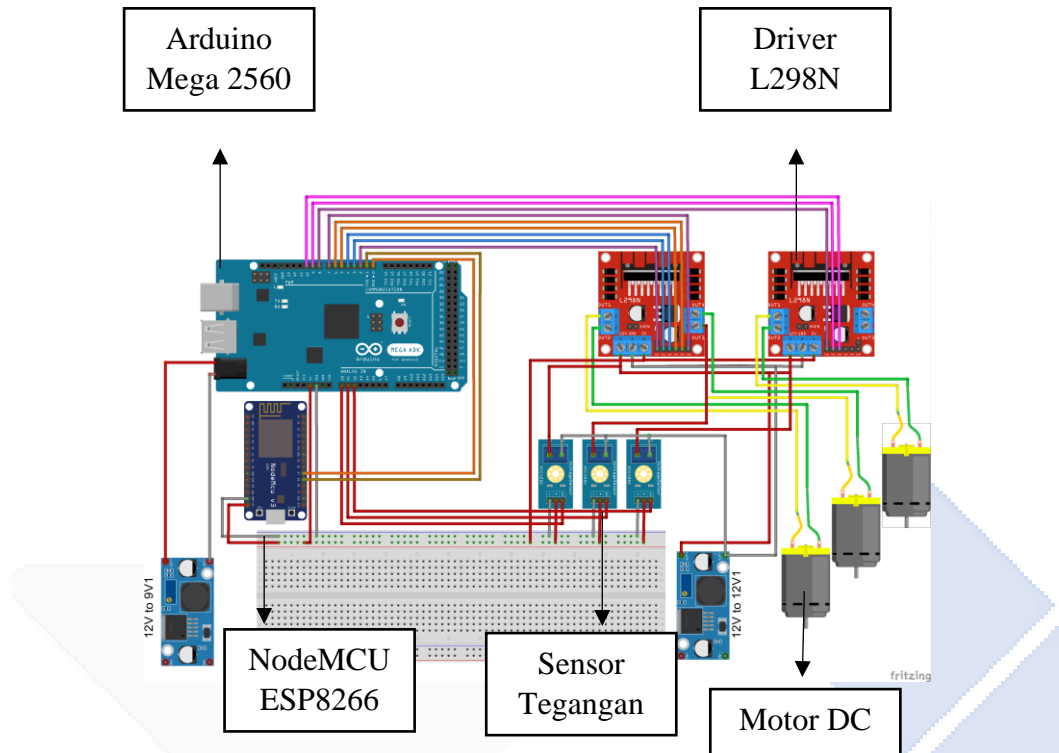
Kesimpulan :

Berdasarkan hasil pengujian dari sensor tegangan yang telah dilakukan, pada pengujian ini didapatkan rata-rata persentase *error* dari sensor tegangan sebesar 0,22% sehingga sensor ini dapat bekerja dengan baik untuk mengukur tegangan. Hal inilah yang mendasari bahwa sensor tegangan ini dapat digunakan dalam pembuatan proyek akhir.

#### **4.4.3 Pengujian Komunikasi Serial NodeMCU ESP8266**

Tujuan pengujian komunikasi serial ini untuk menilai kemampuan dari Arduino Mega 2560 dan NodeMCU dalam mengirim dan menerima data sensor serta jarak maksimum komunikasi serial. Setelah mengetahui kemampuan dan jarak maksimum komunikasi serial maka akan lebih mudah dalam menentukan penempatan posisi dari NodeMCU 2560. Tahap-tahap dalam pengujian sensor tegangan sebagai berikut.

#### 4.4.3.1 Perancangan dan Pembuatan Komunikasi Serial NodeMCU ESP8266



Gambar 4. 38 Skema Rangkaian Komunikasi Serial

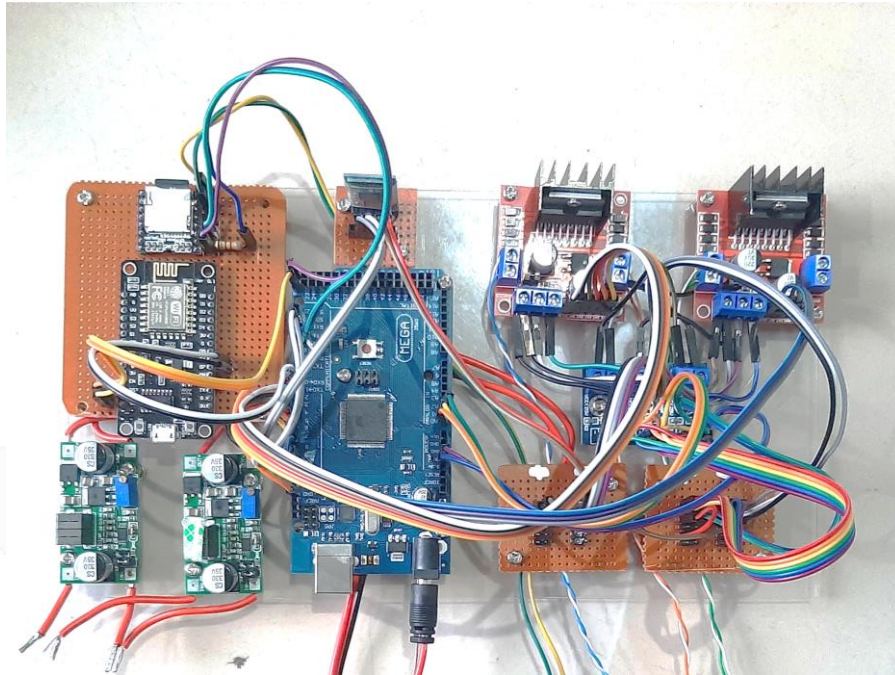
Perancangan *wiring* diagram untuk melakukan komunikasi serial NodeMCU ESP8266 dengan menggunakan aplikasi *fritzing*. Pada aplikasi *fritzing* dihubungkan pin dari NodeMCU ESP8266 ke pin Tx dan Rx Arduino Mega 2560. Setelah *wiring* diagram selesai, selanjutnya hasil dari *wiring* diagram dijadikan acuan untuk menghubungkan masing-masing pin yang digunakan pada komponen asli.

Dilihat dari gambar, berikut adalah penjelasan inisialisasi pin yang digunakan pada rangkaian tersebut:

- Pin Vin NodeMCU ESP8266 dihubungkan dengan pin 5V Arduino Mega 2560
- Pin Gnd NodeMCU ESP8266 dihubungkan dengan pin Gnd Arduino Mega 2560
- Pin D6 NodeMCU ESP8266 dihubungkan dengan pin 1 (Tx) Arduino Mega 2560

- Pin D7 NodeMCU ESP8266 dihubungkan dengan pin 0 (Rx) Arduino Mega 2560

Berikut adalah hasil akhir dari rangkaian komunikasi serial NodeMCU ESP8266.

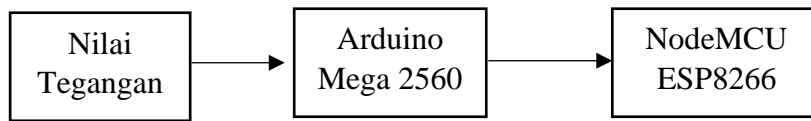


Gambar 4. 39 Rangkaian Pengujian Komunikasi Serial

Setelah selesai merangkai *wiring* diagram komunikasi serial NodeMCU ESP8266, selanjutnya membuat program pada *platform* Arduino IDE. Program dalam komunikasi serial ini terdiri atas dua program yaitu program pengiriman data oleh Arduino Mega 2560 dan program penerimaan data oleh NodeMCU ESP8266 dapat dilihat pada lampiran 1 dan lampiran 2.

#### 4.4.3.2 Prosedur Pengujian Komunikasi Serial NodeMCU ESP8266

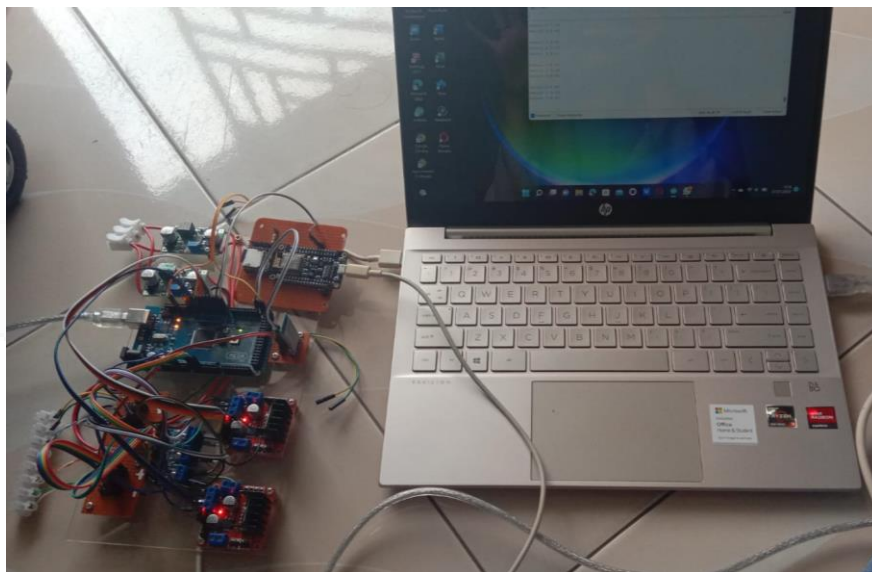
Pengujian komunikasi serial NodeMCU ESP8266 dengan Arduino Mega 2560 menggunakan data sensor tegangan sebagai input. Arduino Mega 2560 akan mengirimkan hasil pembacaan nilai sensor tegangan kemudian mengirimkan data hasil pembacaan nilai sensor ke NodeMCU ESP8266 melalui komunikasi serial dengan pin Tx dan Rx. Berikut adalah blok diagram pengujian komunikasi serial NodeMCU ESP8266.



