

**SISTEM KONTROL GORDEN BERBASIS INTERNET OF THINGS
MENGUNAKAN ALGORITMA KNUTH MORRIS PRATT**

SKRIPSI

Oleh :
MIFTAHUL ULUM
NIM. 16650125



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

**SISTEM KONTROL GORDEN BERBASIS *INTERNET OF THINGS*
MENGUNAKAN ALGORITMA *KNUTH MORRIS PRATT***

SKRIPSI

Oleh:

MIFTAHUL ULUM
NIM. 16650125

Diajukan kepada:

**Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang Untuk
Memenuhi Salah Satu Persyaratan Dalam Memperoleh Gelar Sarjana
Komputer (S.Kom)**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022**

LEMBAR PERSETUJUAN

SISTEM KONTROL GORDEN BERBASIS *INTERNET OF THINGS* MENGUNAKAN ALGORITMA *KNUTH MORRIS PRATT*

SKRIPSI

Oleh :

MIFTAHUL ULUM
NIM. 16650125

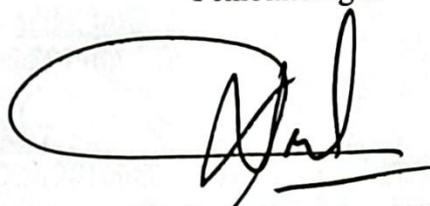
Telah diperiksa dan Disetujui untuk Diuji
Tanggal : 2 Desember 2022

Pembimbing I



Ajib Hanani, M.T
NIDT. 19840731 20160801 1 076

Pembimbing II



Dr. Fresy Nugroho, M.T
NIP. 19710722 201101 1 001

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




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

LEMBAR PENGESAHAN

**SISTEM KONTROL GORDEN BERBASIS *INTERNET OF THINGS*
MENGUNAKAN ALGORITMA *KNUTH MORRIS PRATT***

SKRIPSI

Oleh :

MIFTAHUL ULUM
NIM. 16650125

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal : 9 Desember 2022

Susunan Dewan Penguji

- | | |
|------------------------|-----------------------------------------------------------------|
| 1. Ketua Penguji | : <u>Dr. Totok Chamidy, M.Kom</u>
NIP. 19691222 200604 1 001 |
| 2. Anggota Penguji I | : <u>Juniardi Nur Fadila, M.T</u>
NIP. 19920605 201903 1 015 |
| 3. Anggota Penguji II | : <u>Ajib Hanani, M.T</u>
NIDT. 19840731 20160801 1 076 |
| 4. Anggota Penguji III | : <u>Dr. Fresy Nugroho, M.T</u>
NIP: 19710722 201101 1 001 |

()
()
()
()

Mengetahui :

Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Negeri Maulana Malik Ibrahim Malang




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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Miftahul Ulum

NIM : 16650125

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Sistem Kontrol Gorden Berbasis *Internet of Things*

Menggunakan Algoritma *Knuth Morris Pratt*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan 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 hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 10 Desember 2022

Yang Membuat Pernyataan,



Miftahul Ulum
NIM 16650125

HALAMAN MOTO

حَسْبُنَا اللَّهُ وَنِعْمَ الْوَكِيلُ

"Cukuplah Allah menjadi Penolong kami dan Allah adalah sebaik-baik Pelindung". (QS Ali 'Imran Ayat 173).

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah Puji syukur atas kehadiran Allah SWT, sholawat serta salam kepada Rasulullah SAW. Dengan ini penulis ingin mengungkapkan rasa terima kasih kepada semua pihak yang telah membantu dan mendukung dalam penyelesaian skripsi ini. Ucapan terima kasih ini dipersembahkan kepada:

Kedua orang tua penulis yang dicintai dan hormati, Bapak Drs. Nursalim dan Ibu Isdiyah yang senantiasa memberikan dukungan dan do'a.

Dosen pembimbing penulis Bapak Ajib Hanani, M.T dan Bapak Dr. Fresy Nugroho, M.T yang telah memberikan bimbingan, arahan, dan dukungan kepada penulis selama penyusunan skripsi ini.

Dosen Penguji penulis Bapak Dr. Totok Chamidy, M.Kom dan Bapak Juniardi Nur Fadila, M.T yang telah memberikan masukan dan bimbingan dalam penelitian skripsi ini.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, serta seluruh guru-guru penulis yang telah membimbing dan memberikan ilmunya kepada penulis.

Istri penulis tercinta dan tersayang Rizqa Nur Aziza, S.Trp. dan anak penulis Talitha Kyla Nafisa, yang sudah memberikan dukungan, motivasi, dan do'a dalam penyelesaian skripsi ini.

Almarhum Kakek penulis Bapak H. Muhtarom Muhtar, Nenek penulis Ibu Hj Sumiyati dan Ibu Ni'ayah, yang selalu memberikan dukungan dan do'a.

Kakak penulis Mohammad Farih Fauzi, S.Kom., Adik penulis Hasbi Ashiddiqi, Faldin Aulia Shafi, dan Adhiyat Tijana Langit Jingga yang sudah memberikan dukungan.

Teman-teman seperjuangan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah membantu dan memberikan dukungan kepada penulis selama penyusunan skripsi ini.

Teman-teman "FKMB" yang senantiasa memberi dukungan dan motivasi kepada penulis.

Seluruh keluarga dan rekan yang namanya tidak bisa disebutkan satu-persatu yang sudah memberikan dukungan semangat dan motivasi serta membantu dalam penyelesaian skripsi ini. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah Puji syukur atas ke hadirat Allah SWT yang telah melimpahkan Rahmat serta Hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Sistem Kontrol Gorden Berbasis *Internet of Things* Menggunakan Algoritma Knuth Morris Pratt**” sebagai tugas akhir perkuliahan. Sholawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW atas syafaatnya yang telah menuntun manusia ke jalan yang benar. Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak yang telah membantu terselesaikannya skripsi ini.

Ucapan terima kasih ini penulis sampaikan kepada:

1. Kedua orang tua penulis yang telah memberikan dukungan dan do'a sehingga skripsi ini dapat terselesaikan.
2. Prof. Dr. H. M. Zainuddin, MA, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang dan jajarannya.
3. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang dan jajarannya.
4. Dr. Fachrul Kurniawan ST., M.MT., IPM, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
5. Ajib Hanani, M.T dan Dr. Fresy Nugroho, M.T selaku Dosen Pembimbing skripsi yang sabar dan banyak memberikan arahan serta motivasi selama proses penyusunan skripsi.
6. Dr. Totok Chamidy, M.Kom dan Juniardi Nur Fadila, M.T, selaku Dosen Penguji yang telah membimbing dan memberikan masukan kepada penulis sehingga tercapai hasil skripsi yang lebih baik.

7. Segenap sevitass akademika Jurusan Teknik Informatika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
8. Anggota keluarga dan kerabat yang ada di Banyuwangi maupun di Malang yang selalu memberikan do'a dan semangat kepada penulis.
9. Semua pihak yang ikut membantu dalam menyelesaikan skripsi ini yang tidak bisa disebutkan satu-persatu tanpa mengurangi rasa hormat saya ucapkan terima kasih.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini dapat memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Amin Ya Rabbal 'Alamin.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 10 Desember 2022

Penulis

DAFTAR ISI

COVER	iii
LEMBAR PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
HALAMAN MOTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah.....	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Android	7
2.1 MIT App Inventor	8
2.2 Arduino IDE.....	9
2.3 Algoritma String Matching	10
2.4 Algoritma Knuth Morris Pratt.....	10
2.5 SpeechRecognizer	12
2.6 Stemming	12
2.7 NodeMCU ESP8266 V3	13
2.8 Driver Motor L298N	14
2.9 Motor DC Gearbox dan Pulley(Katrol).....	16
2.10 Kabel Jumper	18
2.11 Adaptor 12VDC	19
2.12 Internet of Things	19
BAB III METODOLOGI PENELITIAN	21
3.1 Desain Dan Implementasi	21
3.1.1 Proses pengambilan Data	22
3.1.2 Proses Stemming	23

3.1.3 Algoritma Knuth Morris Pratt.....	24
3.2 Perancangan Aplikasi.....	26
3.3 Desain Komponen.....	27
3.4 Pengujian Sistem.....	28
BAB IV HASIL DAN PEMBAHASAN	29
4.1 Pembahasan.....	29
4.1.1 Sistem Software	29
4.1.2 Tampilan Aplikasi	29
4.2 Hasil Pengujian <i>Knuth Morris Pratt</i>	35
4.2.1 Proses PreKMP	35
4.2.2 Proses Pengujian Knuth Morris Pratt.....	36
4.2.3 Pengujian I Algoritma Knuth Morris Pratt.....	36
4.2.4 Pengujian II Algoritma Knuth Morris Pratt	37
4.2.5 Pengujian III Algoritma Knuth Morris Pratt	40
4.3 Hasil Pengujian Akurasi.....	46
4.3.1 Pengujian Sistem dengan Confusion Matrix	47
4.3.2 Pengujian Akurasi	49
4.3.3 Pengujian Presisi.....	50
4.3.4 Pengujian Recall	50
4.4 Sistem Hardware	50
4.5 Hasil Pengujian Alat	51
4.6 Integrasi Islam.....	52
BAB V PENUTUP.....	55
5.1 Kesimpulan	55
5.2 Saran.....	55
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Logo Android	7
Gambar 2.2 Logo MIT APP Inventor	8
Gambar 2.3 Logo Arduino IDE	9
Gambar 2.4 Modul NodeMCU ESP8266 V3.....	14
Gambar 2.5 Modul Driver Motor L298N	15
Gambar 2.6 Motor DC Gearbox.....	16
Gambar 2.7 Pulley dan Sabuk.....	17
Gambar 2.8 Kabel Jumper Mikrokontroler	18
Gambar 2.9 Adaptor 12VDC	19
Gambar 3.1 Desain Sistem.....	22
Gambar 3.2 Flowchart Stemming	23
Gambar 3.3 Contoh Hasil Program.....	26
Gambar 3.4 Rancangan Aplikasi.....	27
Gambar 3.5 Rangkaian Komponen	27
Gambar 4.1 Tampilan Awal	29
Gambar 4.2 Kontrol Suara Perintah Buka	30
Gambar 4.3 (1) perintah tidak bisa di ulang (2)perintah salah.....	31
Gambar 4.4 Proses pindah ke screen2.....	32
Gambar 4.5 Sebelum mendapatkan teks	32
Gambar 4.6 Ketika mengklik satu kali tombol Voice.....	32
Gambar 4.7 Ketika menahan tombol Voice	33
Gambar 4.8 Mendapatkan teks dari web.....	33
Gambar 4.9 Sesudah mendapatkan teks.....	33
Gambar 4.10 Ketika Tombol buka di Klik	34
Gambar 4.11 Ketika Tombol Tutup di Klik.....	34
Gambar 4.12 Pesan yang diterima oleh Subscription	35
Gambar 4.13 Nilai LPS pada Pattern	36
Gambar 4.14 Grafik durasi “buka gorden”	41
Gambar 4.15 Grafik durasi “bukalah gordennya”.....	42
Gambar 4.16 Grafik durasi “sekarang buka gordennya”	43
Gambar 4.17 Grafik durasi “sekarang buka gorden dong”	44
Gambar 4.18 Grafik durasi “tolong dong buka gordennya”	45
Gambar 4.19 Grafik rata-rata durasi dari semua pattern.....	46
Gambar 4.20 Rangkaian Komponen	51
Gambar 4.21 Membuka Gorden.....	51
Gambar 4.22 Menutup Gorden	52

DAFTAR TABEL

Tabel 2.1 Spesifikasi Modul NodeMCU ESP8266 V3.....	14
Tabel 2.2 Spesifikasi Modul Driver Motor L298N.....	15
Tabel 2.3 Spesifikasi Motor DC Gearbox.....	17
Tabel 3.1 Fungsi Pattern “BUKA”.....	24
Tabel 3.2 Pencocokan Knuth Morris Pratt.....	25
Tabel 3.3 Confusion matrix.....	28
Tabel 4.1 Pattern “buka gorden(versi gagap)”.....	37
Tabel 4.2 Pattern “tutup gorden(versi gagap)”.....	37
Tabel 4.3 Perintah Salah.....	37
Tabel 4.4 Hasil pengujian dari orang ke-1.....	38
Tabel 4.5 Hasil pengujian dari orang ke-2.....	38
Tabel 4.6 Hasil pengujian dari orang ke-3.....	39
Tabel 4.7 Hasil pengujian dari orang ke-4.....	39
Tabel 4.8 Hasil pengujian dari orang ke-5.....	39
Tabel 4.9 Hasil pengujian “buka gorden”.....	40
Tabel 4.10 Hasil pengujian “bukalah gordennya”.....	41
Tabel 4.11 Hasil pengujian “sekarang buka gordennya”.....	42
Tabel 4.12 Hasil pengujian “sekarang buka gorden dong”.....	43
Tabel 4.13 Hasil pengujian “tolong dong buka gordennya”.....	44
Tabel 4.14 Hasil rata-rata durasi dari semua pattern.....	45
Tabel 4.15 Pattern buka gorden.....	47
Tabel 4.16 Pattern tutup gorden.....	47
Tabel 4.17 Pattern gorden buka.....	48
Tabel 4.18 Pattern gorden tutup.....	48
Tabel 4.18 Bukan Perintah.....	49
Tabel 4.20 Hasil Confusion Matrix.....	49
Tabel 4.21 Keterangan Confusion Matrix.....	49

ABSTRAK

Ulum, Miftahul. 2022 **Sistem Kontrol Gorden Berbasis *Internet of Things* Menggunakan Algoritma *Knuth Morris Pratt***. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T (II) Dr. Fresy Nugrono, M.T

Kata Kunci: *Internet of Things, Knuth Morris Pratt.*

Perkembangan teknologi saat ini mendorong manusia untuk terus menerus berpikir kreatif, tidak hanya menggali penemuan – penemuan baru, tetapi juga dapat memaksimalkan kinerja teknologi yang ada untuk meringankan pekerjaan manusia dalam kehidupan sehari – hari. Kemudahan yang dihasilkan dari perkembangan teknologi yang semakin pesat ini tidak hanya dapat dinikmati oleh seseorang yang berkecimpung dalam dunia teknologi informasi, munculnya *smartphone* yang menjamur di kalangan masyarakat kota maupun desa dipercaya dapat meringankan beban kerja yang ada. Sudah bukan sesuatu yang tidak diketahui oleh masyarakat bahwa fungsi *smartphone* adalah sebagai alat komunikasi jarak jauh yang mempunyai banyak fitur canggih di dalamnya. Adapun sistem operasi yang dimiliki oleh *smartphone* ini ialah salah satunya *android* yang telah dimiliki oleh berbagai kalangan masyarakat. Perancangan aplikasi ini diharapkan mampu membantu seseorang yang sedang sakit dan tidak bisa membuka atau menutup gorden secara langsung dapat melakukannya sambil berbaring atau tetap berada pada posisinya dengan menggunakan *smartphone* yang telah menerapkan aplikasi ini. Dalam penelitian ini penulis ingin merancang suatu penelitian dengan menerapkan aspek – aspek *Internet of Things* yang berjudul “Sistem Kontrol Gorden Berbasis *Internet of Things* Menggunakan Algoritma *Knuth Morris Pratt*”. Sistem kontrol gorden yang telah dibuat menghasilkan suatu alat yang berhubungan dengan aktivitas manusia serta diharapkan dapat memberikan manfaat dalam kehidupan sehari-hari. Hasil analisa sistem kontrol gorden yang dilakukan oleh peneliti dengan menggunakan algoritma *knuth morris pratt* menghasilkan rata-rata durasi dalam pencarian *pattern* 513ms, sedangkan uji akurasi menggunakan rumus *confusion matrix* menghasilkan nilai akurasi sebesar 100%.

ABSTRACT

Ulum, Miftahul. 2022 **Internet of Things Based Control System of Curtain Using the Knuth Morris Pratt Algorithm**. Thesis. Department of Informatics Engineering, Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University of Malang. Advisors: (I) Ajib Hanani, M.T (II) Dr. Fresy Nugrono, M.T

Keywords: *Internet of Things, Knuth Morris Pratt.*

Current technological developments pushing humans to continuously think creatively, not only to explore new discoveries, but also to be able to maximize the performance of existing technology to lighten human work in everyday life. The convenience resulting from the rapid development can not only be enjoyed by someone who is involved in the world of information technology, the emergence of *smartphone* that are mushrooming among urban and rural communities is believed to lighten the existing workload. Many people know that *smartphone* is used as a means of long distance communication tool that have many advanced featured. The operating system owned by the *smartphone*, one of is *android* that has been owned by various circle of society. The design of this application is expected to be able to help someone who is sick and unable to open or close the curtains directly can do so while lying down or staying in position using a *smartphone* that has implemented this application. In this research the writer design some of research with use aspects of the *internet of thing* with the tittle “Internet of Things Based Control System of Curtain Using the Knuth Morris Pratt Algorithm”. Control system of curtain has been created produces a tool related to human activity and is expected to provide benefits in everyday life. Results of curtain control system analysis conducted by researchers with use *Knuth Morris Pratt Algorithm* produces an average duration in *pattern* search of 513ms, while the accuracy test using the *confusion matrix* formula produces an accuracy value of 100%.

مستخلص

العلوم، مفتاح. 2022. نظام رقابة الستار على أساس إنترنت لكل الأشياء باستخدام خوارزمية *Knuth Morris Pratt*.

البحث الجامعي. قسم الهندسة المعلوماتية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج.

المشرف : (1) عجيب هنري الماجستير، (2) الدكتور فرسي نوعروهو الماجستير

الكلمات المفتاحية : إنترنت لكل الأشياء، *Knuth Morris Pratt*

يدفع تطور التكنولوجيا إلى الإنسان على التفكير الإبداعي، ليس فقط كشف المكتشفات الجديدة، ولكن يمكن أيضاً تكميل أداء التكنولوجيا لتسهيل العمل الإنساني في الحياة اليومية. السهولة الناتجة من ناضر تطور التكنولوجيا لا يتمتع بها فقط منكب في تكنولوجيا المعلومات، يعتقد كون الهواتف الذكية في المجتمع المدنية أو القربوية على تخفيف عبء العمل الحالي. والجدير بالذكر أن وظيفة الهواتف الذكية في المجتمع هي وسائل الاتصالات وفيها الميزات المتطورة. وأحد نظام التشغيل المستخدم في الهواتف الذكية هو أندرويد الذي يمتلكه المجتمع المختلفة. يرجى أن يعين تصميم هذا التطبيق متوجعين ولا يمكن أن يفتحوا ويغلقوا الستار مباشرة حتى يأتي باستلقاء أو البقاء في موضعهم باستخدام الهواتف الذكية. يصمم الباحث في هذا البحث بتطبيق جوانب إنترنت لكل الأشياء بموضوع نظام رقابة الستار على أساس إنترنت لكل الأشياء باستخدام خوارزمية *Knuth Morris Pratt*. ينتج نظام رقابة الستار أداة تتعلق بالأنشطة اليومية ويرجى أن يعطي الفوائد في الحياة اليومية. نتائج تحليل نظام رقابة الستار باستخدام خوارزمية *Knuth Morris Pratt* تعطي متوسط مدة في بحث النموذج ٥١٣ م س. بينما ينتج إختبار الدقة باستخدام رمز *confusion matrix* قيمة الدقة تبلغ ١٠٠٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini mendorong manusia untuk terus menerus berpikir kreatif, tidak hanya menggali penemuan – penemuan baru, tetapi juga dapat memaksimalkan kinerja teknologi yang ada untuk meringankan pekerjaan manusia dalam kehidupan sehari – hari. Kemudahan yang dihasilkan dari perkembangan teknologi yang semakin pesat ini tidak hanya dapat dinikmati oleh seseorang yang berkecimpung dalam dunia teknologi informasi, munculnya *smartphone* yang menjamur di kalangan masyarakat kota maupun desa dipercaya dapat meringankan beban kerja yang ada. Sudah bukan sesuatu yang tidak diketahui oleh masyarakat bahwa fungsi *smartphone* adalah sebagai alat komunikasi jarak jauh yang mempunyai banyak fitur canggih di dalamnya. Adapun sistem operasi yang dimiliki oleh *smartphone* ini ialah salah satunya *android* yang telah dimiliki oleh berbagai kalangan masyarakat.

Perkembangan teknologi informasi dan komunikasi saat ini berkembang dengan pesat seiring dengan penemuan dan pengembangan Ilmu Pengetahuan dalam bidang informasi dan komunikasi sehingga mampu menciptakan alat-alat yang mendukung perkembangan teknologi informasi, mulai dari sistem komunikasi sampai dengan alat komunikasi yang searah maupun dua arah (interaktif). Kemajuan tersebut telah memberikan kemudahan-kemudahan dan kesejahteraan bagi kehidupan manusia sekaligus merupakan sarana bagi kesempurnaan manusia sebagai hamba Allah dan khalifah-Nya. Karena Allah telah mengaruniakan

anugerah kenikmatan kepada manusia yang bersifat saling melengkapi yaitu anugerah agama dan kenikmatan teknologi. Salah satu yang tersirat dari firman Allah dalam Al-Qur'an Surat Ar-Rahman Ayat 33, yaitu:

يَا مَعْشَرَ الْجِنِّ وَالْإِنْسِ إِنِ اسْتَطَعْتُمْ أَنْ تَنْفُذُوا مِنْ أَقْطَارِ السَّمَاوَاتِ وَالْأَرْضِ فَانفُذُوا لَا تَنْفُذُونَ إِلَّا بِسُلْطَانٍ

"Wahai golongan jin dan manusia! Jika kamu sanggup menembus (melintasi) penjuru langit dan bumi, maka tembuslah. Kamu tidak akan mampu menembusnya kecuali dengan kekuatan (dari Allah)" (QS. Ar-Rahman : 33).

Sehingga dapat ditarik kesimpulan bahwa dengan adanya perkembangan teknologi yang pesat saat ini, diharapkan juga mampu meningkatkan penyebaran dan pendalaman terhadap agama Islam, mengingat mudahnya penyampaian informasi saat ini dengan penggunaan *smartphone*.

Pemanfaatan *smartphone android* sebagai alat komunikasi dan telepon cerdas kini telah mengalami banyak perkembangan, salah satunya ialah implementasi dari *Internet of Things* yang sudah sering digunakan seperti contohnya *smart watering*, *smart home*, penerapan aplikasi untuk mematikan dan menyalakan lampu dengan menggunakan sensor suara. Adapun permasalahan yang diangkat dalam penelitian ini mengacu pada kemampuan manusia yang berbeda – berbeda dalam melakukan kegiatannya, penelitian ini dirancang untuk mencapai tujuan berupa membantu seseorang yang mengalami sakit fisik atau cacat yang tidak mampu melakukan kegiatan dengan leluasa, perancangan aplikasi ini diharapkan mampu membantu seseorang yang sedang sakit dan tidak bisa membuka atau menutup gorden secara langsung dapat melakukannya sambil berbaring atau tetap berada pada posisinya dengan menggunakan *smartphone* yang telah menerapkan aplikasi ini. Dalam penelitian ini penulis ingin merancang suatu penelitian dengan

menerapkan aspek – aspek *Internet of Things* yang berjudul “Sistem Kontrol Gorden Berbasis *Internet of Things* Menggunakan Algoritma *Knuth Morris Pratt*. Algoritma *Knuth Morris Pratt* merupakan proses pencocokan string (Fau & Ginting, 2017). Algoritma ini dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977, penggunaan Algoritma *Knuth Morris Pratt* pada penelitian ini yaitu guna mencocokkan semua *Pattern* yang masuk atau diterima, sehingga mencapai tujuan penggunaan kalimat yang tepat dalam melakukan perintah melalui sensor suara diantara-Nya yaitu buka gorden, tutup gorden, gorden buka, gorden tutup.

