SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI*

SKRIPSI

Oleh:
ALVA SAPUTRA
NIM. 200605110176

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

SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI*

SKRIPSI

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

Oleh :
ALVA SAPUTRA
NIM. 200605110176

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

HALAMAN PERSETUJUAN

SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI*

SKRIPSI

Oleh :
ALVA SAPUTRA
NIM. 200605110176

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 11 Desember 2025

Pembimbing I,	Pembimbing II,
Ajib Hanani, M.T	Dr. Fachrul Kurniawan, M.MT, IPU
NIP. 198407312023211013	NIP. 19771020 200912 1 001

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI*

SKRIPSI

Oleh :
ALVA SAPUTRA
NIM. 200605110176

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Desember 2025

Susunan Dewan Penguji

Ketua Penguji	: Dr. Yunifa Miftachul Arif, M.T NIP. 19830616 201101 1 004	( )
Anggota Penguji I	: Shoffin Nahwa Utama, M.T NIP. 19860703 202012 1 003	( )
Anggota Penguji II	: Ajib Hanani, M.T NIP. 19840731 202321 1 013	( )
Anggota Penguji III	: Dr. Fachrul Kurniawan, M.MT, IPU NIP. 19771020 200912 1 001	( )

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

Supriyono, M. Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: Alva Saputra
NIM	: 200605110176
Fakultas / Jurusan	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI*

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

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

Malang, 12 Desember 2025
Yang membuat pernyataan,

Alva Saputra
NIM.200605110176

MOTTO

Langkah kecil yang konsisten lebih berarti daripada rencana besar yang hanya disimpan dalam angan.

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur ke hadirat Allah SWT atas segala rahmat, taufik, dan hidayah-Nya, Tak lupa, shalawat dan salam selalu tercurah kepada Nabi Muhammad SAW. karya sederhana ini kupersembahkan sebagai wujud terima kasih yang tulus kepada:

Ayah dan Ibuku tercinta, yang namanya selalu kusebut dalam setiap doa. Terima kasih atas kasih sayang yang tak pernah surut, atas peluh yang tak pernah dihitung, atas nasihat yang tak berhenti meski sering kali tak segera kupahami. Setiap langkah yang kutempuh hingga hari ini adalah hasil dari keikhlasan, kesabaran, dan pengorbanan kalian.

Para dosen dan guru, sejak bangku sekolah hingga perguruan tinggi, yang telah menyalakan cahaya ilmu dalam hidupku. Terima kasih atas setiap pelajaran, kritik, dan bimbingan, baik yang disampaikan di ruang kelas, di ruang konsultasi, maupun dalam interaksi sederhana sehari-hari. Ilmu yang kalian berikan menjadi bekal berharga untuk melangkah menapaki masa depan.

Untuk diriku sendiri, yang telah bertahan sejauh ini. Terima kasih telah memilih untuk tidak menyerah, tetap berdiri meski sering kali ingin berhenti, tetap mencoba meski berkali-kali merasa gagal. Semoga keberhasilan kecil ini menjadi pengingat bahwa setiap usaha, seberapapun pelan, akan menemukan waktunya untuk sampai di garis akhir.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "SISTEM KONTROL KIPAS ANGIN BERBASIS KALIMAT MENGGUNAKAN ALGORITMA *COLUSSI"* Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang Dalam proses penyusunan skripsi ini, penulis menghadapi berbagai hambatan dan keterbatasan, baik dari segi pengetahuan, waktu, maupun kemampuan. Namun berkat bantuan, bimbingan, dukungan, dan doa dari berbagai pihak, akhirnya skripsi ini dapat diselesaikan. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang setulus-tulusnya kepada:

1. Ibu Prof. Dr. Hj Ilfi Nur Diana, M.Si selaku Rektor UIN Maulana Malik Ibrahim Malang, yang telah memberikan kesempatan kepada penulis untuk menempuh pendidikan di perguruan tinggi ini.
2. Dr. Agus Mulyono selaku Dekan Fakultas Sains dan Teknologi yang senantiasa mendukung proses akademik dan memberikan fasilitas bagi mahasiswa.
3. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Ajib Hanani, M.T selaku Dosen Pembimbing I dan Bapak Dr. Fachrul Kurniawan, M.MT, IPU selaku Dosen Pembimbing II, yang dengan sabar meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan,

masukan, dan koreksi kepada penulis dalam setiap tahap penyusunan skripsi, mulai dari penyusunan proposal hingga penyelesaian akhir.

5. Bapak Dr. Yunifa Miftachul Arif, M.T selaku dosen Penguji I dan Bapak Shoffin Nahwa Utama, M.T dosen Penguji II yang telah memberikan masukan dan saran yang membangun sehingga skripsi ini dapat dipertangguhkan dengan lebih baik
6. Bapak/Ibu Dosen di lingkungan Program Studi Teknik Informatika, yang telah memberikan bekal ilmu, wawasan, dan pengalaman berharga selama penulis menempuh studi.
7. Orang tua tercinta, Ayah dan Ibu, yang tidak pernah berhenti mendoakan, menyemangati, dan memberikan dukungan moral maupun materiil. Tanpa kasih sayang dan pengorbanan Ayah dan Ibu, penulis tidak akan dapat sampai pada tahap ini.
8. Seluruh sahabat dan teman seperjuangan, khususnya teman-teman di Program Studi Teknik Informatika angkatan 20, yang selalu menjadi tempat berbagi cerita, berdiskusi, serta saling menguatkan dalam suka dan duka, mulai dari masa perkuliahan hingga penyusunan skripsi ini.
9. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah membantu dalam bentuk apa pun, baik secara langsung maupun tidak langsung.

Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari sempurna. Hal ini tidak terlepas dari keterbatasan pengetahuan, pengalaman, dan kemampuan yang dimiliki penulis. Oleh karena itu, penulis sangat mengharapkan kritik dan saran

yang membangun dari berbagai pihak demi penyempurnaan karya ini di masa yang akan datang. dan harapannya semoga skripsi ini dapat memberikan manfaat kepada pembaca.

Wassalamualaikum Wr. Wb.