Gambar 4. 40 Prosedur Pengujian Komunikasi Serial

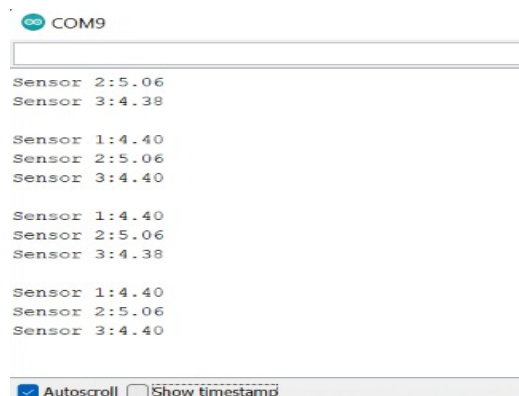
#### 4.4.3.3 Hasil Pengujian Komunikasi Serial

Pengujian komunikasi serial pengiriman data sensor dapat dilihat pada gambar dibawah ini.



Gambar 4. 41 Proses Pengujian Komunikasi Serial

Pengujian komunikasi serial NodeMCU ESP8266 terhadap pengiriman data sensor diperoleh hasil pada serial monitor sebagai berikut.



Gambar 4. 42 Hasil Pengujian Pada Serial Monitor

Kesimpulan:

Berdasarkan pengujian yang telah dilakukan terhadap komunikasi serial NodeMCUESP8266 dapat disimpulkan bahwa pengiriman data dapat dilakukan antara Arduino Mega 2560 dengan NodeMCU ESP8266 melalui komunikasi serial antar pin Tx dan Rx sehingga dapat digunakan untuk sistem *monitoring* pada proyek akhir ini.

#### **4.5 Pengujian Keseluruhan Alat**

Pengujian *prototype* sistem kontrol dan *monitoring* pada mobil listrik secara keseluruhan dilakukan untuk mengetahui performa kinerja alat saat semua sistem telah diprogramkan berdasarkan fungsinya masing-masing. Tidak hanya melakukan pengujian pada sistem kontrol tetapi juga pengujian *software* aplikasi yang telah dibuat apakah dapat mengirimkan perintah suara serta menampilkan data sensor tegangan.

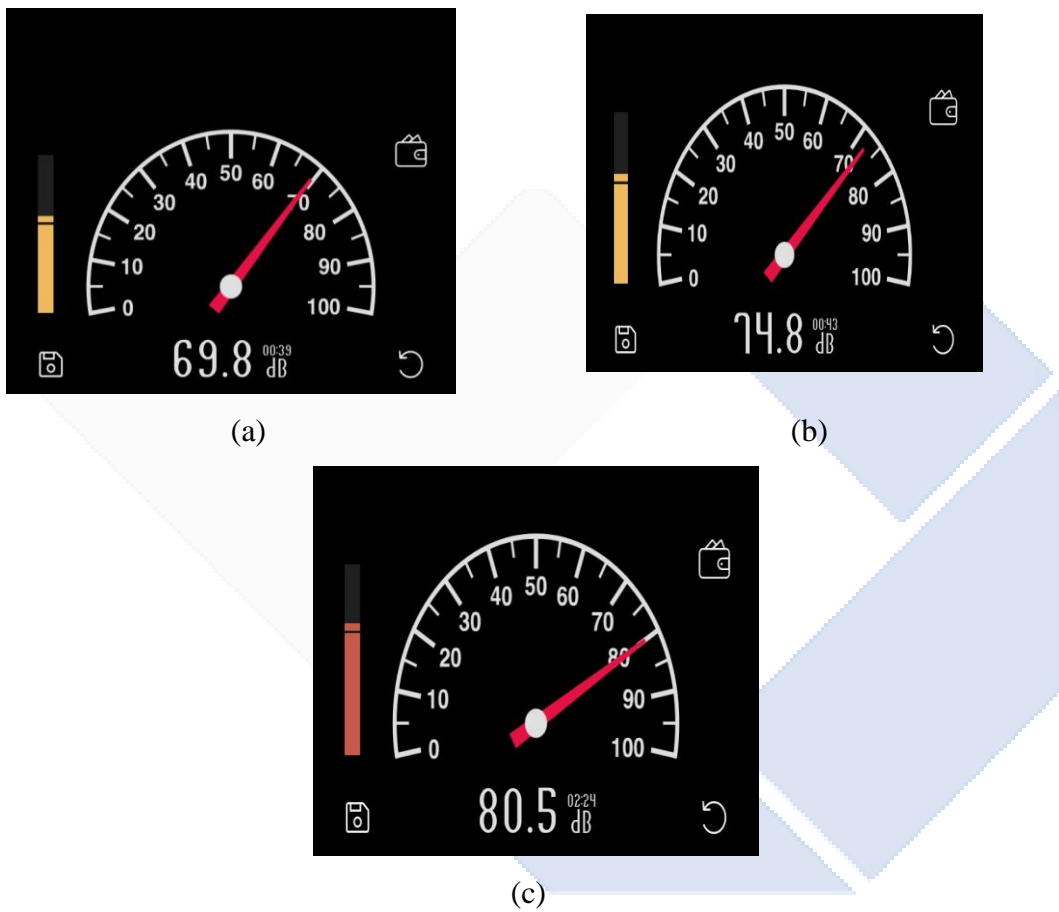
Pengujian alat dilakukan dengan cara memberikan perintah suara melalui aplikasi SKMML untuk menghidupkan dan mematikan *starter*, menurunkan dan menaikkan kaca jendela, menyalakan dan mematikan *wiper* serta pemantauan tegangan yang ada pada masing-masing motor DC pada peralatan *prototype* mobil listrik tersebut. Oleh karena itu langkah-langkah yang dilakukan dalam pengujian alat secara keseluruhan sebagai berikut.

##### **4.5.1 Pengujian pengaruh jarak sumber suara ke smartphone**

Pengujian ini berfungsi untuk mengetahui jarak optimum pemberian perintah suara ke *smartphone*. Pengujian ini dilakukan dengan memberikan perintah suara secara berkala pada jarak mulai dari 10cm – 100cm. Pemberian perintah suara dilakukan secara beragam dengan tingkat intensitas suara yang berbeda-beda. Pengukuran volume perintah suara diukur dengan menggunakan aplikasi *Sound Meter*. Intensitas suara manusia berkisar antara 70 dB sampai dengan 80 dB. Oleh karena itu penentuan sampel volume suara dilakukan berdasarkan pengukuran suara ketika percakapan dimana diperoleh 3 sampel hasil volume suara. Sampel suara 1 dengan volume suara sebesar 70 dB dimana sampel suara ini pengucapan perintah suara dengan volume rendah. Sampel suara 2 dengan volume suara sebesar 75 dB dimana sampel suara ini pengucapan perintah suara dengan volume sedang,

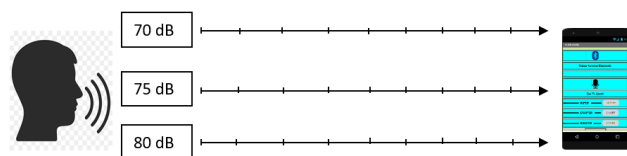
Terakhir sampel suara 3 dengan volume suara sebesar 80 dB dimana sampel suara ini pengucapan perintah suara dengan volume tinggi. Pengukuran volume perintah suara diukur dengan menggunakan aplikasi *Sound Meter* yang terdapat pada *smartphone*.

Proses pengukuran volume suara menggunakan aplikasi *Sound Meter* dapat dilihat pada gambar dibawah ini.



Gambar 4. 43 Hasil Pengukuran Sampel Suara: (a) sampel suara 1; (b) sampel suara 2; (c) sampel suara 3

Prosedur pengujian pengaruh sumber suara ke *smartphone* dapat dilihat melalui gambar dibawah ini.



Gambar 4. 44 Blok Diagram Pengiriman Perintah Suara



Pengujian pengaruh sumber suara ke *smartphone* dilakukan dengan memberikan perintah suara dengan volume suara yang berbeda berdasarkan jarak yang telah ditentukan untuk mengetahui jarak maksimum pendeteksian perintah suara oleh aplikasi SKMML pada *smartphone*. Adapun hasil pengujian pengaruh jarak sumber suara ke *smartphone* dapat dilihat pada tabel dibawah ini.

Tabel 4. 4 Hasil Pengaruh Jarak Sumber Suara ke *Smartphone*

Jarak Pengiriman Perintah (cm)	Volume Suara		
	Suara 1 (70 dB)	Suara 2 (75 dB)	Suara 3 (80 dB)
10	✓	✓	✓
20	✓	✓	✓
30	✓	✓	✓
40	✓	✓	✓
50	✓	✓	✓
60	✓	✓	✓
70	✓	✓	✓
80	✓	✓	✓
90	×	✓	✓
100	×	×	✓
Persentase Keberhasilan	80%	90%	100%

Keterangan:

Tanda ✓ = percobaan berhasil

Tanda × = percobaan gagal

Rumus perhitungan persentase keberhasilan :

$$\text{Persentase Keberhasilan} = \left| \frac{\text{Jumlah Percobaan} - \text{Jumlah Percobaan Gagal}}{\text{Jumlah Percobaan}} \right| \times 100\%$$

Rumus perhitungan *error*:

$$\text{Persentase Error} = \left| \frac{\text{Jumlah Percobaan} - \text{Jumlah Percobaan Berhasil}}{\text{Jumlah Percobaan}} \right| \times 100\%$$

