

**SISTEM KENDALI JENDELA BERBASIS *INTERNET OF THINGS*
MENGUNAKAN ALGORITMA *HORSPool***

SKRIPSI

**Oleh:
ANISA
NIM. 200605110116**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**SISTEM KENDALI JENDELA BERBASIS *INTERNET OF THINGS*
MENGUNAKAN ALGORITMA *HORSPool***

SKRIPSI

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

Oleh :
ANISA
NIM. 200605110116

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN

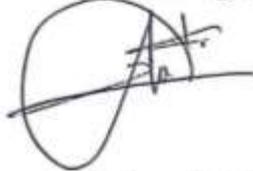
**ISTEM KENDALI JENDELA BERBASIS *INTERNET OF THINGS*
MENGUNAKAN ALGORITMA HORSPOOL**

SKRIPSI

Oleh :
ANISA
NIM. 200605110116

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 21 Mei 2024

Pembimbing I,



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

Pembimbing II,



Dr. Yunifa Miftachul Arif, M. T
NIP. 19830616 201101 1 004

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




Dr. Fachrudin Kurniawan, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

SISTEM KENDALI JENDELA BERBASIS *INTERNET OF THINGS* MENGUNAKAN ALGORITMA *HORSPPOOL*

SKRIPSI

Oleh :
ANISA
NIM. 200605110116

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 21 Mei 2024

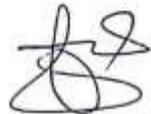
Susunan Dewan Penguji

Ketua Penguji : Dr. M. Amin Hariyadi, M.T
NIP. 19670118 200501 1 001

Anggota Penguji I : Johan Ericka Wahyu P. M. Kom
NIP. 19831213 201903 1 004

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

Anggota Penguji III : Dr. Yunifa Miftachul Arif, M. T
NIP. 19830616 201101 1 004

()
()
()
()

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



PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Anisa
NIM : 200605110116
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Sistem Kendali Jendela Berbasis *Internet of Things*
Menggunakan Algoritma Horspool

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, 01 April 2024
Yang membuat pernyataan,



Anisa
NIM.200605110116

MOTTO

فَإِنَّ مَعَ الْعُسْرِ يُسْرًا إِنَّ مَعَ الْعُسْرِ يُسْرًا

*”Karena sesungguhnya sesudah kesulitan ada kemudahan,
sesungguhnya beserta kesulitan ada kemudahan”*

(QS. Al-Insyirah: 5-6)

HALAMAN PERSEMBAHAN

Puji syukur alhamdulillah atas berkat rahmat Allah SWT hingga skripsi ini dapat diselesaikan dengan lancar. Saya persembahkan karya ini kepada kedua orang tua dan kakak saya yang selalu mendukung, mempercayai dan menyemangati hingga sampai di titik ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dengan menyebut nama Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Kendali Jendela Berbasis *Internet of Things* menggunakan Algoritma *Horspool*” dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW yang telah membawa kita dari zaman jahiliyah menuju zaman yang terang benderang dengan cahaya islam.

Skripsi ini merupakan hasil penelitian yang digunakan untuk memenuhi tugas akhir perkuliahan serta syarat kelulusan Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang. Banyak orang disekeliling penulis yang membantu penyusunan dan penyelesaian tugas skripsi ini. Penulis mengucapkan terima kasih kepada:

1. Prof. Dr. M. Zainuddin, M.A. selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Prof. Dr. Hj. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang.
3. Bapak Dr. Fahrul Kurniawan ST., M.MT., IPM selaku Ketua Program Studi UIN Maulana Malik Ibrahim Malang.
4. Bapak Ajib Hanani, M.T. selaku dosen pembimbing I dan wali dosen yang telah banyak membantu dan memberikan arahan dalam menyelesaikan setiap

permasalah-masalahan yang ditemukan selama perjalanan penelitian dan penulisan skripsi.

5. Bapak Dr. Yunifa Miftachul Arif, M.T selaku dosen pembimbing II yang dengan sabar membimbing dan memberikan arahan dalam proses penulisan integrasi islam dan penulisan skripsi.
6. Bapak Dr. Amin Haryadi, M.T. selaku dosen penguji I dan Bapak Johan Ericka Prakasa, M.Kom. selaku dosen penguji II yang telah memberikan ilmu, kritikan dan saran untuk penulis agar penulis dapat menyelesaikan skripsi dengan baik.
7. Seluruh Dosen dan jajaran staff program studi Teknik Informatika yang senantiasa membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian skripsi penulis.
8. Orang tua penulis Bapak Andy Stahar Anwar dan Ibu Sadariah yang senantiasa selalu mendukung dan memberikan doa-doa terbaik dalam perjalanan penulis menyelesaikan Pendidikan.
9. Kakak Penulis Hendreg Kelana Saputra, Febry Dwi Andar Saputra, dan Dody Augusto Wijaya yang selalu mendukung serta memberikan doa-doa terbaik dalam perjalanan penulis menyelesaikan Pendidikan.
10. Sahabat Zulfa, Viola, Jihan, Zahro, Qorina, Dilla, Izzul, serta teman-teman INTEGER 2020 yang selalu membantu, memotivasi, serta menemani penulis selama perkuliahan ini.
11. Seluruh keluarga dan teman-teman yang tidak dapat disebutkan satu per satu, yang turut dalam memberikan bantuan, dukungan serta doa terbaik untuk penulis.

12. Diri sendiri yang selalu berusaha dan tidak pernah menyerah untuk menuju ke tahap ini.

Penulis menyadari bahwa skripsi ini jauh dari kata sempurna sehingga masih banyak kekurangan yang perlu dibenahi oleh penulis. Oleh karena itu, penulis terbuka dalam menerima saran yang membangun dari pembaca sebagai sarana untuk memperbaiki penelitian. Penulis juga berharap penelitian yang dilakukan akan bermanfaat bagi pembaca dan masyarakat umum.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 01 Juni 2024

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Penelitian	5
1.5 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Internet of Things</i> (IoT).....	7
2.2 <i>Smart Home</i>	8
2.3 <i>String Matching</i>	8
2.4 Algoritma <i>Horspool</i>	9
2.5 <i>Speech Recognition</i>	10
2.6 <i>Stemming</i>	11
2.7 MIT App Inventor	11
2.8 Kerangka Teori.....	12
BAB III METODOLOGI PENELITIAN	14
3.1 Desain dan Implementasi	14
3.1.1 Akuisisi Data	15
3.1.2 Proses <i>Stemming</i>	16
3.1.3 <i>Preprocessing</i> Algoritma <i>Horspool</i>	17
3.1.4 Pencocokan Algoritma <i>Horspool</i>	19
3.2 Perancangan Aplikasi Berbasis <i>Mobile</i>	20
3.3 Rangkaian Komponen	21
3.4 Pengujian Sistem	22
BAB IV HASIL DAN PEMBAHASAN	24
4.1 Implementasi Sistem	24
4.1.1 Halaman Aplikasi	24
4.2 Pengujian Sistem.....	26
4.2.1 Kendali menggunakan <i>Microphone</i>	26
4.2.2 Kendali menggunakan <i>Button</i>	28
4.2.3 Proses <i>Table Bad Match</i>	30

4.2.4 Pengujian Algoritma Horspool	31
4.2.5 Pengujian Algoritma Horspool Menggunakan <i>Confusion Matrix</i>	37
4.3 Rangkaian Hardware	40
4.4 Hasil Pengujian Alat	42
4.5 Integrasi Islam	44
BAB V PENUTUP	46
5.1 Kesimpulan	46
5.2 Saran.....	46
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 3.1 Desain Sistem.....	15
Gambar 3.2 Flowchart Stemming	16
Gambar 3.3 Contoh Implementasi Algoritma Horspool	19
Gambar 3.4 Contoh Pengujian menggunakan Algoritma Horspool	20
Gambar 3.5 Rancangan Aplikasi Berbasis Mobile	20
Gambar 3.6 Rancangan Komponen	21
Gambar 4.1 Tampilan Utama	24
Gambar 4.2 Tampilan Penggunaan Speech Recognition	25
Gambar 4.3 Halaman Tutup Jendela.....	26
Gambar 4.4 Logika Button Mic	27
Gambar 4.5 Output dari Logika Button Mic	28
Gambar 4.6 Button Buka.....	29
Gambar 4.7 Output Terminal dari Logika Button Tutup	29
Gambar 4.8 Button Tutup	29
Gambar 4.9 Output Terminal dari Logika Button Buka	30
Gambar 4.10 Output Tabel Bad Match	31
Gambar 4.11 Grafik “buka jendela”.....	32
Gambar 4.12 Grafik “tolong ditutup jendelanya“	33
Gambar 4.13 Grafik “mohon untuk jendela dibukakan “	34
Gambar 4.14 Grafik “ayo dong jangan dibuka jendelanya“	35
Gambar 4.15 Rata-rata durasi berdasarkan Pattern.....	37
Gambar 4.16 Rangkaian Komponen	41
Gambar 4.17 Memberi Perintah Membuka Jendela.....	42
Gambar 4.18 Output Miniatur Membuka Jendela.....	42
Gambar 4.19 Memberi Perintah Menutup Jendela	43
Gambar 4.20 Output Miniatur Tutup Jendela	43

DAFTAR TABEL

Tabel 2.1 Daftar Jurnal.....	13
Tabel 3.1 Panjang Kata “buka jendela”	17
Tabel 3.2 Indeks Kata Buka	17
Tabel 3.3 Tabel Bad-Match	18
Tabel 3.4 Tabel Confusion Matrix	22
Tabel 4.1 Hasil pengujian kalimat “buka jendela”.....	32
Tabel 4.2 Hasil pengujian kalimat “tolong ditutup jendelanya”	33
Tabel 4.3 Hasil pengujian kalimat “mohon untuk jendela dibukakan”	34
Tabel 4.4 Hasil pengujian kalimat “ayo dong jangan dibuka jendelanya”	35
Tabel 4.5 Rata-rata Durasi Pattern	36
Tabel 4.6 Pattern buka jendela	38
Tabel 4.7 Hasil Confusion Matrix.....	40

ABSTRAK

Anisa. 2024. **Sistem Kendali Jendela Berbasis *Internet Of Things* Menggunakan Algoritma Horspool**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T (II) Dr. Yunifa Miftachul Arif, M. T.

Kata kunci: *Horspool, Internet of Things, Pattern, Smart Home, Speech Recognition.*

Perkembangan *internet of things* memberi kemudahan dalam hidup manusia, khususnya disabilitas. Salah satu perkembangannya adalah *smart home* yang memiliki konsep rumah dapat dikendalikan melalui *smartphone*, seperti kendali jendela hanya dengan tombol pada aplikasi *smartphone*. Selain kendalinya menggunakan tombol dapat juga digunakan dengan *speech recognition*. Untuk merancang *speech recognition* pada aplikasi kendali jendela dibutuhkan sebuah algoritma untuk mencocokkan antara *pattern* atau pola dari algoritma dengan kalimat yang diucapkan. Algoritma yang digunakan adalah algoritma *horspool* dengan cara membandingkan *pattern bad match table* dengan kalimat yang diucapkan mulai dari paling kanan *pattern*. Penelitian ini menghasilkan akurasi sebesar 92% menggunakan 8 *pattern*, 32 kalimat dengan perintah benar, dan 8 kalimat perintah salah.

ABSTRACT

Anisa. 2024. **Internet of Things Based Window Control System Using Horspool Algorithm**. Thesis. Study Program of Informatics Engineering, Faculty of Science and Technology, State Islamic University, Maulana Malik Ibrahim Malang. Advisors: (I) Ajib Hanani, M.T (II) Dr. Yunifa Miftachul Arif, M. T.

The development of the internet of things can provide convenience in human life, especially disability. One of the developments is the smart home which has the concept that the house can be controlled via a smartphone, such as window control with just a button on the smartphone application. In addition to control using buttons, it can also be used with speech recognition. To design speech recognition on window control applications, an algorithm is needed to match the *Pattern* of the algorithm with the spoken sentence. The algorithm used is the horspool algorithm by comparing the bad match table *Pattern* with the spoken sentence starting from the rightmost *Pattern*. This research resulted in an accuracy of 92% using 8 *Patterns*, 32 sentences with correct commands, and 8 sentences with wrong commands.

Keywords: Horspool, Internet of Things, *Pattern*, Smart Home, Voice Recognition.

مستخلص البحث

أنيسة. 2024. نظام التحكم في النوافذ على أساس إنترنت الأشياء باستخدام خوارزمية هورسبول. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: عجيب حناني، الماجستير. المشرف الثاني: د. يونيفة مفتاح العارف، الماجستير.

الكلمات الرئيسية: هورسبول، إنترنت الأشياء، النمط، المنزل الذكي، التعرف على الكلام.

لقد جعل تطور إنترنت الأشياء حياة الإنسان أسهل، خاصة لذوي الإعاقة. أحد التطورات هو المنزل الذكي الذي يحتوي على مفهوم المنزل الذي يمكن التحكم فيه من خلال الهاتف الذكي، مثل التحكم في النوافذ باستخدام زر واحد فقط على تطبيق الهاتف الذكي. بالإضافة إلى التحكم في استخدام الأزرار، يمكن استخدامه أيضا مع التعرف على الكلام. لتصميم التعرف على الكلام في تطبيق التحكم في النافذة، هناك حاجة إلى خوارزمية لمطابقة نمط الخوارزمية مع الجملة المنطوقة. الخوارزمية المستخدمة هي خوارزمية هورسبول من خلال مقارنة نمط جدول المطابقة السيئة مع الجمل المنطوقة بدءا من أقصى يمين النمط. أنتج هذا البحث دقة بنسبة 92٪ باستخدام 8 أنماط و 32 كلمة بأوامر صحيحة و 8 كلمة بأوامر خاطئة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi saat ini, perkembangan teknologi yang berkembang semakin pesat tidak hanya pada satu bidang, tetapi di berbagai bidang kehidupan. Berkembangnya teknologi dapat diketahui dari inovasi yang kian hari memberikan manfaat kemudahan dalam kehidupan manusia, seperti memudahkan pekerjaan manusia yang semula membutuhkan waktu yang lama untuk menyelesaikan suatu pekerjaan, kemudian dengan menggunakan teknologi estimasi waktu untuk menyelesaikan pekerjaan tidak membutuhkan waktu yang lama. Kemudahan lainnya, yaitu memudahkan dalam mencari informasi pengetahuan dengan bijak di internet melalui alat ponsel pintar atau *smartphone*. Melalui manfaat yang dirasakan sehingga penggunaan teknologi tiap tahun semakin meningkat, seperti halnya pengguna internet dan *smartphone*. Berdasarkan laporan data *We Are Social* pengguna internet di Indonesia pada tahun 2023 mencapai 77.0% atau sekitar 212.9 juta jiwa. Kemudian, untuk pengguna *smartphone* mencapai 128% atau setara 353.8 juta (wearesocial.com, 2023). Berdasarkan angka tersebut dapat disimpulkan bahwa rata-rata setiap jiwa memiliki lebih dari satu *smartphone*.

Sesuai dengan namanya ponsel pintar atau *smartphone* memiliki berbagai fitur canggih, salah satunya pemanfaatan fitur *microphone* dengan *google speech recognizer*. Contoh pemanfaatannya adalah menyetel alarm menggunakan suara sehingga tidak perlu menggunakan cara manual dengan *setting* dengan tompo untuk

memudahkan menyetel alarm. Berdasarkan contoh tersebut dapat dikembangkan lagi untuk kehidupan manusia dengan mengkombinasikan fitur *google speech recognizer* dengan *Internet of Things*. Kombinasi dari hal tersebut akan menciptakan sebuah produk untuk *smart home*. *Smart home* adalah sistem dengan bantuan computer yang diaplikasikan pada gedung, rumah, dan bangunan lainnya (Romoadhon & Anamisa, 2017). *Smart home* gambaran rumah yang mencerminkan era saat ini, seperti halnya saat menyalakan lampu rumah tidak perlu dilakukan secara manual dengan klik saklar lampu, tetapi dengan menerapkan *Internet of Things* menggunakan suara untuk kontrol lampu (Hanani & Haryadi, 2020).

Pada penelitian ini akan dikembangkan penelitian teknologi khususnya pada *smart home*, yaitu rumah pintar untuk mereka yang memiliki cacat fisik sehingga dapat mempermudah kegiatan mereka di rumah apabila tidak ada orang di sekitarnya. Salah satu komponen rumah yang digunakan untuk penelitian ini adalah jendela. Dapat diketahui jendela merupakan salah satu komponen rumah yang penting karena dengan adanya jendela dapat mentransfer udara yang segar untuk masuk ke dalam rumah serta mengeluarkan aroma tidak sedap yang ada di dalam rumah. Biasanya penggunaan *smart home* untuk jendela menggunakan sensor suhu yang mana apabila suhu panas maka jendela akan terbuka secara otomatis dan apabila suhu dingin jendela akan menutup secara otomatis. Tetapi, pada penelitian ini ingin mencoba menggunakan perintah suara untuk mengendalikan jendela. Seperti yang dijelaskan sebelumnya, saat pemilik rumah ingin mengeluarkan aroma tidak sedap dan cuaca di luar sedang dingin maka dapat dikendalikan menggunakan perintah suara. Juga, apabila kualitas udara diluar ruangan sedang buruk dan cuaca

sedang panas maka bisa dikendalikan jendela tetap tertutup. Sedangkan, apabila menggunakan sensor suhu maka harus dalam keadaan cuaca hujan atau dingin untuk menutup jendela secara otomatis dan dalam permasalahan yang dihadapi maka jendela harus ditutup walaupun cuaca sedang panas atau tanpa adanya sensor suhu.

Dengan adanya penelitian kendali jendela menggunakan *handphone* dapat membantu disabilitas dalam kegiatannya sehari-hari. Menolong orang lain khususnya disabilitas dengan tujuan memudahkan aktivitasnya merupakan tindakan tolong menolong dalam kebaikan. Dapat diketahui hal tersebut dari integrasi islam dalam surah Al-Maidah ayat 2.

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya”.(Q.S Al-Maidah: 2).

Berdasarkan tafsir singkat kemenag surah Al-Maidah ayat 2 bagian tolong menolong adalah tolong-menolonglah kamu dalam mengerjakan kebajikan, melakukan yang diperintahkan Allah, dan takwa, takut kepada larangannya, dan jangan tolong-menolong dalam berbuat dosa, melakukan maksiat dan permusuhan, sebab yang demikian itu melanggar hukum-hukum Allah. Bertakwalah kepada Allah, takut kepada Allah dengan melakukan perintah-Nya dan meninggalkan larangan-Nya, karena sungguh Allah sangat berat siksaan-Nya kepada orang-orang yang tidak taat kepada-Nya.

