

**AKTIVASI TV BERBASIS *MULTI PATTERN MATCHING*
MENGUNAKAN *WU-MANBER***

SKRIPSI

Oleh:
MUH FAHMA SAHMURA HABIB
NIM. 210605110152



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**AKTIVASI TV BERBASIS *MULTI PATTERN MATCHING*
MENGUNAKAN *WU-MANBER***

SKRIPSI

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

Oleh :
MUH FAHMA SAHMURA HABIB
NIM. 210605110152

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

AKTIVASI TV BERBASIS *MULTI PATTERN MATCHING* MENGUNAKAN *WU-MANBER*

SKRIPSI

Oleh :
MUH FAHMA SAHMURA HABIB
NIM. 210605110152

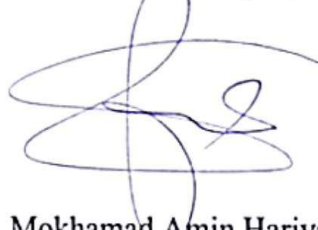
Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 19 Juni 2026

Pembimbing I,



Ajib Hanani, M.T
NIP. 19840731 202321 1 013

Pembimbing II,



Dr. Mokhamad Amin Hariyadi, M.T.
NIP. 19670018 200501 1 001

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

AKTIVASI TV BERBASIS *MULTI PATTERN MATCHING* MENGUNAKAN *WU-MANBER*

SKRIPSI

Oleh :
MUH FAHMA SAHMURA HABIB
NIM. 210605110152

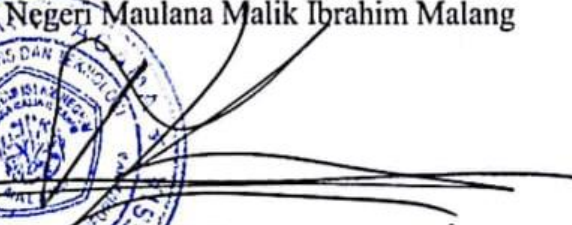
Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 22 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. Ir. Fresy Nugroho, ST., MT, IPM</u> NIP. 19710722 201101 1 001	()
Anggota Penguji I	: <u>Shoffin Nahwa Utama, M.T</u> NIP. 19860703 202012 1 003	()
Anggota Penguji II	: <u>Ajib Hanani, M.T</u> NIP. 19840731 202321 1 013	()
Anggota Penguji III	: <u>Dr. Mokhamad Amin Hariyadi, M.T.</u> NIP. 19670018 200501 1 001	()

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 : MUH FAHMA SAHMURA HABIB
NIM : 210605110152
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : AKTIVASI TV BERBASIS *MULTI PATTERN MATCHING* MENGGUNAKAN *WU-MANBER*

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 Juni 2026

Yang membuat pernyataan,



MUH FAHMA SAHMURA HABIB
NIM. 210605110152

MOTTO

... TUA ITU PASTI DEWASA ITU PILIHAN ...

HALAMAN PERSEMBAHAN

Alhamdulillahirobil Alamin puji syukur atas kehadiran Allah SWT yang telah memberikan segala nikmat dan rahmat-Nya, karena penulis dapat menyelesaikan penelitian ini dengan lancar. Tak lupa Sholawat serta salam senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW yang kita harapkan syafaat beliau di yaumul qiyamah nanti, amin. Penulisan tugas akhir ini dapat terselesaikan dengan lancar karena adanya dukungan dari berbagai pihak. Halaman persembahan ini ditulis sebagai bentuk rasa syukur dan terimakasih penulis kepada semua pihak yang terlibat dalam penelitian ini. Alhamdulillah penulis dapat menyelesaikan penelitian ini walau masih banyak kekurangan. Pencapaian penelitian ini dipersembahkan dengan hati yang senang dan bahagia oleh penulis kepada:

1. Kedua orang tua yang senantiasa memberi kasih sayang, cinta dan doa yang tak terhingga sehingga penulis dapat menyelesaikan penelitian ini. Kemudian, kepada adik penulis yang juga selalu memberikan semangat dan motivasi. Mereka adalah kekuatan bagi penulis dalam proses penyelesaian penelitian ini.
2. Bapak Ajib Hanani, M.T selaku dosen pembimbing I dan Bapak Dr. Mokhammad Amin Hariyadi, M.T selaku dosen pembimbing II yang selalu memberi bimbingan dan arahan dalam membantu penulis menyelesaikan penelitian ini. Kepada Bapak Dr. Ir. Fresy Nugroho, ST., MT, IPM selaku dosen penguji I dan Bapak Shoffin Nahwa Utama, M.T selaku dosen penguji II yang selalu memberi bimbingan, arahan,

masuk, dan perbaikan untuk membantu penulis melengkapi penelitian ini. Tak lupa kepada seluruh Dosen dan Staff Program Studi Teknik Informatika atas ilmu, pengalaman, dan kesempatan belajar selama masa studi penulis dalam masa perkuliahan dan penelitian ini.

3. Keluarga besar Teknik Informatika angkatan 2021 yang telah menemani, kebersamaan, memberikan semangat kepada penulis. Kepada seluruh keluarga, teman serta saudara, terimakasih banyak atas segala kebajikan kepada penulis.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobil Alamin puji syukur atas kehadiran Allah SWT yang telah memberikan segala nikmat dan rahmat-Nya, karena penulis dapat menyelesaikan penelitian ini dengan lancar. Tak lupa Sholawat serta salam senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW yang kita harapkan syafaat beliau di yaumul qiyamah nanti, amin.

Penulis mengucapkan terima kasih banyak kepada seluruh pihak yang terlibat, yang telah memberikan semangat dan bantuannya kepada penulis sehingga penelitian ini dapat selesai dengan lancar. Terimakasih banyak atas dukungannya, yang mana penulis tidak akan melupakan kebaikan kalian semua dan juga penulis tidak dapat membalas satu-persatu, karena penulis hanya bisa mendoakan yang terbaik untuk kalian semua yang terlibat. Untuk itu penulis ingin mengucapkan terima kasih banyak kepada:

1. Prof. Dr. Hj. Ilfi Nurdiana, M.Si., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes., 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 dosen pembimbing I yang selalu memberi bimbingan dan arahan dalam membantu penulis menyelesaikan penelitian ini.

5. Dr. Mokhamad Amin Hariyadi, M.T selaku dosen pembimbing II yang selalu memberi bimbingan dan arahan dalam membantu penulis menyelesaikan penelitian ini.
6. Dr. Ir. Fresy Nugroho, ST., MT, IPM selaku dosen penguji I sekaligus dosen wali penulis yang selalu memberi bimbingan, arahan, masukan, dan perbaikan untuk membantu penulis melengkapi penelitian ini.
7. Shoffin Nahwa Utama, M.T selaku dosen penguji II yang selalu memberi bimbingan, arahan, masukan, dan perbaikan untuk membantu penulis melengkapi penelitian ini.
8. Kedua orang tua dan adik yang selalu memberikan yang terbaik untuk penulis. Semoga beliau diberikan kesehatan dan umur panjang.
9. Abah KH. Marzuqi Mustamar dan Umik Saidah Mustaghfiroh yang telah memberikan semangat kepada penulis. Semoga beliau senantiasa diberi kesehatan dan umur yang panjang dalam memndidik.
10. Nia Faricha S, Si selaku admin Program Studi Teknik Informatika yang selalu sabar memberikan informasi, membantu, dan memberikan arahan selama perkuliahan dan proses penulisan skripsi ini.
11. Seluruh dosen dan jajaran staff Program Studi Teknik Informatika yang telah memberikan ilmu, pengalaman, dan kesempatan belajar selama masa studi penulis dalam masa perkuliahan dan penelitian ini.
12. Seluruh keluarga besar civitas UIN MAULANA MALIK IBRAHIM MALANG yang telah memberikan pengalaman dan pelajaran selama masa perkuliahan.

13. Keluarga besar Ma'had Sunan Ampel Al-Aly.
14. Keluarga besar Mabna Ibnu Sina.
15. Semua para guru penulis, semoga beliau semua diberikan keberkahan.
16. Seluruh saudara penulis yang telah memberikan semangat dalam menyelesaikan penelitian ini.
17. Keluarga besar Teknik Informatika angkatan 2021 yang telah kebersamai selama masa perkuliahan.
18. Seluruh teman seperjuangan di kampus tercinta.
19. Keluarga besar Pondok Pesantren Sabilurrasyad gasek malang.
20. Teman-teman yang telah memberikan semangat, motivasi, dan juga bantuan dalam menyelesaikan penelitian ini.
21. Seluruh pihak yang telah terlibat dalam penelitian ini, yang tidak dapat saya sebutkan satu-persatu, yang telah kebersamai penulis selama masa menempuh pendidikan di kota malang.

Penulis berharap semoga penelitian ini dapat memberikan banyak manfaat untuk semuanya. Penulis menyadari bahwa penelitian ini masih banyak kurangnya, oleh karena itu, penulis bersedia menerima berbagai saran dan masukan yang dapat membantu mengembangkan penelitian ini lebih lanjut kedepanya.

Wa'alaikumsalam Warahmatullahi Wabarakatuh

Malang, 19 Juni 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL	xv
ABSTRAK	xvi
ABSTRACT	xvii
البحث مستخلص.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terkait	7
2.2 <i>MIT App Inventor</i>	9
2.3 Televisi (TV)	10
2.4 Algoritma <i>Wu Manber</i>	12
2.5 Raspberry Pi	18
2.6 IR Transmitter	21
2.7 <i>String Matching</i>	23
2.8 <i>Speech recognition</i>	24
2.9 Arduino Uno.....	27
2.10 Buzzer	31
BAB III DESAIN DAN IMPLEMENTASI	33
3.1 Pengumpulan Data	34
3.2 Data Primer.....	34
3.3 Data Sekunder	35
3.4 Desain dan Perancangan Sistem Kontrol	36
3.5 Tahapan <i>Input</i>	36
3.6 <i>Google Speech Reconition</i>	37
3.7 Tahapan Proses.....	37
3.8 <i>Stopword Removal</i>	38
3.9 <i>Stemming</i>	39
3.10 Pencocokan Algoritma <i>WU-MANBER</i>	39
3.11 Contoh Pencocokan Algoritma <i>WU-MANBER</i>	45
3.12 Tahapan Output.....	49

3.13	Perancangan Aplikasi	49
3.14	Perancangan Alat Sistem Kontrol.....	50
3.15	Implementasi Sistem.....	51
3.16	Pengujian Sistem.....	51
3.17	Perhitungan <i>Error</i> Sistem	52
3.18	Hasil Analisa.....	54
BAB IV UJI COBA DAN PEMBAHASAN		55
4.1	Hasil Implementasi Sistem Kontrol	55
4.1.1	Perangkat Lunak	55
4.1.2	Implementasi Komunikasi	56
4.1.3	Implementasi <i>WU-MANBER</i>	58
4.1.4	Perangkat Keras	64
4.2	Hasil Pengujian.....	66
4.2.1	Hasil Pengujian Perintah Suara.....	66
4.2.2	Hasil Pengujian <i>Stopword Removal</i>	68
4.2.3	Hasil Pengujian <i>Stemming</i>	70
4.2.4	Hasil Pengujian <i>WU-MANBER</i>	71
4.2.5	Hasil Pengujian Output Kode Hex.....	75
4.2.6	Hasil Pengujian sistem.....	77
4.5	Integrasi Islam	80
4.5.1	<i>Muamalah Ma'a Allah</i>	80
4.5.2	<i>Muamalah Ma'a An-Nas</i>	81
4.5.3	<i>Muamalah Ma'a Alam</i>	83
BAB V KESIMPULAN DAN SARAN		85
5.1.	Kesimpulan	85
5.2.	saran	86
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 2. 1 Flowchart <i>WU-MANBER</i>	15
Gambar 2. 2 PIN <i>GPIO</i>	20
Gambar 2. 3 Raspberry Pi 3	18
Gambar 2. 4 IR Transmitter	21
Gambar 2. 5 Arduino Uno.....	27
Gambar 2. 6 PIN Arduino Uno	29
Gambar 2. 7 Buzzer.....	31
Gambar 3. 1 Alur Penelitian	34
Gambar 3. 2 Desain Sistem.....	36
Gambar 3. 3 <i>Diagram Activity</i>	40
Gambar 3. 4 Rancangan Aplikasi.....	49
Gambar 3. 5 Rancangan Komponen	50
Gambar 4. 1 Halaman Utama	56
Gambar 4. 2 Rangkaian Hardware	65
Gambar 4. 3 Blok Diagram Pengujian Perintah Suara.....	67
Gambar 4. 4 Blok Diagram Pengujian <i>Stopword Removal</i>	69
Gambar 4. 5 Blok Diagram Pengujian <i>Stemming</i>	71
Gambar 4. 6 Blok Diagram Pengujian <i>Wu-Manber</i>	74
Gambar 4. 7 Blok Diagram Pengujian Kode <i>HEX</i>	76

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Raspberry Pi 3	20
Tabel 2. 2 Spesifikasi IR Transmitter	22
Tabel 2. 3 Spesifikasi Arduino Uno	28
Tabel 2. 4 PIN Analog Arduino Uno	29
Tabel 2. 5 PIN Digital Arduino Uno	30
Tabel 2. 6 Spesifikasi Buzzer	32
Tabel 3. 1 Kode <i>Hexadecimal</i>	35
Tabel 3. 2 contoh pengujian <i>error</i> sistem	53
Tabel 4. 1 Input Teks dan <i>Pattern</i>	58
Tabel 4. 2 Penyederhanaan Kalimat	59
Tabel 4. 3 Menentukan panjang minimum <i>Pattern</i>	60
Tabel 4. 4 Pembentukan tabel <i>SHIFT</i>	60
Tabel 4. 5 Pembentukan tabel nilai <i>SHIFT</i>	61
Tabel 4. 6 Pembentukan tabel <i>HASH</i>	61
Tabel 4. 7 Indeks Karakter	62
Tabel 4. 8 Verifikasi pola	63
Tabel 4. 9 Hasil pengujian perintah suara	66
Tabel 4. 10 Hasil pengujian <i>Stopword Removal</i>	68
Tabel 4. 11 Hasil pengujian <i>Stemming</i>	70
Tabel 4. 12 Pembentukan tabel <i>SHIFT</i>	72
Tabel 4. 13 Hasil pembentukan tabel <i>HASH</i>	72
Tabel 4. 14 Proses Pencarian <i>WU-MANBER</i>	73
Tabel 4. 15 Verifikasi <i>Pattern</i>	73
Tabel 4. 16 Hasil Pengujian Output Kode Hex	75
Tabel 4. 17 Hasil uji coba	77

ABSTRAK

Habib, Muh Fahma Sahmura. 2026. **Aktivasi TV Berbasis *Multi Pattern Matching* Menggunakan *WU-MANBER***. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T (II) Dr. Mokhamad Amin Hariyadi, M.T.

Kata kunci: Raspberry Pi, *Speech recognition*, Sistem Kontrol TV, *WU-MANBER*.

Kemajuan teknologi memberikan kemudahan dalam menjalankan perintah perangkat elektronik. Penelitian ini bertujuan untuk mengimplementasikan metode *WU-MANBER* untuk pencocokan teks pada sistem kontrol TV berbasis perintah suara guna membantu memudahkan pengoperasian TV tanpa sentuhan langsung pada remot. Penelitian ini menggunakan metode rekayasa sistem yang terdiri dari pengumpulan data, perancangan, implementasi, dan pengujian. Input perintah suara direkam melalui aplikasi *Android* yang dibangun dengan *MIT App Inventor*, kemudian dikonversi menjadi teks menggunakan *Google Speech recognition*. Teks tersebut diproses melalui tahap *stopword removal* dan *Stemming* sebelum dicocokkan dengan pola perintah menggunakan algoritma *WU-MANBER* yang berjalan pada Raspberry Pi 3. Hasil pencocokan diteruskan ke Arduino untuk mengirimkan sinyal melalui IR Transmitter guna menjalankan perintah pada TV, dengan Buzzer sebagai indikator perintah yang diterima. Hasil pengujian terhadap 25 kali percobaan dengan berbagai variasi kalimat perintah menunjukkan bahwa sistem berhasil mengenali 23 perintah dengan benar dan hanya menghasilkan 2 kesalahan, sehingga diperoleh *error rate* sebesar 8%. Kesalahan tersebut terjadi karena tidak ada *Pattern* yang terdaftar dalam program, sehingga perintah tidak dapat dicocokkan dengan pola *Pattern*. Hasil ini menunjukkan bahwa algoritma *WU-MANBER* efektif diterapkan pada sistem kontrol TV berbasis suara guna membantu memudahkan pengoperasian TV, meskipun pengembangan variasi kalimat perintah masih diperlukan untuk meningkatkan akurasi sistem.

ABSTRACT

Habib, Muh Fahma Sahmura. 2026. **TV Activation Based on Multi *Pattern Matching* Using *WU-MANBER***. Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Ajib Hanani, M.T (II) Dr. Mokhamad Amin Hariyadi, M.T.

Keywords: Raspberry Pi, *Speech recognition*, TV Control System, *WU-MANBER*,

Technological advancements have made it easier to execute commands on electronic devices. This study aims to implement the Wu-Manber method for text matching within a voice-controlled TV system, facilitating operation without the need for physical remote control interaction. The research employs a system engineering approach comprising data collection, design, implementation, and testing. Voice commands are captured via an Android application built with MIT App Inventor and converted into text using Google Speech Recognition. The text undergoes stopword removal and stemming before being matched against command patterns using the Wu-Manber algorithm running on a Raspberry Pi 3. Matching results are relayed to an Arduino, which transmits signals via an IR transmitter to execute TV commands, while a buzzer indicates that a command has been received. Testing across 25 trials with various command sentence structures revealed that the system correctly recognized 23 commands and produced only two errors, resulting in an error rate of 8%. These errors occurred because the specific command patterns were not registered in the program, preventing a match. These results demonstrate the effectiveness of the Wu-Manber algorithm in voice-controlled TV systems to simplify operation, although further development of command sentence variations is required to enhance system accuracy.

مستخلص البحث

حبيب، محمد فهمي سهمورا. 2026. تفعيل التلفاز باستخدام خوارزمية *WU-MANBER* القائمة على مطابقة الأنماط المتعددة. بحث جامعي. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك ابراهيم الإسلامية الحكومية مالانج. المشرف: (الأول) عجيب حنائي، م. ت (الثاني). محمد أمين هاريايدي، م. ت.

الكلمات المفتاحية: التعرف على الكلام، الإعاقة الجسدية، نظام التحكم بالتلفاز، *Wu- Manber, Raspberry Pi*.

لقد سهّلت التطورات التكنولوجية عملية تنفيذ الأوامر على الأجهزة الإلكترونية. تهدف هذه الدراسة إلى تطبيق خوارزمية لمطابقة النصوص ضمن نظام تلفزيوني يُحكم فيه صوتياً، مما يتيح التشغيل دون الحاجة إلى استخدام (*Wu-Manber*) "وو-مانبر" (جهاز التحكم عن بُعد) الرهوت كنترول (يدوياً). يعتمد البحث نهج هندسة النظم الذي يشمل مراحل جمع البيانات، والتصميم ثم، "MIT App Inventor" والتنفيذ، والاختبار. يتم التقاط الأوامر الصوتية عبر تطبيق أندرويد تم تطويره باستخدام منصة تخضع النصوص (*Google Speech Recognition*) تحويلها إلى نص باستخدام تقنية التعرف على الكلام من جوجل قبل مطابقتها مع أنماط (*stemming*) واستخلاص الجذور اللغوية (*stopword removal*) لعمليات إزالة الكلمات الشائعة تُرسل نتائج المطابقة إلى لوحة "Raspberry Pi 3" الأوامر المحددة مسبقاً باستخدام خوارزمية "وو-مانبر" التي تعمل على لوحة لتنفيذ أوامر التلفاز، بينما (IR) التي تقوم بدورها بإرسال إشارات عبر جهاز إرسال يعمل بالأشعة تحت الحمراء، "Arduino" إشارة تدل على استلام الأمر. أظهرت الاختبارات -التي شملت 25 محاولة باستخدام هياكل (*buzzer*) يُصدر جهاز تنبيه صوتي %، جمل متنوعة للأوامر- أن النظام تعرف بشكل صحيح على 23 أمراً ووقعت حالات خطأ فقط، مما أدى إلى نسبة خطأ بلغت 8. وقد نتجت هذه الأخطاء عن عدم تسجيل أنماط الأوامر المحددة تلك في البرنامج، مما حال دون إتمام عملية المطابقة. وتُبرز هذه النتائج فعالية خوارزمية "وو-مانبر" في أنظمة التلفاز التي تعمل بالتحكم الصوتي لتبسيط عملية التشغيل، رغم الحاجة إلى مزيد من التطوير ليشمل تنوعاً أكبر في صياغة جمل الأوامر بهدف تعزيز دقة النظام.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi saat ini telah memberikan kemudahan dalam berbagai aspek kehidupan manusia, termasuk dalam hal menjalankan perintah perangkat elektronik dalam kehidupan sehari-hari seperti menonton televisi. Namun, aktivitas seperti menyalakan TV, mengganti saluran, mengatur volume, maupun mematikan TV masih umumnya dilakukan dengan menggunakan remote control secara langsung. Kondisi ini mendorong pentingnya pengembangan teknologi yang dapat membantu memudahkan pengoperasian TV tanpa sentuhan langsung pada remote. Dengan memanfaatkan teknologi pengenalan suara dan sistem kontrol berbasis *Internet of Things (IoT)*, pengguna dapat memberikan perintah secara lebih praktis, cepat, dan efisien hanya melalui suara, sehingga meningkatkan kenyamanan dalam penggunaan perangkat televisi sehari-hari.

Perkembangan teknologi informasi dan komunikasi telah mengalami kemajuan yang sangat pesat dalam beberapa tahun terakhir. Kemajuan tersebut mendorong terciptanya berbagai inovasi yang memberikan kemudahan bagi masyarakat dalam menjalankan aktivitas sehari-hari. Salah satu inovasi yang berkembang adalah teknologi *Internet of Things (IoT)*, yaitu konsep yang memungkinkan perangkat elektronik saling terhubung dan bertukar data melalui jaringan internet. Penerapan teknologi *IoT* pada perangkat televisi memungkinkan pengguna untuk mengendalikan berbagai fungsi televisi, seperti menyalakan, mematikan, mengganti saluran, dan mengatur volume melalui perintah suara tanpa

perlu menggunakan *remote control* secara langsung. Hal ini memberikan kemudahan dan meningkatkan kenyamanan dalam penggunaan perangkat televisi (Fitriani, 2023).