Hasil dari penelitian ini diharapkan dapat menciptakan suatu aplikasi cerdas yang dapat mengontrol gorden yang diantara-Nya ialah membuka dan menutup gorden dengan perintah suara melalui android, dihasilkan dari sebuah aplikasi yang dibuat di *MIT App Inventor* dengan beberapa *tools* elektro yang diantara-Nya adalah NodeMCU ESP8266 V3, Driver Motor L298N, Motor DC Gearbox, kabel *jumper* secukupnya dan Adaptor 12v DC. Menggunakan NodeMCU sebagai mikrokontroler yang mempunyai komponen atau rangkaian elektronik dengan fungsi membaca input, memproses input, dan mengendalikannya, kemudian terdapat program *MIT App Inventor* yang digunakan untuk menciptakan aplikasi perangkat lunak yang dalam implementasinya cukup mudah untuk digunakan. Adapun penggunaan algoritma *knuth morris pratt* ialah untuk melakukan pencarian string dengan menggunakan *Pattern* teks atau suara yang diucapkan sehingga

perintah yang digunakan untuk membuka gorden bisa fleksibel untuk diucapkan seperti “buka gorden ataupun gorden buka”.

1.2 Rumusan Masalah

Adapun rumusan masalah yang didapat yaitu :

- A. Bagaimana implementasi sistem kontrol gorden berbasis *internet of things* menggunakan algoritma *knuth morris pratt*?
- B. Bagaimana analisa sistem kontrol gorden berbasis *internet of things* menggunakan algoritma *knuth morris pratt*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

- A. Implementasi sistem kontrol gorden berbasis *internet of things* menggunakan algoritma *knuth morris pratt*.
- B. Analisa sistem kontrol gorden berbasis *internet of things* menggunakan algoritma *knuth morris pratt*.

1.4 Batasan Masalah

Batasan masalah dibuat guna mencegah melebarnya topik dalam penelitian, adapun batasan – batasan masalah pada penelitian ini ialah :

- A. Rel gorden yang digunakan jenis aluminium kotak berwarna silver.
- B. Menggunakan mikrokontroler NodeMCU ESP8266 V3 dan Driver Motor L298N
- C. Menggunakan satu koneksi jaringan yang sama.
- D. Kontrol gorden hanya buka dan tutup.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

- A. Memudahkan pekerjaan manusia dengan menerapkan *internet of things* yang bisa mengontrol aktivitas gorden melalui *smartphone android*.
- B. Memberi kenyamanan dan kemudahan dalam melakukan aktivitas khususnya untuk para penyandang cacat fisik atau orang yang sudah tua.
- C. Sebagai referensi untuk penelitian selanjutnya.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan pada penelitian adalah sebagai berikut :

A. BAB I PENDAHULUAN

Berisikan poin – poin pendahuluan dalam penelitian yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan

B. BAB II STUDI PUSTAKA

Berisikan kajian – kajian mengenai teori pendukung penelitian, seperti pengertian dari para ahli mengenai teori – teori yang mendukung penelitian.

C. BAB III DESAIN DAN IMPLEMENTASI

Berisikan kerangka pemikiran penelitian, terdapat rancangan – rancangan sistem yang diinginkan dalam penelitian.

D. BAB IV UJI COBA DAN PEMBAHASAN

Berisikan hasil dan pembahasan dari penelitian yang telah dilakukan.

E. BAB V KESIMPULAN DAN SARAN

Berisikan kesimpulan yang didapat dari penelitian, juga saran yang diajukan guna melakukan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Android

Menurut Teguh Arifianto (2011 : 1) android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang berbasis linux. Menurut Hermawan (2011 : 1) android merupakan OS (Operating System) mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. Android merupakan salah satu sistem operasi yang user friendly atau mudah dan bersifat open source yang bisa di kembangkan sendiri oleh siapapun, sistem operasi Android pun mudah digunakan untuk kalangan anak muda maupun tua.

Android merupakan sistem operasi yang diciptakan oleh Google dan merupakan pengembangan dari bahasa Java. Saat ini perkembangan sistem operasi Android sangatlah cepat. Android pertama kali dirilis pada 5 November 2007 dengan nama Android Beta dan pada 23 September 2008 Android 1.0 (Astro) dirilis dan versi terbaru Android adalah 7.0 (Nougat). Berikut ini merupakan logo android pada Gambar 2.1.



Gambar 2.1 Logo Android
Sumber: <https://ridwanromadhon.com/>

2.1 MIT App Inventor

Menurut Amerkashi (2015:15): “*App Inventor Android is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT).* (App Inventor Android adalah sebuah aplikasi *web open-source* asli yang disediakan oleh Google, dan sekarang dikelola oleh *Massachusetts Institute of Technology (MIT)*).”.

Menurut Kamriani dan Roy (2016:2): “*MIT App Inventor 2 is a free, drag-and-drop, blocks-based visual programming language that enables people, regardless of their coding experience, to create mobile apps for Android devices.* (MIT App Inventor 2 adalah gratis, tahan-dan-lepas, suatu bahasa pemrograman visual berbasis blok yang dikenali orang, terlepas dari pengalaman koding mereka, untuk membuat aplikasi *mobile* untuk perangkat Android).” Berdasarkan pengertian-pengertian yang telah didefinisikan oleh para ahli tersebut maka dapat disimpulkan bahwa MIT App Inventor 2 adalah aplikasi berbasis *web open-source* (sumber terbuka) yang disediakan oleh Google untuk membuat aplikasi perangkat Android. Berikut ini merupakan logo MIT APP Inventor pada Gambar 2.2.



Gambar 2.2 Logo MIT APP Inventor
Sumber: <https://psti.unisayogya.ac.id/>

2.2 Arduino IDE

Menurut (Setiawan & Abdullah, 2021), Software yang digunakan untuk memprogram dan mengunggah program ke Arduino adalah Arduino IDE. Kode program yang ditulis untuk Arduino dikenal sebagai Sketsa.

IDE (*Integrated Development Environment*) berarti bentuk alat pengembangan program yang terintegrasi sehingga berbagai keperluan disediakan dan dinyatakan dalam bentuk antarmuka berbasis menu.

Menurut (Endra, 2019), IDE merupakan kependekan dari *Integrated Development Environment*. IDE merupakan program yang digunakan untuk membuat program pada Esp 8266 NodeMcu.

Program yang ditulis dengan menggunakan Software Arduino IDE disebut sebagai sketch. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi .ino. Berikut ini merupakan logo Software Arduino pada Gambar 2.3.



Gambar 2.3 Logo Arduino IDE
Sumber: <http://sistem-komputer-s1.stekom.ac.id/>

Pada Software Arduino IDE, terdapat semacam message box berwarna hitam yang berfungsi menampilkan status, seperti pesan error, compile, dan upload program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.

- a. Verify/Compile, berfungsi untuk mengecek apakah sketch yang dibuat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang dibuat akan dicompile kedalam Bahasa mesin.
- b. Upload, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.

2.3 Algoritma *String Matching*

Algoritma *string matching* dalam bahasa Indonesia dikenal dengan istilah algoritma pencocokan *string*. Persoalan pencarian *string* dirumuskan sebagai berikut:

1. Sebuah teks (*text*), yaitu sebuah (*long*) *string* yang panjangnya n karakter.
2. *Pattern*, yaitu sebuah *string* dengan panjang m karakter ($m < n$) yang akan dicari dalam teks.

Carilah lokasi pertama di dalam teks yang bersesuaian dengan *pattern*. sebuah kata dalam dokumen misalnya menu *Find* dalam *Microsoft Word*. (Astuti, 2017)

Contoh :

Pattern : Buka

Teks : Tolong **Buka** Gorden

↑
target

2.4 Algoritma *Knuth Morris Pratt*

Algoritma *Knuth-Morris-Pratt* adalah salah satu algoritma pencarian *string*, dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya

mempublikasikannya secara bersamaan pada tahun 1977 (Manikandan & Ramyachitra, 2018).

Dalam algoritma KMP dapat diketahui bahwa kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini akan menghemat perbandingan yang selanjutnya akan meningkatkan kecepatan pencarian. Perhitungan pergeseran pada algoritma ini adalah sebagai berikut, bila terjadi ketidakcocokan pada saat *pattern* sejajar dengan teks[$i..i+n-1$], dapat menganggap ketidakcocokan pertama terjadi di antara teks[$i+j$] dan *pattern*[j], dengan $0 < j < n$. Berarti, teks[$i..i+j-1$]=*pattern*[$0..j-1$] dan $a = \text{teks}[i+j]$ tidak sama dengan $b = \text{pattern}[j]$. Ketika kita menggeser, sangat beralasan bila ada sebuah awalan v dari *pattern* akan sama dengan sebagian akhiran u dari sebagian teks. Sehingga kita bisa menggeser *pattern* agar awalan v tersebut sejajar dengan akhiran dari u (Nursobah & Pahrudin, 2019).

KMP dapat ditanggihkan secara efektif untuk *preprocessing* pola, sehingga komponen *Longest Proper Prefix Which Is Suffix* (LPS) dibangun berdasarkan pola yang dicari sebelum program utama dijalankan. LPS adalah *array* dari sebuah *string* pada posisi tertentu yang merupakan awalan yang tepat (artinya tidak ada awalan untuk karakter terakhir dari *string*) dan juga akhiran. LPS ini digunakan untuk pergeseran terbesar yang dapat dilakukan untuk menghindari perbandingan yang tidak perlu (Fathimah Qur', 2020). Kecocokan terjadi jika karakter dalam teks dan karakter dalam pola yang dibandingkan sama, dan sebaliknya jika tidak cocok. (Astuti, 2017).

Preprocessing dilakukan untuk mencari fungsi pinggiran rumus yang digunakan : $b(k)$ = jumlah awalan prefiks dari $P[0..k]$ yang merupakan sufiks dari $P[1..k]$ (Fathimah Qur', 2020).

2.5 *Speech Recognizer*

Secara umum, *speech recognizer* memproses sinyal suara yang masuk dan menyimpannya dalam bentuk digital. Hasil proses digitalisasi tersebut kemudian dikonversi dalam bentuk spektrum suara yang akan *dianalisa* dengan membandingkannya dengan *template* suara pada *database* sistem. (Andriana; Olly V; Riyanto S; Ganjar T; Zulkarnain, 2016).

Sedangkan cara kerja *Speech Recognizer* yang di gunakan oleh peneliti saat ini yaitu *Google Speech Recognizer* memiliki cara kerja mengubah suara menjadi teks kemudian dibandingkan dengan *pattern* pada algoritma *Knuth Morris Pratt*. Dalam penggunaan *Google Speech Recognizer* ini akan mempermudah untuk mengendalikan gorden buka maupun gorden tutup.

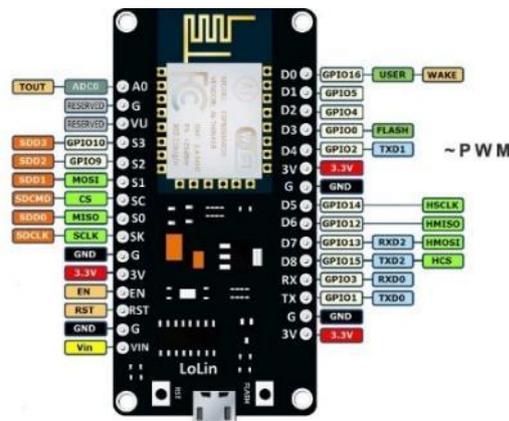
2.6 *Stemming*

Stemming merupakan suatu proses yang terdapat dalam sistem IR yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya (*rootword*) dengan menggunakan aturan-aturan tertentu. Sebagai contoh, kata bersama, kebersamaan, menyamai, akan *distem* ke *root wordnya* yaitu "sama". (Pramudita, 2014).

2.7 NodeMCU ESP8266 V3

Menurut Saputro, 2017 dalam *website* <https://embeddednesia.com/> memberikan pengertian bahwa NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip* ESP8266 dari ESP8266 buatan **Espressif System**, juga *firmware* yang digunakan, yang menggunakan bahasa pemrograman *scripting* Lua. Istilah NodeMCU secara *default* sebenarnya mengacu pada firmware yang digunakan daripada perangkat keras development kit. NodeMCU bisa dianalogikan sebagai board arduino-nya ESP8266. Namun NodeMCU telah *me-package* ESP8266 ke dalam sebuah *board* yang kompak dengan berbagai fitur layaknya mikrokontroler + kapabilitas akses terhadap Wifi juga chip komunikasi USB to serial. Sehingga untuk memprogramnya hanya diperlukan ekstensi kabel data USB persis yang digunakan sebagai kabel data dan kabel *charging* smartphone Android.

NodeMCU ESP8266 V3 ini sejatinya juga sebuah mikrokontroler, seperti Arduino, yang ditambahi dengan modul WiFi ESP8266. Selain terdapat memori untuk menyimpan program, juga tersedia port digital *Input – Output*, sebuah port analog input serta port dengan fungsi khusus seperti serial UART, SPI, I2C dll. Berikut merupakan spesifikasi dan bentuk modul NodeMCU ESP8266 V3 yang ditunjukkan pada Gambar 2.4 dan Tabel 2.1.



Gambar 2.4 Modul NodeMCU ESP8266 V3
Sumber: <http://reslab.sk.fti.unand.ac.id/>

Tabel 2.1 Spesifikasi Modul NodeMCU ESP8266 V3

Mikrokontroler / Chip	ESP8266-12E
Tegangan Input	3.3 ~ 5V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz
USB Port	Micro USB
USB Chip	CH340G

Sumber: <https://www.nn-digital.com/blog/2019/07/27/memulai-pemrograman-nodemcu-esp8266-menggunakan-arduino-ide/>

2.8 Driver Motor L298N

Modul Driver Motor L298N ini adalah sebuah H-Bridge Dual Motor Controller 2A yang memungkinkan kita untuk mengatur arah putaran maupun kecepatan dari satu atau dua motor DC. Selain itu, dengan modul driver motor ini kita juga dapat mengontrol sebuah motor stepper bipolar dengan mudah (Prastyo, 2020) dalam website <https://www.edukasielatronika.com/>

Pada IC L298N terdiri dari transistor-transistor logik (TTL) dengan gerbang nand yang berfungsi untuk memudahkan dalam menentukan arah putaran suatu motor dc maupun motor stepper. Untuk dipasaran sudah terdapat modul driver motor menggunakan IC L298N ini, sehingga lebih praktis dalam penggunaannya karena pin I/O nya sudah terpackage dengan rapi dan mudah digunakan. Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor sehingga motor lebih mudah untuk dikontrol. Berikut spesifikasi dan bentuk modul driver motor L298N ditunjukkan pada Gambar 2.5 dan Tabel 2.2.



Gambar 2.5 Modul Driver Motor L298N

Sumber: <https://www.mahirelektro.com/2020/02/tutorial-menggunakan-driver-motor-l298n-pada-Arduino.html>

Tabel 2.2 Spesifikasi Modul Driver Motor L298N

Tegangan Input	3.2V - 40V.
Driver	Driver Motor L298N Dual H Bridge DC
Catu Daya	5V
Arus puncak	2 Amper
Kisaran operasi	0 - 36 mA
Konsumsi daya maksimum	20W (ketika suhu 75 °C)
Suhu penyimpanan	-25 °C ~ +130 °C
Keluaran pin 10 (sumber tegangan IC) jika berfungsi sebagai pin output	5V
Ukuran	3.4 cm x 4.3 cm x 2.7 cm

Sumber: <https://www.mahirelektro.com/2020/02/tutorial-menggunakan-driver-motor-l298n-pada-Arduino.html>

2.9 Motor DC Gearbox dan *Pulley*(Katrol)

Motor DC merupakan alat yang dilengkapi dengan gearbox. Motor DC ini akan berperan sebagai alat penggerak buka dan tutup gorden yang mempunyai input dc 3 – 6 volt dan akan bergerak atau berputar ketika mendapatkan perintah dari NodeMCU ESP8266.

Menurut (Putri, dkk., 2021) Dinamo DC adalah sebuah perangkat yang mengubah energi listrik menjadi energi kinetik atau gerakan motion. Dinamo ini juga disebut sebagai dinamo arus searah. Dinamo ini memiliki dua terminal dan memerlukan tegangan arus searah untuk dapat menggerakkannya. Berikut ini merupakan bentuk Motor DC Gearbox pada Gambar 2.6.



Gambar 2.6 Motor DC Gearbox

Sumber: <https://medium.com/bisa-ai/komponen-dan-cara-membuat-line-follower-robot-5614ff8c4c09>

Gearbox merupakan alat penyalur tenaga antara as pada dinamo dengan beberapa gear sehingga dapat disebut gearbox. Gearbox ini nantinya akan membantu putaran yang lebih lancar dan tidak memberatkan dinamo.

Menurut (Penu, dkk., 2021) Gearbox adalah suatu alat atau komponen mekanik yang digunakan untuk memindahkan tenaga penggerak untuk menyesuaikan daya atau putaran dari motor dc yang berputar, dan gearbox juga

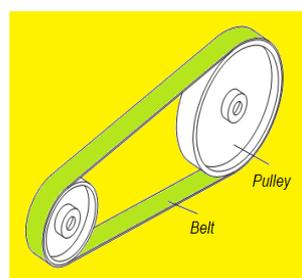
merupakan alat yang dapat meningkatkan daya atau tenaga yang dibutuhkan oleh dinamo. Berikut ini merupakan spesifikasi motor dc gearbox pada Tabel 2.3.

Tabel 2.3 Spesifikasi Motor DC Gearbox

Dimensi Gearbox	panjang 72 mm
	tebal 35 mm
	tinggi 22 mm
Tegangan kerja motor	3 volt – 6 volt
Kecepatan tanpa beban	3V: 90 +/- 10% rpm /
	6V: 200 +/- 10% rpm

Sumber: <https://sismik.stei.itb.ac.id/category/uncategorized/page/4/>

Dalam website <https://en.wikipedia.org/wiki/Katrol> adalah roda pada poros atau poros yang dirancang untuk mendukung gerakan dan perubahan arah kabel atau sabuk yang kencang, atau transfer daya antara poros dan kabel atau sabuk. Dalam kasus katrol yang ditopang oleh rangka atau cangkang yang tidak mentransfer daya ke poros, tetapi digunakan untuk memandu kabel atau mengerahkan gaya, cangkang penopang disebut balok, dan katrol dapat disebut katrol. Berikut ini merupakan bentuk rangkaian katrol dan sabuk pada Gambar 2.7.



Gambar 2.7 Pulley dan Sabuk

Sumber: <https://teknikmesinmanufaktur.blogspot.com/2019/07/pulley-dan-belt.html>

Katrol mungkin memiliki alur atau alur di antara flensa di sekitar kelilingnya untuk menemukan kabel atau sabuk. Elemen penggerak sistem katrol dapat berupa tali, kabel, sabuk, atau rantai.

2.10 Kabel *Jumper*

Menurut Pratama, 2019 dalam *website* <https://otopedia.com/> *Jumper* dalam istilah otomotif merupakan sebuah set kabel elektrik berisolasi yang digunakan untuk menyalakan mesin mobil dengan menggunakan *power* unit seperti aki eksternal. Kabel tersebut disambungkan ke terminal positif dan terminal negatif aki. Untuk memberikan sambungan yang mengikat, ujung-ujung kabel dilengkapi dengan jepitan untuk menjepit masing-masing terminal aki. *Jumper* berfungsi membantu mobil yang memiliki kendala aki tekor atau rusak untuk menghidupkan mesin dengan cara bantuan tenaga listrik dari aki mobil lain. Aki yang dijadikan *pen-jumper* minimal harus memiliki kapasitas dengan aki yang tekor. Jika aki yang tekor memiliki kapasitas 40 AH, maka aki *pen-jumper* minimal harus memiliki aki dengan kapasitas sama.

Kabel *jumper* adalah suatu istilah kabel yang ber-diameter kecil yang di dalam dunia elektronika digunakan untuk menghubungkan dua titik atau lebih dan dapat juga untuk menghubungkan 2 komponen elektronika. Berikut ini merupakan bentuk kabel *jumper* mikrokontroler pada Gambar 2.8.



Gambar 2.8 Kabel *Jumper* Mikrokontroler
Sumber: <https://www.aldyrazor.com/2020/04/kabel-jumper-arduino.html>

2.11 Adaptor 12VDC

Adaptor adalah sebuah rangkaian yang berguna untuk mengubah tegangan AC yang tinggi menjadi DC yang rendah. Adaptor merupakan sebuah alternatif pengganti dari tegangan DC (seperti ;baterai,Aki) karena penggunaan tegangan AC lebih lama dan setiap orang dapat menggunakannya asalkan ada aliran listrik. Adaptor juga banyak di gunakan dalam alat sebagai catu daya, layaknya amplifier, radio, pesawat televisi mini dan perangkat elektronik lainnya, (<https://elektrologi.iptek.web.id/>). Berikut merupakan bentuk Adaptor 12VDC pada Gambar 2.9.



Gambar 2.9 Adaptor 12VDC

Sumber: <https://wikielektronika.com/adaptor-adalah/>

2.12 *Internet of Things*

IoT (Internet of Thing) dapat didefinisikan kemampuan berbagai device yang bisa saling terhubung dan saling bertukar data melalui jaringan internet. IoT merupakan sebuah teknologi yang memungkinkan adanya sebuah pengendalian, komunikasi, kerjasama dengan berbagai perangkat keras, data melalui jaringan internet. Sehingga bisa dikatakan bahwa Internet of Things (IoT) adalah ketika kita menyambungkan sesuatu (things) yang tidak dioperasikan oleh manusia, ke internet (Hardyanto, 2017).

Namun IOT bukan hanya terkait dengan pengendalian perangkat melalui jarak jauh, tapi juga bagaimana berbagi data, memvirtualisasikan segala hal nyata

ke dalam bentuk internet, dan lain-lain. Internet menjadi sebuah penghubung antara sesama mesin secara otomatis. Selain itu juga adanya user yang bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Manfaatnya menggunakan teknologi IoT yaitu pekerjaan yang dilakukan oleh manusia menjadi lebih cepat, muda dan efisien.

BAB III

METODOLOGI PENELITIAN

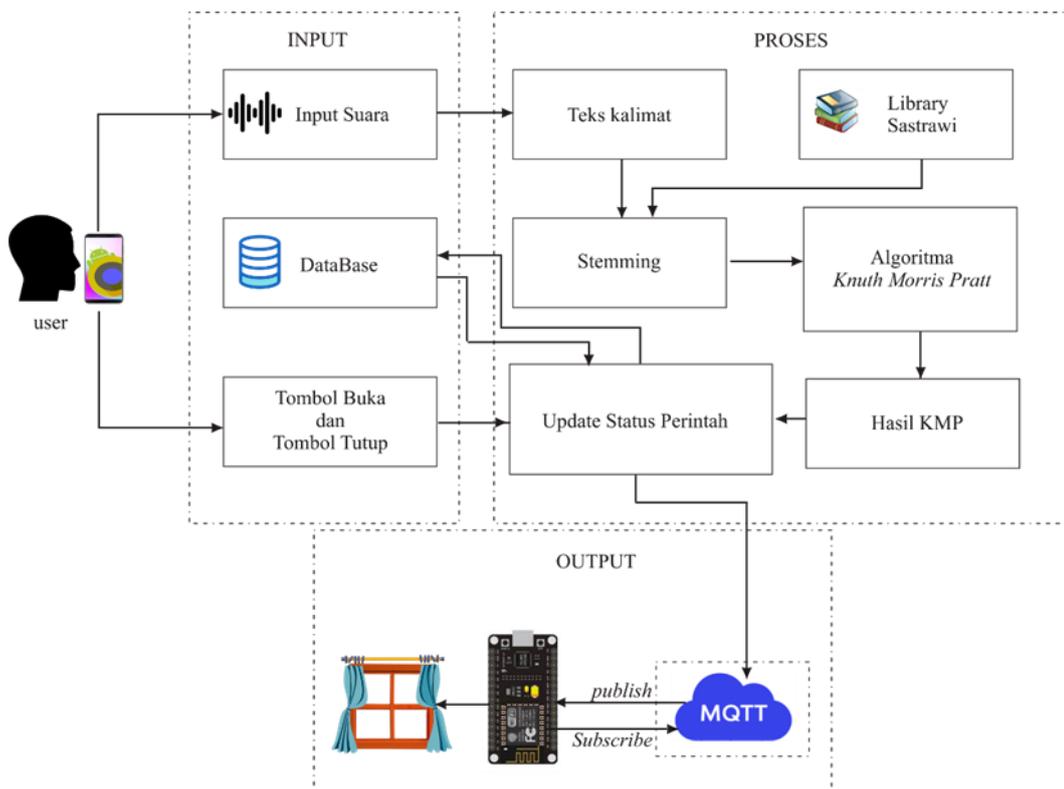
3.1 Desain Dan Implementasi

Dalam penelitian ini terdiri dari *input*, proses, dan *output* yang merupakan pengujian sistem dan perancangan. Pada tahapan *input* merupakan tahapan dalam pengambilan data kemudian data tersebut akan diproses pada bagian proses, *user* dapat melakukan kontrol suara maupun tombol. Jika *user* melakukan dengan kontrol suara maka akan menggunakan fitur *Google SpeechRecognition* yang akan diolah ke dalam bentuk teks dan diproses ke *Stemming* dengan menggunakan *Library Sastrawi* kemudian melakukan proses pencocokan *pattern* dengan metode algoritma *Knuth Morris Pratt*. Setelah *pattern* ditemukan maka hasilnya akan dilanjutkan ke proses mengubah teks perintah menjadi data perintah kemudian membandingkan data tersebut dengan data yang tersimpan di *Database*. Jika data perintah sama dengan data perintah di *Database* maka proses dihentikan, tetapi ketika data tidak sama dengan data perintah di *Database* maka akan dilanjutkan ke proses penyimpanan data ke *Database (Update)* lalu mengirim data ke *MQTT (Publish)* dan selanjutnya data perintah akan mengendalikan *NodeMCU ESP8266* untuk membuka atau menutup gorden.