Malang, 19 Desember 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Saputra, Alva. 2025. **Sistem Kontrol Kipas Angin Berbasis Kalimat Menggunakan Algoritma Colussi**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T, (II) Dr. Ir. Fachrul Kurniawan, M.MT, IPU.

Kata Kunci: Algoritma *Colussi*, Deteksi Suara, ESP32, *Internet of Things*, *Smart Home*.

Penelitian ini merancang dan mengimplementasikan sistem kontrol kipas angin berbasis perintah kalimat berbahasa Indonesia menggunakan algoritma *Colussi* sebagai metode pencocokan pola teks. Sistem terdiri dari tiga komponen utama, yaitu aplikasi Android berbasis MIT App Inventor yang berfungsi sebagai antarmuka pengguna sekaligus media input suara, backend API berbasis PHP yang dilengkapi library Sastrawi untuk proses *stemming* dan pencocokan pola kalimat, serta mikrokontroler ESP32 yang berfungsi sebagai pengendali kipas melalui modul relay dengan komunikasi protokol MQTT menggunakan broker HiveMQ. Sistem dirancang untuk mengenali perintah dasar seperti "nyalakan kipas" dan "matikan kipas", termasuk variasi sinonim dan struktur kalimat disertai negasi. Pengujian dilakukan sebanyak 60 percobaan, terdiri dari kalimat langsung (tanpa negasi), kalimat sopan, variasi struktur kalimat, serta kalimat negasi. Hasil pengujian menunjukkan bahwa sistem berhasil mengeksekusi perintah secara benar sebanyak 40 percobaan (66,7%), sementara 20 percobaan (33,3%) mengalami kesalahan interpretasi. Kesalahan dominan terjadi pada kalimat negasi seperti "jangan nyalakan kipas" atau "tolong tidak menghidupkan kipas", di mana sistem masih kesulitan membedakan maksud pembalikan perintah sehingga menghasilkan aksi yang berlawanan. Selain itu, sistem mampu merespons perintah dengan waktu rata-rata pemrosesan berkisar antara 578 – 9034 ms, bergantung pada kompleksitas kalimat dan stabilitas jaringan. Hasil penelitian menunjukkan bahwa algoritma *Colussi* memiliki potensi untuk diterapkan pada sistem kendali berbasis kalimat dalam Bahasa Indonesia, khususnya untuk pola kalimat sederhana dan langsung. Namun demikian, akurasi masih perlu ditingkatkan pada penanganan kalimat kompleks, terutama yang melibatkan negasi dan variasi semantik. Pengembangan lebih lanjut dapat dilakukan dengan menambahkan dataset kalimat yang lebih beragam, integrasi analisis semantik yang lebih dalam, atau penggabungan metode kecerdasan buatan untuk meningkatkan tingkat keberhasilan sistem.

ABSTRACT

Saputra, Alva. 2025. **Fan Control System Based on Sentences Using the Colussi Algorithm**. Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, State Islamic University Maulana Malik Ibrahim Malang. Supervisors: (I) Mr. Ajib Hanani, M.T (II) Dr. Fachrul Kurniawan, M.MT, IPU.

This research designs and implements a fan control system based on Indonesian sentence commands using the Colussi algorithm as a text pattern-matching method. The system consists of three main components, namely an Android application developed using MIT App Inventor which functions as a user interface as well as a voice input medium, a PHP-based backend API equipped with the Sastrawi library for stemming and sentence pattern matching, and an ESP32 microcontroller that controls the fan through a relay module using the MQTT communication protocol with HiveMQ as the broker. The system is designed to recognize basic commands such as "nyalakan kipas" (turn on the fan) and "matikan kipas" (turn off the fan), including synonym variations and sentence structures containing negation. Testing was conducted with 60 trials, consisting of direct commands (without negation), polite sentence commands, sentence structure variations, and negation commands. The test results show that the system successfully executed commands correctly in 40 trials (66.7%), while 20 trials (33.3%) resulted in misinterpretation. Most errors occurred in negation sentences such as "jangan nyalakan kipas" (do not turn on the fan) or "tolong tidak menghidupkan kipas" (please do not turn on the fan), where the system still had difficulty distinguishing the reversal intent of the command, resulting in the opposite action. In addition, the system is able to respond to commands with an average processing time ranging from 578 to 9034 ms, depending on sentence complexity and network stability. The results indicate that the Colussi algorithm has potential to be applied in sentence-based control systems in the Indonesian language, particularly for simple and direct sentence patterns. However, system accuracy still needs improvement in handling complex sentences, especially those involving negation and semantic variations. Further development can be carried out by adding a more diverse sentence dataset, integrating deeper semantic analysis, or combining artificial intelligence methods to improve the system's performance and success rate.

Keywords: Colussi Algorithm, ESP32, Internet of Things, Smart Home, Speech Recognition.

الملخص

سابوترا، ألفا. 2025 نظام التحكم في المروحة المعتمد على الجمل باستخدام خوارزمية كولوسي .رسالة بكالوريوس. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة الدولة الإسلامية مولانا مالك إبراهيم مالانغ. المشرفان: (الأول) الأستاذ أجيب حناني، ماجستير الهندسة (الثاني) الدكتور فحر الكرنياوان، ماجستير الهندسة، عضو معهد المهندسين المحترفين(IPU)

الكلمات المفتاحية: إنترنت الأشياء، المنزل الذكي، خوارزمية كولوسي، التعرف على الصوت, ESP32

