

**SISTEM KENDALI KURSI RODA MENGGUNAKAN  
ALGORITMA *FINITE STATE AUTOMATA***

**SKRIPSI**

Oleh :  
**DANENDRA FARREL BHAGAWANTA INDARU**  
**NIM. 210605110057**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

**SISTEM KENDALI KURSI RODA MENGGUNAKAN  
ALGORITMA *FINITE STATE AUTOMATA***

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :  
**DANENDRA FARREL BHAGAWANTA INDARU**  
**NIM. 210605110057**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2025**

## HALAMAN PERSETUJUAN

### SISTEM KENDALI KURSI RODA MENGGUNAKAN ALGORITMA *FINITE STATE AUTOMATA*

## SKRIPSI

Oleh :

**DANENDRA FARREL BHAGAWANTA INDARU**  
**NIM. 210605110057**

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 19 Oktober 2025

Pembimbing I,



Ajib Hanani, M.T  
NIP. 19840731 202321 1 103

Pembimbing II,



Dr. Agung Teguh Wibowo Almais, M.T  
NIP. 19860301 202321 1 016

Mengetahui,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom  
NIP. 19841010 201903 1 012

## HALAMAN PENGESAHAN

### SISTEM KENDALI KURSI RODA MENGGUNAKAN ALGORITMA *FINITE STATE AUTOMATA*

#### SKRIPSI

Oleh :

DANENDRA FARREL BHAGAWANTA INDARU  
NIM. 210605110057

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )

Tanggal: 6 November 2025

#### Susunan Dewan Penguji

Ketua Penguji : Dr. Ir. Yunifa Miftachul Arif, M.T  
NIP. 19830616 201101 1 004

Anggota Penguji I : Nur Fitriyah Ayu Tunjung Sari, M.Cs  
NIP. 19911226 202012 2 001

Anggota Penguji II : Ajib Hanani, M.T  
NIP. 19840731 202321 1 013

Anggota Penguji III : Dr. Agung Teguh Wibowo Almais, M.T  
NIP. 19860301 202321 1 016

Mengetahui dan Mengesahkan,  
Ketua Program Studi Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom  
NIP. 19841010 201903 1 012

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Danendra Farrel Bhagawanta Indaru  
NIM : 210605110057  
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika  
Judul Skripsi : Sistem Kendali Kursi Roda Menggunakan Algoritma *Finite State Automata*.

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 19 Oktober 2025  
Yang membuat pernyataan,



Danendra Farrel Bhagawanta Indaru  
NIM. 210605110057

## **MOTTO**

*... Carpe Diem (Manfaatkan hari ini)...*

## HALAMAN PERSEMBAHAN

*Alhamdulillahi rabbil 'alamin*, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW, suri teladan umat manusia, yang telah membimbing kita dari zaman jahiliyah menuju jalan Islam yang penuh cahaya.

Dengan penuh rasa syukur dan kerendahan hati, karya ini penulis persembahkan kepada diri penulis sendiri sebagai wujud ikhtiar dan kesabaran dalam menuntaskan perjalanan akademik; kepada kedua orang tua tercinta atas doa, kasih sayang, dan pengorbanan yang tiada henti; kepada para dosen pembimbing dan pengajar atas bimbingan, ilmu, dan arahan yang berharga kepada saudara, kerabat, sahabat, serta teman seperjuangan yang selalu memberikan doa dan semangat serta kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyelesaian skripsi ini. Semoga karya sederhana ini dapat memberikan manfaat.

## **KATA PENGANTAR**

*Assalamualaikum wr wb.*

Segala puji dan syukur yang tak terhingga, penulis mengucapkan Alhamdulillah kepada Allah SWT atas segala limpahan rahmat, karunia dan petunjuk-Nya dan tak lupa, shalawat, serta salam selalu tercurah kepada Nabi Muhammad SAW. Proses penyusunan skripsi ini tidak terlepas dari bantuan, doa, serta dukungan dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan rendah hati, penulis mengucapkan terima kasih secara khusus kepada:

1. Prof. Dr. Ilfi Nur Diana selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si. selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ajib Hanani, M.T selaku pembimbing utama dan Dr. Agung Teguh Wibowo Almais selaku pembimbing II yang memberi bimbingan, arahan, serta dorongan dalam penyusunan skripsi ini.
5. Dr. Ir. Yunifa Miftachul Arif, M.T selaku penguji I dan Nur Fitriyah Ayu Tunjung Sari, M.Cs selaku penguji II yang telah memberikan masukan untuk penyusunan skripsi ini

6. Seluruh Dosen, Admin, Laboran dan Civitas Akademik Program Studi Teknik Informatika yang telah memberi ilmu dan pengalaman selama masa perkuliahan.
7. Mama Rosmalia Simatupang dan Papa Habil Septa Indaru, kedua orang tua penulis yang selalu menjadi sumber kekuatan dan motivasi terbesar penulis. Terima kasih atas doa, semangat, serta cinta dan dukungan yang selalu ada. Mama dan Papa selalu menjadi sumber semangat dan penguat hati. Tanpa Mama dan Papa penulis tidak akan sampai sejauh ini. Terima kasih atas segala pengorbanan dan cinta yang tak ternilai sepanjang perjalanan penulis.
8. Kakak tersayang Izdhihar Syifa Aurellia Indaru terima kasih untuk dukungan yang selalu kakak berikan hingga penulis dapat kekuatan untuk terus melangkah maju.
9. Darin Marwa Fadiyah yang telah menjadi sosok istimewa yang selalu memberikan motivasi dan penyemangat dalam penulisan skripsi ini.
10. Seluruh teman-teman Teknik Informatika, khususnya angkatan 2021 'Aster', terima kasih atas waktu, pertemanan, dan pengalaman yang telah dilalui bersama.

Malang, 19 Oktober 2025

Penulis

## DAFTAR ISI

<b>HALAMAN PENGAJUAN .....</b>	<b>ii</b>
<b>HALAMAN PERSETUJUAN .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iv</b>
<b>PERNYATAAN KEASLIAN TULISAN .....</b>	<b>v</b>
<b>MOTTO .....</b>	<b>vi</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>viii</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xiii</b>
<b>ABSTRAK .....</b>	<b>xiii</b>
<b>ABSTRACT .....</b>	<b>xv</b>
<b>الملخص.....</b>	<b>xvi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Pernyataan Masalah .....	3
1.3    Tujuan Penelitian .....	3
1.4    Batasan Masalah .....	3
1.5    Manfaat Penelitian .....	4
<b>BAB II STUDI PUSTAKA.....</b>	<b>5</b>
2.1    Penelitian Terkait.....	5
2.2 <i>MIT App Inventor</i> .....	8
2.3    Algoritma <i>Finite State Automata</i> (FSA).....	9
2.4    Raspberry Pi.....	10
2.5 <i>Motor DC</i> dan Sistem Transmisi Daya (Gear dan Rantai) .....	11
2.5.1 <i>Gear Mini 66T</i> .....	12
2.5.2    Rantai Mini GP 25H.....	13
2.6    Aki Motor 12V 7Ah.....	13
2.7 <i>Motor Driver BTS7960 43A</i> .....	14
2.8 <i>Stemming</i> .....	14
2.9 <i>Speech Recognition</i> .....	15
2.10 <i>String Matching</i> .....	16
<b>BAB III DESAIN DAN IMPLEMENTASI.....</b>	<b>17</b>
3.1    Akuisisi Data.....	17
3.1.1    Data Primer.....	17
3.1.2    Data Sekunder .....	18
3.2    Desain .....	19
3.2.1    Tahapan <i>Input</i> .....	20
3.2.2    Tahapan <i>Process</i> .....	21
3.2.3    Tahapan <i>Output</i> .....	23
3.3 <i>Google Speech Recognition</i> .....	24
3.4 <i>Stemming</i> .....	26
3.5 <i>Stopword Removal</i> .....	27
3.6    Algoritma <i>Finite State Automata</i> .....	28
3.6.1    Inisialisasi dan Persiapan <i>Input</i> .....	30
3.6.2    Proses Iterasi dan Transisi <i>State</i> .....	30

3.6.3	Logika <i>Self-Loop</i> dan Penyelesaian.....	31
3.6.4	Penentuan Hasil Akhir.....	32
3.7	Perancangan Aplikasi .....	33
3.8	Rangkaian Komponen.....	34
3.9	Pengujian Sistem .....	37
<b>BAB IV UJI COBA DAN PEMBAHASAN.....</b>		<b>41</b>
4.1	Data Pengujian.....	41
4.1.1	Sistem <i>Software</i> .....	41
4.1.2	Sistem <i>Hardware</i> .....	42
4.2	Rangkaian Desain Kursi Roda.....	44
4.3	Pengujian Sistem .....	47
4.3.1	Hasil Pengujian dengan bobot 65 kg .....	47
4.3.2	Hasil Pengujian Dengan Bobot 58kg .....	50
4.4	Hasil Pengujian Total.....	51
4.5	Integrasi Islam .....	53
4.5.1	Landasan Etis: Prinsip Keseimbangan Duniawi dan Ihsan .....	54
4.5.2	Landasan Filosofis: Prinsip Penundukan Alam dan Dorongan Berpikir.....	55
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>58</b>
5.1	Kesimpulan .....	58
5.2	Saran .....	58

## **DAFTAR PUSTAKA**

## DAFTAR GAMBAR

Gambar 2. 1 Input Maju .....	9
Gambar 2. 2 <i>Raspberry Pi 3</i> .....	10
Gambar 2. 3 Motor DC .....	11
Gambar 2. 4 <i>Motor Driver BTS7960 43A</i> .....	14
Gambar 3. 1 Alur Penelitian .....	17
Gambar 3. 2 Desain Sistem.....	24
Gambar 3. 3 Proses <i>Stemming</i> .....	27
Gambar 3. 4 Cara Kerja <i>Finite State Automata</i> .....	29
Gambar 3. 5 Rancangan Aplikasi .....	34
Gambar 3. 6 Rancangan Sistem Pada Fritzing.....	37
Gambar 4. 1 Halaman Utama.....	42
Gambar 4. 2 Rangkaian Hardware.....	43
Gambar 4. 3 Tampak Kursi Roda Dari Belakang .....	44
Gambar 4. 4 Tampak Kursi Roda Dari Depan.....	45
Gambar 4. 5 Tampak Kursi Roda Dari Kanan.....	46
Gambar 4. 6 Tampak Kursi Roda Dari Kiri.....	46

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu .....	7
Tabel 2. 2 Spesifikasi <i>Raspberry Pi 3</i> .....	11
Tabel 2. 3 Spesifikasi Motor DC .....	12
Tabel 2. 4 <i>Spesifikasi Motor Driver BTS7960 43A</i> .....	14
Tabel 3. 1 Pencocokan <i>Finite State Automata</i> .....	33
Tabel 3. 2 Contoh Pengujian <i>Error Sistem</i> .....	38
Tabel 4. 1 Hasil Pengujian Bobot 65kg .....	48
Tabel 4. 2 Hasil Pengujian Bobot 58kg .....	50
Tabel 4. 3 Hasil Pengujian Total.....	52

## ABSTRAK

Indaru, Danendra Farrel Bhagawanta. 2025. **Sistem Kendali Kursi Roda Menggunakan Algoritma Finite State Automata.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T (II) Dr. Agung Teguh Wibowo Almais, M.T.

**Kata kunci:** *Algoritma Finite State Automata, Kursi roda, Penyandang disabilitas, Raspberry Pi 3*

Keterbatasan mobilitas menjadi hambatan besar bagi penyandang disabilitas fisik. Penelitian ini bertujuan mengembangkan sistem kendali kursi roda berbasis perintah suara untuk meningkatkan kemandirian pengguna. Sistem dirancang secara terintegrasi, di mana perintah suara dari pengguna diproses melalui sebuah aplikasi Android dan kemudian dikirim ke mikrokontroler. Metode *Finite State Automata* (FSA) diterapkan sebagai logika utama untuk mengenali pola perintah dasar seperti maju, mundur, belok kanan, dan kiri. Perintah yang teridentifikasi kemudian dijalankan oleh aktuator motor pada kursi roda. Hasil pengujian menunjukkan bahwa sistem mampu mengenali perintah suara dengan tingkat *error* 10%, yang membuktikan efektivitas sistem sebagai teknologi bantu yang dapat meningkatkan kemandirian mobilitas pengguna.

## ABSTRACT

Indaru, Danendra Farrel Bhagawanta. 2025. ***Wheelchair Control System Using Finite State Automata Algorithm.*** Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Ajib Hanani, M.T (II) Dr. Agung Teguh Wibowo Almuis, M.T.

**Key words:** *Finite State Automata Algorithm, People with disabilities, Raspberry Pi 3, Wheelchair*

Mobility limitations pose a significant obstacle for individuals with physical disabilities. This research aims to develop a voice command-based control system for wheelchairs to enhance user independence. The system is designed in an integrated manner, where user voice commands are processed through an Android application and then sent to a microcontroller. The Finite State Automata (FSA) method is applied as the core logic to recognize patterns of basic movement commands such as forward, backward, right, and left. The identified commands are then executed by the wheelchair's motor actuators. Test results show that the system can recognize voice commands with an error rate of 10%, demonstrating its effectiveness as an assistive technology for improving user mobility independence.

## الملخص

إندارو، داندرا فاريل بجاجانتا. 2025. نظام التحكم في الكراسي المتحركة باستخدام خوارزمية آلة محدودة الحالات. البحث الجامعي. قسم هندسة المعلومات، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالاج. المشرف الأول: أجيبي حناني، الماجستير؛ المشرف الثاني: د. أغونج تيغوه وبيبوو أليس، الماجستير.

**الكلمات الرئيسية:** خوارزمية آلة محدودة حالات، كرسي متحرك، أشخاص ذوي إعاقة، راسيري باي 3

تعد محدودية الحركة عقبة كبيرة للأشخاص ذوي الإعاقة الجسدية. هدف هذا البحث إلى تطوير نظام تحكم في الكراسي المتحركة يعتمد على الأوامر الصوتية لتعزيز استقلالية مستخدميها. تم تصميم النظام بشكل متكامل، حيث تتم معالجة الأوامر الصوتية من المستخدم عبر تطبيق أندرويد ومن ثم إرسالها إلى المتحكم الدقيق. تم تطبيق خوارزمية آلة محدودة الحالات (*Finite State Automata*) كمنطق رئيسي للتعرف على أنماط الأوامر الأساسية مثل التقدم، التراجع، الانعطاف إلى اليمين واليسار. بعد ذلك، يتم تنفيذ الأوامر التي تم التعرف عليها بواسطة الحرك في الكرسي المتحرك. أظهرت نتائج الاختبارات أن النظام قادر على التعرف على الأوامر الصوتية بنسبة خطأ قدرها 10%， مما يثبت فعالية النظام كتقنية مساعدة يمكنها تعزيز استقلالية الحركة لدى المستخدمين.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Mobilitas merupakan salah satu aspek penting dalam kehidupan sehari-hari. Bagi penyandang disabilitas fisik yang bergantung pada kursi roda, keterbatasan dalam mengendalikan kursi roda secara mandiri dapat menjadi hambatan dalam aktivitas sehari-hari. Ketergantungan pada orang lain untuk mengoperasikan kursi roda dapat mengurangi kemandirian dan menurunkan kualitas hidup mereka.