- Perhitungan persentase keberhasilan dan presentase *error* pada suara 1

$$\text{Persentase Keberhasilan} = \left| \frac{10-2}{10} \right| \times 100\% = 80\%$$

$$\text{Persentase Error} = \left| \frac{10-8}{10} \right| \times 100\% = 20\%$$

- Perhitungan persentase keberhasilan dan presentase *error* pada suara 2

$$\text{Persentase Keberhasilan} = \left| \frac{10-1}{10} \right| \times 100\% = 90\%$$

$$\text{Persentase Error} = \left| \frac{10-9}{10} \right| \times 100\% = 10\%$$

- Perhitungan persentase keberhasilan dan presentase *error* pada suara 3

$$\text{Persentase Keberhasilan} = \left| \frac{10-0}{10} \right| \times 100\% = 100\%$$

$$\text{Persentase Error} = \left| \frac{10-10}{10} \right| \times 100\% = 0\%$$

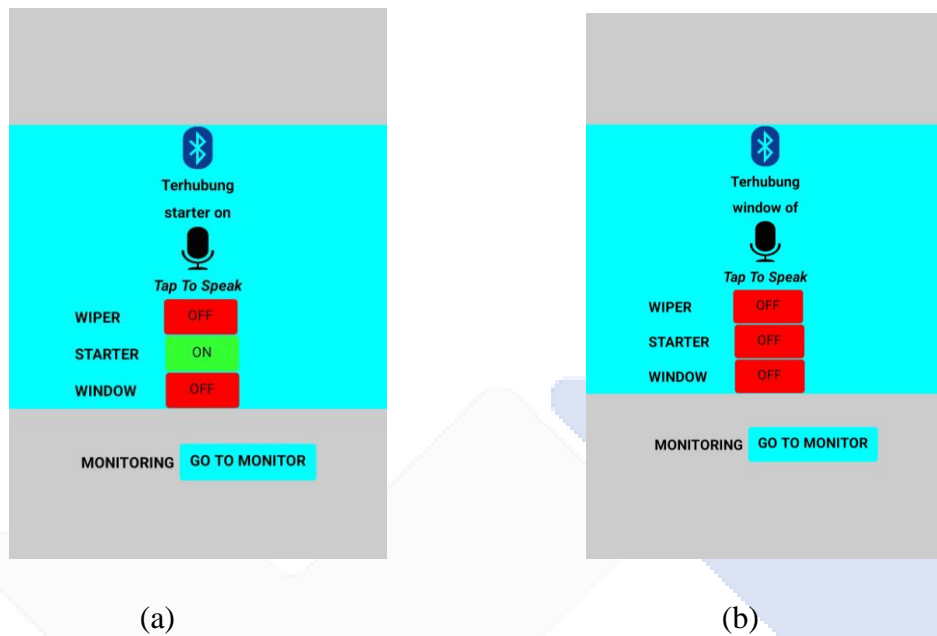
Kesimpulan:

Dari hasil pengujian yang telah dilakukan pengaruh jarak pemberian perintah suara ke *smartphone* bergantung pada besar dan kecilnya volume suara. Pada tabel hasil pengujian dapat dilihat ketika memberikan perintah suara dengan volume 70 dB jarak maksimum pemberian perintah suara 80 cm, perintah suara dengan volume 77 dB jarak maksimum pemberian perintah suara 100 cm, dan perintah suara dengan volume 83 dB jarak maksimum pemberian perintah suara 100 cm. Maka dapat disimpulkan jarak optimum pemberian perintah suara adalah 80 cm.

#### 4.5.2 Pengujian pengontrolan starter dengan perintah suara

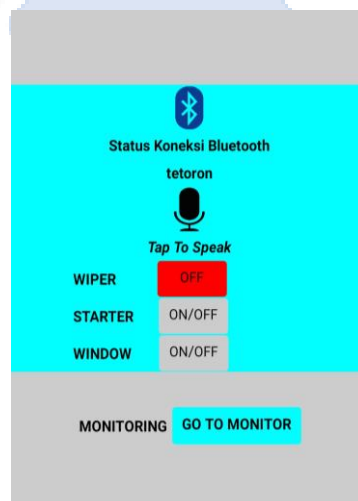
Pengujian *starter* ini dilakukan untuk mengetahui respon motor *starter* ketika diberikan perintah suara apakah bekerja sesuai rancangan. Pengujian ini dilakukan dengan mengucapkan kode perintah “Starter ON” pada *screen voice control*. Kode perintah ini digunakan untuk menyalakan speaker dan motor DC starter. *Speaker* akan mengeluarkan suara proses *starting* pada mobil sesungguhnya, hal ini berfungsi sebagai indikator sebelum motor DC berputar. Kemudian pengujian untuk mematikan motor DC starter dengan mengucapkan kode perintah “Starter OFF” pada *screen voice control*. Kode perintah ini digunakan untuk mematikan perputaran motor DC *starter*. Setelah itu dilakukan observasi kejelasan pengenalan kata dan durasi pengiriman perintah suara serta proses pelaksanaan kontrol dari perintah yang telah disebutkan.

Berikut gambar tampilan dari *screen voice control* pada aplikasi SKMML saat kondisi *starter ON* dan *starter OFF*.



Gambar 4. 45 Tampilan *Screen Voice Control*: (a) tampilan saat kondisi *starter on*; (b) tampilan saat kondisi *starter off*

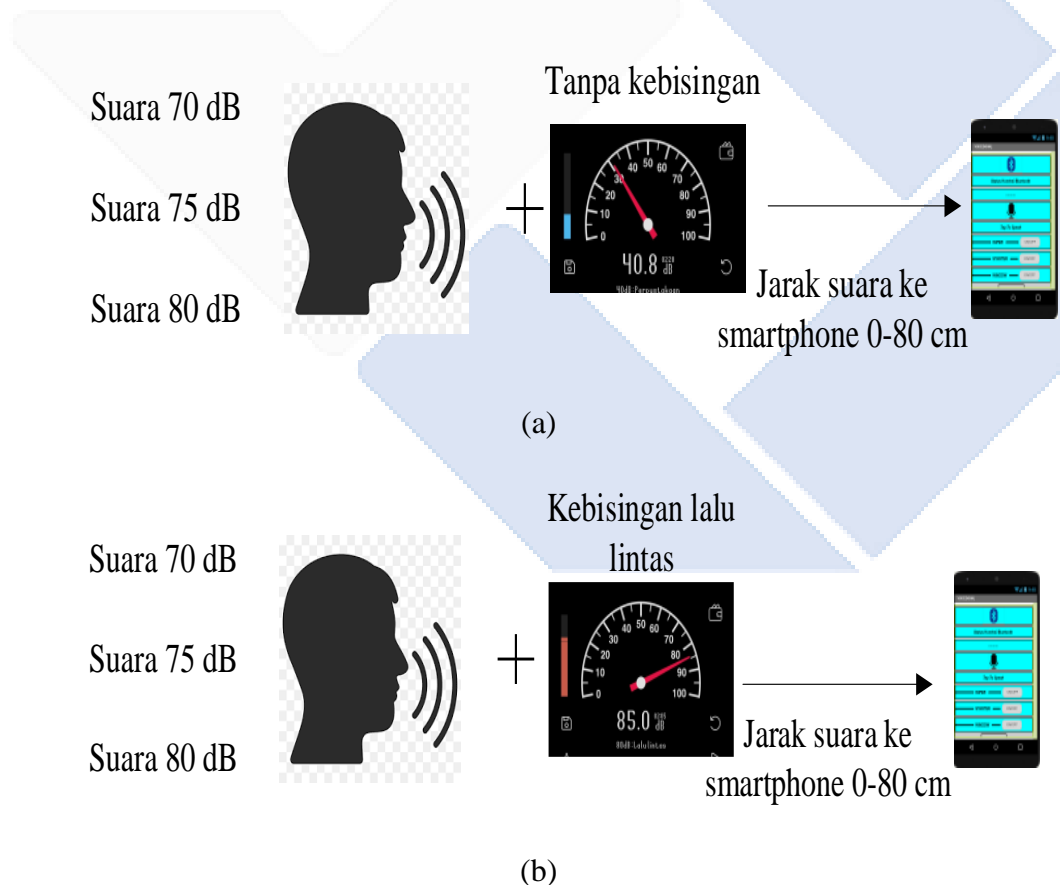
Berikut gambar tampilan aplikasi ketika gagal melakukan pendeteksian perintah suara pada pengontrolan *starter*.



Gambar 4. 46 Tampilan Aplikasi Ketika Gagal Mendeteksi Perintah Suara

Pengiriman perintah suara dalam pengujian ini menggunakan tiga sampel volume suara yaitu volume suara 70 dB (suara rendah), volume suara 75 dB (suara

sedang), dan volume suara 80 dB (suara tinggi). Pengujian dilakukan dalam dua kondisi yaitu kondisi tanpa kebisingan dan kondisi dengan kebisingan terhadap masing-masing sampel volume suara. Pengujian pada kondisi tanpa kebisingan dilakukan pada arena sunyi dengan intensitas suara sebesar 40 dB. Pengujian pada kondisi dengan kebisingan dilakukan pada arena lalu lintas dengan intensitas kebisingan sebesar 85 dB. Pengukuran intensitas suara pada kedua kondisi diukur dengan menggunakan aplikasi *Sound Meter* yang terdapat pada *smartphone*. Perhitungan persentase keberhasilan pendeteksian pengiriman perintah suara dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi terhadap kode perintah untuk mengontrol *starter* dengan jarak sumber suara pengguna terhadap *smartphone* ialah 80 cm. Prosedur pengujian pengiriman perintah suara pada masing-masing kondisi dapat dilihat pada gambar dibawah ini.



Gambar 4. 47 Pengujian Pengontrolan *Starter*: (a) kondisi tanpa kebisingan; (b) kondisi dengan kebisingan

Tabel 4. 5 Hasil Pengujian Perintah Suara Pengontrolan *Starter* dengan Volume Suara 70 dB

Pengujian	Volume Suara 70 dB (Rendah)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Starter On	100%	94%
Starter Off	100%	94%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan *starter* dilakukan dengan memberikan perintah suara dengan volume rendah (suara 70 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 94% yang dihitung dari 47 kali percobaan berhasil serta persentase kegagalan sebesar 6% yang dihitung dari 3 kali percobaan yang gagal.

Tabel 4. 6 Hasil Pengujian Perintah Suara Pengontrolan *Starter* dengan Volume Suara 75 dB

Pengujian	Volume Suara 75 dB (Sedang)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Starter On	100%	96%
Starter Off	100%	96%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan *starter* dilakukan dengan memberikan perintah suara dengan volume sedang (suara 75 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 96% yang hitung dari

48 kali percobaan berhasil serta persentase kegagalan sebesar 4% yang dihitung dari 2 kali percobaan yang gagal..