صُمِّم هذا البحث ونُفِّذ نظامُ تحكمٍ بمروحة كهربائية يعتمد على أوامر الجمل باللغة الإندونيسية باستخدام خوارزمية Colussi بوصفها طريقةً لمطابقة أنماط النصوص. يتكوّن النظام من ثلاثة مكوّنات رئيسة، وهي: تطبيق أندرويد مبني باستخدام منصة MIT App Inventor يعمل كواجهة استخدام ووسيط لإدخال الصوت، وواجهة خلفية (Backend API) مبنية بلغة PHP ومزوّدة بمكتبة Sastrawi لإجراء عملية التجذير اللغوي (Stemming) ومطابقة أنماط الجمل، بالإضافة إلى متحكّم دقيق ESP32 يعمل كوحدة تحكم بالمروحة من خلال وحدة Relay باستخدام بروتوكول الاتصال MQTT عبر خادم HiveMQ. وقد صُمِّم النظام للتعرّف على الأوامر الأساسية مثل ''تشغيل المروحة'' و''إيقاف المروحة''، بما في ذلك المترادفات وتنوّعات البنية اللغوية المصحوبة بالنفي. تم إجراء الاختبارات بعدد 60 تجربة، شملت أوامر مباشرة دون نفي، وأوامر بأسلوب مهذّب،وتنوّعاً في تركيب الجملة، إضافةً إلى الجمل المنفية. وأظهرت نتائج الاختبار أنّ النظام نجح في تنفيذ الأوامر بشكل صحيح في 40 تجربة (66.7٪)، بينما حدثت أخطاء تفسير في 20 تجربة (33.3٪). وكانت معظم الأخطاء في الجمل التي تحتوي على صيغة النهي مثل: ''لا تُشغّل المروحة'' أو ''رجاءً لا تقم بتشغيل المروحة''، حيث واجه النظام صعوبة في فهم المقصود من عكس الأمر مما أدى أحيانًا إلى تنفيذ الإجراء المعاكس. كما أظهر النظام قدرة على الاستجابة للأوامر بزمن معالجة يتراوح بين 578 – 9034 مللي ثانية حسب تعقيد الجملة واستقرار الشبكة .وتشير نتائج البحث إلى أنّ خوارزمية Colussi تمتلك إمكانية جيدة للتطبيق في أنظمة التحكم المعتمدة على الأوامر النصية باللغة الإندونيسية، خصوصًا للجمل البسيطة والمباشرة. ومع ذلك، لا تزال هناك حاجة لتحسين الدقة في معالجة الجمل المعقّدة، خاصة تلك التي تحتوي على النفي والتنوّعات الدلالية. ويمكن تطوير العمل مستقبلاً من خلال توسيع مجموعة بيانات الجمل المستخدمة، ودمج تحليل دلالي أعمق، أو الاستفادة من تقنيات الذكاء الاصطناعي لرفع مستوى أداء النظام ودقته.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era perkembangan teknologi yang pesat, *Internet of Things* (IoT) telah menjadi salah satu pendorong utama dalam transformasi digital, terutama di ranah *smart home*. Konsep ini menghubungkan berbagai perangkat rumah tangga ke jaringan internet, memungkinkan interaksi dan kontrol yang lebih cerdas dan efisien (Anggoro, 2021). Dengan kemampuan untuk mengintegrasikan sensor, perangkat lunak, dan konektivitas internet, IoT memberikan peluang besar untuk meningkatkan fungsionalitas perangkat rumah tangga, termasuk kipas angin.

Kipas angin, sebagai salah satu perangkat rumah tangga yang umum digunakan, dapat menjadi lebih mudah dioperasikan melalui sistem kontrol berbasis IoT, terutama bagi pengguna yang mengalami keterbatasan fisik, seperti disabilitas gerak. Orang dengan disabilitas, terutama mereka yang memiliki keterbatasan dalam mobilitas, sering kali menghadapi kesulitan dalam mengontrol perangkat elektronik secara manual (Utomo et al., 2024). Untuk mengatasi hambatan ini, teknologi IoT memungkinkan perangkat dikendalikan dari jarak jauh melalui aplikasi ponsel pintar atau perintah suara.

Kipas angin konvensional memiliki beberapa keterbatasan, seperti pengoperasian manual yang merepotkan, pengaturan kecepatan yang terbatas, dan potensi pemborosan energi (Sudrajat & Rofifah, 2023). Keterbatasan ini mendorong pengembangan sistem kontrol kipas angin dengan berbasis IoT. Sistem ini menawarkan solusi inovatif untuk mengatasi keterbatasan kipas angin

konvensional. Pengguna dapat mengontrol kipas angin dari mana saja dengan mudah dan nyaman menggunakan smartphone. Sistem ini juga memungkinkan pengaturan yang lebih presisi untuk mendapatkan tingkat pendinginan yang diinginkan, serta membantu menghemat energi dengan mematikan kipas angin ketika tidak digunakan.

Pada penelitian ini penulis ingin menggunakan algoritma *Colussi* yang merupakan pengembangan dari algoritma *Knuth-Morris-Pratt* yang merupakan proses pencocokan *String* (Novia Intan Pratiwi, 2020). Penggunaan algoritma *Colussi* pada penelitian ini yaitu guna mencocokan semua *patern* yang diterima oleh sistem, sehingga mendapatkan kalimat yang sesuai dalam melakukan perintah melalui sensor suara seperti: nyalakan kipas angin, matikan kipas angin.

Algoritma *Colussi* yang awalnya dirancang untuk mempercepat pencocokan pola pada teks, diterapkan dalam sistem ini untuk meningkatkan efisiensi pencarian dan pengolahan data pada perintah kontrol. Algoritma ini dipilih karena kemampuannya yang cepat dan efisien dalam mengidentifikasi pola-pola spesifik dalam waktu singkat, yang relevan dalam konteks sistem pengenalan suara atau kontrol perintah berbasis teks.

Sistem kontrol kipas angin berbasis IoT merupakan solusi inovatif dan efisien untuk kebutuhan pendinginan ruangan di era modern. Sistem ini menawarkan berbagai keuntungan bagi pengguna, seperti kemudahan dan kenyamanan. Sistem ini juga memiliki potensi pengembangan yang luas untuk menghadirkan solusi yang lebih canggih dan adaptif di masa depan khususnya dalam membantu orang disabilitas yang mempunyai keterbatasan dalam bergerak.

Sebagaimana hadist yang diriwayatkan oleh Imam Muslim dari Abu Hurairah yang berbunyi:

مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَبِ الدُّنْيَا نَفَّسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرَبِ يَوْمِ الْقِيَامَةِ وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ يَسَّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالآخِرَةِ

"*Siapapun yang menyelesaikan kesulitan seorang mukmin dari berbagai kesulitan-kesulitan dunia, niscaya Allah akan memudahkan kesulitan-kesulitannya pada hari kiamat. Siapa yang memudahkan orang yang sedang kesulitan niscaya Allah mudahkan baginya di dunia dan akhirat*."