Agar proses sistem penelitian berjalan sebagaimana mestinya maka diperlukan sebuah algoritma untuk memperoleh kecocokan perintah suara (Sunny, 2009). Terkadang dalam penggunaan algoritma sering kali tidak akurat, sehingga perlu dihitung agar dapat diketahui keakuratan algoritma tersebut. Pada penelitian ini algoritma yang digunakan adalah algoritma *horspool* karena algoritma ini adalah turunan dari boyer-moore yang dimana pada algoritma dalam proses *string matching boyer-moore* menggunakan dua tabel, yaitu *bad-character shift* dan *good suffix shift*. Tetapi, pada algoritma *horspool* hanya menggunakan satu tabel, yaitu *bad-character shift* (Waruwu & Hondro, 2019). Jadi, dalam proses implementasinya lebih sederhana dan mudah dipahami. Untuk proses menggunakan algoritma *horspool*, pertama menentukan *pattern* sebagai kata kunci, kemudian menghitung jumlah karakter *pattern* dan dikurangi dengan jumlah indeks dari *pattern* (Frigustini et al., 2018). Apabila sudah dihitung maka akan dicocokkan dengan kalimat yang ada. Apabila cocok maka berhenti pada iterasi pertama dan apabila tidak cocok maka akan bergeser sesuai dengan hasil pengurangan karakter *pattern*.

Implementasi *Internet of Things* menggunakan perintah suara membutuhkan sebuah algoritma, seperti algoritma *horspool* karena dengan adanya algoritma dapat mengetahui letak kata kunci yang dicari dari kalimat yang tersusun. Apabila tidak menggunakan algoritma maka akan susah mencari kata kunci dalam kalimat yang tersusun dari banyaknya karakter atau abjad. Oleh karena itu, penulis memutuskan untuk membuat “Sistem Kendali Jendela Berbasis *Internet of Things* menggunakan Algoritma *Horspool*”.

1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dipaparkan, rumusan masalah yang diambil adalah mengenai bagaimana tingkat akurasi algoritma *horspool* dalam sistem kendali jendela berbasis *Internet of Things* untuk membantu disabilitas.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini untuk mengetahui tingkat akurasi algoritma *horspool* dalam sistem kendali jendela berbasis *Internet of Things* sehingga dapat diketahui berapa persen keberhasilan penggunaan *speech recognition* menggunakan *horspool* dalam membantu disabilitas.

1.4 Batasan Penelitian

Adapun batasan dalam penelitian sebagai berikut:

- a. Jendela tipe geser
- b. Kendali jendela buka dan tutup.

1.5 Manfaat Penelitian

Beberapa manfaat dari penelitian, diantaranya:

- a. Untuk masyarakat, memudahkan orang yang cacat fisik atau lansia dalam mengontrol jendela tanpa perlu manual untuk membuka dan menutup jendela. Kemudian, memberikan kenyamanan seseorang tanpa perlu berdiri dan beranjak ke jendela untuk membuka serta menutup jendela.
- b. Untuk pendidikan, dalam membuat sistem kendali jendela berbasis IoT dapat dijadikan sebagai pendekatan inovatif dalam proses belajar mengenai teknologi

sehingga untuk kedepannya bisa mengembangkan teknologi melalui penelitian ini.

- c. Untuk mengetahui seberapa akurat algoritma *horspool* dalam mendeteksi *pattern* pada sebuah teks yang diucapkan *user*.
- d. Hasil dari penelitian ini dapat dikembangkan lagi untuk sistem kendali jendela otomatis yang lebih canggih.

BAB II

TINJAUAN PUSTAKA

2.1 *Internet of Things (IoT)*

Internet of Things atau yang sering disingkat IoT adalah suatu objek memiliki kemampuan dalam mentransfer data melalui jaringan sehingga tidak perlu ada interaksi manusia. IoT mulai ada pada saat konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan internet. IoT biasanya didefinisikan terhadap RFID, yaitu sebagai metode komunikasi. Selain itu, juga mencakup teknologi sensor, baik nirkabel maupun kode QR yang sudah tidak asing dalam kehidupan sehari-hari. Kemampuan dari IoT yang dapat diketahui lainnya adalah dalam berbagi data menjadi remote control (Setiawan, 2021).

Menurut Casagras (*Coordination and support action for global RFID-related activities and standardisation*), IoT berfungsi sebagai infrastruktur jaringan yang ada dan menggunakan internet sebagai pengembangan jaringan yang menghubungkan objek fisik dan virtual melalui eksploitasi data *capture* serta kemampuan komunikasi.

Menurut SAP (*Systeme, Anwendungen und Produkte*), mengemukakan IoT adalah dunia yang di dalamnya jaringan informasi disatukan dengan objek fisik secara berkesinambungan dan berperan aktif dalam melaksanakan proses bisnis serta memperhatikan privasi dan keamanan.

Menurut EPOSS, IoT adalah sebuah jaringan yang dibentuk oleh objek yang memiliki identitas di internet yang beroperasi menggunakan kecerdasan antarmuka

agar terhubung dan saling komunikasi dengan pengguna, konteks sosial dan lingkungan.

Cara kerja *Internet of Things* menggunakan pemrograman yang memiliki perintah untuk menghasilkan interaksi antar mesin yang terhubung melalui internet. Oleh karena itu, dalam proses kerja IoT tidak ada campur tangan manusia dalam proses interaksinya. Manusia hanya mengatur dan mengawasi mesin yang bekerja saat proses interaksi objek dengan mesin-mesin.

2.2 *Smart Home*

Smart home atau rumah pintar adalah penggabungan teknologi dengan aplikasi yang dengan namanya, yaitu di rumah atau gedung. Adapun fungsi diciptakan *smart home* antara lain, untuk meningkatkan keamanan, memudahkan dan memberikan kenyamanan penghuninya. *Smart home* memiliki sistem yang biasanya terdiri dari perangkat monitoring, kontrol, dan beberapa perangkat lainnya yang diakses di komputer, seperti *cloud computing* (Putri & Yendri, 2018).

2.3 *String Matching*

String merupakan sebuah susunan karakter yang dapat berupa alfabet, angka, dan lain-lainnya. Implementasi pada *string* yang diketahui, seperti kata, frase dan kalimat. Pencocokan *string* itu sangat penting dari proses pencarian karakter dalam sebuah teks, dokumen, dan lain-lain. Kemudian, untuk hasil dari penggunaan *string matching* itu tergantung dari metode *string matching* yang digunakan. Untuk metode atau algoritma dari pencocokan *string*, seperti algoritma *zhu-takaoka*, *horspool*, dan lain-lainnya (Ernawati et al., 2019). Menurut

Dictionary of Algorithms and Data Structures, Natiina Insitute of Standars, pencocokan *string* merupakan sebuah permasalahan dalam mencari atau menemukan pola susunan dari karakter di dalam karakter *string* lainnya.

Beberapa penelitian terdahulu dilakukan untuk *string matching* menggunakan algoritma. Pada riset oleh Dongoran et al. (2021) dilakukan pencarian file pada sistem operasi linux. Dapat diketahui dalam proses pencarian file yang banyak dilakukan secara manual memerlukan waktu yang lama sehingga diimplementasikan sebuah algoritma untuk mempersingkat pencarian file dengan menggunakan algoritma *zhu-thakaoka*. Hasil dari penelitian tersebut sangat efektif dalam proses pencarian file.

Penelitian lainnya dari Mesi & Oktarina (2021) dilakukan proses pencarian pencatatan obat menggunakan algoritma *horspool*. Penelitian ini dilakukan untuk mempersingkat waktu dalam memantau ketersediaan obat sehingga dapat meringankan pekerjaan pegawai apotek. Hasil dari penelitian ini akurat dan cepat dalam proses pencarian stok obat.

Selanjutnya, riset oleh Filcha & Hayaty (2019) dilakukan *string matching* pada dokumen tugas mahasiswa menggunakan algoritma *rabin-karp*. Tujuan penelitian ini untuk memudahkan deteksi plagiarisme dengan cepat dan tepat. Hasil dari penelitian ini adalah algoritma *rabin-karp* sangat cocok untuk *string matching* dengan pola yang panjang berdasarkan perhitungan *confusion matrix*, yaitu 90%.

2.4 Algoritma Horspool

Algoritma *horspool* adalah sebuah algoritma turunan dari algoritma *boyer-moore* yang ditemukan oleh R. Nigel Horspool. Karena algoritma *horspool* turunan

dari algoritma *boyer-moore* maka proses kerjanya sama, yaitu nilai pergeseran karakternya dimulai dari yang paling kanan. Adapun perbedaan dari *boyer-moore* dengan *horspool*, yaitu pada algoritma *horspool* tidak melakukan lompatan berdasarkan karakter pada kata kunci *pattern* yang tidak cocok dengan karakter atau abjad dari kalimat (Mesi & Oktarina, 2021).

Pada riset oleh Fuadi et al., (2020) algoritma *horspool* digunakan untuk proses pencarian doa Nabi dan Rasul. Dapat diketahui dalam proses pencarian doa di Al-Qur'an secara manual membutuhkan waktu yang lama sehingga diperlukan penelitian ini untuk mempermudah proses pencarian. Hasil dari pencarian doa Nabi dan Rasul dengan mengimplementasikan algoritma *horspool* adalah cepat dan akurat.

Penelitian terdahulu lainnya dari riset Abidin et al. (2023) adalah mengembangkan algoritma *horspool* dalam proses *translate* bahasa Indonesia ke bahasa daerah. Penelitian ini dilakukan untuk melestarikan bahasa daerah serta memudahkan pengguna memahami bahasa daerah yang tidak mereka ketahui. Hasil penelitian ini dengan menggunakan algoritma *horspool* adalah berjalan dengan akurat.

2.5 *Speech Recognition*

Speech Recognition adalah sebuah proses yang terjadi secara otomatis dalam mengidentifikasi kata-kata yang diucapkan manusia melalui sinyal suara (Mohamad Salman Farizi et al., 2022). Implementasi nyata dalam kehidupan sehari-hari adalah penggunaan *Google Assistant* dari aplikasi android. Penggunaannya

mengatakan “ok *google*, nyalakan alarm jam 4 pagi” maka *google assistant* akan menyetel alarm jam 4 pagi.

Melalui *google assistant* dapat dikembangkan dan diimplementasikan menjadi proses perintah sistem kendali menggunakan suara. Kemudian akan diintegrasikan pada mikrokontroler melalui *smartphone* dalam bentuk aplikasi. Jadi, dapat mengendalikan atau mengontrol perangkat elektronik di rumah menggunakan perintah suara tanpa harus secara manual.

2.6 Stemming

Stemming adalah sebuah proses dalam mencari kata dasar dalam suatu kata di kalimat. Proses *stemming* meliputi penghilangan imbuhan dari awalan, akhiran, dan gabungan awalan akhiran pada suatu kata turunan. Penggunaan *stemming* dengan mengganti bentuk suatu kata menjadi kata dasar dari sebuah kata yang sesuai dengan aturan kata morfologi bahasa Indonesia. Jadi, imbuhan dari suatu kata dihilangkan agar bentuknya menjadi kata dasar, contoh kata membuka agar menjadi kata dasar maka imbuhan awalan mem-nya dihilangkan sehingga kata dasarnya menjadi buka (Dinata et al., 2020).

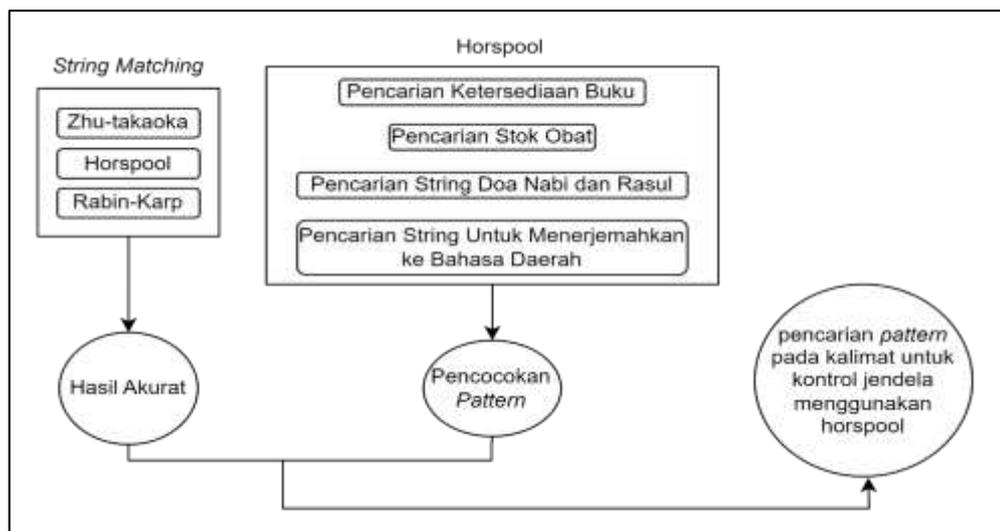
2.7 MIT App Inventor

Aplikasi MIT App Inventor adalah sebuah tempat pembuatan aplikasi android atau *mobile* berbasis *website* dengan cara *drag and drop* atau dengan kata lain tanpa perlu menggunakan *code* untuk tampilan serta logikanya. MIT App inventor ini berbasis visual *block programming* sehingga tidak perlu mempelajari bahasa pemrograman yang begitu banyak jenisnya dalam membuat aplikasi serta

code untuk mengatur layout dan komponen lainnya. Dengan adanya MIT App Inventor dapat memudahkan dan mempersingkat waktu dalam pembuatan aplikasi perangkat lunak *mobile* dengan sistem operasi *android* (psti.unisayogya.ac.id, 2020).

2.8 Kerangka Teori

Sebelum melakukan penelitian pencarian pattern menggunakan algoritma *horspool* perlu mengacu pada penelitian terkait sebelumnya. Berikut gambar kerangka teori yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 Kerangka Teori

Pada Gambar 2.1 kerangka teori dalam penelitian ini dilihat dari pendekatan *string matching*. Dalam proses *string matching* dibutuhkan sebuah algoritma untuk mencapai hasil yang akurat. Berdasarkan penelitian terkait sebelumnya algoritma yang digunakan, yaitu *zhu-thakaoka*, *horspool*, dan *rabin-karp*. Kemudian, dilakukan pendekatan dari algoritma *horspool* untuk diketahui macam-macam penggunaan algoritma tersebut dalam berbagai studi kasus sehingga memutuskan

penggunaan *horspool* dalam penelitian ini.

Tabel 2.1 Daftar Jurnal

No	Peneliti (Tahun)	Algoritma dan Studi Kasus	Metode Penelitian	Hasil Penelitian
1	(Dongoran et al., 2021)	Implementasi Algoritma Zhu-Takaoka Pada Data Repository Linux Berbasis Web	<i>Zhu-thakaoka</i>	sangat efektif dalam proses pencarian file
2	(Mesi & Oktarina, 2021)	Penerapan Algoritma Horspool Pada Sistem Pendataan Obat Pada Apotek Fajar Mas	<i>Horspool</i>	akurat dan cepat dalam proses pencarian stok obat
3	(Filcha & Hayaty, 2019)	Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa	<i>Rabin-Karp</i>	Akurat dalam memproses pola yang panjang
4.	(Kusnadi & Wicaksono, 2017)	Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop	- <i>Horspool</i> - <i>Zhu-thakaoka</i>	- Uji coba 1, <i>horspool</i> lebih cepat 19.82845% - Uji coba 2, <i>horspool</i> lebih cepat 15.9442%
5.	(Waruwu & Hondro, 2019)	Algoritma <i>horspool</i> digunakan untuk pencarian buku pada aplikasi katalog buku perpustakaan	<i>Horspool</i>	Algoritma <i>horspool</i> termasuk algoritma yang proses pencarian <i>pattern</i> cepat dalam pola pendek
6.	(Fuadi et al., 2020)	Aplikasi Doa Para Nabi Dan Rasul Dalam Al-Qur'an Menggunakan Algoritma <i>Horspool</i> Berbasis Android	<i>Horspool</i>	Penggunaan <i>horspool</i> akurat dalam proses pencarian doa dalam Al-Qur'an
7.	(Abidin et al., 2023)	Algoritma <i>horspool</i> digunakan untuk aplikasi <i>translate</i> indonesia ke bahasa daerah	<i>Horspool</i>	Hasil pengujian algoritma <i>horspool</i> dalam menerjemahkan kata hanya 1 detik

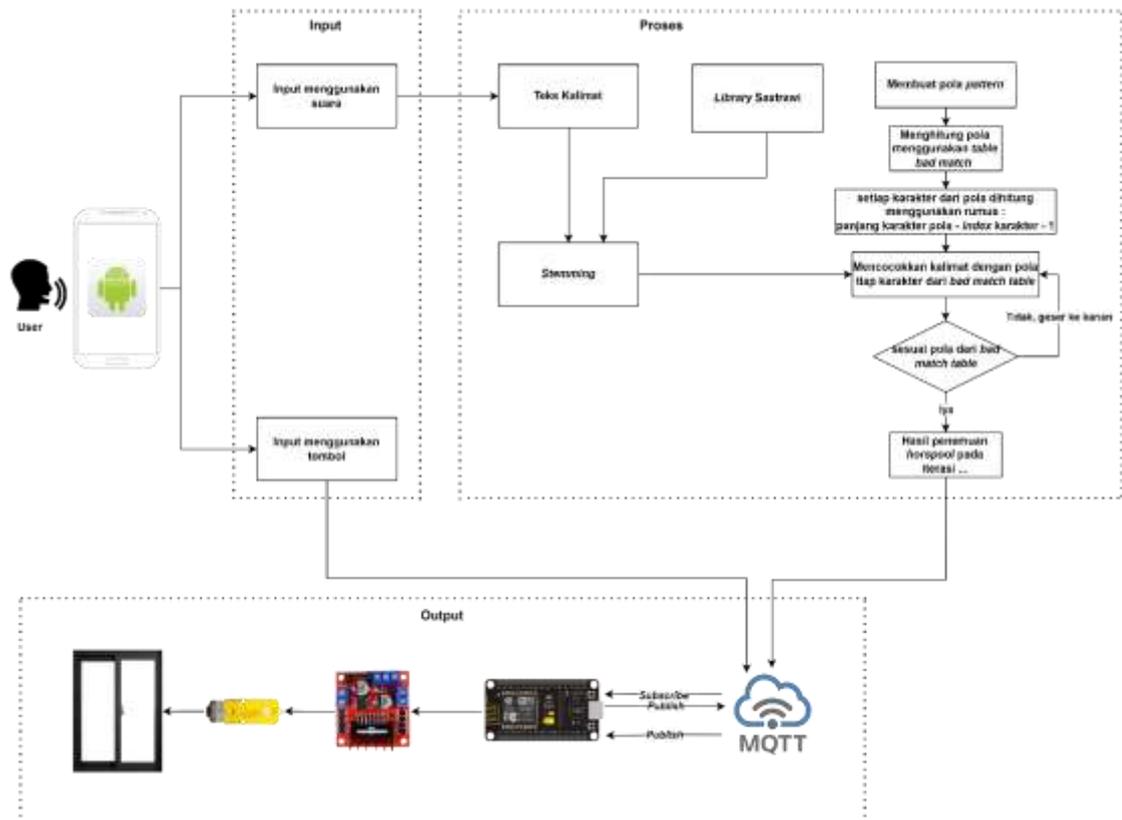
Berdasarkan Tabel 2.1 pada penelitian terkait sebelumnya menghasilkan algoritma *horspool* yang terbaik dalam proses pencarian pola yang pendek dibandingkan dengan *rabin-karp* dan *zhu-thakaoka* termasuk cepat dalam proses pencarian *string*. Hasil dari algoritma *rabin-karp* akurat dengan pencarian pola yang panjang dan *zhu-thakaoka* yang dibandingkan dengan *horspool* menghasilkan *horspool* lebih cepat dalam uji coba sebanyak 2 kali dalam penelitian (Kusnadi & Wicaksono, 2017).