Jika *user* menggunakan kontrol tombol maka akan dilanjutkan ke proses mengubah tekanan tombol menjadi data perintah kemudian membandingkan data tersebut dengan data yang tersimpan di *Database*. Jika data perintah sama dengan data perintah di *Database* maka proses dihentikan, tetapi ketika data tidak sama

dengan data perintah di *Database* maka akan dilanjutkan ke proses penyimpanan data ke *Database (Update)* lalu mengirim data ke *MQTT (Publish)* dan selanjutnya data perintah akan mengendalikan *NodeMCU ESP8266* untuk membuka atau menutup gorden.

Pengujian sistem dilakukan dengan 2 cara, pertama penguji melakukan pengujian menggunakan algoritma *Knuth Morris Pratt* untuk mencari pencocokan *pattern* dan yang kedua menentukan seberapa akurat sistem yang dibangun dengan menggunakan *confusion matrix*. Desain sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Sistem

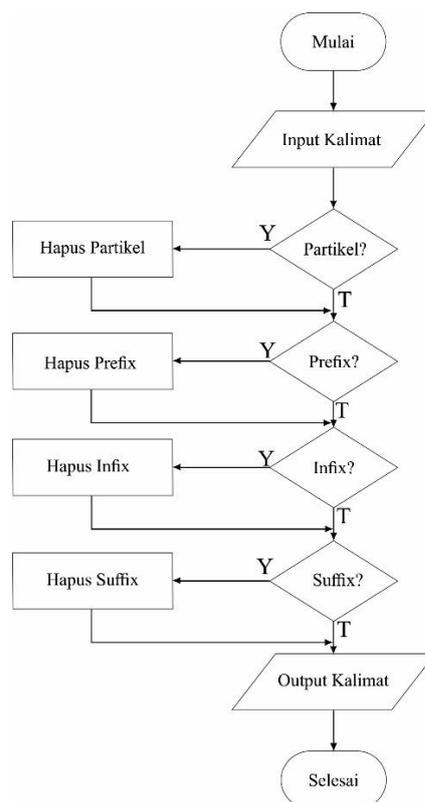
3.1.1 Proses pengambilan Data

Data yang digunakan pada penelitian ini memiliki data primer dan data sekunder yang mana data primer merupakan data dari inputan suara *Speech*

Recognition dengan menggunakan *Google Speech* dalam Bahasa Indonesia dan di konversi menjadi teks. Selanjutnya untuk data sekunder adalah data *pattern* yang dibuat oleh peneliti untuk kata kunci. Berikut data sekunder yang dibuat oleh peneliti : buka gorden, gorden buka, tutup gorden, dan gorden tutup.

3.1.2 Proses *Stemming*

Stemming adalah proses perubahan kalimat atau kata yang berlebihan menjadi kata baku. Dalam proses tersebut sistem akan mengolah menjadi kata-kata dasar dalam Bahasa Indonesia. Proses *Stemming* dengan *Library Sastrawi* dalam pemrograman php dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Stemming

3.1.3 Algoritma Knuth Morris Pratt

Algoritma *knuth morris pratt* merupakan algoritma pencarian *string* dengan cara yang lebih besar pergeserannya dibanding algoritma sebelumnya yaitu algoritma *brute force*. Dalam hal ini algoritma *knuth morris pratt* memiliki tahap *preprocessing* yang dapat bergeser secara efektif dengan adanya komponen LPS (*Longest Proper Prefix Which Is Suffix*). Pada tahap *preprocessing* ini peneliti menggunakan rumus $b(k)$ = jumlah awalan prefiks dari $P[0..k]$ yang merupakan sufiks dari $P[1..k]$. berikut contoh fungsi pinggiran dengan *pattern* “BUKA” pada tabel 3.1.

Tabel 3.1 Fungsi *Pattern* “BUKA”

j	0	1	2	3
P[j]	B	U	K	A
k	0	0	0	0
B(k)	0	0	0	0

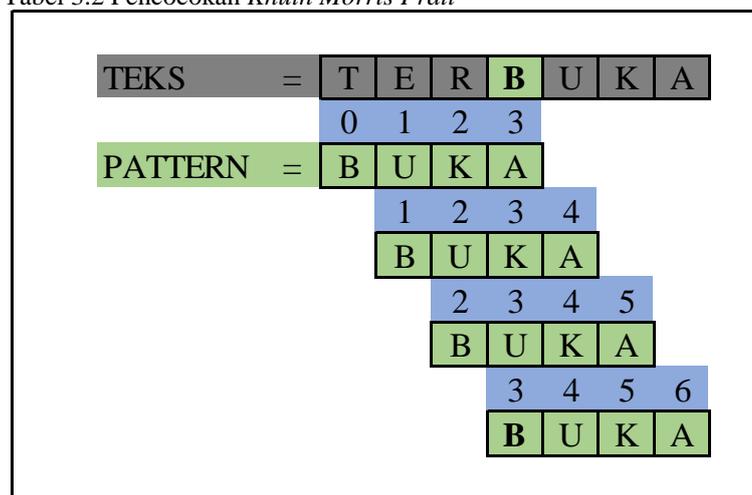
Pada tabel diatas memiliki karakter *Pattern* “BUKA” yang mana diberikan indeks $[0...k]$ yang disimpan pada baris j. Lalu hasil dari baris k merupakan hasil dari -1 pada setiap nilai baris j. Selanjutnya melakukan perhitungan fungsi pinggiran pada baris k. Pada indeks $j=0$ yang menghasilkan nilai $B(k)=0$. Begitu juga pada indeks $j=1$ dan $k=0$ dan menghasilkan $B(k)=0$ karena belum memenuhi definisi dari fungsi pinggiran $B(k)$. Dalam kasus ini nilai yang dihasilkan k_1, k_2, k_3 adalah 0 karena tidak ada karakter yang sama pada *Pattern* begitu juga definisi dari fungsi pinggiran $B(k)_1, B(k)_2, B(k)_3$ bernilai 0. Maka LPS ditemukan dari *Pattern* “BUKA” bernilai 0.

Berikut ini adalah proses pencarian yang dilakukan untuk mencocokkan *Pattern* dengan teks, pencocokan karakter per karakter yang dimulai dari paling kiri dalam *Pattern*. Setiap kali ditemukan ketidakcocokan, maka dilanjutkan ke indeks berikutnya sampai *Pattern* dan teks sama. Berikut contoh kasus pencocokan pada tabel 3.2.

Teks : TERBUKA

Pattern : BUKA

Tabel 3.2 Pencocokan *Knuth Morris Pratt*



Pencocokan di atas merupakan proses pencarian dengan teks “TERBUKA” dan *Pattern* “BUKA”, yang mana ketika kata “BUKA” pada teks ditemukan yaitu dalam indeks 3, sehingga akan menggerakkan Motor untuk membuka gorden, begitu juga ketika yang ditemukan pada teks kata “TUTUP” maka Motor akan bergerak untuk menutup gorden.

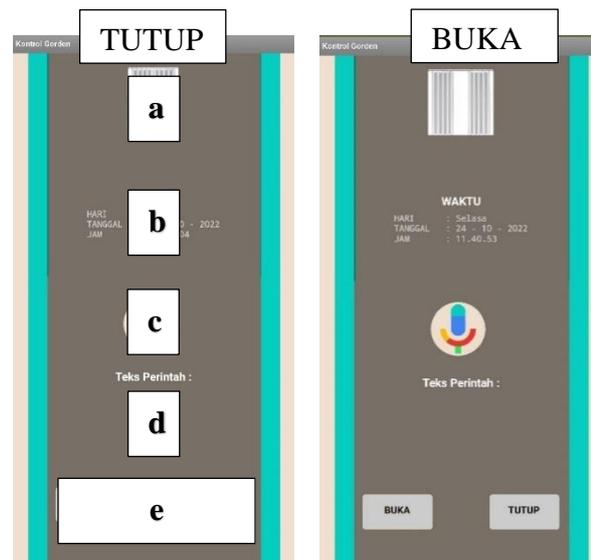
Teks asli : tolong dong bukakan gordennya
Teks Stemming : tolong dong buka gorden
Pattern : buka gorden
LPS Stemming : [0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Ditemukan pada indeks ke : 10
Status : membuka gorden
Waktu perintah terakhir : 2022-12-04 10:05:41

Gambar 3.3 Contoh Hasil Program

Pada gambar 3.3 merupakan tampilan hasil program yang menampilkan hasil dari perintah suara android kemudian suara dijadikan teks yang mana contoh hasil teksnya adalah “tolong dong bukakan gordennya”. Berikutnya proses *stemming* dengan *library* sastrawi yang hasilnya *stemming* adalah “tolong dong buka gorden”, selanjutnya dicocokkan dengan *Pattern* yang sudah ditentukan oleh peneliti, dan menghasilkan *LPS Stemming* [000000 0000 1234 5678910] yang ditemukan pada indeks ke 10 lalu status yang dilakukan oleh motor adalah membuka dan menampilkan waktu saat perintah terakhir.

3.2 Perancangan Aplikasi

Pada bagian perancangan aplikasi ini bertujuan untuk memberikan kemudahan dalam membangun sistem. Sehingga, setiap tampilan pada sistem tersebut dapat dipahami fungsi dan tujuannya. Berikut untuk tampilan rancangan aplikasi dapat dilihat pada Gambar 3.3.



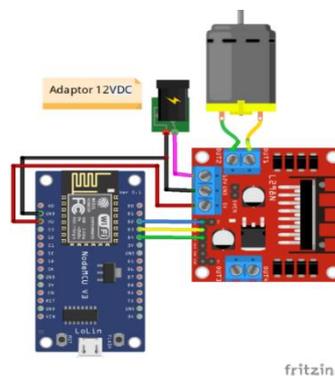
Gambar 3.4 Rancangan Aplikasi

Penjelasan Gambar :

- Gambar status gorden.
- Tampilan waktu saat perintah terakhir.
- Untuk perintah suara menggunakan *Speech Recognizer*.
- Teks perintah suara yang di ucapkan.
- Untuk perintah dengan tombol.

3.3 Desain Komponen

Desain komponen ini merupakan rangkaian komponen elektro yang digunakan oleh peneliti, antara lain: NodeMCU ESP8266, Driver Motor L298N, Motor, konektor dan adaptor. Rangkaian komponen dapat dilihat pada gambar 3.5.



Gambar 3.5 Rangkaian Komponen

3.4 Pengujian Sistem

Pada tahap ini peneliti akan membahas mengenai pengujian sistem dengan menggunakan rumus *confusion matrix* sebagai tujuan untuk mengetahui keakurasian pada sistem. Berikut langkah penjelasan mengenai rumus *confusion matrix* pada Tabel 3.3.

Tabel 3.3 *Confusion matrix*

Kenyataan \ Prediksi	Merespons	Tidak Merespons
	TP	FN
Merespons	TP	FN
Tidak Merespons	FP	TN

Keterangan:

TP (*True Positive*) :Jumlah dari hasil yang berhasil merespons.

TN (*True Negative*) :Jumlah dari hasil yang berhasil tidak merespons.

FP (*False Positive*) :Jumlah dari hasil yang seharusnya tidak merespons tetapi merespons.

FN (*False Negative*) :Jumlah dari hasil yang seharusnya merespons tetapi tidak merespons.

Ada 3 penilaian *confusion matrix* yaitu akurasi, presisi, dan *recall*. Berikut

$$\text{rumusnya : Akurasi} = \frac{TP+TN}{Total} \quad (3.1)$$

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (3.2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3.3)$$

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

Pada bagian ini menjelaskan tentang sistem yang telah dirancang baik dari pembahasan sistem *software* dan *hardware*.

4.1.1 Sistem *Software*

Sistem *software* pada penelitian ini menjelaskan tentang perancangan aplikasi android yang dibuat melalui *MIT App Inventor* menggunakan objek Visual Blok. Fungsi dari aplikasi ini adalah untuk mengendalikan gorden dengan bantuan mikrokontroler yang sudah dirancang.

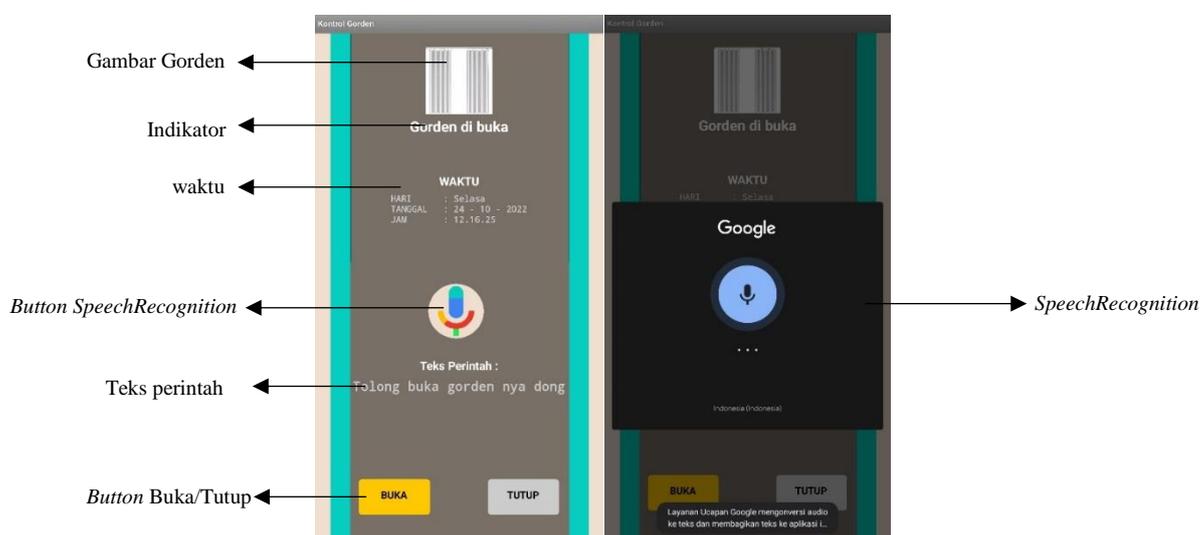
4.1.2 Tampilan Aplikasi

Pada Tampilan aplikasi yang telah dibangun memiliki 2 halaman. Berikut tampilan awal aplikasi dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Awal

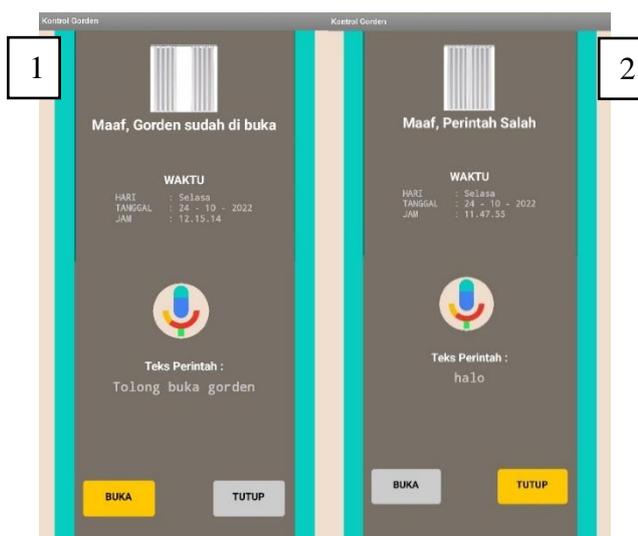
Pada Gambar 4.1 Terdapat *title bar* yang berfungsi sebagai jendela yang menampilkan nama pada halaman tersebut, *title bar* pada halaman ini yaitu home. Gambar logo pada aplikasi ini berfungsi sebagai simbol untuk aplikasi yang dibuat oleh peneliti. *Button* Mulai merupakan tombol yang berfungsi untuk pindah ke *Screen* selanjutnya ketika di klik oleh pengguna. Teks versi pada aplikasi ini merupakan penanda yang sengaja dibuat oleh peneliti untuk memberi informasi kepada pengguna bahwa versi saat ini yaitu versi 1.0.



Gambar 4.2 Kontrol Suara Perintah Buka

Pada Gambar 4.2 Menampilkan halaman kedua, pada halaman ini terdapat *title bar* yang berfungsi sebagai jendela yang menampilkan nama Kontrol Gorden. Gambar gorden yang ada di halaman ini dapat berubah-ubah, ketika pengguna melakukan perintah buka maka menampilkan gambar gorden yang terbuka dan ketika perintah tutup maka menampilkan gambar gorden yang menutup. Indikator merupakan hasil yang dikirim dari web berupa tulisan dan akan ditampilkan pada halaman ini. Waktu akan menampilkan waktu ketika pengguna melakukan perintah. *Button SpeechRecognition* merupakan tombol yang digunakan oleh pengguna

untuk melakukan perintah suara. Teks perintah yang diucapkan oleh pengguna akan ditampilkan pada halaman ini. *Button* buka dapat digunakan untuk membuka gorden secara manual dan *Button* tutup dapat digunakan untuk menutup gorden secara manual.



Gambar 4.3 (1) perintah tidak bisa di ulang (2)perintah salah

1. Peneliti melakukan perintah dengan menggunakan *Speech Recognition* “Tolong buka gorden” ketika gorden sudah terbuka maka akan menampilkan pesan “Maaf, Gorden sudah di buka”.
2. Peneliti melakukan perintah dengan menggunakan *Speech Recognition* “halo” maka akan menampilkan pesan “Maaf, perintah salah”

4.1.3 Desain Program Visual Blok

Desain program Visual Blok pada penelitian ini menjelaskan tentang rancangan sistem dari aplikasi Kontrol Gorden atau perintah – perintah yang terletak pada objek visual blok di sistem aplikasi Kontrol Gorden.

1. Visual Blok *Screen 1*

a. Visual Blok *Button1.Click*



Gambar 4.4 Proses pindah ke *screen2*

Pada Gambar 4.4, Terdapat Gambar Visual Blok *Button1.Click* yang mana ketika tombol di tekan maka akan diproses ke *Screen2*.

2. Visual Blok *Screen 2*

a. Visual Blok *SpeechRecognizer1.BeforeGettingText*



Gambar 4.5 Sebelum mendapatkan teks

Pada Gambar 4.5, Terdapat Gambar Visual Blok *SpeechRecognizer1.BeforeGettingText* menjelaskan ketika pengguna sebelum mengucapkan atau memerintah apa pun maka tampilan pada Label Teks itu kosong.

b. Visual Blok *Voice.Click*



Gambar 4.6 Ketika mengklik satu kali tombol *Voice*

Pada Gambar 4.6, Terdapat Gambar Visual Blok *Voice.Click* yang ketika pengguna menekan tombol *Voice* tanpa menahannya maka akan keluar pemberitahuan “Tekan dan tahan untuk bicara!”

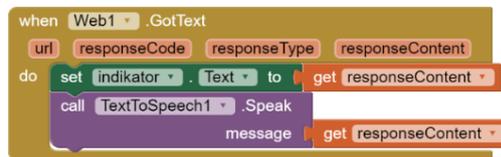
c. Visual Blok *Voice.LongClick*



Gambar 4.7 Ketika menahan tombol *Voice*

Pada Gambar 4.7, Terdapat Gambar Visual Blok *Voice.LongClick* yang mana ketika pengguna menekan tombol *Voice* dan menahannya maka bisa melanjutkan untuk mengucapkan kalimat perintah.

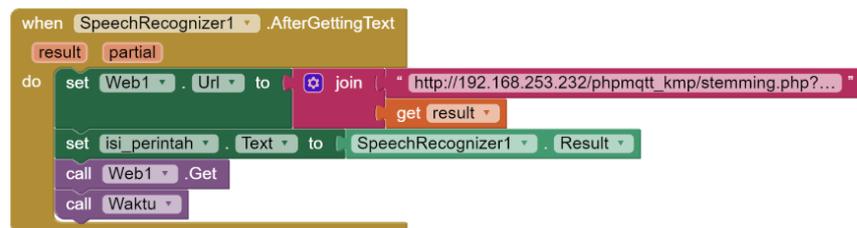
d. Visual Blok *Web1.GotText*



Gambar 4.8 Mendapatkan teks dari web

Pada Gambar 4.8, Terdapat Gambar Visual Blok *Web1.GotText* yang mana jika mendapatkan teks dari web akan di tampilkan ke indikator dan mengeluarkan pesan suara.

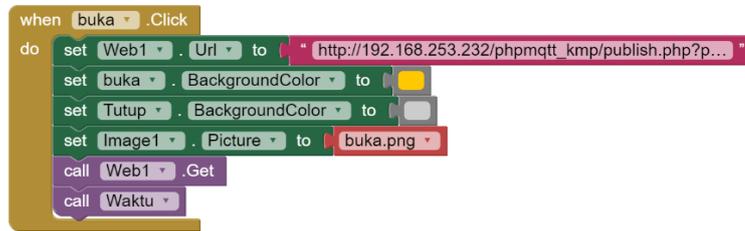
e. Visual Blok *SpeechRecognizer1.AfterGettingText*



Gambar 4.9 Sesudah mendapatkan teks

Pada Gambar 4.9, Terdapat Gambar Visual Blok *SpeechRecognizer1.AfterGettingText* yang mana sesudah mendapatkan teks melalui suara maka teks dikirim dengan cara mengaktifkan *Web1* yang selanjutnya dilakukan *stemming* pada alamat web tersebut.