Imam Nawawi dalam bukunya *Riyadh as-Salihin* menjelaskan hadits tersebut, bahwa meringankan kesusahan muslim di dunia, akan menuai pahala berupa keringanan di hari kiamat, hari penuh kegentaran bagi seluruh umat manusia. Menolong mereka yang tengah tertimpa kesulitan, merupakan amal mulia yang dicintai Allah SWT (Riyadh As-Salihin,). Dengan saling membantu, kita dapat meringankan beban satu sama lain dan memperkuat ukhuwah Islamiyah. Ingatlah bahwa pahala dari Allah SWT sangatlah besar bagi mereka yang menolong saudaranya yang muslim.

1.2 Pernyataan Masalah

Permasalahan dalam penelitian ini adalah bagaimana cara mengontrol kipas angin berbasis *Internet of Things* (IoT) yang dapat dikendalikan melalui aplikasi Android melalui perintah suara, sehingga mampu memberikan kemudahan penggunaan khususnya bagi pengguna yang memiliki keterbatasan mobilitas.

1.3 Batasan Masalah

Agar penelitian ini terfokus dan tidak menyimpang dari tujuan yang telah ditetapkan, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Penelitian hanya berfokus pada integrasi sistem kendali kipas angin berbasis IoT, di mana pengendalian dilakukan secara jarak jauh menggunakan aplikasi Android berbasis MIT App Inventor yang terhubung dengan *backend* server dan modul ESP32.
2. Sistem hanya dirancang untuk mengenali perintah kalimat Bahasa Indonesia sederhana, khususnya perintah dasar untuk menyalakan dan mematikan kipas angin beserta beberapa variasi kalimat dan sinonimnya.
3. Sistem difokuskan pada fungsi kendali ON/OFF, sehingga tidak mencakup pengaturan tingkat kecepatan kipas angin atau fungsi otomatisasi lanjutan.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan mengimplementasikan sistem kontrol kipas angin berbasis *Internet of Things* (IoT) yang dapat dikendalikan melalui aplikasi Android menggunakan perintah suara, sehingga mampu memberikan kemudahan penggunaan, meningkatkan efektivitas pengoperasian perangkat, serta membantu pengguna yang memiliki keterbatasan mobilitas dalam mengontrol kipas angin secara lebih praktis dan efisien.

1.5 Manfaat Penelitian

Peneliti berharap hasil dari penelitian ini mampu memudahkan pekerjaan manusia dengan menerapkan *internet of things* yang bisa mengontrol kipas angin melalui smartphone android khususnya pada penderita divabel. Selain itu, dapat digunakan sebagai landasan penelitian selanjutnya dalam konsep sistem kontrol *smart home*.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Hasil dari penelitian terdahulu dapat dijadikan panduan untuk mempelajari lebih jauh kelebihan dan kekurangan komponen tertentu. Penelitian ini akan berhubungan dengan beberapa penelitian sebelumnya yang mencangkup penggunaan IoT dalam peralatan rumah khususnya kipas angin. Iyappan dkk., (2022) dalam jurnalnya ia mempelajari bagaimana sistem *smart home* dapat bekerja dengan perintah suara dengan menggunakan *google voice* yang terhubung dengan jaringan lokal ataupun dengan menggunakan *mobile data.* Hasil dari penelitian tersebut didapatkan bahwa perangkat rumah dapat bekerja sesuai dengan perintah yang di inputkan melalui *google voice*.

Dalam penelitian yang dilakukan oleh Arrosyidah dkk.(2023) yang berjudul "Pengembangan Sistem Kipas Otomatis Berbasis Internet Of Things (Iot) Menggunakan Sensor PIR Dan DHT11" berhasil membuat prototipe sistem otomatis yang dapat mengendalikan kipas angin berdasarkan suhu yang dideteksi oleh sensor pada jarak tertentu.

Penelitian yang dilakukan oleh Tjandra Setiati dkk., (2023) dengan judul "Sistem Kendali Kipas Angin Otomatis Dengan Sistem Monitoring Berbasis IoT" menjelaskan tentang permasalahan pada perangkat kipas angin yang diakibatkan oleh motor kipas angin yang terbakar akibat tidak terkontrolnya penggunaan kipas angin. Peneliti berhasil membuat sistem yang dapat mengontrol nyala dan

kecepatan kipas angin dengan sensor suhu DHT11 dan ESP8266 sebagai mikrokontroller melalui aplikasi android.

Dalam penelitian yang dilakukan oleh (Filipe et al., 2021)dengan judul "*Voice-Activated Smart Home Controller Using Machine Learning*" menjelaskan tentang penggunaan *Voice Controller* dalam peralatan rumah tangga dan bangunan rumah. Peneliti telah berhasil mengimplementasikan *Voice Controller* pada objek tirai yang dapat melakukan perintah membuka dan menutup tirai melalui *smart device*.

Dalam penelitian yang dilakukan oleh (Isyanto et al., 2020) yang berjudul "*Performance of Smart Personal Assistant Applications Based on Speech Recognition Technology using IoT-based Voice Commands*" menjelaskan tentang performa Sistem *Smart Personal Assistant* menghasilkan bahwa sistem yang menggunakan *Google Assistant* memiliki performa *reponding* 95% lebih tinggi dibandingkan *Apple Siri* yang memiliki performa *responding* 80%.

Perbedaan penelitian sebelumnya dengan penelitain yang dilakukan adalah metode yang digunakan, seperti yang terlihat di tabel 2.1.