BAB III

METODOLOGI PENELITIAN

3.1 Desain dan Implementasi

Desain dan implementasi pada penelitian ini memiliki beberapa tahapan, diantaranya perancangan sistem, *input*, proses, *output*, dan pengujian sistem. Input dalam penelitian ini dapat melalui tombol dan kendali suara menggunakan *google speech recognizer* untuk membuka serta menutup jendela. Untuk *input* dengan tombol akan langsung ke tahap *output* dengan hasil yang sesuai pada tombol yang diklik. Sedangkan, apabila dengan input kendali suara menggunakan *google speech recognizer*, maka akan melalui tahap akuisisi data yang dimana apabila data telah siap untuk diproses maka akan lanjut pada tahap proses. Data yang siap diproses adalah hasil suara yang diubah menjadi teks. Setelah diubah menjadi teks, proses selanjutnya adalah *stemming* menggunakan *library* sastrawi, yaitu mengubah kata atau karakter dalam teks menjadi kata dasar agar menjadi baku. Setelah teks sudah baku maka akan dilakukan pencocokkan teks dengan *pattern* menggunakan algoritma *horspool*. Proses algoritma *horspool* diawali dengan membuat pola *pattern*. Kemudian, dilanjutkan perhitungan tabel nilai tiap karakter pada *pattern* yang disebut *table bad match*. Selanjutnya, dilakukan pencocokan *pattern* dengan kalimat yang sudah diproses pada tahapan *stemming* dan dilakukan perulangan atau iterasi hingga bertemu *pattern* yang sesuai. Berikut alur desain dan implementasinya pada Gambar 3.1.



Gambar 3.1 Desain Sistem

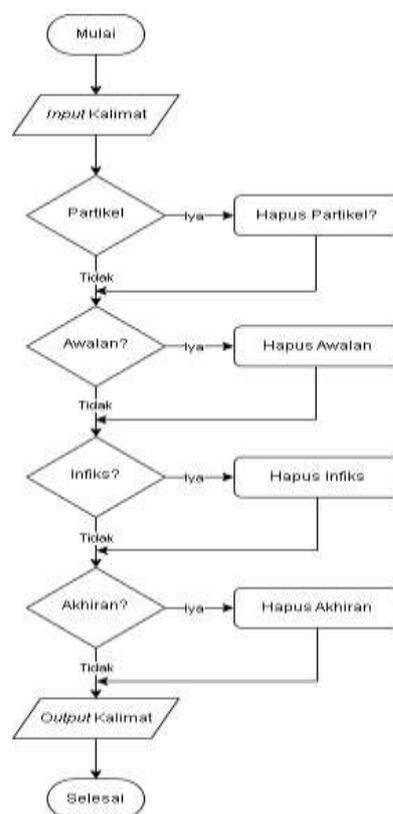
Input dapat menggunakan suara dan tombol. Jika menggunakan suara akan masuk ke proses algoritma horspool untuk diproses kalimat yang telah diucapkan *user*. Sedangkan, apabila menggunakan tombol maka langsung ke output MQTT atau tidak diproses menggunakan algoritma *horspool*.

3.1.1 Akuisisi Data

Data yang digunakan untuk penelitian ini berupa primer dan sekunder. Untuk data primer, diperoleh melalui input kendali suara menggunakan *google speech recognizer*. Sedangkan untuk data sekunder, melalui kata kunci *pattern*, seperti buka jendela, tutup jendela, jendela buka, dan jendela tutup.

3.1.2 Proses *Stemming*

Sebelumnya untuk proses *stemming* bahasa yang digunakan adalah bahasa Indonesia. Seperti yang telah dijelaskan pada bab 2 tinjauan pustaka, *stemming* adalah sebuah proses mengubah kata yang memiliki imbuhan menjadi kata dasar. Untuk diproses dalam pemrograman, pada penelitian ini menggunakan *library sastrawi* dengan bahasa pemrograman PHP. Alur *stemming* menggunakan *library sastrawi* ditunjukkan pada *flowchart* Gambar 3.2.



Gambar 3.2 *Flowchart Stemming*

Penggunaan *stemming* terdiri dari penghapusan beberapa kriteria, yakni partikel, awalan, infiks, dan akhiran. Awalan penambahan morfem pada awal kata, contoh awalan “ber” pada “bermain”. Infiks merupakan morfem yang diletakkan di

tengah kata atau membentuk kata kerja, contoh kata dasar “baca” digunakan pada “me” menjadi “membaca”. Selanjutnya, akhiran penambahan morfem pada akhir kata, contoh akhiran “kan” pada kata “meletakkan”. Partikel merupakan morfem yang bukan dari awalan, akhiran, dan infiks contohnya kata “sangat”.

3.1.3 Preprocessing Algoritma Horspool

Sebelum melakukan pencocokan menggunakan *horspool*, hal-hal yang harus dilakukan, sebagai berikut:

- a. Menentukan kata kunci *pattern*. Misal kata kunci *pattern* “buka jendela”.
- b. Menghitung panjang kata kunci *pattern*.

Contoh:

Tabel 3.1 Panjang Kata “buka jendela”

B	U	K	A		J	E	N	D	E	L	A
1	2	3	4	5	6	7	8	9	10	11	12

Jadi, panjangnya 12.

- c. Setelah itu menghitung indeks dari setiap karakter kata kunci *pattern*.

Tabel 3.2 Indeks Kata Buka

B	U	K	A		J	E	N	D	E	L	A
0	1	2	3	4	5	6	7	8	9	10	11

Jadi, indeksnya 11.

- d. Langkah selanjutnya, menghitung *value* masing-masing karakter kecuali karakter paling kanan atau indeks terakhir dan karakter yang tidak ada pada *pattern* langsung diberi *value* sesuai nilai panjang karakter. Rumus menghitung *value* menggunakan rumus (3.1):

$$value = m - i - 1 \quad (3.1)$$

Keterangan:

m = panjang karakter

i indeks masing-masing karakter

Contoh perhitungan *pattern* buka jendela menggunakan rumus (3.1).

(B) $value = 12 - 0 - 1 = 11$

(U) $value = 12 - 1 - 1 = 10$

(K) $value = 12 - 2 - 1 = 9$

(A) $value = 12 - 3 - 1 = 8$

() $value = 12 - 4 - 1 = 7$

(J) $value = 12 - 5 - 1 = 6$

(E) $value = 12 - 6 - 1 = 5$

(N) $value = 12 - 7 - 1 = 4$

(D) $value = 12 - 8 - 1 = 3$

(E) $value = 12 - 9 - 1 = 2$

(L) $value = 12 - 10 - 1 = 1$

(A) $value = 12$ (karakter terakhir)

Setiap karakter terakhir pada *pattern* akan bernilai sepanjang karakter *pattern*.

- e. Setelah mendapatkan nilai *value*, langkah terakhir membuat tabel *bad-match* yang berisi *pattern* dan nilai pergeserannya pada tiap *window* dapat dilihat pada

Tabel 3.3.

Tabel 3.3 Tabel *Bad-Match*

<i>Letter</i>	B	U	K	A		J	E	N	D	L	*
<i>Value</i>	11	10	9	8	7	6	2	4	3	1	12

Berdasarkan Tabel 3.3 apabila ada karakter yang sama maka hanya salah satu karakter yang dimasukkan dengan aturan karakter yang paling terakhir nilainya dimasukkan. Contoh pada karakter E awalnya bernilai 5 berubah menjadi 2.

3.1.4 Pencocokan Algoritma *Horspool*

Setelah mendapatkan *value* masing-masing karakter *pattern*, langkah selanjutnya:

- Membandingkan *pattern* dengan karakter pada teks paling kanan yang sesuai dengan panjang *pattern*.
- Apabila terjadi ketidakcocokan antara teks dengan *pattern* maka akan bergeser sesuai dengan karakter teks paling kanan dari nilai karakter *pattern* yang telah dihitung sebelumnya pada proses *preprocessing*.

Berikut implementasinya menggunakan contoh kasus teks “buka jendela”

D	I	A		B	U	K	A		J	E	N	D	E	L	A
B	U	K	A		J	E	N	D	E	L	A				

Pada perulangan 1 berhenti di karakter N yang dimana memiliki nilai pergeseran 4.

D	I	A		B	U	K	A		J	E	N	D	E	L	A
				B	U	K	A		J	E	N	D	E	L	A

Pada perulangan 2 sudah cocok atau sesuai dengan *pattern*.

Gambar 3.3 Contoh Implementasi Algoritma *Horspool*

Pada Gambar 3.3 mengimplementasikan kalimat “dia buka jendela” dan ditemukan pola “buka jendela” pada iterasi atau perulangan kedua.

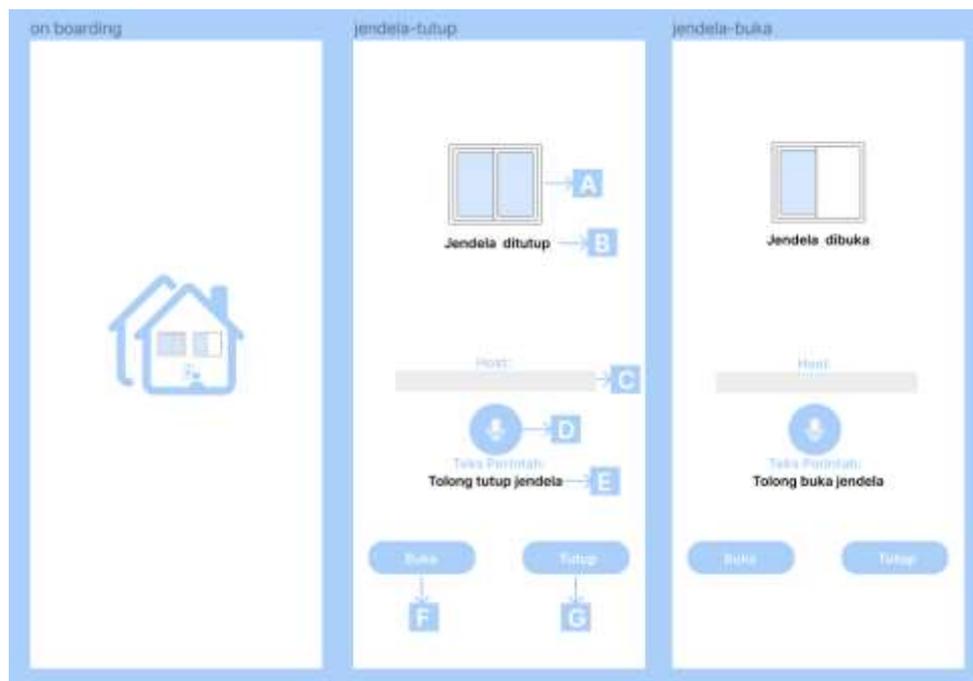
Untuk proses dalam pemrograman php berikut contoh pengujian algoritma *horspool* menggunakan suara

Pola ditemukan pada iterasi ke-2
 Teks sebelum stemming: Tolong dibukakan jendela
 Teks setelah stemming: tolong buka jendela

Gambar 3.4 Contoh Pengujian menggunakan Algoritma *Horspool*

Pada gambar 3.4 merupakan output dari *code* algoritma *horspool* dengan *pattern* buka jendela, teks awal tolong dibukakan jendela, dan setelah *stemming* tolong buka jendela sehingga didapatkan pola buka jendela pada iterasi kedua.

3.2 Perancangan Aplikasi Berbasis *Mobile*



Gambar 3.5 Rancangan Aplikasi Berbasis *Mobile*

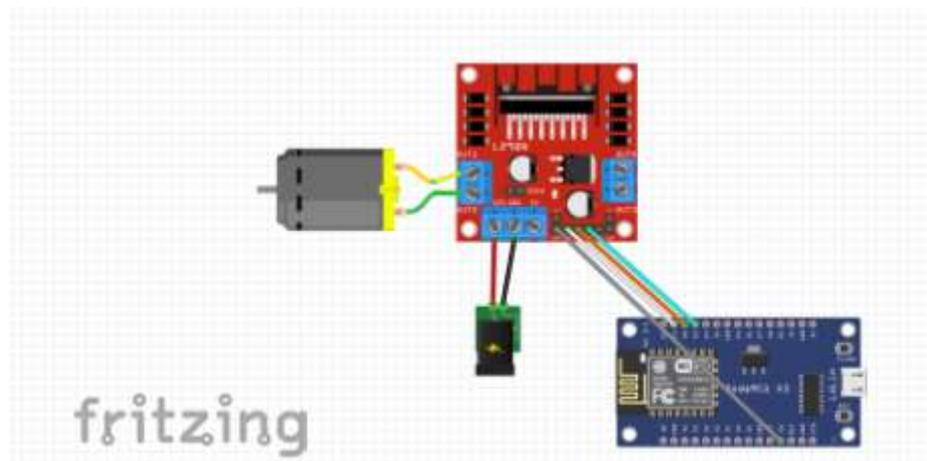
Keterangan:

- A = Output jendela terbuka atau tertutup
- B = Keterangan tulisan kondisi jendela
- C = Host untuk *url* ngrok
- D = Penggunaan untuk sistem kendali suara

E = Hasil suara diubah menjadi teks
F = Tombol untuk membuka jendela
G = Tombol untuk menutup jendela

3.3 Rangkaian Komponen

Rangkaian komponen yang digunakan dalam penelitian dapat dilihat pada gambar 3.6. Nama komponen tersebut adalah motor dc, adaptor 12v, Motor Driver L298N, dan NodeMCU ESP8266.



Gambar 3.6 Rancangan Komponen

Motor DC digunakan untuk penggerak jendela, Motor Driver L298N digunakan sebagai pengendali untuk motor dc seperti mengatur seberapa cepat gerakan motor dc, NodeMCU ESP8266 digunakan sebagai otak yang mengatur motor untuk melakukan sesuatu seperti membuka dan menutup jendela, adaptor sebagai pemberi tenaga pada motor. Berikut gambar rancangan komponen yang dibuat di *fritzing*.

3.4 Pengujian Sistem

Fungsi adanya tahapan pengujian sistem adalah untuk mengukur sistem apakah sudah tepat dan dapat digunakan. Hal yang diukur adalah algoritma pada penelitian ini menggunakan *confusion matrix*. *Confusion matrix* merupakan sebuah tabel perhitungan tingkat akurasi pada sebuah data yang kemudian berisi informasi data uji benar dan data uji salah.

Sebelum masuk ke perhitungan *confusion matrix* terlebih dahulu mencari data *ground truth* atau data referensi kebenaran. Ground truth pada sistem ini adalah pola kata yang dianggap benar yang harus ditemukan dalam teks pada proses pencocokan *pattern*. Berikut data *ground truth* yang digunakan:

- Buka Jendela
- Jendela Buka
- Tutup Jendela
- Jendela Tutup
- Jangan Buka Jendela
- Jangan Tutup Jendela
- Jendela Jangan Buka
- Jendela Jangan Tutup

Setelah memiliki data *ground truth*, masuk ke perhitungan *confusion matrix* untuk mengetahui tingkat akurasi algoritma *horspool* menggunakan rumus akurasi.

Tabel 3.4 Tabel *Confusion Matrix*

		Prediksi	
		Merespon Aktif	Tidak Bereaksi
Kondisi <i>Real</i>	Merespon Aktif	32	3
	Tidak Bereaksi	0	4

Keterangan:

- a. TP (*True Positive*) ialah input perintah benar dan algoritma mengatakan perintah tersebut benar
- b. TN (*True Negative*) ialah input perintah salah dan algoritma mengatakan perintah tersebut salah
- c. FP (*False Positive*) ialah input perintah salah dan algoritma mengatakan perintah tersebut benar
- d. FN (*False Negative*) ialah input perintah benar dan algoritma mengatakan perintah tersebut salah

Kemudian, untuk rumus menghitung akurasi dapat dilihat pada rumus (3.2).

$$accuracy = \frac{TP + TN}{TP + TF + FP + FN} \quad (3.2)$$

BAB IV

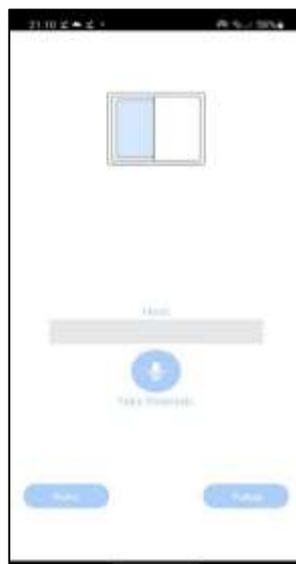
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada tahap implementasi sistem adalah hasil aplikasi dari perancangan yang telah dibuat sebelumnya. Kemudian, melalui aplikasi tersebut dijadikan sebagai sistem uji yang siap digunakan. Untuk server aplikasi agar terhubung dengan *broker MQTT* menggunakan platform *ngrok* yang dapat diakses secara gratis.