Tabel 4. 7 Hasil Pengujian Perintah Suara Pengontrolan *Starter* dengan Volume Suara 80 dB

Pengujian	Volume Suara 80 dB (Tinggi)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Starter On	100%	98%
Starter Off	100%	98%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan *starter* dilakukan dengan memberikan perintah suara dengan volume tinggi (suara 80 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 98% yang dihitung dari 49 kali percobaan berhasil serta persentase kegagalan sebesar 2% yang dihitung dari 1 kali percobaan yang gagal.

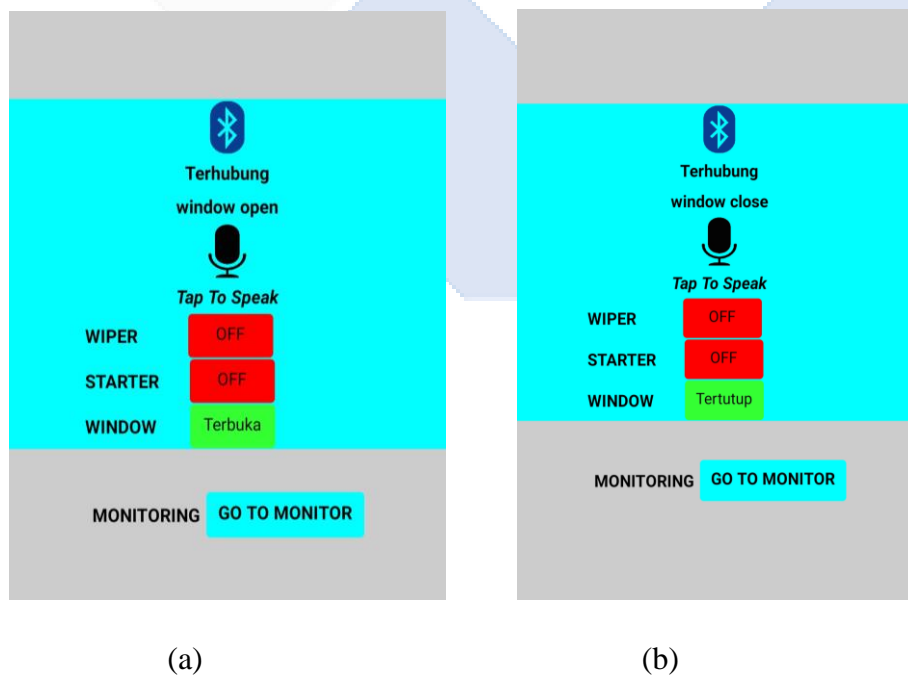
Kesimpulan:

Dari tabel pengujian diatas dapat dilihat bahwa persentase keberhasilan pengiriman perintah suara terdeteksi oleh aplikasi SKMML dipengaruhi oleh faktor kebisingan, besar kecilnya volume suara, dan pengucapan kalimat perintah yang diucapkan oleh user sehingga mempengaruhi kinerja dari *speech recognizer*. Eksekusi perintah suara dengan kode perintah “STARTER ON” menghasilkan sistem kontrol pada mobil listrik mengeluarkan bunyi proses *starting* pada mobil asli dari *speaker* dalam waktu 5 detik kemudian diikuti dengan perputaran motor DC *starter* sebagai indikator dari mesin mobil telah menyala. Kemudian eksekusi dari perintah suara dengan kode perintah “STARTER OFF” menghasilkan sistem kontrol mematikan perputaran motor DC *starter*.

### 4.5.3 Pengujian pengontrolan pergerakan naik turun kaca jendela dengan perintah suara.

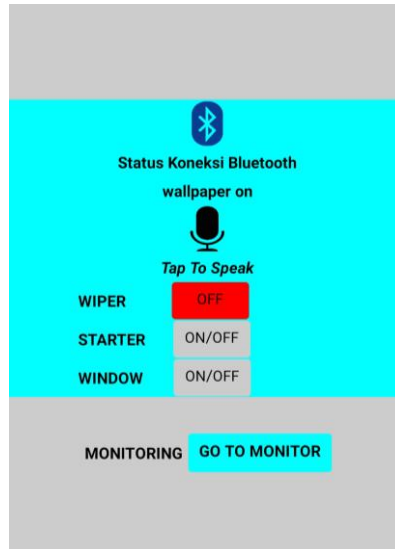
Pengujian ini dilakukan untuk mengetahui respon motor *window* ketika diberikan perintah suara apakah akan melakukan pergerakan naik atau turun dari kaca jendela pada *prototype* mobil listrik. Pengujian ini dilakukan dengan mengucapkan kode perintah “WINDOW OPEN” pada *screen voice control*. Kode perintah ini digunakan untuk menggerakkan kaca jendela ke bawah atau membuat kaca jendela dalam kondisi terbuka. Pada proses pergerakan ini motor DC akan berputar searah jarum jam. Kemudian mengucapkan kode perintah “WINDOW CLOSE” pada *screen voice control*. Kode perintah ini digunakan untuk menggerakkan kaca jendela ke atas atau membuat motor DC berputar bellawanan arah jarum jam. Setelah itu dilakukan observasi kejelasan pengenalan kata dan durasi pengiriman perintah suara serta proses pelaksanaan kontrol dari perintah yang telah disebutkan.

Berikut gambar tampilan dari *screen voice control* pada aplikasi SKMML saat kondisi *window open* dan *window close*.



Gambar 4. 48 Tampilan *Voice Control*: (a) tampilan saat kondisi kaca jendela terbuka; (b) tampilan saat kondisi kaca jendela tertutup

Berikut gambar tampilan aplikasi ketika gagal melakukan pendeteksian perintah suara pada pengontrolan kaca jendela.



Gambar 4. 49 Tampilan Aplikasi Ketika Gagal Mendeteksi Kode Perintah

Pengiriman perintah suara dalam pengujian ini menggunakan tiga sampel volume suara yaitu volume suara 70 dB (suara rendah), volume suara 75 dB (suara sedang), dan volume suara 80 dB (suara tinggi). Pengujian dilakukan dalam dua kondisi yaitu kondisi tanpa kebisingan dan kondisi dengan kebisingan terhadap masing-masing sampel volume suara. Pengujian pada kondisi tanpa kebisingan dilakukan pada arena sunyi dengan intensitas suara sebesar 40 dB. Pengujian pada kondisi dengan kebisingan dilakukan pada arena lalu lintas dengan intensitas kebisingan sebesar 85 dB. Pengukuran intensitas suara pada kedua kondisi diukur dengan menggunakan aplikasi *Sound Meter* yang terdapat pada *smartphone*. Perhitungan persentase keberhasilan pendeteksian pengiriman perintah suara dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi terhadap kode perintah untuk mengontrol pergerakan kaca jendela dengan jarak sumber suara pengguna terhadap *smartphone* ialah 80 cm. Prosedur pengujian pengiriman perintah suara pada masing-masing kondisi dapat dilihat pada gambar dibawah ini.





(a)



(b)

Gambar 4. 50 Pengujian Pengontrolan Kaca Jendela: (a) kondisi tanpa kebisingan; (b) kondisi dengan kebisingan

Tabel 4. 8 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 70 dB

Pengujian	Volume Suara 70 dB (Rendah)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Window Open	100%	94%
Window Close	100%	94%
Window Off	100%	94%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan pergerakan kaca jendela dilakukan dengan memberikan perintah suara dengan volume rendah (suara 70 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-

masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 94% yang dihitung dari 47 kali percobaan berhasil serta persentase kegagalan sebesar 6% yang dihitung dari 3 kali percobaan yang gagal.

Tabel 4. 9 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 75 dB

Pengujian	Volume Suara 75 dB (Sedang)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Window Open	100%	96%
Window Close	100%	96%
Window Off	100%	96%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan pergerakan kaca jendela dilakukan dengan memberikan perintah suara dengan volume sedang (suara 75 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 96% yang hitung dari 48 kali percobaan berhasil serta persentase kegagalan sebsar 4% yang dihitung dari 2 kali percobaan yang gagal.

Tabel 4. 10 Hasil Pengujian Perintah Suara Pengontrolan Kaca Jendela dengan Volume Suara 80 dB

Pengujian	Volume Suara 80 dB (Tinggi)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Window Open	100%	98%
Window Close	100%	98%
Window Off	100%	98%

Keterangan:

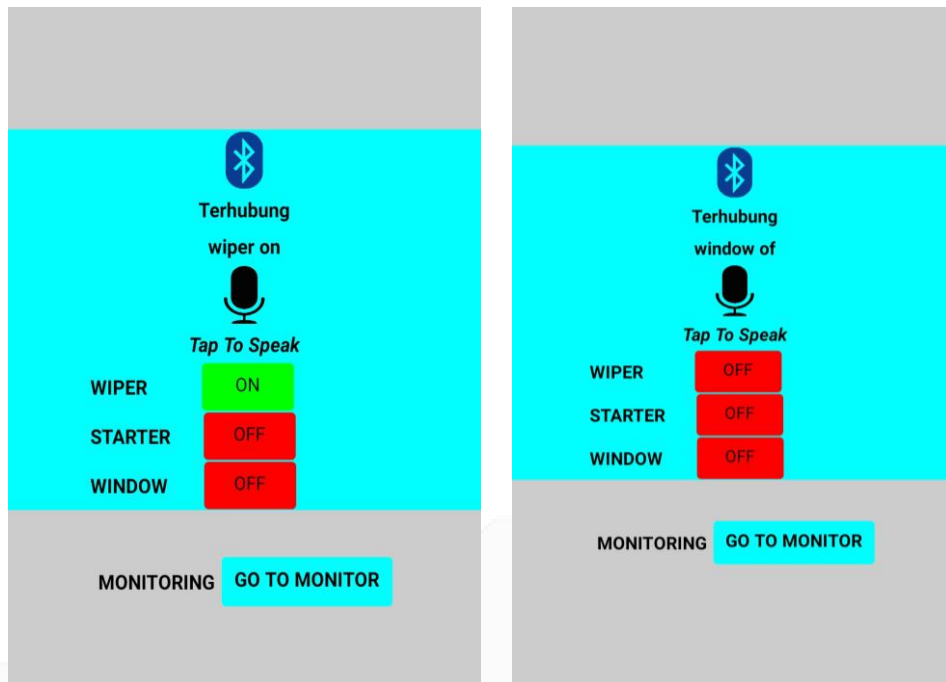
Pengujian pengiriman perintah suara untuk pengontrolan pergerakan kaca jendela dilakukan dengan memberikan perintah suara dengan volume tinggi (suara 80 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 98% yang dihitung dari 49 kali percobaan berhasil serta persentase kegagalan sebesar 2% yang dihitung dari 1 kali percobaan yang gagal.