Salah satu solusi yang dapat diterapkan untuk mengatasi permasalahan tersebut adalah dengan memanfaatkan teknologi *Internet of Things (IoT)*, yang memungkinkan berbagai perangkat saling terhubung dan dapat dikendalikan melalui jaringan internet. Dengan mengintegrasikan teknologi *IoT* dan sistem pengenalan perintah suara, pengguna dapat mengoperasikan televisi secara langsung melalui perintah suara tanpa memerlukan sentuhan fisik pada remote TV. Melalui pendekatan ini, aktivitas seperti menyalakan TV, mengganti saluran, mengatur volume, maupun mematikan TV dapat dilakukan dengan lebih mudah, praktis, dan efisien sehingga meningkatkan kenyamanan pengguna dalam mengoperasikan televisi.

Speech recognition merupakan sebuah proses pengubahan sinyal suara menjadi bentuk teks kalimat. Proses ini dirancang untuk membangun sistem yang memungkinkan komputer menerima input berupa kata-kata yang diucapkan secara verbal. Dengan teknologi ini, suatu perangkat dapat mengenali serta memahami kalimat yang diucapkan melalui proses digitalisasi suara, yaitu dengan mencocokkan sinyal yang diterima dengan pola-pola tertentu yang sebelumnya telah tersimpan di dalam sistem perangkat tersebut (Faroqi *et al.*, 2016).

Sistem ini diharapkan dapat memberikan kemudahan bagi pengguna dalam mengoperasikan televisi. Hal tersebut sejalan dengan firman Allah SWT dalam surah *Al-Maidah* ayat 2 yang menganjurkan umat manusia untuk saling membantu

dalam melakukan kebaikan. Ayat tersebut menjelaskan bahwa tolong-menolong yang dianjurkan dalam Islam adalah tolong-menolong dalam kebaikan dan ketakwaan, sementara tolong-menolong dalam perbuatan dosa dan kemungkaran tidak dibenarkan dalam ajaran Islam (Zabidi, 2020).

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَحْلُوا شَعَائِرَ اللَّهِ وَلَا الشُّهُرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَائِدَ وَلَا آمِينَ الْبَيْتِ الْحَرَامَ يَبْتَغُونَ فَضْلًا مِّن رَّبِّهِمْ وَرِضْوَانًا وَإِذَا حَلَلْتُمْ فَاصْطَادُوا وَلَا يَجْرِمَنَّكُمْ شَنَاٰنُ قَوْمٍ أَن صَدُّوكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ أَن تَعْتَدُوا وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ عَاثُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“Wahai orang-orang yang beriman, janganlah kamu melanggar syiar-syiar (kesucian) Allah,193) jangan (melanggar kehormatan) bulan-bulan haram,194) jangan (mengganggu) hadyu (hewan-hewan kurban)195) dan qalā'id (hewan-hewan kurban yang diberi tanda),196) dan jangan (pula mengganggu) para pengunjung Baitulharam sedangkan mereka mencari karunia dan rida Tuhannya!197) Apabila kamu telah bertahalul (menyelesaikan ihram), berburulah (jika mau). Janganlah sekali-kali kebencian(-mu) kepada suatu kaum, karena mereka menghalang-halangi dari Masjidilharam, mendorongmu berbuat melampaui batas (kepada mereka). Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya (Q.S. Al-Maidah: 2).”

Menurut M. Quraish Shihab dalam Tafsir Al-Misbah, ayat ini menegaskan bahwa umat Islam diperintahkan untuk bekerja sama atau saling tolong menolong dalam segala bentuk kebaikan yang membawa manfaat bagi individu maupun masyarakat. Kerja sama (*ta'awun*) tidak dibatasi hanya pada sesama Muslim, tetapi mencakup semua pihak selama bertujuan mewujudkan kebajikan dan ketakwaan.

Rasulullah SAW juga bersabda dalam sebuah hadist shahih yang diriwayatkan oleh Imam Bukhori No. 30 pada kitab *Fathul Bari* mengenai pentingnya tolong-menolong. Berikut merupakan hadist tersebut:

صحيح البخاري ٣٠: حَدَّثَنَا عَبْدُ الرَّحْمَنِ بْنُ الْمُبَارَكِ حَدَّثَنَا حَمَّادُ بْنُ زَيْدٍ حَدَّثَنَا أَيُّوبُ وَيُونُسُ عَنْ الْحَسَنِ عَنْ الْأَخْنَفِ بْنِ قَيْسٍ قَالَ ذَهَبْتُ لِأَنْصُرَ هَذَا الرَّجُلَ فَلَقِيَنِي أَبُو بَكْرَةَ فَقَالَ أَيْنَ تُرِيدُ قُلْتُ أَنْصُرَ هَذَا الرَّجُلَ قَالَ ارْجِعْ فَإِنِّي سَمِعْتُ رَسُولَ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ يَقُولُ إِذَا تَقَى الْمُسْلِمَانِ بِسَيَفَيْهِمَا فَالْقَاتِلُ وَالْمَقْتُولُ فِي النَّارِ فَقُلْتُ يَا رَسُولَ اللَّهِ هَذَا الْقَاتِلُ فَمَا بَالُ الْمَقْتُولِ قَالَ إِنَّهُ كَانَ حَرِيصًا عَلَى قَتْلِ صَاحِبِهِ

Shahih Bukhari 30: Telah menceritakan kepada kami Abdurrahman bin Al Mubarak Telah menceritakan kepada kami Hammad bin Zaid Telah menceritakan kepada kami Ayyub dan Yunus dari Al Hasan dari Al Ahnaf bin Qais berkata: aku datang untuk menolong seseorang kemudian bertemu Abu Bakrah, maka dia bertanya: "Kamu mau kemana?" Aku jawab: "hendak menolong seseorang" dia berkata: "Kembalilah, karena aku pernah mendengar Rasulullah shallallahu 'alaihi wa sallam bersabda: "Jika dua orang muslim saling bertemu (untuk berkelahi) dengan menghunus pedang masing-masing, maka yang terbunuh dan membunuh masuk neraka." aku pun bertanya: "Wahai Rasulullah, ini bagi yang membunuh, tapi bagaimana dengan yang terbunuh?" Maka Nabi shallallahu 'alaihi wa sallam menjawab: "Dia juga sebelumnya sangat ingin untuk membunuh temannya."

Menurut penjelasan Ibnu Hajar Al-Asqalani dalam kitab Fathul Bari, hadis riwayat Imam Bukhari No. 30 mengandung peringatan agar seorang Muslim tidak membantu atau terlibat dalam konflik yang mengarah pada pertumpahan darah tanpa dasar kebenaran yang jelas. Rasulullah SAW menjelaskan bahwa baik pembunuh maupun yang terbunuh mendapatkan ancaman karena keduanya memiliki keinginan untuk saling membinasakan. Hadis ini menunjukkan bahwa prinsip tolong-menolong dalam Islam harus didasarkan pada kebajikan dan ketakwaan, bukan pada tindakan yang mengandung permusuhan, kekerasan, atau kezaliman.

Namun, dalam penerapannya, sistem perintah suara sering kali mengalami kendala dalam pencocokan dan pencarian kata kunci suara secara cepat dan akurat. Oleh karena itu, untuk mengatasi permasalahan tersebut, maka digunakan algoritma WU-MANBER, yang mana algoritma WU-MANBER sendiri merupakan sebuah

algoritma pencocokan string yang efisien dan mampu mengenali pola dengan toleransi kesalahan. Penggunaan algoritma ini memungkinkan sistem mengenali perintah suara meskipun terdapat sedikit kesalahan pengucapan atau kebisingan lingkungan.

Dengan menggabungkan *teknologi Internet of Things (IoT)*, perintah suara, dan algoritma *WU-MANBER*, sistem kontrol TV yang dikembangkan diharapkan mampu memberikan solusi yang lebih praktis dan efisien dalam pengoperasian televisi. Sistem ini memungkinkan pengguna untuk mengendalikan berbagai fungsi TV melalui perintah suara tanpa memerlukan sentuhan langsung pada *remote control*, sehingga dapat meningkatkan kemudahan, kenyamanan, dan pengalaman pengguna dalam berinteraksi dengan perangkat elektronik di rumah.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana pengguna dapat mengontrol TV dengan perintah suara menggunakan metode *WU-MANBER* untuk pencocokan teks.

1.3 Batasan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Hanya dapat digunakan pada satu jenis TV.
2. Kontrol TV berupa menyalakan, mematikan, menaikkan volume, menurunkan volume, mengontrol menu berupa kanan, kiri, atas, dan bawah.

3. Sistem kontrol TV hanya menerima *input* berupa perintah suara yang diucapkan dalam bahasa Indonesia.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah mengimplementasikan metode *WU-MANBER* untuk pencocokan teks pada sistem kontrol TV untuk memberikan kemudahan dan kenyamanan bagi pengguna dalam mengoperasikan televisi melalui perintah suara.

1.5 Manfaat Penelitian

Berdasarkan tujuan, Penelitian ini diharapkan memberikan beberapa manfaat sebagai berikut:

1. Sistem ini memberikan kemudahan dan kenyamanan bagi pengguna dalam mengoperasikan televisi melalui perintah suara, sehingga berbagai fungsi TV dapat dikendalikan tanpa menggunakan remote control secara langsung.
2. Menjadikan penelitian ini sebagai rujukan untuk penelitian lebih lanjut.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Peneliti sebelumnya telah menjelaskan beberapa penelitiannya untuk menjadi perbandingan dan bahan referensi bagi peneliti. Peneliti juga berupaya untuk menghindari adanya kemiripan dengan studi yang dimiliki oleh peneliti sebelumnya. Pada sub bab ini, peneliti memberikan hasil dari penelitian sebelumnya, sebagai berikut:

Ada beberapa peneliti yang sebelumnya menggunakan algoritma *WU-MANBER* dalam penerapannya. Adapun penelitiannya yaitu yang dilakukan oleh (B. B. Putri & Ernawati, 2015) yang menerapkan algoritma *WU-MANBER* untuk pencarian kesamaan DNA. Penelitian tersebut memberikan hasil kecocokan DNA dengan waktu yang lebih efisien dan efektif dengan menggunakan algoritma *WU-MANBER*.

Penelitian lain menunjukkan bahwa penerapan algoritma *WU-MANBER* dalam pencarian pola kata pada file teks dapat digunakan sebagai simulasi untuk membantu memahami cara kerja algoritma yang dikembangkan oleh Sun Wu dan Udi Manber dalam menyelesaikan permasalahan *multi-Pattern matching*. Algoritma ini bekerja dengan memanfaatkan tabel *SHIFT*, *HASH*, dan *PREFIX* guna meningkatkan efisiensi proses pencarian. Dengan mekanisme tersebut, algoritma Sun Wu dan Udi Manber terbukti mampu melakukan pencarian terhadap sekumpulan pola kata dengan waktu eksekusi yang relatif singkat (Chandra, 2013).

Selain itu penelitian yang dilakukan oleh (PyrgIoTis *et al.*, 2012) yang meneliti tentang implementasi paralel algoritma *WU-MANBER* menggunakan kerangka kerja *OpenCL*. Penelitian tersebut menjelaskan bahwa algoritma *WU-MANBER* dengan *OpenCL* yang digunakan untuk pencocokan beberapa pola telah disajikan. Untuk evaluasi kinerja, waktu proses praktis dari fase pencarian telah dibandingkan dengan waktu berturut-turut yang sesuai. Hasil eksperimen menunjukkan hasil bahwa implementasi parallel mencapai peningkatan kecepatan yang signifikan.

Selain itu juga ada penelitian pada Sistem deteksi intrusi jaringan dan perangkat lunak antivirus digunakan untuk mendeteksi lalu lintas berbahaya, seperti serangan *denial-of-service (DoS)*, *worm*, dan *malware*. Setiap serangan memiliki ciri khas tersendiri, sehingga diperlukan deteksi berbasis tanda tangan yang mengandalkan pencocokan pola. Namun, pencocokan pola merupakan tugas yang kompleks dan memerlukan banyak waktu, memori, serta sumber daya komputasi. Untuk mengatasinya, digunakan algoritma *WU-MANBER* yang mampu mempercepat proses pencarian pola, dengan hasil eksperimen menunjukkan kinerja yang sangat cepat (Aldwairi *et al.*, 2020).

Pada penelitian yang dilakukan oleh (M. Ulum, 2022) digunakan algoritma *Knuth-Morris-Pratt (KMP)* untuk melakukan pencarian dan pencocokan pola pada sistem kontrol gorden. Penelitian ini bertujuan untuk meningkatkan efisiensi dan keakuratan sistem dalam mengenali pola perintah yang diberikan pengguna. Hasil pengujian menunjukkan bahwa algoritma KMP mampu memberikan kinerja yang sangat baik, dengan rata-rata durasi pencarian pola hanya 513 ms dan tingkat

akurasi mencapai 100%. Pengujian dilakukan dengan memanfaatkan confusion matrix sebagai alat evaluasi performa sistem. Objek yang digunakan dalam penelitian ini adalah gorden, yang menjadi bukti bahwa algoritma KMP dapat diimplementasikan dengan baik pada sistem kendali berbasis pencocokan pola dan mampu menghasilkan respon yang cepat serta akurat.

Terakhir pada penelitian yang dilakukan oleh (Rabbani, 2021), algoritma *Boyer-Moore* diterapkan untuk melakukan *Pattern matching* atau pencarian pola pada sistem kontrol pagar rumah. Penelitian ini bertujuan untuk memastikan sistem mampu mengenali perintah dengan cepat dan tepat sehingga dapat meningkatkan keamanan serta kenyamanan pengguna. Dalam pengujiannya, dilakukan pengolahan terhadap 50 kalimat perintah dengan 4 pola berbeda sebagai data uji. Hasil penelitian menunjukkan bahwa algoritma *Boyer-Moore* mampu memberikan kinerja optimal dengan tingkat akurasi mencapai 100%. Selain itu, waktu proses pencarian pola rata-rata hanya sekitar 0,00672 ms, yang menunjukkan kemampuan algoritma ini dalam melakukan pencocokan pola secara sangat cepat. Temuan ini membuktikan bahwa *Boyer-Moore* dapat diandalkan untuk implementasi pada sistem kontrol pagar rumah yang membutuhkan kecepatan tinggi dan akurasi penuh.

2.2 MIT App Inventor

MIT App Inventor, dikembangkan oleh *Massachusetts Institute of Technology (MIT)*, adalah platform inovatif untuk pembuatan aplikasi Android tanpa memerlukan keahlian pemrograman yang kompleks. Platform ini populer

karena kemudahan penggunaannya dan berbagai aplikasi *mobile* yang dapat dibuat dengan cepat (G. Putri, 2024).

MIT App Inventor merupakan sebuah aplikasi inovatif hasil pengembangan *Google* dan *MIT* yang bertujuan untuk memperkenalkan serta memudahkan proses *pengembangan* pemrograman berbasis Android. Aplikasi ini mengubah bahasa pemrograman yang awalnya bersifat tekstual dan kompleks menjadi bahasa pemrograman berbasis visual, yaitu dengan menggunakan metode drag and drop berupa susunan blok-blok pemrograman. (Nugraha & Adriansyah, 2019).

MIT App Inventor merupakan sebuah program yang digunakan untuk merancang dan membuat aplikasi berbasis Android. Aplikasi ini merupakan produk yang dikembangkan oleh *Google* dan dikenal mudah dipelajari serta digunakan dalam proses pembuatan aplikasi. Dalam penggunaannya, *App Inventor* membantu pengguna untuk mendesain dan membangun aplikasi *Android* yang berbasis antarmuka Java serta halaman *web*. (M. B. Ulum & Badri, 2023).

2.3 Televisi (TV)

Televisi merupakan media elektronik yang bersifat audio-visual, artinya dapat dinikmati melalui indra penglihatan sekaligus pendengaran. Penonton tidak hanya dapat melihat gambar yang ditampilkan, tetapi juga dapat mendengar suara yang menyertainya serta memahami narasi dari tayangan tersebut. Televisi memiliki tiga karakteristik utama, yaitu bersifat audio-visual karena dapat didengar dan dilihat secara bersamaan, mampu menampilkan serta menerjemahkan gambar, dan memiliki cara pengoperasian yang lebih kompleks dibandingkan media elektronik lain seperti radio. (Verdian *et al.*, 2023).

Selain itu, televisi juga dapat dimaknai sebagai sebuah teknologi yang memiliki pengaruh besar terhadap perkembangan teknologi informasi di berbagai belahan dunia. Pada masa awal kemunculannya, televisi merupakan perpaduan antara teknologi optik, mekanik, dan elektronik yang berfungsi untuk merekam, menampilkan, sekaligus menyiarkan gambar secara visual (Budiman, 2017).

Sedangkan Nurpadila *et al.* (2020), Televisi disebutkan sebagai media komunikasi massa yang menggabungkan unsur audio dan visual, sehingga masyarakat dapat melihat tampilan visual yang disajikan sekaligus mendengar suara yang dihasilkan televisi. Televisi memiliki beberapa fungsi, di antaranya sebagai sarana penyampaian informasi, media edukasi, sarana kontrol sosial, fungsi hiburan, serta media yang menjembatani komunikasi antarwilayah secara geografis.

Televisi merupakan media massa yang memiliki keunggulan dan kekuatan besar, di antaranya kemampuannya dalam membentuk imaji, opini, bahkan perilaku masyarakat. Bagi masyarakat, televisi sebagai media massa dapat dimanfaatkan untuk berbagai kepentingan, baik di bidang ekonomi, politik, maupun sosial budaya. Melalui televisi, masyarakat juga dapat memperoleh dan menyerap beragam informasi terkait perekonomian, seperti informasi mengenai produk hasil produksi, baik yang berasal dari dalam negeri maupun luar negeri (Sudiana, 1986).

Di sisi lain, televisi sebagai media massa yang telah hampir satu abad menjadi media favorit masyarakat, kini mulai mengalami pergeseran posisi akibat hadirnya media baru, yaitu media sosial yang didukung oleh perkembangan internet. Masyarakat Indonesia pun mulai beralih dalam pola penggunaan media, tidak hanya berdampak pada menurunnya minat terhadap media cetak, tetapi juga

memengaruhi keberadaan media elektronik seperti televisi. Bahkan, saat ini siaran televisi dapat diakses dan diatur melalui smartphone atau gadget (Abdullah & Puspitasari, 2018).

2.4 Algoritma Wu Manber

Algoritma WU-MANBER merupakan salah satu algoritma *multiple Pattern matching* yang dikenal sebagai algoritma pencocokan *string* dengan kinerja terbaik. Tingkat keberhasilan *algoritma WM* sangat bergantung pada berbagai tabel yang dihasilkan dari proses pengolahan *Pattern* sebelumnya. Algoritma ini menghitung nilai *HASH* dari blok *Suffix Pattern* menggunakan teknik *HASHing*, kemudian menghubungkan seluruh *Pattern* yang memiliki blok *Suffix* sama ke dalam sebuah daftar yang disimpan dalam tabel *HASH*. Selanjutnya, nilai *HASH* dihitung untuk blok karakter pada *window* pencocokan, lalu dicocokkan dengan tabel *HASH* untuk menemukan entri yang memiliki *Suffix Pattern* yang sesuai dengan blok karakter tersebut. Algoritma ini pertama kali dikembangkan oleh Sun Wu dan Udi Manber pada tahun 1994. Konsep pemrosesan awal tabel pada algoritma ini diadaptasi dari salah satu varian *algoritma Boyer-Moore* dalam menerapkan aturan *bad character rule*. (Velda, 2019).

Tidak hanya itu algoritma *WU-MANBER* dapat diartikan juga sebagai algoritma pencocokan string yang pada umumnya digunakan untuk pencarian pola secara efisien di dalam teks. Algoritma ini merupakan penggabungan dari algoritma *Boyer-Moore* dan *Knuth-Morris-Pratt* untuk menghasilkan pencocokan pola yang cepat dan akurat. Algoritma *WU-MANBER* membagi pola menjadi beberapa

subpola dan menggunakannya untuk mencari kecocokan dalam teks, sehingga lebih efisien dan pola yang dicari relatif pendek (GeeksforGeeks, 2025).

Algoritma *WU-MANBER* merupakan algoritma berbasis *HASHing* yang mengadopsi konsep pergeseran karakter buruk (*bad character SHIFT*) dari algoritma *Boyer-Moore*. Nilai pergeseran karakter buruk tersebut digunakan untuk menentukan jarak pergeseran maksimum apabila terjadi ketidakcocokan pola. Dalam proses kerjanya, algoritma *WU-MANBER* menghitung nilai *HASH* dari blok *sufiks* suatu pola dengan menggunakan teknik *HASHing*, kemudian menghubungkan seluruh pola yang memiliki blok *sufiks* sama ke dalam satu daftar yang disimpan sebagai entri pada tabel *HASH*. Selanjutnya, nilai *HASH* dihitung pula untuk blok karakter yang terdapat dalam jendela pencocokan, lalu dicocokkan dengan tabel *HASH* guna menemukan entri yang memiliki *sufiks* pola sama dengan blok karakter tersebut (Bhardwaj & Garg, 2015).