Seiring dengan perkembangan teknologi, berbagai inovasi telah dikembangkan untuk meningkatkan aksesibilitas bagi penyandang disabilitas. Salah satu pendekatan yang dapat diterapkan adalah penggunaan sistem kendali otomatis yang memungkinkan penggunanya untuk mengendalikan kursi roda dengan lebih mudah dan efisien. Dalam penelitian ini, dikembangkan sebuah sistem kendali kursi roda berbasis algoritma *Finite State Automata* (FSA). Algoritma ini memiliki keunggulan dalam mengenali pola perintah dan mengeksekusi pergerakan secara terstruktur.

Dengan menerapkan *Finite State Automata* (FSA), sistem kendali kursi roda dapat memproses perintah secara efisien, memastikan keakuratan pola yang dikenali, serta meningkatkan responsivitas sistem. Implementasi teknologi ini diharapkan dapat memberikan solusi bagi penyandang disabilitas dalam meningkatkan mobilitas mereka secara mandiri tanpa harus bergantung pada orang lain.

Sejalan dengan tujuan ini, Al-Qur'an memberikan panduan kepada manusia untuk tolong-menolong, sebagaimana firman Allah:

وَابْتَغِ فِيمَا أَتَكَ اللَّهُ الدَّارَ الْآخِرَةَ وَلَا تَنْسِ نَصِيبِكَ مِنَ الدُّنْيَا وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ وَلَا تَنْعِ  
الْفَسَادَ فِي الْأَرْضِ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ

*"Dan, carilah pada apa yang telah dianugerahkan Allah kepadamu (pahala) negeri akhirat, tetapi janganlah kamu lupakan bagianmu di dunia. Berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik kepadamu dan janganlah kamu berbuat kerusakan di bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.." (QS. Al-Qasas: 77).*

Dalam kitab Tafsir Ibnu Katsir pada ayat ini dijelaskan bahwa makna "janganlah kamu lupakan bahagianmu dari (kenikmatan) duniawi" adalah dorongan untuk memanfaatkan segala potensi dan anugerah Allah di dunia (termasuk ilmu pengetahuan dan teknologi) untuk mencapai kemaslahatan, bukan sekadar kenikmatan semata. Tafsir ini menggarisbawahi bahwa inovasi teknologi adalah bagian dari pemanfaatan karunia dunia yang diizinkan dan dianjurkan. Lebih lanjut, perintah "berbuat baiklah (*ahsin*)" menjadi landasan etis (filosofis) bahwa tujuan utama pengembangan teknologi *assistive* ini adalah untuk mewujudkan kebaikan (ihsan), yaitu memberikan kemudahan dan meningkatkan aksesibilitas bagi penyandang disabilitas.

Berdasarkan latar belakang di atas, dibuatlah sistem kendali kursi roda menggunakan algoritma *Finite State Automata* (FSA). Sistem yang dibangun diharapkan dapat diimplementasikan untuk membantu penyandang disabilitas fisik, terutama mereka yang mengalami keterbatasan mobilitas dalam mengoperasikan kursi roda secara mandiri. Dengan adanya sistem ini, diharapkan dapat meningkatkan kemandirian dan kenyamanan bagi penyandang disabilitas dalam

menjalankan aktivitas sehari-hari tanpa bergantung pada bantuan orang lain. Teknologi ini juga diharapkan dapat memberikan kontribusi positif dalam pengembangan perangkat *assistive technology* yang lebih inklusif dan mudah diakses.

### **1.2 Pernyataan Masalah**

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah bagaimana performa algoritma *Finite State Automata* (FSA) sebagai sistem kendali kursi roda dalam meningkatkan kemandirian mobilitas dan mengurangi ketergantungan pada tenaga manusia bagi penyandang disabilitas fisik?

### **1.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah Merancang dan mengembangkan sistem kendali kursi roda yang dapat meningkatkan kemandirian mobilitas penyandang disabilitas fisik melalui perintah suara.

### **1.4 Batasan Masalah**

Dalam penelitian ini, sistem kendali kursi roda dibatasi hanya dapat mengenali perintah suara sederhana seperti maju, mundur, belok kiri, belok kanan. Proses pengenalan suara difokuskan pada penggunaan bahasa Indonesia dengan kata-kata tertentu yang telah ditentukan sebelumnya. Penelitian ini juga hanya menitikberatkan pada fungsi kendali gerak kursi roda tanpa mencakup pengembangan fitur tambahan seperti sensor tabrakan, navigasi otomatis, ataupun integrasi dengan *Internet of Things* (IoT). Selain itu, uji coba sistem dilakukan pada

lingkungan terbatas di dalam ruangan (indoor), sehingga tidak mempertimbangkan faktor eksternal seperti kebisingan berlebih, kondisi permukaan jalan yang tidak rata, maupun pengaruh cuaca.

### **1.5 Manfaat Penelitian**

Manfaat penelitian ini mencakup beberapa aspek yang diharapkan menjadi solusi inovatif untuk meningkatkan mobilitas penyandang disabilitas dan menjadi referensi bagi pengembangan teknologi asistif di masa depan.

## **BAB II**

### **STUDI PUSTAKA**

#### **2.1 Penelitian Terkait**

Sejumlah penelitian sebelumnya telah dijelaskan sebagai bahan referensi dan perbandingan. Penelitian ini juga berupaya untuk tidak menyerupai studi-studi yang sudah dilakukan. Pada bagian kajian pustaka, peneliti menyajikan berbagai hasil dari penelitian-penelitian terdahulu.

Terdapat beberapa penelitian sebelumnya yang menggunakan algoritma *Finite State Automata* (FSA). Salah satunya adalah penelitian oleh (Ginting, 2024) yang mengimplementasikan *Finite State Automata* (FSA) dalam perancangan *vending machine*, khususnya untuk mesin perjual minuman Boba. Peneliti tersebut memberikan hasil yang membuktikan bahwa algoritma *Finite State Automata* (FSA) dapat mengoptimalkan desain dan pengoperasian mesin jual, meningkatkan efisiensi dan kepuasan pengguna.

Selain itu, penelitian sebelumnya yang menggunakan konsep *Finite State Automata* (FSA) dalam berbagai aplikasi validasi data. Salah satunya adalah penelitian oleh (Kusuma et al., 2022) yang mengimplementasikan konsep *Finite State Automata* (FSA) dalam validasi permohonan pengajuan layanan akun digital pada aplikasi Menara Masjid. Penelitian tersebut menunjukkan bahwa penerapan *Finite State Automata* (FSA) dapat mempercepat proses validasi data, mengurangi kesalahan pencatatan, serta memastikan bahwa data yang disimpan telah sesuai dengan format yang ditentukan.

Berdasarkan penelitian oleh (Huda et al., 2023) mengimplementasikan *Finite State Automata* (FSA) pada sistem perparkiran kendaraan bermotor di Bandara Raja Haji Fisabilillah Tanjungpinang. Studi ini menunjukkan bahwa penggunaan FSA dapat meningkatkan efisiensi dan akurasi sistem parkir dengan mengotomatiskan proses validasi tiket serta mengurangi potensi kesalahan input pengguna.

Sementara itu, penelitian oleh (Hidayat et al., 2021) menerapkan konsep FSA pada *vending machine* untuk simulasi pemesanan masakan Padang. Penelitian ini membuktikan bahwa FSA dapat digunakan sebagai logika dasar dalam pengoperasian *vending machine*, memungkinkan transaksi pemesanan makanan dilakukan secara otomatis dan lebih efisien tanpa perlu interaksi langsung dengan penjual.

Penelitian yang dilakukan oleh (Kurniawan et al., 2022) memiliki persamaan dimana mengembangkan kursi roda berbasis *IoT* menggunakan NodeMCU sebagai mikrokontroler dan aplikasi berbasis *Android* untuk mengendalikan kursi roda. Selain itu, kedua sistem memanfaatkan Motor DC sebagai penggerak utama dan jaringan internet sebagai media komunikasi. Namun, perbedaannya terletak pada metode yang digunakan, di mana penelitian Simplisius Kurniawan menggunakan metode *Proportional Integral Derivative Controller* (PID) untuk mengatur kecepatan motor dan kestabilan gerakan kursi roda melalui aplikasi *Blynk*, sedangkan penelitian ini menggunakan algoritma *Finite State Automata* (FSA) untuk mengenali pola perintah suara yang dikonversi menjadi teks melalui *speech recognition*, kemudian diterjemahkan menjadi perintah gerakan

seperti maju, mundur, belok kanan, dan belok kiri tanpa menggunakan sensor garis atau sensor ultrasonik.

Di sisi lain, terdapat penelitian yang membahas perancangan kursi roda berbasis mikrokontroler yang dilakukan oleh (Mardianto et al., 2024) penelitian ini menggunakan objek yang sama kursi roda. Namun, penelitian tersebut tidak berfokus pada penggunaan algoritma tertentu. Selain itu terdapat perbedaan *hardware* berupa *Arduino Uno*. Tabel 2. 1 dibawah merupakan penelitian terdahulu yang telah diuraikan:

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
1.	Muhammad Daffa Ginting	Implementasi <i>Finite State Automata</i> ( <i>FSA</i> ) pada aplikasi simulasi <i>Vending Machine</i> minuman BOBA	<i>Finite State Automata</i>	<i>FSA</i> secara efektif mengelola proses transaksi, memastikan output akurat dari minuman yang dipilih berdasarkan input pengguna dan metode pembayaran.
2.	Muhammad Romadhona Kusuma, Windu Gata Laela Kurniawati, Friyadi, Ahmad Baihaqi	Penerapan <i>Finite State Automata</i> pada Validasi Permohonan Pengajuan Layanan Akun Digital	<i>Finite State Automata</i>	Penelitian menyimpulkan bahwa menerapkan <i>Finite State Automata</i> ( <i>FSA</i> ) dalam proses validasi pengiriman akun digital secara signifikan mengurangi waktu yang diperlukan untuk memproses aplikasi.
3.	Dwi Nurul Huda, Amalia Zahara, Danil Hardinata	Pemanfaatan Konsep <i>Finite State Automata</i> Pada Sistem Perparkiran Kendaraan Bermotor Bandara Raja Haji Fisabilillah	<i>Finite State Automata</i>	Penelitian ini menyimpulkan bahwa penerapan sistem parkir terkomputerisasi menggunakan <i>Finite State Automata</i> ( <i>FSA</i> ) dapat secara signifikan meningkatkan efisiensi dan akurasi parkir kendaraan di Bandara Raja Haji Fisabilillah di Tanjungpinang
4.	Syarif Hidayat, Fadillah Said, Fakihotun Titiani	Desain Konsep <i>Finite State Automata</i> ( <i>FSA</i> ) Pada Simulasi <i>Vending Machine</i> ( <i>VM</i> ) Masakan Padang	<i>Finite State Automata</i>	Penelitian menyimpulkan bahwa <i>Finite State Automata</i> ( <i>FSA</i> ) dapat berfungsi sebagai logika fundamental untuk mensimulasikan Mesin Penjual Otomatis ( <i>VM</i> ) khusus untuk Masakan Padang.
5.	Simplisius Yekri Kurniawan, Aries Boedi	Kursi Roda Otomatis Berbasis <i>IoT</i> ( <i>Internet Of Things</i> )	PID	Penelitian menyimpulkan bahwa kursi roda otomatis, yang dikendalikan melalui aplikasi smartphone, secara signifikan

No	Nama Peneliti	Judul Penelitian	Metode	Hasil Penelitian
	Setiawan, Wahyu Dirgantara	Menggunakan Metode <i>PID</i> ( <i>Proporional Integral Derivative Controller</i> )		meningkatkan kenyamanan pengguna dan mobilitas bagi para penyandang cacat. Penerapan metode kontrol PID memastikan navigasi responsif dan penghindaran rintangan, berkontribusi pada efektivitas kursi roda secara keseluruhan.
6.	Syahrizan, Satriyo, Eko Mardianto, Rianda Tarkan	Perancangan Dan Implementasi Sistem Kendali Kursi Roda Berbasis Mikrokontroler	-	Penelitian menyimpulkan bahwa sistem kontrol kursi roda yang dikembangkan secara signifikan meningkatkan kemandirian dan mobilitas pengguna dengan keterbatasan fisik.

## 2.2 *MIT App Inventor*

*MIT App Inventor* adalah platform yang dirilis pada Maret 2010. *Google released App Inventor which is currently being maintained by Massachusetts Institute of Technology*. *App Inventor* menyediakan antarmuka pengguna grafis dengan semua komponen yang diperlukan untuk membangun aplikasi dan memungkinkan siapa pun untuk membangun aplikasi untuk memenuhi kebutuhan mereka. Aplikasi yang dikembangkan menggunakan alat ini dapat dengan mudah dipindahkan ke ponsel, dibagikan dengan orang lain, atau bahkan dikirim ke *Google Play Store* untuk didistribusikan ke seluruh dunia(Kaddipujar et al., 2022).

Menurut (Putri et al., 2024), *Mit App Inventor* memungkinkan kita untuk membuat dan memprogram secara mandiri aplikasi untuk *Android* tanpa mempelajari berbagai macam bahasa pemrograman. *MIT App Inventor* dapat disimpulkan sebagai platform pengembangan aplikasi berbasis *Android* yang dirancang untuk mempermudah siapa saja, termasuk pemula, dalam membuat aplikasi tanpa memerlukan pengetahuan mendalam tentang bahasa pemrograman. Platform ini menyediakan antarmuka grafis dan komponen yang lengkap untuk

membangun aplikasi, memungkinkan pengguna untuk membuat, membagikan, dan mendistribusikan aplikasi mereka dengan mudah, termasuk melalui *Google Play Store*.

### 2.3 Algoritma *Finite State Automata* (FSA)

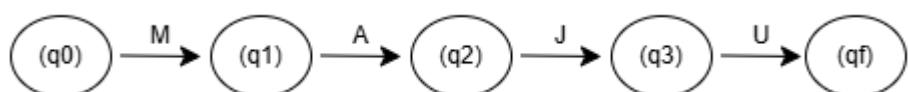
Algoritma *Finite State Automata* (FSA) adalah salah satu algoritma pencocokan *string* yang efisien karena menggunakan pendekatan *automata* untuk mengenali pola dengan cepat dan akurat. Algoritma ini memanfaatkan *state* dan transisi antar *state* untuk menentukan apakah sebuah teks mengandung pola tertentu. Konsep dasarnya memungkinkan sistem untuk memproses *input* tanpa mengulang pemeriksaan dari awal ketika terjadi ketidakcocokan, berkat adanya tabel transisi yang telah dibangun selama tahap *preprocessing* (Ikhsan et al., 2023).