4.1.1 Halaman Aplikasi

Tampilan awal halaman aplikasi dapat dilihat pada gambar 4.1 yang menampilkan beberapa komponen gambar dan *text* untuk menampilkan status jendela. Kemudian, terdapat *button microphone*, *button* buka, dan *button* tutup yang berfungsi untuk mengontrol jendela. Selain itu, terdapat komponen *input text* untuk memasukkan *url ngrok* agar dapat mengakses server yang menyimpan logika *horspool* dan pengiriman pesan ke MQTT tanpa perlu *hosting*.



Gambar 4.1 Tampilan Utama



Gambar 4.2 Tampilan Penggunaan Speech Recognition

Pada saat *button mic* di *click* maka akan menampilkan fitur *speech* dari *google* yang akan mengubah kalimat yang diucapkan menjadi teks dapat dilihat pada Gambar 4.2. Setelah menjadi teks maka akan diproses oleh algoritma *horspool* untuk mencocokkan kalimat dengan *pattern* yang sudah ditentukan.

Pada tampilan Gambar 4.3 adalah saat algoritma *horspool* telah menemukan *pattern* pada kalimat yang telah diucapkan dan akan muncul hasil output, semisal “Jendela dibuka”. Kemudian akan mengirim pesan dari teks output ke MQTT untuk diproses pada *hardware* jendela.



Gambar 4.3 Halaman Tutup Jendela

Pada Gambar 4.3 adalah halaman yang berfungsi untuk tampilan menutup jendela menggunakan fitur *mic* ataupun *button*.

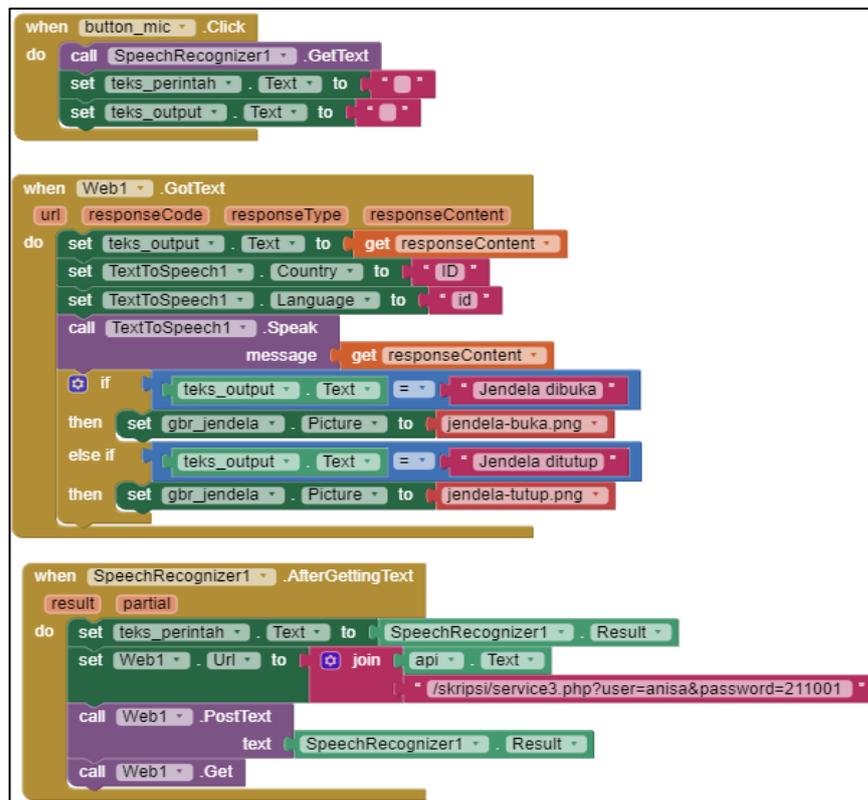
4.2 Pengujian Sistem

Pengujian sistem dilakukan menggunakan rumus akurasi pada *confussion matrix*. Tujuan dilakukan perhitungan akurasi untuk mengetahui apakah algoritma *horspool* dapat berfungsi dengan baik dan akurat saat diimplementasikan pada sistem kendali jendela dalam mendeteksi kalimat yang diperintahkan pengguna atau *user* berdasarkan *pattern* yang sudah ditentukan.

4.2.1 Kendali menggunakan *Microphone*

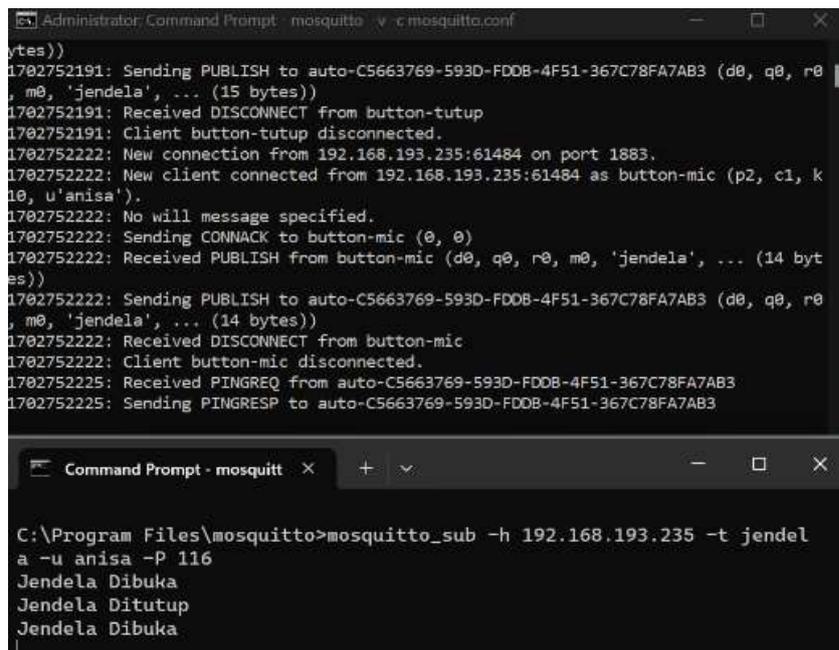
Perancangan kendali *microphone* menggunakan aplikasi MIT App Inventor dimana terdiri beberapa fungsi, diantaranya *button_mic*, *Web1*,

dan *SpeechRecognizer1*. Berikut Gambar 4.4 untuk proses kendali microphone pada MIT App Inventor.



Gambar 4.4 Logika *Button Mic*

Urutan logika diawali dengan klik *button mic* kemudian akan memanggil fitur *google speech* untuk mendeteksi kalimat yang diucapkan. Setelah itu, memanggil fungsi *text to speech* dari MIT App Inventor untuk merubah kalimat yang diucapkan menjadi teks. Setelah muncul teks pada keterangan *teks_perintah* akan dicari *pattern* dari *code horspool* melalui *url* API pada teks untuk output pada jendela apakah perintah tersebut sesuai atau tidak. Penjelasan lengkapnya dapat dilihat melalui Gambar 4.4.



```

Administrator: Command Prompt  mosquitto v c:mosquitto.conf
ytes))
1702752191: Sending PUBLISH to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3 (d0, q0, r0, m0, 'jendela', ... (15 bytes))
1702752191: Received DISCONNECT from button-tutup
1702752191: Client button-tutup disconnected.
1702752222: New connection from 192.168.193.235:61484 on port 1883.
1702752222: New client connected from 192.168.193.235:61484 as button-mic (p2, c1, k10, u'anisa').
1702752222: No will message specified.
1702752222: Sending CONNACK to button-mic (0, 0)
1702752222: Received PUBLISH from button-mic (d0, q0, r0, m0, 'jendela', ... (14 bytes))
1702752222: Sending PUBLISH to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3 (d0, q0, r0, m0, 'jendela', ... (14 bytes))
1702752222: Received DISCONNECT from button-mic
1702752222: Client button-mic disconnected.
1702752225: Received PINGREQ from auto-C5663769-593D-FDDB-4F51-367C78FA7AB3
1702752225: Sending PINGRESP to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3

Command Prompt - mosquitto x + v
C:\Program Files\mosquitto>mosquitto_sub -h 192.168.193.235 -t jendela -u anisa -P 116
Jendela Dibuka
Jendela Ditutup
Jendela Dibuka

```

Gambar 4.5 Output dari Logika Button Mic

Output Gambar 4.5 berisi informasi bahwa *button mic* dengan *id client button-mic* berhasil *publish* pesan “Jendela Ditutup” ke *subscribe* dengan topik yang sesuai. Tidak lupa mengisi *username* dan *password* yang valid agar hanya pengguna yang berhak dapat menggunakan atau memakai *topic* jendela.

4.2.2 Kendali menggunakan *Button*

Untuk memproses *button* buka atau tutup memerlukan logika yang dirancang pada halaman *block* MIT App Inventor. Sebelum klik *button*, terlebih dahulu mengisi komponen *input* yang *28able2828* “api” dengan *url publish ngrok*. Kemudian, *button* tersebut dapat diklik dan akan diproses ke *code* buka.php atau tutup.php yang memiliki *library* PHPMQTT yang berfungsi untuk menghubungkan ke MQTT.



Gambar 4.6 Button Buka

Pada Gambar 4.6 adalah logika *button* buka menggunakan fungsi *click* yang dimana saat *button* diklik akan diarahkan ke API server laptop pada *code* buka.php untuk membuka jendela.

```

Administrator: Command Prompt - mosquitto -> -c mosquitto.conf
DB-4F51-367C78FA7AB3 (p2, c1, k60, u'anisa').
1702752164: No will message specified.
1702752164: Sending CONNACK to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3 (0, 0)
1702752164: Received SUBSCRIBE from auto-C5663769-593D-FDDB-4F51-367C78FA7AB3
1702752164:      jendela (QoS 0)
1702752164: auto-C5663769-593D-FDDB-4F51-367C78FA7AB3 0 jendela
1702752164: Sending SUBACK to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3
1702752167: New connection from 192.168.193.235:61479 on port 1883.
1702752167: New client connected from 192.168.193.235:61479 as button-buka (p2, c1,
k10, u'anisa').
1702752167: No will message specified.
1702752167: Sending CONNACK to button-buka (0, 0)
1702752167: Received PUBLISH from button-buka (d0, q0, r0, m0, 'jendela', ... (14 by
tes))
1702752167: Sending PUBLISH to auto-C5663769-593D-FDDB-4F51-367C78FA7AB3 (d0, q0, r0
, m0, 'jendela', ... (14 bytes))
1702752167: Received DISCONNECT from button-buka
1702752167: Client button-buka disconnected.

C:\Program Files\mosquitto>mosquitto_sub -h 192.168.193.235 -t jendel
a -u anisa -P 116
Jendela Dibuka

```

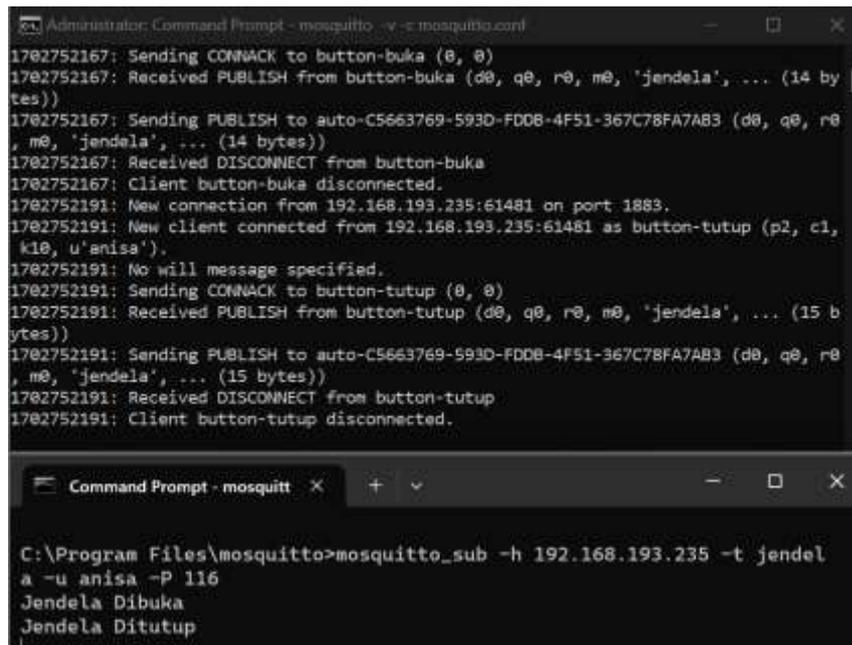
Gambar 4.7 Output Terminal dari Logika Button Tutup

Output Gambar 4.7 berisi informasi bahwa *button mic* dengan *id client* *button-mic* berhasil *publish* pesan “Jendela Ditutup” ke *subscribe* dengan topik yang sesuai. Tidak lupa mengisi *username* dan *password* yang valid agar hanya pengguna yang berhak dapat menggunakan atau memakai *topic* jendela.



Gambar 4.8 Button Tutup

Pada logika Gambar 4.8 menggunakan fungsi *click* pada mit app inventor yang dimana saat *button* diklik akan diarahkan 30able30 pada server laptop code tutup.php untuk menutup jendela. Kemudian memanggil fungsi Web1 untuk menampilkan status output pada aplikasi



```

Administrator: Command Prompt - mosquitto - v - c mosquitto.com
1702752167: Sending CONNACK to button-buka (0, 0)
1702752167: Received PUBLISH from button-buka (d0, q0, r0, m0, 'jendela', ... (14 bytes))
1702752167: Sending PUBLISH to auto-C5663769-593D-FDDB-4F51-367C78FA7A83 (d0, q0, r0, m0, 'jendela', ... (14 bytes))
1702752167: Received DISCONNECT from button-buka
1702752167: Client button-buka disconnected.
1702752191: New connection from 192.168.193.235:61481 on port 1883.
1702752191: New client connected from 192.168.193.235:61481 as button-tutup (p2, c1, k10, u'anisa').
1702752191: No will message specified.
1702752191: Sending CONNACK to button-tutup (0, 0)
1702752191: Received PUBLISH from button-tutup (d0, q0, r0, m0, 'jendela', ... (15 bytes))
1702752191: Sending PUBLISH to auto-C5663769-593D-FDDB-4F51-367C78FA7A83 (d0, q0, r0, m0, 'jendela', ... (15 bytes))
1702752191: Received DISCONNECT from button-tutup
1702752191: Client button-tutup disconnected.

Command Prompt - mosquitto x + v - - □ x
C:\Program Files\mosquitto>mosquitto_sub -h 192.168.193.235 -t jendela -u anisa -P 116
Jendela Dibuka
Jendela Ditutup

```

Gambar 4.9 Output Terminal dari Logika Button Buka

Pada Gambar 4.9 adalah proses dikliknya *button* maka akan bertindak sebagai *publish* untuk mengirim pesan yang sesuai dengan keterangan *button* ke *subscribe*. Kemudian, pesan yang tampil pada *subscribe* akan di *publish* kembali ke esp8266 untuk memproses kendali jendela.

4.2.3 Proses *Table Bad Match*

Pada proses *table bad match* berfungsi untuk mencari nilai dari setiap karakter pada *pattern*. Fungsinya agar dapat diketahui berapa pergeseran ke kanan sesuai nilai karakter *pattern* sehingga dapat ditemukan pola pada suatu teks atau *message*. Untuk prosesnya dimulai dari menentukan *index* pada *pattern* dan

menghitung panjang *pattern*. Kemudian, dihitung nilai karakter *bad match* menggunakan rumus yang telah dipaparkan pada bab 3. Berikut hasil *table bad match* yang dapat dilihat pada Gambar 4.10.

Pattern: buka jendela Panjang Pattern: 12 Tabel Pergeseran Bad Character: b: 11 u: 10 k: 9 a: 8 : 7 j: 6 e: 2 n: 4 d: 3 l: 1 Pattern: jendela buka Panjang Pattern: 12 Tabel Pergeseran Bad Character: j: 11 e: 7 n: 9 d: 8 l: 6 a: 5 : 4 b: 3 u: 2 k: 1 Pattern: tutup jendela Panjang Pattern: 13 Tabel Pergeseran Bad Character: t: 10 u: 9 p: 8 : 7 j: 6 e: 2 n: 4 d: 3 l: 1	Pattern: jangan buka jendela Panjang Pattern: 19 Tabel Pergeseran Bad Character: j: 6 a: 8 n: 4 g: 15 : 7 b: 11 u: 10 k: 9 e: 2 d: 3 l: 1 Pattern: jangan jendela buka Panjang Pattern: 19 Tabel Pergeseran Bad Character: j: 11 a: 5 n: 9 g: 15 : 4 e: 7 d: 8 l: 6 b: 3 u: 2 k: 1	Pattern: jangan tutup jendela Panjang Pattern: 20 Tabel Pergeseran Bad Character: j: 6 a: 15 n: 4 g: 16 : 7 t: 10 u: 9 p: 8 e: 2 d: 3 l: 1 Pattern: jangan jendela tutup Panjang Pattern: 20 Tabel Pergeseran Bad Character: j: 12 a: 6 n: 10 g: 16 : 5 e: 8 d: 9 l: 7 t: 2 u: 1
---	---	--

Gambar 4.10 Output Tabel Bad Match

Gambar tersebut menampilkan percobaan semua *pattern* dengan penjelasan panjang dari tiap *pattern* dengan perhitungan pergeseran tiap karakter pada *pattern*.

4.2.4 Pengujian Algoritma Horspool

Algoritma *Horspool* pada penelitian ini diimplementasikan pada pengujian kalimat perintah yang diantaranya terdapat kata yang akan dicari sesuai pola *pattern* untuk dicocokkan. Pada pengujiannya peneliti menggunakan 34 kalimat perintah

dengan 8 *pattern*. Berikut beberapa hasil pengujian yang dimulai dari kalimat “buka jendela” pada Tabel 4.1.

Tabel 4.1 Hasil pengujian kalimat “buka jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	buka jendela	buka jendela	1	3,3991367
Perintah 2	buka jendela	buka jendela	1	3,3845521
Perintah 3	buka jendela	buka jendela	1	3,3063913
Perintah 4	buka jendela	buka jendela	1	3,4051865
Perintah 5	buka jendela	buka jendela	1	3,3899512
Perintah 6	buka jendela	buka jendela	1	3,3456323
Perintah 7	buka jendela	buka jendela	1	3,3502624
Perintah 8	buka jendela	buka jendela	1	3,3674219
Perintah 9	buka jendela	buka jendela	1	3,3768361
Perintah 10	buka jendela	buka jendela	1	3,3812163
Rata-rata				3,37065868

Setelah percobaan dilakukan, berikut hasil grafik waktu dari pengujian kalimat “buka jendela” dengan *pattern* buka jendela dapat dilihat pada Gambar 4.11 di bawah ini.