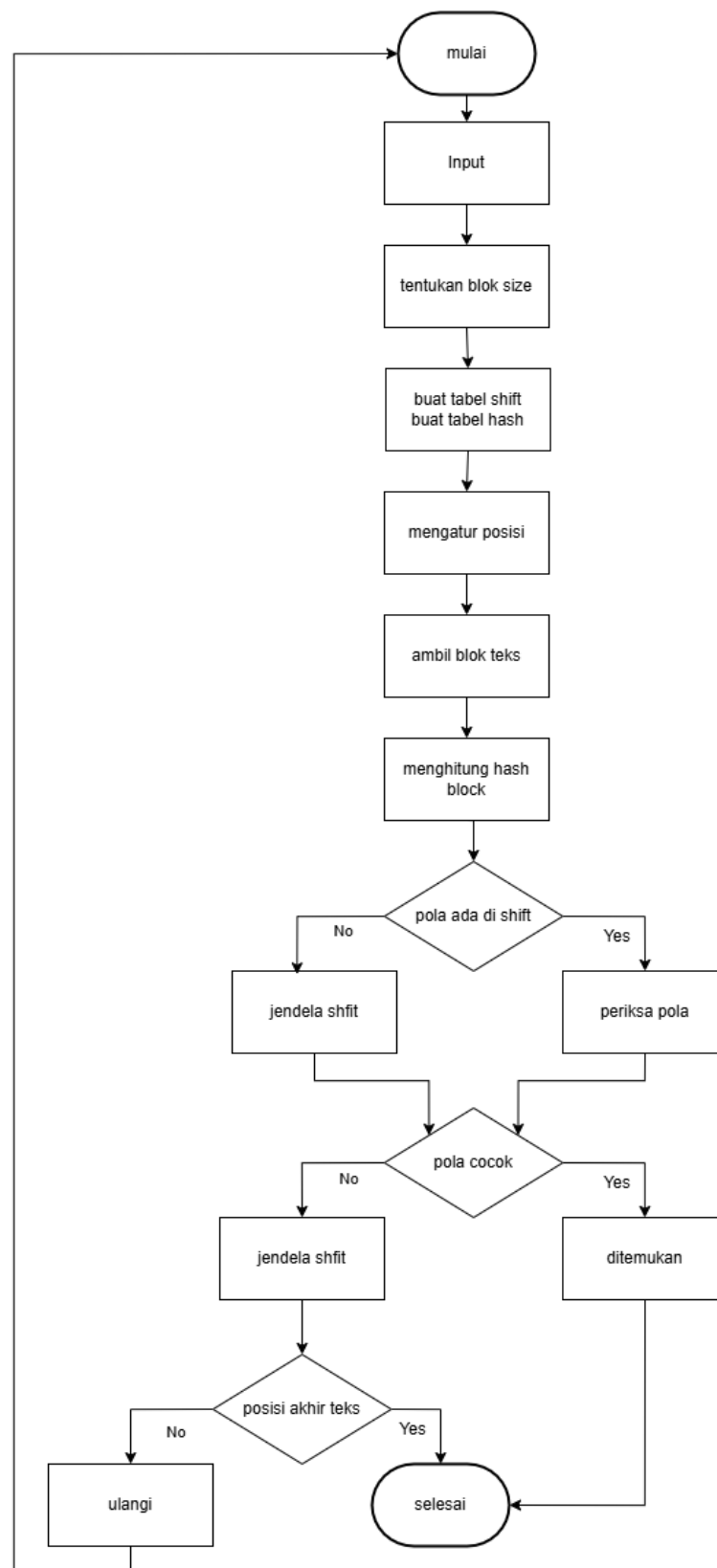
Proses pelacakan *string* seringkali mengalami kegagalan akibat adanya kesalahan penulisan, baik pada *string* yang dicari maupun pada file teks yang menjadi sumber pencarian. Hal ini dapat terjadi karena pengguna tidak mengingat dengan tepat ejaan *string* yang dimaksud, atau karena memang terdapat kesalahan pengejaan pada file teks tersebut. Untuk mengatasi permasalahan ini, diperlukan suatu metode pelacakan *string* yang memperhitungkan toleransi kesalahan, sehingga pencarian dapat dilakukan secara tidak eksak atau *approximate* (Wu & Manber, 1992).

Cara kerja algoritma *WU-MANBER* yaitu memiliki dua tahap, yaitu pertama adalah *preprocessing* yang meliputi pembentukan tabel *SHIFT*, pembentukan tabel

HASH, dan pembentukan tabel *Prefix*. Kedua yaitu tahapan *scanning* yang dimulai dari *match window* hingga proses selesai menelusuri seluruh teks (Putri *et al.*, 2015). Sebagaimana penjelasan Malav & Rasool (2014), bahwa algoritma WM memajukan algoritma pencocokan *string* multi pola dengan memperkenalkan konsep baru dengan menggeser atau melewati karakter yang tidak perlu. Kemudian karakter yang cocok dalam fase pencarian berdasarkan blok karakter akan ditemukan, sehingga meningkatkan kecepatan pencarian dan lebih cepat dari pada algoritma sebelumnya. Algoritma ini dibagi menjadi dua fase, yaitu fase pra-pemrosesan dan fase pemindaian.

Algoritma yang dikembangkan oleh Sun Wu dan Udi Manber dipublikasikan pada Mei 1994. Metode *WU-MANBER* ini lahir dari hasil pengembangan ide algoritma *Boyer-Moore* yang dilakukan oleh Sun Wu dan Udi Manber untuk menyelesaikan permasalahan multi-*Pattern matching* (B. B. Putri & Ernawati, 2015).

Algoritma *WU-MANBER* memiliki tahapan-tahapan sistematis dalam penyelesaian masalah. Menurut Putri & Ernawati, (2015), berikut cara kerja *WU-MANBER* dapat dilihat pada gambar 2.1.



Gambar 2. 1 Flowchart WU-MANBER

Flowchart algoritma *Wu-Manber* pada Gambar 2.1 menggambarkan tahapan proses pencocokan pola yang digunakan dalam sistem pengendali televisi berbasis perintah suara. Proses diawali dengan sistem menerima masukan berupa teks hasil konversi suara (*speech-to-text*) yang sebelumnya telah melalui tahap *preprocessing*, seperti mengubah seluruh huruf menjadi huruf kecil, menghapus tanda baca, serta menghilangkan kata-kata yang tidak berpengaruh terhadap proses pencarian pola. Teks hasil *preprocessing* tersebut selanjutnya digunakan sebagai masukan utama dalam proses pencocokan menggunakan algoritma *Wu-Manber*.

Setelah menerima masukan, sistem menentukan ukuran blok (*block size*) yang akan digunakan dalam proses pencarian. Pada penelitian ini digunakan ukuran blok sebanyak dua karakter (*block size* = 2) karena memberikan keseimbangan antara kecepatan pencarian dan efisiensi penggunaan memori. Ukuran blok ini menjadi dasar dalam pembentukan tabel-tabel yang digunakan oleh algoritma *Wu-Manber*, yaitu *Shift Table* dan *Hash Table*.

Tahap berikutnya adalah proses *preprocessing* terhadap seluruh pola perintah yang telah tersimpan di dalam sistem, seperti nyala, mati, volume naik, volume turun, dan mute. Pada tahap ini sistem membangun *Shift Table* yang berfungsi untuk menentukan jumlah pergeseran jendela pencarian ketika tidak ditemukan kecocokan, serta *Hash Table* yang berisi nilai hash dari blok-blok pola sehingga proses pencarian kandidat pola dapat dilakukan dengan lebih cepat. Pembentukan kedua tabel tersebut hanya dilakukan satu kali sebelum proses pencarian dimulai.

Selanjutnya sistem mengatur posisi awal pencarian pada teks masukan, kemudian mengambil blok karakter sesuai ukuran blok yang telah ditentukan. Blok karakter tersebut dihitung nilai *hash*-nya menggunakan fungsi *hash* yang sama dengan yang digunakan pada tahap *preprocessing*. Nilai *hash* yang diperoleh kemudian dibandingkan dengan isi *Shift Table* untuk mengetahui apakah terdapat kemungkinan kecocokan dengan pola yang telah disimpan.

Apabila nilai *hash* tidak ditemukan pada *Shift Table*, sistem tidak melakukan pemeriksaan karakter secara langsung, melainkan menggeser jendela pencarian sesuai nilai *shift* yang telah ditentukan. Mekanisme ini memungkinkan algoritma melewati beberapa karakter sekaligus sehingga proses pencarian menjadi lebih cepat dibandingkan metode pencocokan karakter satu per satu. Sebaliknya, apabila nilai *hash* menunjukkan adanya kandidat pola yang sesuai, sistem akan melanjutkan proses dengan melakukan pemeriksaan pola secara lengkap melalui perbandingan setiap karakter antara teks masukan dan pola yang tersimpan.

Jika hasil pemeriksaan menunjukkan seluruh karakter sesuai dengan pola yang dicari, maka sistem menyatakan bahwa pola telah ditemukan. Selanjutnya sistem mengambil kode *HEX* yang telah dipetakan pada pola tersebut dan mengirimkannya ke Raspberry Pi untuk dipancarkan melalui LED inframerah sebagai sinyal kendali menuju televisi. Namun, apabila hasil pemeriksaan karakter menunjukkan bahwa pola tidak cocok, sistem kembali menggeser jendela pencarian sesuai nilai *shift* dan mengulangi proses pengambilan blok serta perhitungan *hash* hingga seluruh teks selesai diperiksa.

Proses pencarian akan terus berulang sampai sistem mencapai posisi akhir teks. Apabila hingga akhir teks tidak ditemukan pola yang sesuai, maka algoritma mengakhiri proses tanpa mengirimkan kode *HEX* ke perangkat televisi. Sebaliknya, apabila pola berhasil ditemukan sebelum mencapai akhir teks, proses pencarian dihentikan dan sistem langsung mengeksekusi perintah sesuai fungsi televisi yang telah dipetakan. Dengan mekanisme tersebut, algoritma *Wu-Manber* mampu melakukan pencocokan pola secara efisien karena mengurangi jumlah perbandingan karakter melalui penggunaan *Shift Table* dan *Hash Table*, sehingga waktu pencarian menjadi lebih cepat dan sesuai untuk diterapkan pada sistem pengendali televisi berbasis perintah suara.

2.5 Raspberry Pi



Gambar 2. 2 Raspberry Pi 3

Raspberry Pi merupakan komputer berukuran mini yang dapat dihubungkan dengan televisi atau monitor serta keyboard. Meskipun berukuran kecil, perangkat ini memiliki kemampuan yang mumpuni dan dapat dimanfaatkan untuk berbagai proyek elektronik, serta mampu menjalankan berbagai fungsi layaknya PC desktop atau komputer pada umumnya (Fernando *et al.*, 2022).

Raspberry Pi merupakan sebuah perangkat yang praktis dari segi ukuran, namun memiliki fungsi yang kompleks untuk memenuhi berbagai kebutuhan

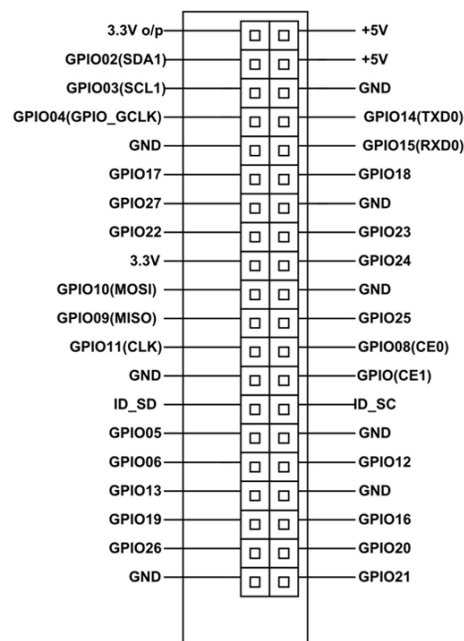
manusia, mulai dari penggunaannya sebagai mikrokontroler, server, hingga pengolahan citra digital. Gagasan di balik terciptanya Raspberry Pi sendiri bermula dari keinginan untuk melahirkan generasi baru programmer (Wijaya *et al.*, 2017).

Raspberry Pi, yang juga dikenal dengan istilah RasPi, merupakan sebuah *Single Board Computer (SBC)* berukuran sebesar kartu kredit yang dikembangkan oleh Raspberry Pi *Foundation* di Inggris dengan tujuan untuk mendorong pembelajaran ilmu komputer dasar di lingkungan sekolah. Perangkat ini menggunakan sistem *on a chip (SoC)* mulai dari *Broadcom BCM2835* hingga *BCM2837* pada Raspberry Pi 3, yang juga dilengkapi dengan prosesor *ARM1176JZF-S* dan bahkan prosesor *quad-core ARMv8 64-bit* berkecepatan *1.2GHz* pada Raspberry Pi 3, *GPU VideoCore IV*, serta kapasitas *RAM* hingga *1 GB* (Astri, 2016).

Sistem operasi Raspbian merupakan sistem operasi berbasis Debian yang dioptimalkan khusus untuk digunakan pada komputer mini Raspberry Pi. Sistem operasi ini dilengkapi dengan sejumlah program standar serta beberapa program pendukung yang berfungsi untuk menjalankan perangkat keras pada komputer mini Raspberry Pi tersebut (Rasyid *et al.*, 2013).

Raspberry Pi 3 memiliki jumlah pin *GPIO* sebanyak 40 pin, dengan beberapa istilah penting yang perlu dipahami dalam konfigurasinya. Pertama, pin 3.3V dan 5V berfungsi untuk memberikan tegangan ke berbagai komponen seperti sensor, LED, motor, dan relay, yang dihubungkan ke pin VCC pada komponen tersebut. Kedua, pin GND atau Ground, yang dihubungkan ke pin ground atau kutub negatif (-) pada LED, sensor, motor, maupun relay. Terakhir, pin *GPIO*

merupakan pin yang dapat dikendalikan melalui bahasa pemrograman Python, di mana pin ini dapat diatur untuk aktif, mati, atau menyala berdasarkan kondisi tertentu sesuai dengan program yang telah dibuat. Berikut pin *GPIO* pada Raspberry Pi dapat dilihat pada gambar 2.2.



Gambar 2. 3 PIN *GPIO*

Tabel 2. 1 Spesifikasi *Raspberry Pi 3*

Spesifikasi	Keterangan
<i>Soc</i>	<i>1.2GHz 64-bit quad-core ARMv8 CPU</i>
<i>Procesor</i>	<i>ARVv8</i>
<i>RAM</i>	<i>1 GB SDRAM 400MHz</i>
<i>GPU</i>	<i>VideoCore IV 3D graphics core</i>
<i>Wireless Adapter/LAN</i>	<i>802.1 In Wireless LAN</i>
<i>Bluetooth</i>	<i>Bluetooth 4.1 (built in), Bluetooth Low Energy (BLE)</i>
<i>GPIO</i>	<i>40 Pin</i>
<i>Port USB</i>	<i>4 USB Ports</i>
<i>Card Stroge</i>	<i>Micro SD card slot (now push-pull rather than push-push)</i>
<i>Jaringan</i>	<i>Ethernet Port</i>
<i>External Audio and Video</i>	<i>Full HDMI port, Camera interface (CSI), Display interface (DSI), Combined 3.5mm audio jack and composite video</i>
<i>Sistem Operassi</i>	<i>Debian GNU/Linux,Fedora, Arch Linux ARM, RISC OS</i>

Tabel 2.1 menunjukkan spesifikasi Raspberry Pi 3 yang digunakan sebagai perangkat utama dalam penelitian ini. Raspberry Pi 3 memiliki prosesor *ARMv8 Quad-Core 1,2 GHz*, *RAM 1 GB*, *GPU VideoCore IV*, serta dilengkapi konektivitas *Wi-Fi*, *Bluetooth*, *Ethernet*, dan 40 pin *GPIO* yang mendukung integrasi dengan berbagai perangkat keras. Selain itu, Raspberry Pi 3 menyediakan port *USB*, *HDMI*, slot *microSD*, serta mendukung beberapa sistem operasi berbasis *Linux*.

2.6 IR Transmitter



Gambar 2. 4 IR Transmitter

IR Transmitter merupakan komponen elektronika yang berfungsi untuk mengidentifikasi cahaya *infrared*, yang biasanya dirancang dalam satu modul khusus yang disebut *IR Detector Photomodules*. *IR Detector Photomodules* sendiri merupakan chip *detektor infrared* digital yang di dalamnya terdiri dari *fotodiode* dan *amplifier*. Pada bagian *transmitter*, komponen ini dibangun dari sebuah *LED infrared*, yang apabila dibandingkan dengan *LED* pada umumnya, memiliki karakteristik sinyal yang tidak tampak oleh mata manusia (Deswar & Pradana, 2021).

Infrared transmitter merupakan sebuah modul yang berfungsi untuk mengirimkan data melalui gelombang infra merah dengan frekuensi carrier sebesar

38 kHz. Modul ini dapat digunakan sebagai output dalam berbagai aplikasi transmisi data nirkabel, seperti pada bidang robotika, sistem pengamanan, dan lain sebagainya. Pemancar yang digunakan dalam sistem ini terdiri atas sebuah *Light Emitting Diode (LED)*, yaitu bahan semikonduktor yang dapat memancarkan cahaya monokromatik tidak koheren ketika dialiri tegangan maju. LED infra merah sendiri merupakan jenis diode yang khusus memancarkan cahaya infra merah, dengan salah satu contoh aplikasi sederhananya dapat ditemukan pada remote TV. Pada dasarnya, LED infra merah merupakan diode PN silikon biasa yang dikemas dalam kotak transparan (Yusniati, 2018).

Cahaya pada *LED* dihasilkan dari proses penggabungan *elektron* dan *hole* yang terjadi pada persambungan antara dua jenis semikonduktor, di mana setiap proses penggabungan tersebut disertai dengan pelepasan energi. Dalam penggunaannya, *LED* infra merah dapat diaktifkan dengan tegangan *Direct Current (DC)* untuk keperluan transmisi atau sensor pada jarak dekat, maupun dengan tegangan *Alternating Current (AC)* pada frekuensi 30-40 kHz untuk keperluan transmisi atau sensor pada jarak jauh (Yusniati, 2018).

Tabel 2. 2 Spesifikasi IR Transmitter

Spesifikasi
<i>High reliability</i>
<i>2.54mm Lead spacing</i>
<i>Low forward voltage</i>
<i>Good spectral matching to Si photodetector</i>
<i>Pb Free</i>
<i>This product itself will remain within RoHS compliant version.</i>
<i>Compliance with EU REACH</i>
<i>Compliance Halogen Free(Br < 900ppm, Cl < 900ppm, Br+Cl < 1500ppm)</i>

Tabel 2.2 menunjukkan spesifikasi IR Transmitter yang digunakan sebagai media untuk mengirimkan sinyal inframerah ke televisi. Komponen ini memiliki tingkat keandalan yang tinggi, tegangan maju (*forward voltage*) yang rendah, serta karakteristik spektrum yang sesuai dengan *fotodetektor* silikon sehingga mampu menghasilkan transmisi sinyal inframerah yang optimal. Selain itu, IR Transmitter telah memenuhi standar lingkungan seperti *RoHS*, *EU REACH*, dan *Halogen Free*, sehingga aman digunakan sebagai komponen pendukung dalam sistem pengendalian televisi berbasis Raspberry Pi.

2.7 String Matching

Berdasarkan definisi yang dikemukakan oleh *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)*, *string* diartikan sebagai sebuah susunan karakter, baik berupa angka, alfabet, maupun karakter lainnya, yang umumnya direpresentasikan dalam bentuk struktur data *array*. *String* sendiri dapat berbentuk kata, frasa, atau bahkan kalimat (Astuti, 2017).

Pencocokan string merupakan salah satu tahapan penting dalam proses pencarian string (*string searching*) pada sebuah dokumen, di mana hasil pencarian tersebut sangat bergantung pada teknik atau metode pencocokan yang digunakan. Berdasarkan definisi dari *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology (NIST)*, pencocokan string (*string matching*) dimaknai sebagai suatu permasalahan dalam menemukan pola susunan karakter dari sebuah string di dalam string lain atau di dalam bagian isi suatu teks (Astuti, 2017).

Dalam berbagai perangkat lunak praktis yang telah ada saat ini, algoritma pencocokan string menjadi salah satu komponen dasar dalam proses implementasinya. *String matching* sendiri digunakan untuk menemukan satu atau lebih *string*, yang disebut sebagai *Pattern*, di dalam suatu string lain yang disebut sebagai *text*, yaitu *string input* yang akan dicocokkan dengan isi dari *string text* tersebut (Nababan & Jannah, 2019).

Terdapat dua teknik utama dalam algoritma *string matching*. Pertama, *exact matching*, yaitu teknik pencocokan *string* yang hasilnya harus sama persis atau identik dengan *string* yang di-*input*, seperti yang diterapkan pada algoritma *WU-MANBER*, *Needleman-Wunsch*, *Smith-Waterman*, *Knuth-Morris-Pratt*, dan *Boyer-Moore-Horspool*. Kedua, *approximate matching*, yaitu teknik pencocokan string yang hasilnya dapat mengandung sedikit perbedaan, sehingga tidak harus sama persis dengan string yang di-*input*, seperti yang diterapkan pada algoritma *Fuzzy String Searching*, *Rabin-Karp*, dan *Brute Force*. (Nababan & Jannah, 2019).

2.8 Speech recognition

Speech recognition merupakan proses pengubahan sinyal akustik yang ditangkap melalui mikrofon atau telepon menjadi rangkaian kata. Kata-kata yang berhasil dikenali tersebut selanjutnya dapat dimanfaatkan sebagai hasil akhir untuk berbagai aplikasi, seperti *command and control*, *input data*, maupun pembuatan dokumen. Teknologi *speech recognizer* pertama kali dikenalkan pada tahun 1952, dengan salah satu perangkat awal yang terkenal adalah *IBM Shoebox*, yang diperkenalkan pada tahun 1963 dalam ajang *New York World's Fair* (Rumopa, 2015).

Speech recognition merupakan teknologi yang dapat memungkinkan sistem komputer mengenali atau memahami perkataan manusia (Siahaan *et al.*, 2020). Sedangkan berpendapat bahwa *Speech recognition* merupakan bagian dari kemajuan teknologi di bidang kecerdasan mesin yang semakin terlihat dengan banyaknya perangkat teknologi berbasis komputer yang kini dilengkapi dengan fitur perintah suara atau *Speech recognition*.

Speech recognition, atau yang dikenal pula dengan istilah pengenalan pembicaraan maupun perintah suara otomatis, merupakan suatu proses pengubahan kata-kata yang diucapkan menjadi bentuk teks. Istilah "perintah suara" terkadang digunakan untuk merujuk pada sistem yang perlu dilatih terlebih dahulu agar dapat mengenali suara pembicara tertentu, sebagaimana diterapkan pada sebagian besar perangkat lunak pengenalan suara di komputer desktop. Sistem yang dirancang untuk mengenali pembicara tertentu ini dapat menyederhanakan proses penerjemahan ucapan menjadi teks. Berbeda halnya dengan pengenalan pembicaraan dalam arti yang lebih luas, yang merujuk pada teknologi yang mampu mengenali ucapan dari siapa pun, seperti yang diterapkan pada sistem *call center* yang dapat memahami suara dari berbagai pengguna. Selain itu, *Speech recognition* juga termasuk salah satu bentuk *biometric recognition*, karena sistem komputer mengenali ucapan berdasarkan karakteristik suara yang telah dikonversi menjadi *digital voice print*. (Rumopa, 2015).