*Finite State Automata* (FSA) memiliki *state* yang banyaknya berhingga, dan dapat berpindah-pindah dari suatu *state* ke *state* lain. Secara formal *Finite State Automata* (FSA) dinyatakan oleh 5 tupel atau  $M=(Q, \Sigma, \delta, S, F)$  (Meiyanti & Sandy, 2024).

Keterangan:

- a)  $Q$  adalah himpunan state / kedudukan.
- b)  $\Sigma$  adalah himpunan symbol input / masukan / abjad.
- c)  $\delta$  adalah fungsi transisi.
- d)  $S$  adalah state awal atau dudukan awal (initial state).
- e)  $F$  adalah himpunan state akhir.

di bawah ini menunjukkan bagaimana sistem mengelola *input string* “maju” untuk mengenali pola perintah:



Gambar 2. 1 Input Maju

Penjelasan:

- a)  $q_0$  = Sistem memulai di state ini. Jika *input* adalah "m", sistem berpindah ke  $q_1$ .
- b)  $q_1$  = Jika input berikutnya "a", sistem berpindah ke  $q_2$ .
- c)  $q_2$  = Jika input berikutnya "j", sistem berpindah ke  $q_3$ .
- d)  $q_3$  = Jika input berikutnya "u", sistem berpindah ke  $q_f$  (final state).
- e)  $q_f$  = Sistem mengenali string "maju" sebagai pola yang valid

Jika ada karakter yang tidak sesuai di salah satu transisi, sistem akan kembali ke *state* awal atau ke *state* tertentu sesuai fungsi transisinya. Dengan pendekatan ini, algoritma dapat mengenali pola perintah dengan cepat dan akurat tanpa perlu memeriksa ulang dari awal.

## 2.4 Raspberry Pi

*Raspberry Pi* adalah sebuah komputer *mini single-board* yang dikembangkan oleh *Raspberry Pi Foundation*, dirancang untuk mendukung pembelajaran komputer dan pengembangan sistem berbasis mikrokontroler. Jenis *Raspberry Pi* yang umum digunakan untuk sistem seperti ini adalah *Raspberry Pi 3 Model B* atau *Raspberry Pi 4 Model B*, karena memiliki kombinasi yang baik antara kinerja prosesor, jumlah pin GPIO, dan konektivitas. *Raspberry Pi* menjalankan sistem operasi berbasis *Linux* seperti *Raspberry Pi OS*, yang mendukung bahasa pemrograman seperti *Python*, sangat cocok untuk automasi, dan pengendalian perangkat keras secara *real-time*. yang ditunjukan pada Gambar 2. 2 dan Tabel 2. 2 di bawah ini.



Gambar 2. 2 *Raspberry Pi 3*

Tabel 2. 2 Spesifikasi *Raspberry Pi 3*

Prosesor	Broadcom BCM2837 Quad Core 1.2 Ghz 64-Bit
Arsitektur	ARM
Sub-Arsitektur	ARMv8
Ram	1 GB
Jaringan Wireless	Wifi dan Bluetooth Low Energy
Ethernet	10/100 Mbps
GPIO	40 pin
USB	4x USB 2.0 port

## 2.5 Motor DC dan Sistem Transmisi Daya (Gear dan Rantai)

Motor DC (Arus Searah) adalah perangkat elektromagnetik yang berfungsi mengubah energi listrik menjadi energi gerak mekanik berupa putaran. Motor ini beroperasi berdasarkan prinsip gaya Lorentz, di mana interaksi medan magnet dan arus listrik menghasilkan gerakan putar. Dalam sistem kontrol gerakan yang memerlukan daya dan kecepatan respons yang baik, Motor DC sering dipilih karena kemudahan kontrol kecepatannya melalui pengaturan tegangan. Motor ini beroperasi pada tegangan tertentu dan mulai berputar saat mendapatkan sinyal perintah dari mikrokontroler seperti NodeMCU.

Untuk aplikasi pada sistem kursi roda, motor ini beroperasi tanpa *gearbox* internal, yang berarti putaran murni dihasilkan oleh motor dan memerlukan sistem transmisi eksternal. Berikut adalah contoh Motor DC dan spesifikasi pada Gambar 2. 3 dan Tabel 2. 3.



Gambar 2. 3 Motor DC

Tabel 2. 3 Spesifikasi Motor DC

Rated speed	2750 RPM
Rated output power	250 Watt
Voltage	12V
No-load current	0.7-1.4A
Weight	1.8 Kg

Karena Motor DC yang digunakan tidak dilengkapi *gearbox* internal, mekanisme transmisi daya menjadi komponen krusial untuk menurunkan kecepatan putaran (RPM tinggi dari motor) sekaligus meningkatkan torsi yang dibutuhkan untuk menggerakkan roda kursi roda.

### 2.5.1 *Gear Mini 66T*

Dalam sistem mekanis, *Gear* (gigi) berfungsi sebagai perangkat penting untuk mentransmisikan torsi dan gerakan rotasi secara presisi dari satu poros ke poros lainnya. *Gear* adalah komponen utama yang menentukan rasio reduksi, yaitu perbandingan kecepatan putar dan torsi antara poros *input* dan poros *output*. Dalam desain kursi roda ini, *Gear Mini 66T* secara spesifik digunakan sebagai *gear* yang digerakkan (*driven gear*) yang dipasang pada poros roda. Angka 66T menunjukkan bahwa *gear* ini memiliki 66 mata gigi (*teeth*). Penggunaan *gear* dengan jumlah gigi yang jauh lebih besar dibandingkan *gear* penggerak pada motor bertujuan untuk menciptakan rasio reduksi yang tinggi, yang sangat krusial. Rasio reduksi tinggi ini secara efektif akan meningkatkan torsi (gaya putar) yang diperlukan untuk menggerakkan beban berat sekaligus menurunkan kecepatan putaran (RPM) motor ke level yang aman dan memadai untuk mobilitas, menjamin sistem memiliki daya dorong yang kuat dan terkontrol.

### 2.5.2 Rantai Mini GP 25H

Rantai (chain) berfungsi sebagai komponen krusial yang berperan sebagai penghubung dan pemindah daya utama antara *gear* motor (*drive gear*) dengan *Gear Mini 66T (driven gear)* yang terpasang pada poros roda. Dalam sistem kursi roda ini, digunakan Rantai Mini GP 25H. Pemilihan jenis rantai ini didasarkan pada ukurannya yang ringkas kode 25H mengindikasikan *pitch* (jarak antar pin) dan spesifikasi kekuatannya sehingga sangat ideal untuk transmisi daya yang efisien dalam ruang terbatas. Sistem transmisi yang mengandalkan rantai dan *gear* memastikan transfer daya dari motor ke roda terjadi secara positif, yang berarti tidak terjadi selip (slip). Hal ini menjamin pergerakan kursi roda menjadi stabil, efisien, dan memiliki presisi tinggi dalam setiap perintah kontrol gerakan.

### 2.6 Aki Motor 12V 7Ah

Aki motor 12V 7Ah adalah jenis baterai kering (*Sealed Lead Acid/SLA*) yang umum digunakan sebagai sumber daya listrik pada kendaraan bermotor, sistem UPS, dan berbagai perangkat elektronik portabel. Dalam konteks penelitian ini, aki digunakan sebagai sumber daya utama untuk menyuplai listrik ke motor DC yang menggerakkan kursi roda.

Aki ini memiliki tegangan nominal sebesar 12 volt dan kapasitas penyimpanan energi sebesar 7 *ampere-hour* (Ah). Nilai 7Ah menunjukkan bahwa aki ini mampu menyuplai arus sebesar 7 ampere selama 1 jam, atau 3.5 ampere selama 2 jam, sebelum kehabisan daya, tergantung pada beban sistem

## 2.7 ***Motor Driver BTS7960 43A***

*Driver Motor BTS7960 43A* adalah modul modul pengendali motor DC yang dirancang untuk menangani arus tinggi hingga 43A, cocok untuk proyek-proyek yang memerlukan tenaga besar seperti kursi roda elektrik, robot berukuran besar, dan kendaraan kecil berbasis mikrokontroler. Modul ini menggunakan dua buah *IC* *BTS7960* dari Infineon yang mendukung konfigurasi *H-Bridge*, memungkinkan kontrol arah putaran motor serta kecepatan melalui sinyal PWM (*Pulse Width Modulation*) (Bugade et al., 2024). Ditunjukan pada Gambar 2. 4 dan Tabel 2. 4.



Gambar 2. 4 *Motor Driver BTS7960 43A*

Tabel 2. 4 *Spesifikasi Motor Driver BTS7960 43A*

Catu Daya Motor	5.5V - 27V.
Catu Daya Driver	5V
Logic TTL Driver	5V
Total Pin	8

## 2.8 ***Stemming***

*Stemming* adalah proses dalam pemrosesan bahasa alami (*Natural Language Processing* atau *NLP*) yang bertujuan untuk mengurangi sebuah kata ke bentuk dasarnya atau akar katanya (*root word*). Proses ini penting untuk menghilangkan imbuhan seperti awalan, akhiran, atau sisipan sehingga kata-kata

yang memiliki makna dasar yang sama dapat dikenali sebagai satu entitas. Dalam konteks pengenalan perintah suara untuk sistem kendali kursi roda, *stemming* digunakan untuk menyederhanakan teks hasil konversi suara sebelum dilakukan pencocokan pola menggunakan algoritma *Finite State Automata* (FSA).

Menurut (Anwar et al., 2020), *stemming* digunakan untuk meningkatkan efisiensi pencarian teks karena sistem tidak perlu memproses variasi kata yang berbeda-beda. Sebagai contoh, kata-kata seperti "membuka", "dibuka", "membukakan", atau "pembukaan" semuanya akan direduksi menjadi akar kata "buka". Dengan demikian, algoritma pencocokan pola tidak perlu mempertimbangkan setiap bentuk kata, cukup berfokus pada akar katanya saja.

## 2.9 *Speech Recognition*

*Speech Recognition* atau Pengenalan Suara, yang juga dikenal secara akademis sebagai *Automatic Speech Recognition* (ASR), adalah sebuah disiplin ilmu di bawah naungan ilmu komputer dan linguistik komputasional yang berfokus pada pengembangan teknologi untuk mengidentifikasi dan mentranskripsi bahasa lisan manusia menjadi bentuk teks. Secara fundamental, tujuan utama dari sistem ASR adalah untuk menjembatani kesenjangan komunikasi antara manusia dan mesin, memungkinkan interaksi yang lebih alami dan intuitif. Alih-alih menggunakan perangkat input konvensional seperti *keyboard* atau *mouse*, pengguna dapat berinteraksi dengan sistem komputer melalui modalitas paling alami bagi manusia, yaitu suara.

Teknologi ini merupakan komponen inti dari bidang Interaksi Manusia-Komputer (*Human-Computer Interaction - HCI*), yang mengubah paradigma

interaksi dari yang eksplisit (memerlukan perintah tertulis) menjadi implisit dan percakapan (*conversational*). Kemampuannya untuk mengubah sinyal akustik yang kompleks menjadi data terstruktur (teks) menjadikannya teknologi fundamental bagi berbagai aplikasi cerdas di era modern.

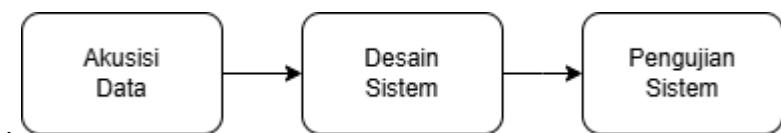
## 2.10 *String Matching*

*String Matching* adalah proses untuk menemukan kemunculan sebuah pola (*pattern*) dalam sebuah teks atau *string* (Gulo, 2022). Tujuan utama dari *string matching* adalah untuk menentukan apakah sebuah pola tertentu ada di dalam teks dan, jika ada, di mana letak pola tersebut. Teknik ini banyak digunakan dalam berbagai aplikasi, seperti pencarian teks, pengolahan bahasa alami (*Natural Language Processing*), dan sistem pencarian.

## **BAB III**

### **DESAIN DAN IMPLEMENTASI**

Pada bab ini dibahas secara rinci alur penelitian dari sistem kendali kursi roda menggunakan algoritma *Finite Automata* yang dimulai dengan desain untuk menentukan arsitektur dan fungsionalitas yang diperlukan. Tahap berikutnya adalah proses pengambilan data. Tahap berikutnya adalah implementasi sistem dari desain yang telah dirancang. Selanjutnya, dilakukan pengujian sistem untuk memastikan kinerjanya, diakhiri dengan hasil akurasi untuk mengevaluasi efektivitas sistem yang telah dibuat. Adapun alur penelitian dalam pembuatan sistem control ini dapat dilihat pada Gambar 3. 1.



Gambar 3. 1 Alur Penelitian

#### **3.1 Akuisisi Data**

Data yang digunakan dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer digunakan untuk menghasilkan perintah. Kemudian, data sekunder merupakan kata kunci atau *pattern*. Berikut perjelasan lebih lanjut:

##### **3.1.1 Data Primer**

Pada tahap awal penelitian ini, akuisisi data primer dilakukan melalui metode *input* suara. Proses perekaman dan pengenalan suara tersebut memanfaatkan fitur *Google Speech Recognition* yang terintegrasi, dengan konfigurasi khusus untuk mengenali dan memproses ujaran dalam Bahasa

Indonesia. Hasil dari proses pengenalan suara ini adalah konversi *input audio* menjadi representasi tekstual. Data teks yang telah dihasilkan selanjutnya akan menjadi masukan utama bagi tahap pemrosesan berikutnya, di mana teknik analisis menggunakan *Finite State Automata* (FSA) akan diimplementasikan. FSA akan digunakan untuk memodelkan dan menganalisis pola-pola sekuensial atau struktur linguistik yang terdapat dalam data teks tersebut, memungkinkan ekstraksi informasi atau validasi kaidah-kaidah tertentu yang menjadi fokus penelitian ini.

### **3.1.2 Data Sekunder**

Selain akuisisi data primer, penelitian ini juga melibatkan pemanfaatan data sekunder yang secara spesifik dirancang dan didefinisikan oleh peneliti. Data sekunder ini berperan krusial dalam fase pencocokan (*matching*) perintah suara, di mana sistem akan membandingkan *input* suara yang telah dikonversi menjadi teks dengan serangkaian perintah baku yang telah ditetapkan. Perintah-perintah suara yang menjadi bagian dari data sekunder ini meliputi 'maju', 'mundur', 'belok kanan', dan 'belok kiri'. Setiap perintah ini merepresentasikan instruksi kontrol fundamental untuk operasional kursi roda. Tujuan utama dari penggunaan data sekunder ini adalah untuk membangun basis referensi yang kokoh, sehingga sistem mampu secara akurat menginterpretasikan dan menentukan respons yang sesuai dari kursi roda terhadap setiap perintah suara yang diberikan oleh pengguna. Dengan demikian, data sekunder ini menjadi jembatan antara *input* suara non-standar dan aksi kontrol standar yang diharapkan dari sistem.