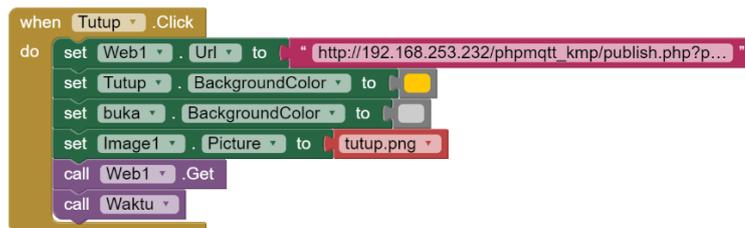
f. Visual Blok buka.Click



Gambar 4.10 Ketika Tombol buka di Klik

Pada Gambar 4.10, Terdapat Gambar Visual Blok *buka.Click* yang mana saat tombol buka di klik akan mengaktifkan web1 selanjutnya berakibat membuka gorden.

g. Visual Blok Tutup.Click



Gambar 4.11 Ketika Tombol Tutup di Klik

Pada Gambar 4.11, Terdapat Gambar Visual Blok *Tutup.Click* yang mana saat tombol Tutup di klik akan mengaktifkan web1 selanjutnya berakibat menutup gorden. Di mana *link* pada web1.Url mengarah ke *publish* yang tersimpan di dalam *file* berformat *php* dan akan diproses ke *database* untuk pengecekan data. Apakah perintah sama dengan data yang ada di *database*, kalau tidak sama maka server MQTT akan berjalan dan mengirim pesan 1 atau 0 ke *NodeMCU ESP8266*. Tetapi ketika data yang ada di *database* sama dengan isi perintah maka pesan akan dikirim kembali ke *Smartphone* untuk memberi tahu (keluar pesan) “Maaf, Gorden sudah di buka” begitu juga kalau perintah tutup akan keluar notifikasi “Maaf, Gorden sudah di tutup”.

```

Local server
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-125-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Tue Nov 29 21:33:36 2022 from 127.0.0.1
root@ubuntu:~# cd /www/wwwroot/kontrolgorden.online/tata/mqtt
root@ubuntu:/www/wwwroot/kontrolgorden.online/tata/mqtt# php subscribe.php
Wed, 30 Nov 2022 18:31:26 +0800: not had something in a while so ping
Wed, 30 Nov 2022 18:31:26 +0800: ping sent
Wed, 30 Nov 2022 18:31:26 +0800: Received CMD: 13 (PINGRESP)
Wed, 30 Nov 2022 18:31:26 +0800: Fetching: 0 bytes
Wed, 30 Nov 2022 18:31:35 +0800: Received CMD: 3 (PUBLISH)
Wed, 30 Nov 2022 18:31:35 +0800: Fetching: 23 bytes
Msg Recieved: Wed, 30 Nov 2022 18:31:35 +0800
Topic: /mqtt/IOT2022/gorden
pesan: 0
Wed, 30 Nov 2022 18:31:37 +0800: not had something in a while so ping
Wed, 30 Nov 2022 18:31:37 +0800: ping sent
Wed, 30 Nov 2022 18:31:37 +0800: Received CMD: 13 (PINGRESP)
Wed, 30 Nov 2022 18:31:37 +0800: Fetching: 0 bytes
Wed, 30 Nov 2022 18:31:40 +0800: Received CMD: 3 (PUBLISH)
Wed, 30 Nov 2022 18:31:40 +0800: Fetching: 23 bytes
Msg Recieved: Wed, 30 Nov 2022 18:31:40 +0800
Topic: /mqtt/IOT2022/gorden
pesan: 1

```

Gambar 4.12 Pesan yang diterima oleh *Subscription*

Pada gambar 4.12 merupakan pesan “0” yang mana untuk perintah tutup dan pesan “1” untuk perintah buka. Pesan tersebut diterima oleh *Subscription* yang diterima oleh *publish* dengan *topic* “mqtt/IOT2022/gorden”.

4.2 Hasil Pengujian *Knuth Morris Pratt*

Penelitian ini melakukan hasil pengujian kalimat teks pada aplikasi Kontrol Gorden. Hal ini dilakukan untuk mengetahui *LPS Stemming* pada algoritma *knuth morris pratt*.

4.2.1 Proses *PreKMP*

Proses *preKMP* dilakukan untuk mendapatkan nilai *LPS* yang mana nilai *LPS* melakukan pencarian dari indeks paling kiri pada *pattern*. Hal ini dilakukan pencacahan masing-masing karakter pada *pattern*. dimulai dari indeks kiri sampai paling kanan. Sehingga nilai *LPS* dapat ditemukan seperti pada gambar 4.13.

PATTERN : buka gorden
LPS : [000000000000]
PATTERN : tutup gorden
LPS : [001200000000]
PATTERN : gorden buka
LPS : [000000000000]
PATTERN : gorden tutup
LPS : [000000000000]

Gambar 4.13 Nilai LPS pada *Pattern*

Pada gambar 4.13 merupakan hasil dari nilai *LPS* pada *pattern* (buka gorden, tutup gorden, gorden buka, gorden tutup).

4.2.2 Proses Pengujian *Knuth Morris Pratt*

Pada proses pengujian *Knuth Morris Pratt* ini dilakukan setelah mendapatkan nilai *LPS* pada proses sebelumnya. Algoritma *Knuth Morris Pratt* melakukan pengujian dengan pergeseran *pattern* untuk mencocokkan teks. Apabila dalam proses pengujian benar dan data tidak sama dengan data yang ada di *database* maka akan mengirimkan perintah 1 atau 0 pada *subscribe* MQTT yang dilakukan dari *Alibaba Cloud ECS* dengan kontrol panel yang bernama *aanel*.

4.2.3 Pengujian I Algoritma *Knuth Morris Pratt*

Sistem kontrol gorden ini menggunakan algoritma *knuth morris pratt* untuk mengimplementasikan pengujian kalimat perintah teks versi gagap kemudian dicocokkan dengan *pattern*. Berikut contoh pengujian versi gagap yang dilakukan oleh peneliti dapat dilihat pada tabel 4.1 dan 4.2.

Tabel 4.1 *Pattern* “buka gorden(versi gagap)”

Pattern "buka gorden"				
No.	Teks Stemming	Pattern	Durasi (ms)	LPS Stemming
1	tolong bubu bubuka gorden	buka gorden	482	[000000 1212 121234 5678910]
2	bububu buka gorden	buka gorden	748	[121212 1234 5678910]
3	bu u u u u u u k k k buka gorden	buka gorden	514	[12 0 0 0 0 0 0 0 0 0 1234 5678910]
4	tototolong buka gorden	buka gorden	462	[101010100000 12345 67891011]
5	sesesekakarang bubuka gorden	buka gorden	286	[00000000000000 121234 5678910]

Tabel 4.2 *Pattern* “tutup gorden(versi gagap)”

Pattern "tutup gorden"				
No.	Teks Stemming	Pattern	Durasi (ms)	LPS Stemming
1	tolong tututu tutup gorden	tutup gorden	440	[100000 123412 12345 67891011]
2	tututu tutup gorden	tutup gorden	225	[123412 12345 67891011]
3	tu u u u u u u p p p tutup gorden	tutup gorden	113	[12 0 0 0 0 0 0 0 0 0 12345 67891011]
4	tototolong tutup gorden	tutup gorden	2	[000000000000 1234 5678910]
5	sesesekakarang tutututup gorden	tutup gorden	205	[00000000000000 123412345 567891011]

Dari pengujian pada tabel 4.1 dan 4.2 merupakan contoh kalimat teks perintah versi gagap yang dibuat oleh peneliti. Selanjutnya ada 10 contoh kalimat teks perintah yang salah, dapat dilihat pada tabel 4.3.

Tabel 4.3 Perintah Salah

Perintah salah		
No.	Teks asli	Teks Stemming
1	b b u u k a k a a a gorden	b b u u k a k a a a gorden
2	t t u u t u p gorden	t t u u t u p gorden
3	b b u k a k a a a gor den	b b u k a k a a a gor den
4	t t u t u t u p gor den	t t u t u t u p gor den
5	buka gor gor gorden	buka gor gor gorden
6	tutup gor gorden	tutup gor gorden
7	gor den bu ka	gor den bu ka
8	gor den tu tup	gor den tu tup
9	g o go gor den bububuka	g o go gor den bububuka
10	g o go gor den tutututup	g o go gor den tutututup

4.2.4 Pengujian II Algoritma *Knuth Morris Pratt*

Sistem kontrol gorden ini menggunakan algoritma *knuth morris pratt* untuk mengimplementasikan pengujian kalimat perintah teks yang dicocokkan dengan

pattern. Pengujian yang dilakukan dari 5 orang terdapat 20 kalimat teks yang berbeda-beda dari 4 *pattern*. Yang mana per orang menggunakan 1 *pattern* buka gorden, 1 *pattern* tutup gorden, 1 *pattern* gorden buka dan 1 *pattern* gorden tutup. Berikut pengujiannya dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil pengujian dari orang ke-1

Perintah dari orang 1				
Fajri Aldi				
No.	Teks Asli	Teks Stemming	Pattern	Durasi (ms)
1	tolong bukalah gordennya	tolong buka gorden	buka gorden	930
2	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	740
3	gordennya buka dong	gorden buka dong	gorden buka	62
4	gordennya tutup dong	gorden tutup dong	gorden tutup	409
Rata-Rata Durasi (ms)				535

Pada Tabel 4.4 merupakan hasil pengujian dari orang 1 dengan rata-rata durasi 535ms. Setelah itu dilanjutkan oleh pengujian orang ke 2 yang dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil pengujian dari orang ke-2

Perintah dari orang 2				
Rizqa Nur Aziza				
No.	Teks Asli	Teks Stemming	Pattern	Durasi (ms)
1	membukalah gorden	buka gorden	buka gorden	679
2	menutuplah gordenn	tutup gorden	tutup gorden	449
3	gorden membuka dong	gorden buka dong	gorden buka	658
4	gorden menutup dong	gorden tutup dong	gorden tutup	206
Rata-Rata Durasi (ms)				498

Pada Tabel 4.5 merupakan hasil pengujian dari orang 2 dengan rata-rata durasi 498ms. Setelah itu dilanjutkan oleh pengujian orang ke 3 yang dapat dilihat pada tabel 4.6.

Tabel 4.6 Hasil pengujian dari orang ke-3

Perintah dari orang 3				
Faldin Aulia Shafi				
No.	Teks Asli	Teks Stemming	Pattern	Durasi (ms)
1	tolong bukakan gorden	tolong buka gorden	buka gorden	34
2	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	836
3	gorden bukalah	gorden buka dong	gorden buka	543
4	gorden tutuplah	gorden tutup dong	gorden tutup	954
Rata-Rata Durasi (ms)				592

Pada Tabel 4.6 merupakan hasil pengujian dari orang 3 dengan rata-rata durasi 592ms. Setelah itu dilanjutkan oleh pengujian orang ke 4 yang dapat dilihat pada tabel 4.7.

Tabel 4.7 Hasil pengujian dari orang ke-4

Perintah dari orang 4				
Hasbi Ashiddiqi				
No.	Teks Asli	Teks Stemming	Pattern	Durasi (ms)
1	sekarang membuka gorden	sekarang buka gorden	buka gorden	622
2	sekarang menutup gorden	sekarang tutup gorden	tutup gorden	169
3	tolong dong gordennya buka	tolong dong gorden buka	gorden buka	650
4	tolong dong gordennya tutup	tolong dong gorden tutup	gorden tutup	852
Rata-Rata Durasi (ms)				573

Pada Tabel 4.7 merupakan hasil pengujian dari orang 4 dengan rata-rata durasi 573ms. Setelah itu dilanjutkan oleh pengujian orang ke 5 yang dapat dilihat pada tabel 4.8.

Tabel 4.8 Hasil pengujian dari orang ke-5

Perintah dari orang 5				
Muhammad Lukmanul Khakim				
No.	Teks Asli	Teks Stemming	Pattern	Durasi (ms)
1	bukakan gorden	buka gorden	buka gorden	442
2	tutupkan gorden	tutup gorden	tutup gorden	614
3	gordennya dibuka	gorden buka	gorden buka	777
4	gordennya ditutup	gorden tutup	gorden tutup	176
Rata-Rata Durasi (ms)				502

Pada Tabel 4.8 merupakan hasil pengujian dari orang 5 dengan rata-rata durasi 502ms. setelah semuanya mendapatkan nilai durasi maka selanjutnya dijumlahkan semua rata-rata durasinya kemudian di bagi dan menghasilkan rata-rata durasi dari keseluruhan 540ms.

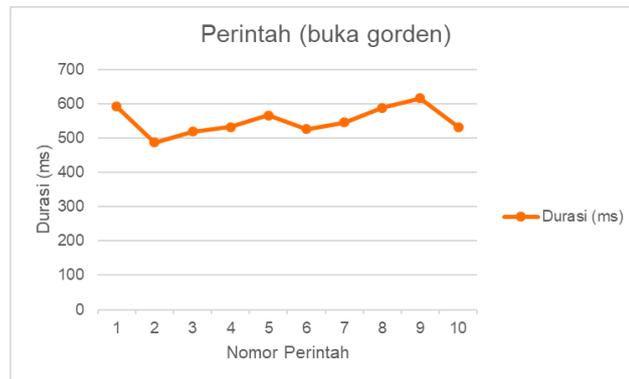
4.2.5 Pengujian III Algoritma *Knuth Morris Pratt*

Sistem kontrol gorden ini menggunakan algoritma *knuth morris pratt* untuk mengimplementasikan pengujian kalimat perintah teks kemudian dicocokkan dengan *pattern*. Pengujian yang dilakukan oleh peneliti terdapat 40 kalimat teks dengan 4 *pattern*. Perhitungan akurasi, presisi dan *recall* pada sistem kontrol gorden ini menggunakan rumus *Confusion Matrix* berdasarkan rumus pada bab 3. Berikut pengujian menggunakan kalimat perintah “buka gorden” pada tabel 4.9 .

Tabel 4.9 Hasil pengujian “buka gorden”

Perintah (buka gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	buka gorden	buka gorden	buka gorden	592
2	buka gorden	buka gorden	buka gorden	487
3	buka gorden	buka gorden	buka gorden	519
4	buka gorden	buka gorden	buka gorden	532
5	buka gorden	buka gorden	buka gorden	566
6	buka gorden	buka gorden	buka gorden	526
7	buka gorden	buka gorden	buka gorden	546
8	buka gorden	buka gorden	buka gorden	588
9	buka gorden	buka gorden	buka gorden	616
10	buka gorden	buka gorden	buka gorden	532
Rata-rata Durasi (ms)				550
Indeks	[0 1 2 3 4 5 6 7 8 9]			
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]			
Pergeseran Indeks ke	0			

Dari pengujian pada tabel 4.9 didapatkan grafik durasi yang dapat dilihat pada gambar 4.14.



Gambar 4.14 Grafik durasi “buka gorden”

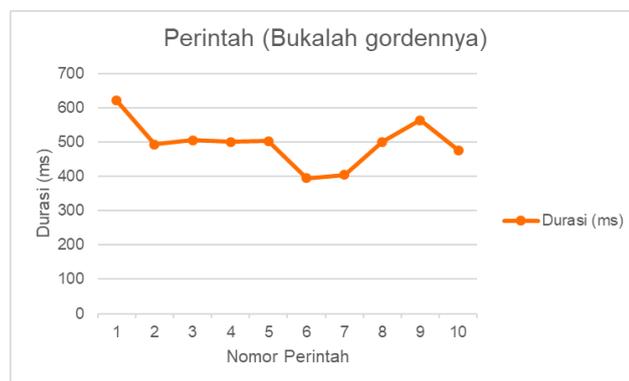
Setelah dilakukan pengujian kalimat perintah “buka gorden” sebanyak 10 kali memiliki hasil pengujian akurasi 100% presisi 100% *recall* 100% dengan rata-rata durasi 550ms. Setelah itu dilakukan pengujian kalimat perintah “bukalah gordennya” yang dapat dilihat pada tabel 4.10.

Tabel 4.10 Hasil pengujian “bukalah gordennya”

Perintah (Bukalah gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Bukalah gordennya	Buka gorden	buka gorden	621
2	Bukalah gordennya	Buka gorden	buka gorden	493
3	Bukalah gordennya	Buka gorden	buka gorden	505
4	Bukalah gordennya	Buka gorden	buka gorden	500
5	Bukalah gordennya	Buka gorden	buka gorden	503
6	Bukalah gordennya	Buka gorden	buka gorden	395
7	Bukalah gordennya	Buka gorden	buka gorden	404
8	Bukalah gordennya	Buka gorden	buka gorden	501
9	Bukalah gordennya	Buka gorden	buka gorden	564
10	Bukalah gordennya	Buka gorden	buka gorden	476
Rata-rata Durasi (ms)				496

Indeks	[0 1 2 3 4 5 6 7 8 9]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	0

Dari pengujian pada tabel 4.10 didapatkan grafik durasi yang dapat dilihat pada gambar 4.15.



Gambar 4.15 Grafik durasi “bukalah gordennya”

Setelah dilakukan pengujian kalimat perintah “bukalah gordennya” sebanyak 10 kali memiliki hasil pengujian akurasi 100% presisi 100% *recall* 100% dengan rata-rata durasi 496ms. Setelah itu dilakukan pengujian kalimat perintah “sekarang buka gordennya” yang dapat dilihat pada tabel 4.11.

Tabel 4.11 Hasil pengujian “sekarang buka gordennya”

Perintah (sekarang buka gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang buka gordennya	sekarang buka gorden	buka gorden	376
2	sekarang buka gordennya	sekarang buka gorden	buka gorden	536
3	sekarang buka gordennya	sekarang buka gorden	buka gorden	543
4	sekarang buka gordennya	sekarang buka gorden	buka gorden	544
5	sekarang buka gordennya	sekarang buka gorden	buka gorden	510
6	sekarang buka gordennya	sekarang buka gorden	buka gorden	456
7	sekarang buka gordennya	sekarang buka gorden	buka gorden	546
8	sekarang buka gordennya	sekarang buka gorden	buka gorden	513
9	sekarang buka gordennya	sekarang buka gorden	buka gorden	414
10	sekarang buka gordennya	sekarang buka gorden	buka gorden	419
Rata-rata Durasi (ms)				486

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	8

Dari pengujian pada tabel 4.11 didapatkan grafik durasi yang dapat dilihat pada gambar 4.16.



Gambar 4.16 Grafik durasi “sekarang buka gordennya”

Setelah dilakukan pengujian kalimat perintah “sekarang buka gordennya” sebanyak 10 kali memiliki hasil pengujian akurasi 100% presisi 100% *recall* 100% dengan rata-rata durasi 486ms. Setelah itu dilakukan pengujian kalimat perintah “sekarang buka gorden dong” yang dapat dilihat pada tabel 4.12.

Tabel 4.12 Hasil pengujian “sekarang buka gorden dong”

Perintah (sekarang buka gorden dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	556
2	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	430
3	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	533
4	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	531
5	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	513
6	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	553
7	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	454
8	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	389
9	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	603
10	sekarang buka gorden dong	sekarang buka gorden dong	buka gorden	480
Rata-rata Durasi (ms)				504

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 0 0 0 0]
Pergeseran Indeks ke	8

Dari pengujian pada tabel 4.12 didapatkan grafik durasi yang dapat dilihat pada gambar 4.17.



Gambar 4.17 Grafik durasi “sekarang buka gorden dong”

Setelah dilakukan pengujian kalimat perintah “sekarang buka gorden dong” sebanyak 10 kali memiliki hasil pengujian akurasi 100% presisi 100% *recall* 100% dengan rata-rata durasi 504ms. Setelah itu dilakukan pengujian kalimat perintah “tolong dong buka gordennya” yang dapat dilihat pada tabel 4.13.

Tabel 4.13 Hasil pengujian “tolong dong buka gordennya”

Perintah (Tolong dong buka gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	887
2	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	110
3	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	127
4	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	472
5	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	407
6	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	553
7	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	489
8	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	460
9	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	456
10	Tolong dong buka gordennya	Tolong dong buka gorden	buka gorden	364
Rata-rata Durasi (ms)				433

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]
LPS Stemming	[0 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	10

Dari pengujian pada tabel 4.13 didapatkan grafik durasi yang dapat dilihat pada gambar 4.18.



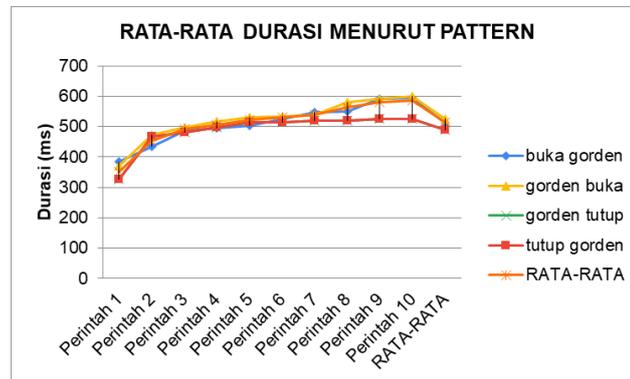
Gambar 4.18 Grafik durasi “tolong dong buka gordennya”

Setelah dilakukan pengujian kalimat perintah “tolong dong buka gordennya” sebanyak 10 kali memiliki hasil pengujian akurasi 100% presisi 100% *recall* 100% dengan rata-rata durasi 433ms. Hasil pengujian berikutnya dapat dilihat pada lampiran yang sudah cantumkan oleh peneliti pada bagian akhir. Berikut hasil rata-rata durasi dari semua kalimat perintah dan semua *pattern* (buka gorden, tutup gorden, gorden buka dan gorden tutup) dapat dilihat pada tabel 4.14.

Tabel 4.14 Hasil rata-rata durasi dari semua *pattern*

RATA-RATA DURASI MENURUT PATTERN					
PERINTAH	RATA-RATA DURASI PATTERN (ms)				
	buka gorden	gorden buka	gorden tutup	tutup gorden	RATA-RATA
Perintah 1	385	370	325	326	352
Perintah 2	433	472	440	468	453
Perintah 3	486	497	503	480	492
Perintah 4	496	518	508	498	505
Perintah 5	504	532	536	514	522
Perintah 6	526	533	544	515	530
Perintah 7	547	540	553	521	540
Perintah 8	550	581	606	521	565
Perintah 9	592	591	614	526	581
Perintah 10	593	601	627	526	587
RATA-RATA	511	524	526	490	513

Dari pengujian pada tabel 4.14 didapatkan grafik durasi yang dapat dilihat pada gambar 4.19.



Gambar 4.19 Grafik rata-rata durasi dari semua *pattern*

Pada Gambar 4.19 merupakan grafik rata-rata durasi dari semua kalimat perintah dan semua *pattern*. Yang mana nilai rata-rata dari perintah 1 adalah 352ms, perintah 2 adalah 453ms, perintah 3 adalah 492ms, perintah 4 adalah 505ms, perintah 5 adalah 522ms, perintah 6 adalah 530ms, perintah 7 adalah 540ms, perintah 8 adalah 565ms, perintah 9 adalah 581ms, perintah 10 adalah 587ms, selanjutnya nilai rata-rata dari *pattern* “buka gorden” adalah 511ms, rata-rata dari *pattern* “gorden buka” adalah 524ms, rata-rata dari *pattern* “tutup gorden” adalah 490ms, dan rata-rata dari *pattern* “gorden tutup” adalah 526ms. Setelah semuanya mendapatkan nilai kemudian di cari rata-rata dari keseluruhan yang hasilnya adalah 513ms.

4.3 Hasil Pengujian Akurasi

Penelitian ini melakukan hasil pengujian sistem pada aplikasi Kontrol Gorden. Hal ini dilakukan agar mengetahui tingkat akurasi pada sistem yang telah dirancang menggunakan rumus *Confusion matrix*.

4.3.1 Pengujian Sistem dengan *Confusion Matrix*

Pada pengujian sistem menggunakan *Confusion matrix* untuk mengetahui tingkat keakurasian aplikasi Kontrol Gorden Berbasis *Internet Of Things* dengan perintah suara. Berikut hasil dari uji coba dengan menggunakan rumus *confusion matrix* dengan *Pattern* buka gorden, tutup gorden, gorden buka, gorden tutup.

Tabel 4.15 *Pattern* buka gorden

No	Kata	Pattern	Prediksi		Kenyataan		Keterangan
			Merespon	Tdk Merespon	Merespon	Tdk Merespon	
1	buka gorden	buka gorden	√		√		TP
2	bukalah gordennya	buka gorden	√		√		TP
3	sekarang buka gordennya	buka gorden	√		√		TP
4	sekarang buka gorden dong	buka gorden	√		√		TP
5	tolong dong buka gordennya	buka gorden	√		√		TP
6	tolong dibuka gordennya	buka gorden	√		√		TP
7	tolong bukakan gorden	buka gorden	√		√		TP
8	tolong bukalah gordennya	buka gorden	√		√		TP
9	membukalah gorden	buka gorden	√		√		TP
10	sekarang membuka gorden dong	buka gorden	√		√		TP

Pada tabel 4.15 merupakan pengujian kata dari *Pattern* buka gorden yang diperoleh hasil *True Positive* (TP), yang mana prediksi dan kenyataan adalah merespons.