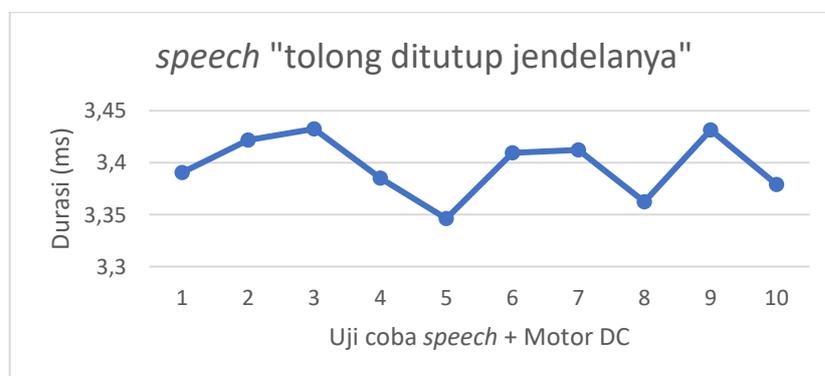
Gambar 4.11 Grafik “buka jendela”

Berdasarkan percobaan kalimat “buka jendela” sebanyak 10 kali hasilnya sesuai dengan perintah, yakni buka jendela. Kemudian, rata-rata waktu yang dihasilkan dari percobaan tersebut adalah 3,37065868 ms. Percobaan selanjutnya pada Tabel 4.2 dengan kalimat “tolong ditutup jendelanya”.

Tabel 4.2 Hasil pengujian kalimat “tolong ditutup jendelanya”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong ditutup jendelanya	tutup jendela	2	3,3906142
Perintah 2	tolong ditutup jendelanya	tutup jendela	2	3,4217231
Perintah 3	tolong ditutup jendelanya	tutup jendela	2	3,4325415
Perintah 4	tolong ditutup jendelanya	tutup jendela	2	3,3852614
Perintah 5	tolong ditutup jendelanya	tutup jendela	2	3,3462435
Perintah 6	tolong ditutup jendelanya	tutup jendela	2	3,4095303
Perintah 7	tolong ditutup jendelanya	tutup jendela	2	3,4124351
Perintah 8	tolong ditutup jendelanya	tutup jendela	2	3,3625142
Perintah 9	tolong ditutup jendelanya	tutup jendela	2	3,4314233
Perintah 10	tolong ditutup jendelanya	tutup jendela	2	3,3789311
Rata-rata				3,39712177

Setelah percobaan dilakukan, berikut hasil grafik waktu dari pengujian kalimat “tolong ditutup jendelanya” dengan *pattern* tutup jendela dapat dilihat pada Gambar 4.12 di bawah ini.



Gambar 4.12 Grafik “tolong ditutup jendelanya”

Berdasarkan percobaan kalimat “tolong ditutup jendelanya” sebanyak 10 kali hasilnya sesuai dengan perintah, yakni tutup jendela. Kemudian, rata-rata waktu yang dihasilkan dari percobaan tersebut adalah 3,39712177 ms. Percobaan selanjutnya pada Tabel 4.3 dengan kalimat “mohon untuk jendela dibuka”.

Tabel 4.3 Hasil pengujian kalimat “mohon untuk jendela dibukakan”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon untuk jendela dibukakan	jendela buka	3	3,4015368
Perintah 2	mohon untuk jendela dibukakan	jendela buka	3	3,3700351
Perintah 3	mohon untuk jendela dibukakan	jendela buka	3	3,4420241
Perintah 4	mohon untuk jendela dibukakan	jendela buka	3	3,3771739
Perintah 5	mohon untuk jendela dibukakan	jendela buka	3	3,3742653
Perintah 6	mohon untuk jendela dibukakan	jendela buka	3	3,4542159
Perintah 7	mohon untuk jendela dibukakan	jendela buka	3	3,3852415
Perintah 8	mohon untuk jendela dibukakan	jendela buka	3	3,4349162
Perintah 9	mohon untuk jendela dibukakan	jendela buka	3	3,3435104
Perintah 10	mohon untuk jendela dibukakan	jendela buka	3	3,3677448
Rata-rata				3,3950664

Setelah percobaan dilakukan, berikut hasil grafik waktu dari pengujian kalimat “mohon untuk jendela dibukakan” dengan *pattern* jendela buka dapat dilihat pada Gambar 4.13.



Gambar 4.13 Grafik “mohon untuk jendela dibukakan”

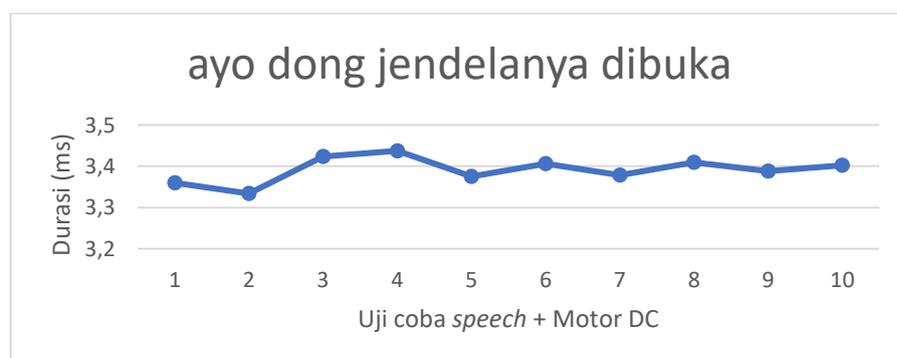
Berdasarkan percobaan kalimat “mohon untuk jendela dibukakan” sebanyak 10 kali hasilnya sesuai dengan perintah, yakni jendela buka. Kemudian, rata-rata waktu yang dihasilkan dari percobaan tersebut adalah 3,3950664 ms.

Percobaan selanjutnya pada Tabel 4.4 dengan kalimat “ayo dong jangan dibuka jendelanya”.

Tabel 4.4 Hasil pengujian kalimat “ayo dong jangan dibuka jendelanya”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,3594951
Perintah 2	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,3338224
Perintah 3	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,4236733
Perintah 4	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,4372948
Perintah 5	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,3750798
Perintah 6	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,4058231
Perintah 7	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,3782687
Perintah 8	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,4093331
Perintah 9	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,3877243
Perintah 10	ayo dong jangan dibuka jendelanya	jangan buka jendela	2	3,4019876
Rata-rata				3,39125022

Setelah percobaan dilakukan, berikut hasil grafik waktu dari pengujian kalimat “ayo dong jangan dibuka jendelanya” dengan *pattern* jangan buka jendela dapat dilihat pada Gambar 4.14.



Gambar 4.14 Grafik “ayo dong jangan dibuka jendelanya”.

Berdasarkan percobaan kalimat “mohon untuk jendela dibukakan” sebanyak 10 kali hasilnya sesuai dengan perintah, yakni jangan buka jendela atau tutup jendela. Kemudian, rata-rata waktu yang dihasilkan dari percobaan tersebut adalah 3,39125022 ms. Untuk hasil pengujian selanjutnya pada lampiran yang dicantumkan di bagian akhir penulisan.

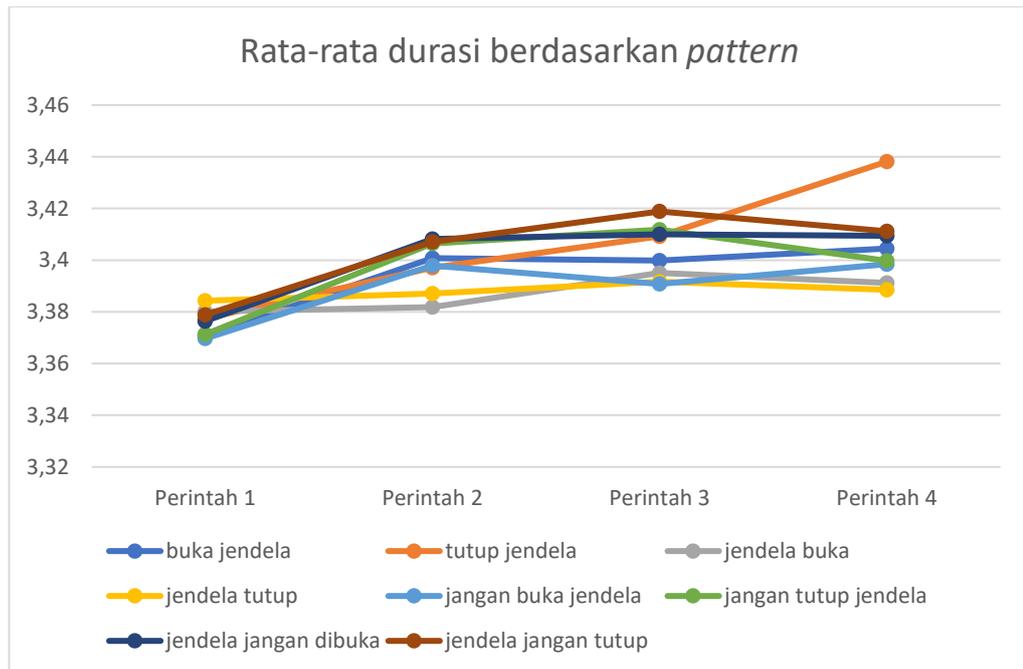
Setelah beberapa pengujian telah dilakukan, langkah terakhir menghitung rata-rata durasi *pattern* berdasarkan beberapa perintah. Hal tersebut dilakukan agar dapat diketahui berapa lama eksekusi rata-rata waktu tiap *pattern* dan perintah *pattern* mana yang eksekusi waktunya paling lama dan paling cepat. Berikut hasil grafik rata-rata durasi *pattern*.

Tabel 4.5 Rata-rata Durasi *Pattern*

	Rata-rata Durasi <i>Pattern</i>							
	buka jendela	tutup jendela	jendela buka	jendela tutup	jangan buka jendela	jangan tutup jendela	jendela jangan dibuka	jendela jangan tutup
Perintah 1	3,37066	3,377419	3,380092	3,384308	3,369667	3,371333	3,376411	3,378846
Perintah 2	3,40081	3,397122	3,381798	3,387083	3,397918	3,406415	3,408218	3,407074
Perintah 3	3,39988	3,409276	3,395066	3,39167	3,390776	3,411807	3,409997	3,418854
Perintah 4	3,4045	3,438134	3,39125	3,388555	3,398446	3,399757	3,409397	3,411177
Rata-rata	3,39396	3,405488	3,387052	3,387904	3,389202	3,397328	3,401006	3,403988

Berdasarkan pengujian dari 36 kalimat dapat dilihat dalam bentuk grafik berdasarkan Gambar 4.15 yang menampilkan grafik durasi dari *pattern* yang terdiri dari 32 kalimat. Durasi terlama pada *pattern* “tutup jendela” dengan kalimat “saya ingin sekarang menutup jendela” yang diketahui penemuan pola *pattern* “tutup jendela” pada iterasi ke 5 sehingga hasil durasinya cukup tinggi, yaitu

3,438133991 ms. Kemudian, kalimat yang durasinya paling rendah pada pattern “jangan buka jendela” dengan kalimat “jangan buka jendela” yang ditemukan pola pattern pada iterasi ke-1 sehingga durasi dalam menemukan pola tersebut sangat cepat, yaitu 3,36966726 ms.



Gambar 4.15 Rata-rata durasi berdasarkan *Pattern*

4.2.5 Pengujian Algoritma Horspool Menggunakan *Confusion Matrix*

Setelah dilakukannya pengujian kalimat perintah terhadap *pattern* menggunakan algoritma *horspool*, langkah selanjutnya adalah menghitung tingkat akurasi algoritma *horspool* menggunakan *confusion matrix*. Pada implementasi perhitungan *confusion matrix* lebih berfokus pada rumus akurasi yang telah dipaparkan pada bab 3. Kemudian yang dihitung adalah berdasarkan 8 *pattern*, diantaranya buka jendela, tutup jendela, jendela buka, jendela tutup, jangan buka jendela, jangan tutup jendela, jendela jangan dibuka, dan jendela jangan tutup.

Tabel 4.6 *Pattern* buka jendela

No	Teks/Kalimat Perintah	Pattern	Prediksi		Kondisi Real		Ket.
			Merespon Aktif	Tidak Bereaksi	Merespon Aktif	Tidak Bereaksi	
1	buka jendela	buka jendela	✓		✓		TP
2	tolong dibuka jendelanya	buka jendela	✓		✓		TP
3	mohon sekarang membuka jendela	buka jendela	✓		✓		TP
4	ayo dong jendelanya dibuka	buka jendela	✓		✓		TP
5	tolong buka segera jendela itu	buka jendela	✓			✓	FN
6	tutup jendela	tutup jendela	✓		✓		TP
7	tolong ditutup jendelanya	tutup jendela	✓		✓		TP
8	mohon sekarang menutup jendela	tutup jendela	✓		✓		TP
9	ayo dong ditutup jendelanya	tutup jendela	✓		✓		TP
10	Saya minta tutup itu jendela	tutup jendela	✓			✓	FN
11	jendela buka	jendela buka	✓		✓		TP
12	mohon untuk jendela dibukakan	jendela buka	✓		✓		TP
13	ayo dong jendelanya dibuka	jendela buka	✓		✓		TP
14	tolong jendela dibukakan	jendela buka	✓		✓		TP
15	jendela mohon dibuka	jendela buka	✓			✓	FN
16	jendela tutup	jendela tutup	✓		✓		TP
17	mohon untuk jendela ditutupkan	jendela tutup	✓		✓		TP
18	ayo dong jendelanya ditutup	jendela tutup	✓		✓		TP
19	tolong jendela ditutup	jendela tutup	✓		✓		TP
20	jangan buka jendela	jangan buka jendela	✓		✓		TP

No	Teks/Kalimat Perintah	Pattern	Prediksi		Kondisi Real		Ket.
			Merespon Aktif	Tidak Bereaksi	Merespon Aktif	Tidak Bereaksi	
21	tolong jangan membuka jendela	jangan buka jendela	✓		✓		TP
22	saya ingin jangan membukakan jendela	jangan buka jendela	✓		✓		TP
23	ayo dong jangan dibuka jendelanya	jangan buka jendela	✓		✓		TP
24	jangan tutup jendela	jangan tutup jendela	✓		✓		TP
25	tolong jangan membuka jendela	jangan tutup jendela	✓		✓		TP
26	saya ingin jangan menutup jendela	jangan tutup jendela	✓		✓		TP
27	ayo dong jangan ditutup jendelanya	jangan tutup jendela	✓		✓		TP
28	jendela jangan dibuka	jendela jangan buka	✓		✓		TP
29	mohon jendela jangan dibuka	jendela jangan buka	✓		✓		TP
30	saya mau jendelanya jangan dibuka	jendela jangan buka	✓		✓		TP
31	tolong dong jendelanya jangan membuka	jendela jangan buka	✓		✓		TP
32	jendela jangan ditutup	jendela jangan tutup	✓		✓		TP
33	mohon jendela jangan ditutup	jendela jangan tutup	✓		✓		TP
34	saya mau jendelanya jangan ditutup	jendela jangan tutup	✓		✓		TP
35	tolong dong jendelanya jangan menutup	jendela jangan tutup	✓		✓		TP
36	Tolong			✓		✓	TN
37	Mohon buka pintu			✓		✓	TN
38	Segera tutup			✓		✓	TN
39	Ayo			✓		✓	TN

Pada Tabel 4.6 Pengujian dari *pattern* menghasilkan 32 merespon aktif pada prediksi dengan kondisi real disebut *true positive* (TP) dan 3 prediksi merespon aktif tetapi kondisi *real* tidak bereaksi disebut *False Negative* (FN) dikarenakan diantara pola terdapat kata, seperti contoh pada salah satu kalimat “jendela mohon dibuka” dengan *pattern* “jendela buka” diantara kalimat terdapat kata “mohon” sehingga pola terpisah dan tidak terdeteksi.

Tabel 4.7 Hasil *Confusion Matrix*

		Prediksi	
		Merespon Aktif	Tidak Bereaksi
Kondisi Real	Merespon Aktif	32	3
	Tidak Bereaksi	0	4

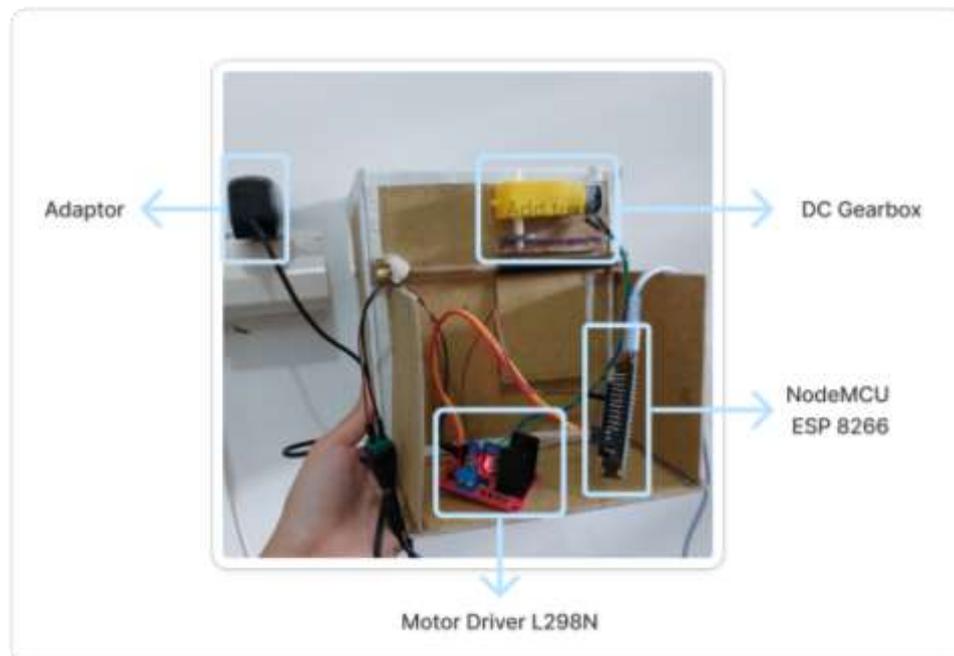
Setelah didapatkan hasil *confusion matrix*, langkah selanjutnya menghitung akurasi menggunakan rumus $\frac{TP+TN}{TP+TF+FP+FN} \times 100\%$

$$accuracy = \frac{32 + 4}{32 + 4 + 0 + 3} \times 100\% = 0,92$$

4.3 Rangkaian Hardware

Rangkaian pada penelitian ini menggunakan NodeMCU ESP8266 sebagai mikrokontroler untuk menjalankan Motor DC Gearbox yang dimana perintahnya dikirimkan terlebih dahulu ke Motor Driver L298N. Pada saat *user* menggunakan *button mic*, hasil suara akan *diconvert* menjadi teks. Teks tersebut akan dikirim ke server untuk diketahui apakah sudah sesuai *pattern*. Apabila sudah sesuai maka akan dikirim perintahnya ke NodeMCU ESP8266 menggunakan MQTT dan mikrokontroler tersebut akan menjalankan perintah dari MQTT ke Motor Driver L298N. Output dapat dilihat dari Motor DC Gearbox sesuai perintah yang tampil

pada pesan MQTT. Berikut gambar rangkaian komponennya berdasarkan implementasi dari aplikasi *fritzing*.