### 3.2 Desain

Penelitian ini mengimplementasikan metodologi pengujian yang terstruktur dan terdiri dari tiga tahapan utama, dirancang untuk mengevaluasi kinerja dan fungsionalitas sistem secara komprehensif. Tahap pertama adalah fase *Input*, yang memegang peranan krusial dalam akusisi data awal. Pada tahap ini, sistem berfungsi untuk mengumpulkan data suara dari pengguna secara *real-time*. Selain itu, tahap *input* juga bertanggung jawab untuk mengidentifikasi dan mempersiapkan pola data (*pattern*) yang akan digunakan sebagai referensi. Pola data ini mencakup serangkaian perintah suara yang telah didefinisikan sebelumnya, seperti 'maju', 'mundur', 'belok kanan', dan 'belok kiri', yang esensial untuk mengontrol pergerakan kursi roda.

Setelah data terkumpul, proses beralih ke Tahap kedua, yaitu Proses. Pada tahapan ini, data suara yang telah berhasil direkam akan mengalami serangkaian transformasi. Mula-mula, data suara tersebut akan dikonversi menjadi format teks menggunakan teknologi pengenalan suara. Selanjutnya, teks yang dihasilkan akan melalui tahap *preprocessing* yang intensif. *Preprocessing* ini mencakup dua teknik utama: *Stemming*, yang bertujuan untuk mengurangi kata-kata menjadi bentuk dasar atau akar kata, dan *Stopword Removal*, yang berfungsi untuk menghilangkan kata-kata umum yang tidak memiliki makna signifikan dalam konteks perintah (misalnya, 'yang', 'dan', 'adalah'). Tujuan dari *preprocessing* ini adalah untuk meningkatkan efisiensi dan akurasi dalam tahap pencocokan pola. Setelah *preprocessing*, sistem akan melakukan pencocokan pola (*pattern matching*) menggunakan algoritma *Finite State Automata* (FSA). FSA dirancang untuk secara

efektif mencocokkan teks hasil *preprocessing* dengan pola-pola perintah yang telah ditentukan. Hasil dari proses pencocokan ini adalah identifikasi perintah yang paling sesuai dengan input pengguna.

Terakhir, hasil dari tahap proses akan diteruskan ke Tahap ketiga, yaitu *Output*. Pada tahap ini, perintah yang telah teridentifikasi dan divalidasi akan diterjemahkan menjadi aksi fisik. Sistem akan mengirimkan sinyal kontrol ke motor penggerak kursi roda, yang kemudian akan mengendalikan mekanisme pergerakan maju, mundur, belok kanan, atau belok kiri sesuai dengan perintah yang diberikan oleh pengguna. Dengan demikian, ketiga tahapan ini bekerja secara sekuensial untuk memastikan bahwa setiap perintah suara dari pengguna dapat diinterpretasikan dan dieksekusi dengan tepat oleh sistem kursi roda.

Berikut ini adalah penjelasan lebih lanjut mengenai setiap tahapan dalam pengujian sistem:

### **3.2.1      Tahapan *Input***

Tahap Input merupakan gerbang awal dalam alur kerja sistem, di mana akuisisi data primer dilakukan. Pada fase ini, interaksi utama pengguna dengan sistem adalah melalui antarmuka kontrol suara. Ketika pengguna membuka aplikasi dan memberikan perintah, sistem akan segera memanfaatkan fitur canggih dari *Google Speech Recognition*. Teknologi ini berperan vital dalam mengonversi gelombang suara yang ditangkap menjadi representasi tekstual yang dapat diproses lebih lanjut. Akurasi konversi ini sangat penting karena teks inilah yang akan menjadi fondasi untuk semua analisis berikutnya.

Setelah konversi suara menjadi teks berhasil dilakukan, data tekstual tersebut tidak langsung diproses untuk pencocokan. Sebaliknya, teks akan melalui serangkaian proses normalisasi dimulai dengan tahapan *Stemming*. Untuk implementasi *stemming* ini, penelitian ini secara spesifik menggunakan *library Sastrawi*, sebuah pustaka pengolahan bahasa alami yang dirancang khusus untuk Bahasa Indonesia. *Stemming* dengan Sastrawi bertujuan untuk mereduksi setiap kata dalam teks perintah menjadi bentuk dasarnya atau akar katanya, sehingga variasi morfologis kata tidak menghambat proses pencocokan. Misalnya, kata 'majukan' dan 'majuin' akan direduksi menjadi 'maju'.

Setelah teks telah melalui proses *stemming*, langkah selanjutnya adalah melakukan proses pencocokan pola (*pattern matching*). Metode yang dipilih untuk tahap krusial ini adalah *Finite State Automata* (FSA). FSA dirancang untuk secara efisien membandingkan teks yang telah di *stemming* dengan serangkaian pola perintah yang telah didefinisikan sebelumnya (misalnya, 'belok kanan', 'maju', 'mundur'). Keberhasilan FSA dalam mencocokkan pola akan menentukan perintah mana yang akan dieksekusi oleh sistem, menjadikannya jembatan antara input suara pengguna dan aksi fisik kursi roda.

### **3.2.2      Tahapan *Process***

Tahapan Proses merupakan jantung operasional sistem, di mana semua logika komputasi dan interpretasi data dilakukan. Seluruh kegiatan pada tahapan ini dieksekusi oleh *Raspberry Pi 3*, sebuah komputer *single-board* yang berfungsi sebagai otak pengendali. Data yang menjadi masukan utama pada tahap ini adalah Teks Kalimat yang merupakan hasil konversi dari *input* suara pengguna pada

Tahapan *Input*. Setelah teks diterima, data tersebut tidak langsung dicocokkan, melainkan akan melalui serangkaian proses *Pre-processing* yang sangat esensial untuk meningkatkan kualitas data dan mengoptimalkan performa pencocokan pola.

Setelah data teks diperoleh, langkah pertama dalam tahap proses adalah *Pre-processing*. Fase ini sangat krusial untuk meningkatkan kualitas data teks dan memastikan efisiensi serta akurasi pada tahap pencocokan pola. Dalam penelitian ini, teknik *pre-processing* yang diterapkan meliputi dua metode utama:

1. *Stemming*: Metode ini bertujuan untuk mereduksi setiap kata dalam teks perintah menjadi bentuk dasar atau akar katanya. Implementasi *stemming* akan menghilangkan imbuhan (*prefix*, *suffix*, *confix*) yang sering kali terdapat pada kata-kata Bahasa Indonesia, sehingga kata-kata dengan makna dasar yang sama namun memiliki bentuk morfologis yang berbeda dapat dikenali sebagai entitas tunggal. Ini sangat membantu dalam standardisasi *input* teks.
2. *Stopword Removal*: Teknik ini melibatkan penghapusan kata-kata yang dianggap sebagai '*stopwords*' dari teks. *Stopwords* adalah kata-kata umum yang sering muncul dalam bahasa namun tidak memberikan kontribusi makna yang signifikan terhadap inti perintah (misalnya, 'yang', 'dan', 'adalah', 'untuk'). Penghapusan *stopwords* akan mengurangi dimensi data dan fokus pada kata kunci yang relevan, sehingga mempercepat proses dan mengurangi *noise*.

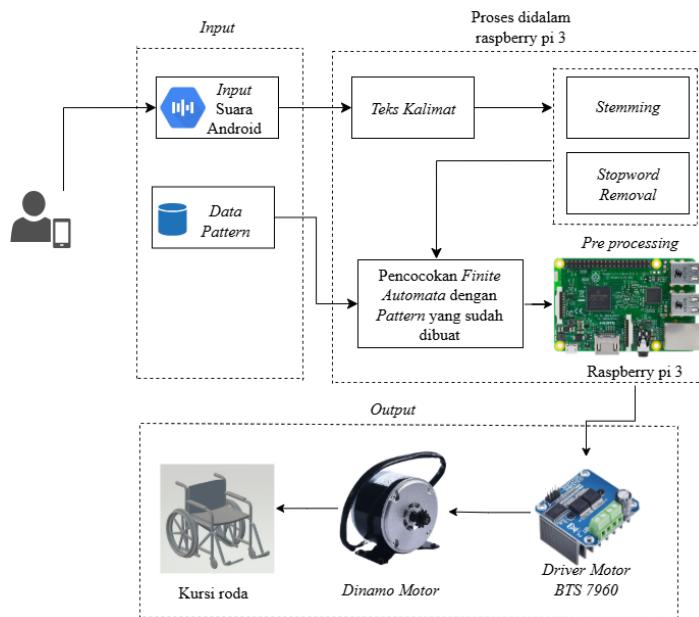
Setelah data teks telah bersih dan terstandardisasi melalui *pre-processing*, langkah berikutnya yang fundamental adalah Proses Pencocokan Pola (*Pattern*

*Matching*). Untuk tugas ini, sistem akan memanfaatkan metodologi *Finite State Automata* (FSA). FSA dirancang untuk secara efisien membandingkan teks hasil *pre-processing* dengan sekumpulan pola atau '*patterns*' perintah yang telah didefinisikan sebelumnya oleh peneliti. Proses pencocokan ini bertujuan untuk mengidentifikasi perintah yang paling sesuai atau paling mendekati dengan *input* pengguna. Keberhasilan pencocokan pola ini akan secara langsung menentukan 'hasil dari pengujian sistem', yaitu identifikasi perintah kontrol kursi roda yang spesifik (misalnya, 'maju', 'mundur', 'belok kanan', 'belok kiri'), yang kemudian akan diteruskan ke tahap output untuk eksekusi fisik.

### 3.2.3 Tahapan *Output*

Tahapan *Output* merupakan kulminasi dari seluruh proses, di mana hasil interpretasi perintah suara diubah menjadi aksi fisik yang dapat dirasakan oleh pengguna. Dari *Raspberry Pi 3*, sinyal kontrol digital yang merepresentasikan perintah yang telah berhasil diidentifikasi pada Tahapan Proses (misalnya, 'maju', 'belok kanan') akan dikirimkan. Sinyal ini kemudian diterima oleh *Driver Motor BTS 7960*. *Driver motor* ini memainkan peran vital sebagai antarmuka daya, karena sinyal dari *Raspberry Pi* biasanya memiliki tegangan dan arus yang rendah dan tidak cukup kuat untuk langsung menggerakkan motor. *Driver Motor BTS 7960* bertugas untuk memperkuat sinyal kontrol tersebut menjadi daya yang memadai, sekaligus mengatur arah dan kecepatan arus yang mengalir ke motor. Dari *driver motor*, arus listrik yang telah diatur akan dialirkan menuju Dinamo Motor. Dinamo motor ini adalah komponen elektromekanis yang mengubah energi listrik menjadi energi kinetik, yaitu putaran mekanis.

Dalam konteks kursi roda, dinamo motor ini secara langsung terhubung dengan roda-roda penggerak. Akhirnya, pergerakan yang dihasilkan oleh Dinamo Motor akan diterjemahkan menjadi manuver fisik Kursi Roda. Ini mencakup pergerakan maju, mundur, belok kanan, atau belok kiri, yang semuanya secara akurat merefleksikan perintah suara asli yang diberikan oleh pengguna dan telah berhasil diinterpretasikan oleh sistem. Dengan demikian, Tahapan *Output* memastikan bahwa ada korespondensi fungsional yang mulus antara niat verbal pengguna dan eksekusi gerak fisik oleh kursi roda, menegaskan efektivitas sistem dalam memberikan kontrol yang intuitif dan responsif. Desain sistem pada penelitian ini dapat dilihat pada dibawah ini:



Gambar 3. 2 Desain Sistem

### 3.3 *Google Speech Recognition*

Dalam implementasi sistem kontrol kursi roda berbasis suara ini, komponen kunci pada tahap akusisi data *input* adalah pemanfaatan teknologi pengenalan

suara, yang secara spesifik menggunakan *Google Speech Recognition*. *Google Speech Recognition* merupakan salah satu layanan *cloud-based* terkemuka yang disediakan oleh *Google*, yang memiliki kemampuan untuk mengonversi ujaran manusia (audio) menjadi teks tertulis (transkripsi). Teknologi ini beroperasi dengan memanfaatkan model pembelajaran mesin yang telah dilatih pada *dataset audio* dan teks yang sangat besar dan beragam, memungkinkan pengenalan berbagai bahasa, aksen, serta konteks ujaran.

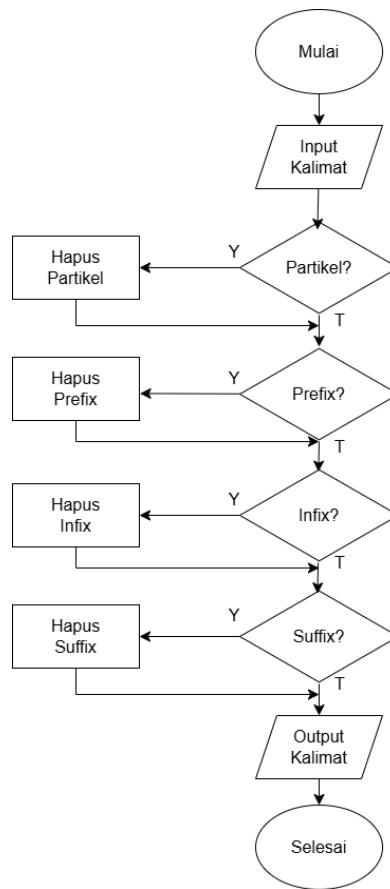
Prinsip kerja *Google Speech Recognition* melibatkan beberapa tahapan kompleks. Pertama, *input audio* dari mikrofon perangkat pengguna (dalam kasus ini, perangkat *Android*) akan ditangkap dan diubah menjadi representasi digital. Data audio digital ini kemudian dikirimkan melalui jaringan yang terhubung di jaringan yang sama ke *server local Rasberry pi 3*. Di server, sinyal audio diproses melalui serangkaian algoritma canggih yang mencakup pemrosesan sinyal suara, segmentasi ujaran, serta pemodelan akustik dan linguistik. Pemodelan akustik bertugas untuk mengenali fonem-fonem (unit bunyi terkecil dalam bahasa) dari sinyal suara, sementara pemodelan linguistik menggunakan pengetahuan tentang tata bahasa, sintaksis, dan semantik untuk membangun urutan kata yang paling mungkin. Untuk aplikasi ini, konfigurasi layanan *Google Speech Recognition* diatur secara spesifik untuk mengenali Bahasa Indonesia, memastikan bahwa perintah-perintah yang diucapkan oleh pengguna dalam bahasa tersebut dapat diinterpretasikan.

### 3.4 *Stemming*

Dalam penelitian ini, *stemming* digunakan sebagai salah satu tahapan dalam *preprocessing* data pada sistem kendali kursi roda berbasis algoritma *Finite State Automata*. *Stemming* adalah proses mengubah kata berimbuhan menjadi bentuk dasar atau akar katanya. Proses ini bertujuan untuk menyederhanakan teks hasil konversi dari suara agar lebih mudah dicocokkan dengan pola perintah yang telah ditentukan.