Tabel 4.16 *Pattern* tutup gorden

No	Kata	Pattern	Prediksi		Kenyataan		Keterangan
			Merespon	Tdk Merespon	Merespon	Tdk Merespon	
1	tutup gorden	tutup gorden	√		√		TP
2	tutuplah gordennya	tutup gorden	√		√		TP
3	sekarang tutup gordennya	tutup gorden	√		√		TP
4	sekarang tutup gorden dong	tutup gorden	√		√		TP
5	tolong dong tutup gordennya	tutup gorden	√		√		TP
6	tolong ditutup gordennya	tutup gorden	√		√		TP
7	tolong tutupkan gorden	tutup gorden	√		√		TP
8	tolong tutuplah gordennya	tutup gorden	√		√		TP
9	menutuplah gorden	tutup gorden	√		√		TP
10	sekarang menutup gorden dong	tutup gorden	√		√		TP

Pada tabel 4.16 merupakan pengujian kata dari *Pattern* tutup gorden yang diperoleh hasil *True Positive* (TP), yang mana prediksi dan kenyataan adalah merespons.

Tabel 4.17 *Pattern* gorden buka

No	Kata	Pattern	Prediksi		Kenyataan		Keterangan
			Merespon	Tdk Merespon	Merespon	Tdk Merespon	
1	gorden buka	gorden buka	√		√		TP
2	gorden bukalah	gorden buka	√		√		TP
3	sekarang gordennya buka	gorden buka	√		√		TP
4	sekarang gorden dibuka dong	gorden buka	√		√		TP
5	tolong dong gordennya buka	gorden buka	√		√		TP
6	tolong gordennya dibukakan	gorden buka	√		√		TP
7	tolong gorden bukakan	gorden buka	√		√		TP
8	tolong gordennya bukalah	gorden buka	√		√		TP
9	gorden membukalah	gorden buka	√		√		TP
10	sekarang gorden membuka dong	gorden buka	√		√		TP

Pada tabel 4.17 merupakan pengujian kata dari *Pattern* gorden buka yang diperoleh hasil *True Positive* (TP), yang mana prediksi dan kenyataan adalah merespons.

Tabel 4.18 *Pattern* gorden tutup

No	Kata	Pattern	Prediksi		Kenyataan		Keterangan
			Merespon	Tdk Merespon	Merespon	Tdk Merespon	
1	gorden tutup	gorden tutup	√		√		TP
2	gorden tutuplah	gorden tutup	√		√		TP
3	sekarang gordennya tutup	gorden tutup	√		√		TP
4	sekarang gorden ditutup dong	gorden tutup	√		√		TP
5	tolong dong gordennya tutup	gorden tutup	√		√		TP
6	tolong gordennya ditutupkan	gorden tutup	√		√		TP
7	tolong gorden tutupkan	gorden tutup	√		√		TP
8	tolong gordennya tutuplah	gorden tutup	√		√		TP
9	gorden menutuplah	gorden tutup	√		√		TP
10	sekarang gorden menutup dong	gorden tutup	√		√		TP

Pada tabel 4.18 merupakan pengujian kata dari *Pattern* gorden tutup yang diperoleh hasil *True Positive* (TP), yang mana prediksi dan kenyataan adalah merespons. Selanjutnya ada 10 teks yang bukan perintah dapat dilihat pada tabel 4.19.

Tabel 4.19 Bukan Perintah

No	Kata	Pattern	Prediksi		Kenyataan		Keterangan
			Merespon	Tdk Merespon	Merespon	Tdk Merespon	
1	Hai	Bukan Perintah		√		√	TN
2	Halo	Bukan Perintah		√		√	TN
3	Dimana	Bukan Perintah		√		√	TN
4	Iya	Bukan Perintah		√		√	TN
5	Tidak	Bukan Perintah		√		√	TN
6	Bagaimana	Bukan Perintah		√		√	TN
7	Siapa	Bukan Perintah		√		√	TN
8	Saya	Bukan Perintah		√		√	TN
9	Kamu	Bukan Perintah		√		√	TN
10	Sedang Apa	Bukan Perintah		√		√	TN

Pada tabel 4.19 merupakan pengujian dari kata yang bukan perintah diperoleh hasil *True Negative* (TN), yang mana prediksi dan kenyataan adalah tidak merespons. Berikut untuk nilai hasil keseluruhan respons dalam tabel 4.20.

Tabel 4.20 Hasil *Confusion Matrix*

		Prediksi	
		Merespon	Tidak Merespon
Kenyataan	Merespon	40	0
	Tidak Merespon	0	10

Keterangan :

Tabel 4.21 Keterangan *Confusion Matrix*

TP	40
FP	0
FN	0
TN	10
TOTAL	50

Setelah melakukan uji coba menggunakan perhitungan *Confusion matrix* maka selanjutnya dilakukan pengujian akurasi, pengujian presisi, dan pengujian *recall*

4.3.2 Pengujian Akurasi

Pada pengujian akurasi peneliti menggunakan rumus $\frac{TP+TN}{Total}$. Dari pengujian akurasi dengan rumus tersebut diperoleh hasil :

$$\text{Akurasi} = \frac{TP+TN}{\text{Total}} = \frac{40+10}{50} = 1 = 100\%$$

4.3.3 Pengujian Presisi

Pada pengujian presisi peneliti menggunakan rumus $\frac{TP}{TP+FP}$. Dari pengujian presisi dengan rumus tersebut diperoleh hasil :

$$\text{Presisi} = \frac{TP}{TP+FP} = \frac{40}{40+0} = 1 = 100\%$$

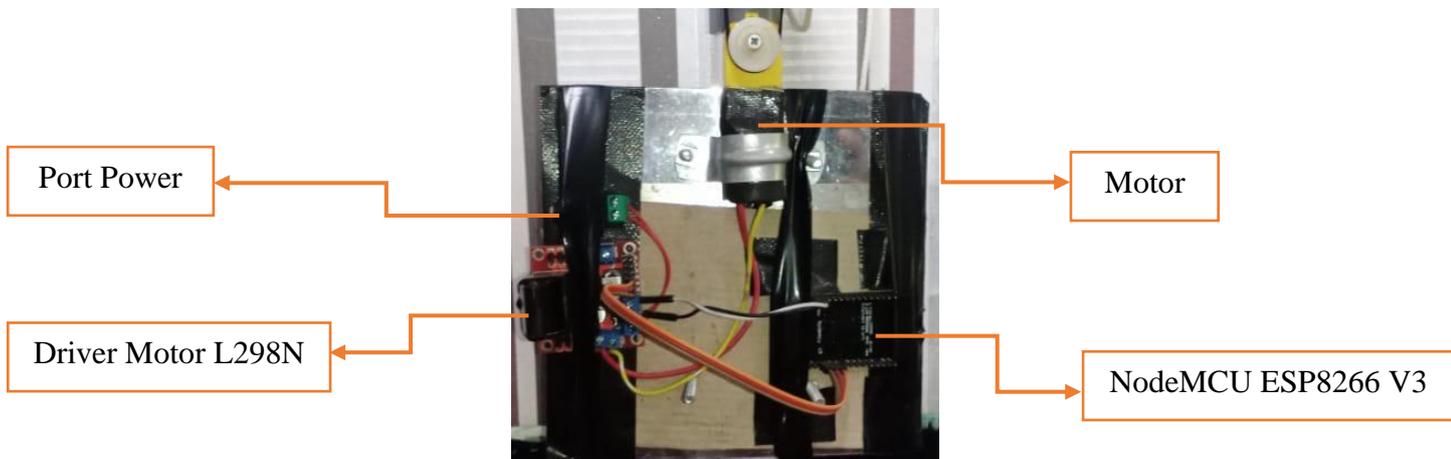
4.3.4 Pengujian Recall

Pada pengujian *recall* peneliti menggunakan rumus $\frac{TP}{TP+FN}$. Dari pengujian *recall* dengan rumus tersebut diperoleh hasil :

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{40}{40+0} = 1 = 100\%$$

4.4 Sistem Hardware

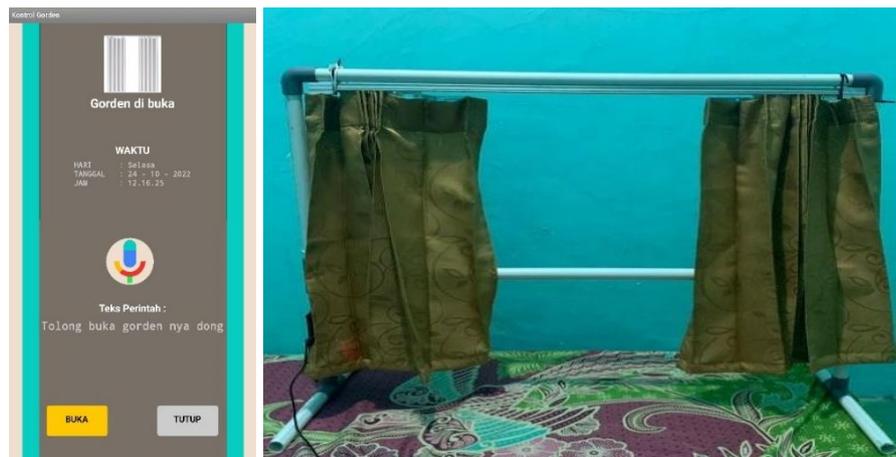
Penelitian ini menggunakan *mikrokontroler NodeMCU ESP8266* yang menjadi alat utama untuk menjalankan *hardware* yang akan mengirim perintah pada *Driver Motor L298N*. Suara dari aplikasi yang berhasil akan mengirimkan data menuju *NodeMCU ESP8266* untuk dikirimkan kepada *Driver Motor L298N*. *NodeMCU ESP8266* ini memanfaatkan komunikasi serial untuk saling bertukar data dengan aplikasi Kontrol Gorden. Pada gambar ini menunjukkan komponen utama yang sudah dirancang dengan komponen lainnya.



Gambar 4.20 Rangkaian Komponen

4.5 Hasil Pengujian Alat

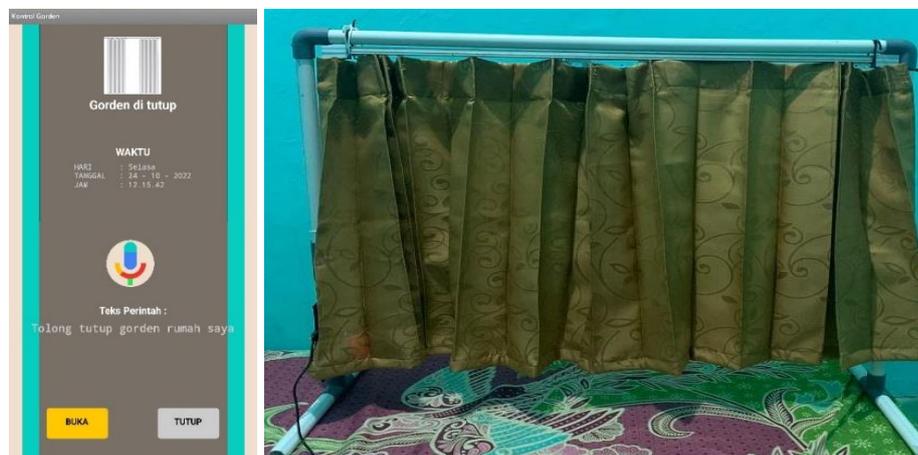
Pada penelitian ini, peneliti melakukan pengujian aplikasi yang dikoneksikan ke jaringan WIFI rumah dan NodeMCU ESP8266 V3 dihidupkan, setelah beberapa detik NodeMCU ESP8266 V3 terhubung di WIFI yang sama. Maka aplikasi Kontrol Gorden dapat dijalankan, berikut ini gambar aplikasi dan gambar alat-alatnya (Rangkaian Gorden) :



Gambar 4.21 Membuka Gorden

Gambar 4.21 merupakan tampilan aplikasi dan rangkaian gorden yang terperintah, dengan kata “tolong buka gordennya dong” yang di sebutkan oleh

peneliti dan menghasilkan gorden terbuka. Berikut gambar ketika peneliti memerintah untuk menutup gorden :



Gambar 4.22 Menutup Gorden

Gambar 4.22 merupakan tampilan aplikasi dan rangkaian gorden yang terperintah, dengan kata “tolong tutup gorden rumah saya” yang di sebutkan oleh peneliti dan menghasilkan gorden tertutup.

4.6 Integrasi Islam

Sistem kontrol gorden menggunakan perintah suara ini dirancang untuk mencapai tujuan berupa membantu atau menolong seseorang yang mengalami sakit fisik atau cacat yang tidak mampu melakukan kegiatan dengan leluasa. Hal ini sangat sesuai dengan apa yang telah difirmankan Allah SWT di dalam Al-Qur'an Surat Al-Ma'idah Ayat 2 yang berbunyi :

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَى

“Dan tolong-menolonglah kamu dalam (perbuatan) kebaikan dan ketakwaan”(QS. Al-Ma'idah : 2).

Menurut KH. Ahmad Zaeni Dahlan dalam kajian Surat Al Maidah Ayat 2 menjelaskan : “(Dan tolong menolonglah kalian dalam) menjalankan (kebajikan)

yaitu segala hal yang dapat membawa kebaikan dan kemaslahatan, (dan) tolong menolonglah dalam melaksanakan (Ketakwaan) yaitu segala hal yang dapat menghindarkan seorang dari keburukan dan kerugian”.

Firman Allah SWT dalam Al-Qur’an, menjelaskan akan Ke Maha besarannya seperti dalam Al-Qur’an.Surat Ali ‘Imran ayat 190 yang berbunyi :

إِنَّ فِي خَلْقِ السَّمُوتِ وَالْأَرْضِ وَأَخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِأُولِي الْأَلْبَابِ

“*Sesungguhnya dalam penciptaan langit dan bumi, dan silih bergantinya malam dan siang terdapat tanda-tanda bagi orang-orang yang berakal*”(QS. Ali ‘Imran : 190).

Sesungguhnya di dalam penciptaan langit dan bumi, dari tidak ada menjadi ada serta tanpa ada contoh sebelumnya, dan di dalam pergantian malam dan siang serta perbedaan panjang dan pendeknya waktu, benar-benar terdapat bukti-bukti nyata bagi orang-orang yang berakal sehat yang menunjukkan mereka kepada Sang Maha Pencipta alam semesta, hanya Dia Yang berhak disembah (Tafsir Al-Mukhtashar Ibnu Katsir).

Manusia adalah makhluk yang berakal yang diciptakan Allah SWT yang diberikan kemampuan melalui akal nya menangkap atau menerima pancaran ilmu Allah SWT. sehingga mampu mencipta atau berkreasi menemukan hal-hal yang sebelumnya belum ditemkan orang lain, tentunya dengan izin Allah SWT.

Sebagaimana dinyatakan dalam Al-Qur’an surat Al Alaq Ayat 4-5 :

الَّذِي عَلَّمَ بِالْقَلَمِ - عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ

"Yang mengajar (manusia) dengan kalam. Dia mengajarkan manusia apa yang tidak diketahuinya" (QS. Al-Alaq : 4-5).

Sistem kontrol gorden adalah salah satu penelitian saat ini yang memerlukan perangkat akal untuk mewujudkannya, sehingga bisa bermanfaat untuk orang yang membutuhkannya.

BAB V

PENUTUP

5.1 Kesimpulan

Implementasi dari sistem kontrol gorden dalam penelitian ini menggunakan metode *Knuth Morris Pratt* yang dipadukan dengan *internet of things*. Sistem kontrol gorden yang telah dibuat menghasilkan suatu alat yang berhubungan dengan aktivitas manusia serta diharapkan dapat memberikan manfaat dalam kehidupan sehari-hari.

Hasil analisa sistem kontrol gorden yang dilakukan oleh peneliti dengan menggunakan algoritma *knuth morris pratt* menghasilkan rata-rata durasi dalam pencarian *pattern* 513ms, sedangkan uji akurasi menggunakan rumus *confusion matrix* menghasilkan nilai akurasi sebesar 100%.

5.2 Saran

Sistem kontrol gorden menggunakan algoritma *knuth morris pratt* ini tentu masih memiliki kekurangan, sehingga dapat dikembangkan lagi oleh penelitian selanjutnya dengan menggunakan tambahan berbagai alat seperti lampu rumah, kipas angin, pintu, dan lainnya.

DAFTAR PUSTAKA

- Al-Qur'an dan terjemahan. Kementerian Agama Republik Indonesia. 2017.
- Adiono, T., Putra, R. W., Fathany, M. Y., & Adijarto, W. (2021). Desain Sistem Rumah Cerdas Berbasis Topologi Mesh dan Protokol Wireless Sensor Network yang Efisien. *INKOM Journal*, 65 - 72.
- Dewi, N. H., Rohmah, M. F., & Zahara, S. (n.d.). Prototype Smart Home dengan Modul NodeMCU ESP8266 Berbasis Internet of Things (IoT). 1 - 9.
- Effendi, Y. (2018). Internet of Things (IoT) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis Mobile. *Jurnal Ilmiah Ilmu Komputer Vol. 4, No. 1*, 19 - 26.
- Endra, R. Y., Cucus, A., Afandi, F. N., & Syahputra, M. B. (2019). Model Smart Room Dengan Menggunakan Mikrokontroler Arduino Untuk Efisiensi Sumber Daya. *Explore: Jurnal Sistem informasi dan telematika (Telekomunikasi, Multimedia dan Informatika)*, 10(1).
- Firdaus, Basyir, M., & Finawan, A. (2021). Rancang Bangun Prototype Sistem Kendali Keamanan Pada Jendela Pintar Berbasis Internet of Things. *Jurnal Listrik Telekomunikasi Elektronika Vol. 18, No. 2*, 78 - 85.
- Fragastia, V. A., & Rahmad, I. F. (2019). Penerapan Internet Of Things (IoT) Untuk Mendeteksi Kadar Alkohol Pada Pengendara Mobil. *IESM Journal (Industrial Engineering System and Management Journal)*, 1(1), 11-19.
- Iklima, Z. (2020). Rancang Bangun Prototipe Sistem Kontrol Terdistribusi untuk Pemantauan dan Pengendali Ketinggian Permukaan Air pada 5 Pintu Air Berbasis IoT Menggunakan Socket.IO. *Jurnal Teknologi Elektro*, 153 - 158.
- Ilham, M., & Mirza, A. H. (2020). Penerapan Algoritma Knuth Morris Pratt Dalam Fitur Pencarian Pengarsipan Dokumen Pada Sma Plus Negeri 17 Palembang. *Journal of Software Engineering Ampera*, 1(2), 110-121.
- Kedoh, A. R., Nursalim, Djahi, H., & Pollo, D. E. (n.d.). Sistem Kontrol Rumah Berbasis Internet of Things (IoT) Menggunakan Arduino Uno.
- Mughni, F. I., Riyadi, S., & Putra, A. P. (2020). Sistem Pengendali Background dan Lampu Pada Studio Foto Menggunakan Raspberry Pi dan Berbasis Website.
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, 5(2), 697-711.
- Noviansyah, M., & Saiyar, H. (2019). Perancangan Alat Kontrol Relay Lampu Rumah Via Mobile. *Jurnal Akrab Juara*, 85 - 97.

- Nurrahmadi, F., Rullah, I., Lagumdjiza, L., & Kardian, A. R. (2018). Perancangan Alat Gorden dan Lampu Untuk Smarthome Berbasis Mikrokontroler dan Android. *Seminar Nasional Teknologi Informasi dan Komunikasi Vol. 2*, 38 - 41.
- Prihatmoko, D. (2016). Penerapan Internet of Things (IoT) Dalam Pembelajaran di Unisnu Jepara. *Jurnal SIMETRIS Vol. 7, No. 2*, 567 - 574.
- Putri, A. M., Istiasih, H., & Santoso, R. (2021). Rancang Bangun Mesin Penyemprot Cat Dinding Menggunakan Dinamo DC. *Nusantara of Engineering*, 4(1), 10.
- Risma, Farida, & Andriani, S. (2021). Android Mobile Learning : MIT App Inventor dan Pengembangannya pada Pembelajaran Matematika. *Alphamath*, Vol. 7 No.1 .
- Samsugi, S., Ardiansyah, & Kastutara, D. (2017). Internet of Things (IoT) : Sistem Kendali Jarak Jauh Berbasis Arduino dan Modul Wifi Esp8266. *Prosiding Seminar Nasional XII "Rekayasa Teknologi Industri dan Informasi"*, 295 - 303.
- Samsugi, S., Ardiansyah, & Kastutara, D. (2018). Arduino dan Modul Wifi ESP8266 Sebagai Media Kendali Jarak Jauh dengan Antarmuka Berbasis Android. *Jurnal Teknoinfo* , Vol.12 No.1 23 - 27.
- Saragih, A. S. (2016). Implementasi Algoritma Knuth-Morris-Pratt Pada Pencarian Kumpulan Rumus Matematika. *Jurnal Ilmiah INFOTEK*, 1(2).
- Satrio, W., Syauqy, D., & Fitriyah, H. (2018). Implementasi Kontrol Lampu dan Kipas Secara Implisit Menggunakan Suara Dengan Metode Text Processing Berbasis Embedded System. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 7067 - 7073.
- Setiawan, A., & Abdullah, D. (2021). Implementasi Internet of Things Pada Alat Hand Sanitizer Otomatis Menggunakan Telegram Messenger Bot Berbasis ESP8266. *Rekursif: Jurnal Informatika*, 9(2), 137-143.
- Suprianto, D., Firdaus, V. A.-H., Agustina, R., & Wibowo, D. W. (2019). *Microcontroller Arduino Untuk Pemula*. Malang: Jasakom.
- Wasista, S., Setiawardhana, Saraswati, D. A., & Susanto, E. (2019). *Aplikasi Internet of Things (IoT) dengan Arduino dan Android*. Yogyakarta: CV Budi Utama.
- Penu, M., Gunawan, P., & Huda, M. M. (2021). The DESIGN AND DEVELOPMENT OF COMBAT ROBOT SYSTEM ON THE CIA VERSION N2MR3 USING THE INTERNET OF THINGS (IOT) BASED RASPBERRY PI 4.0. *Jurnal Telkommil*, 2(Oktober), 1-10.

LAMPIRAN

Spesifikasi *Alibaba Cloud ECS*

Peneliti menggunakan *Alibaba Cloud ECS* sebagai server, terinstal Sistem Operasi *Ubuntu* yang digunakan dalam membangun sistem *Internet of Things*.

Berikut spesifikasinya :

- Sistem Operasi : Ubuntu 20.04 64bit
- CPU : 1 Core
- Ram : 1 Gb
- Penyimpanan : 40 Gb
- Zona Wilayah : Malaysia (Kuala Lumpur Zona A)

Dalam *Alibaba Cloud ECS* dengan Sistem Operasi *Ubuntu* sudah terinstal kontrol panel (*aanel*), yang mana digunakan peneliti untuk mengendalikan *website* serta menjalankan terminal dari *aanel* untuk menjalankan subscribe MQTT. Berikut alamat *websitenya* <https://kontrolgorden.online/tata/mqtt>. *Website* yang berisi hasil perintah dari android, tabel pengujian dan video demo.