Tabel 2.1 Perbedaan dengan Penelitian Terdahulu

No.	Peneliti (Tahun)	Metode	Hasil Penelitian	Perbedaan
1.	Iyappan dkk.(2022)	Voice Controller	Sistem dapat bekerja dengan menggunakan koneksi router wifi ataupun dengan menggunakan mobile data	Penelitian tersebut fokus pada otomatisasi rumah secara umum. Penelitian ini khusus pada kontrol kipas angin dengan algoritma *colussi*
2.	(Arrosyidah et al., n.d.)	Sensor Suhu	Kipas angin dapat bekerja secara otomatis sesuai dengan suhu yang terdeteksi dengan sensor suhu	Penelitian tersebut berfokus kepada sistem otomatisasi kipas angin, Penelitian ini fokus pada kontrol kipas angin atau algoritma pencocokan string *colussi*

3.	Tjandra Setiati dkk., (2023)	Monitoring system	Sistem dapat mengendalikan kipas angin dengan menggunakan sensor suhu dan perintah melalui smartphone	Tidak menggunakan algoritma *colussi* atau pencocokan string
4.	(Filipe et al., 2021)	Speech Recognition	Sistem dapat bekerja mengendalikan tirai melalui perintah suara.	Penelitian tersebut berfokus kepada sistem *smart home* secara umum khususnya pada tirai, Penelitian ini berfokus pada kontrol kipas angin
5.	(Isyanto et al., 2020)	Speech Recognition	Sistem *Smart Personal Assistant* yang menggunakan *Google Assistant* memiliki performa *reponding* 95% dibandingkan *Apple Siri* yang memiliki performa *responding* 80%	Penelitian tersebut berfokus pada pengujian performa sistem *Smart Personal Assistant* pada *Smart Home*, Penelitian ini berfokus untuk implementasi sistem kontrol kipas angin

2.2 *Internet Of Things* (IoT)

Internet of Things (IoT) merupakan konsep yang menggambarkan jaringan perangkat fisik yang terhubung satu sama lain melalui internet, memungkinkan mereka untuk mengumpulkan dan berbagi data. Perangkat ini bisa berupa sensor, perangkat rumah tangga, kendaraan, dan banyak lagi. IoT menggabungkan tiga dimensi utama: internet (jaringan global), things (benda fisik), dan semantik (informasi yang bermakna). Menurut (Atzori et al., 2010) arsitektur IoT umumnya terdiri dari beberapa lapisan: *Perception Layer* yang mencakup sensor dan aktuator untuk mengumpulkan data dari lingkungan fisik; *Network Layer* yang bertanggung jawab untuk mengirim data dari sensor ke pusat data atau *server cloud* menggunakan teknologi seperti Wi-Fi, Bluetooth, Zigbee, dan jaringan seluler; *Processing Layer* yang melibatkan analisis dan pemrosesan data di *server cloud* atau *edge computing devices*; dan *Application Layer* yang mencakup aplikasi dan

layanan yang digunakan oleh pengguna akhir seperti aplikasi *smart home*, kesehatan, atau transportasi. Beberapa teknologi kunci yang mendukung IoT meliputi sensor, komunikasi nirkabel, *cloud computing*, dan *edge computing*.

2.3 *Smart Home*

Smart Home atau rumah pintar adalah konsep rumah yang dilengkapi dengan perangkat dan sistem otomatis yang dapat dikendalikan dari jarak jauh melalui internet. Teknologi *smart home* memungkinkan pengguna untuk mengelola dan mengontrol berbagai aspek rumah mereka, seperti pencahayaan, keamanan, suhu, dan peralatan rumah tangga, dari perangkat mobile atau komputer(Vaidya & Vishwakarma, n.d.). Menurut (Balta-Ozkan et al., 2014), rumah pintar meningkatkan efisiensi energi, kenyamanan, dan keamanan penghuninya. Komponen utama dalam sistem *smart home* meliputi perangkat pintar seperti termostat pintar, lampu pintar, kamera keamanan, dan peralatan rumah tangga pintar; *hub* atau *gateway* yang berfungsi sebagai pusat kontrol yang menghubungkan dan mengendalikan berbagai perangkat pintar; aplikasi mobile atau web yang digunakan oleh pengguna untuk mengontrol dan memantau perangkat *smart home* dari jarak jauh, serta sensor dan aktuator untuk mendeteksi perubahan lingkungan dan mengambil tindakan(Hanani & Hariyadi, 2020). Manfaat utama dari penerapan teknologi smart home meliputi efisiensi energi, kenyamanan, keamanan, dan konektivitas. Namun, implementasi *smart home* juga menghadapi berbagai tantangan seperti keamanan dan privasi, kompatibilitas antara berbagai perangkat dari produsen yang berbeda, biaya awal yang tinggi, dan kompleksitas pengaturan dan konfigurasi.

2.4 Kipas Angin

Kipas angin adalah perangkat listrik yang memutar baling-baling untuk menghasilkan angin. Selain mendinginkan dan menyegarkan udara, kipas angin juga meningkatkan sirkulasi udara, menurunkan bahaya polusi udara dalam ruangan, dan berfungsi sebagai pengering. Di antara banyak manfaat kipas angin adalah biayanya yang rendah, kemudahan pemasangan, portabilitas, kurangnya kebutuhan perawatan, dan kemampuan untuk dibersihkan di rumah (Tjandra Setiati dkk., 2023).

2.5 *Speech recognition*

Speech recognition atau pengenalan suara adalah teknologi yang memungkinkan sistem komputer untuk menerima, memproses, dan memahami instruksi dalam bentuk ucapan manusia, kemudian menerjemahkannya menjadi teks atau perintah yang dapat dijalankan oleh perangkat. Perkembangan *speech recognition* semakin pesat seiring dengan meningkatnya kebutuhan interaksi manusia dengan mesin secara lebih natural, cepat, dan efisien. Teknologi ini banyak diaplikasikan pada asisten virtual, sistem otomasi rumah, penerjemah suara, hingga perangkat bantu bagi penyandang disabilitas(Setyawan & Ceng Giap, 2022).

Hal ini bertujuan untuk memberikan kemudahan kepada pengguna dalam aktivitas sehari-hari(Ritonga et al., 2019). Terutama orang-orang yang sudah lanjut usia ataupun yang mengalami cacat fisik. Dan untuk mengatasi berbagai masalah yang terjadi pada peralatan elektronik seperti tombol ON/OFF yang rusak yang dapat mengakibatkan pemborosan energi listrik.