Sebagai contoh, jika pengguna memberikan perintah suara seperti "majukan kursi roda", maka sistem akan mengubahnya menjadi teks terlebih dahulu. Namun, dalam teks tersebut, kata "majukan" memiliki imbuhan "-kan", sehingga dapat menyebabkan ketidaksesuaian saat dilakukan pencocokan pola (*pattern matching*). Oleh karena itu, melalui proses *stemming*, kata "majukan" akan diubah menjadi bentuk dasarnya, yaitu "maju", sehingga lebih mudah dicocokkan dengan pola yang telah disiapkan dalam algoritma *Finite State Automata* (FSA).

Dalam implementasinya, *stemming* dilakukan dengan menggunakan pustaka (*library*) pemrosesan bahasa alami, seperti Sastrawi untuk bahasa Indonesia. Setelah kata-kata dalam teks dikonversi ke bentuk dasar, sistem akan melanjutkan proses pencocokan pola menggunakan *Finite State Automata* (FSA) untuk menentukan perintah yang sesuai, seperti maju, mundur, belok kanan, atau belok kiri. Proses *Stemming* pada penelitian ini dapat dilihat dalam Gambar 3. 3 dibawah ini:

Gambar 3. 3 Proses *Stemming*

### 3.5 *Stopword Removal*

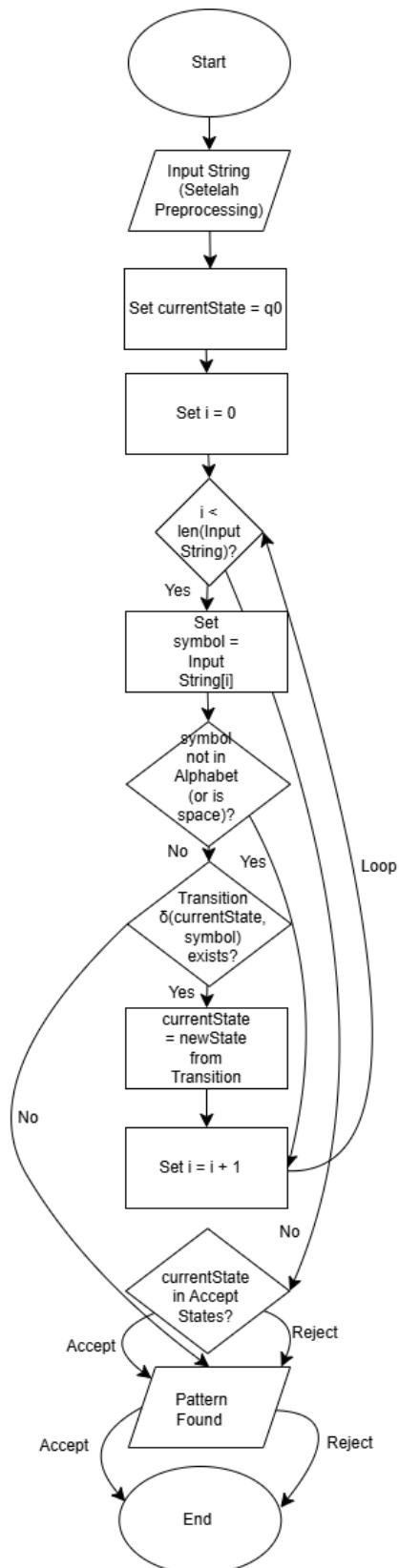
Dalam penelitian ini, *stopword removal* digunakan sebagai salah satu tahapan dalam *preprocessing* data sebelum dilakukan pencocokan pola (*pattern matching*) menggunakan *Finite State Automata* (FSA). *Stopword removal* adalah proses menghilangkan kata-kata yang dianggap tidak memiliki makna signifikan dalam analisis teks, seperti "dan", "yang", "dengan", "adalah", serta kata-kata umum lainnya yang tidak memengaruhi pemahaman makna utama dari perintah pengguna.

Tahapan ini penting karena perintah suara yang diberikan pengguna sering kali mengandung kata-kata tambahan yang tidak diperlukan dalam proses pencocokan pola. Sebagai contoh, jika pengguna memberikan perintah suara seperti "Tolong majukan kursi roda ke depan" maka, setelah proses konversi suara ke teks, kalimat tersebut akan diproses dengan *stopword removal*, sehingga hanya menyisakan kata-kata utama seperti "majukan kursi roda"

Penghapusan kata-kata tidak penting ini bertujuan untuk menyederhanakan teks, mengurangi kompleksitas pemrosesan, serta meningkatkan keakuratan dalam mencocokkan pola perintah menggunakan *Finite State Automata* (FSA). Dengan cara ini, sistem dapat lebih efektif dalam mengenali perintah dan mengonversinya ke dalam aksi yang sesuai, seperti maju, mundur, belok kanan, atau belok kiri.

### 3.6 Algoritma *Finite State Automata*

Algoritma *Finite State Automata* (FSA) adalah salah satu metode yang efektif untuk melakukan pencocokan *string* dengan cara merepresentasikan pola dalam bentuk *state* diagram. Setiap *state* merepresentasikan kemajuan pencocokan pola hingga selesai, dengan setiap *input* karakter memindahkan *state* sistem ke *state* berikutnya. Dalam penelitian ini, algoritma *Finite State Automata* (FSA) digunakan untuk mencocokkan perintah suara yang telah dikonversi menjadi teks dengan pola-pola yang telah ditentukan untuk mengendalikan kursi roda. dibawah merupakan *flowchart* mengenai cara kerja pada *Finite State Automata* (FSA):

Gambar 3. 4 Cara Kerja *Finite State Automata*

*Flowchart* tersebut merepresentasikan logika yang dieksekusi oleh *processInput()* untuk menentukan apakah sebuah *string input* (yang telah diproses sebelumnya) sesuai dengan *pattern* perintah yang telah didefinisikan. Proses ini tidak hanya mencocokkan *pattern* kata kunci, tetapi juga dirancang khusus untuk mengabaikan sisa *input* yang tidak relevan setelah kata kunci ditemukan, berkat implementasi *self-loop* pada *state* akhir.

### 3.6.1 Inisialisasi dan Persiapan Input

Proses pengenalan pola dimulai dengan mengambil *string input* yang telah melalui tahap *preprocessing* (*stopword removal* dan *stemming*), misalnya "maju" atau "maju kursi".

1. *Input String*: Mesin menerima *string input* yang bersih (\$clean\_stemmed).
2. *Reset State*: Langkah pertama di dalam algoritma adalah mengatur ulang *state* mesin. *currentState* diatur ke *state* awal, yaitu q0 (\$this->reset()).
3. Inisialisasi *Index*: Sebuah *index* (variabel i) diatur ke 0 untuk memulai pembacaan karakter dari awal *string*.

### 3.6.2 Proses Iterasi dan Transisi State

Setelah inisialisasi, mesin akan melakukan iterasi sekuensial (karakter demi karakter) melalui keseluruhan *string input*.

1. Pengecekan Karakter (*Loop*): Mesin akan terus berulang selama *index* i masih lebih kecil dari panjang total *string input* (i < len(Input String)?).
2. Pembacaan Simbol: Mesin membaca satu karakter (*symbol*) pada posisi i.

3. Filter Alfabet (*Skip Logic*): Sebelum memproses transisi, *flowchart* menunjukkan keputusan *symbol in alphabet?*. Ini adalah implementasi dari logika kode if (!in\_array(...)) yang berfungsi untuk melewatkkan (skip) karakter apa pun yang tidak relevan (seperti tanda baca atau simbol acak) dengan langsung melanjutkan ke iterasi berikutnya (langkah Set  $i = i + 1$ ).
4. Pengecekan Transisi (Inti FSA): Jika simbol *valid*, mesin akan memeriksa Transition  $\delta(\text{currentState}, \text{symbol}) \text{ exists?}$ . Ini adalah inti dari FSA, di mana mesin memeriksa tabel transisi (\$this->transitions) untuk menentukan apakah ada aturan perpindahan dari *state* saat ini (*currentState*) menggunakan simbol yang sedang dibaca.
  - a. Jika Tidak Ada (*No/Reject*): Jika tidak ada aturan transisi yang ditemukan (misalnya, *state* saat ini  $q_0$  (awal) tetapi simbol yang dibaca adalah 'x'), itu berarti *string input* tidak sesuai dengan pola yang diharapkan. Algoritma segera berhenti dan mengembalikan *Pattern Not Found (Reject)* atau *false*.
  - b. Jika Ada (*Yes/Accept*): Jika transisi ditemukan, *currentState* diperbarui ke *state* baru (*newState*) sesuai dengan aturan transisi tersebut.

### 3.6.3 Logika *Self-Loop* dan Penyelesaian

Setelah inisialisasi, mesin akan melakukan iterasi sekuensial (karakter demi karakter) melalui keseluruhan *string input*. Bagian terpenting dari desain FSA ini adalah cara menangani *input* setelah *pattern* utama ditemukan (misalnya, kata "kursi" setelah "maju").

1. Ketika *pattern* utama (misal, "m-a-j-u") selesai diproses, *currentState* akan mencapai *Final State* (misal, \$q\_4\$).
2. Pada saat pembangunan FSA (bukan di *flowchart*), *Final State* ini telah dilengkapi dengan transisi *self-loop* untuk semua karakter dalam alfabet (misal,  $\delta(q_4, ' ') \rightarrow q_4$ ;  $\delta(q_4, 'k') \rightarrow q_4$ ; dst.).
3. Oleh karena itu, ketika *flowchart* berada di *Final State* dan memproses sisa *string* (" kursi"), pengecekan *Transition*  $\delta(currentState, symbol) \ exists?$  akan selalu Yes. *currentState* akan diperbarui, tetapi kembali ke dirinya sendiri (*currentState = currentState*).
4. Ini memastikan mesin tetap berada di *Final State* dan tidak me-*Reject string* hanya karena ada sisa *input*.

#### 3.6.4 Penentuan Hasil Akhir

Setelah *loop* selesai (panah No dari  $i < \text{len}(\text{Input String})?$ ), algoritma akan melakukan pengecekan akhir:

1. Pengecekan *Accept State*: Mesin memeriksa apakah *currentState* saat ini berada di dalam daftar *Accept States* (*currentState in Accept States?*).
  2. Hasil:
    - a. Jika Yes: Ini berarti *string input* berhasil dicocokkan dengan pola dari awal hingga akhir (atau sisa *input* diabaikan oleh *self-loop*).
- Mesin mengembalikan *Pattern Found* (Accept) atau *true*.

- b. Jika No: Ini berarti *string input* telah selesai dibaca tetapi mesin tidak berakhir di *state* penerima. Mesin mengembalikan *Pattern Not Found (Reject)* atau *false*.

Berikut adalah contoh kasus pencocokan pola menggunakan algoritma *Finite Automata* menggunakan teks “MAJUKAN KURSI RODA” dan *pattern* “MAJU” pada Tabel 3. 1:

Tabel 3. 1 Pencocokan *Finite State Automata*

Index	Karakter	State Sebelumnya	Valid Alfabet	State Berikutnya	Keterangan
0	M	q0	Yes	q1	Transisi berhasil
1	A	q1	Yes	q2	Transisi berhasil
2	J	q2	Yes	q3	Transisi berhasil
3	U	q3	Yes	q4	Accept state tercapai
4	K	q4	No	q0	Karakter tidak valid
5	A	q0	No	q0	Karakter tidak valid
6	N	q0	No	q0	Karakter tidak valid
7	K	q0	No	q0	Karakter tidak valid
8	U	q0	No	q0	Karakter tidak valid
9	R	q0	No	q0	Karakter tidak valid
10	S	q0	No	q0	Karakter tidak valid
11	I	q0	No	q0	Karakter tidak valid
12	R	q0	No	q0	Karakter tidak valid
13	O	q0	No	q0	Karakter tidak valid
14	D	q0	No	q0	Karakter tidak valid
15	A	q0	No	q0	Karakter tidak valid

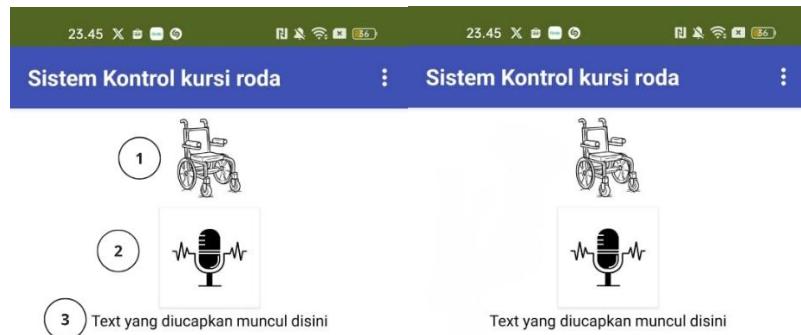
### Keterangan:

Pencocokan pola ditemukan ketika *automaton* mencapai *state* akhir pada indeks ke-3. Setelah pola “MAJU” ditemukan, perintah suara diterjemahkan menjadi tindakan untuk menggerakkan kursi roda maju.

### 3.7 Perancangan Aplikasi

Pencocokan Pada bagian perancangan aplikasi ini bertujuan untuk memberikan kemudahan dalam membangun *system*. Sehingga, setiap tampilan

pada *system* tersebut dapat dipahami fungsi dan tujuannya. Berikut untuk tampilan rancangan aplikasi dapat dilihat pada Gambar 3. 5



Gambar 3. 5 Rancangan Aplikasi

Penjelasan Gambar :

1. Gambar status kursi roda
2. Untuk perintah suara menggunakan *Speech Recognizer*
3. Suara yang diproses menjadi teks

### 3.8 Rangkaian Komponen

Implementasi perangkat keras sistem kontrol kursi roda berbasis suara ini melibatkan integrasi beberapa komponen elektronik esensial yang dirancang untuk bekerja secara sinergis, mulai dari unit pemroses utama hingga aktuator motorik. Komponen sentral yang bertindak sebagai "otak" sistem adalah *Raspberry Pi 3 Model B*. *Raspberry Pi 3* dipilih karena kapabilitas komputasinya yang memadai untuk menjalankan algoritma pemrosesan teks, *pre-processing*, dan *Finite Automata*, serta ketersediaan pin *General Purpose Input/Output* (GPIO) yang fleksibel untuk antarmuka dengan komponen lain.

Sumber daya listrik utama untuk menggerakkan seluruh sistem berasal dari dua unit Aki (Accu) 12 Volt 12 Ah yang dihubungkan secara *paralel*. Konfigurasi paralel ini bertujuan untuk meningkatkan kapasitas arus total (Ah) tanpa mengubah

nilai tegangan nominal (12 Volt), sehingga mampu menyediakan daya yang lebih stabil dan tahan lama untuk semua komponen yang haus daya, khususnya dinamo motor.