Gambar 4.16 Rangkaian Komponen

DC Gearbox diletakkan di atas jendela berfungsi sebagai roda untuk menggerakkan maju mundur jendela. Kemudian, adaptor yang terletak pada colokan listrik sebelah kiri yang berfungsi sebagai daya untuk menggerakkan jendela. Selanjutnya, Motor Driver L298N terletak ditengah untuk mengendalikan pergerakan Motor DC Gearbox. Terakhir, NodeMCU ESP8266 yang diletakkan di bawah sebelah kanan berfungsi sebagai modul wifi yang menangkap pesan melalui MQTT, seperti “jendela dibuka” maka akan memproses Motor Driver L298N dan Motor DC Gearbox untuk membuka jendela.

4.4 Hasil Pengujian Alat

Untuk proses pengujian alat disini meliputi aplikasi kontrol jendela dan *hardware*.



Gambar 4.17 Memberi Perintah Membuka Jendela

Pada Gambar 4.17 merupakan tampilan jendela saat ingin dibuka dengan kalimat perintah “tolong dibuka jendelanya ya”.



Gambar 4.18 Output Miniatur Membuka Jendela

Pada Gambar 4.18 adalah tampilan jendela terbuka saat diberi perintah “tolong dibuka jendelanya ya” sehingga tampilan akhir jendela tersebut terbuka.



Gambar 4.19 Memberi Perintah Menutup Jendela

Pada Gambar 4.19 merupakan tampilan jendela saat ingin ditutup dengan kalimat perintah “tolong tutup jendelanya ya”



Gambar 4.20 Output Miniatur Tutup Jendela

Pada Gambar 4.20 adalah tampilan setelah kalimat perintah sudah sesuai dengan *pattern* tutup jendela maka miniatur jendela akan tertutup.

4.5 Integrasi Islam

Allah SWT memberikan manusia kemampuan berpikir atau akal kepada manusia yang diketahui dalam Surah Al-Baqarah ayat 31

وَعَلَّمَ آدَمَ الْأَسْمَاءَ كُلَّهَا ثُمَّ عَرَضَهُمْ عَلَى الْمَلَائِكَةِ فَقَالَ أَنْبِئُونِي بِأَسْمَاءِ هَؤُلَاءِ إِنْ كُنْتُمْ صَادِقِينَ

“Dan Dia mengajarkan kepada Adam nama-nama (benda-benda) seluruhnya, kemudian mengemukakannya kepada para Malaikat lalu berfirman: "Sebutkanlah kepada-Ku nama benda-benda itu jika kamu mamang benar orang-orang yang benar!". (QS. Al-Baqarah:31)

Pada surah Al-Baqarah ayat 31 terkandung keutamaan Adam atas malaikat berkat apa yang telah dikhususkan oleh Allah baginya berupa ilmu tentang nama-nama segala sesuatu, sedangkan para malaikat diperintahkan untuk bersujud kepada Adam.

Sesungguhnya bagian ini didahulukan atas bagian tersebut (yang mengandung perintah Allah kepada para malaikat untuk bersujud kepada Adam) karena bagian ini mempunyai ikatan erat dengan ketidaktahuan para malaikat tentang hikmah penciptaan khalifah, yaitu disaat mereka menanyakan hal tersebut. Kemudian Allah Swt memberitahukan bahwa Dia mengetahui apa yang tidak mereka ketahui. Karena itulah Allah menyebutkan bagian ini sesudah hal tersebut, untuk menjelaskan kepada mereka keutamaan Adam, berkat kelebihan yang dimilikinya diatas mereka berupa ilmu pengetahuan tentang nama-nama segala sesuatu. Untuk itu Allah Swt berfirman “Dan Dia mengajarkan kepada Adam nama-nama (benda-benda) seluruhnya” (Tafsir Ibnu Katsir).

Oleh karena itu, dengan kemampuan Nabi Adam AS dalam memberi nama sesuatu merupakan bukti bahwa manusia memiliki akal yang diciptakan oleh Allah

SWT. Seperti halnya pada sistem yang dibuat dalam penelitian ini, salah satunya mencari pola pada suatu kalimat menggunakan algoritma horspool dalam penggunaan *speech recognizer* yang memerlukan akal agar sistem berjalan sesuai yang diinginkan. Oleh karena itu, dengan diberinya akal dapat menciptakan sesuatu yang bermanfaat serta memudahkan manusia.

Sesuai dengan hadist riwayat muslim tentang memudahkan orang lain yang berbunyi.

وَمَنْ يَسِّرْ عَلَى مُعْسِرٍ يَسِّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ

“Barangsiapa memudahkan orang yang kesulitan Maka Allâh memudahkan baginya (dari kesulitan) di dunia dan akhirat.” (H.R Muslim)

Dengan sistem ini dapat memudahkan baik disabilitas maupun nondisabilitas sehingga proses kendali jendela tidak diperlukan secara manual, dengan kata lain dapat dilakukan secara otomatis. Seperti halnya, seseorang sedang berada di lantai 1, kemudian ingin membuka jendela di lantai 2. Melalui sistem ini orang tersebut hanya perlu menggunakan aplikasi sebagai kendali. Oleh karena itu, dengan memudahkan orang lain maka Allah juga akan memudahkan kita

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian kendali jendela menggunakan algoritma *horspool* berjalan dengan baik. Hasil akurasi algoritma ini adalah 92% yang dimana dilakukan percobaan 35 kalimat dengan 8 *pattern*. Pada 32 kalimat berhasil dan sesuai *pattern*, tetapi pada 3 kalimat tidak berhasil dikarenakan diantara kalimat polanya terpisah, contohnya “tolong jendela itu dibuka” diantara pola jendela buka terdapat kata “itu”. Untuk rata-rata durasi pengujian algoritma ini adalah 3,395741207 ms. Durasi terlama pada kalimat “saya ingin sekarang menutup jendela” bernilai 3,438133991 ms yang dimana *pattern* tutup jendela ditemukan pada iterasi ke 5. Sedangkan durasi tercepat pada kalimat “jangan dibuka jendela” yang bernilai 3,36966726 ms. Oleh karena itu, lama atau cepat durasi tergantung ditemukannya pola iterasi pada kalimat semakin banyak iterasi maka semakin lama durasi dan sebaliknya.

5.2 Saran

Sistem kendali jendela menggunakan algoritma *horspool* ini masih memiliki kekurangan sehingga diperlukan pengembangan kedepannya agar dapat menjadikan sistem yang lebih baik,.Saran dari peneliti adalah menambahkan bahasa lainnya, seperti bahasa inggris atau bahasa asing lainnya dalam memproses kalimat yang diucapkan

DAFTAR PUSTAKA

- Abidin, Z., Mustafa, M. S., & SY, H. (2023). Penerapan Horspool Algorithm pada Aplikasi Translate Indonesia ke Bahasa Daerah Berbasis Android: Studi Kasus 29 Bahasa. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 26–36. <https://doi.org/10.57152/malcom.v4i1.1000>
- Dinata, R. K., Safwandi, S., Hasdyna, N., & Mahendra, R. (2020). Kombinasi Algoritma Brute Force dan Stemming pada Sistem Pencarian Mashdar. *CESS (Journal of Computer Engineering, System and Science)*, 5(2), 273. <https://doi.org/10.24114/cess.v5i2.17989>
- Dongoran, K. A., Hasibuan, N. A., Hutabarat, S., Ginting, G., & Ramadan Siregar, S. (2021). Implementasi Algoritma Zhu-Takaoka Pada Data Repository Linux Berbasis Web. *Jurnal Riset Komputer*, 8(2), 2407–389. <https://doi.org/10.30865/jurikom.v7i6.2542>
- Ernawati, E., Johar, A., & Setiawan, S. (2019). Implementasi Metode String Matching Untuk Pencarian Berita Utama Pada Portal Berita Berbasis Android (Studi Kasus: Harian Rakyat Bengkulu). *Pseudocode*, 6(1), 77–82. <https://doi.org/10.33369/pseudocode.6.1.77-82>
- Filcha, A., & Hayaty, M. (2019). Implementasi Algoritma Rabin-Karp untuk Pendeteksi Plagiarisme pada Dokumen Tugas Mahasiswa. *JUITA : Jurnal Informatika*, 7(1), 25. <https://doi.org/10.30595/juita.v7i1.4063>
- Frigustini, V., Erlanshari, A., & Andreswari, D. (2018). Implementasi Algoritma Horspool Pada Aplikasi Kamus Bahasa Lintang - Indonesia Berbasis Android. *Rekursif: Jurnal Informatika*, 6(1). <https://doi.org/10.33369/rekursif.v6i1.3072>
- Fuadi, W., Risawandi, R., & Yanti, R. (2020). Aplikasi Doa Para Nabi Dan Rasul Dalam Al-Qur'an Menggunakan Algoritma Horspool Berbasis Android. *TECHSI - Jurnal Teknik Informatika*, 12(1), 1. <https://doi.org/10.29103/techsi.v12i1.1702>
- Hanani, A., & Haryadi, M. A. (2020). Smart Home Berbasis Iot Menggunakan Arduino Uno Dan Suara Pada Google Assistant. *Stmikasia*, 14, 1–8. <https://doi.org/https://doi.org/10.32815/jitika.v14i1.456>
- Kusnadi, A., & Wicaksono, A. K. (2017). Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop. *Jurnal ULTIMA Computing*, 9(1), 12–16. <https://doi.org/10.31937/sk.v9i1.568>
- Mesi, E., & Oktarina, D. (2021). Penerapan Algoritma Horspool Pada Sistem Pendataan Obat Pada Apotek Fajar Mas. *Seminar Nasional Informatika*, 79–85. <https://www.ejournal.pelitaindonesia.ac.id/ojs32/index.php/SENATIKA/art>

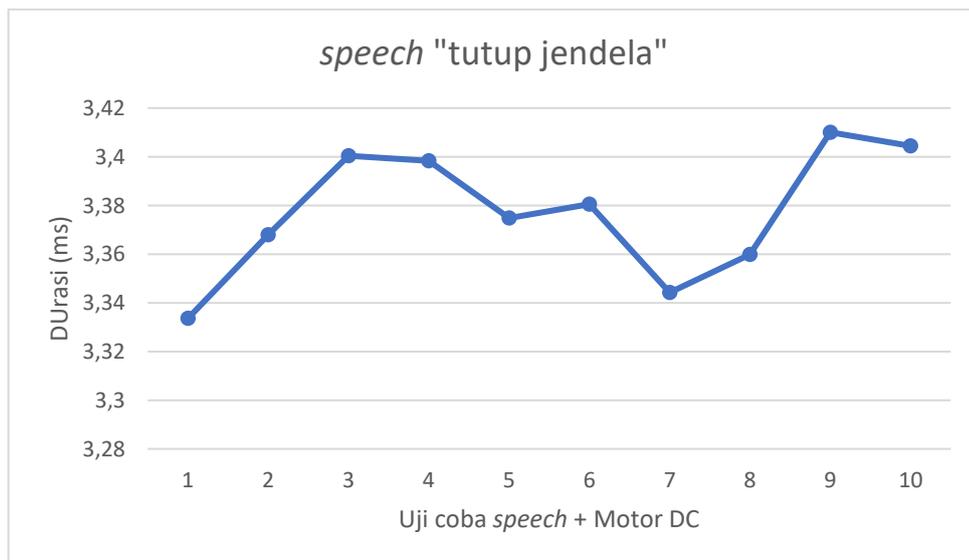
icle/view/1138

- Mohamad Salman Farizi, Somantri, S., & Yustiana, I. (2022). Implementasi Speech Recognition Pada Sistem Kendali Perangkat Elektronik Rumah Berbasis IoT (Internet Of Things) dan Mobile Application. *ZONAsi: Jurnal Sistem Informasi*, 4(2), 157–166. <https://doi.org/10.31849/zn.v4i2.10662>
- psti.unisayogya.ac.id. (2020). *Apa Itu MIT App Inventor*. <https://psti.unisayogya.ac.id/2020/01/06/apa-itu-mit-app-inventor-berikut-penjelasan/>
- Putri, R. E., & Yendri, D. (2018). Sistem Pengontrolan Dan Keamanan Rumah Pintar (Smart Home) Berbasis Android. *Journal on Information Technology and Computer Engineering*, 2(01), 1–6. <https://doi.org/10.25077/jitce.2.01.1-6.2018>
- Romoadhon, A. S., & Anamisa, D. R. (2017). Sistem Kontrol Peralatan Listrik pada Smart Home Menggunakan Android. *Rekayasa*, 10(2), 116. <https://doi.org/10.21107/rekayasa.v10i2.3613>
- Setiawan, R. (2021). Memahami Apa Itu Internet of Things. In *8 September 2021* (Vol. 1, Issue September). Tiga Ebook.
- Sunny, A. S. (2009). Speech Recognition Menggunakan Algoritma Program Dinamis. *Speech Recognition Menggunakan Algoritma Program Dinamis*, 4.
- Waruwu, F. T., & Hondro, R. K. (2019). Penerapan Algoritma Horspool pada Aplikasi Katalog Buku Perpustakaan. *Prosiding Seminar Nasional Riset Information Science (SENARIS)*, 1(September), 881. <https://doi.org/10.30645/senaris.v1i0.95>
- wearesocial.com. (2023). *Digital 2023 Indonesia The Essential To The Latest Connected Behaviours*. <https://wearesocial.com/id/blog/2023/01/digital-2023/>

LAMPIRAN

Lampiran 1. Pengujian Kalimat "tutup jendela"

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tutup jendela	tutup jendela	1	3,3335914
Perintah 2	tutup jendela	tutup jendela	1	3,3679225
Perintah 3	tutup jendela	tutup jendela	1	3,4004181
Perintah 4	tutup jendela	tutup jendela	1	3,3983313
Perintah 5	tutup jendela	tutup jendela	1	3,3748678
Perintah 6	tutup jendela	tutup jendela	1	3,3804931
Perintah 7	tutup jendela	tutup jendela	1	3,3441915
Perintah 8	tutup jendela	tutup jendela	1	3,3598737
Perintah 9	tutup jendela	tutup jendela	1	3,4100511
Perintah 10	tutup jendela	tutup jendela	1	3,4044541
rata-rata				3,37741946



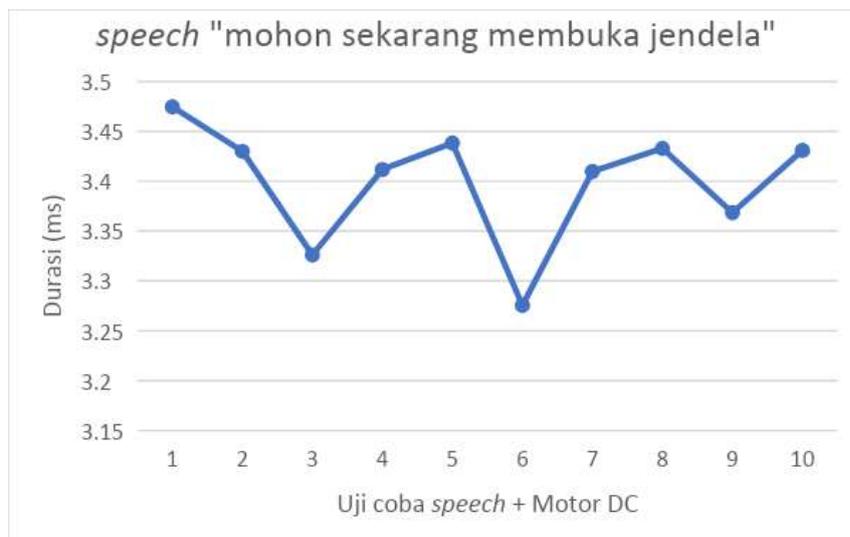
Lampiran 2. Pengujian Kalimat “tolong dibuka jendelanya”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong dibuka jendelanya	buka jendela	2	3,4294145
Perintah 2	tolong dibuka jendelanya	buka jendela	2	3,2956513
Perintah 3	tolong dibuka jendelanya	buka jendela	2	3,4341322
Perintah 4	tolong dibuka jendelanya	buka jendela	2	3,4221642
Perintah 5	tolong dibuka jendelanya	buka jendela	2	3,4226352
Perintah 6	tolong dibuka jendelanya	buka jendela	2	3,3551119
Perintah 7	tolong dibuka jendelanya	buka jendela	2	3,3953415
Perintah 8	tolong dibuka jendelanya	buka jendela	2	3,4135514
Perintah 9	tolong dibuka jendelanya	buka jendela	2	3,4143326
Perintah 10	tolong dibuka jendelanya	buka jendela	2	3,4257611
rata-rata				3,40080959