2.6 Algoritma *String Matching*

Algoritma *String matching* adalah sebuah teknik dalam ilmu komputer dan teori algoritma yang digunakan untuk mencari kemunculan *substring* (pola) dalam sebuah teks (Raihan Azis et al., n.d.). Metode ini memiliki berbagai aplikasi dalam ilmu komputer, seperti dalam pemrosesan teks, pencarian informasi, bioinformatika, dan analisis data. Algoritma *string matching* berfungsi untuk menemukan satu atau beberapa lokasi di mana pola tersebut muncul dalam teks(Khancome & Boonjing, 2012). Rumus dasar dari algoritma *string matching* adalah:

1. Pencocokan Pola (*Pattern Matching*): Algoritma mencari kecocokan pola dalam teks dengan mencocokkan pola karakter demi karakter.
2. *Prefix Function* (Fungsi Awalan): Algoritma menggunakan fungsi awalan (*prefix function*) untuk menentukan perpindahan yang optimal saat pencocokan pola dalam teks. Fungsi ini mengidentifikasi prefiks terpanjang dari pola yang juga merupakan sufiks dari pola tersebut.

String matching adalah teknik penting dalam ilmu komputer dengan berbagai aplikasi praktis. Berbagai algoritma string matching telah dikembangkan untuk memenuhi kebutuhan berbagai konteks pencarian, masing-masing dengan kekuatan dan kelemahannya sendiri. Pemilihan algoritma yang tepat tergantung pada karakteristik teks dan pola serta kebutuhan spesifik dari aplikasi.

2.7 Algoritma *Colussi*

Algoritma *Colussi* merupakan salah satu dari sekian banyak algoritma yang digunakan untuk pencocokan string. Algoritma ini dirancang oleh Alberto Apostolico dan Alessandro Colussi pada tahun 1990 dan memperkenalkan metode baru untuk mempercepat proses pencarian pola dalam teks(Bardadi, 2019). Berbeda dengan algoritma pencocokan string tradisional seperti *Knuth-Morris-Pratt* (KMP) dan *Boyer-Moore*. Algoritma *Colussi* menggabungkan elemen-elemen dari kedua metode tersebut untuk mencapai efisiensi yang lebih tinggi dalam pencocokan string.

Algoritma *Colussi* menggunakan pendekatan yang mirip dengan KMP dalam hal *preprocessing* pola dan menggunakan tabel penyesuaian yang mirip dengan *shift table* pada algoritma *Boyer-Moore* (Mabrouk et al., 1996). Dengan demikian, algoritma ini dapat melompati beberapa karakter dalam teks yang sedang dicocokkan, yang menghasilkan waktu pencocokan yang lebih cepat dalam banyak kasus praktis.

Algoritma Colussi bekerja dengan membagi pola yang dicari menjadi beberapa bagian dan memproses masing-masing bagian tersebut secara terpisah untuk mempercepat pencarian. Algoritma ini melakukan pencocokan dengan memanfaatkan informasi tentang struktur internal pola untuk meminimalkan jumlah karakter teks yang perlu diperiksa.

Proses pencocokan dimulai dengan preprocessing pola untuk membuat tabel penyesuaian. Setelah itu, algoritma mulai mencocokkan pola dengan teks menggunakan tabel tersebut untuk menentukan berapa banyak karakter yang bisa

dilewati saat terjadi ketidakcocokan. Proses ini memungkinkan pencarian yang lebih cepat dengan mengurangi jumlah perbandingan karakter yang harus dilakukan.

2.8 Node MCU ESP32

NodeMCU ESP32 adalah modul mikrokontroler yang menggabungkan fitur Wi-Fi dan Bluetooth, menjadikannya pilihan populer untuk proyek *Internet of Things* (IoT). Modul ini memiliki kemampuan pemrosesan yang kuat dan kompatibilitas dengan berbagai sensor dan aktuator, memungkinkan pengembangan aplikasi IoT yang kompleks dengan konektivitas nirkabel(Aghenta & Iqbal, 2019).

2.9 MIT App Inventor

MIT App Inventor adalah platform pembuatan aplikasi berbasis *blockchain* yang memungkinkan pengguna membuat aplikasi Android dengan mudah tanpa memiliki pengalaman pemrograman yang luas. Platform ini memungkinkan pengguna untuk membuat aplikasi Android secara visual menggunakan konsep *block-based* programming. Dengan antarmuka *drag and drop*, Pengguna dapat memanfaatkan antarmuka sederhana tanpa perlu menulis kode dalam bahasa pemrograman konvensional (Zulkarnain et al., 2024). Salah satu contoh implementasinya adalah merancang aplikasi yang menggunakan MQTT untuk mengontrol LED dan menampilkan data sensor.

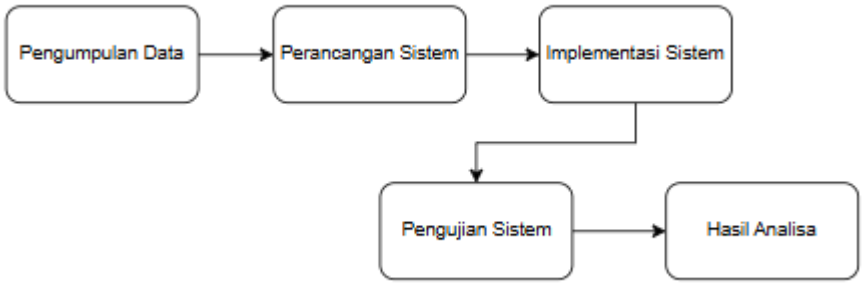
2.10 MQTT

MQTT adalah protokol komunikasi ringan yang ditujukan untuk perangkat dengan *bandwidth* dan latensi rendah. Protokol ini menggunakan pendekatan *publish* dan *subscribe*, di mana perangkat (klien) dapat mengirim atau menerima pesan melalui server yang dikenal sebagai broker(Mishra & Kertesz, 2020). MQTT sangat ideal untuk aplikasi IoT karena kemampuannya mengelola data dalam jumlah kecil dan berkomunikasi secara real time.

BAB III

DESAIN PENELITIAN

Bab ini membahas perancangan sistem kontrol kipas angin yang dikendalikan melalui perintah kalimat dan diproses menggunakan algoritma *Colussi*. Tujuan dari bab ini adalah untuk menjelaskan rancangan arsitektur sistem, desain perangkat keras dan perangkat lunak, serta alur data yang terjadi selama sistem berlangsung. Sistem ini mengintegrasikan teknologi *Internet of Things* (IoT), pengenalan suara, dan pencocokan pola berbasis teks dengan menggunakan algoritma *Colussi*.