Dari sumber daya Aki 12 Volt, tegangan pertama-tama disalurkan ke modul *Step Down Converter LM2596 DC-DC*. Fungsi utama modul ini adalah untuk menurunkan tegangan 12 Volt dari aki menjadi tegangan 5 Volt yang stabil. Tegangan 5 Volt ini sangat krusial karena merupakan tegangan operasi standar untuk *Raspberry Pi 3*. Dengan demikian, *LM2596* memastikan bahwa *Raspberry Pi* menerima pasokan daya yang sesuai dan aman.

Kemudian, untuk mengendalikan pergerakan kursi roda, sistem menggunakan dua buah Dinamo Motor 12 Volt 250 Watt. Motor-motor ini adalah aktuator utama yang bertanggung jawab untuk menggerakkan roda kursi. Untuk mengatur arah dan kecepatan putaran dinamo motor, digunakan modul *Driver Motor BTS 7960*. Modul *BTS 7960* adalah *driver motor H-bridge* berdaya tinggi yang mampu mengelola arus besar yang dibutuhkan oleh dinamo motor 250 Watt, sekaligus melindungi *Raspberry Pi* dari *back-EMF* dan tegangan tinggi.

Interkoneksi antar komponen dilakukan sebagai berikut:

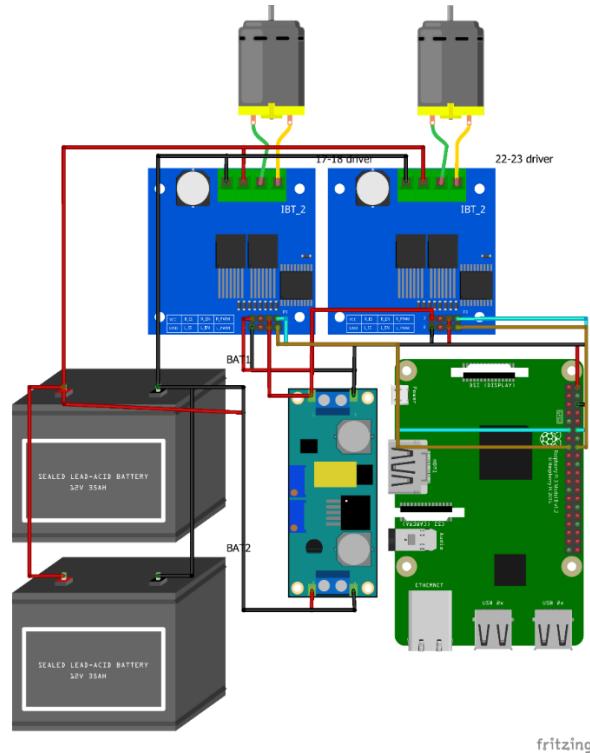
1. Dari Aki ke *LM2596*: Output 12 Volt dari rangkaian paralel aki disambungkan ke input modul *Step Down LM2596*.
2. Dari *LM2596* ke *Raspberry Pi 3*: Output 5 Volt dari *LM2596* disambungkan untuk menyuplai daya ke *Raspberry Pi 3*.
3. Dari *LM2596* ke *Driver Motor BTS 7960*: Tegangan 5 Volt dari *LM2596* juga digunakan untuk menyuplai daya logik *BTS 7960*, dengan

menghubungkan pin *output* 5 Volt *LM2596* ke pin *VCC* dan *GND* pada *Driver Motor BTS 7960*.

4. Kontrol Dinamo Motor dari *Raspberry Pi* ke *BTS 7960*:
  - a. Untuk mengaktifkan dan mengontrol arah putaran dinamo motor kanan, pin RPWM (*Right Pulse Width Modulation*) dan LPWM (*Left Pulse Width Modulation*) pada *Driver Motor BTS 7960* dicolokkan ke pin GPIO 18 dan GPIO 23 pada *Raspberry Pi 3*.
  - b. Sementara itu, untuk mengaktifkan dan mengontrol dinamo motor kiri, pin RPWM dan LPWM pada *Driver Motor BTS 7960* yang terhubung ke motor kiri dicolokkan ke pin GPIO 17 dan GPIO 22 pada *Raspberry Pi 3*. Penggunaan *Pulse Width Modulation (PWM)* melalui pin GPIO ini memungkinkan kontrol kecepatan motor secara presisi.
5. Koneksi Dinamo Motor ke *BTS 7960*: Pin M- dan M+ pada *Driver Motor BTS 7960* disambungkan langsung ke terminal positif (+) dan negatif (-) dari masing-masing dinamo motor.
6. Penyuplai Daya Motor dari Aki ke *BTS 7960*: Tegangan 12 Volt langsung dari rangkaian paralel aki disambungkan ke pin B+ (*Battery Positive*) dan B- (*Battery Negative*) pada *Driver Motor BTS 7960*. Koneksi ini menyediakan daya tinggi yang diperlukan oleh dinamo motor, terpisah dari daya logik.

Rangkaian interkoneksi ini memastikan bahwa *Raspberry Pi 3* dapat menginterpretasikan perintah suara dan mengubahnya menjadi sinyal kontrol

digital, yang kemudian diperkuat dan diterjemahkan oleh *Driver Motor BTS 7960* untuk menggerakkan dinamo motor dan pada akhirnya mengendalikan pergerakan kursi roda secara akurat dan efisien. gambar rangkaian komponen dapat dilihat pada Gambar 3. 6.



Gambar 3. 6 Rancangan Sistem Pada Fritzing

### 3.9 Pengujian Sistem

Pada tahap ini, dilakukan pengujian terhadap sistem untuk mengukur tingkat kesalahan (*error rate*) dalam pengenalan dan pemrosesan perintah suara untuk mengendalikan kursi roda . Pengujian ini bertujuan untuk mengevaluasi seberapa besar tingkat *error* yang terjadi saat sistem menjalankan fungsi-fungsi yang diharapkan, seperti menggerakkan kursi roda maju, mundur, belok kanan, atau belok kiri berdasarkan perintah suara yang diberikan.

Rumus perhitungan rata-rata *error* digunakan untuk menghitung persentase kesalahan yang terjadi selama pengujian sistem (Wardana, 2011), sebagaimana ditunjukkan pada persamaan berikut:

$$\mu = \frac{\sum Ei}{n} \times 100\% \quad (3.2)$$

**Keterangan:**

$\mu$  : Rata-rata *error* dalam bentuk persen (%)

$n$  : Jumlah total percobaan

$\sum Ei$  : Jumlah kesalahan dari semua percobaan

Rumus perhitungan rata-rata *error* digunakan untuk menghitung persentase kesalahan yang terjadi selama pengujian sistem, sebagaimana ditunjukkan pada persamaan berikut. Proses pengujian dilakukan sebanyak 20 kali, dengan merekap hasil keberhasilan atau kegagalan di setiap percobaan. Setiap uji coba dianggap berhasil jika perintah suara dikenali dan dijalankan dengan benar, serta gagal jika terjadi kesalahan dalam pengenalan atau pelaksanaan perintah. Untuk mempermudah perhitungan, setiap keberhasilan diberi nilai 0 dan setiap kegagalan diberi nilai 1.

Rata-rata *error* dihitung dengan membagi jumlah kesalahan yang terjadi dengan total jumlah percobaan, kemudian dikalikan dengan 100% untuk mendapatkan persentase rata-rata *error*. Data ini penting untuk mengetahui tingkat keandalan sistem dan mengidentifikasi area yang perlu perbaikan lebih lanjut.

Tabel 3. 2 Contoh Pengujian *Error* Sistem

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
1.	Majukan kursi roda	Maju	Benar	Maju dengan baik	0
2.	Tolong majukan kursi roda	Maju	Benar	Maju dengan baik	0

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
3.	Kursi roda maju sekarang	Maju	Benar	Maju dengan baik	0
4.	Maju kursinya	Maju	Benar	Maju dengan baik	0
5.	Ayo gerakkan kursi roda maju	Maju	Benar	Maju dengan baik	0
6.	Mundurkan kursi roda	Mundur	Benar	Mundur dengan baik	0
7.	Tolong mundurkan kursi roda	Mundur	Benar	Mundur dengan baik	0
8.	Kursi roda mundur sekarang	Mundur	Benar	Mundur dengan baik	0
9.	Ayo mundurkan kursinya	Mundur	Benar	Mundur dengan baik	0
10.	Gerakkan kursi roda ke belakang	Mundur	Benar	Mundur dengan baik	0
11.	Belokkan kursi roda ke kanan	Belok kanan	Salah	Tidak bergerak	1
12.	Tolong belok ke kanan	Belok kanan	Benar	Membelok kekanan dengan baik	0
13.	Arahkan kursi roda ke kanan	Belok kanan	Salah	Tidak bergerak	1
14.	Kursi roda belok kanan sekarang	Belok kanan	Benar	Membelok kekanan dengan baik	0
15.	Belok kanan pelan-pelan	Belok kanan	Salah	Tidak bergerak	1
16.	Belokkan kursi roda ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0
17.	Tolong belok ke kiri	Belok kiri	Salah	Tidak bergerak	1
18.	Arahkan kursi roda ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0
19.	Kursi roda belok kiri sekarang	Belok kiri	Benar	Membelok kekiri dengan baik	0
20.	Belok kiri pelan-pelan	Belok kiri	Salah	Tidak bergerak	1
$\Sigma Ei$ (Jumlah Kesalahan)					5

Perhitungan Rata-rata *Error*:

$$\mu = \frac{\Sigma Ei}{n} \times 100\%$$

$$\mu = \frac{5}{20} \times 100\% \quad (3.3)$$

$$\mu = 25\%$$

Tabel 3. 2 diatas hasil pengujian menunjukkan bahwa dari 20 percobaan, terdapat 5 kesalahan dalam pengenalan atau pelaksanaan perintah suara. Dengan demikian, rata-rata *error* sistem adalah sebesar 25%. Persentase ini memberikan gambaran tentang seberapa andal sistem dalam mengenali perintah suara, sekaligus menjadi dasar evaluasi untuk meningkatkan performa sistem di masa mendatang.

## **BAB IV**

### **UJI COBA DAN PEMBAHASAN**

#### **4.1 Data Pengujian**

Sistem dibangun melalui dua bagian utama, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*) yang saling terintegrasi. Pada sisi perangkat lunak, implementasi dimulai dari aplikasi *android* yang dibuat menggunakan *MIT APP Inventor*. Aplikasi ini berfungsi untuk menerima perintah suara dari pengguna, kemudian mengirimkannya ke *server local* berbasis PHP yang berjalan di *Raspberry Pi*.

Sementara itu, dari sisi perangkat keras, sistem dikendalikan menggunakan *Raspberry Pi* yang terhubung ke dua motor DC melalui dua modul *driver BTS7960*. Setiap motor dihubungkan ke satu *BTS7960* secara terpisah untuk mendukung sistem kendali dua arah (*differential control*), yaitu kemampuan berbelok ke kanan atau ke kiri. Sumber daya berasal dari dua aki 12V. *Raspberry Pi* juga dilengkapi dengan modul *step-down LM2596* untuk menurunkan tegangan menjadi 5V bagi sistem kontrol. Berikut penjelasan secara rinci aplikasi, konektivitas, serta rangkaian *hardware* yang digunakan:

##### **4.1.1 Sistem Software**

Sistem *software* pada penelitian ini merupakan bagian yang berperan dalam mengolah *input* perintah suara menjadi instruksi kontrol yang dapat dieksekusi oleh perangkat keras. Proses dimulai dari aplikasi *Android* yang dirancang menggunakan *MIT App Inventor*. Aplikasi ini dilengkapi dengan fitur pengenalan

suara berbasis *Google Speech Recognition* untuk menangkap ucapan pengguna dalam bentuk kalimat seperti “tolong majukan kursi roda” atau “belok ke kiri”.



Gambar 4. 1 Halaman Utama

Gambar 4. 1 diatas merupakan halaman utama terdapat beberapa fitur yang ditampilkan. Berikut penjelasan mengenai fitur pada halaman utama:

1. Ikon mikrofon berfungsi untuk mengkonversi perintah suara menjadi teks.
2. Label berfungsi untuk menampilkan teks yang sudah dikonversi.

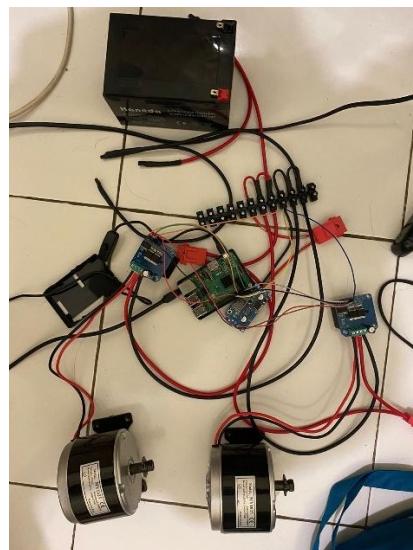
#### 4.1.2 Sistem *Hardware*

Sistem *hardware* dalam penelitian ini merupakan komponen fisik yang berfungsi untuk menggerakkan kursi roda berdasarkan perintah yang dikirim dari sistem *software*. Komponen utama dalam sistem ini adalah *Raspberry Pi*, yang berperan sebagai mikrokontroler pusat. *Raspberry Pi* menerima perintah dari *server*

lokal, lalu mengeksekusinya melalui antarmuka *GPIO (General Purpose Input Output)* untuk mengendalikan dua buah motor DC.

Kedua motor DC ini dihubungkan ke dua buah modul *driver motor* *BTS7960* secara terpisah. Setiap *BTS7960* digunakan untuk mengendalikan satu motor secara independen, sehingga memungkinkan pengaturan arah gerak kursi roda menggunakan prinsip *differential control* yaitu dengan memberi kecepatan atau arah yang berbeda ke motor kiri dan kanan. Misalnya, untuk belok kanan, motor kiri bergerak sedangkan motor kanan berhenti.

Sumber daya sistem ini berasal dari dua buah aki 12V yang disusun secara *paralel*. Tegangan ini digunakan langsung untuk menggerakkan motor DC. Namun, karena *Raspberry Pi* hanya menerima input 5V, digunakan modul *step-down converter* *LM2596* untuk menurunkan tegangan 12V menjadi 5V yang stabil dan aman untuk *Raspberry Pi*. Gambar 4. 2 dibawah merupakan rangkaian *hardware*



Gambar 4. 2 Rangkaian Hardware

Secara keseluruhan, sistem hardware dirancang agar dapat berfungsi sebagai eksekutor dari perintah suara yang dikirimkan oleh pengguna. Kombinasi

antara *driver motor* *BTS7960*, motor DC, *Raspberry Pi*, dan sumber daya dari aki menjadikan sistem ini mampu memberikan respons pergerakan sesuai arah perintah suara seperti maju, mundur, belok kanan, dan belok kiri.

## 4.2 Rangkaian Desain Kursi Roda

Penempatan perangkat keras (*hardware*) pada sistem kursi roda ini dilakukan dengan memperhatikan aspek kemudahan dan keamanan. Komponen utama yang digunakan terdiri dari *Raspberry Pi*, motor driver, baterai, modul relay, serta motor DC penggerak roda. Semua komponen diatur sedemikian rupa agar tidak mengganggu kenyamanan pengguna dan tetap menjaga stabilitas kursi roda saat digunakan.