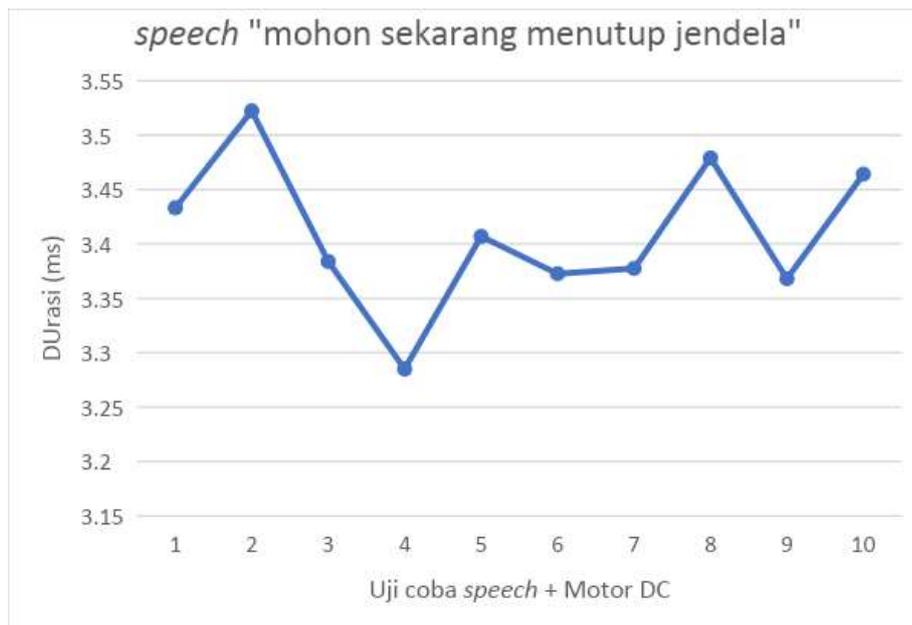
Lampiran 3. Pengujian Kalimat “mohon sekarang membuka jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon sekarang membuka jendela	buka jendela	3	3,4746712
Perintah 2	mohon sekarang membuka jendela	buka jendela	3	3,4300621
Perintah 3	mohon sekarang membuka jendela	buka jendela	3	3,3260734
Perintah 4	mohon sekarang membuka jendela	buka jendela	3	3,4118531
Perintah 5	mohon sekarang membuka jendela	buka jendela	3	3,4383622
Perintah 6	mohon sekarang membuka jendela	buka jendela	3	3,2755443
Perintah 7	mohon sekarang membuka jendela	buka jendela	3	3,4098618
Perintah 8	mohon sekarang membuka jendela	buka jendela	3	3,4330124
Perintah 9	mohon sekarang membuka jendela	buka jendela	3	3,3684613
Perintah 10	mohon sekarang membuka jendela	buka jendela	3	3,4309331
rata-rata				3,3950664



Lampiran 4. Pengujian Kalimat “mohon sekarang menutup jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon sekarang menutup jendela	tutup jendela	4	3,4331643
Perintah 2	mohon sekarang menutup jendela	tutup jendela	4	3,5223811
Perintah 3	mohon sekarang menutup jendela	tutup jendela	4	3,3839232
Perintah 4	mohon sekarang menutup jendela	tutup jendela	4	3,2850525
Perintah 5	mohon sekarang menutup jendela	tutup jendela	4	3,4068532
Perintah 6	mohon sekarang menutup jendela	tutup jendela	4	3,3726213
Perintah 7	mohon sekarang menutup jendela	tutup jendela	4	3,3774714
Perintah 8	mohon sekarang menutup jendela	tutup jendela	4	3,4790304
Perintah 9	mohon sekarang menutup jendela	tutup jendela	4	3,3680922
Perintah 10	mohon sekarang menutup jendela	tutup jendela	4	3,4641712
rata-rata				3,40927608



Lampiran 5. Pengujian Kalimat “saya ingin sekarang membuka jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya ingin sekarang membuka jendela	buka jendela	3	3,3508232
Perintah 2	saya ingin sekarang membuka jendela	buka jendela	3	3,4073854
Perintah 3	saya ingin sekarang membuka jendela	buka jendela	3	3,5051487
Perintah 4	saya ingin sekarang membuka jendela	buka jendela	3	3,2488513
Perintah 5	saya ingin sekarang membuka jendela	buka jendela	3	3,3520421
Perintah 6	saya ingin sekarang membuka jendela	buka jendela	3	3,3966944
Perintah 7	saya ingin sekarang membuka jendela	buka jendela	3	3,4813233
Perintah 8	saya ingin sekarang membuka jendela	buka jendela	3	3,4263285
Perintah 9	saya ingin sekarang membuka jendela	buka jendela	3	3,5470405
Perintah 10	saya ingin sekarang membuka jendela	buka jendela	3	3,3293443
rata-rata				3,40449817



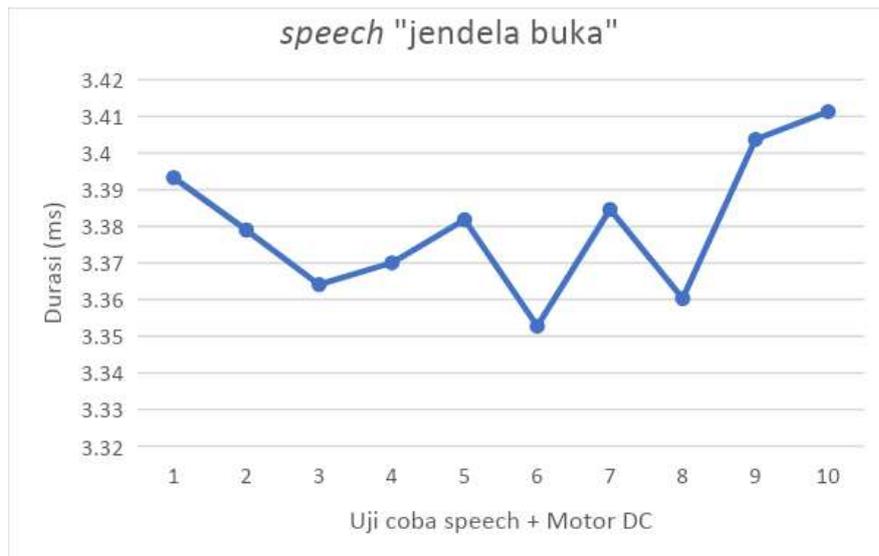
Lampiran 6. Pengujian kalimat “saya ingin sekarang menutup jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya ingin sekarang menutup jendela	tutup jendela	5	3,3576455
Perintah 2	saya ingin sekarang menutup jendela	tutup jendela	5	3,5733124
Perintah 3	saya ingin sekarang menutup jendela	tutup jendela	5	3,4581123
Perintah 4	saya ingin sekarang menutup jendela	tutup jendela	5	3,5153462
Perintah 5	saya ingin sekarang menutup jendela	tutup jendela	5	3,5446944
Perintah 6	saya ingin sekarang menutup jendela	tutup jendela	5	3,3331352
Perintah 7	saya ingin sekarang menutup jendela	tutup jendela	5	3,4208765
Perintah 8	saya ingin sekarang menutup jendela	tutup jendela	5	3,3805117
Perintah 9	saya ingin sekarang menutup jendela	tutup jendela	5	3,3734245
Perintah 10	saya ingin sekarang menutup jendela	tutup jendela	5	3,4242812
rata-rata				3,438133991



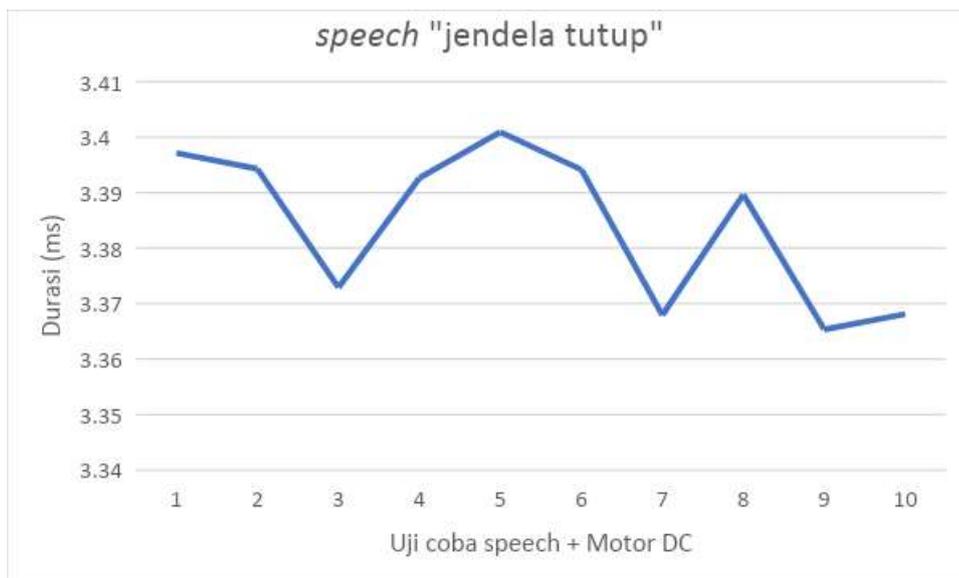
Lampiran 7. Pengujian kalimat “jendela buka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jendela buka	jendela buka	1	3,3932714
Perintah 2	jendela buka	jendela buka	1	3,3789822
Perintah 3	jendela buka	jendela buka	1	3,3640725
Perintah 4	jendela buka	jendela buka	1	3,3700431
Perintah 5	jendela buka	jendela buka	1	3,3818124
Perintah 6	jendela buka	jendela buka	1	3,3527513
Perintah 7	jendela buka	jendela buka	1	3,3846269
Perintah 8	jendela buka	jendela buka	1	3,3603344
Perintah 9	jendela buka	jendela buka	1	3,4037132
Perintah 10	jendela buka	jendela buka	1	3,4113143
rata-rata				3,38009217



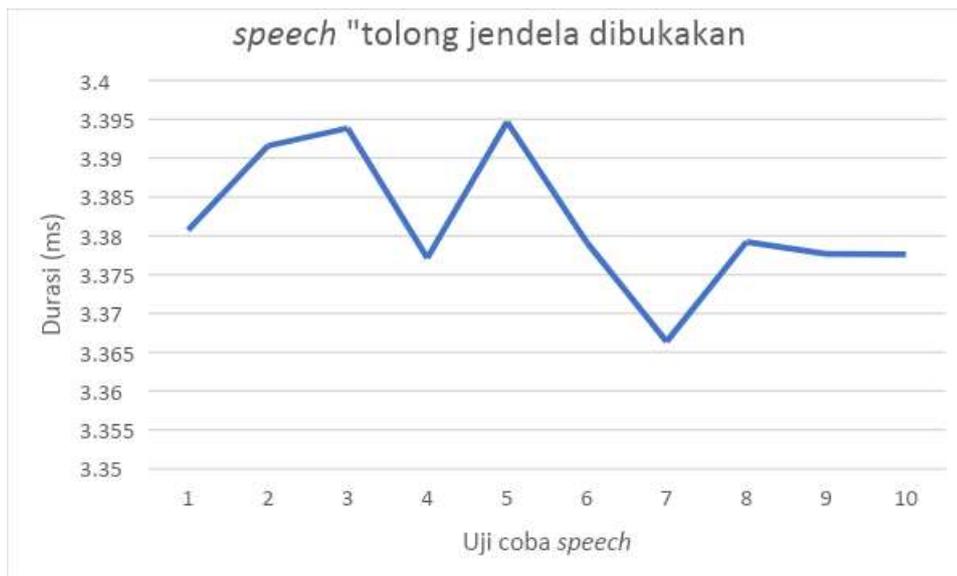
Lampiran 8. Pengujian kalimat “jendela tutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jendela tutup	jendela tutup	1	3,3971544
Perintah 2	jendela tutup	jendela tutup	1	3,3942754
Perintah 3	jendela tutup	jendela tutup	1	3,3729231
Perintah 4	jendela tutup	jendela tutup	1	3,3926118
Perintah 5	jendela tutup	jendela tutup	1	3,4009239
Perintah 6	jendela tutup	jendela tutup	1	3,3941436
Perintah 7	jendela tutup	jendela tutup	1	3,3679322
Perintah 8	jendela tutup	jendela tutup	1	3,3896716
Perintah 9	jendela tutup	jendela tutup	1	3,3653297
Perintah 10	jendela tutup	jendela tutup	1	3,3681112
rata-rata				3,38430769



Lampiran 9. Pengujian kalimat “tolong jendela dibukakan”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong jendela dibukakan	jendela buka	2	3,3807324
Perintah 2	tolong jendela dibukakan	jendela buka	2	3,3915941
Perintah 3	tolong jendela dibukakan	jendela buka	2	3,3938622
Perintah 4	tolong jendela dibukakan	jendela buka	2	3,3771445
Perintah 5	tolong jendela dibukakan	jendela buka	2	3,3946772
Perintah 6	tolong jendela dibukakan	jendela buka	2	3,3791432
Perintah 7	tolong jendela dibukakan	jendela buka	2	3,3663623
Perintah 8	tolong jendela dibukakan	jendela buka	2	3,3792191
Perintah 9	tolong jendela dibukakan	jendela buka	2	3,3776551
Perintah 10	tolong jendela dibukakan	jendela buka	2	3,3775933
rata-rata				3,38179834



Lampiran 10. Pengujian kalimat “tolong jendela ditutupkan”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong jendela ditutupkan	jendela tutup	2	3,3587612
Perintah 2	tolong jendela ditutupkan	jendela tutup	2	3,4221631
Perintah 3	tolong jendela ditutupkan	jendela tutup	2	3,4051322
Perintah 4	tolong jendela ditutupkan	jendela tutup	2	3,3753156
Perintah 5	tolong jendela ditutupkan	jendela tutup	2	3,3700115
Perintah 6	tolong jendela ditutupkan	jendela tutup	2	3,3845924
Perintah 7	tolong jendela ditutupkan	jendela tutup	2	3,3727551
Perintah 8	tolong jendela ditutupkan	jendela tutup	2	3,4022439
Perintah 9	tolong jendela ditutupkan	jendela tutup	2	3,3791374
Perintah 10	tolong jendela ditutupkan	jendela tutup	2	3,4007163
rata-rata				3,38708287



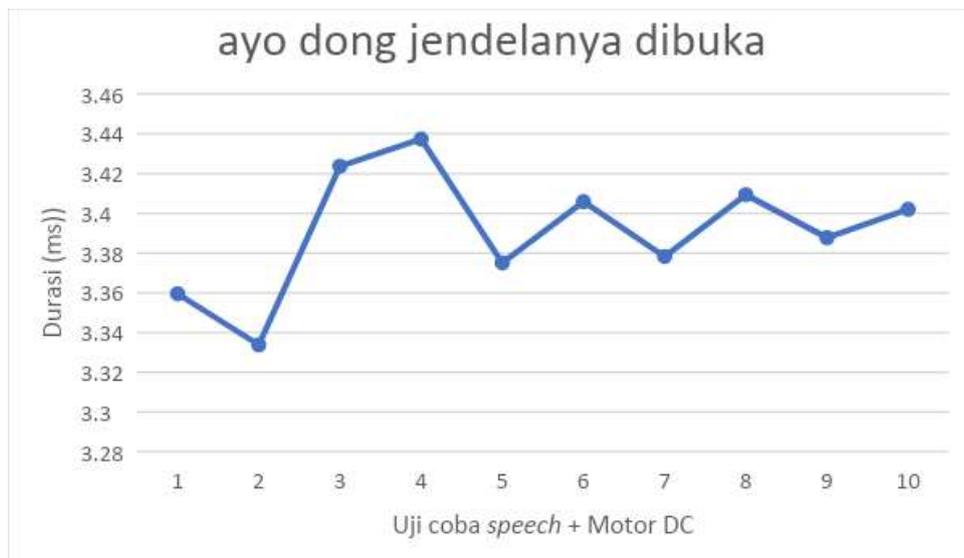
Lampiran 11. Pengujian kalimat “mohon untuk jendela ditutupkan”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon untuk jendela ditutupkan	jendela tutup	2	3,4056125
Perintah 2	mohon untuk jendela ditutupkan	jendela tutup	2	3,4528191
Perintah 3	mohon untuk jendela ditutupkan	jendela tutup	2	3,4125325
Perintah 4	mohon untuk jendela ditutupkan	jendela tutup	2	3,3589164
Perintah 5	mohon untuk jendela ditutupkan	jendela tutup	2	3,2204243
Perintah 6	mohon untuk jendela ditutupkan	jendela tutup	2	3,3842597
Perintah 7	mohon untuk jendela ditutupkan	jendela tutup	2	3,4074758
Perintah 8	mohon untuk jendela ditutupkan	jendela tutup	2	3,4098891
Perintah 9	mohon untuk jendela ditutupkan	jendela tutup	2	3,4452964
Perintah 10	mohon untuk jendela ditutupkan	jendela tutup	2	3,4194772
rata-rata				3,3916703



Lampiran 12. Pengujian kalimat “ayo dong jendelanya dibuka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	ayo dong jendelanya dibuka	jendela buka	2	3,3594951
Perintah 2	ayo dong jendelanya dibuka	jendela buka	2	3,3338224
Perintah 3	ayo dong jendelanya dibuka	jendela buka	2	3,4236733
Perintah 4	ayo dong jendelanya dibuka	jendela buka	2	3,4372948
Perintah 5	ayo dong jendelanya dibuka	jendela buka	2	3,3750798
Perintah 6	ayo dong jendelanya dibuka	jendela buka	2	3,4058231
Perintah 7	ayo dong jendelanya dibuka	jendela buka	2	3,3782687
Perintah 8	ayo dong jendelanya dibuka	jendela buka	2	3,4093331
Perintah 9	ayo dong jendelanya dibuka	jendela buka	2	3,3877243
Perintah 10	ayo dong jendelanya dibuka	jendela buka	2	3,4019876
rata-rata				3,39125022



Lampiran 13. Pengujian kalimat “ayo dong jendela ditutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	ayo dong jendelanya ditutup	jendela tutup	2	3,4333518
Perintah 2	ayo dong jendelanya ditutup	jendela tutup	2	3,3799325
Perintah 3	ayo dong jendelanya ditutup	jendela tutup	2	3,4232949
Perintah 4	ayo dong jendelanya ditutup	jendela tutup	2	3,3808691
Perintah 5	ayo dong jendelanya ditutup	jendela tutup	2	3,3625144
Perintah 6	ayo dong jendelanya ditutup	jendela tutup	2	3,4056316
Perintah 7	ayo dong jendelanya ditutup	jendela tutup	2	3,3389987
Perintah 8	ayo dong jendelanya ditutup	jendela tutup	2	3,3692645
Perintah 9	ayo dong jendelanya ditutup	jendela tutup	2	3,3534327
Perintah 10	ayo dong jendelanya ditutup	jendela tutup	2	3,4382614
rata-rata				3,38855516