Gambar 3. 1 Desain Penelitian

Tahapan pertama dalam pengembangan sistem ini adalah melakukan studi pustaka yang berkaitan dengan algoritma pencocokan teks serta teknologi sistem IoT. Studi ini bertujuan untuk memperoleh landasan teori yang kuat tentang berbagai metode string matching, termasuk algoritma *Colussi*, serta cara kerja komunikasi perangkat IoT yang dapat dikendalikan dari jarak jauh. Dari studi tersebut, dipilihlah pendekatan berbasis kalimat dan algoritma *Colussi* sebagai

metode utama untuk mencocokkan perintah pengguna terhadap perintah yang telah dikenali sistem.

Selanjutnya, dilakukan analisis kebutuhan perangkat keras dan perangkat lunak. Pada tahap ini, ditentukan komponen yang diperlukan dalam pembangunan sistem seperti mikrokontroler ESP32, modul relay, dan kipas angin sebagai perangkat output. Di sisi perangkat lunak, dibutuhkan aplikasi mobile berbasis MIT App Inventor, server API untuk pemrosesan kalimat, serta broker MQTT sebagai media komunikasi antara aplikasi dan perangkat fisik. Analisis ini penting untuk memastikan semua komponen mampu saling terintegrasi secara optimal.

Tahap berikutnya adalah perancangan sistem, yang meliputi penggambaran alur kerja sistem secara menyeluruh, mulai dari input suara yang diubah menjadi teks, pengolahan kalimat, hingga pengiriman perintah kendali ke perangkat melalui MQTT. Desain ini juga memperhitungkan integrasi algoritma *Colussi* dan penanganan kalimat yang mengandung negasi seperti “jangan” atau “tidak”, agar sistem dapat memahami konteks perintah dengan benar.

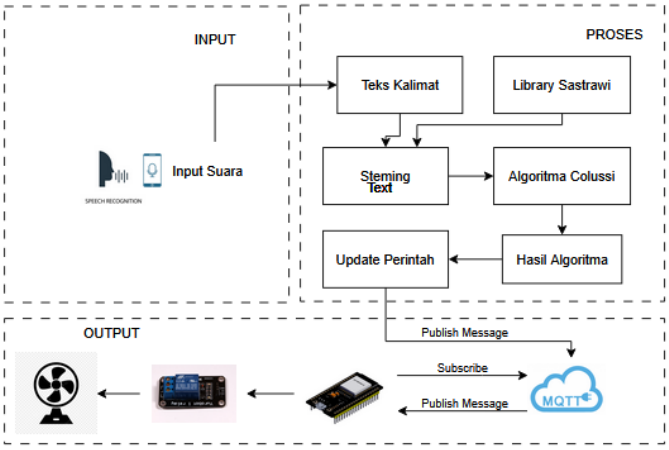
Setelah desain siap, dilakukan implementasi sistem. Perangkat keras dirakit dan dikonfigurasi, aplikasi MIT App Inventor dibangun untuk mengirimkan perintah suara, dan backend API dikembangkan untuk memproses teks perintah dari pengguna. Integrasi algoritma *Colussi* dilakukan di sisi *backend*, yang bertugas mencocokkan kalimat hasil pengenalan suara dengan pola-pola perintah yang telah ditentukan, seperti “nyalakan kipas” atau “matikan kipas”. Hasil pencocokan tersebut menentukan apakah sistem akan memberikan perintah ON atau OFF ke kipas angin.

Setelah sistem terintegrasi, dilakukan pengujian sistem untuk mengetahui sejauh mana sistem dapat mengenali dan merespons perintah pengguna dengan benar. Pengujian ini juga mencakup skenario kalimat dengan berbagai variasi bentuk dan kata negasi untuk memastikan sistem dapat membedakan konteks yang sebenarnya. Akhir dari tahapan ini adalah melakukan analisis hasil, yaitu mengevaluasi tingkat keberhasilan sistem dalam mengenali perintah dan keefektifan penggunaan algoritma *Colussi* dalam proses pencocokan teks.

3.1 Analisis dan Perancangan

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. Pengguna melakukan kontrol dengan menggunakan suara, kemudian digunakan Google Speech Recognition untuk mengolah data pengenalan suara menjadi bentuk teks kemudian memulai proses stemming menggunakan Library Sastrawi.

Kemudian proses pencocokan *pattern* dilakukan dengan menggunakan algoritma *Colussi*. Setelah *pattern* teridentifikasi, maka akan dilakukan proses penerjemahan teks perintah menjadi data perintah, Selanjutnya data dikirim ke MQTT *(Publish)*, dan terakhir NodeMCU ESP32 akan dikendalikan dengan perintah data untuk menyalakan atau mematikan kipas angin dengan menggunakan modul *relay* seperti yang ditunjukkan pada gambar 3.1.



Gambar 3. 2 Desain Sistem

Gambar tersebut menggambarkan arsitektur sistem kendali kipas angin berbasis *Internet of Things* (IoT) dengan perintah suara. Sistem dimulai dari input suara pengguna yang ditangkap melalui aplikasi mobile yang dibangun menggunakan MIT App Inventor. Suara tersebut diubah menjadi teks melalui fitur *Speech Recognition* yang tersedia di dalam aplikasi. Kalimat hasil pengenalan suara kemudian dikirim ke server lokal yang menjalankan API untuk pengolahan lebih lanjut dengan menggunakan *ngrok* supaya server dapat diakses melalui internet.

Setelah kalimat diterima oleh server, proses dilanjutkan dengan stemming menggunakan library Sastrawi, yaitu proses untuk mengubah kata berimbuhan menjadi kata dasarnya. Misalnya, kata “menyalakan” akan diubah menjadi “nyala” dan “mematikan” menjadi “mati”. Tahapan ini penting agar sistem dapat memahami inti perintah dari pengguna secara konsisten, meskipun disampaikan dalam berbagai bentuk kalimat.

Hasil stemming kemudian diproses menggunakan algoritma *string matching Colussi*. Algoritma ini digunakan untuk mencocokkan teks perintah

dengan pola yang telah ditentukan sebelumnya, seperti “nyala kipas” atau “mati kipas”. Sistem juga dilengkapi dengan logika tambahan untuk mendeteksi kata negasi, seperti “jangan”, “tidak”, atau “nggak”. Jika ditemukan kata negasi, maka perintah yang dikenali akan dibalik; misalnya, “jangan nyalakan kipas” akan dipahami sebagai perintah untuk tidak menghidupkan kipas.