Gambar 4. 3 Tampak Kursi Roda Dari Belakang

Pada bagian belakang kursi roda, ditempatkan sebuah wadah logam yang berfungsi sebagai kotak kontrol utama. Di dalam wadah tersebut terdapat baterai berkapasitas 12V, *Raspberry Pi* sebagai pusat pengendali sistem, serta *driver motor* yang berfungsi untuk mengatur kecepatan dan arah putaran motor berdasarkan perintah yang diterima dari sistem pengenalan suara. Penempatan seluruh

komponen di dalam kotak ini bertujuan untuk melindungi perangkat dari benturan, debu, serta gangguan lingkungan sekitar. Kabel-kabel penghubung disusun rapi menggunakan jalur kabel agar tidak mengganggu pergerakan kursi roda maupun pengguna.



Gambar 4. 4 Tampak Kursi Roda Dari Depan

Kemudian, dua buah motor DC ditempatkan di bagian kanan dan kiri belakang kursi roda yang terhubung langsung ke roda utama. Motor ini berfungsi sebagai penggerak utama yang memungkinkan kursi roda bergerak maju, mundur, serta berbelok ke kanan dan kiri. Posisi pemasangan motor disesuaikan agar distribusi daya ke roda seimbang dan pergerakan kursi roda tetap stabil. Setiap motor dihubungkan dengan *motor driver* yang dikontrol melalui *Raspberry Pi* menggunakan sinyal digital yang dikirim berdasarkan hasil proses pengenalan perintah suara.



Gambar 4. 5 Tampak Kursi Roda Dari Kanan

Posisi pemasangan motor DC beserta gear dan rantainya diatur agar distribusi tenaga ke kedua roda seimbang, sehingga kursi roda dapat bergerak dengan stabil tanpa condong ke salah satu sisi. Setiap motor dikendalikan melalui *motor driver* yang dihubungkan dengan *Raspberry Pi*, di mana sinyal kendali dikirim berdasarkan hasil pengenalan perintah suara.



Gambar 4. 6 Tampak Kursi Roda Dari Kiri

Selain itu, Komponen ini ditempatkan di area yang mudah dijangkau, namun tetap aman dari sentuhan pengguna. Seluruh jalur kabel dirapikan

menggunakan pelindung kabel agar tidak terinjak atau tersangkut saat kursi roda bergerak.

Secara keseluruhan, penempatan *hardware* ini dirancang agar sistem tetap ringkas, aman, dan efisien, tanpa mengubah fungsi utama kursi roda sebagai alat bantu mobilitas. Penambahan sistem *gear* dan rantai pada kedua sisi roda memungkinkan tenaga motor disalurkan secara optimal, sehingga pergerakan kursi roda menjadi lebih responsif dan bertenaga.

### **4.3 Pengujian Sistem**

Proses pengujian pengenalan kalimat perintah ini dilakukan secara menyeluruh dengan total 40 kali pengetesan 20 kali diuji dengan orang berbobot 65 kg dan 20 kali diuji dengan orang berbobot 58 kg. Seluruh pengujian ini dilaksanakan di lingkungan *indoor* (di dalam ruangan) dengan skenario yang mencakup empat pola pergerakan dasar: Maju, Mundur, Belok Kanan, dan Belok Kiri. Untuk memastikan akurasi dan independensi data pada setiap pengetesan, diterapkan jeda 5 detik antara setiap kali perintah diujikan.

#### **4.3.1 Hasil Pengujian dengan bobot 65 kg**

Proses pengujian pengenalan kalimat perintah dilakukan sebanyak 20 kali yang diuji oleh orang dewasa dengan bobot seberat 65 kg dengan skenario perintah yang berbeda, yang mencakup empat *pattern* pergerakan dasar, yaitu: Maju, Mundur, Belok Kanan, dan Belok Kiri. Setiap pengujian perintah dilakukan jeda selama 5 detik dikarenakan aplikasi akan menangkap perintah *input* secara

*continuously recognition enabled* agar pengguna tidak perlu menekan tombol untuk memberi perintah ke kursi roda. Hasil pengujian dapat dilihat pada dibawah ini:

Tabel 4. 1 Hasil Pengujian Bobot 65kg

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
1	Majukan kursi roda	Maju	Benar	Maju tetapi sedikit serong kekanan	0
2	Tolong majukan kursi roda	Maju	Benar	Maju tetapi sedikit serong kekanan	0
3	Kursi roda maju sekarang	Maju	Benar	Maju tetapi sedikit serong kekanan	0
4	Maju kursinya	Maju	Benar	Maju tetapi sedikit serong kekanan	0
5	Ayo gerakkan kursi roda maju	Maju	Benar	Maju tetapi sedikit serong kekanan	0
6	Mundurkan kursi roda	Mundur	Benar	Mundur tetapi sedikit serong kekanan	0
7	Tolong mundurkan kursi roda	Mundur	Benar	Mundur tetapi sedikit serong kekanan	0
8	Kursi roda mundur sekarang	Mundur	Benar	Mundur tetapi sedikit serong kekanan	0
9	Ayo mundurkan kursinya	Mundur	Benar	Mundur tetapi sedikit serong kekanan	0
10	Gerakkan kursi roda ke belakang	Mundur	Benar	Mundur tetapi sedikit serong kekanan	0
11	Belokkan kursi roda ke kanan	Belok kanan	Benar	Kurang membelok kekanan	0
12	Tolong belok ke kanan	Belok kanan	Benar	Kurang membelok kekanan	0
13	Arahkan kursi roda ke kanan	Belok kanan	Salah	Tidak bergerak	1
14	Kursi roda belok kanan sekarang	Belok kanan	Benar	Kurang membelok kekanan	0
15	Belok kanan pelan-pelan	Belok kanan	Benar	Kurang membelok kekanan	0
16	Belokkan kursi roda ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0
17	Tolong belok ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0
18	Arahkan kursi roda ke kiri	Belok kiri	Salah	Tidak bergerak	1
19	Kursi roda belok kiri sekarang	Belok kiri	Benar	Membelok kekiri dengan baik	0
20	Belok kiri pelan-pelan	Belok kiri	Benar	Membelok kekiri dengan baik	0

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
$\sum Ei$ (Jumlah Kesalahan)					2

Sebagaimana disajikan pada Tabel 4. 1 telah dilakukan pengujian performa sistem sebanyak 20 kali pada bobot 65kg. Pengujian ini mencakup empat jenis perintah, yaitu “Maju”, “Mundur”, “Belok Kanan”, dan “Belok Kiri”. Hasil pengujian menunjukkan bahwa sistem berhasil mengidentifikasi 18 perintah dengan benar dan menghasilkan 2 kesalahan klasifikasi. Kesalahan tersebut terjadi pada proses pencocokan untuk frasa perintah “Arahkan kursi roda ke kanan” dan “Arahkan kursi roda ke kiri”. Tingkat error sistem dari hasil ini akan dihitung menggunakan persamaan berikut.

$$\mu = \frac{\sum Ei}{n} \times 100\% \quad (4. 1)$$

Pada Tabel 4. 1 dapat diketahui bahwa dari 20 percobaan sistem berhasil sebanyak 18 kali dan terdapat kesalahan sebanyak 2 kali. Adapun perhitungan pengujian *error* sistem dengan 2 kali salah adalah sebagai berikut:

$$\begin{aligned} \mu &= \frac{\sum Ei}{n} \times 100\% \\ \mu &= \frac{2}{20} \times 100\% \\ \mu &= 10\% \end{aligned} \quad (4. 2)$$

Dapat diketahui bahwa dari percobaan *Finite State Automata* (FSA) sistem memiliki error sebesar 10%.

### 4.3.2 Hasil Pengujian Dengan Bobot 58kg

Pengujian kedua dilakukan terhadap pengguna dewasa dengan bobot 58 kg, menggunakan skenario yang sama seperti pada pengujian pertama. Proses pengujian dilakukan sebanyak 20 kali dengan variasi kalimat perintah yang berbeda, meliputi empat pola pergerakan utama: Maju, Mundur, Belok Kanan, dan Belok Kiri. Setiap perintah diujikan dengan jeda waktu tertentu untuk memastikan sistem dapat menangkap *input* suara secara kontinu melalui fitur *continuous recognition enabled*. Hasil pengujian dapat dilihat pada table berikut:

Tabel 4. 2 Hasil Pengujian Bobot 58kg

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
1	Majukan kursi roda	Maju	Benar	Maju dengan baik	0
2	Tolong majukan kursi roda	Maju	Benar	Maju dengan baik	0
3	Kursi roda maju sekarang	Maju	Benar	Maju dengan baik	0
4	Maju kursinya	Maju	Benar	Maju dengan baik	0
5	Ayo gerakkan kursi roda maju	Maju	Benar	Maju dengan baik	0
6	Mundurkan kursi roda	Mundur	Benar	Mundur dengan baik	0
7	Tolong mundurkan kursi roda	Mundur	Benar	Mundur dengan baik	0
8	Kursi roda mundur sekarang	Mundur	Benar	Mundur dengan baik	0
9	Ayo mundurkan kursinya	Mundur	Benar	Mundur dengan baik	0
10	Gerakkan kursi roda ke belakang	Mundur	Benar	Mundur dengan baik	0
11	Belokkan kursi roda ke kanan	Belok kanan	Benar	Kurang membelok kekanan	0
12	Tolong belok ke kanan	Belok kanan	Benar	Kurang membelok kekanan	0
13	Arahkan kursi roda ke kanan	Belok kanan	Salah	Tidak bergerak	1
14	Kursi roda belok kanan sekarang	Belok kanan	Benar	Kurang membelok kekanan	0
15	Belok kanan pelan-pelan	Belok kanan	Benar	Kurang membelok kekanan	0
16	Belokkan kursi roda ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0
17	Tolong belok ke kiri	Belok kiri	Benar	Membelok kekiri dengan baik	0

No	Perintah	Pattern	Hasil Pengujian	Keterangan	Jumlah Kesalahan
18	Arahkan kursi roda ke kiri	Belok kiri	Salah	Tidak bergerak	1
19	Kursi roda belok kiri sekarang	Belok kiri	Benar	Membelok kekiri dengan baik	0
20	Belok kiri pelan-pelan	Belok kiri	Benar	Membelok kekiri dengan baik	0
<b><math>\Sigma E_i</math> (Jumlah Kesalahan)</b>					2

Sebagaimana ditunjukkan pada Tabel 4. 2 di atas, pengujian dengan bobot pengguna 58 kg menghasilkan performa yang serupa dengan pengujian sebelumnya. Dari total 20 kali percobaan, sistem mampu mengenali 18 perintah dengan benar dan mengalami 2 kali kesalahan klasifikasi. Kesalahan pengenalan terjadi pada kalimat perintah “Arahkan kursi roda ke kanan” dan “Arahkan kursi roda ke kiri”, di mana pola kata tersebut tidak terdaftar pada basis data pola perintah yang digunakan dalam sistem *Finite State Automata* (FSA). Perhitungan *error* sistem dapat dilakukan menggunakan persamaan berikut:

$$\mu = \frac{\Sigma E_i}{n} \times 100\%$$

$$\mu = \frac{2}{20} \times 100\% \quad (4. 3)$$

$$\mu = 10\%$$

Dapat diketahui bahwa dari percobaan *Finite State Automata* (FSA) sistem memiliki error sebesar 10%.

#### 4.4 Hasil Pengujian Total

Pengujian sistem pengenalan kalimat perintah berbasis *Finite State Automata* (FSA) dilakukan untuk mengevaluasi tingkat keberhasilan sistem dalam mengenali empat pola perintah utama, yaitu Maju, Mundur, Belok Kanan, dan

Belok Kiri. Pengujian dilakukan pada dua pengguna dengan bobot tubuh berbeda, yaitu 65 kg dan 58 kg, masing-masing dengan jumlah percobaan sebanyak 20 kali, sehingga total pengujian sistem berjumlah 40 kali percobaan. Setiap pola perintah diujikan sebanyak 10 kali, dengan variasi kalimat yang berbeda namun memiliki makna perintah yang sama. Berikut merupakan hasil pengujian sistem secara keseluruhan:

Tabel 4. 3 Hasil Pengujian Total

No	Pattern	Diujikan Oleh	Jumlah Uji Coba	Benar	Salah
1.	Maju	Bobot 58 kg dan 65 kg	10 kali	10	0
2.	Mundur	Bobot 58 kg dan 65 kg	10 kali	10	0
3.	Belok Kanan	Bobot 58 kg dan 65 kg	10 kali	8	2
4.	Belok Kiri	Bobot 58 kg dan 65 kg	10 kali	8	2
<b>Total Uji Coba</b>			<b>40 kali</b>	<b>36</b>	<b>4</b>

Berdasarkan tabel Tabel 4. 3 di atas, dapat dilihat bahwa sistem memiliki tingkat keberhasilan yang cukup tinggi dalam mengenali perintah suara. Untuk perintah Maju dan Mundur, sistem mampu mengenali seluruh kalimat dengan benar sebanyak 10 dari 10 percobaan. Hal ini menunjukkan bahwa pola kalimat yang digunakan untuk kedua perintah tersebut telah sesuai dengan pola yang terdaftar dalam sistem FSA, sehingga proses pencocokan berjalan optimal tanpa adanya kesalahan klasifikasi.

Sementara itu, pada perintah Belok Kanan dan Belok Kiri, sistem hanya mampu mengenali 8 dari 10 percobaan dengan benar dan terjadi 2 kesalahan pada tiap pola perintah tersebut. Kesalahan ini disebabkan oleh variasi frasa yang tidak terdapat dalam basis pola FSA, seperti penggunaan kata “Arahkan kursi roda ke kanan/kiri” yang tidak terdefinisi dalam *pattern* perintah. Akibatnya, sistem tidak

dapat mengenali perintah tersebut dan tidak memberikan respons yang sesuai terhadap instruksi pengguna.

Secara keseluruhan, dari 40 kali pengujian, sistem berhasil mengenali 36 perintah dengan benar dan mengalami 4 kesalahan pengenalan, sehingga dapat dihitung tingkat *error total* sistem sebagai berikut:

$$\begin{aligned}\mu &= \frac{\Sigma E_i}{n} \times 100\% \\ \mu &= \frac{4}{40} \times 100\% \\ \mu &= 10\%\end{aligned}\tag{4.7}$$

Dengan demikian *error rate*-nya sebesar 10%. Hasil ini menunjukkan bahwa sistem sudah mampu bekerja dengan baik dalam mengenali perintah suara dasar untuk menggerakkan kursi roda, meskipun masih diperlukan penyempurnaan dalam penanganan variasi kalimat perintah agar sistem lebih fleksibel dan responsif terhadap berbagai bentuk perintah pengguna.