Pengenalan suara merupakan salah satu aplikasi teknologi yang terus berkembang. Sistem ini memungkinkan interaksi manusia dengan komputer dengan cara memasukkan data melalui suara. Salah satu penerapannya adalah

meningkatkan efisiensi di industri manufaktur, misalnya dengan mengontrol mesin hanya melalui perintah suara. Proses ini diawali dengan mengonversi spektrum suara menjadi data digital, kemudian diubah ke dalam bentuk diskrit untuk dianalisis (Rumopa, 2015).

Selain itu, *Speech recognition* dapat diartikan sebagai proses identifikasi suara berdasarkan kata yang diucapkan. Parameter yang dibandingkan ialah tingkat penekanan suara yang kemudian akan dicocokkan dengan template database yang tersedia. Sistem perintah suara berdasarkan orang yang berbicara inilah yang dinamakan *speaker recognition*. *Speech recognition* (juga dikenal sebagai perintah suara otomatis, pengenalan komputer pidato, pidato ke teks) mengkonversi kata yang diucapkan dengan teks. Para "*Speech recognition*" istilah kadang-kadang digunakan untuk merujuk pada sistem pengenalan yang harus dilatih untuk pembicara tertentu seperti halnya bagi sebagian besar perangkat lunak pengenalan desktop. Menyadari pembicara dapat menyederhanakan tugas menerjemahkan pidato. *Speech recognition* adalah solusi yang lebih luas yang mengacu pada teknologi yang dapat mengenali pidato tanpa ditargetkan pada satu pembicara seperti panggilan sistem yang dapat mengenali suara sewenang-wenangnya (Nada *et al.*, 2019).

Speech recognition, atau yang biasa dikenal dengan istilah *Automatic Speech recognition (ASR)*, merupakan suatu bentuk perkembangan teknik dan sistem yang memungkinkan komputer untuk dapat menerima input berupa kata-kata yang diucapkan secara verbal. Melalui teknologi ini, suatu perangkat dapat mengenali dan memahami kata-kata yang diucapkan dengan cara mendigitalisasi

kata tersebut, kemudian mencocokkan sinyal digital yang dihasilkan dengan pola tertentu yang telah tersimpan di dalam sistem perangkat. Proses pengubahan kata yang diucapkan menjadi sinyal digital dilakukan dengan mengonversi gelombang suara menjadi sekumpulan angka, yang selanjutnya disesuaikan dengan kode-kode tertentu guna mengidentifikasi kata yang dimaksud. Hasil identifikasi kata yang diucapkan tersebut dapat ditampilkan dalam bentuk teks tulisan, atau dapat pula dibaca oleh perangkat teknologi sebagai sebuah perintah untuk melaksanakan suatu tugas, misalnya penekanan tombol pada telepon genggam yang dilakukan secara otomatis melalui komando suara (Siahaan *et al.*, 2020).

2.9 Arduino Uno



Gambar 2. 5 Arduino Uno

Arduino uno merupakan sebuah alat elektronika yang bersifat *open source* baik digunakan dalam *hardware* atau *software*. Arduino Uno memiliki 14 pin input output yang dapat digunakan. Koneksi yang digunakan dalam arduino ini menggunakan koneksi *USB*. Dalam penggunaannya arduino ini menggunakan bahasa pemrograman bahasa C dengan menggunakan bantuan pustaka bagi pengembang (Laudira, 2020).

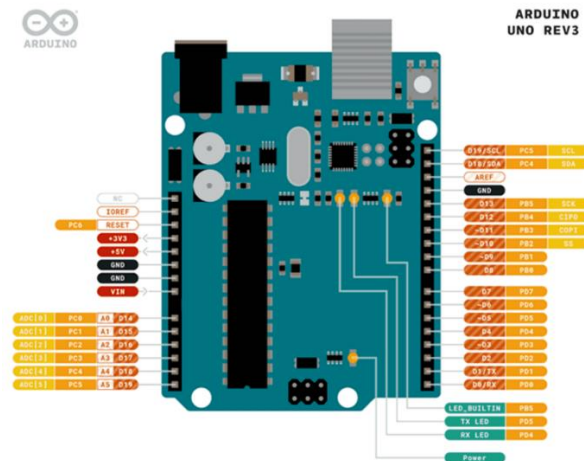
Arduino Uno merupakan salah satu jenis papan (*board*) berisi mikrokontroler berukuran sebesar kartu kredit, yang dilengkapi dengan sejumlah pin sebagai sarana komunikasi dengan perangkatlain. Arduino sendiri merupakan mikrokontroler serbaguna yang dapat diprogram, di mana program yang dijalankan pada Arduino biasa disebut dengan istilah *sketch*. Secara umum, Arduino terdiri atas dua komponen utama, yaitu papan sirkuit fisik yang sering disebut sebagai mikrokontroler, serta perangkat lunak atau *IDE* yang berjalan pada komputer dan berfungsi sebagai compiler (Nadziroh, 2021).

Tabel 2. 3 Spesifikasi Arduino Uno

Spesifikasi	Keterangan
<i>Microcontroller</i>	<i>ATmega328P-AU</i>
<i>Operating Voltage</i>	<i>5V</i>
<i>Input Voltage (recommended)</i>	<i>7-12V</i>
<i>Input Voltage (limits)</i>	<i>6-20V</i>
<i>Digital I/O Pins</i>	<i>14 (of which 6 provide provide PWM output)</i>
<i>Analog Input Pins</i>	<i>6</i>
<i>DC Current per I/O Pin</i>	<i>40 mA</i>
<i>DC Current for 3.3V Pin</i>	<i>50 mA</i>
<i>Flash Memory</i>	<i>32 KB of which 0.5 KB used by bootloader</i>
<i>SRAM</i>	<i>2 KB</i>
<i>EEPROM</i>	<i>1 KB</i>
<i>Clock Speed</i>	<i>16 MHz</i>

Tabel 2.3 menunjukkan spesifikasi Arduino Uno yang digunakan sebagai mikrokontroler pada penelitian ini. Arduino Uno menggunakan mikrokontroler *ATmega328P-AU* dengan tegangan operasi 5 V, kecepatan *clock* 16 MHz, memori 32 KB *Flash*, 2 KB *SRAM*, dan 1 KB *EEPROM*. Selain itu, Arduino Uno dilengkapi dengan 14 pin digital, 6 pin analog, serta mendukung komunikasi dengan berbagai perangkat elektronik. Pada penelitian ini, Arduino Uno berfungsi untuk

mengendalikan IR Transmitter dalam mengirimkan kode *HEX*. Berikut pin pada Arduino Uno dapat dilihat pada gambar 2.6.



Gambar 2. 6 PIN Arduino Uno

Tabel 2. 4 PIN Analog Arduino Uno

Pin	Fungsi	Tipe	Deskripsi
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 / GPIO
10	A1	Analog/GPIO	Analog input 1 / GPIO
11	A2	Analog/GPIO	Analog input 2 / GPIO
12	A3	Analog/GPIO	Analog input 3 / GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

Tabel 2.4 menunjukkan fungsi masing-masing pin analog pada Arduino Uno. Selain digunakan sebagai masukan analog melalui pin A0–A5, beberapa pin juga dapat berfungsi sebagai *GPIO* serta mendukung komunikasi I2C melalui pin A4 (*SDA*) dan A5 (*SCL*). Selain itu, terdapat pin catu daya, *ground*, *reset*, *IOREF*, dan *VIN* yang mendukung pengoperasian papan Arduino.

Tabel 2. 5 PIN Digital Arduino Uno

Pin	Fungsi	Tipe	Deskripsi
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Tabel 2.5 menunjukkan fungsi masing-masing pin digital pada Arduino Uno. Pin digital D0–D9 dapat digunakan sebagai *GPIO* untuk menerima maupun mengirimkan sinyal digital, sedangkan pin *SS*, *MOSI*, *MISO*, dan *SCK* digunakan untuk komunikasi *SPI*. Selain itu, tersedia pin *AREF* sebagai referensi tegangan analog, *GND* sebagai *ground*, serta pin *A4/SDA* dan *A5/SCL* yang juga mendukung komunikasi *I2C*. Pada penelitian ini, pin digital Arduino Uno dimanfaatkan untuk mengendalikan IR Transmitter sehingga dapat mengirimkan kode *HEX* ke televisi sesuai perintah yang telah diproses oleh Raspberry Pi.

2.10 Buzzer



Gambar 2. 7 Buzzer

Buzzer merupakan salah satu komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara, dengan prinsip kerja yang pada dasarnya hampir serupa dengan loudspeaker. Buzzer terdiri dari kumparan yang terpasang pada elemen suara, di mana ketika kumparan tersebut dialiri arus listrik, kumparan akan berubah menjadi elektromagnet. Kumparan ini selanjutnya akan bergerak ke arah dalam atau luar, bergantung pada arus listrik yang mengalir serta polaritas magnet yang dihasilkan. Gerakan bolak-balik dari kumparan yang terpasang pada elemen suara inilah yang menyebabkan udara bergetar dan menghasilkan suara. Pada umumnya, buzzer dimanfaatkan sebagai indikator bahwa suatu proses telah selesai dilakukan, atau dapat pula digunakan sebagai indikator adanya kesalahan pada sebuah alat (Nadziroh, 2021).

Buzzer merupakan komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Komponen ini terdiri dari kumparan yang terpasang pada *diafragma*, di mana ketika kumparan tersebut dialiri arus listrik, kumparan akan berubah menjadi elektromagnet. Karena terpasang pada diafragma,

setiap gerakan yang terjadi pada kumparan akan menggerakkan diafragma secara bolak-balik, sehingga menyebabkan udara bergetar dan menghasilkan suara (Laudira, 2020).

Tabel 2. 6 Spesifikasi Buzzer

Spesifikasi	Keterangan
<i>Rated Voltage</i>	<i>6V DC</i>
<i>Operating Voltage</i>	<i>4 to 8V DC</i>
<i>Rated Current*</i>	$\leq 30mA$
<i>Sound Output at 10cm*</i>	$\geq 85dB$
<i>Resonant Frequency</i>	$2300 \pm 300Hz$
<i>Tone</i>	<i>Continuous</i>
<i>Operating Temperature</i>	$-25^{\circ}C$ to $+80^{\circ}C$
<i>Storage Temperature</i>	$-30^{\circ}C$ to $+85^{\circ}C$
<i>Weight</i>	<i>2g</i>

Tabel 2.6 menunjukkan spesifikasi buzzer yang digunakan sebagai perangkat keluaran (output) pada sistem. Buzzer bekerja pada tegangan operasi 4–8 V DC dengan tegangan nominal 6 V DC, arus maksimum 30 mA, dan menghasilkan suara dengan intensitas minimal 85 dB pada frekuensi resonansi 2300 ± 300 Hz. Pada penelitian ini, buzzer berfungsi sebagai indikator suara untuk memberikan notifikasi bahwa sistem telah menerima dan memproses perintah yang diberikan oleh pengguna.

BAB III

DESAIN DAN IMPLEMENTASI

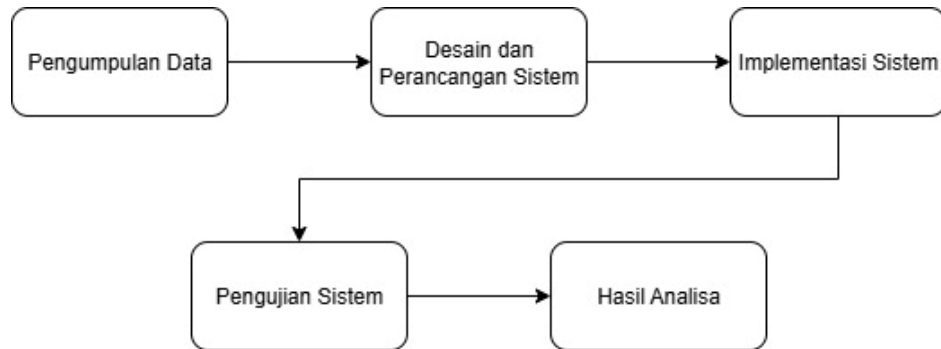
Pada bab ini dibahas secara sistematis seluruh alur penelitian yang dilakukan dalam perancangan sistem kontrol TV. Proses penelitian diawali dengan tahap pengumpulan data, yang menjadi dasar utama dalam pengembangan sistem. Data yang dikumpulkan meliputi kebutuhan pengguna, spesifikasi perangkat, serta pola interaksi yang akan diterapkan. Tahap berikutnya adalah desain dan perancangan sistem, yang bertujuan untuk menghasilkan gambaran menyeluruh mengenai arsitektur sistem, alur kerja, dan komponen yang diperlukan.

Setelah perancangan selesai, dilakukan tahap implementasi, yaitu proses mengubah rancangan menjadi sistem nyata yang dapat dijalankan. Implementasi mencakup pembuatan perangkat keras, pemrograman perangkat lunak, serta integrasi keduanya agar sistem berfungsi sesuai tujuan.

Selanjutnya dilakukan pengujian sistem untuk memastikan bahwa sistem berjalan sesuai spesifikasi, dapat merespons perintah dengan baik, dan memberikan keluaran yang akurat. Pengujian ini juga membantu menemukan potensi kesalahan atau kelemahan yang perlu diperbaiki.

Tahap terakhir adalah analisis dan evaluasi, yang bertujuan untuk menilai tingkat keberhasilan sistem, mengukur kinerjanya, serta memastikan bahwa sistem memberikan manfaat sesuai kebutuhan pengguna. Dengan penyusunan tahap-tahap penelitian yang terstruktur dan terencana ini, diharapkan seluruh proses dapat berjalan lancar dan menghasilkan sistem yang sesuai dengan tujuan penelitian.

Adapun alur penelitian dalam pembuatan sistem kontrol ini dapat dilihat pada gambar 3.1.



Gambar 3. 1 Alur Penelitian

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri dari dua jenis, yaitu data primer dan data sekunder. Data primer digunakan untuk menghasilkan perintah. Sedangkan data sekunder digunakan untuk kata kunci atau *Pattern*. Berikut merupakan penjelasan lebih lanjut mengenai data yang digunakan dalam penelitian ini:

3.2 Data Primer

Data primer diperoleh dari suara user yang diinputkan pada saat pengujian sistem yang nantinya akan menghasilkan perintah dengan format teks yang mana nantinya akan diproses menggunakan *WU-MANBER Algorithm*. Proses perekaman suara tersebut memanfaatkan fitur *Google Speech recognition* yang terintegrasi, dengan konfigurasi khusus yaitu sebuah library untuk mengenali dan memproses suara dalam Bahasa Indonesia. Hasil dari proses pengenalan suara ini adalah konversi input suara menjadi teks. Data teks yang telah dihasilkan selanjutnya akan menjadi masukan utama bagi tahap pemrosesan berikutnya, menggunakan *WU-*

MANBER. *WU-MANBER* akan digunakan untuk menganalisis pola-pola teks yang terdapat dalam data teks tersebut yang kemudian akan menemukan pola yang cocok dan program akan menjalankan tugasnya. Selain itu juga ada data primer berupa kode *hexadecimal* yang diperoleh dari penangkapan sinyal *IR* remot TV menggunakan *IR Receiver*. Adapun data kode *hex* dapat dilihat pada tabel dibawah ini.

Tabel 3. 1 Kode Hexadecimal

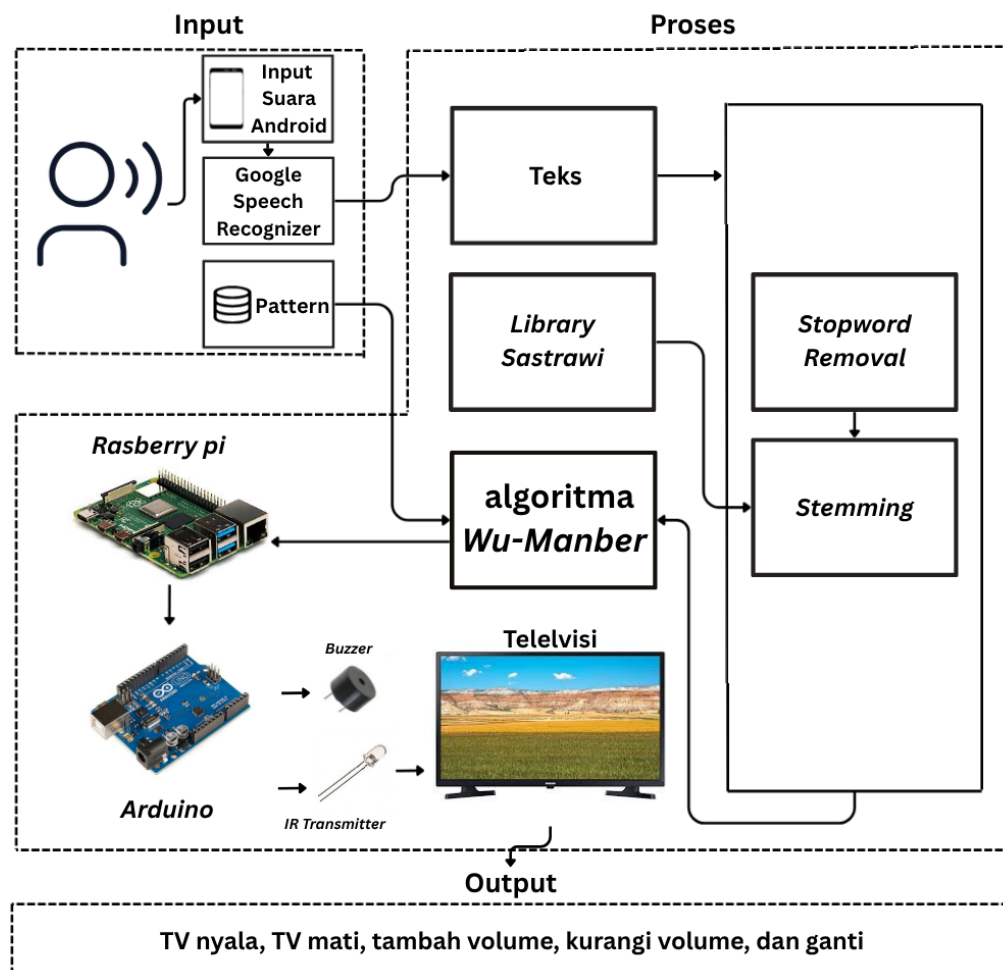
Kode Hex	Fungsi
0xF708FB04	Menyalakan
0xF708FB04	Mematikan
0xFD02FB04	Volume naik
0xFC03FB04	Volume turun
0xF609FB04	Mute
0x837CFB04	Menu
0xBF40FB04	Atas
0xBE41FB04	Bawah
0xF906FB04	Kanan
0xF807FB04	Kiri
0xBB44FB04	Pilih
0xBC43FB04	Pengaturan
0xD728FB04	Kembali
0xA45BFB04	Keluar
0xAC53FB04	Program
0xF40BFB04	Masukan

3.3 Data Sekunder

Data sekunder diperoleh dari data *Pattern* yang diinputkan sebelumnya oleh peneliti. Data sekunder ini menjadi pemeran utama dalam pencocokan (*matching*) dengan perintah suara, di mana sistem akan membandingkan input suara yang telah dikonversi menjadi teks dengan berbagai *Pattern* yang telah ditetapkan. Adapun perintah suara yang dijadikan sebagai pencocok teks yaitu berupa, “Nyala”, “Mati”, “Tambah”, “Kurang”, “Maju”, “Mundur”.

3.4 Desain dan Perancangan Sistem Kontrol

Pada penelitian ini, sistem kontrol terdiri dari tiga tahapan utama yaitu *input*, proses, output. Adapun desain sistem dalam penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3. 2 Desain Sistem

3.5 Tahapan Input

Tahapan *input* merupakan tahapan penambahan data. Pertama yaitu input berupa *Pattern* atau pola yang digunakan sebagai acuan untuk pencarian pencocokan teks. Kemudian yang kedua yaitu *input* pada sistem ini berupa perintah suara yang akan diterima oleh perangkat Raspberry Pi. Ketika pengguna

mengucapkan perintah suara, maka sistem akan memanfaatkan fitur canggih yaitu *Google Speech recognition*. Kemudian perangkat (*Android*) akan otomatis merekam perintah yang sudah diberikan yang mana kemudian fitur tersebut akan mengonversi gelombang suara yang telah diinputkan menjadi sebuah teks.

3.6 Google Speech Reconition

Dalam penerapan sistem kontrol TV menggunakan suara ini, komponen utama pada tahap input data atau suara 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 ucapan manusia (audio) menjadi teks tertulis (teks).

Prinsip kerja *Google Speech recognition* melibatkan beberapa tahapan kompleks. Pertama, input audio dari mikrofon perangkat pengguna (*Android*) akan ditangkap dan diubah menjadi teks digital. Data teks digital ini kemudian dikirimkan melalui jaringan yang terhubung di jaringan yang sama ke server local Raspberry pi 3. 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 dijelaskan.

3.7 Tahapan Proses

Tahapan proses ini merupakan upaya untuk menghasilkan sesuatu yang baru atau memperbarui sistem yang sudah ada guna meningkatkan kinerjanya sehingga dapat mencapai hasil yang diinginkan (B. B. Putri & Ernawati, 2015). Tahapan

pemrosesan berfungsi untuk menginterpretasikan dan menganalisis data *input* yang telah diperoleh dari pengguna. Pada tahap ini, Raspberry Pi mencocokkan hasil perintah suara dengan daftar perintah yang telah diprogram sebelumnya. Untuk memastikan proses pencocokan berlangsung cepat dan efisien, digunakan algoritma *WU-MANBER*. Algoritma ini memungkinkan sistem mengenali kata kunci dari perintah suara dengan akurasi tinggi, sehingga respon yang diberikan sesuai dengan instruksi pengguna.