Kesimpulan:

Dari tabel pengujian diatas dapat dilihat bahwa persentase keberhasilan pengiriman perintah suara terdeteksi oleh aplikasi SKMML dipengaruhi oleh faktor kebisingan, besar kecilnya volume suara, dan pengucapan kalimat perintah yang diucapkan oleh user sehingga mempengaruhi kinerja dari *speech recognizer*. Eksekusi perintah suara dengan kode perintah “WINDOW OPEN” menghasilkan sistem kontrol pada mobil listrik menyebabkan pergerakan turun dari kaca jendela pada mobil asli. Kemudian eksekusi dari perintah suara dengan kode perintah “WINDOW CLOSE” menghasilkan pergerakan naik dari kaca jendela pada mobil listrik. Selanjutnya eksekusi dari perintah suara dengan kode perintah “WINDOW OFF” menghasilkan motor dc *window* akan berhenti.

#### **4.5.4 Pengujian pengontrolan pergerakan wiper menggunakan perintah suara.**

Pengujian ini dilakukan untuk mengetahui respon motor *wiper* ketika diberikan perintah suara apakah akan melakukan pergerakan *wiper*. Pengujian ini dilakukan dengan mengucapkan kode perintah “WIPER ON” pada *screen voice control*. Kode perintah ini digunakan untuk menyalakan motor. Pada proses pergerakan ini motor DC akan berputar untuk menggerakkan *wiper* pada *protoype* mobil listrik, Kemudian mengucapkan kode perintah “WIPER OFF” pada *screen voice control*. Kode perintah ini digunakan untuk mematikan putaran motor DC. Setelah itu dilakukan observasi kejelasan pengenalan kata dan durasi pengiriman perintah suara serta proses pelaksanaan kontrol dari perintah yang telah disebutkan.

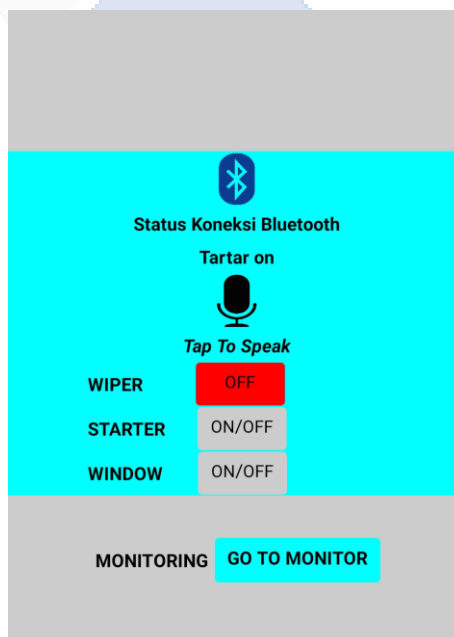


(a)

(b)

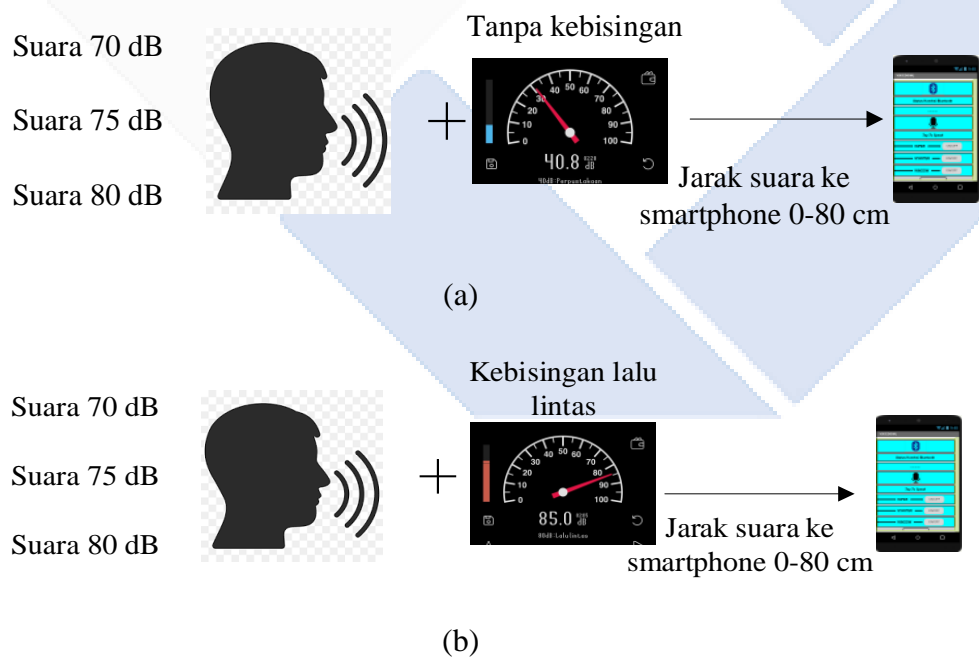
Gambar 4. 51 Tampilan *Voice Control*: (a) tampilan saat kondisi wiper on; (b) tampilan saat kondisi wiper off

Berikut gambar tampilan aplikasi ketika gagal melakukan pendeteksian perintah suara pada pengontrolan *wiper*.



Gambar 4. 52 Tampilan Aplikasi Ketika Gagal Melakukan Pendeteksian Perintah Suara

Pengiriman perintah suara dalam pengujian ini menggunakan tiga sampel volume suara yaitu volume suara 70 dB (suara rendah), volume suara 75 dB (suara sedang), dan volume suara 80 dB (suara tinggi). Pengujian dilakukan dalam dua kondisi yaitu kondisi tanpa kebisingan dan kondisi dengan kebisingan terhadap masing-masing sampel volume suara. Pengujian pada kondisi tanpa kebisingan dilakukan pada arena sunyi dengan intensitas suara sebesar 40 dB. Pengujian pada kondisi dengan kebisingan dilakukan pada arena lalu lintas dengan intensitas kebisingan sebesar 85 dB. Pengukuran intensitas suara pada kedua kondisi diukur dengan menggunakan aplikasi *Sound Meter* yang terdapat pada *smartphone*. Perhitungan persentase keberhasilan pendeteksian pengiriman perintah suara dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi terhadap kode perintah untuk mengontrol *wiper* dengan jarak sumber suara pengguna terhadap *smartphone* ialah 80 cm. Prosedur pengujian pengiriman perintah suara pada masing-masing kondisi dapat dilihat pada gambar dibawah ini.



Gambar 4. 53 Pengujian Pengontrolan *Wiper*: (a) kondisi tanpa kebisingan; (b) kondisi dengan kebisingan

Tabel 4. 11 Hasil Pengujian Perintah Suara Pengontrolan *Wiper* dengan Volume Suara 70 dB

Pengujian	Volume Suara 70 dB (Rendah)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Wiper On	100%	94%
Wiper Off	100%	94%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan pergerakan *wiper* dilakukan dengan memberikan perintah suara dengan volume rendah (suara 70 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 94% yang dihitung dari 47 kali percobaan berhasil serta persentase kegagalan sebesar 6% yang dihitung dari 3 kali percobaan yang gagal.

Tabel 4. 12 Hasil Pengujian Perintah Suara Pengontrolan *Wiper* dengan Volume Suara 75 dB

Pengujian	Volume Suara 75 dB (Sedang)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Wiper On	100%	96%
Wiper Off	100%	96%

Keterangan:

Pengujian pengiriman perintah suara untuk pengontrolan pergerakan *wiper* dilakukan dengan memberikan perintah suara dengan volume sedang (suara 75 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 96%

yang hitung dari 48 kali percobaan berhasil serta persentase kegagalan sebesar 4% yang dihitung dari 2 kali percobaan yang gagal..

Tabel 4. 13 Hasil Pengujian Perintah Suara Pengontrolan *Wiper* dengan Volume Suara 80 dB

Pengujian	Volume Suara 80 dB (Tinggi)	
	Persentase Keberhasilan(%)	
	Tanpa Kebisingan (40 dB)	Dengan Kebisingan (85 dB)
Wiper On	100%	98%
Wiper Off	100%	98%

**Keterangan:**

Pengujian pengiriman perintah suara untuk pengontrolan pergerakan *wiper* dilakukan dengan memberikan perintah suara dengan volume tinggi (suara 75 dB) pada dua kondisi. Pengujian ini dilakukan sebanyak 50 kali percobaan pada masing-masing kondisi. Pada kondisi tanpa kebisingan persentase keberhasilan sebesar 100% dan pada kondisi dengan kebisingan persentase keberhasilan sebesar 98% yang dihitung dari 49 kali percobaan berhasil serta persentase kegagalan sebesar 2% yang dihitung dari 1 kali percobaan yang gagal.