#### 4.5 Integrasi Islam

Penelitian sistem kendali kursi roda menggunakan algoritma *Finite Automata*, yang dirancang untuk memudahkan mobilitas penyandang disabilitas, secara mendalam mencerminkan perintah dan nilai-nilai luhur dalam agama Islam. Prinsip-prinsip ini menjadi landasan filosofis dan etis bagi pengembangan teknologi yang berorientasi pada kemaslahatan umat. Integrasi nilai-nilai ini diperkuat melalui dua landasan utama dalam Al-Qur'an.

#### 4.5.1 Landasan Etis: Prinsip Keseimbangan Duniawi dan Ihsan

Pilar pertama integrasi Islam adalah prinsip keseimbangan hidup dan perintah berbuat baik kepada sesama. Hal ini sejalan dengan firman Allah SWT dalam Surah Al-Qasas ayat 77:

وَابْتَغِ فِيمَا أَنْتَكَ اللَّهُ الدَّارُ الْآخِرَةَ وَلَا تَنْسِ نَصِيبِكَ مِنَ الدُّنْيَا وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ وَلَا تَبْغِ  
الْفَسَادَ فِي الْأَرْضِ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ

*“Dan, carilah pada apa yang telah dianugerahkan Allah kepadamu (pahala) negeri akhirat, tetapi janganlah kamu lupakan bagianmu di dunia. Berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik kepadamu dan janganlah kamu berbuat kerusakan di bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.” (Q.S Al-Qashash 28:77)*

Para mufasir, seperti Imam Ibnu Katsir dalam *Tafsir Al-Qur'an Al-'Azhim* dan Imam Al-Qurtubi dalam *Al-Jami' li Ahkam Al-Qur'an*, menjelaskan bahwa ayat ini mengandung perintah fundamental untuk mencapai keseimbangan antara kehidupan akhirat dan dunia. Frasa "janganlah kamu lupakan bahagianmu dari (kenikmatan) duniawi" diinterpretasikan sebagai anjuran untuk memanfaatkan segala potensi, anugerah, dan ilmu pengetahuan yang diberikan Allah termasuk kemampuan berpikir dan inovasi guna mencapai kemaslahatan di dunia. Selain itu, perintah eksplisit "berbuat baiklah (*ahsin*) kepada orang lain sebagaimana Allah telah berbuat baik, kepadamu" menjadi landasan etika universal yang mendorong umat Muslim untuk memberikan manfaat dan kebaikan bagi sesama.(ilham, 2024)

Prinsip keseimbangan dan perintah *ihsan* (berbuat baik) ini sangat relevan dengan pengembangan sistem kendali kursi roda otomatis. Proyek ini merupakan wujud nyata dari upaya berbuat baik dan memanfaatkan bahagian duniawi (ilmu pengetahuan dan teknologi). Dengan menyediakan solusi yang memudahkan

mobilitas penyandang disabilitas, penelitian ini berkontribusi pada peningkatan kualitas hidup dan kemaslahatan umat manusia. Pengembangan sistem kontrol kursi roda menggunakan algoritma *Finite State Automata* (FSA) melibatkan pemanfaatan teknologi modern. Kecanggihan sistem ini, dengan memanfaatkan teknologi terkini, tentu tidak terlepas dari kebesaran dan kuasa Allah SWT yang telah menganugerahkan akal dan kemampuan berpikir kepada manusia. Dalam konteks Surah Al-Qasas ayat 77, pengembangan teknologi ini dapat dipahami sebagai salah satu bentuk upaya manusia untuk merealisasikan perintah berbuat baik (*ihsan*) dan memenuhi anjuran untuk tidak melupakan bahagian dunia dalam rangka menciptakan solusi yang meringankan beban dan memberikan kemudahan bagi mereka yang membutuhkan. Ini adalah manifestasi dari kemampuan manusia untuk berinovasi dan menciptakan solusi untuk mengatasi keterbatasan fisik, seolah-olah "membantu" mereka bergerak dengan "kekuatan" ilmu dan teknologi yang diilhamkan Allah, memastikan bahwa teknologi diarahkan untuk kemaslahatan umat manusia.

#### **4.5.2 Landasan Filosofis: Prinsip Penundukan Alam dan Dorongan Berpikir**

Pilar kedua adalah landasan filosofis yang memberikan kerangka berpikir bagi penelitian dan pengembangan teknologi. Pilar ini menegaskan bahwa seluruh ciptaan Allah telah ditundukkan untuk dimanfaatkan manusia melalui akal. Hal ini ditegaskan dalam Surah Al-Jatsiyah ayat 13:

وَسَحَّرَ لَكُمْ مَا فِي السَّمَاوَاتِ وَمَا فِي الْأَرْضِ جَمِيعًا مِنْهُ، إِنَّ فِي ذَلِكَ لَعَائِدَاتٍ لِّقَوْمٍ يَتَفَكَّرُونَ

*“Dan Dia menundukkan untukmu apa yang ada di langit dan apa yang ada di bumi semuanya, (sebagai rahmat) daripada-Nya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kekuasaan Allah) bagi kaum yang berpikir.”* (Q.S Al-Jatsiyah 45:13)

Para mufasir, seperti Imam Fakhruddin Ar-Razi dalam *Mafatih al-Ghaib* dan Imam Ibnu Katsir dalam *Tafsir Al-Qur'an Al-'Azhim*, menjelaskan bahwa ayat ini mengandung makna filosofis yang sangat mendalam terkait peran manusia di muka bumi dan anugerah Allah SWT. Frasa "Dia menundukkan untukmu apa yang ada di langit dan apa yang ada di bumi semuanya" mengindikasikan bahwa segala potensi alam semesta, baik yang terlihat maupun yang tidak, diciptakan dan dijadikan mudah untuk dimanfaatkan oleh manusia. Ini bukan sekadar penaklukan fisik, tetapi juga anugerah akal dan kemampuan untuk memahami, meneliti, dan mengelola sumber daya alam serta hukum-hukumnya. Tujuan utama dari "penundukan" ini adalah agar manusia dapat mengambil pelajaran dan menemukan "tanda-tanda kekuasaan Allah" bagi kaum yang berpikir (*li qaumin yatafakkarun*). Ini adalah dorongan eksplisit dalam Al-Qur'an untuk terus menggunakan akal, melakukan observasi, eksperimen, dan inovasi. (*Al-Jasiyah Ayat 13 - Bahasa Indonesia*, n.d.)

Prinsip dasar penundukan alam semesta dan dorongan untuk berpikir ini sangat relevan dengan pengembangan sistem kendali kursi roda otomatis yang memanfaatkan teknologi informasi. Proyek ini merupakan manifestasi nyata dari perintah Ilahi untuk memanfaatkan anugerah akal dan potensi alam. Manusia, dengan kecerdasannya, diberi kemampuan untuk memahami prinsip-prinsip fisika, elektronika, dan ilmu komputasi, yang kemudian diwujudkan dalam bentuk teknologi informasi yang kompleks. Penggunaan algoritma *Finite State Automata*

(FSA), integrasi *Google Speech Recognition*, serta rancangan perangkat keras seperti *Raspberry Pi* dan *driver motor*, adalah bentuk aplikasi ilmu pengetahuan yang memungkinkan kita "menundukkan" prinsip-prinsip alam dan kecanggihan teknologi untuk mengatasi keterbatasan fisik. Melalui inovasi ini, manusia tidak hanya memecahkan masalah praktis mobilitas, tetapi juga memberikan solusi yang meningkatkan kemandirian dan kualitas hidup penyandang disabilitas, yang merupakan wujud nyata dari pemanfaatan rahmat Allah SWT.

Lebih jauh, pengembangan teknologi informasi dalam konteks ini tidak hanya berhenti pada fungsi praktis semata, tetapi juga mengembangkan nilai spiritual. Setiap *line of code*, setiap komponen elektronik yang terhubung, dan setiap fungsi yang bekerja, adalah hasil dari proses berpikir, merancang, dan menciptakan yang diilhamkan oleh anugerah akal dari Allah SWT. Ini adalah bentuk *tadabbur* (perenungan mendalam) dan *tafakkur* (pemikiran) terhadap ciptaan-Nya. Oleh karena itu, teknologi informasi seharusnya tidak hanya dilihat sebagai alat netral, tetapi sebagai sarana untuk mewujudkan kemaslahatan, keadilan, dan kasih sayang, terutama bagi mereka yang membutuhkan. Dengan mengintegrasikan nilai-nilai luhur Islam, pengembangan teknologi seperti kursi roda ini menjadi sebuah ibadah dan upaya untuk lebih mendekatkan diri kepada Allah SWT dengan memanfaatkan anugerah yang telah diberikan-Nya untuk kebaikan seluruh umat manusia.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Penelitian ini telah berhasil merancang dan mengimplementasikan sistem kendali kursi roda menggunakan algoritma *Finite State Automata* (FSA), dengan tujuan utama meningkatkan kemandirian mobilitas penyandang disabilitas fisik melalui perintah suara. Implementasi sistem ini mencakup integrasi komponen perangkat lunak seperti aplikasi *Android* yang dikembangkan dengan *MIT App Inventor*, *Google Speech Recognition* untuk konversi suara ke teks, serta proses *preprocessing* teks (*stemming* dengan Sastrawi dan *stopword removal*) dan eksekusi algoritma FSA yang semuanya berjalan pada *Raspberry Pi 3* sebagai mikrokontroler utama. Sementara itu, komponen perangkat keras seperti *driver motor BTS7960*, motor DC, dan aki 12V berfungsi sebagai aktuator yang menerjemahkan perintah digital menjadi gerakan fisik. Hasil pengujian menunjukkan bahwa sistem ini memiliki *error rate* pengenalan perintah sebesar 10% dalam lingkungan terbatas, membuktikan efektivitas algoritma FSA dalam memproses pola perintah suara.

#### **5.2 Saran**

Berdasarkan hasil dan keterbatasan penelitian ini, terdapat beberapa saran untuk pengembangan sistem lebih lanjut di masa mendatang:

1. Ekspansi Pengujian Lingkungan Uji coba sistem perlu dilakukan pada lingkungan luar ruangan (*outdoor*) dengan mempertimbangkan faktor

eksternal seperti kebisingan yang lebih tinggi, variasi kondisi permukaan tanah (tidak rata), dan faktor cuaca. Hal ini akan menguji ketahanan dan keandalan sistem dalam skenario penggunaan nyata.

2. Integrasi Fitur Tambahan Pengembangan dapat mencakup penambahan fitur kritis seperti sensor atau navigasi otomatis untuk meningkatkan keamanan dan fungsionalitas kursi roda, seperti yang telah dibatasi dalam penelitian

## DAFTAR PUSTAKA

- Al-Jasiyah Ayat 13—Bahasa Indonesia. (n.d.). Retrieved November 17, 2025, from <https://qurano.com/id/45-al-jasiyah/ayat-13/>
- Anwar, M. S., Subroto, I. M. I., & Mulyono, S. (2020). Sistem Pencarian E-Jurnal Menggunakan Metode Stopword Removal dan Stemming Berbasis Android. <https://api.semanticscholar.org/CorpusID:219649684>
- Bugade, V. S., Chougale, P., Katkar, P. K., Lavhate, S., & Bhosale, A. (2024). BTS7960 Motor Drive: Precision Speed Modulation for PMDC Motors. 2024 3rd International Conference for Advancement in Technology (ICONAT), 1–5.
- Ginting, M. D. (2024). Implementasi Finite State Automata (FSA) Pada Aplikasi Simulasi Vending Machine Minuman Boba. *Scientica: Jurnal Ilmiah Sains Dan Teknologi*, 3(2), 65–73.
- Gulo, Y. N. (2022). Penerapan Algoritma Hamming Distance Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Indonesia. *Journal Global Technology Computer*. <https://api.semanticscholar.org/CorpusID:266201696>
- Hidayat, S., Said, F., Titiani, F., & Gata, W. (2021). Desain Konsep Finite State Automata (Fsa) Pada Simulasi Vending Machine (Vm) Masakan Padang. *Journal of Information System, Informatics and Computing*, 5(1), 134–143.
- Huda, D. N., Zahara, A., & Hardinata, D. (2023). Pemanfaatan Konsep Finite State Automata Pada Sistem Perparkiran Kendaraan Bermotor Bandara Raja Haji Fisabilillah Tanjungpinang. *Jurnal Bangkit Indonesia*, 12(2), 34–39.
- Ikhsan, M., Supiyandi, S., & Syuhada, M. F. (2023). Implementation of String Matching Algorithm with Finite Automata in The Indonesian-Korean Dictionary Application. *Sainstek : Jurnal Sains Dan Teknologi*. <https://api.semanticscholar.org/CorpusID:266804953>
- ilham. (2024, December 21). Menerapkan Etos Ihsan dalam Setiap Sendi Kehidupan. *Muhammadiyah*. <https://muhammadiyah.or.id/2024/12/menerapkan-etos-ihsan-dalam-setiap-sendi-kehidupan/>
- Kaddipujar, M., Rajan, J., & B.D., K. (2022). Mobile Application Development Using MIT App Inventor: An Experiment at Raman Research Institute Library. *Bulletin of the AAS*. <https://doi.org/10.3847/25c2cfeb.d68a2a42>

- Kurniawan, S. Y., Setiawan, A. B., & Dirgantara, W. (2022). Kursi Roda Otomatis Berbasis IoT (Internet Of Things) Menggunakan Metode PID (Proportional Integral Derivative Controller). *SinarFe7*, 5(1), 79–84.
- Kusuma, M. R., Gata, W., Kurniawati, L., Friyadie, F., & Baihaqi, A. (2022). Penerapan Finite State Automata Pada Validasi Permohonan Pengajuan Layanan Akun Digital Menara Masjid. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 12(1), 73–87.
- Mardianto, E., Tarkan, R., & others. (2024). Perancangan Dan Implementasi Sistem Kendali Kursi Roda Berbasis Mikrokontroler: Perancangan Dan Implementasi Sistem Kendali Kursi Roda Berbasis Mikrokontroler. *Electrical Network Systems and Sources*, 3(2), 22–29.
- Meiyanti, R., & Sandy, C. (2024). Implementation of Finite State Automata Method in Text to Speech Conversion System. *Journal of Advanced Computer Knowledge and Algorithms*, 1, 59. <https://doi.org/10.29103/jacka.v1i3.16917>
- Putri, M., Jannah, M., Fitriati, I., & Ridwan, R. (2024). Effectiveness of Using Mit App Inventor at STKIP Taman Siswa Bima. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, 12, 351. <https://doi.org/10.26418/justin.v12i2.76628>
- Salam, A. (2021, September 10). Tafsir Surah Al-Qasas Ayat 77: Akhirat Harus, Dunia Jangan Dilupakan. *Tafsir Al Quran | Referensi Tafsir di Indonesia*. <https://tafsiralquran.id/tafsir-surah-al-qasas-ayat-77-akhirat-harus-dunia-jangan-dilupakan/>
- Wardana, M. M. J. (2011). Pengenalan Suara Menggunakan Algoritma Fast Fourier Transform (FFT) Dengan Algoritma Mel Frequency Cepstrum Coeffisient (MFCC) Sebagai Ekstrasi Ciri.