Dalam perancangan sistem kontrol ini, dilakukan *preprocessing text* yang mencakup beberapa tahap, yaitu *stopword removal*, dan *Stemming*. Tahapan ini bertujuan untuk membersihkan dan menyiapkan data teks agar lebih terstruktur sehingga mudah diproses. Setelah teks melalui tahap *preprocessing*, data tersebut siap digunakan untuk proses pencocokan pola menggunakan algoritma *WU-MANBER*. Proses pencocokan pola dilakukan dengan mencocokkan hasil teks yang sudah diproses dengan *Pattern* yang telah didefinisikan di dalam kode program. Dengan demikian, sistem dapat mengenali perintah dengan lebih akurat dan menjalankan perintah sesuai instruksi pengguna. Berikut adalah penjelasan rinci mengenai tahapan-tahapan proses yang digunakan dalam perancangan sistem ini.

3.8 Stopword Removal

Stopword removal merupakan tahapan dalam proses penyaringan kata-kata penting dari data yang diperoleh, dengan cara menghilangkan kata-kata yang dianggap tidak diperlukan atau tidak memberikan kontribusi makna yang signifikan (Zhafira *et al.*, 2021). *Stopword removal* dilakukan dengan tujuan untuk menghapus daftar kata-kata umum yang tidak memiliki makna penting atau tidak mengandung

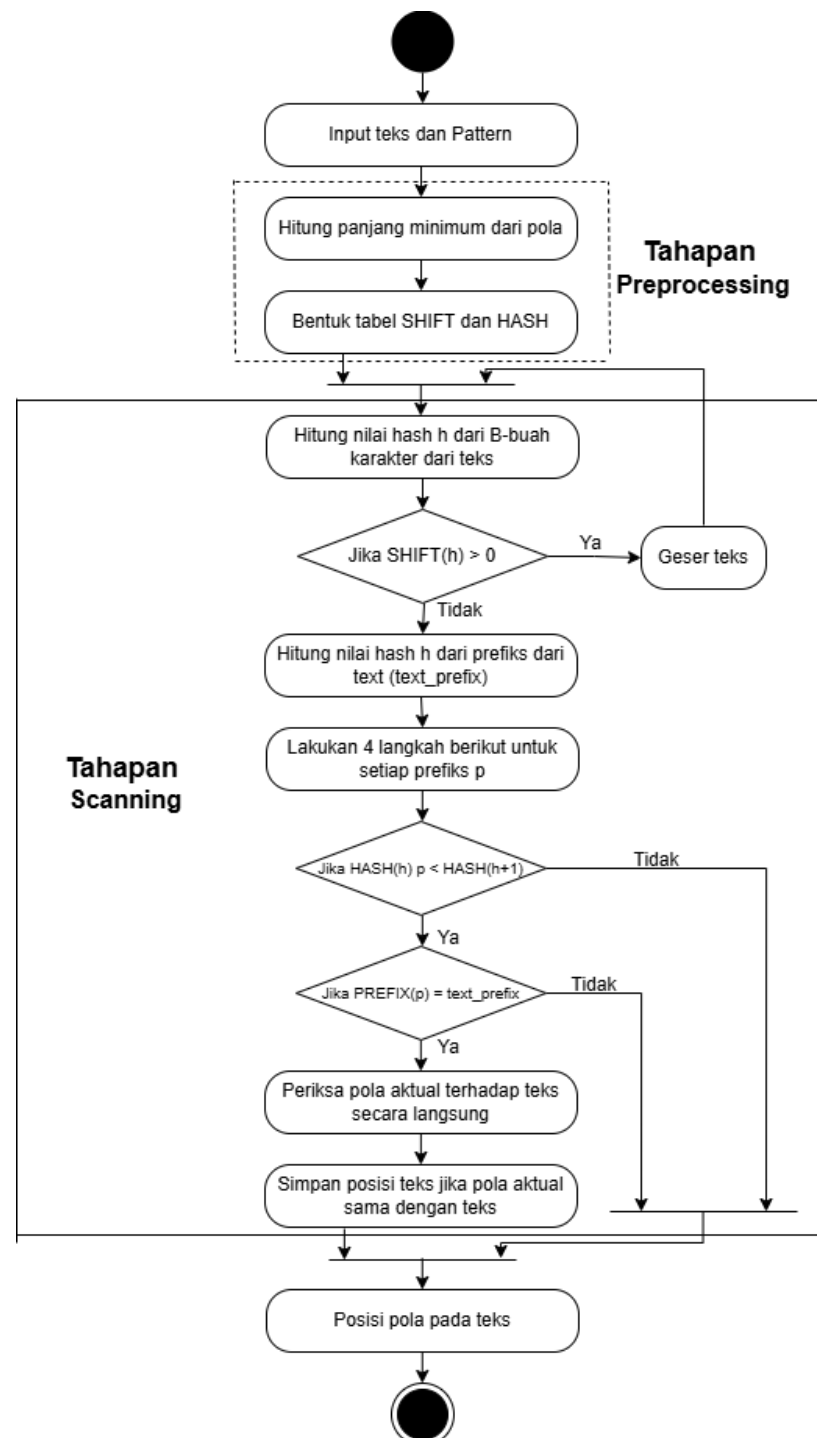
informasi yang dibutuhkan, sehingga dapat mengurangi jumlah kata yang perlu disimpan oleh sistem (V. A. Putri, 2023).

3.9 Stemming

Stemming merupakan suatu proses pemetaan terhadap berbagai variasi bentuk morfologi suatu kata, yang kemudian dikembalikan ke bentuk dasarnya (*root word*) (Zhafira *et al.*, 2021). *Stemming* dilakukan dengan tujuan untuk mengubah kata yang memiliki imbuhan maupun kata berulang menjadi satu bentuk kata dasar (V. A. Putri, 2023).

3.10 Pencocokan Algoritma WU-MANBER

Algoritma *WU-MANBER* memiliki tahapan-tahapan sistematis dalam penyelesaian masalah. Berikut cara kerja *WU-MANBER* dapat dilihat pada gambar 3.3.



Gambar 3. 3 Diagram Activity

Sementara itu (B. B. Putri & Ernawati, 2015) menyebutkan bahwa algoritma *WU-MANBER* mempunyai dua tahap pemrosesan yaitu *preprocesssing* dan *scanning*, tahapan pertama dalam proses pencocokan string menggunakan

algoritma *WU-MANBER* adalah tahap persiapan (preprocessing) terhadap kumpulan pola yang akan dicocokkan dengan teks. Pada tahap ini, dibangun tiga buah tabel utama, yaitu tabel *SHIFT*, tabel *HASH*, dan tabel *Prefix*. Diasumsikan bahwa B merupakan *Block Size* dengan nilai $B = 2$ atau $B = 3$, teks dinyatakan sebagai $T = t_1, t_2, \dots, t_n$ dengan n sebagai panjang dari T , p merupakan pola asli, k adalah panjang pola, dan m adalah minimum length dari pola, sehingga himpunan pola P dapat dinyatakan sebagai $P = k[p] - m$, atau $P = \{P_1, P_2, \dots, P_j\}$, dengan j menyatakan banyaknya pola dalam himpunan P .

Pembentukan tabel *SHIFT* dilakukan dengan mengasumsikan bahwa kunci *SHIFT* $X = x_1 \dots x_s$ merupakan sejumlah s buah karakter sepanjang B pada teks yang akan dibaca, di mana X dipetakan ke posisi i dalam tabel *SHIFT*, yaitu $HASH(X) = i$. Berdasarkan pemetaan tersebut, terdapat dua kasus yang dapat terjadi. Kasus pertama adalah ketika X tidak dapat ditemukan sebagai sebuah substring pada sembarang pola p . Pada kasus ini, dapat dilakukan pergeseran sebanyak $m - B + 1$ buah karakter pada teks, karena apabila dilakukan pergeseran dengan nilai yang lebih kecil, proses pembandingan antara B buah karakter terakhir dari teks dengan substring dari salah satu pola akan menghasilkan ketidakcocokan. Oleh karena itu, nilai $m - B + 1$ inilah yang kemudian disimpan sebagai nilai *SHIFT*. Kasus kedua adalah ketika X ditemukan pada beberapa pola, di mana kemunculan X dicari pada posisi paling kanan dari setiap pola. Apabila q adalah nilai indeks paling besar dari X , maka diasumsikan bahwa X berada pada posisi q dari pola tersebut dan tidak lebih besar dari q pada pola lainnya, sehingga nilai $m - q$ disimpan sebagai *SHIFT*.

Pembentukan tabel *HASH* dilakukan berdasarkan prinsip bahwa selama nilai pergeseran lebih besar daripada 0, pergeseran dapat dilakukan dan proses pembacaan dapat dilanjutkan. Namun, ketika nilai pergeseran sama dengan 0, terdapat kemungkinan bahwa substring yang bersangkutan pada teks cocok dengan beberapa pola pada daftar pola. Untuk mencegah perbandingan substring dengan setiap pola pada daftar pola satu per satu, digunakan sebuah teknik *HASH* guna meminimalkan jumlah pola yang perlu dibandingkan. Sebelumnya telah dihitung sebuah pemetaan dari B buah karakter ke sebuah nilai integer yang digunakan sebagai indeks ke tabel *SHIFT*, dan nilai integer yang sama juga digunakan sebagai indeks ke tabel *HASH*. Entri ke-*i* dari tabel *HASH*, yaitu *HASH*[*i*], terdiri dari sebuah penunjuk (pointer) ke daftar pola di mana B buah karakter di-*HASH* ke *i*. Tabel *HASH* secara khusus bersifat jarang (sparse) karena hanya menyimpan pola sementara dari tabel *SHIFT* yang menyimpan semua kemungkinan string berukuran B. Hal ini merupakan penggunaan memori yang kurang efisien, tetapi memungkinkan penggunaan kembali fungsi *HASH* sehingga dapat menghemat banyak waktu komputasi. Apabila *h* adalah nilai *HASH* dari *Suffix* pada teks dan diasumsikan *SHIFT*[*h*] = 0, maka nilai dari *HASH*[*h*] adalah sebuah penunjuk *p* yang menunjuk ke dua tabel terpisah secara bersamaan, yaitu daftar penunjuk ke pola (*PAT_POINT*) yang diurutkan berdasarkan nilai *HASH* dari B buah karakter terakhir setiap pola. Penunjuk *p* menunjuk ke awal dari daftar pola dengan nilai *HASH* sebesar *h*, dan untuk menemukan akhir dari daftar tersebut, dilakukan penambahan terhadap penunjuk hingga nilainya sama dengan nilai pada *HASH*[*h*+1], karena keseluruhan daftar telah diurutkan berdasarkan nilai *HASH*.

Sebaliknya, jika $SHIFT[h] \neq 0$, maka $HASH[h] = HASH[h+1]$, karena tidak ada pola yang memiliki *Suffix* yang di-HASH ke h . Sebagai tambahan, disimpan pula sebuah tabel lain yang disebut tabel *Prefix*.

Pembentukan tabel *Prefix* diperlukan karena *Suffix* tidak hanya sering muncul pada teks, tetapi juga dapat muncul pada beberapa pola sekaligus. Kondisi ini menyebabkan terjadinya tabrakan (collision) pada tabel *HASH*, di mana seluruh pola dengan *Suffix* yang sama akan dipetakan ke entri yang sama pada tabel tersebut. Ketika *Suffix* yang bersangkutan ditemukan pada teks dengan nilai *SHIFT* sebesar 0, setiap pola dengan *Suffix* tersebut harus dicoba satu per satu untuk melihat kecocokannya dengan teks. Untuk mempercepat proses ini, diperkenalkan tabel *Prefix*, yang memetakan B' buah karakter pertama dari semua pola, dengan nilai $B' = 2$ yang disarankan oleh penemu algoritma sebagai nilai yang baik. Ketika ditemukan sebuah *SHIFT* bernilai 0 dan proses akan menuju ke tabel *HASH* untuk menentukan ada tidaknya kecocokan, nilai pada tabel *Prefix* akan diperiksa terlebih dahulu. Untuk setiap *Suffix*, tabel *HASH* tidak hanya mencakup daftar seluruh pola dengan *Suffix* tersebut, tetapi juga mencakup nilai *HASH* dari prefiksnya. Prefiks yang bersangkutan pada teks dihitung dengan menggeser $m - B'$ buah karakter ke kiri, kemudian digunakan untuk memfilter pola yang memiliki *Suffix* sama tetapi prefiks berbeda. Metode pemfilteran ini efektif karena kecil kemungkinannya terdapat pola berbeda yang memiliki *Prefix* dan *Suffix* yang sama sekaligus. Pendekatan ini juga merupakan bentuk penyeimbangan yang baik, karena kerja tambahan untuk menghitung fungsi *HASH* dari prefiks hanya dilakukan ketika nilai

SHIFT sebesar 0, yaitu pada kondisi di mana terdapat banyak pola dengan kemiripan tinggi.

Setelah tahap preprocessing selesai, dilakukan tahap scanning, yaitu proses pencocokan pola terhadap teks menggunakan tabel-tabel yang telah dibangun sebelumnya. Proses scanning dimulai dengan mengambil posisi tempat pointer berhenti sebanyak m buah karakter teks, yang disebut sebagai *match window*. Selanjutnya, diambil B buah karakter *Suffix* dari *match window* tersebut, kemudian nilai dari tabel *SHIFT* diambil sesuai dengan *Suffix* yang bersangkutan. Apabila nilai *SHIFT* tidak ditemukan, maka nilai *SHIFT* diatur sebesar $m - B + 1$ karakter ke arah kanan. Namun, apabila nilai *SHIFT* yang ditemukan adalah 0, maka dicari posisi pada daftar tabel *HASH* yang menunjukkan kemungkinan terdapat kecocokan dengan teks pada *match window*. Pada kondisi ini, diambil pula B' buah karakter *Prefix* dari *match window* dengan menggeser $m - B'$ karakter ke kiri. Apabila B' karakter tersebut sesuai dengan nilai pada tabel *Prefix* milik pola yang bersangkutan, maka pola aktual pada daftar tabel *HASH* akan diperiksa lebih lanjut. Jika pola aktual tersebut ditemukan cocok dengan teks, maka teks yang cocok akan disimpan sebagai hasil pencocokan. Setelah itu, pointer akan digeser sebanyak satu karakter ke arah kanan, dan seluruh proses dari pengambilan *match window* hingga pemeriksaan kecocokan diulang kembali hingga seluruh teks selesai ditelusuri. Apabila hasil pencocokan yang diperoleh sesuai dengan salah satu perintah yang tersedia, Raspberry Pi akan memutuskan aksi yang harus dijalankan. Tahap ini juga dapat mencakup proses validasi, misalnya memeriksa apakah perintah yang

diberikan benar dan dapat dijalankan, atau mengabaikan perintah yang tidak dikenali oleh sistem.

3.11 Contoh Pencocokan Algoritma *WU-MANBER*

Sebagai contoh penerapan algoritma *WU-MANBER*, langkah pertama yang dilakukan adalah menentukan parameter yang akan digunakan. Misalkan terdapat kumpulan pola atau kata kunci perintah yang ingin dideteksi oleh sistem, yaitu $P1 = \text{"nyala"}$ dengan panjang $k = 5$, $P2 = \text{"matikan"}$ yang apabila ditinjau dengan *window* 5 karakter dari posisi tergeser menghasilkan "matik", "atika", dan "tikan", serta $P3 = \text{"nyalakan"}$ dengan panjang $k = 8$. Berdasarkan kumpulan pola tersebut, nilai m atau minimum length ditentukan sebagai panjang pola terpendek, yaitu 5, yang berasal dari pola "nyala". Selanjutnya, nilai B atau *Block Size* ditetapkan sebesar 2, dan nilai B' atau *Prefix length* juga ditetapkan sebesar 2. Teks yang akan diproses dalam contoh ini adalah "tolong nyalakan tv", yang kemudian diindeks per karakter mulai dari posisi 1, dengan spasi diperlakukan sebagai karakter biasa, sehingga diperoleh rangkaian karakter t-o-l-o-n-g-(spasi)-n-y-a-l-a-k-a-n-(spasi)-t-v dengan total panjang teks $n = 18$.

Pada tahap preprocessing, dilakukan pembentukan tabel *SHIFT* dengan cara mengambil $B = 2$ karakter terakhir dari setiap jendela sepanjang m karakter pada masing-masing pola, mengingat nilai *SHIFT* dihitung berdasarkan *window* selebar m . Karena $m = 5$, maka peninjauan dilakukan terhadap 5 karakter pertama dari setiap pola sesuai definisi *window*, sehingga pola $P1 = \text{"nyala"}$ memiliki *window* "nyala" karena k sama dengan m , sedangkan pola $P3 = \text{"nyalakan"}$ memiliki beberapa *window*, yaitu "nyala", "yalak", "alaka", dan "lakan". Untuk setiap

window tersebut, diambil 2 karakter terakhirnya, kemudian dihitung posisi q dari akhir *window* di dalam pola asli, sehingga nilai *SHIFT* dapat diperoleh melalui rumus $SHIFT = m - q$. Pada pola "nyala" dengan *window* "nyala", *Suffix* yang diperoleh adalah "la", dengan posisi q dari kanan pola berada di posisi 5 atau tepat di akhir pola, sehingga nilai $SHIFT = m - q = 5 - 5 = 0$. Adapun untuk pasangan dua karakter yang tidak muncul pada *window* manapun di seluruh pola, nilai *SHIFT* ditetapkan sebesar $m - B + 1 = 5 - 2 + 1 = 4$. Dengan demikian, tabel *SHIFT* yang terbentuk memuat entri X berupa dua karakter, misalnya "la", dengan nilai $SHIFT[HASH(X)] = 0$.

Pembentukan tabel *HASH* dilakukan untuk setiap entri dengan nilai *SHIFT* sebesar 0, di mana tabel ini menyimpan daftar pointer yang mengarah ke pola-pola dengan *Suffix* m -karakter yang berakhiran dua karakter tertentu. Sebagai contoh, untuk h sebagai *HASH* dari *Suffix* "la", maka $HASH[h]$ menunjuk ke pola $P1$ ("nyala") dan $P3$ pada *window* "nyala". Selanjutnya, pembentukan tabel *Prefix* dilakukan dengan menyimpan $B' = 2$ karakter pertama dari setiap *window* pola yang telah disebutkan sebelumnya, yaitu *Prefix* "ny" untuk *window* "nyala" milik $P1$, *Prefix* "ny" untuk *window* "nyala" milik $P3$, *Prefix* "ya" untuk *window* "yalak" milik $P3$, *Prefix* "al" untuk *window* "alaka" milik $P3$, dan *Prefix* "la" untuk *window* "lakan" milik $P3$.

Setelah tahap preprocessing selesai, dilakukan tahap scanning terhadap teks "tolong nyalakan tv". Pointer mula-mula diposisikan pada posisi awal teks ditambah $m - 1$, karena *match window* pertama mencakup m karakter pertama dari teks. Pada iterasi pertama, pointer berada di posisi 5 dengan *match window* berupa

karakter ke-1 hingga ke-5, yaitu "tolon". *Suffix* sepanjang $B = 2$ karakter dari *window* tersebut adalah "on", dan ketika dicek pada tabel *SHIFT*, *Suffix* ini tidak ditemukan, sehingga nilai *SHIFT* ditetapkan sebesar $m - B + 1 = 5 - 2 + 1 = 4$ dan pointer bergerak ke posisi $5 + 4 = 9$.

Pada iterasi kedua, pointer berada di posisi 9 dengan *match window* berupa karakter ke-5 hingga ke-9, yaitu "ng ny". *Suffix* dua karakter dari *window* ini adalah "ny", yang ketika dicek pada tabel *SHIFT* tidak ditemukan secara langsung sebagai *Suffix* pola, karena "ny" pada kumpulan pola yang digunakan berperan sebagai *Prefix* dan bukan sebagai *Suffix*. Oleh karena itu, nilai *SHIFT* kembali ditetapkan sebesar 4, sehingga pointer bergerak ke posisi $9 + 4 = 13$.

Pada iterasi ketiga, pointer berada di posisi 13 dengan *match window* berupa karakter ke-9 hingga ke-13, yaitu "yalak". *Suffix* dua karakter dari *window* ini adalah "ak", yang ketika dicek pada tabel *SHIFT* ditemukan dengan nilai *SHIFT* = 0, sehingga proses dilanjutkan ke tabel *HASH*. Pencarian *HASH*("ak") menunjukkan bahwa entri tersebut menunjuk ke pola P3 ("nyalakan") pada *window* "yalak". Selanjutnya, diambil *Prefix* $B' = 2$ karakter dari teks dengan menggeser $m - B' = 5 - 2 = 3$ karakter ke kiri dari *window*, yaitu karakter pada posisi 9 hingga 10, sehingga diperoleh "ny". Ketika *Prefix* ini dibandingkan dengan *Prefix*["yalak"] yang bernilai "ya", hasilnya tidak cocok, sehingga proses pemfilteran berhasil mencegah perbandingan penuh yang tidak diperlukan, dan pointer bergerak satu posisi ke kanan menjadi 14.

Pada iterasi keempat, pointer berada di posisi 14 dengan *match window* berupa karakter ke-10 hingga ke-14, yaitu "alaka". *Suffix* dua karakter dari *window*

ini adalah "ka", yang ketika dicek pada tabel *SHIFT* ditemukan dengan nilai *SHIFT* = 0, sehingga proses dilanjutkan ke tabel *HASH*. Pencarian *HASH*("ka") menunjukkan bahwa entri tersebut menunjuk ke pola P3 pada *window* "alaka". *Prefix* B' dari teks kemudian diambil dengan menggeser 3 karakter ke kiri dari *window*, yaitu karakter pada posisi 10 hingga 11, sehingga diperoleh "al". Ketika *Prefix* ini dibandingkan dengan *Prefix*["alaka"] yang juga bernilai "al", hasilnya cocok, sehingga proses dilanjutkan dengan memeriksa pola aktual pada daftar tabel *HASH*. Pemeriksaan dilakukan dengan membandingkan keseluruhan *window* teks "alaka" terhadap pola penuh P3 = "nyalakan" pada posisi yang bersesuaian, yaitu dengan menggeser verifikasi ke kiri mulai dari posisi 8. Hasil verifikasi menunjukkan bahwa teks pada posisi 8 hingga 15, yaitu "nyalakan", cocok sepenuhnya dengan pola P3, sehingga hasil pencocokan ini disimpan sebagai temuan pola "nyalakan" pada posisi 8 sampai 15 dari teks, dan pointer kemudian bergerak satu posisi ke kanan menjadi 15.