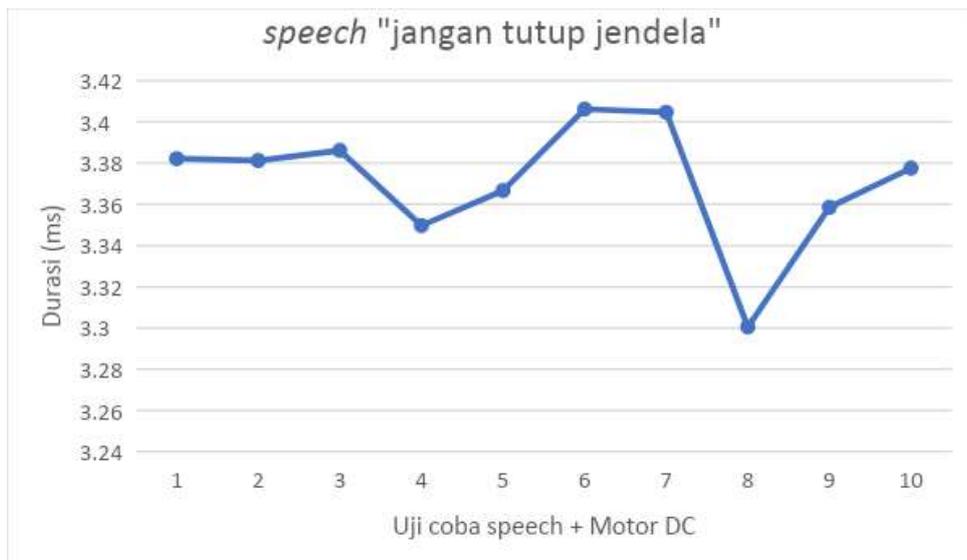
Lampiran 14. Pengujian kalimat “jangan buka jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jangan buka jendela	jangan buka jendela	1	3,3584954
Perintah 2	jangan buka jendela	jangan buka jendela	1	3,3937561
Perintah 3	jangan buka jendela	jangan buka jendela	1	3,3825334
Perintah 4	jangan buka jendela	jangan buka jendela	1	3,3728913
Perintah 5	jangan buka jendela	jangan buka jendela	1	3,3808439
Perintah 6	jangan buka jendela	jangan buka jendela	1	3,3568258
Perintah 7	jangan buka jendela	jangan buka jendela	1	3,3664332
Perintah 8	jangan buka jendela	jangan buka jendela	1	3,3841862
Perintah 9	jangan buka jendela	jangan buka jendela	1	3,3936527
Perintah 10	jangan buka jendela	jangan buka jendela	1	3,3070546
rata-rata				3,36966726



Lampiran 15. Pengujian kalimat “jangan tutup jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jangan tutup jendela	jangan tutup jendela	1	3,3821322
Perintah 2	jangan tutup jendela	jangan tutup jendela	1	3,3812835
Perintah 3	jangan tutup jendela	jangan tutup jendela	1	3,3861598
Perintah 4	jangan tutup jendela	jangan tutup jendela	1	3,3496751
Perintah 5	jangan tutup jendela	jangan tutup jendela	1	3,3666641
Perintah 6	jangan tutup jendela	jangan tutup jendela	1	3,4061671
Perintah 7	jangan tutup jendela	jangan tutup jendela	1	3,4046553
Perintah 8	jangan tutup jendela	jangan tutup jendela	1	3,3004916
Perintah 9	jangan tutup jendela	jangan tutup jendela	1	3,3585819
Perintah 10	jangan tutup jendela	jangan tutup jendela	1	3,3775231
rata-rata				3,37133337



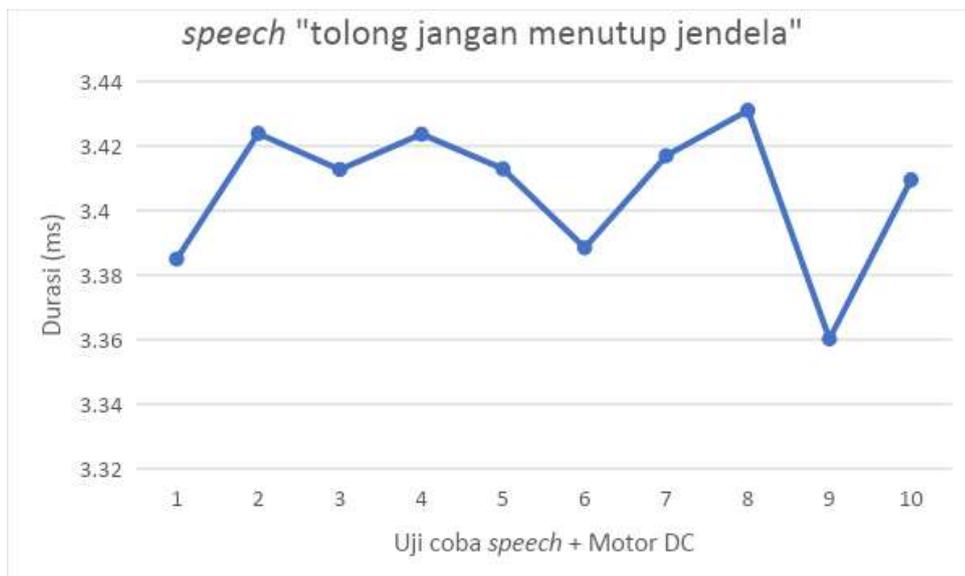
Lampiran 16. Pengujian kalimat “tolong jangan membuka jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong jangan membuka jendela	jangan buka jendela	2	3,4277231
Perintah 2	tolong jangan membuka jendela	jangan buka jendela	2	3,3943654
Perintah 3	tolong jangan membuka jendela	jangan buka jendela	2	3,4271922
Perintah 4	tolong jangan membuka jendela	jangan buka jendela	2	3,3677832
Perintah 5	tolong jangan membuka jendela	jangan buka jendela	2	3,3673748
Perintah 6	tolong jangan membuka jendela	jangan buka jendela	2	3,3790575
Perintah 7	tolong jangan membuka jendela	jangan buka jendela	2	3,4472224
Perintah 8	tolong jangan membuka jendela	jangan buka jendela	2	3,3612435
Perintah 9	tolong jangan membuka jendela	jangan buka jendela	2	3,3977243
Perintah 10	tolong jangan membuka jendela	jangan buka jendela	2	3,4094941
rata-rata				3,39791805



Lampiran 17. Pengujian kalimat “tolong jangan menutup jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong jangan menutup jendela	jangan tutup jendela	2	3,3849523
Perintah 2	tolong jangan menutup jendela	jangan tutup jendela	2	3,4238665
Perintah 3	tolong jangan menutup jendela	jangan tutup jendela	2	3,4126843
Perintah 4	tolong jangan menutup jendela	jangan tutup jendela	2	3,4236431
Perintah 5	tolong jangan menutup jendela	jangan tutup jendela	2	3,4128924
Perintah 6	tolong jangan menutup jendela	jangan tutup jendela	2	3,3884537
Perintah 7	tolong jangan menutup jendela	jangan tutup jendela	2	3,4169377
Perintah 8	tolong jangan menutup jendela	jangan tutup jendela	2	3,4310328
Perintah 9	tolong jangan menutup jendela	jangan tutup jendela	2	3,3601931
Perintah 10	tolong jangan menutup jendela	jangan tutup jendela	2	3,4094933
rata-rata				3,406414914



Lampiran 18. Pengujian kalimat “saya ingin jangan membukakan jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,3880145
Perintah 2	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,4009235
Perintah 3	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,3049321
Perintah 4	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,4444581
Perintah 5	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,3609841
Perintah 6	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,4240943
Perintah 7	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,3551598
Perintah 8	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,4373919
Perintah 9	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,4013765
Perintah 10	saya ingin jangan membukakan jendela	jangan buka jendela	2	3,3904239
rata-rata				3,39077587



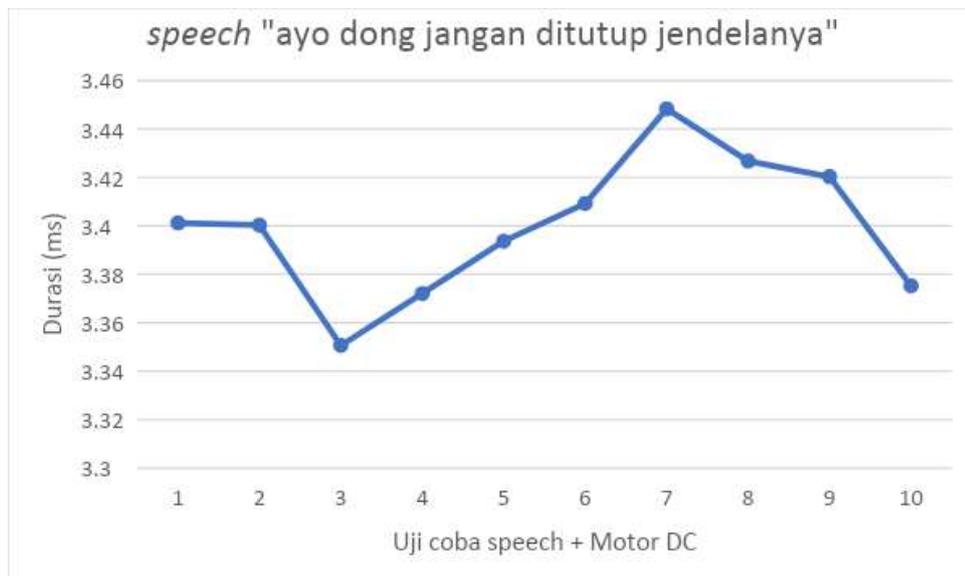
Lampiran 19. Pengujian kalimat “saya ingin jangan menutupkan jendela”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4195926
Perintah 2	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4135832
Perintah 3	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,3955691
Perintah 4	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4151484
Perintah 5	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4302147
Perintah 6	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4385394
Perintah 7	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,3644854
Perintah 8	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4006928
Perintah 9	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4274675
Perintah 10	saya ingin jangan menutupkan jendela	jangan tutup jendela	3	3,4127759
rata-rata				3,4118069



Lampiran 20. Pengujian kalimat “ayo dong jangan ditutup jendelanya”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4011797
Perintah 2	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4002556
Perintah 3	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,3506221
Perintah 4	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,3720891
Perintah 5	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,3936667
Perintah 6	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4092174
Perintah 7	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4482783
Perintah 8	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4267433
Perintah 9	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,4202936
Perintah 10	ayo dong jangan ditutup jendelanya	jangan tutup jendela	2	3,3752279
rata-rata				3,39975737



Lampiran 21. Pengujian kalimat “jendela jangan buka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jendela jangan buka	jendela jangan buka	1	3,3826892
Perintah 2	jendela jangan buka	jendela jangan buka	1	3,3689437
Perintah 3	jendela jangan buka	jendela jangan buka	1	3,3834692
Perintah 4	jendela jangan buka	jendela jangan buka	1	3,4014268
Perintah 5	jendela jangan buka	jendela jangan buka	1	3,3450278
Perintah 6	jendela jangan buka	jendela jangan buka	1	3,4099702
Perintah 7	jendela jangan buka	jendela jangan buka	1	3,3999843
Perintah 8	jendela jangan buka	jendela jangan buka	1	3,3983465
Perintah 9	jendela jangan buka	jendela jangan buka	1	3,3440758
Perintah 10	jendela jangan buka	jendela jangan buka	1	3,3301739
rata-rata				3,37641074



Lampiran 22. Pengujian kalimat “jendela jangan tutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	jendela jangan tutup	jendela jangan tutup	1	3,3909615
Perintah 2	jendela jangan tutup	jendela jangan tutup	1	3,2928771
Perintah 3	jendela jangan tutup	jendela jangan tutup	1	3,3802716
Perintah 4	jendela jangan tutup	jendela jangan tutup	1	3,4017755
Perintah 5	jendela jangan tutup	jendela jangan tutup	1	3,4062229
Perintah 6	jendela jangan tutup	jendela jangan tutup	1	3,3651835
Perintah 7	jendela jangan tutup	jendela jangan tutup	1	3,4159763
Perintah 8	jendela jangan tutup	jendela jangan tutup	1	3,4021143
Perintah 9	jendela jangan tutup	jendela jangan tutup	1	3,4028981
Perintah 10	jendela jangan tutup	jendela jangan tutup	1	3,3301759
rata-rata				3,37884567



Lampiran 23. Pengujian kalimat “mohon jendela jangan dibuka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon jendela jangan dibuka	jendela jangan buka	2	3,4155497
Perintah 2	mohon jendela jangan dibuka	jendela jangan buka	2	3,4086925
Perintah 3	mohon jendela jangan dibuka	jendela jangan buka	2	3,3531531
Perintah 4	mohon jendela jangan dibuka	jendela jangan buka	2	3,3973454
Perintah 5	mohon jendela jangan dibuka	jendela jangan buka	2	3,3881243
Perintah 6	mohon jendela jangan dibuka	jendela jangan buka	2	3,4296247
Perintah 7	mohon jendela jangan dibuka	jendela jangan buka	2	3,3982631
Perintah 8	mohon jendela jangan dibuka	jendela jangan buka	2	3,4703459
Perintah 9	mohon jendela jangan dibuka	jendela jangan buka	2	3,4168529
Perintah 10	mohon jendela jangan dibuka	jendela jangan buka	2	3,4042293
rata-rata				3,40821809



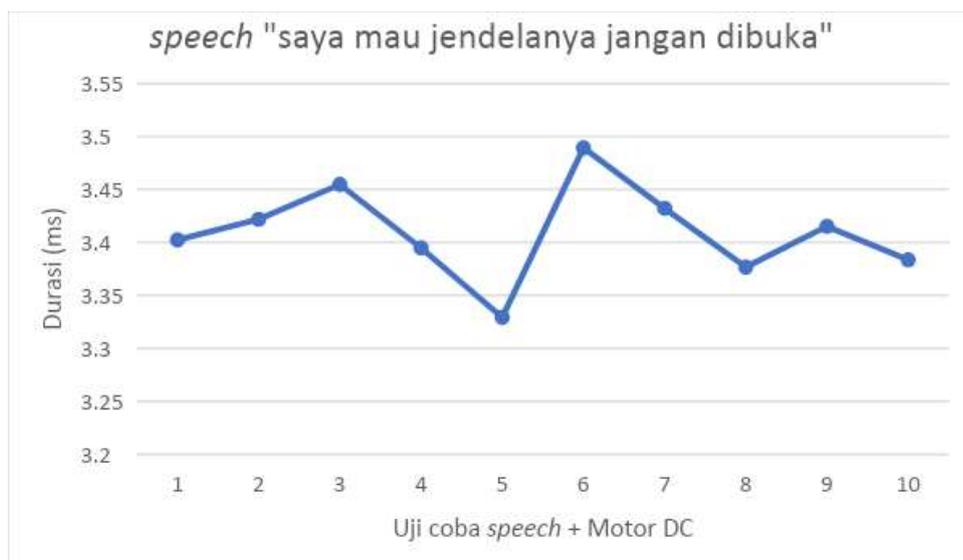
Lampiran 24. Pengujian kalimat “mohon jendela jangan ditutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4325618
Perintah 2	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4176826
Perintah 3	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4149438
Perintah 4	mohon jendela jangan ditutup	jendela jangan tutup	2	3,3692285
Perintah 5	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4132355
Perintah 6	mohon jendela jangan ditutup	jendela jangan tutup	2	3,3993423
Perintah 7	mohon jendela jangan ditutup	jendela jangan tutup	2	3,3980867
Perintah 8	mohon jendela jangan ditutup	jendela jangan tutup	2	3,3862298
Perintah 9	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4194714
Perintah 10	mohon jendela jangan ditutup	jendela jangan tutup	2	3,4199539
rata-rata				3,407073634



Lampiran 25. Pengujian kalimat “saya mau jendelanya jangan dibuka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4023654
Perintah 2	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4218773
Perintah 3	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4546995
Perintah 4	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,3947943
Perintah 5	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,3292798
Perintah 6	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4892756
Perintah 7	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4321677
Perintah 8	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,3767451
Perintah 9	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,4152258
Perintah 10	saya mau jendelanya jangan dibuka	jendela jangan buka	3	3,3835442
rata-rata				3,40999747



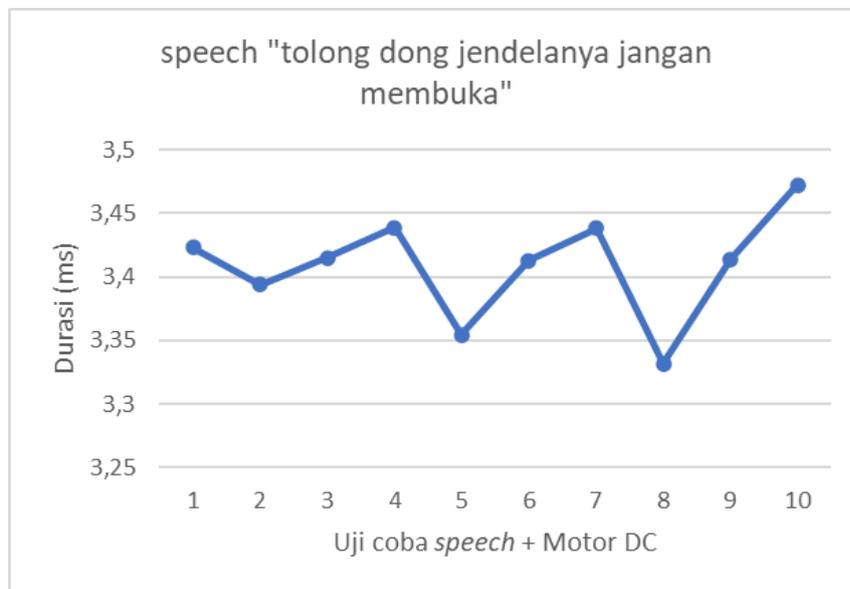
Lampiran 26. Pengujian kalimat “saya mau jendelanya jangan ditutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4068809
Perintah 2	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4230776
Perintah 3	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4310916
Perintah 4	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4087633
Perintah 5	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4202448
Perintah 6	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4017375
Perintah 7	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4133668
Perintah 8	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4235579
Perintah 9	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4341662
Perintah 10	saya mau jendelanya jangan ditutup	jendela jangan tutup	4	3,4256579
rata-rata				3,41885445



Lampiran 27. Pengujian kalimat “tolong dong jendelanya jangan membuka”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4231475
Perintah 2	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,3937947
Perintah 3	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4151968
Perintah 4	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4389664
Perintah 5	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,3543261
Perintah 6	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4126228
Perintah 7	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4380765
Perintah 8	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,3317894
Perintah 9	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4137187
Perintah 10	tolong dong jendelanya jangan membuka	jendela jangan buka	3	3,4723349
rata-rata				3,40939738



Lampiran 28. Pengujian kalimat “tolong dong jendelanya jangan menutup”

No	Kalimat	Pattern	Pattern ditemukan pada iterasi	Durasi (ms)
Perintah 1	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,3997612
Perintah 2	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,3786229
Perintah 3	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4076798
Perintah 4	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4190276
Perintah 5	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4130178
Perintah 6	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4046884
Perintah 7	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4088635
Perintah 8	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4104654
Perintah 9	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4347485
Perintah 10	tolong dong jendelanya jangan menutup	jendela jangan tutup	3	3,4348916
rata-rata				3,41117667