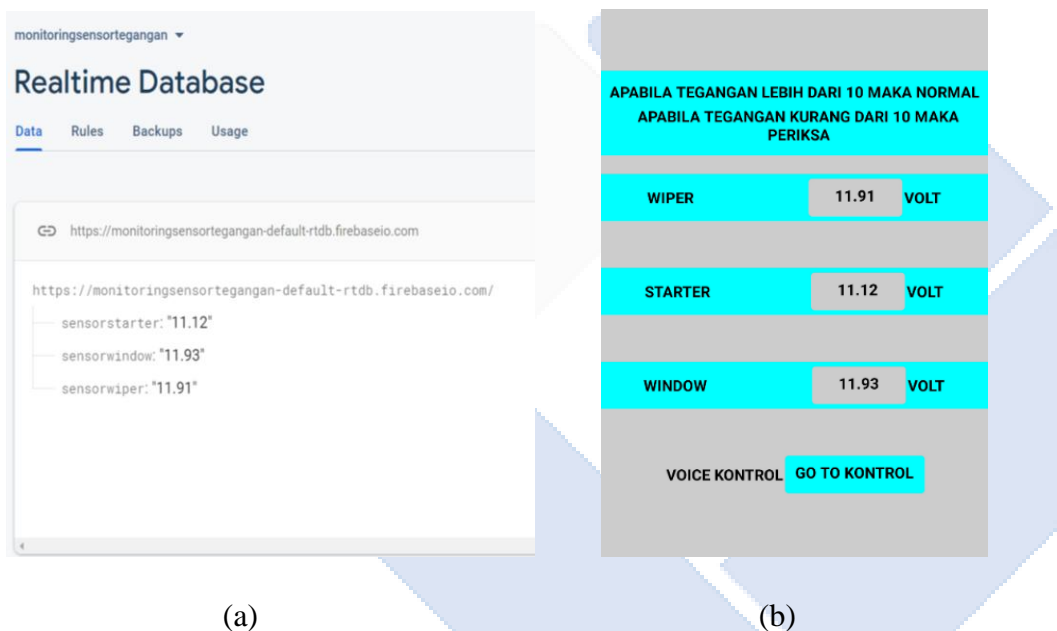
**Kesimpulan:**

Dari tabel pengujian diatas dapat dilihat bahwa persentase keberhasilan pengiriman perintah suara terdeteksi oleh aplikasi SKMML dipengaruhi oleh faktor kebisingan, besar kecilnya volume suara, dan pengucapan kalimat perintah yang diucapkan oleh user sehingga mempengaruhi kinerja dari *speech recognizer*. Eksekusi perintah suara dengan kode perintah “WIPER ON” menghasilkan sistem kontrol pada mobil listrik menghidupkan motor DC kemudian motor DC menggerakkan konstruksi wiper. Selanjutnya eksekusi dari perintah suara dengan kode perintah “WIPER OFF” menghasilkan sistem kontrol mematikan perputaran motor DC wiper.

#### 4.5.5 Pengujian monitoring data sensor tegangan pada aplikasi SKMML

Pengujian ini berfungsi untuk menganalisa koneksi *firebase* dengan aplikasi SKMML yang sudah terinstal pada *smartphone* pengguna apakah sistem *monitoring* berbasis IoT telah berfungsi dengan baik. Pengujian ini dilakukan dengan membandingkan data sensor yang tampil pada *real time database firebase* sama dengan yang tampil pada *screen monitoring* perangkat pada aplikasi SKMML.

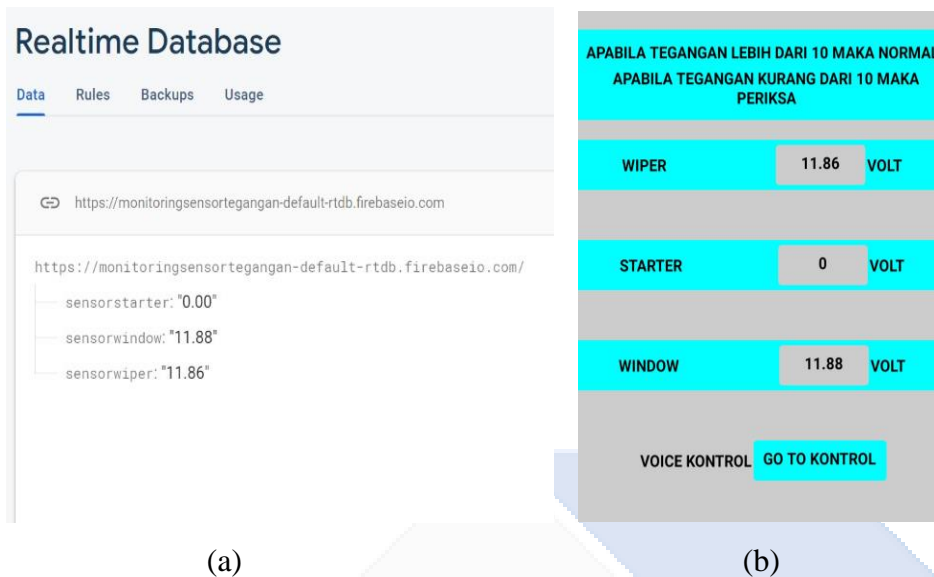
- Pengujian monitoring data sensor tegangan ketika semua peralatan berfungsi dengan baik



Gambar 4. 54 Hasil Pengujian *Monitoring* Sensor Tegangan: (a) Tampilan data sensor pada *firebase*; (b) Tampilan data sensor pada *screen monitoring*



- Pengujian *monitoring* data sensor tegangan ketika kondisi *starter* rusak.



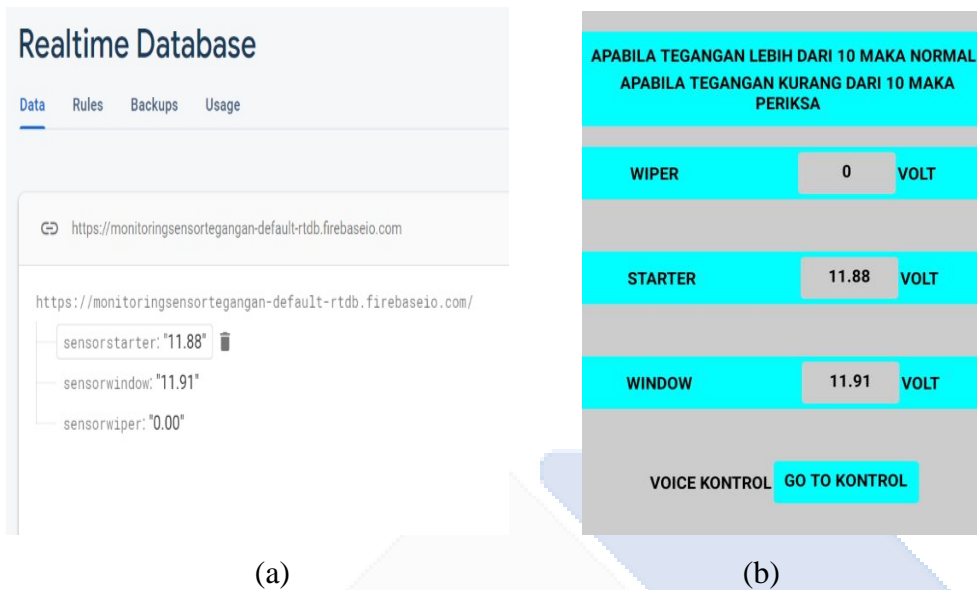
Gambar 4. 55 Hasil Pengujian *Monitoring* Sensor Tegangan Ketika *Starter* Rusak: (a) Data Sensor Tegangan Pada *Firebase*; (b) Tampilan *Screen Monitoring*

- Pengujian *monitoring* data sensor tegangan ketika kondisi *window* rusak.



Gambar 4. 56 Hasil Pengujian *Monitoring* Data Sensor Tegangan Ketika *Window* Rusak: (a) Data sensor tegangan pada *firebase*; (b) Tampilan *Screen Monitoring*

- Pengujian monitoring data sensor tegangan ketika kondisi *wiper* rusak.



Gambar 4. 57 Hasil Pengujian *Monitoring* Data Sensor Tegangan Ketika Kondisi *Wiper* Rusak: (a) Data sensor tegangan pada *firebase*; (b) Tampilan *screen monitoring*

Tabel 4. 14 Hasil Pengujian Sistem *Monitoring* Data Sensor Tegangan

Kondisi	Pengujian Sistem Monitoring Data Sensor Tegangan					Persentase Keberhasilan
	1	2	3	4	5	
Semua Baik	✓	✓	✓	✓	✓	100%
Starter Rusak	✓	✓	✓	✓	✓	100%
Window Rusak	✓	✓	✓	✓	✓	100%
Wiper Rusak	✓	✓	✓	✓	✓	100%

Keterangan:

Tanda ✓ = percobaan berhasil

Tanda ✗ = percobaan gagal

Rumus perhitungan persentase keberhasilan:

$$\text{Persentase Keberhasilan} = \left| \frac{\text{Jumlah Percobaan} - \text{Jumlah Percobaan Gagal}}{\text{Jumlah Percobaan}} \right| \times 100\%$$

Rumus perhitungan persentase *error*:

$$\text{Persentase Error} = \left| \frac{\text{Jumlah Percobaan} - \text{Jumlah Percobaan Berhasil}}{\text{Jumlah Percobaan}} \right| \times 100\%$$

- Perhitungan persentase keberhasilan pada tabel

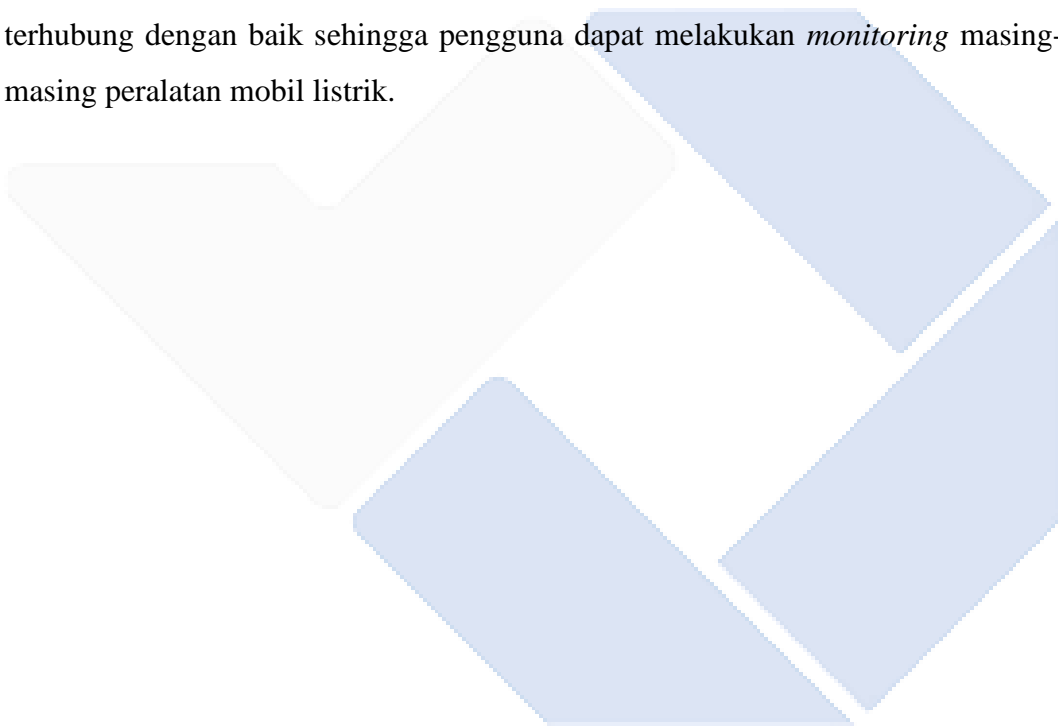
$$\text{Persentase Keberhasilan} = \left| \frac{5-0}{5} \right| \times 100\% = 100\%$$