Pada iterasi kelima, pointer berada di posisi 15 dengan *match window* berupa karakter ke-11 hingga ke-15, yaitu "lakan". *Suffix* dua karakter dari *window* ini adalah "an", yang ketika dicek pada tabel *SHIFT* ditemukan dengan nilai *SHIFT* = 0, sehingga proses dilanjutkan ke tabel *HASH*. Pencarian *HASH*("an") menunjukkan bahwa entri tersebut menunjuk ke pola P2 dan P3. *Prefix* dari teks kemudian diambil pada posisi 11 hingga 12, yaitu "la". Ketika *Prefix* ini dibandingkan dengan *Prefix* P3["lakan"] yang bernilai "la", hasilnya cocok, sedangkan ketika dibandingkan dengan *Prefix* P2["tikan"] yang bernilai "ti", hasilnya tidak cocok sehingga pola P2 terfilter dari proses verifikasi. Pemeriksaan

pola aktual P3 ("nyalakan") yang telah ditemukan cocok pada iterasi sebelumnya menunjukkan hasil yang sama, sehingga tidak ditemukan pencocokan baru pada iterasi ini, dan pointer bergerak satu posisi ke kanan menjadi 16. Proses scanning kemudian berlanjut dengan cara yang sama hingga pointer mencapai akhir teks pada posisi $n = 18$, dan tidak ditemukan kemunculan pola lain selain "nyalakan" yang telah teridentifikasi sebelumnya.

3.12 Tahapan Output

Tahapan output merupakan hasil dari proses pengolahan data yang telah dilakukan pada tahap sebelumnya. Jika perintah suara berhasil dikenali dan diproses, Raspberry Pi akan mengirimkan data ke arduino kemudian arduino akan mengirimkan sinyal sesuai perintah. Pada sistem ini, keluaran berupa sinyal inframerah (IR) yang dikirim melalui IR Transmitter untuk mengontrol televisi sesuai perintah pengguna.

3.13 Perancangan Aplikasi

Adapun rancangan aplikasi yang digunakan dalam penelitian ini dapat dilihat pada gambar.



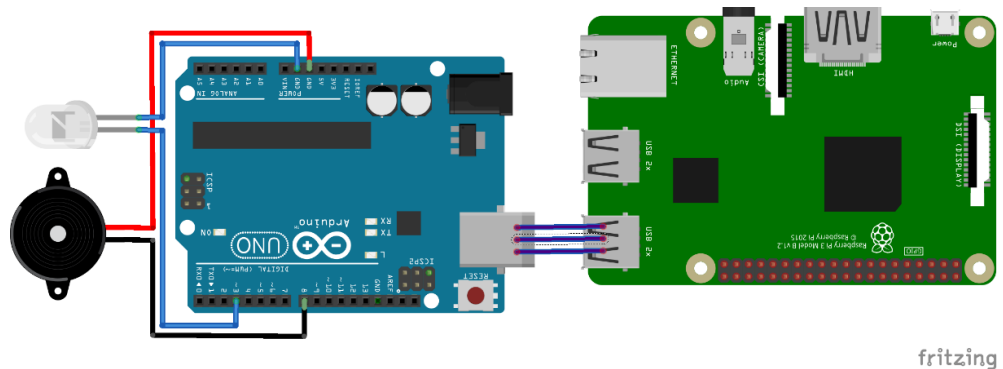
Gambar 3. 4 Rancangan Aplikasi

Keterangan:

1. Mikrofon
2. Hasil perintah suara yang telah diubah menjadi teks

3.14 Perancangan Alat Sistem Kontrol

Adapun rancangan alat yang digunakan dalam penelitian ini terdiri Raspberry Pi, IR Transmitter yang dapat dilihat pada gambar 3.5



Gambar 3. 5 Rancangan Komponen

Gambar tersebut menunjukkan rancangan komponen yang terdiri dari Raspberry Pi, dan IR Transmitter yang telah dirakit. IR Transmitter berfungsi mengirimkan kode heksadesimal ke televisi setelah menerima perintah. Setiap tombol pada remote TV memiliki kode heksadesimal masing-masing, yang diperoleh dengan menangkap sinyal dari remote menggunakan sensor IR. Kode-kode tersebut kemudian dimasukkan ke dalam program sistem agar dapat menjalankan perintah sesuai fungsinya.

Raspberry Pi berperan sebagai otak yang menerima *input* suara melalui perangkat (Android), memprosesnya, lalu mengeksekusi perintah. Rancangan ini memungkinkan sistem mendeteksi perintah suara dan memberikan respons, seperti menyalakan atau mematikan TV, mengganti saluran, serta menambah atau mengurangi volume.

3.15 Implementasi Sistem

Implementasi sistem dilakukan dengan merakit komponen utama, yaitu Raspberry Pi, arduino, dan IR transmitter, yang kemudian ditempatkan pada sebuah kotak pelindung agar rapi dan aman. Raspberry Pi dan arduino difungsikan sebagai pusat kendali, dihubungkan dengan perangkat (*Android*) untuk menangkap suara, dan dengan IR Transmitter untuk mengirimkan sinyal ke televisi.

Tahap berikutnya adalah pembuatan program menggunakan bahasa Python untuk mengintegrasikan seluruh komponen. Program ini menangkap suara dari perangkat (*Android*), kemudian *Google Speech recognition* akan mengonversinya menjadi teks, mencocokkannya dengan pola perintah yang telah ditentukan menggunakan algoritma *WU-MANBER*, kemudian mengirimkan sinyal IR yang sesuai ke televisi. Setelah proses implementasi selesai, dilakukan pengujian dengan berbagai perintah seperti menyalakan dan mematikan TV, mengontrol , serta mengatur volume.

3.16 Pengujian Sistem

Setelah tahap implementasi selesai, dilakukan pengujian sistem untuk memastikan seluruh komponen bekerja sesuai dengan rancangan. Pengujian dilakukan dengan memberikan berbagai perintah suara, seperti menyalakan dan mematikan TV, mengganti saluran, serta menaikkan dan menurunkan volume. Setiap perintah diuji berulang kali untuk memastikan konsistensi respons dan keandalan sistem. Selain itu, pengujian ini juga bertujuan untuk mengidentifikasi kemungkinan terjadinya error, baik pada proses perintah suara maupun pada pengiriman sinyal IR ke televisi. Hasil pengujian digunakan sebagai dasar evaluasi

untuk memperbaiki kekurangan dan meningkatkan kinerja sistem agar sesuai dengan tujuan penelitian.

3.17 Perhitungan *Error* Sistem

Pada tahap ini dilakukan pengujian sistem untuk mengukur tingkat kesalahan (*error rate*) dalam proses pengenalan dan pemrosesan perintah suara. Pengujian ini bertujuan untuk mengevaluasi sejauh mana sistem dapat menjalankan perintah sesuai dengan yang diharapkan, sekaligus mengidentifikasi kesalahan yang terjadi selama proses pengenalan atau eksekusi perintah. Hasil pengujian akan memberikan gambaran mengenai kinerja sistem, apakah sudah memenuhi standar akurasi yang ditentukan atau masih memerlukan perbaikan. Untuk mengetahui besarnya tingkat kesalahan, digunakan perhitungan rata-rata *error* dari setiap pengujian. Rumus perhitungan rata-rata error ditunjukkan pada persamaan berikut.

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

Keterangan:

μ : Rata-rata error dalam bentuk persen (%)
 n : Jumlah percobaan
 $\sum Ei$: Jumlah kesalahan dari semua percobaan

Proses pengujian pada penelitian ini dilakukan untuk merekap hasil keberhasilan maupun kegagalan pada setiap percobaan. Dalam setiap uji coba, sistem dianggap berhasil apabila perintah suara dapat dikenali dengan tepat dan dijalankan sesuai fungsi yang diharapkan. Sebaliknya, sistem dianggap gagal apabila terjadi kesalahan dalam pengenalan perintah atau kesalahan dalam pelaksanaan perintah. Karena tujuan pengujian adalah untuk menghitung rata-rata *error* sistem, maka setiap pengujian yang berhasil diberikan nilai 0, sedangkan

setiap pengujian yang gagal diberikan nilai 1. Nilai-nilai tersebut kemudian dihitung rata-ratanya untuk memperoleh tingkat error sistem secara keseluruhan.

Hasil pengujian ini digunakan untuk menghitung rata-rata *error* sistem. Perhitungan dilakukan dengan membagi jumlah kesalahan yang terjadi dengan total pengujian, kemudian dikalikan 100% untuk memperoleh persentase rata-rata *error*. Data ini penting untuk mengetahui tingkat keandalan sistem serta menjadi acuan dalam menentukan kebutuhan perbaikan lebih lanjut.

Tabel dibawah ini merupakan contoh pengujian eror sistem dalam 11 kali percobaan.

Tabel 3. 2 contoh pengujian error sistem

Percobaan	Perintah	<i>Pattern</i>	Hasil	<i>Error</i>
1.	Nyalakan TV	Nyala	Benar	0
2.	Tolong nyalakan TV	Nyala	Benar	0
3.	TV nyala	Nyala	Benar	0
4.	Matikan TV	Mati	Salah	1
5.	Tolong matikan TV	Mati	Benar	0
6.	TV Mati	Mati	Benar	0
7.	Geser ke kanan	kanan	salah	1
8.	Kanan	Kanan	Benar	0
9.	Atas	Atas	Benar	0
10.	Kiri	Kiri	Salah	1
11.	Bawah Lagi	Bawah	Salah	1
Jumlah Kesalahan				4 (36,36%)

Perhitungan rata-rata eror:

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

$$\mu = \frac{4}{11} \times 100\%$$

$$\mu = 36,36\%$$

Pada tabel diatas terdapat beberapa kesalahan yang dilakukan sistem. adapun kesalahan atau rata-rata eror pada percobaan diatas adalah sebesar 36,36%.

Gambaran ini memberikan tentang seberapa akurat sistem dalam mengenali perintah suara, sekaligus menjadi bahan evaluasi untuk meningkatkan kinerja sistem di penelitian selanjutnya.

3.18 Hasil Analisa

Setelah nantinya akan dilakukan serangkaian pengujian, peneliti akan mengidentifikasi faktor-faktor yang memengaruhi kinerja sistem. Hasil pengujian tersebut menjadi dasar untuk memahami secara lebih mendalam kelebihan serta kekurangan yang dimiliki sistem.

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Hasil Implementasi Sistem Kontrol

Sistem kontrol yang diterapkan dalam sistem ini sesuai dengan rancangan awal yang sudah dijelaskan pada bab sebelumnya. pada sistem kontrol ini terdapat dua bagian utama, yaitu *software* atau perangkat lunak dan *hardware* atau perangkat keras yang saling terhubung. Pada bagian perangkat lunak, penerapan dimulai dari aplikasi *android* yang dibuat menggunakan *MIT APP Inventor*. Aplikasi ini digunakan untuk menerima perintah suara dari pengguna yang sudah diubah menjadi teks, kemudian akan dikirimkan server lokal berbasis *PHP* di Raspberry Pi.

Dari sistem perangkat keras ini program dijalankan menggunakan Raspberry Pi yang terhubung ke arduino, kemudian dihubungkan lagi dengan IR Transmitter dan Buzzer, yang mana Buzzer digunakan untuk indikator bahwa sistem sudah berhasil menerima masukan kemudian memproses dan mengirimkan sinyal. Berikut penjelasan secara jelas mengenai tentang perangkat keras dan perangkat lunak pada sistem yang digunakan:

4.1.1 Perangkat Lunak

Perangkat lunak atau *software* pada sistem ini merupakan bagian yang digunakan untuk mengolah masukan atau input berupa perintah suara. Yang mana perintah tersebut menjadi perintah untuk mengontrol TV yang kemudian akan diproses oleh perangkat keras. Aplikasi di buat menggunakan *software MIT App*

Inventor. Dalam aplikasi ini terdapat fitur pengenalan suara menggunakan *Google Speech recognition* untuk menangkap ucapan suara dari pengguna yang kemudian akan diubah menjadi bentuk teks seperti “tolong nyalakan TV”.



Gambar 4. 1 Halaman Utama

Gambar diatas merupakan halaman utama pada aplikasi yang terdapat beberapa fitur yang ditampilkan. Berikut penjelasannya:

1. Ikon *microphone* yang berfungsi untuk mengubah perintah suara menjadi teks.
2. Label ini berfungsi untuk menampilkan teks.

4.1.2 Implementasi Komunikasi

Dalam sistem ini, komunikasi antara perangkat lunak dan perangkat keras dilakukan melalui satu jaringan internet yang sama. Penerapan komunikasi ini memungkinkan interaksi antara aplikasi yang berjalan sebagai perangkat lunak pengguna dengan Raspberry Pi sebagai pengendali proses, yang kemudian

diteruskan ke arduino untuk memproses dan mengirimkan sinyal kendali kepada TV.

Proses dimulai dengan mengaktifkan *speech recognizer* yang berfungsi untuk menangkap dan mengubah ucapan pengguna menjadi teks. Ketika pengguna selesai berbicara, kemudian akan diubah menjadi sebuah teks secara otomatis dan akan menghasilkan sebuah teks hasil dari proses pengenalan suara tersebut.

Teks yang tersimpan kemudian di proses dan ditampilkan kepada pengguna di layar aplikasi. Selain itu, *speech recognizer* mengaktifkan proses *looping*, sehingga sistem dapat terus mendengarkan dan mengenali suara pengguna secara terus-menerus tanpa harus dihidupkan ulang secara manual.

Setelah perintah suara berhasil dikenali dan diubah menjadi bentuk teks, sistem kemudian mempersiapkan proses pengiriman data. Pada proses ini teks akan dikirim melalui alamat *IP* lokal dari perangkat Raspberry Pi yang terhubung dalam satu jaringan yang sama dengan perangkat lunak yang menjalankan aplikasi.

Setelah alamat *IP* tujuan ditetapkan, teks hasil pengenalan suara dikirimkan dari aplikasi ke Raspberry Pi menggunakan metode *HTTP POST* karena memungkinkan pengiriman data secara langsung melalui jaringan lokal dengan syarat kedua perangkat harus saling terhubung ke dalam satu jaringan yang sama.

Pada Raspberry Pi berfungsi sebagai web server yang telah diprogram sebelumnya akan menerima permintaan *HTTP POST*. Data teks yang dikirimkan dari aplikasi kemudian diterima dan diproses oleh Raspberry Pi sebagai pengendali utama sistem. Hasil pemrosesan tersebut selanjutnya diteruskan ke Arduino, yang bertugas untuk mengeksekusi perintah dalam bentuk sinyal kendali ke TV.

Setelah seluruh proses pengiriman dan pemrosesan selesai dilakukan, sistem secara otomatis mengulang-ulang proses pendegaran pada proses *Speech Recognizer* agar aplikasi siap untuk mendengarkan dan memproses ucapan pengguna berikutnya. Oleh karena itu, sistem dapat bekerja secara terus-menerus tanpa memerlukan sentuhan langsung dari pengguna untuk mengaktifkan kembali proses pengenalan suara.

4.1.3 Implementasi *WU-MANBER*

Pada alat ini algoritma *WU-MANBER* merupakan tahap yang paling utama yaitu proses pencocokan kalimat perintah dengan pola yang sudah ditentukan, yang mana nantinya akan menghasilkan perintah sesuai dengan apa yang telah diperintahkan. Pada tahap awal sistem menerima hasil pengenalan suara berupa teks dan *Pattern* yang akan dicari.

Tabel 4. 1 Input Teks dan *Pattern*

Parameter	Nilai
Teks (T)	"tolong nyalakan tv"
<i>Pattern</i> (P)	"nyala"
Panjang <i>Pattern</i>	5 karakter

Tabel 4.1 menunjukkan parameter yang digunakan pada proses pencocokan algoritma Wu-Manber. Teks (T) berupa hasil konversi suara pengguna yaitu "tolong nyalakan tv", sedangkan pattern (P) yang dicocokkan adalah "nyala" dengan panjang 5 karakter. Pattern tersebut digunakan sebagai kata kunci untuk mendeteksi perintah menyalakan televisi. Selanjutnya, algoritma Wu-Manber akan melakukan proses pencocokan antara pattern dan teks untuk menentukan apakah perintah yang diberikan sesuai dengan salah satu pola yang telah disimpan pada sistem.

Selanjutnya yaitu tahap penyederhanaan kalimat. sebelum kalimat dicocokkan dengan *Pattern*, sistem terlebih dahulu menyederhanakan kalimat dari pengguna melalui dua proses, yaitu penghapusan kata tidak penting (*stopword removal*) dan pengubahan kata ke bentuk dasarnya (*Stemming*). Dengan demikian pemrosesan teks input sebelum masuk ke *WU-MANBER* dapat dilihat pada tabel berikut ini.

Tabel 4. 2 Penyederhanaan Kalimat

Tahap	Proses	Hasil
<i>Stopword Removal</i>	Kata "tolong" termasuk dalam daftar <i>stopword</i> tambahan (['tolong', 'sekarang', 'ya', 'dong', ...]) sehingga dihapus	"nyalakan tv"
<i>Stemming</i>	Kata "nyalakan" di-stem oleh Sastrawi menjadi bentuk dasarnya	"nyala tv"

Tabel 4.2 menunjukkan hasil tahap *preprocessing* sebelum proses pencocokan menggunakan algoritma *Wu-Manber*. Pada tahap *stopword removal*, kata "tolong" dihapus karena termasuk dalam daftar *stopword* tambahan sehingga menghasilkan teks "nyalakan tv". Selanjutnya, pada tahap *stemming*, kata "nyalakan" diubah ke bentuk dasarnya menjadi "nyala", sehingga teks akhir yang digunakan dalam proses pencocokan adalah "nyala tv".

Hasil akhir yang akan dicocokkan dengan *Pattern* adalah "nyala tv". Seluruh daftar perintah yang tersimpan dalam sistem juga telah melalui proses penyederhanaan yang sama sejak awal, sehingga perintah "nyalakan tv" sudah tersimpan dalam bentuk "nyala tv". Oleh karena itu, kalimat pengguna dan perintah yang tersimpan memiliki bentuk yang sama.

Kemudian tahap preprocessing yang bertujuan untuk membangun tabel *SHIFT* dan *HASH* yang digunakan untuk mempercepat proses pencarian. Pertama yaitu proses menentukan panjang minimum *pattern*.

Tabel 4. 3 Menentukan panjang minimum *Pattern*

<i>Pattern</i>	Panjang
nyala	5

Tabel 4.3 menunjukkan *pattern* yang digunakan pada proses pencocokan menggunakan algoritma *Wu-Manber*. *Pattern* yang digunakan adalah "nyala" dengan panjang 5 karakter. Informasi panjang *pattern* ini digunakan sebagai acuan dalam proses preprocessing untuk membentuk tabel yang diperlukan serta menentukan proses pencocokan antara *pattern* dengan teks masukan. Sehingga, $m=5$, dan Ukuran blok (B) adalah : $B=2$.

Kemudian yang kedua yaitu proses pembentukan tabel *SHIFT*, yang mana pada tahap ini *pattern* "nyala" dipecah menjadi blok 2 karakter.

Tabel 4. 4 Pembentukan tabel *SHIFT*

Posisi	Blok
1-2	ny
2-3	ya
3-4	al
4-5	la

Tabel 4.4 menunjukkan pembagian *pattern* menjadi beberapa blok berukuran 2 karakter ($block\ size = 2$) pada tahap preprocessing algoritma *Wu-Manber*. *Pattern* "nyala" dibagi menjadi empat blok, yaitu "ny", "ya", "al", dan "la", yang selanjutnya digunakan dalam pembentukan tabel *Shift* dan *Hash* untuk mempercepat proses pencocokan *pattern* dengan teks masukan. Yang mana nilai *SHIFT* dihitung menggunakan rumus: $SHIFT(x)=m-B-i$.

Keterangan:

1. m = panjang *Pattern* = 5
2. B = ukuran blok = 2
3. i = posisi blok

Tabel 4. 5 Pembentukan tabel nilai *SHIFT*

Blok	Nilai <i>SHIFT</i>
ny	3
ya	2
al	1
la	0

Keterangan:

1. Nilai *SHIFT* = 0 menunjukkan blok *Suffix* terakhir dari *Pattern*.
2. Nilai *SHIFT* > 0 menunjukkan teks dapat digeser tanpa perlu pemeriksaan lebih lanjut.

Tabel 4.5 menunjukkan nilai *Shift* yang dihasilkan untuk setiap blok pada *pattern* "nyala". Nilai *Shift* diperoleh berdasarkan posisi masing-masing blok terhadap blok terakhir pada *pattern*, sehingga blok "la" sebagai blok terakhir memiliki nilai 0, sedangkan blok "al", "ya", dan "ny" masing-masing memiliki nilai 1, 2, dan 3. Nilai ini digunakan oleh algoritma *Wu-Manber* untuk menentukan jumlah pergeseran saat proses pencocokan berlangsung.

Kemudian yang ketiga yaitu pembentukan tabel *HASH* yang berfungsi untuk menyimpan setiap blok pada *Pattern*.

Tabel 4. 6 Pembentukan tabel *HASH*

<i>HASH</i>	Blok
H(ny)	ny
H(ya)	ya
H(al)	al
H(la)	la

Tabel 4.6 menunjukkan tabel *Hash* yang dibentuk pada tahap *preprocessing* algoritma *Wu-Manber*. Setiap blok berukuran 2 karakter dari *pattern* "nyala", yaitu "ny", "ya", "al", dan "la", dipetakan ke dalam nilai *Hash* sesuai bloknnya. Tabel *Hash* ini digunakan untuk mempercepat proses pencarian dengan mengidentifikasi blok yang akan dibandingkan dengan teks masukan selama proses pencocokan. tabel

HASH digunakan untuk mengetahui apakah blok karakter pada teks memiliki kemungkinan cocok dengan *Pattern*.

Setelah tahap *preprocessing* pada proses diatas selesai, selanjutnya algoritma akan masuk pada tahap *scanning*. Pada tahap ini algoritma mulai melakukan pencarian pada teks. Pada kasus ini, teks yang akan diproses pada saat proses pencarian yaitu “tolong nyalakan tv”.

Tabel 4. 7 Indeks Karakter

Posisi	Karakter
1	t
2	o
3	l
4	o
5	n
6	g
7	Spasi
8	n
9	y
10	a
11	l
12	a
13	k
14	a
15	n
16	spasi
17	t
18	v

Tabel 4.7 menunjukkan *indeks* setiap karakter pada teks masukan "tolong nyalakan tv". Pemberian *indeks* dilakukan untuk memudahkan proses pencocokan pada algoritma *Wu-Manber*, karena setiap karakter memiliki posisi yang akan digunakan sebagai acuan dalam pembentukan blok dan pergeseran (*shift*) selama proses pencarian pattern di dalam teks.