Pengujian teks perintah “tolong dibuka gordennya”

Perintah (Tolong dibuka gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong dibuka gordennya	tolong buka gorden	buka gorden	778
2	Tolong dibuka gordennya	tolong buka gorden	buka gorden	451
3	Tolong dibuka gordennya	tolong buka gorden	buka gorden	559
4	Tolong dibuka gordennya	tolong buka gorden	buka gorden	621
5	Tolong dibuka gordennya	tolong buka gorden	buka gorden	185
6	Tolong dibuka gordennya	tolong buka gorden	buka gorden	646
7	Tolong dibuka gordennya	tolong buka gorden	buka gorden	650
8	Tolong dibuka gordennya	tolong buka gorden	buka gorden	364
9	Tolong dibuka gordennya	tolong buka gorden	buka gorden	607
10	Tolong dibuka gordennya	tolong buka gorden	buka gorden	399
Rata-rata Durasi (ms)				526

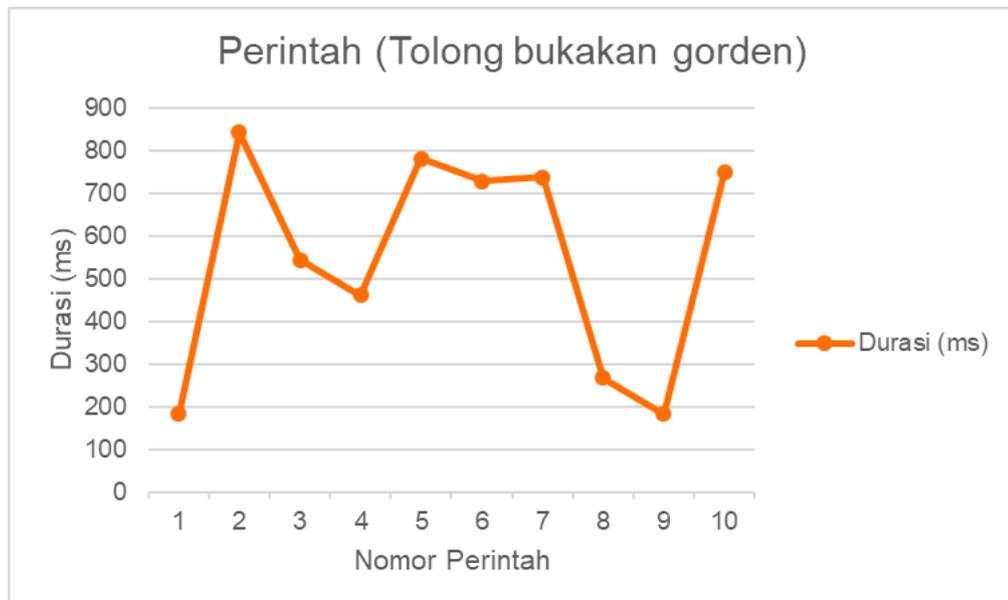
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong bukakan gorden”

Perintah (Tolong bukakan gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong bukakan gorden	tolong buka gorden	buka gorden	183
2	Tolong bukakan gorden	tolong buka gorden	buka gorden	842
3	Tolong bukakan gorden	tolong buka gorden	buka gorden	543
4	Tolong bukakan gorden	tolong buka gorden	buka gorden	460
5	Tolong bukakan gorden	tolong buka gorden	buka gorden	781
6	Tolong bukakan gorden	tolong buka gorden	buka gorden	728
7	Tolong bukakan gorden	tolong buka gorden	buka gorden	737
8	Tolong bukakan gorden	tolong buka gorden	buka gorden	266
9	Tolong bukakan gorden	tolong buka gorden	buka gorden	183
10	Tolong bukakan gorden	tolong buka gorden	buka gorden	749
Rata-rata Durasi (ms)				547

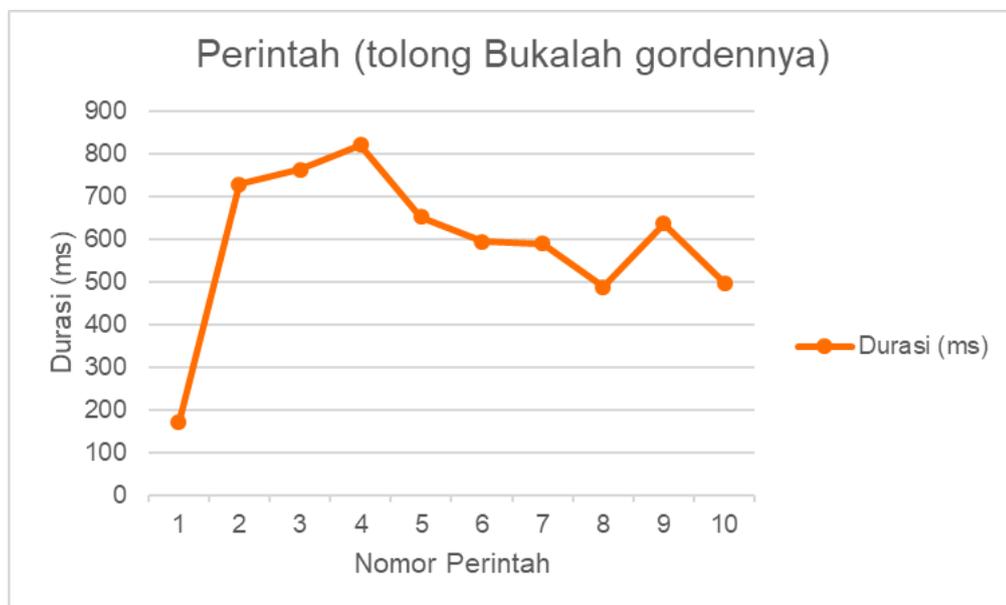
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong bukalah gordennya”

Perintah (tolong Bukalah gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong Bukalah gordennya	tolong buka gorden	buka gorden	170
2	tolong Bukalah gordennya	tolong buka gorden	buka gorden	728
3	tolong Bukalah gordennya	tolong buka gorden	buka gorden	763
4	tolong Bukalah gordennya	tolong buka gorden	buka gorden	820
5	tolong Bukalah gordennya	tolong buka gorden	buka gorden	651
6	tolong Bukalah gordennya	tolong buka gorden	buka gorden	594
7	tolong Bukalah gordennya	tolong buka gorden	buka gorden	588
8	tolong Bukalah gordennya	tolong buka gorden	buka gorden	487
9	tolong Bukalah gordennya	tolong buka gorden	buka gorden	635
10	tolong Bukalah gordennya	tolong buka gorden	buka gorden	497
Rata-rata Durasi (ms)				593

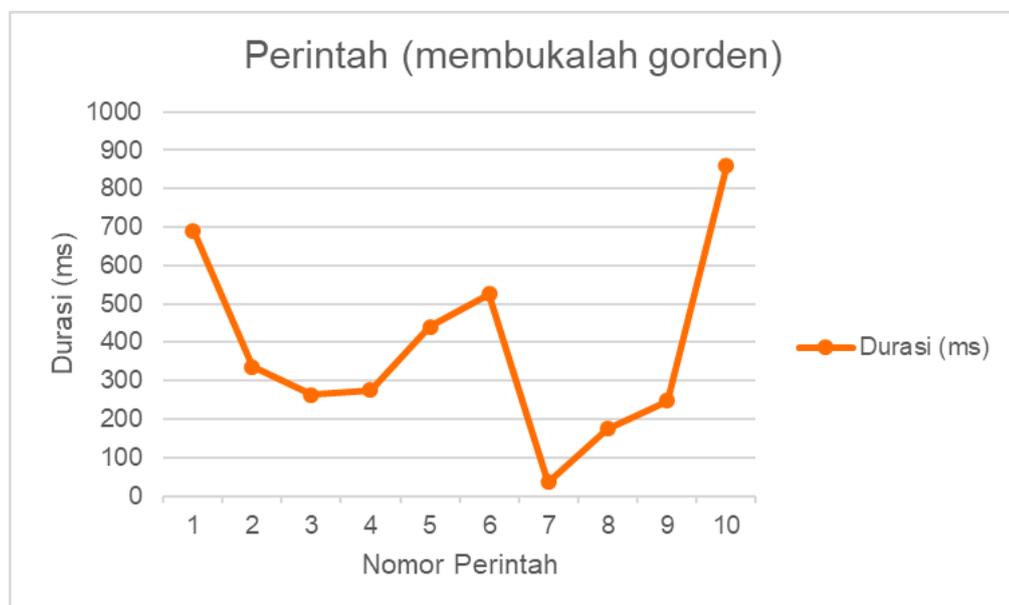
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “membukalah gorden”

Perintah (membukalah gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	membukalah gorden	buka gorden	buka gorden	690
2	membukalah gorden	buka gorden	buka gorden	336
3	membukalah gorden	buka gorden	buka gorden	263
4	membukalah gorden	buka gorden	buka gorden	276
5	membukalah gorden	buka gorden	buka gorden	441
6	membukalah gorden	buka gorden	buka gorden	526
7	membukalah gorden	buka gorden	buka gorden	35
8	membukalah gorden	buka gorden	buka gorden	174
9	membukalah gorden	buka gorden	buka gorden	248
10	membukalah gorden	buka gorden	buka gorden	860
Rata-rata Durasi (ms)				385

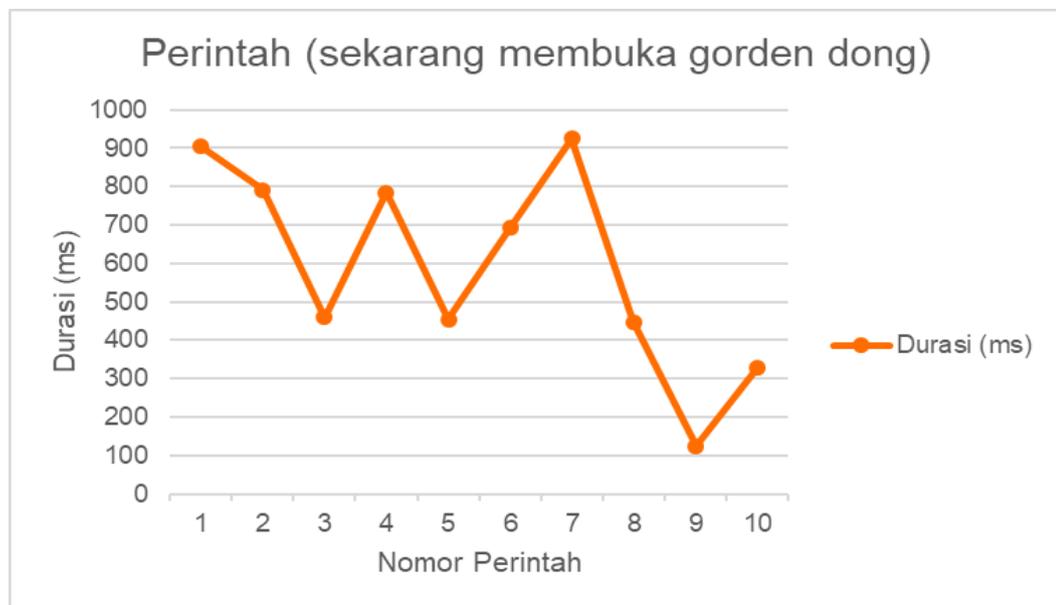
Indeks	[0 1 2 3 4 5 6 7 8 9]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang membuka gorden dong”

Perintah (sekarang membuka gorden dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	904
2	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	791
3	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	460
4	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	784
5	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	455
6	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	693
7	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	926
8	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	448
9	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	125
10	sekarang membuka gorden dong	sekarang buka gorden dong	buka gorden	329
Rata-rata Durasi (ms)				592

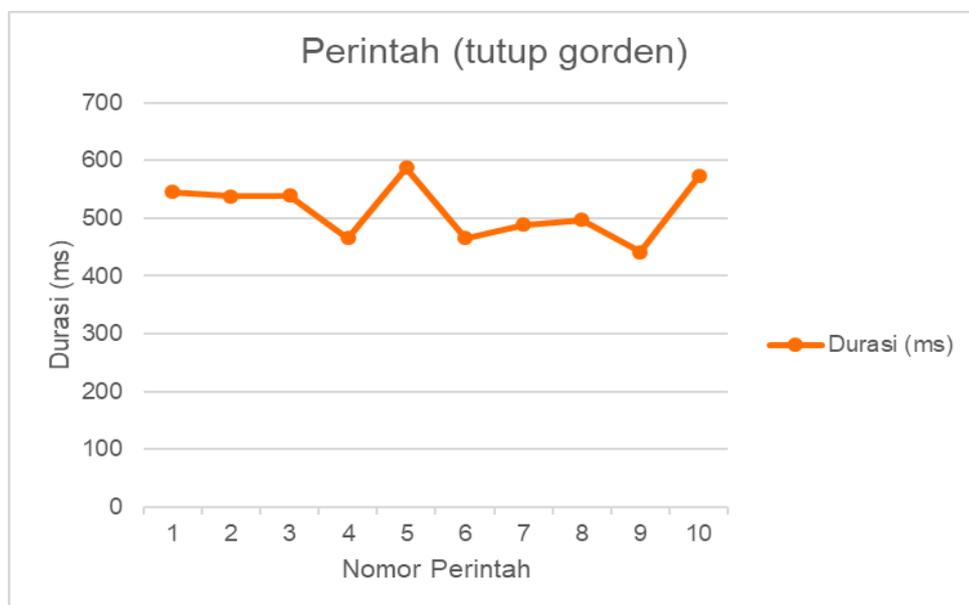
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 0 0 0 0]
Pergeseran Indeks ke	8



Pengujian teks perintah “tutup gorden”

Perintah (tutup gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tutup gorden	tutup gorden	tutup gorden	546
2	tutup gorden	tutup gorden	tutup gorden	538
3	tutup gorden	tutup gorden	tutup gorden	539
4	tutup gorden	tutup gorden	tutup gorden	466
5	tutup gorden	tutup gorden	tutup gorden	587
6	tutup gorden	tutup gorden	tutup gorden	466
7	tutup gorden	tutup gorden	tutup gorden	489
8	tutup gorden	tutup gorden	tutup gorden	497
9	tutup gorden	tutup gorden	tutup gorden	442
10	tutup gorden	tutup gorden	tutup gorden	573
Rata-rata Durasi (ms)				514

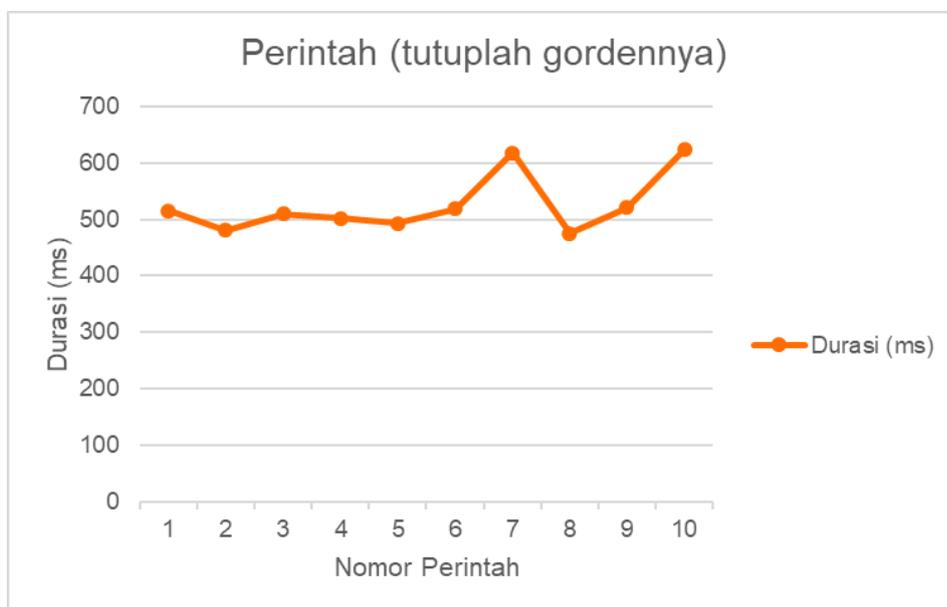
Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “tutuplah gordennya”

Perintah (tutuplah gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tutuplah gordennya	tutup gorden	tutup gorden	515
2	tutuplah gordennya	tutup gorden	tutup gorden	481
3	tutuplah gordennya	tutup gorden	tutup gorden	509
4	tutuplah gordennya	tutup gorden	tutup gorden	502
5	tutuplah gordennya	tutup gorden	tutup gorden	493
6	tutuplah gordennya	tutup gorden	tutup gorden	519
7	tutuplah gordennya	tutup gorden	tutup gorden	617
8	tutuplah gordennya	tutup gorden	tutup gorden	475
9	tutuplah gordennya	tutup gorden	tutup gorden	521
10	tutuplah gordennya	tutup gorden	tutup gorden	623
Rata-rata Durasi (ms)				526

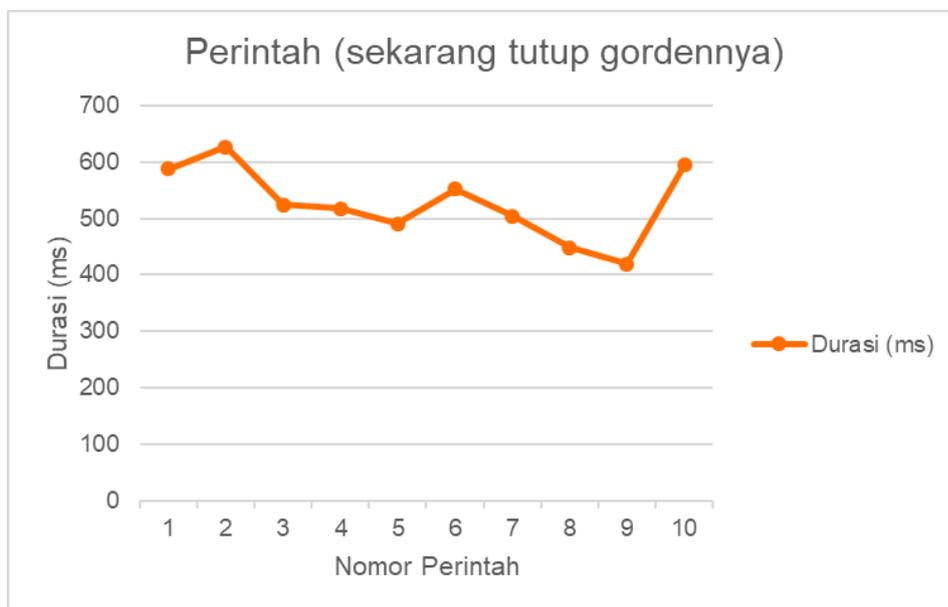
Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang tutup gordennya”

Perintah (sekarang tutup gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	587
2	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	626
3	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	524
4	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	517
5	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	491
6	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	552
7	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	504
8	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	448
9	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	419
10	sekarang tutup gordennya	sekarang tutup gorden	tutup gorden	594
Rata-rata Durasi (ms)				526

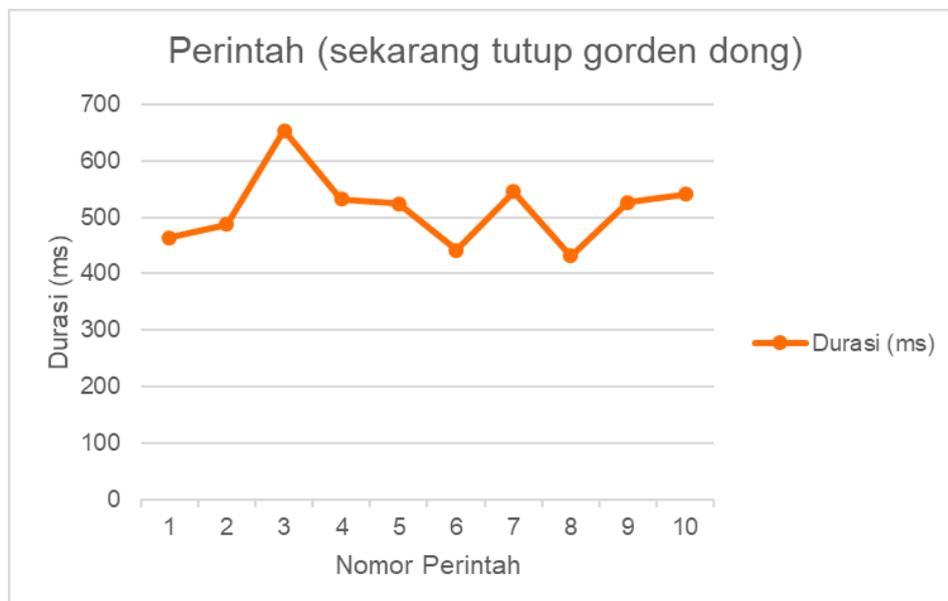
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	8



Pengujian teks perintah “sekarang tutup gorden dong”

Perintah (sekarang tutup gorden dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	464
2	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	487
3	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	653
4	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	532
5	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	524
6	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	442
7	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	545
8	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	432
9	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	526
10	sekarang tutup gorden dong	sekarang tutup gorden dong	tutup gorden	541
Rata-rata Durasi (ms)				515

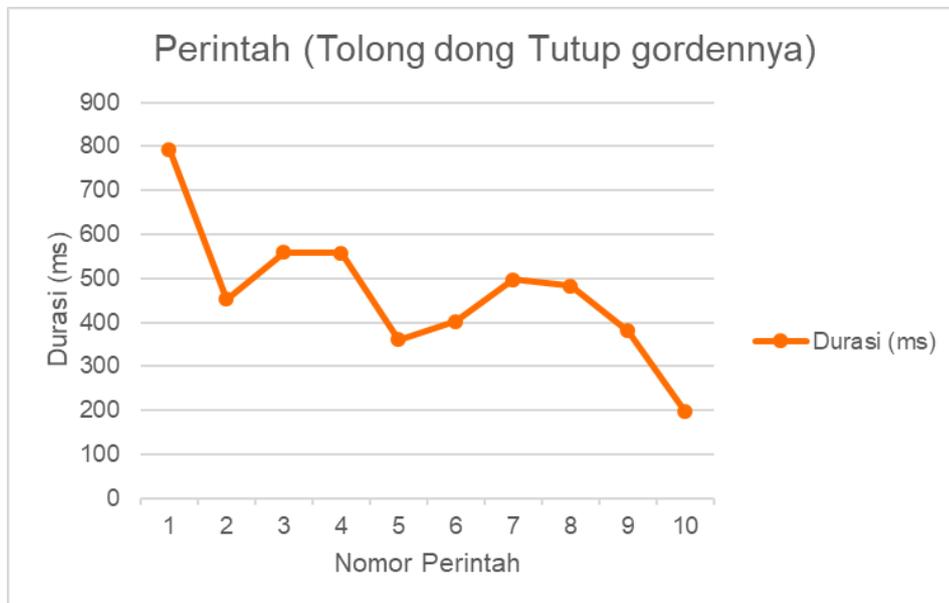
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11 0 0 0 0]
Pergeseran Indeks ke	8



Pengujian teks perintah “tolong dong tutup gordennya”

Perintah (Tolong dong Tutup gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	794
2	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	452
3	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	559
4	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	557
5	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	360
6	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	402
7	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	497
8	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	483
9	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	380
10	Tolong dong Tutup gordennya	Tolong dong Tutup gorden	tutup gorden	197
Rata-rata Durasi (ms)				468

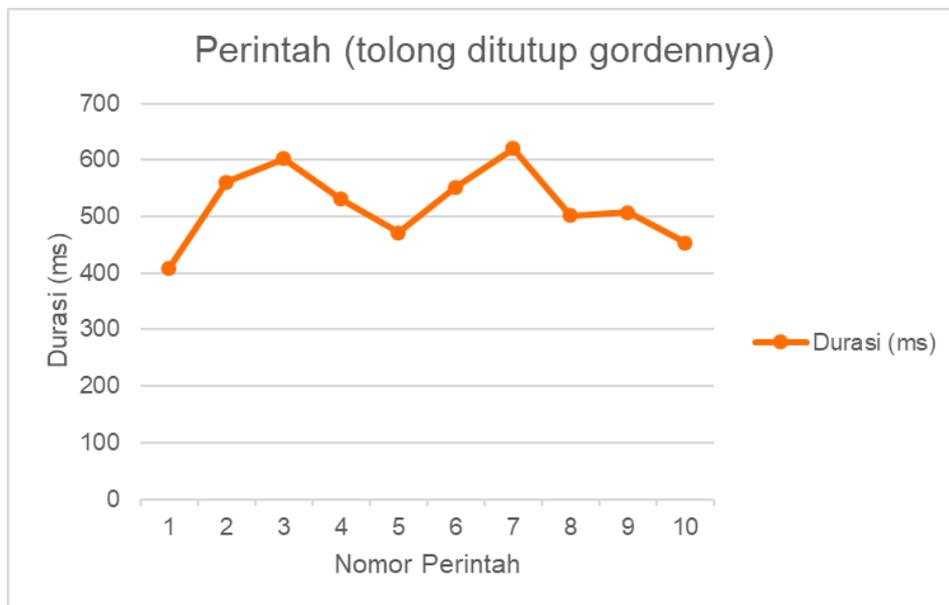
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]
LPS Stemming	[1 0 0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	10