- Perhitungan persentase *error* pada tabel

$$\text{Persentase Error} = \left| \frac{5-5}{5} \right| \times 100\% = 0\%$$

Kesimpulan:

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa hasil monitoring yang tampil pada *screen monitoring* aplikasi SKMML sama dengan *realtime database* pada *platform firebase*. Hal ini menunjukkan bahwa koneksi antara Arduino Mega 2560, NodeMCU ESP8266, *firebase*, dan aplikasi SKMML terhubung dengan baik sehingga pengguna dapat melakukan *monitoring* masing-masing peralatan mobil listrik.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari hasil pengujian yang telah dilakukan dengan memberikan perintah suara dan *monitoring* kondisi dari masing-masing peralatan mobil listrik melalui aplikasi SKMML diperoleh kesimpulan sebagai berikut.

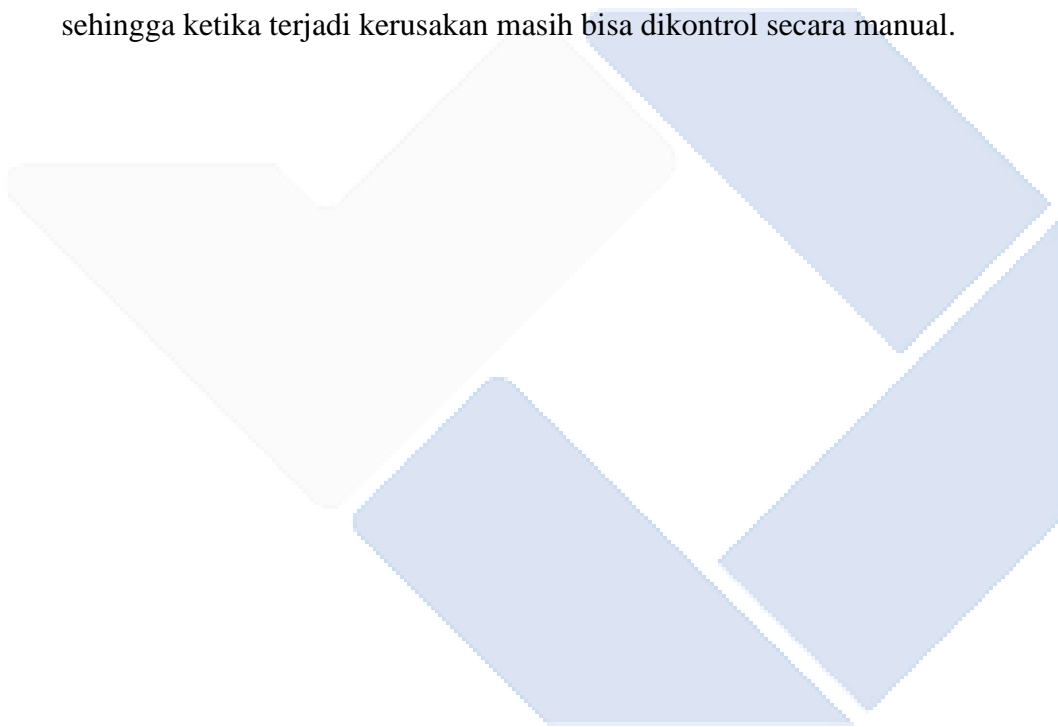
1. Pemberian perintah suara untuk mengontrol *starter*, *wiper*, dan kaca jendela pada mobil listrik dilakukan melalui aplikasi SKMML yang telah dibuat. Pengucapan dalam pemberian perintah suara disarankan pada volume 70 dB – 80 dB dengan artikulasi yang benar serta jarak optimal pemberian perintah suara ke *smartphone* ialah 80 cm sehingga *speech recognizer* pada aplikasi SKMML dapat mendeteksi perintah tersebut.
2. Alat mampu menyalakan sistem kontrol melalui perintah suara dengan persentase keberhasilan 100% dari 50 kali percobaan pada kondisi tanpa kebisingan dan persentase keberhasilan 96% pada kondisi dengan kebisingan.
3. Durasi pengiriman data perintah suara melalui Modul Bluetooth HC-05 ke Arduino Mega 2560 selama 4 detik, hal ini dikarenakan terdapat proses mengubah perintah suara menjadi sebuah teks dan pencocokan data tersebut pada kode-kode perintah yang telah dibuat.
4. Pengiriman data sensor tegangan untuk *screen monitoring* rata-rata mengalami *delay* sebesar 4 detik, hal ini dikarenakan terjadi dua kali komunikasi yaitu komunikasi serial antara Arduino Mega 2560 dengan NodeMCU ESP8266 serta komunikasi IoT antara NodeMCU ESP8266 dengan *Firebase*.

#### **5.2 Saran**

Proyek akhir yang dibuat ini masih terdapat beberapa kekurangan yang bisa dijadikan sebagai saran untuk pengembangan proyek akhir selanjutnya yaitu:

1. Proyek akhir ini harus bisa mengontrol peralatan mobil listrik dengan perintah suara dengan melebihi persentase 96% ketika kondisi dengan kebisingan.

2. Pengiriman data perintah suara pada aplikasi SKMML melalui modul Bluetooth HC-05 mengalami *delay* selama 4 detik karena terdapat proses pengenalan perintah suara. Untuk kedepannya diharapkan menemukan program pengenalan perintah suara dan pengiriman data yang lebih cepat sehingga setiap perintah suara bisa langsung dieksekusi dengan durasi yang lebih cepat.
3. Sistem kontrol yang dibuat pada proyek akhir ini buat secara otomatis sehingga jika sistem kontrol rusak maka tidak mudah untuk diperbaiki, oleh karena itu untuk pengembangan proyek akhir ini menambah sistem kontrol manual sehingga ketika terjadi kerusakan masih bisa dikontrol secara manual.



## DAFTAR PUSTAKA

- [1] R. Mulyadi, D. K. Artika and M. Khalil, "Perancangan Sistem Kelistrikan Perangkat Elektronik Pada Mobil Listrik," *Elemen*, pp. 7-12, June 2019.
- [2] M. Aziz, Y. Marcelliono, I. A. Rizki, S. A. Ikhwanud and J. W. Simatupang, "Studi Analisis Perkembangan Teknologi dan Dukungan Pemerintah Terkait Mobil Listrik," vol. 22, pp. 45-55, 2020.
- [3] F. Pratama, "Car Voice Menggunakan Smartphone dengan Sensor Bluetooth sebagai Powerstart dan Powerstop pada Mobil Listrik," Politeknik Negeri Sriwijaya, Palembang, 2019.
- [4] R. A. Subekti, H. Sudibyo, V. Susanti, H. M. Saputra and A. Hartanto, *Peluang dan Tantangan Pengembangan Mobil Listrik Nasional*, 1 ed., Jakarta: LIPI Press, 2014.
- [5] A. C. Ashar, M. and A. Sifaunajah, "Penerapan Game Rdukasi "Speak English" Pada Sekolah Dasar Menggunakan Teknologi Speech Recognition," *Sains dan Teknologi*, vol. 11, p. 10, 2019.
- [6] B. A. Faruq, H. R. Herlianto and S. P. H. Simbolon, "Speech Recognition," 2019. [Online]. Available: <https://mti.binus.ac.id/2019/05/08/speech-recognition/>. [Accessed 20 April 2022].
- [7] K. U. Ariawan, G. S. Santyadiputra and I. W. Sutaya, "Design of Hexapod Robot Movement Based on Arduino Mega 2560," *Physics:Conference Series*, pp. 1-2, 2019.
- [8] A. Razor, "Arduino Mega 2560: Pengertian, Harga, dan Spesifikasi," 2021. [Online]. Available: <https://www.aldyrazor.com/2020/05/arduino-mega-2560-adalah.html>. [Accessed 19 April 2022].

- [9] R. Toyib and C. Saputra, "Prototype Robot Lengan dengan Kontrol Jarak Jauh Menggunakan Bluetooth HC-05 dan Kamera," *Pseudocode*, vol. VIII, p. 13, 2021.
- [10] D. Elektro, "Pengenalan Modul Bluetooth HC-05," *Dunia Elektro*, 19 January 2020. [Online]. Available: <https://www.sekolahotomasi.com/2020/01/pengenalan-modul-bluetooth-hc-05.html>. [Accessed 20 April 2022].
- [11] N. H. L. Dewi, M. F. Rohmah and S. Zahara, "Prototype Smarth Home dengan Modul NodeMCU ESP8266 Berbasis Internet Of Things (IoT)," p. 3, 2019.
- [12] M. Y. Efendi and J. E. Chandra, "Implementasi Internet of Things Pada Sistem Kendali Lampu Rumah Menggunakan Telegram Messenger Bot dan NodeMCU ESP8266," *A Hardware & Computation*, vol. XIX, no. 1, p. 16, 2019.
- [13] R. Muhardian and Krismadinata, "Kendali Kecepatan Motor DC dengan Kontroller PID dan Antarmuka Visual Basic," *Teknik Elektro dan Vokasional*, vol. 06, p. 329, 2020.
- [14] Badi, "Motor DC : Pengertian, Fungsi, Prinsip Kerja, Jenis Bagian," *The City Foundry*, 21 January 2022. [Online]. Available: <https://thecityfoundry.com/motor-dc/>. [Accessed 22 April 2022].
- [15] A. Alfstudio, "L298N Motor Driver," *Teknik Elektro*, 23 August 2021. [Online]. Available: <https://www.teknikelektro.com/2021/08/l298n-motor-driver.html>. [Accessed 22 April 2022].
- [16] I. Rudiatmadja, "Rancang Bangun dan Monitoring Charger Baterai dengan Metode Charging Otomasi Menggunakan Rangkaian Sensor Tegangan dan Regulator Arus Berbasis Arduino Mega 2568," *PhD Thesis Undip*, 2018.