Proses selanjutnya yaitu mengambil blok pertama pada teks, karena $B = 2$, maka algoritma mengambil 2 karakter dari posisi akhir jendela pencarian. Misalnya

diperoleh blok: ng. Nilai *HASH* dihitung: $h = \text{HASH}(ng)$. Karena blok "ng" tidak ada pada tabel *HASH* maka: $\text{SHIFT}(h) > 0$. Maka teks langsung digeser sejumlah nilai *SHIFT* tanpa memeriksa seluruh *Pattern*.

Setelah melakukan pencarian dengan cara mengambil blok pertama pada teks, dan tidak menemukan kecocokan maka algoritma akan melakukan pergeseran hingga menemukan *Suffix Pattern*. Setelah beberapa kali pergeseran diperoleh blok: la. Blok "la" ditemukan pada tabel *HASH* dengan: $\text{SHIFT}(la) = 0$. Karena nilai $\text{SHIFT} = 0$ maka algoritma memasuki tahap verifikasi.

Tahap selanjutnya yaitu pemeriksaan *Prefix*, setelah ditemukan *Suffix* yang sesuai, sistem memeriksa *Prefix Pattern*. *Pattern*: "nyala", *Prefix*: "nya". *Prefix* pada Teks, Teks: nyalakan. *Prefix* yang diperoleh: nya. Oleh karena itu karena *Prefix Pattern* "nya", dan *Prefix* Teks "nya" menghasilkan hasil yang cocok maka algoritma akan melanjutkan ke tahap verifikasi penuh.

Tabel 4. 8 Verifikasi pola

Posisi	<i>Pattern</i>	Teks	Hasil
1	n	n	Cocok
2	y	y	Cocok
3	a	a	Cocok
4	l	l	Cocok
5	a	a	Cocok

Tabel 4.8 menunjukkan proses verifikasi pola setelah algoritma *Wu-Manber* menemukan kandidat kecocokan pada teks. Setiap karakter pada *pattern* "nyala" dibandingkan dengan karakter pada teks pada posisi yang sesuai, dan seluruh karakter menghasilkan status cocok, sehingga *pattern* dinyatakan berhasil ditemukan di dalam teks masukan. Seluruh karakter cocok sehingga *Pattern*

ditemukan. Oleh karena itu algoritma berhasil menemukan *Pattern* "nyala" pada teks "tolong nyalakan tv" pada posisi karakter ke-8.

4.1.4 Perangkat Keras

Perangkat keras dalam penelitian ini merupakan komponen fisik yang berfungsi untuk mengirimkan sinyal berdasarkan perintah yang telah dikirim dari perangkat lunak. Komponen utama yang digunakan dalam penelitian ini yaitu Raspberry Pi yang berperan sebagai mikrokontroler pusat yang kemudian dihubungkan ke arduino. Raspberry Pi menerima perintah dari server lokal, Raspberry Pi mulai mengeksekusi perintah dan diteruskan ke arduino melalui kabel *USB* dan selanjutnya arduino akan mengeksekusi perintah melalui *PIN (Personal Identification Number)* untuk mengirimkan sinyal ke TV menggunakan IR Transmitter.

IR Transmitter ini dihubungkan ke *PIN* pada arduino untuk mengirimkan sinyal atau kode *HEX* yang sebelumnya sudah pernah direkam menggunakan IR Receiver. Selain itu pada sistem ini juga dilengkapi Buzzer yang berfungsi untuk memberi tanda atau indikator bahwa ada perintah yang sesuai dengan *Pattern* yang ada pada program yang kemudian diteruskan oleh IR Transmitter.

Sumber daya pada sistem ini berasal dari listrik yang kemudian dihubungkan ke Raspberry Pi menggunakan adaptor 5V 3A yang mengubah dari tegangan AC ke DC, karena pada *Raspberri Pi* Hanya menerima tegangan input sebesar 5V agar dapat bekerja dengan stabil.



Gambar 4. 2 Rangkaian Hardware

Berdasarkan gambar 4.2, terdapat beberapa komponen elektronika antara lain yang pertama ada Raspberry Pi yang bertugas sebagai pengontrol sistem yang selanjutnya diteruskan ke arduino, yang mana pada komponen tersebut semua proses perintah yang masuk akan di proses dan dilanjutkan ke arduino. Pada arduino perintah yang masuk akan diproses dan akan dikirimkan sinyalnya melalui IR Transmitter untuk mengontrol TV, seperti menyalakan, mematikan, menambah volume, mengurangi volume, dan mengontrol TV. IR Transmitter mengirim sinyal inframerah ke TV untuk menjalankan perintah berdasarkan kode *hexadecimal* yang telah direkan sebelumnya menggunakan IR Receiver. IR Receiver ini berperan sebagai penangkap kode *HEX* dari remote asli TV. Sistem ini di buat agar pengguna dapat mengontrol TV secara mudah menggunakan perintah suara tanpa harus menombol tombol pada remot sehingga dapat memudahkan penggunaan dalam mengoperasikan TV.

4.2 Hasil Pengujian

4.2.1 Hasil Pengujian Perintah Suara

Tabel 4. 9 Hasil pengujian perintah suara

No	Perintah Suara	Hasil <i>Speech recognition</i>	Jumlah kesalahan
1	Tolong nyalakan TV	tolong nyalakan tv	0
2	Matikan TV	matikan tv	0
3	Saya ingin menaikkan volume	saya ingin menaikkan volume	0
4	Suara sangat kencang tolong turunkan volume	suara sangat kencang tolong turunkan volume	0
5	Saya ingin membuka menu	saya ingin membuka menu	0
Total			0

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

$$\mu = \frac{0}{5} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.9 menunjukkan hasil pengujian *speech recognition* terhadap lima perintah suara yang diberikan pengguna. Seluruh perintah berhasil dikonversi menjadi teks sesuai dengan ucapan yang diberikan tanpa mengalami kesalahan. Hasil pengujian menunjukkan bahwa tingkat kesalahan yang diperoleh adalah 0%, sehingga proses pengenalan suara pada sistem dapat bekerja dengan baik dan menghasilkan teks yang siap diproses pada tahap pencocokan menggunakan algoritma Wu-Manber. Berikut blok diagram pengujian perintah suara dapat dilihat pada gambar 4.3.



Gambar 4. 3 Blok Diagram Pengujian Perintah Suara

Blok diagram pada Gambar 4.3 menunjukkan alur pengujian speech recognition pada sistem. Proses diawali ketika pengguna mengucapkan perintah yang kemudian diterima oleh mikrofon. Selanjutnya, suara diproses menggunakan *Google Speech API* untuk dikonversi menjadi teks. Hasil konversi tersebut ditampilkan pada aplikasi dan dibandingkan dengan perintah asli yang diucapkan pengguna. Setelah itu, dilakukan perhitungan jumlah kesalahan antara hasil pengenalan suara dengan perintah sebenarnya untuk mengetahui tingkat akurasi sistem. Hasil perhitungan tersebut kemudian digunakan sebagai dasar dalam mengevaluasi kinerja proses pengenalan suara yang diterapkan pada sistem kontrol TV berbasis perintah suara.

4.2.2 Hasil Pengujian *Stopword Removal*

Tabel 4. 10 Hasil pengujian *Stopword Removal*

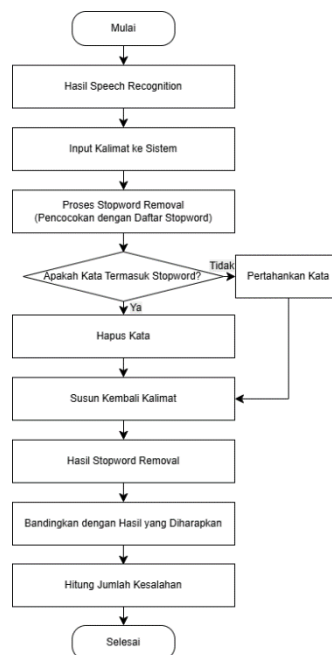
No	Hasil <i>Speech recognition</i>	Hasil <i>Stopword Removal</i>	Jumlah kesalahan
1	tolong nyalakan tv	nyalakan tv	0
2	matikan tv	matikan tv	0
3	saya ingin menaikkan volume	menaikkan volume	0
4	suara sangat kencang tolong turunkan volume	suara sangat kencang turunkan volume	0
5	saya ingin membuka menu	membuka menu	0
Total			0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{5} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.10 menunjukkan hasil pengujian pada tahap *stopword removal*. Pada tahap ini, kata-kata yang tidak memengaruhi makna utama perintah, seperti "tolong", "saya", dan "ingin", dihapus dari hasil *speech recognition* sehingga menghasilkan kalimat yang lebih sederhana dan berisi kata-kata penting saja. Seluruh data berhasil diproses sesuai dengan daftar *stopword* yang digunakan, sehingga tingkat kesalahan yang diperoleh adalah 0%. Berikut blok diagram pengujian *stopword removal* dapat dilihat pada gambar 4.4.



Gambar 4. 4 Blok Diagram Pengujian *Stopword Removal*

Blok diagram pada Gambar 4.4 menunjukkan alur pengujian *stopword removal* pada sistem. Proses diawali dengan hasil *speech recognition* yang berupa teks hasil konversi suara pengguna. Teks tersebut kemudian menjadi masukan pada proses *stopword removal*, yaitu pencocokan setiap kata dengan daftar *stopword* yang telah ditentukan. Apabila suatu kata termasuk *stopword*, maka kata tersebut akan dihapus, sedangkan kata yang bukan *stopword* akan dipertahankan. Setelah seluruh kata diproses, sistem menyusun kembali kalimat sehingga diperoleh hasil *stopword removal*. Hasil tersebut kemudian dibandingkan dengan keluaran yang diharapkan, selanjutnya dihitung jumlah kesalahan yang terjadi untuk mengevaluasi kinerja proses *stopword removal* pada sistem.

4.2.3 Hasil Pengujian *Stemming*

Tabel 4. 11 Hasil pengujian *Stemming*

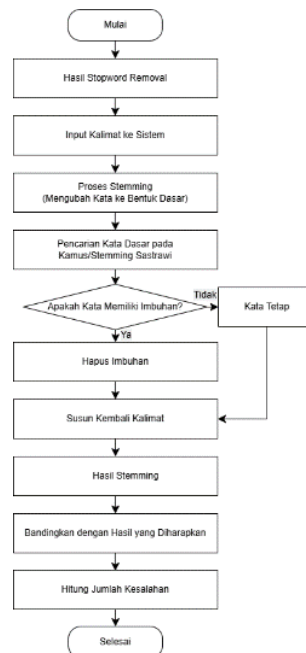
No	Hasil <i>Stopword Removal</i>	Hasil <i>Speech Stemming</i>	Jumlah kesalahan
1	nyalakan tv	nyala tv	0
2	matikan tv	mati tv	0
3	volume naik	naik volume	0
4	suara sangat kencang turunkan volume	suara sangat kencang turun volume	0
5	membuka menu	buka menu	0
Total			0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{5} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.11 menunjukkan hasil pengujian pada tahap *stemming*, yaitu proses mengubah setiap kata menjadi bentuk dasarnya. Kata-kata seperti "nyalakan" menjadi "nyala", "matikan" menjadi "mati", "turunkan" menjadi "turun", dan "membuka" menjadi "buka", sedangkan susunan kata pada perintah ketiga disesuaikan menjadi "naik volume" agar sesuai dengan pola yang telah ditentukan pada sistem. Seluruh data berhasil diproses dengan benar sehingga tingkat kesalahan yang diperoleh adalah 0%. Berikut blok diagram pengujian *stemming* dapat dilihat pada gambar 4.4.



Gambar 4. 5 Blok Diagram Pengujian *Stemming*

Blok diagram pada Gambar 4.5 menunjukkan alur pengujian *stemming* pada sistem. Proses diawali dengan hasil *stopword removal* yang menjadi masukan untuk proses *stemming*. Selanjutnya, setiap kata diperiksa untuk mencari bentuk kata dasarnya menggunakan kamus *stemming* Sastrawi. Apabila kata mengandung imbuhan, maka imbuhan tersebut akan dihapus sehingga diperoleh kata dasar, sedangkan kata yang tidak memiliki imbuhan akan dipertahankan. Setelah seluruh kata diproses, sistem menyusun kembali kalimat dan menghasilkan keluaran akhir dari proses *stemming*. Hasil tersebut kemudian dibandingkan dengan hasil yang diharapkan, selanjutnya dihitung jumlah kesalahan yang terjadi untuk mengevaluasi tingkat keberhasilan proses *stemming* pada sistem.

4.2.4 Hasil Pengujian WU-MANBER

Pengujian algoritma *Wu-Manber* dilakukan menggunakan teks hasil *speech recognition* yaitu "tolong nyalakan tv" dengan *pattern* yang dicari adalah "nyala".

Pengujian ini bertujuan untuk mengetahui proses pembentukan tabel, pencarian, hingga verifikasi *pattern* pada sistem.

Tabel 4. 12 Pembentuka tabel *SHIFT*

Blok	<i>SHIFT</i>	Jumlah Kesalahan
ny	3	0
ya	2	0
al	1	0
la	0	0
Total		0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{4} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.12 menunjukkan hasil pembentukan tabel *Shift* pada *pattern* "nyala". Setiap blok memperoleh nilai *shift* berdasarkan posisinya terhadap blok terakhir. Seluruh proses pembentukan tabel berjalan dengan baik tanpa ditemukan kesalahan, sehingga tingkat kesalahan yang diperoleh adalah 0%.

Tabel 4. 13 Hasil pembentukan tabel *HASH*

<i>HASH</i>	Blok	Jumlah Kesalahan
H(ny)	ny	0
H(ya)	ya	0
H(al)	al	0
H(la)	la	0
Total		0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{4} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.13 menunjukkan hasil pembentukan tabel *Hash* dari setiap blok *pattern* "nyala". Seluruh blok berhasil dipetakan ke dalam tabel *Hash* sesuai proses *preprocessing* algoritma *Wu-Manber* tanpa terjadi kesalahan, sehingga nilai tingkat kesalahan sebesar 0%.

Tabel 4. 14 Proses Pencarian *WU-MANBER*

Iterasi	Blok Teks	Nilai <i>SHIFT</i>	Keputusan	Jumlah Kesalahan
1	ng	> 0	Geser teks	0
2	g	> 0	Geser teks	0
3	ny	3	Geser teks	0
4	ya	2	Geser teks	0
5	al	1	Geser teks	0
6	la	0	Verifikasi	0
Total				0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{6} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.14 menunjukkan proses pencarian *pattern* menggunakan algoritma *Wu-Manber*. Algoritma melakukan pergeseran berdasarkan nilai *Shift* hingga menemukan blok terakhir "la" dengan nilai *Shift* = 0. Kondisi tersebut menandakan kandidat *pattern* telah ditemukan sehingga proses dilanjutkan ke tahap verifikasi. Selama proses pencarian tidak ditemukan kesalahan sehingga tingkat kesalahan sebesar 0%.

Tabel 4. 15 Verifikasi *Pattern*

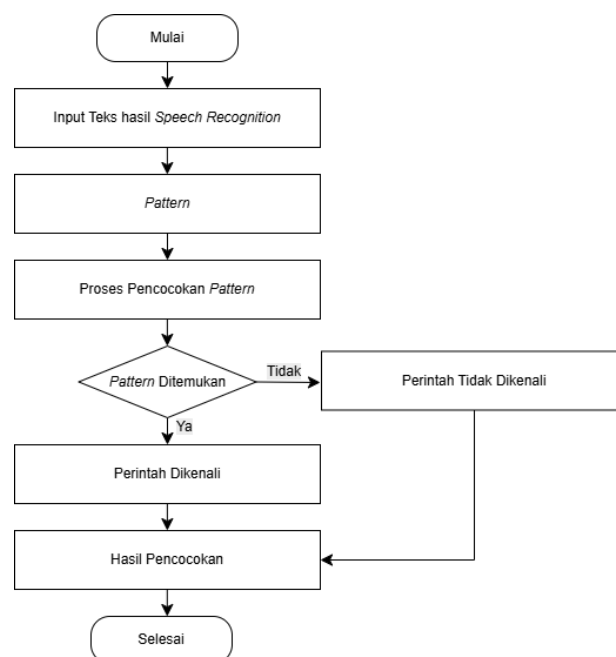
Posisi	<i>Pattern</i>	Teks	Hasil	Jumlah Kesalahan
1	n	n	Cocok	0
2	y	y	Cocok	0
3	a	a	Cocok	0
4	l	l	Cocok	0
5	a	a	Cocok	0
Total				0

$$= \frac{\sum E_i}{n} \times 100\%$$

$$\mu = \frac{0}{5} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.15 menunjukkan hasil verifikasi setelah kandidat *pattern* ditemukan. Seluruh karakter pada *pattern* "nyala" sesuai dengan karakter pada teks sehingga diperoleh lima kecocokan. Hasil tersebut menunjukkan bahwa *pattern* berhasil dikenali oleh algoritma *Wu-Manber* dan sistem dapat melanjutkan proses dengan mengirimkan kode HEX untuk menyalakan televisi. Tingkat kesalahan pada tahap verifikasi adalah 0%. Berikut blok diagram pengujian algoritma *Wu-Manber* dapat dilihat pada gambar 4.6.



Gambar 4. 6 Blok Diagram Pengujian *Wu-Manber*

Blok diagram pada Gambar 4.6 menunjukkan alur pengujian algoritma *Wu-Manber* pada sistem. Proses diawali dengan teks hasil *speech recognition* yang menjadi masukan untuk proses pencocokan. Selanjutnya, sistem membandingkan

teks tersebut dengan kumpulan pattern atau pola perintah yang telah disimpan menggunakan algoritma *Wu-Manber*. Apabila ditemukan *pattern* yang sesuai, sistem akan mengenali perintah dan menghasilkan keluaran berupa nama perintah yang akan digunakan pada tahap berikutnya. Sebaliknya, jika tidak ditemukan *pattern* yang sesuai, sistem akan memberikan hasil bahwa perintah tidak dikenali. Hasil pencocokan tersebut kemudian digunakan untuk mengevaluasi kemampuan algoritma *Wu-Manber* dalam mengenali perintah suara yang diberikan oleh pengguna.

4.2.5 Hasil Pengujian Output Kode HEX

Tabel 4. 16 Hasil Pengujian Output Kode Hex

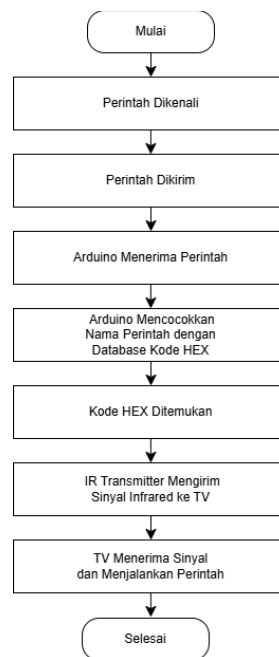
No	<i>Pattern</i>	Fungsi TV	Kode HEX	Jumlah kesalahan
1	nyala tv	Menyalakan TV	0xF708FB04	0
2	mati tv	Mematikan TV	0xF708FB04	0
3	volume naik	Volume +	0xFD02FB04	0
4	volume turun	Volume -	0xFC03FB04	0
5	mute	Mute	0xF609FB04	0
6	menu	Menu	0x837CFB04	0
7	atas	Navigasi Atas	0xBF40FB04	0
8	bawah	Navigasi Bawah	0xBE41FB04	0
9	kanan	Navigasi Kanan	0xF906FB04	0
10	kiri	Navigasi Kiri	0xF807FB04	0
11	pilih	OK / Enter	0xBB44FB04	0
12	pengaturan	Pengaturan	0xBC43FB04	0
13	keluar	Keluar	0xA45BFB04	0
14	program	Program	0xAC53FB04	0
15	masukan	Input	0xF40BFB04	0
Total				0

$$= \frac{\sum Ei}{n} \times 100\%$$

$$\mu = \frac{0}{15} \times 100\%$$

$$\mu = 0\%$$

Tabel 4.16 menunjukkan hasil pengujian keluaran kode *HEX* yang dikirimkan oleh sistem setelah proses pencocokan menggunakan algoritma *Wu-Manber* berhasil dilakukan. Setiap *pattern* yang dikenali, seperti "nyala tv", "mati tv", "volume naik", dan perintah lainnya, menghasilkan kode *HEX* yang sesuai dengan fungsi tombol pada remote TV. Seluruh pengujian berhasil menghasilkan kode *HEX* yang benar tanpa ditemukan kesalahan, sehingga tingkat kesalahan yang diperoleh adalah 0%. Hasil ini menunjukkan bahwa sistem mampu menerjemahkan perintah suara menjadi sinyal inframerah yang tepat untuk mengendalikan televisi. Berikut blok diagram pengujian output kode *HEX* dapat dilihat pada gambar 4.7.



Gambar 4. 7 Blok Diagram Pengujian Kode *HEX*

Blok diagram pada Gambar 4.7 menjelaskan proses pengiriman perintah dari hasil identifikasi algoritma *Wu-Manber* hingga televisi menjalankan perintah yang diberikan. Setelah *Wu-Manber* berhasil mengenali pola perintah suara pengguna, Raspberry Pi mengirimkan nama perintah tersebut ke Arduino melalui komunikasi serial. Selanjutnya, Arduino mencocokkan nama perintah dengan

daftar kode HEX yang telah tersimpan pada program. Apabila kode HEX yang sesuai ditemukan, Arduino mengirimkan sinyal inframerah melalui IR Transmitter menuju televisi. Sensor inframerah pada TV kemudian menerima sinyal tersebut dan mengeksekusi fungsi sesuai kode yang dikirim, seperti menyalakan atau mematikan TV, mengatur volume, membuka menu, maupun menjalankan fungsi lainnya. Setelah perintah berhasil dijalankan oleh televisi, proses dinyatakan selesai dan sistem kembali siap menerima perintah suara berikutnya.