Pengujian teks perintah “tolong ditutup gordennya”

Perintah (tolong ditutup gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	409
2	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	561
3	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	603
4	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	531
5	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	472
6	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	552
7	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	620
8	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	502
9	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	508
10	tolong ditutup gordennya	tolong tutup gorden	tutup gorden	454
Rata-rata Durasi (ms)				521

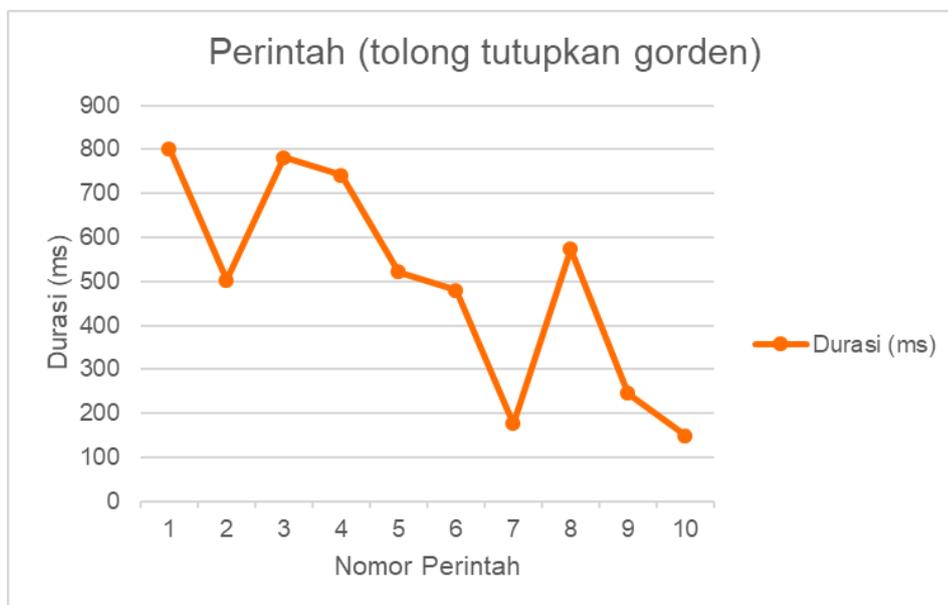
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[1 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong tutupkan gorden”

Perintah (tolong tutupkan gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	802
2	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	503
3	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	782
4	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	742
5	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	522
6	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	480
7	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	177
8	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	573
9	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	246
10	tolong tutupkan gorden	tolong tutup gorden	tutup gorden	149
Rata-rata Durasi (ms)				498

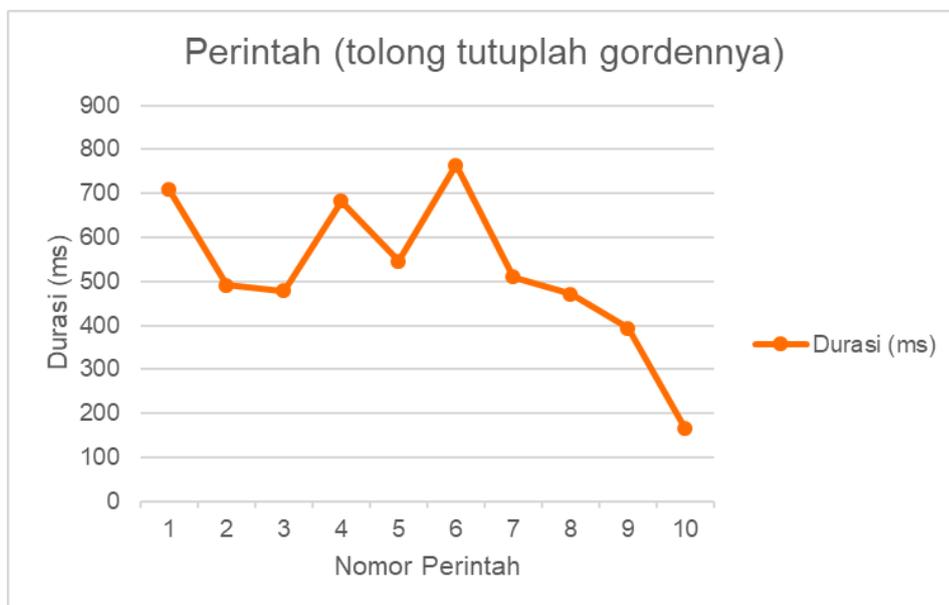
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[1 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong tutuplah gordennya”

Perintah (tolong tutuplah gordennya)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	708
2	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	492
3	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	479
4	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	683
5	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	546
6	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	765
7	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	510
8	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	471
9	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	394
10	tolong tutuplah gordennya	tolong tutup gorden	tutup gorden	165
Rata-rata Durasi (ms)				521

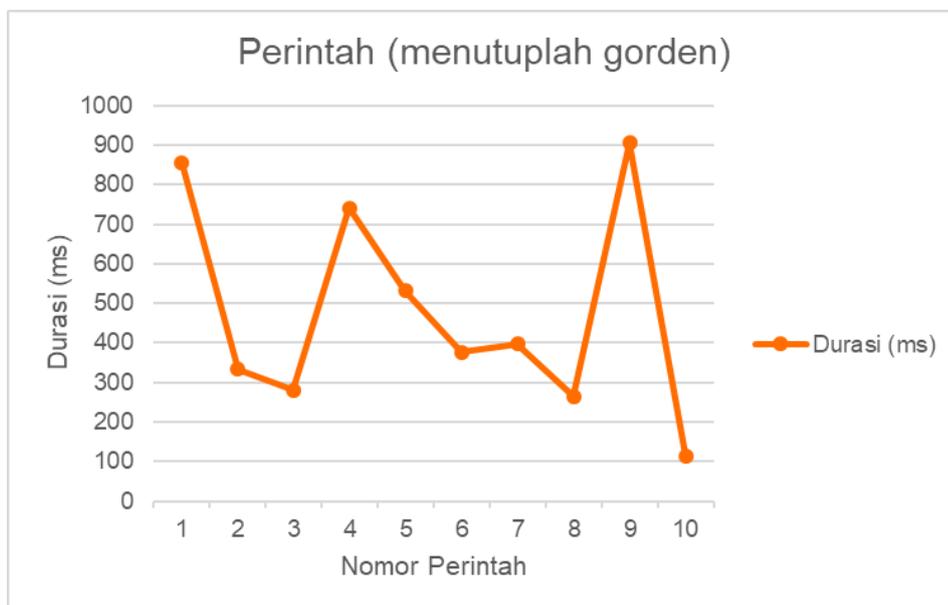
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[1 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “menutuplah gorden”

Perintah (menutuplah gorden)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	menutuplah gorden	tutup gorden	tutup gorden	854
2	menutuplah gorden	tutup gorden	tutup gorden	333
3	menutuplah gorden	tutup gorden	tutup gorden	280
4	menutuplah gorden	tutup gorden	tutup gorden	741
5	menutuplah gorden	tutup gorden	tutup gorden	531
6	menutuplah gorden	tutup gorden	tutup gorden	376
7	menutuplah gorden	tutup gorden	tutup gorden	397
8	menutuplah gorden	tutup gorden	tutup gorden	264
9	menutuplah gorden	tutup gorden	tutup gorden	905
10	menutuplah gorden	tutup gorden	tutup gorden	114
Rata-rata Durasi (ms)				480

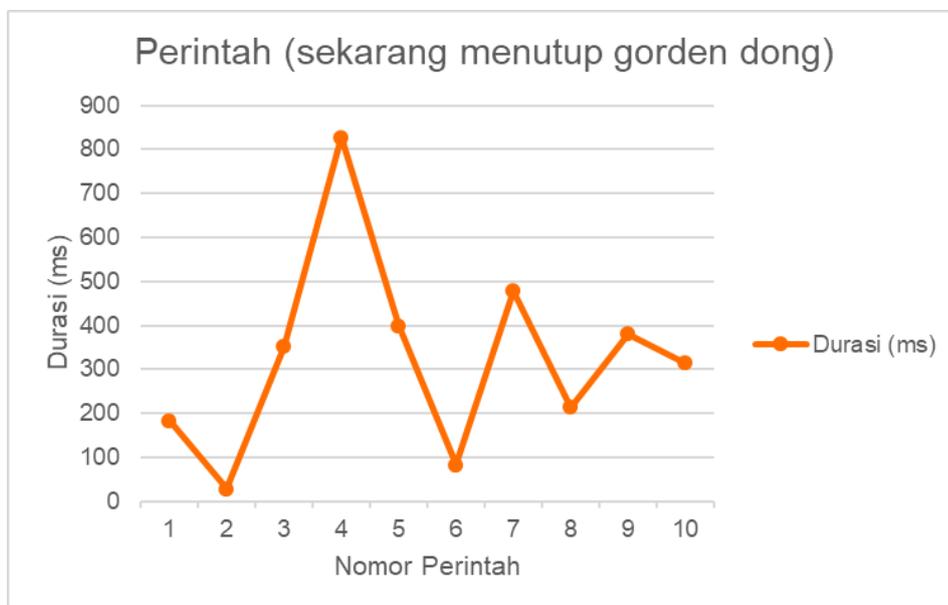
Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang menutup gorden dong”

Perintah (sekarang menutup gorden dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	184
2	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	28
3	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	352
4	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	828
5	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	400
6	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	83
7	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	478
8	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	214
9	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	380
10	sekarang menutup gorden dong	sekarang tutup gorden dong	tutup gorden	314
Rata-rata Durasi (ms)				326

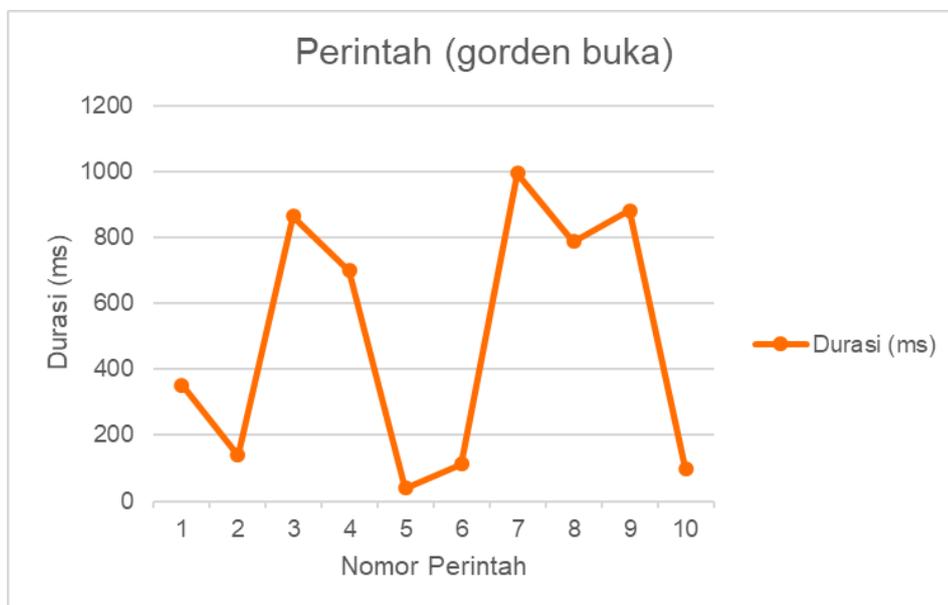
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
LPS Stemming	[0 0 0 0 0 0 0 1 2 3 4 5 6 7 8 9 10 11 0 0 0 0]
Pergeseran Indeks ke	8



Pengujian teks perintah “gorden buka”

Perintah (gorden buka)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden buka	gorden buka	gorden buka	353
2	gorden buka	gorden buka	gorden buka	139
3	gorden buka	gorden buka	gorden buka	865
4	gorden buka	gorden buka	gorden buka	700
5	gorden buka	gorden buka	gorden buka	40
6	gorden buka	gorden buka	gorden buka	112
7	gorden buka	gorden buka	gorden buka	995
8	gorden buka	gorden buka	gorden buka	787
9	gorden buka	gorden buka	gorden buka	881
10	gorden buka	gorden buka	gorden buka	98
Rata-rata Durasi (ms)				497

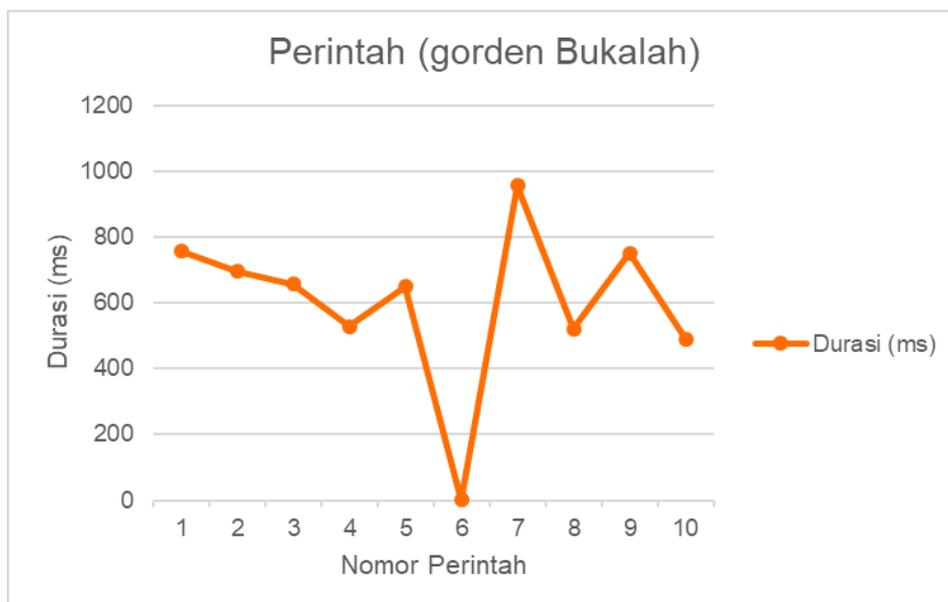
Indeks	[0 1 2 3 4 5 6 7 8 9]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	0



Pengujian teks perintah “gorden bukalah”

Perintah (gorden Bukalah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden Bukalah	gorden buka	gorden buka	757
2	gorden Bukalah	gorden buka	gorden buka	695
3	gorden Bukalah	gorden buka	gorden buka	658
4	gorden Bukalah	gorden buka	gorden buka	529
5	gorden Bukalah	gorden buka	gorden buka	649
6	gorden Bukalah	gorden buka	gorden buka	1
7	gorden Bukalah	gorden buka	gorden buka	956
8	gorden Bukalah	gorden buka	gorden buka	521
9	gorden Bukalah	gorden buka	gorden buka	751
10	gorden Bukalah	gorden buka	gorden buka	490
Rata-rata Durasi (ms)				601

Indeks	[0 1 2 3 4 5 6 7 8 9]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang gordennya buka”

Perintah (sekarang gordennya buka)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gordennya buka	sekarang gorden buka	gorden buka	576
2	sekarang gordennya buka	sekarang gorden buka	gorden buka	700
3	sekarang gordennya buka	sekarang gorden buka	gorden buka	258
4	sekarang gordennya buka	sekarang gorden buka	gorden buka	130
5	sekarang gordennya buka	sekarang gorden buka	gorden buka	692
6	sekarang gordennya buka	sekarang gorden buka	gorden buka	55
7	sekarang gordennya buka	sekarang gorden buka	gorden buka	569
8	sekarang gordennya buka	sekarang gorden buka	gorden buka	847
9	sekarang gordennya buka	sekarang gorden buka	gorden buka	794
10	sekarang gordennya buka	sekarang gorden buka	gorden buka	101
Rata-rata Durasi (ms)				472

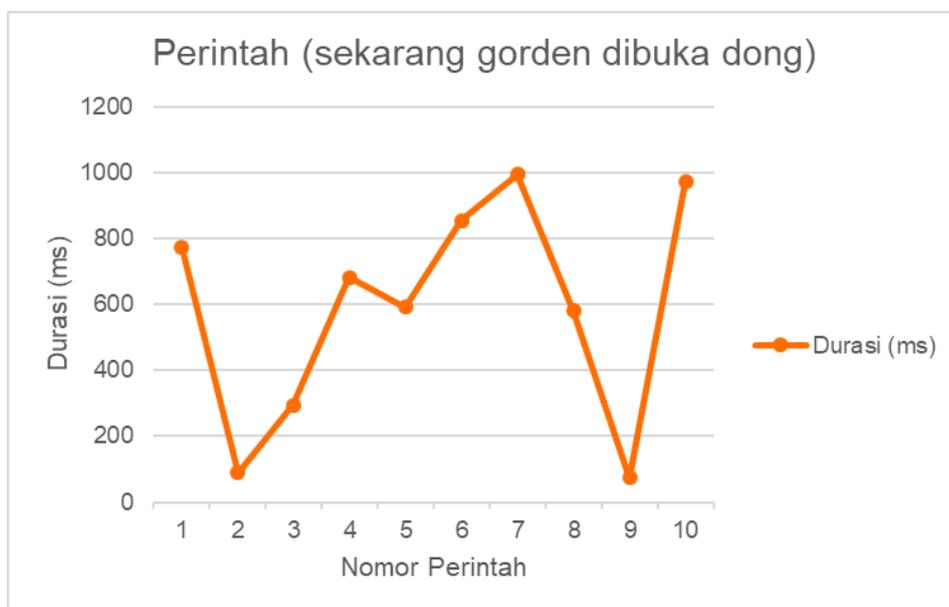
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17]
LPS Stemming	[0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	8



Pengujian teks perintah “sekarang gorden dibuka dong”

Perintah (sekarang gorden dibuka dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	772
2	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	89
3	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	294
4	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	682
5	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	592
6	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	855
7	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	994
8	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	580
9	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	74
10	sekarang gorden dibuka dong	sekarang gorden buka dong	gorden buka	973
Rata-rata Durasi (ms)				591

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21]
LPS Stemming	[0 0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 0 0 0 1]
Pergeseran Indeks ke	8



Pengujian teks perintah “tolong dong gordennya buka”

Perintah (Tolong dong gordennya buka)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	938
2	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	710
3	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	721
4	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	448
5	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	21
6	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	446
7	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	605
8	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	148
9	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	637
10	Tolong dong gordennya buka	Tolong dong gorden buka	gorden buka	501
Rata-rata Durasi (ms)				518

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]
LPS Stemming	[0 0 0 0 0 1 0 0 0 1 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	10



Pengujian teks perintah “tolong gordennya dibukakan”

Perintah (tolong gordennya dibukakan)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gordennya dibukakan	tolong gorden buka	gorden buka	916
2	tolong gordennya dibukakan	tolong gorden buka	gorden buka	528
3	tolong gordennya dibukakan	tolong gorden buka	gorden buka	71
4	tolong gordennya dibukakan	tolong gorden buka	gorden buka	503
5	tolong gordennya dibukakan	tolong gorden buka	gorden buka	503
6	tolong gordennya dibukakan	tolong gorden buka	gorden buka	877
7	tolong gordennya dibukakan	tolong gorden buka	gorden buka	80
8	tolong gordennya dibukakan	tolong gorden buka	gorden buka	670
9	tolong gordennya dibukakan	tolong gorden buka	gorden buka	481
10	tolong gordennya dibukakan	tolong gorden buka	gorden buka	689
Rata-rata Durasi (ms)				532

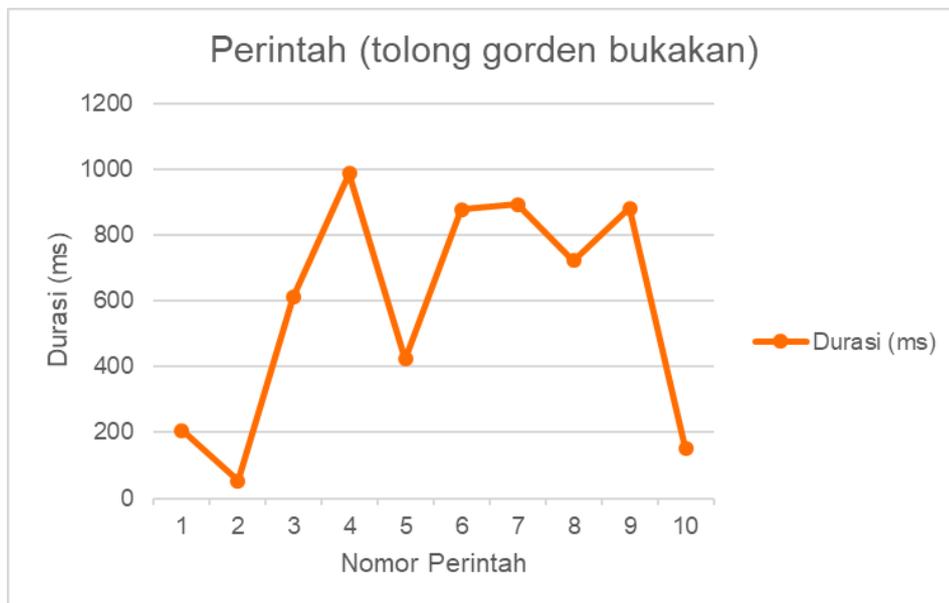
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong gorden bukakan”

Perintah (tolong gorden bukakan)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gorden bukakan	tolong gorden buka	gorden buka	207
2	tolong gorden bukakan	tolong gorden buka	gorden buka	53
3	tolong gorden bukakan	tolong gorden buka	gorden buka	612
4	tolong gorden bukakan	tolong gorden buka	gorden buka	987
5	tolong gorden bukakan	tolong gorden buka	gorden buka	424
6	tolong gorden bukakan	tolong gorden buka	gorden buka	878
7	tolong gorden bukakan	tolong gorden buka	gorden buka	893
8	tolong gorden bukakan	tolong gorden buka	gorden buka	722
9	tolong gorden bukakan	tolong gorden buka	gorden buka	881
10	tolong gorden bukakan	tolong gorden buka	gorden buka	151
Rata-rata Durasi (ms)				581

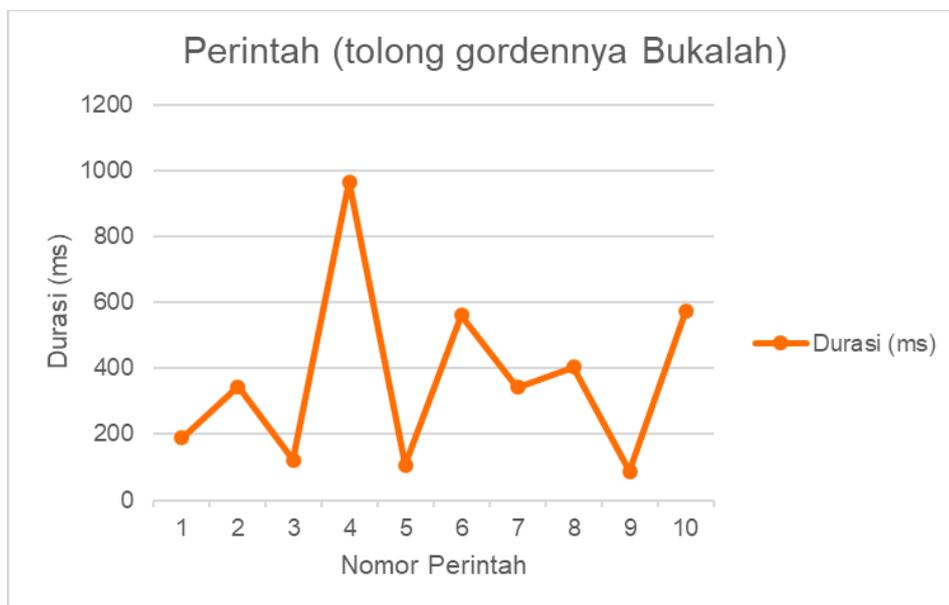
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong gordennya bukalah”