- [17] A. Imron, T. Andromeda and B. Setiyono, "Perancangan Akuisisi Data Pada Panel RTU PT.PLN (Persero) Berplatform Android," *Information Technology and Electronics Engineering*, vol. 7, p. 2, 2018.
- [18] A. Muchta, "Fungsi dan Pengertian Aki (Baterai) Pada Kendaraan," *Auto Expose*, 26 February 2018. [Online]. Available: <https://www.autoexpose.org/2018/02/fungsi-dan-pengertian-aki-baterai.html>. [Accessed 17 April 2022].
- [19] E. "Mengenal Perangkat Lunak Arduino IDE," *Kmtech*, 8 October 2021. [Online]. Available: <https://www.autoexpose.org/2018/02/fungsi-dan-pengertian-aki-baterai.html>. [Accessed 20 April 2022].
- [20] K. S. Salamah, T. M. Kadarina and Z. Iklima, "Pengenalan MIT App INventor Untuk Siswa/i di Wilayah Kembang Utara," *Abdi Masyarakat (JAM)*, vol. 5, pp. 5-9, 2020.
- [21] D. Intern, "Apa itu Firebase? Pengertian, Jenis-jenis, dan Fungsi Kegunaannya?," 25 November 2020. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>. [Accessed 16 April 2022].
- [22] Myspslolution, "Cara Kerja Konsep Internet of Things," 25 March 2019. [Online]. Available: <http://www.myspsolution.com/news-events/cara-kerja-konsep-internet-of-things/>. [Accessed 22 April 2022].
- [23] S. Kumar, T. Prayag and M. Zymbler, "Internet of Things is a revolutionary," *Big Data*, pp. 2-21, 2019.





**LAMPIRAN 1**  
**(Riwayat Hidup Perorangan)**

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Fajri  
Tempat, Tanggal Lahir : Pangkalpinang, 30 Oktober 2000  
Alamat Rumah : Jl. Kutilang 1 Air Salemba  
No. HP : 081339323695  
Email : fajrisebelas@gmail.com  
Jenis Kelamin : Laki-laki  
Agama : Islam



### 2. Riwayat Pendidikan

- |   |               |
|---|---------------|
| 1. SD Negeri 10 Pangkalpinang                   | Lulus 2013    |
| 2. SMP Negeri 3 Pangkalpinang                   | Lulus 2016    |
| 3. SMK Negeri 2 Pangkalpinang                   | Lulus 2019    |
| 4. Politeknik Manufaktur Negeri Bangka Belitung | 2019-sekarang |

### 3. Pengalaman Kerja

Praktik Kerja Lapangan di Divisi Electrical Engineering PT IOT Integrasi Otomasi

Sungailiat, 12 Agustus 2022

Fajri

## DAFTAR RIWAYAT HIDUP

### 1. Data Pribadi

Nama Lengkap : Rizkia Meilani  
Tempat, Tanggal Lahir : Belinyu, 26 Mei 2001  
Alamat Rumah : Jl. Pahlawan 12, Belinyu, Bangka  
No. HP : 081271051086  
Email : rzmeilani26@gmail.com  
Jenis Kelamin : Perempuan  
Agama : Islam



### 2. Riwayat Pendidikan

5. SD Negeri 2 Belinyu Lulus 2013  
6. SMP Negeri 1 Belinyu Lulus 2016  
7. SMA Negeri 1 Belinyu Lulus 2019  
8. Politeknik Manufaktur Negeri Bangka Belitung 2019-sekarang

### 3. Pengalaman Kerja

Praktik Kerja Lapangan di Divisi Pengadaan PT Timah Tbk

Sungailiat, 12 Agustus 2022

Rizkia Meilani



# **LAMPIRAN 2**

## Program Arduino

```
#include <SoftwareSerial.h>
#include <DFPlayer_Mini_Mp3.h> //memanggil library DFPlayer mini
#include <Wire.h>

//BT(19, 18); //pin RX dan TX
//NODEMCU(0, 1); //pin RX dan TX
//DFPLAYER(17, 16); //pin RX dan TX

#define ENA 2 // deklarasi pin ENA
#define IN1 3 // deklarasi pin IN1
#define IN2 4 // deklarasi pin IN2
#define ENB 7 // deklarasi pin ENB
#define IN3 5 // deklarasi pin IN3
#define IN4 6 // deklarasi pin IN4
#define ENC 8 // deklarasi pin ENC
#define IN5 9 // deklarasi pin IN5
#define IN6 10 // deklarasi pin IN6

String voice;
float sensor_1 = A0; //pin yg di gunakan
float sensor_2 = A1;
float sensor_3 = A2;

float sensorValue_1 = 0;
float sensorValue_2 = 0;
float sensorValue_3 = 0;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  Serial2.begin(9600); //baud komunikasi pada 9600

  mp3_set_serial (Serial2);
  delay(500);
  mp3_set_volume(20);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN5, OUTPUT);
  pinMode(IN6, OUTPUT);
  pinMode(ENC, OUTPUT);
}

void loop ()
{
```

```

//baca permintaan
String minta = "";
//baca permintaan
while (Serial.available() > 0)
{
  minta += char(Serial.read());
}

//buang spasi data
minta.trim();
//uji
if (minta == "Ya")
{
  kirimdata();
}
minta = "";
delay(1000);

////////////////////////////////////
while (Serial1.available()) {
  delay (10);
  char c = Serial1.read ();
  if (c == '#') {
    break;
  }
  voice += c;
}

if (voice.length() > 0) {
  Serial.println(voice);

  if (voice == "wiper on" ) {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    analogWrite(ENA, 55);
    delay(2000); // Jeda 2 detik
  }
  if (voice == "starter on" ) {
    mp3_play (1); //memainkan lagu 1 pada folder mp3 yang sudah
    direname dengan nama 0001.mp3
    delay (5000); //jeda 5 detik
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENB, 255);
    delay(2000); // Jeda 2 detik
  }
  if (voice == "window open" ) {
    digitalWrite(IN5, HIGH);
    digitalWrite(IN6, LOW);
    analogWrite(ENC, 23);
    delay(2000); // Jeda 2 detik
  }
  else if (voice == "wiper off" ) {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
  }
}

```

```

        analogWrite(ENA, 0);
        delay(2000);
    }
    else if (voice == "starter of" ) {
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
        analogWrite(ENB, 0);
        delay(2000);
    }
    else if (voice == "window close" ) {
        digitalWrite(IN5, LOW);
        digitalWrite(IN6, HIGH);
        analogWrite(ENC, 23);
        delay(2000);
    }
    else if (voice == "window of" ) {
        digitalWrite(IN5, LOW);
        digitalWrite(IN6, LOW);
        analogWrite(ENC, 0);
        delay(2000);
    }
}
voice = "";
}
}
void kirimdata()
{
    sensorValue_1 = ((analogRead(sensor_1) * 0.00489) * 5);
    sensorValue_2 = ((analogRead(sensor_2) * 0.00489) * 5);
    sensorValue_3 = ((analogRead(sensor_3) * 0.00489) * 5);

    //    Serial.print("Sensor 1: ");
    //    Serial.println(sensorValue_1);
    //
    //    Serial.print("Sensor 2: ");
    //    Serial.println(sensorValue_2);
    //
    //    Serial.print("Sensor 3: ");
    //    Serial.println(sensorValue_3);

    String datakirim = String(sensorValue_1) + "#" +
String(sensorValue_2) + "#" + String(sensorValue_3);
    //kirim data
    Serial.println(datakirim);
}

```

# **LAMPIRAN 3**





## Program NodeMCU ESP8266

```
#include <SoftwareSerial.h>
SoftwareSerial DataSerial(12,14); //D6TX dan D7RX d5rx
#include <FirebaseArduino.h>
#include<ESP8266WiFi.h>
#define FIREBASE_HOST "monitoringsensortegangan-default-
rtdb.firebaseio.com"
#define FIREBASE_AUTH "xDBYKmFkMrU2xLrU2z6FpMjwFdWmSyXpQ3bJsvj7"
#define WIFI_SSID "Kos Nol Dua"
#define WIFI_PASSWORD "satutambahsatu"

unsigned long previousMillis =0;
const long interval = 3000;

//Variable array data parsing
String arrData[3];

void setup() {
  Serial.begin(115200);
  DataSerial.begin(9600);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= interval)
  {
    previousMillis = currentMillis;

    String data ="";
    while(DataSerial.available(>0)
    {
      data += char(DataSerial.read());
    }
    data.trim();

    if(data != "")
    {
      //parsing data
      int index = 0;
      for(int i=0; i<=data.length(); i++)
      {
        char delimiter ='#';
        if (data[i] != delimiter)
```

```
        arrData[index] += data[i];
    else
        index++;
}
//pastikan lengkap
if (index == 2)
{
    //tampilkan ke monitor
    Serial.println("Sensor 1:" + arrData[0]); //Sensor 1
    Serial.println("Sensor 2:" + arrData[1]); //Sensor 2
    Serial.println("Sensor 3:" + arrData[2]); //Sensor 3
    Serial.println ();
    Firebase.setString("sensorwiper", (arrData[0]));
    Firebase.setString("sensorstarter", (arrData[1]));
    Firebase.setString("sensorwindow", (arrData[2]));

}
arrData[0] = "";
arrData[1] = "";
arrData[2] = "";
}
//minta data ke arduino
DataSerial.println("Ya");
}
}
```