4.2.6 Hasil Pengujian sistem

Proses pengujian sistem ini dilakukan untuk mengetahui proses kerja sistem, apakah sistem bekerja sesuai dengan perintah dan berfungsi sesuai dengan program yang telah diatur sebelumnya. Selain itu, pengujian juga dilakukan untuk mengetahui potensi *error* pada sistem. Pengujian pengenalan kalimat perintah ini dilakukan sebanyak 25 kali percobaan dengan beberapa varian jenis perintah. Seluruh pengujian ini dilakukan didalam ruangan dengan memberikan berbagai jenis perintah.

Tabel 4. 17 Hasil uji coba

No	Perintah	<i>Pattern</i>	Hasil Pengujian	Keterangan	Jumlah Kesalahan
1.	Nyalakan TV	Nyala tv	Benar	Nyala	0
2.	Matikan TV	Mati tv	Benar	Mati	0
3.	Tambah Volume	Tambah volume	Benar	Tambah	0
4.	Kurangi Volume	Kurangi volume	Benar	Kurang	0
5.	Menu	Menu	Benar	Menu	0
6.	Atas	Atas	Benar	Atas	0
7.	Bawah	Bawah	Benar	Bawah	0
8.	Kanan	Kanan	Benar	Kanan	0
9.	Kiri	Kiri	Benar	Kiri	0
10.	Pilih	pilih	Benar	Pilih	0
11.	Oke	oke	Benar	Oke	0
12.	Pengaturan	pengaturan	Benar	pengaturan	0

13.	Tolong nyalakan TV	Nyala tv	Benar	Nyala	0
14.	Nyalakan TV sekarang	Nyala tv	Benar	Nyala	0
15.	Saya ingin menyalakan TV	Nyala tv	Benar	Nyala	0
16.	Tolong tambah volume sekarang	Tambah volume	Benar	Tambah	0
17.	Saya ingin menambah volume sekarang	Tambah volume	Benar	Tambah	0
18.	Tolong kurangi volume sekarang	Kurangi volume	Benar	Kurang	0
19.	Saya ingin mengurangi volume	Kurangi volume	Benar	Kurang	0
20.	Ayo matikan TV	matikan tv	Benar	Mati	0
21.	saya pulang dari pasar, kemudian ingin menyalakan TV	Nyalakan tv	Benar	Nyala	0
22.	Saya ingin melihat menu	menu	Benar	Menu	0
23.	Geser ke kanan	kanan	salah	Tidak ada respon	1
24.	Geser ke kiri	kanan	salah	Tidak ada respon	1
25.	keluar	keluar	Benar	Keluar	0
Jumlah Kesalahan					2 (8%)

Dari hasil pengujian pada tabel 4.17 telah dilakukan pengujian sistem sebanyak 25 kali. Pengujian ini memerintahkan berbagai jenis perintah yang mungkin orang ucapkan ketika ingin mengontrol TV. Hasil pengujian menunjukkan bahwa sistem berhasil mengenali 23 perintah dengan baik dan benar dan dapat menjalankan perintah TV sesuai dengan perintah, dan menghasilkan 2 kesalahan pengujian. Kesalahan tersebut terjadi pada proses pencocokan untuk kalimat perintah “geser ke kanan” dan “geser ke kiri”. Adapun tingkat error sistem dari hasil pengujian dapat dihitung menggunakan persamaan berikut ini.

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

Pada 25 kali percobaan sistem berhasil mengenali perintah sebanyak 23 perintah dan terdapat kesalahan sebanyak 2 kali. Adapun pengujian *error* sistem dengan 2 kali kesalahan dapat dihitung sebagai berikut:

$$\begin{aligned}\mu &= \frac{\sum Ei}{n} \times 100\% \\ \mu &= \frac{2}{25} \times 100\% \\ \mu &= 8\%\end{aligned}$$

Dapat diketahui bahwa dari 25 kali percobaan menggunakan berbagai jenis perintah sistem ini memiliki *error* sebesar 8%.

Pengujian sistem pengenalan kalimat perintah menggunakan metode *Wu-Mamber* dilakukan untuk menguji tingkat keberhasilan sistem dalam mengenali perintah-perintah yang diucapkan oleh pengguna. Berdasarkan pengujian di atas, dapat dilihat bahwa sistem memiliki tingkat keberhasilan yang tinggi dalam mengenali perintah suara. Hal tersebut menunjukkan bahwa perintah kalimat atau kata yang digunakan telah sesuai dengan pola atau *Pattern* yang terdaftar dalam algoritma *WU-MANBER*, sehingga program dapat berjalan sesuai dengan yang diharapkan.

Sementara itu, terdapat 2 kali kesalahan dalam pengujian sistem, pada perintah geser ke kanan dan geser ke kiri. Kesalahan tersebut disebabkan karena tidak ada *Pattern* di dalam program, sehingga pada saat proses pencocokan perintah dengan *Pattern*, perintah tidak dikenali. Oleh karena itu sistem tidak dapat mengenali perintah tersebut dan tidak memberikan respons yang diharapkan pengguna.

Dengan demikian *error rate*-nya sebesar 8%. Hasil ini menunjukkan bahwa sistem sudah dapat bekerja dengan baik dan benar dalam mengenali perintah suara untuk mengontrol TV, meskipun masih banyak yang harus disempurnakan dalam pengembangan kalimat-kalimat perintah yang lebih banyak variasi lagi. Sehingga sistem dapat bekerja lebih maksimal lagi terhadap berbagai bentuk perintah pengguna.

4.5 Integrasi Islam

Penelitian ini dibuat untuk memudahkan pengoperasian TV bagi pengguna. Oleh karena itu, dalam penerapannya sistem ini tidak lepas dari norma-norma islam dan dapat mencerminkan nilai-nilai spiritual dalam agama islam. Prinsip-prinsip ini diharapkan dapat menjadi landasan atau dasar dalam pengembangan teknologi yang berhubungan dengan kemaslahatan umat manusia.

4.5.1 *Muamalah Ma'a Allah*

Muamalah Ma'a Allah merupakan bentuk hubungan antara seorang hamba dengan Allah yang mana mensyukuri atas akal dan ilmu merupakan sebagian bentuk ibadah. Dalam penelitian ini, membantu memudahkan pekerjaan atas apa yang orang lain kesulitan dalam melakukan hal tersebut itu merupakan bentuk rasa wujud syukur atas diberikanya akal, dan ilmu. Allah berfirman dalam QS. *An-Nahl* ayat 78:

وَاللَّهُ أَخْرَجَكُمْ مِنْ بُطُونِ أُمَّهَاتِكُمْ لَا تَعْلَمُونَ شَيْئًا وَجَعَلَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ وَالْأَفْئِدَةَ ۚ لَعَلَّكُمْ تَشْكُرُونَ

“Allah mengeluarkan kamu dari perut ibumu dalam keadaan tidak mengetahui sesuatu pun dan Dia menjadikan bagi kamu pendengaran, penglihatan, dan hati nurani (akal) agar kamu bersyukur.” (QS. *An-Nahl*: 78).

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, QS. *An-Nahl* ayat 78 menjelaskan bahwa manusia dilahirkan tanpa memiliki pengetahuan yang siap pakai. Allah kemudian menganugerahkan pendengaran, penglihatan, dan hati sebagai sarana memperoleh ilmu dan memahami realitas. Penyebutan pendengaran lebih dahulu menunjukkan pentingnya proses menerima informasi sejak awal kehidupan, sedangkan hati berfungsi mengolah, menimbang, dan memberi makna terhadap informasi tersebut. Karena itu, tujuan akhir dari anugerah indera dan akal adalah agar manusia bersyukur dengan menggunakan seluruh potensi itu sesuai petunjuk Allah.

Hubungan surah *An-Nahl* ayat 78 dengan penelitian ini terletak pada mensyukuri atas akal dan ilmu yang telah Allah berikan, bukan hanya membantu memudahkan meraka dalam hal pengoperasian TV tetapi juga menjadi perantara untuk mendapatkan rahmat dari Allah dengan cara mensyukuri atas akal dan ilmu yang telah Allah berikan.

4.5.2 *Muamalah Ma'a An-Nas*

Muamalah Ma'a An-Nas merupakan bentuk hubungan antara seseorang dengan kehidupan sosial. Adapun dalam penelitian ini, alat dibuat untuk meberikan kemudahan bagi pengguna dalam pengoperasian TV. Teknologi ini digunakan untuk memudahkan bagi pengguna dalam mengontrol TV. Hal ini sesuai dengan ajaran islam yaitu saling tolong-menolong dalam hal kebaikan. Allah berfirman dalam QS. *Al-Maidah* ayat 2:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَحْلُوا شَعَائِرَ اللَّهِ وَلَا الشُّهُرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَائِدَ وَلَا آمِينَ الْبَيْتِ الْحَرَامَ يَبْتَغُونَ فَضْلًا مِّن رَّبِّهِمْ وَرِضْوَانًا ۚ وَإِذَا حَلَلْتُمْ فَاصْطَادُوا ۚ وَلَا يَجْرِمَنَّكُمْ شَنَاٰنُ قَوْمٍ أَن صَدُّوكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ أَن تَعْتَدُوا ۚ وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ ۚ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ عَاثُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“Wahai orang-orang yang beriman, janganlah kamu melanggar syiar-syiar kesucian Allah, jangan melanggar kehormatan bulan-bulan haram, jangan mengganggu hadyu (hewan-hewan kurban)¹⁹⁵ dan qalā'id (hewan-hewan kurban yang diberi tanda), dan jangan (pula mengganggu) para pengunjung Baitulharam sedangkan mereka mencari karunia dan rida Tuhannya! Apabila kamu telah bertahalul (menyelesaikan ihram), berburulah (jika mau). Janganlah sekali-kali kebencian(-mu) kepada suatu kaum, karena mereka menghalang-halangi dari Masjidilharam, mendorongmu berbuat melampaui batas (kepada mereka). Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya.” (QS. Al-Maidah: 2).

Hubungan antara surat *Al-Maidah* ayat 2 dengan penelitian ini terletak pada hal tolong-menolong dalam hal kebaikan. Pada penelitian, alat dibuat untuk membantu memudahkan pengguna dalam hal pengoperasian TV tanpa harus menggunakan remot fisik menggunakan teknologi berbasis suara.

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, ayat ini menegaskan bahwa umat Islam diperintahkan untuk bekerja sama atau saling tolong menolong dalam segala bentuk kebaikan yang membawa manfaat bagi individu maupun masyarakat. Kerja sama (*ta'awun*) tidak dibatasi hanya pada sesama Muslim, tetapi mencakup semua pihak selama bertujuan mewujudkan kebajikan dan ketakwaan.

Sebagaimana telah di pesankan oleh Rasulullah bahwa kita sebagai manusia hendaklah mampu memberikan kemanfaatan serta kebaikan kepada orang lain, sesuai dengan tuntunan beliau yang tertuang dalam hadis riwayat Imam Ahmad:

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia (atau orang lain)." (HR. Imam Ahmad).

Menurut Imam Al-Munawi dalam kitab *Faidh al-Qadir* menjelaskan bahwa seseorang yang memberikan manfaat kepada manusia berarti telah menjalankan

salah satu tujuan utama syariat Islam, yaitu mewujudkan kemaslahatan (*jalb al-mashalih*) dan menghilangkan kemudharatan (*dar' al-mafasid*). Semakin besar manfaat yang diberikan kepada banyak orang, semakin besar pula nilai kebbaikannya di sisi Allah.

4.5.3 *Muamalah Ma'a Alam*

Muamalah Ma'a Alam merupakan hubungan seseorang dengan alam semesta sebagai sesama ciptaan Allah. Dalam hal ini alat yang dirancang untuk memudahkan pengoperasian TV, tidak hanya untuk membantu kemudahan bagi sesama manusia tetapi juga pemanfaatan apa yang telah Allah berikan dari alam. Hal ini sesuai dengan ajaran islam yaitu pemanfaatan alam secara seimbang dan tidak berlebihan. Allah berfirman dalam QS. *Al-Jasyiah* ayat 13:

وَسَخَّرَ لَكُم مَّا فِي السَّمُوتِ وَمَا فِي الْأَرْضِ جَمِيعًا مِّنْهُ إِنَّ فِي ذَلِكَ لَآيَاتٍ لِّقَوْمٍ يَتَفَكَّرُونَ

Dia telah menundukkan (pula) untukmu apa yang ada di langit dan apa yang ada di bumi semuanya (sebagai rahmat) dari-Nya. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kebesaran Allah) bagi kaum yang berpikir. (QS. Al-Jasyiah: 13).

Menurut M. Quraish Shihab dalam *Tafsir Al-Misbah*, menjelaskan bahwa Allah telah menundukkan seluruh alam semesta untuk kepentingan manusia, baik yang berada di langit maupun di bumi. Yang dimaksud dengan yang ada di langit antara lain matahari, bulan, bintang, hujan, dan berbagai fenomena alam lainnya. Sedangkan yang ada di bumi meliputi tumbuhan, hewan, air, tanah, dan berbagai sumber daya alam yang dapat dimanfaatkan manusia. Penundukan ini bukan berarti manusia bebas mengeksploitasi alam tanpa batas, melainkan sebagai bentuk kasih sayang dan karunia Allah agar manusia dapat memenuhi kebutuhan hidupnya serta

menjalankan tugas sebagai khalifah di bumi. Semua nikmat tersebut berasal dari Allah semata dan menjadi bukti kekuasaan-Nya bagi orang-orang yang mau berpikir dan merenungkan ciptaan-Nya.

Hubungan antara surat *Al-Jasyah* ayat 13 dengan penelitian ini yaitu manusia diberikan kekuasaan untuk memanfaatkan segala ciptaanya termasuk gelombang suara dan listrik yang telah diproses sedemikian rupa menjadi sebuah alat kontrol TV berbasis suara.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini telah berhasil mengimplementasikan metode *WU-MANBER* untuk pencocokan teks pada sistem kontrol TV bagi pengguna, dengan tujuan yaitu memberikan kemudahan dalam pengoperasian TV bagi pengguna melalui perintah suara. Implementasi dalam sistem ini menerapkan komponen perangkat lunak seperti aplikasi *android* yang dibangun menggunakan *software MIT App Inventor*, *Google Speech recognition* untuk mengubah suara ke teks, serta proses *preprocessing* teks (*steeming* dengan sastrawi dan *stopword removal*) dan penerapan algoritma *WU-MANBER* yang semuanya berjalan pada Raspberry Pi 3 sebagai otak utama yang kemudian diteruskan ke arduino. Sementara itu, komponen perangkat keras seperti IR Transmitter berfungsi sebagai nirkabel yang akan mengirimkan sinyal menggunakan cahaya inframerah dari perintah yang telah masuk, kemudian ada Buzzer yang berfungsi sebagai indikator bahwa ada perintah masuk dan sesuai dengan *Pattern* yang kemudian di teruskan oleh IR Tansmitter. Hasil pengujian sistem menunjukkan bahwa sistem ini memiliki *error rate* pengenalan perintah suara sebesar 8% dalam kondisi terbatas. Oleh karena itu, sistem membuktikan bahwa tingkat keberhasilan algoritma *WU-MANBER* dalam memproses perintah suara sangat efektif.

5.2. saran

Penelitian pada sistem kontrol TV ini masih terdapat banyak kekurangan yang perlu dikembangkan lagi oleh peneliti selanjutnya, antara lain:

1. Pengembangan terkait jumlah teks perintah yang lebih banyak atau lebih bervariasi lagi yang disesuaikan dengan tombol pada remot TV, atau dapat mengenali teks perintah yang lebih kompleks, seperti “naikkan volume menjadi 15”, “matikan dalam 5 menit”.
2. Mengembangkan sistem agar dapat bekerja secara *offline* tidak tergantung pada layanan *Google Speech recognition* sehingga alat tetap dapat digunakan dalam keadaan tanpa internet.
3. Penambahan fitur perintah suara menggunakan berbagai bahasa di dunia untuk pengenalan suara menjadi lebih baik dan dapat digunakan oleh berbagai pengguna di dunia.

DAFTAR PUSTAKA

- Aldwairi, M., Flaifel, Y., & Mhaidat, K. (2020). *Efficient WU-MANBER Pattern Matching Hardware for Intrusion and Malware Detection*. <http://arxiv.org/abs/2003.00405>
- Astuti, W. (2017). Analisis String Matching Pada Judul Skripsi Dengan Algoritma Knuth-Morris Pratt (Kmp). *ILKOM Jurnal Ilmiah*, 9(2), 167–172. <https://doi.org/10.33096/ilkom.v9i2.136.167-172>
- Bhardwaj, V., & Garg, V. (2015). Efficient Wu Manber String Matching Algorithm for Large Number of Patterns. *International Journal of Computer Applications*, 132(17), 29–33. <https://doi.org/10.5120/ijca2015907707>
- Chandra, T. (2013). *Simulasi Pencarian Pola Kata Pada File Teks Berdasarkan Multi Pattern Matching Dengan Metode WU-MANBER*. 18.
- Deswar, F. A., & Pradana, R. (2021). *MONITORING SUHU PADA RUANG SERVER MENGGUNAKAN WEMOS D1 R1 BERBASIS INTERNET OF THINGS (IOT)*. 12(1), 25–32.
- Farogi, A., WS, M. S., & Nugraha, R. (2016). Perancangan sistem kontrol otomatis lampu menggunakan metode pengenalan suara berbasis arduino. *TELKA-Jurnal Telekomunikasi, Elektronika, Komputasi Dan Kontrol*, 2(2), 106–117.
- Fernando, M., Jasa, L., & Hartati, R. S. (2022). Monitoring system kecepatan dan arah angin berbasis *Internet of Things (IoT)* menggunakan Raspberry Pi 3. *Majalah Ilmiah Teknologi Elektro*, 21(1), 135.
- Fitriani. (2023). *PENGUNAAN TEKNOLOGI INFORMASI DALAM MENGELOLA ADMINISTRASI SEKOLAH DI SMPN 2 PAREPARE*. 107.
- Laudira, A. (2020). *Palang pintu perlintasan kereta api otomatis berbasis Arduino Uno*.
- Nababan, A. A., & Jannah, M. (2019). Algoritma String Matching Brute Force Dan Knuth-Morris-Pratt Sebagai Search Engine Berbasis Web Pada Kamus Istilah Jaringan Komputer. *Jurnal Mantik Penusa*, 3(Desember), 87–94.
- Nada, Q., Ridhuandi, C., Santoso, P., & Apriyanto, D. (2019). *Speech recognition dengan Hidden Markov Model untuk Pengenalan dan Pelafalan Huruf Hijaiyah. JURNAL AI-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI*, 5(1), 19. <https://doi.org/10.36722/sst.v5i1.319>
- Nadziroh, F. (2021). *ALAT DETEKSI INTENSITAS CAHAYA BERBASIS ARDUINO UNO SEBAGAI PENANDA PERGANTIAN WAKTU SIANG-*

MALAM BAGI TUNANETRA. 1(3).

- Nugraha, A., & Adriansyah, A. (2019). *Analisa Kendali Dan Pemantauan Pintu Perlintasan Kereta Api Berbasis IoT (Internet of Things) Menggunakan Aplikasi MIT Inventor. 10(3), 145–151.*
- Putri, B. B., & Ernawati, D. P. (2015). IMPLEMENTASI METODE WU-MANBER BERDASARKAN MULTI PATTERN MATCHING DALAM PENCARIAN KESAMAAN DNA (Studi kasus: DNA Kanker Hati). *Jurnal Rekursif, 3(2).*
- Putri, G. (2024). Implementasi MIT App Inventor untuk Aplikasi Translator Suara ke Teks di Platform Android. *Jurnal Ilmu Komputer Dan Informatika| E-ISSN: 3063-9026, 1(1), 1–5.*
- Putri, V. A. (2023). *Analisis sentimen kebijakan program pemahadan tahun 2023 UIN Walisongo Semarang dengan metode naive bayes classifier.*
- PyrgIoTis, T. K., Kouzinopoulos, C. S., & Margaritis, K. G. (2012). Parallel implementation of the WU-MANBER algorithm using the OpenCL framework. *IFIP Advances in Information and Communication Technology, 382 AICT(PART 2), 576–583.* https://doi.org/10.1007/978-3-642-33412-2_59
- Rabbani, M. K. (2021). *Sistem kendali pagar rumah berbasis Internet of Things dengan perintah suara menggunakan algoritma Boyer-Moore.* Universitas Islam Negeri Maulana Malik Ibrahim.
- Rumopa, V. W. (2015). *KONTROL PENERANGAN RUANGAN MENGGUNAKAN SENSOR SUARA (SPEECH RECOGNITION) BERBASIS ANDROID.* Politeknik Negeri Manado.
- Siahaan, M., Harsana Jasa, C., Anderson, K., Rosiana, M. V., Lim, S., & Yudianto, W. (2020). Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra. *Journal of Information System and Technology (JOINT), 01 (02)(02), 186–193.*
- Ulum, M. (2022). *Sistem Kontrol Gorden Berbasis Internet of Things menggunakan Algoritma Knuth Morris Pratt.* Universitas Islam Negeri Maulana Malik Ibrahim.
- Ulum, M. B., & Badri, F. (2023). *SISTEM MONITORING CUACA DAN PERINGATAN BANJIR BERBASIS IOT DENGAN MENGGUNAKAN. 11(3), 319–328.*
- Verdian, A., Wantoro, A., Utami, Y. T., Hatta, S. R. M. J. S., Metro, M. K., Nomor, J. S. B., & Lampung, R. B. (2023). *PENERAPAN LOGIKA FUZZY DENGAN FIS MAMDANI PADA PROTOTYPE VOLUME TELEVISI*

SECARA OTOMATIS. *Jurnal Teknik Dan Sistem Komputer (JTIKOM)*, Vol. 4, No. 1, P.

Yusniati. (2018). *Penggunaan Sensor Infrared Switching Pada Motor DC Satu Phasa*. 1099, 90–96.

Zabidi, A. (2020). Kelompok Sosial Dalam Masyarakat Perspektif Qs. Al-Maidah Ayat 2. *Borneo: Journal of Islamic Studies*, 3(2), 42–58. <https://doi.org/10.37567/borneo.v3i2.262>

Zhafira, D. F., Rahayudi, B., & Indriati, I. (2021). Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes dan Pembobotan TF-IDF Berdasarkan Komentar pada Youtube. *Jurnal Sistem Informasi, Teknologi Informasi, Dan Edukasi Sistem Informasi*, 2(1), 55–63. <https://doi.org/10.25126/justsi.v2i1.24>