Perintah (tolong gordennya Bukalah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	190
2	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	345
3	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	122
4	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	966
5	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	106
6	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	562
7	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	343
8	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	403
9	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	87
10	tolong gordennya Bukalah	tolong gorden Buka	gorden buka	572
Rata-rata Durasi (ms)				370

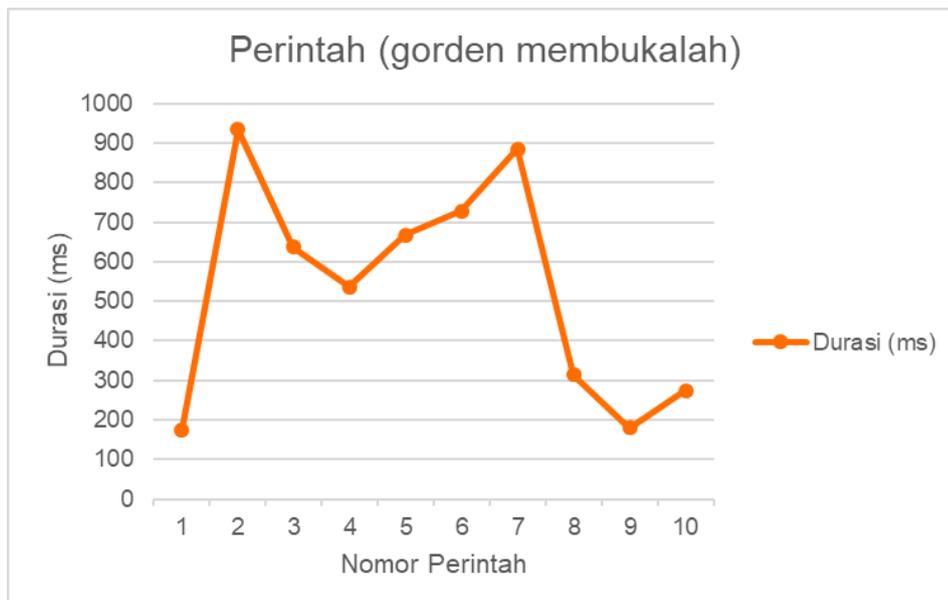
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
LPS Stemming	[0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	6



Pengujian teks perintah “gorden membukalah”

Perintah (gorden membukalah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden membukalah	gorden buka	gorden buka	175
2	gorden membukalah	gorden buka	gorden buka	934
3	gorden membukalah	gorden buka	gorden buka	637
4	gorden membukalah	gorden buka	gorden buka	536
5	gorden membukalah	gorden buka	gorden buka	667
6	gorden membukalah	gorden buka	gorden buka	728
7	gorden membukalah	gorden buka	gorden buka	884
8	gorden membukalah	gorden buka	gorden buka	315
9	gorden membukalah	gorden buka	gorden buka	180
10	gorden membukalah	gorden buka	gorden buka	274
Rata-rata Durasi (ms)				533

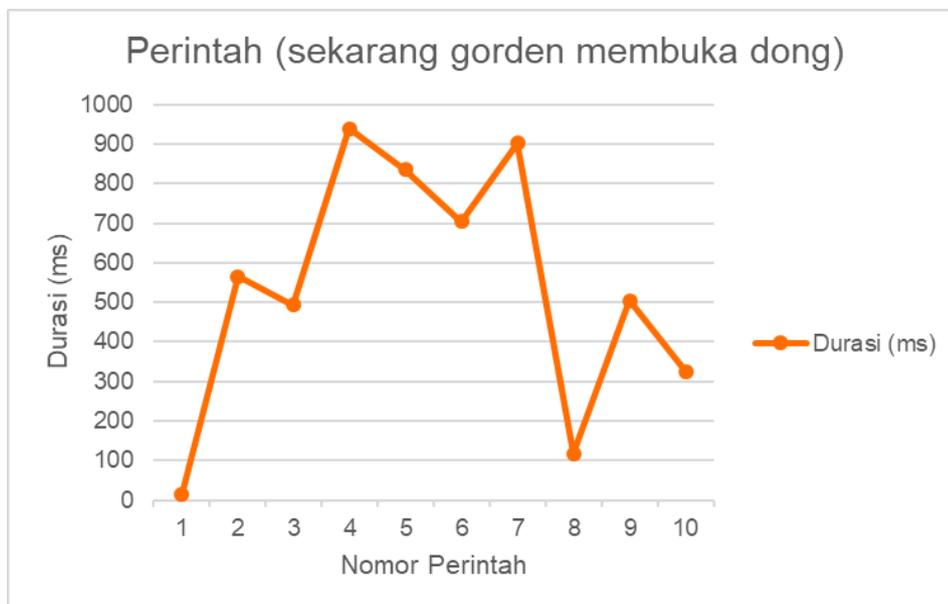
Indeks	[0 1 2 3 4 5 6 7 8 9]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang gorden membuka dong”

Perintah (sekarang gorden membuka dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	16
2	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	565
3	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	493
4	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	939
5	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	836
6	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	705
7	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	902
8	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	118
9	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	504
10	sekarang gorden membuka dong	sekarang gorden buka dong	gorden buka	325
Rata-rata Durasi (ms)				540

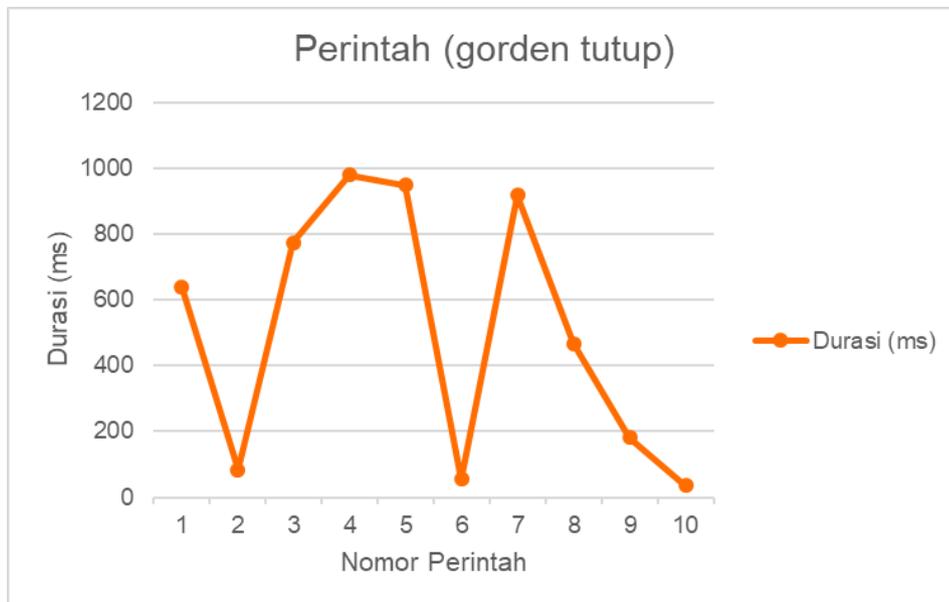
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21]
LPS Stemming	[0 0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 0 0 0 1]
Pergeseran Indeks ke	8



Pengujian teks perintah “gorden tutup”

Perintah (gorden tutup)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden tutup	gorden tutup	gorden tutup	637
2	gorden tutup	gorden tutup	gorden tutup	83
3	gorden tutup	gorden tutup	gorden tutup	775
4	gorden tutup	gorden tutup	gorden tutup	980
5	gorden tutup	gorden tutup	gorden tutup	949
6	gorden tutup	gorden tutup	gorden tutup	54
7	gorden tutup	gorden tutup	gorden tutup	917
8	gorden tutup	gorden tutup	gorden tutup	468
9	gorden tutup	gorden tutup	gorden tutup	181
10	gorden tutup	gorden tutup	gorden tutup	36
Rata-rata Durasi (ms)				508

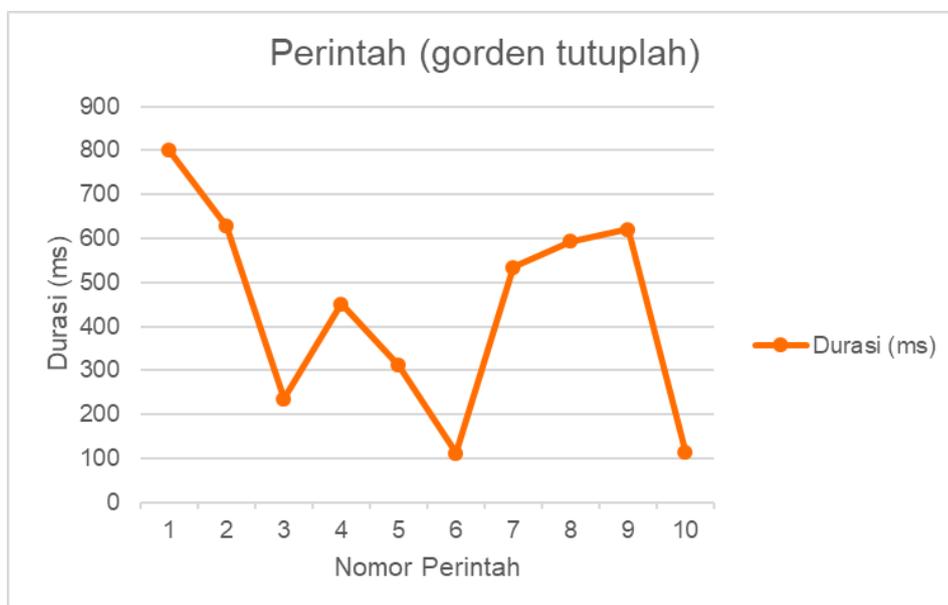
Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “gorden tutuplah”

Perintah (gorden tutuplah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden tutuplah	gorden tutup	gorden tutup	800
2	gorden tutuplah	gorden tutup	gorden tutup	628
3	gorden tutuplah	gorden tutup	gorden tutup	235
4	gorden tutuplah	gorden tutup	gorden tutup	451
5	gorden tutuplah	gorden tutup	gorden tutup	312
6	gorden tutuplah	gorden tutup	gorden tutup	112
7	gorden tutuplah	gorden tutup	gorden tutup	535
8	gorden tutuplah	gorden tutup	gorden tutup	593
9	gorden tutuplah	gorden tutup	gorden tutup	621
10	gorden tutuplah	gorden tutup	gorden tutup	114
Rata-rata Durasi (ms)				440

Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang gordennya tutup”

Perintah (sekarang gordennya tutup)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	762
2	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	897
3	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	279
4	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	530
5	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	233
6	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	661
7	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	667
8	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	749
9	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	979
10	sekarang gordennya tutup	sekarang gorden tutup	gorden tutup	299
Rata-rata Durasi (ms)				606

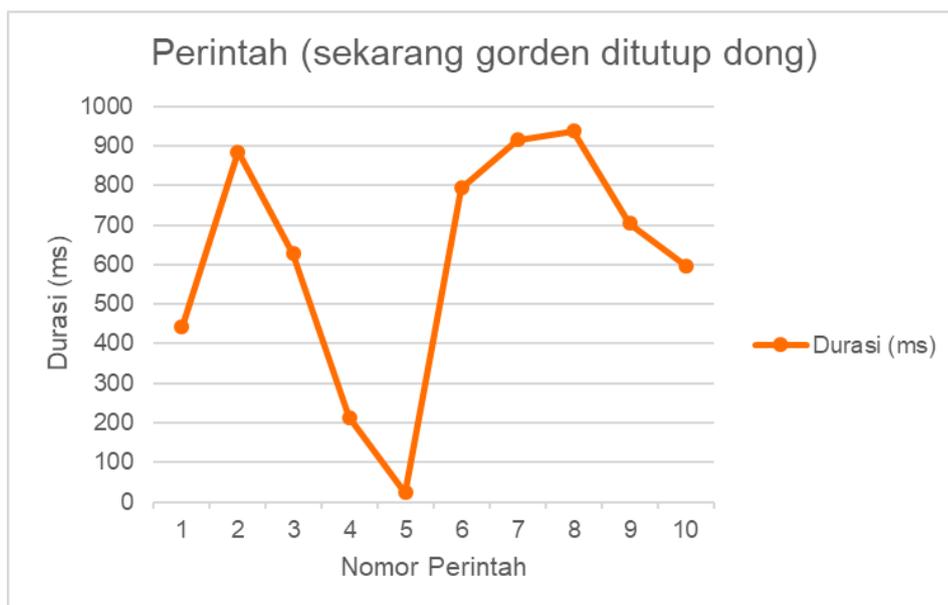
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18]
LPS Stemming	[0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	8



Pengujian teks perintah “sekarang gorden ditutup dong”

Perintah (sekarang gorden ditutup dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	442
2	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	885
3	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	627
4	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	214
5	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	24
6	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	794
7	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	915
8	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	938
9	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	705
10	sekarang gorden ditutup dong	sekarang gorden tutup dong	gorden tutup	597
Rata-rata Durasi (ms)				614

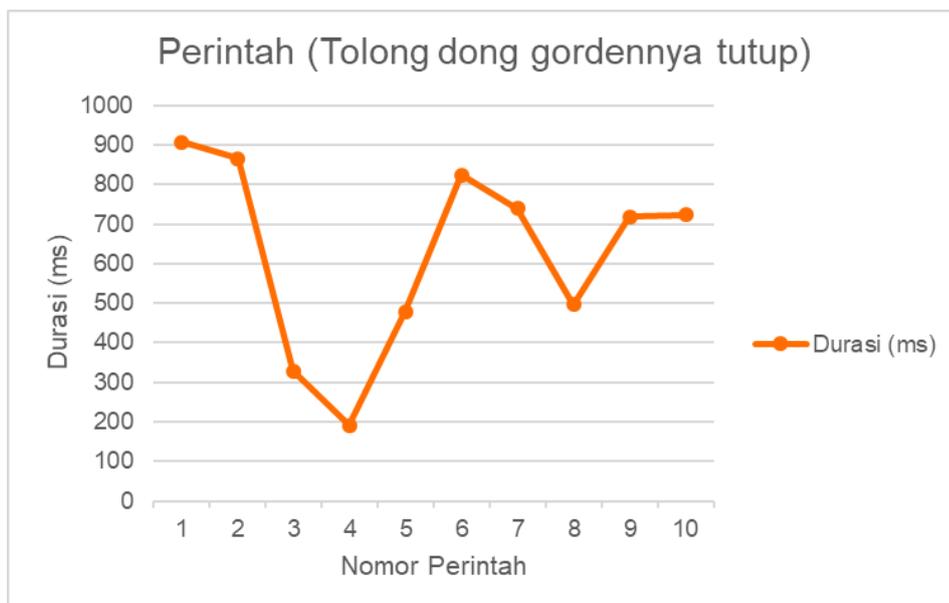
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
LPS Stemming	[0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11 0 0 0 1]
Pergeseran Indeks ke	8



Pengujian teks perintah “tolong dong gordennya tutup”

Perintah (Tolong dong gordennya tutup)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	907
2	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	865
3	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	328
4	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	191
5	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	479
6	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	824
7	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	739
8	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	496
9	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	718
10	Tolong dong gordennya tutup	Tolong dong gorden tutup	gorden tutup	724
Rata-rata Durasi (ms)				627

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]
LPS Stemming	[0 0 0 0 1 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	10



Pengujian teks perintah “tolong gordennya ditutupkan”

Perintah (tolong gordennya ditutupkan)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	273
2	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	603
3	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	497
4	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	604
5	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	258
6	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	935
7	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	37
8	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	497
9	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	776
10	tolong gordennya ditutupkan	tolong gorden tutup	gorden tutup	547
Rata-rata Durasi (ms)				503

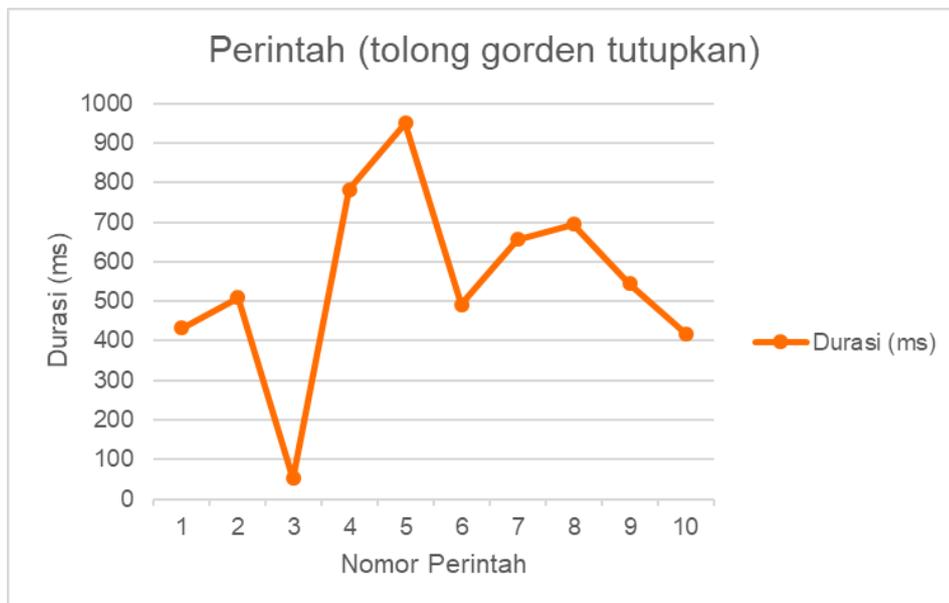
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong gorden tutupkan”

Perintah (tolong gorden tutupkan)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	433
2	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	510
3	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	52
4	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	782
5	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	951
6	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	492
7	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	656
8	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	694
9	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	545
10	tolong gorden tutupkan	tolong gorden tutup	gorden tutup	418
Rata-rata Durasi (ms)				553

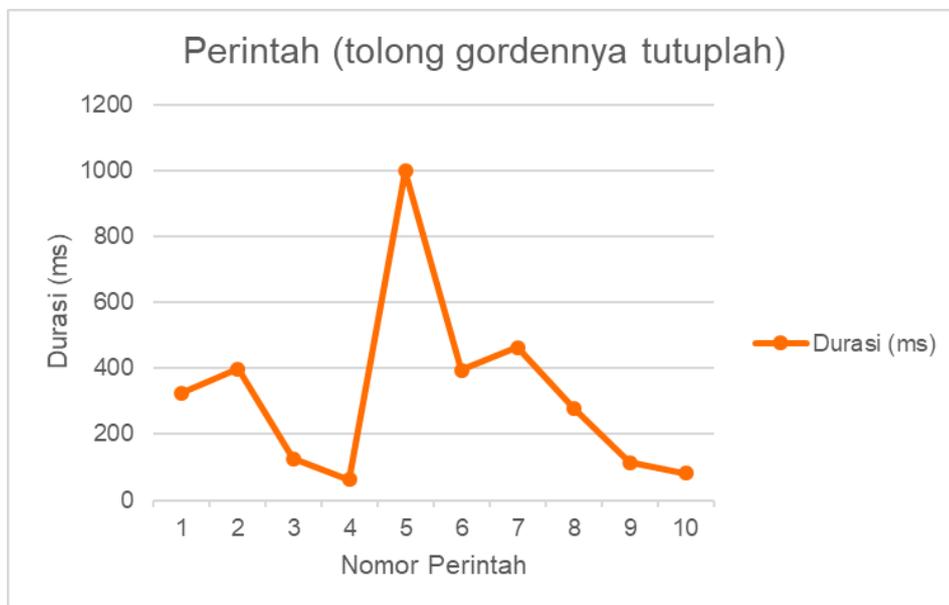
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “tolong gordennya tutuplah”

Perintah (tolong gordennya tutuplah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	326
2	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	398
3	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	126
4	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	63
5	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	999
6	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	395
7	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	464
8	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	280
9	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	115
10	tolong gordennya tutuplah	tolong gorden tutup	gorden tutup	81
Rata-rata Durasi (ms)				325

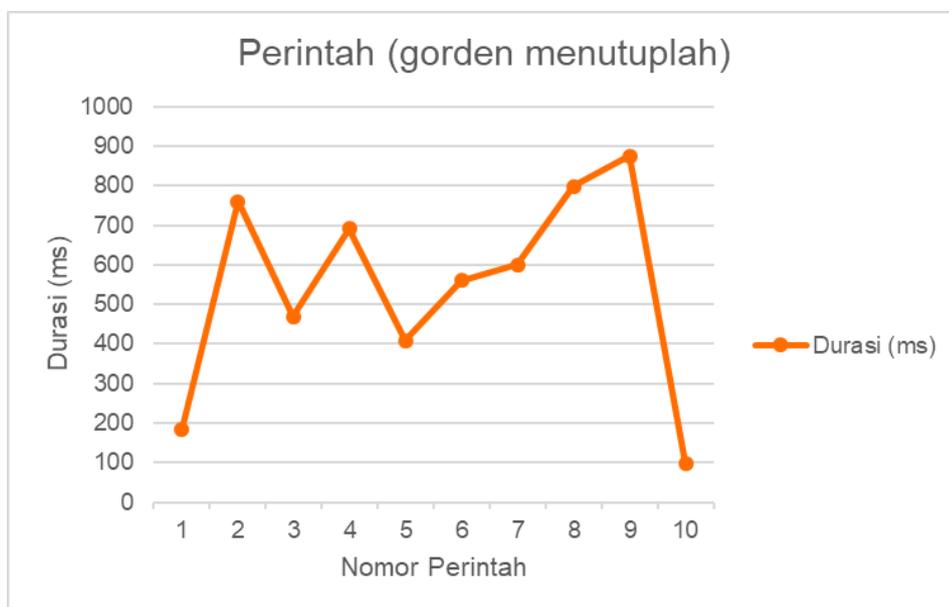
Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16]
LPS Stemming	[0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	6



Pengujian teks perintah “gorden menutuplah”

Perintah (gorden menutuplah)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	gorden menutuplah	gorden tutup	gorden tutup	185
2	gorden menutuplah	gorden tutup	gorden tutup	760
3	gorden menutuplah	gorden tutup	gorden tutup	469
4	gorden menutuplah	gorden tutup	gorden tutup	692
5	gorden menutuplah	gorden tutup	gorden tutup	408
6	gorden menutuplah	gorden tutup	gorden tutup	560
7	gorden menutuplah	gorden tutup	gorden tutup	600
8	gorden menutuplah	gorden tutup	gorden tutup	798
9	gorden menutuplah	gorden tutup	gorden tutup	875
10	gorden menutuplah	gorden tutup	gorden tutup	97
Rata-rata Durasi (ms)				544

Indeks	[0 1 2 3 4 5 6 7 8 9 10]
LPS Stemming	[1 2 3 4 5 6 7 8 9 10 11]
Pergeseran Indeks ke	0



Pengujian teks perintah “sekarang gorden meutup dong”

Perintah (sekarang gorden menutup dong)				
No	Teks Perintah	Teks Stemming	Pattern	Durasi (ms)
1	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	583
2	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	646
3	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	67
4	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	102
5	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	921
6	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	636
7	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	329
8	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	675
9	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	967
10	sekarang gorden menutup dong	sekarang gorden tutup dong	gorden tutup	436
Rata-rata Durasi (ms)				536

Indeks	[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
LPS Stemming	[0 0 0 0 0 0 1 1 2 3 4 5 6 7 8 9 10 11 0 0 0 1]
Pergeseran Indeks ke	8