Setelah keputusan perintah ditentukan, sistem akan mengirimkan pesan melalui protokol MQTT ke broker HiveMQ. Pesan ini berisi instruksi ON atau OFF dan dipublikasikan ke topik tertentu, misalnya iot/kipas. Di sisi output, sebuah mikrokontroler ESP32 yang terhubung dengan modul relay dan kipas angin akan *subscribe* ke topik tersebut. Ketika menerima pesan dari broker, ESP32 akan mengaktifkan atau menonaktifkan kipas sesuai dengan perintah yang diterima.

Secara keseluruhan, sistem ini menggabungkan teknologi *speech recognition*, *natural language processing* dengan Sastrawi, algoritma pencocokan string Colussi, dan komunikasi IoT berbasis MQTT untuk menghasilkan sistem kendali perangkat rumah tangga yang cerdas dan responsif terhadap perintah suara manusia.

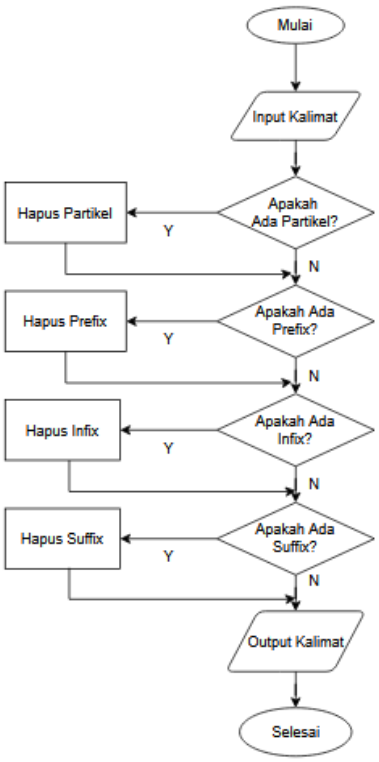
3.1.1 Proses Pengambilan Data

Data yang digunakan dalam penelitian ini terdiri dari data primer dan data sekunder. Data primer diperoleh dari hasil input suara pengguna melalui fitur *speech recognition* Google Speech dalam Bahasa Indonesia, yang kemudian dikonversi menjadi teks berupa kalimat perintah seperti “nyalakan kipas”, “matikan kipas”, “hidupkan kipas”, dan “padamkan kipas”, termasuk variasi kalimat dengan kata bantu serta larangan seperti “tolong nyalakan kipas”, “tolong matikan kipas”,

dan “jangan nyalakan kipas”. Sementara itu, data sekunder berupa data pattern kata kunci yang ditentukan oleh peneliti sebagai acuan pencocokan pola, seperti pattern “nyala kipas”, “mati kipas”, “hidup kipas”, dan “padam kipas” yang digunakan dalam proses pengenalan perintah oleh sistem.

3.1.2 Proses *Stemming*

Stemming adalah proses pengubahan kalimat atau kata yang berimbuhan menjadi kata baku (Rifai & Winarko, 2019). 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. 3 Proses Stemming

Dalam konteks pemrosesan bahasa alami dan algoritma *string matching*, *stemming* membantu mencocokkan kata dengan bentuk dasar yang sama, meskipun

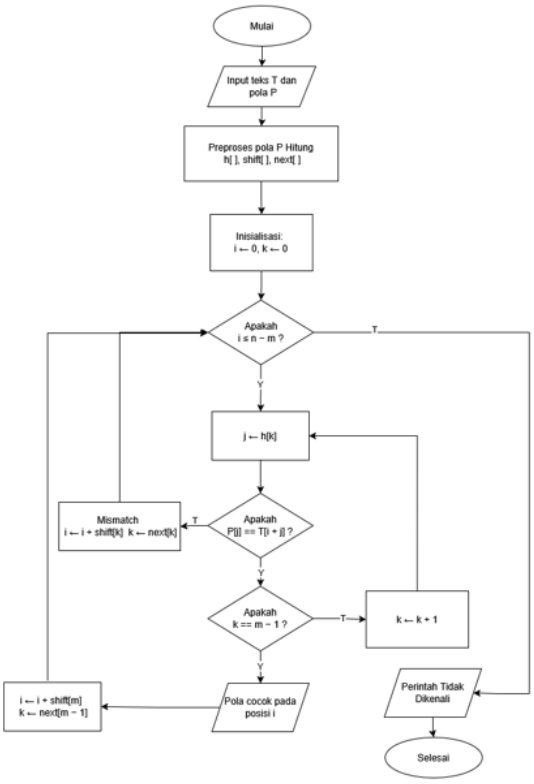
kata tersebut muncul dalam berbagai variasi imbuhan(Chafid dkk., 2020). Proses ini sangat penting untuk meningkatkan efisiensi dan akurasi pencarian teks. Berikut langkah-langkah proses *stemming*:

1. Identifikasi Awalan (*Prefix*): Awalan adalah imbuhan yang muncul di depan kata dasar. Dalam bahasa Indonesia, beberapa contoh awalan adalah "me-", "di-", "ber-", dan "ter-". Pada kalimat "Tolong nyalakan kipas angin", kata "nyalakan" memiliki awalan "me-", yang ditandai oleh awalan "ny-" dalam bentuk singkatannya. Algoritma stemming akan menghilangkan awalan ini sehingga menjadi kata dasar "nyala".
2. Penghapusan Akhiran (*Suffix*): Akhiran adalah imbuhan yang terletak di akhir kata dasar. Pada contoh “nyalakan”, terdapat akhiran "-kan" yang menunjukkan bahwa kata tersebut merupakan bentuk perintah. Algoritma *stemming* akan menghapus akhiran ini untuk mendapatkan kata dasar, yaitu “nyala”.
3. Identifikasi Partikel: Partikel merupakan kata yang tidak mengubah arti dasar kata tetapi menambahkan nuansa tertentu dalam kalimat. Contoh partikel dalam bahasa Indonesia termasuk "lah", "pun", dan "kah". Walaupun kalimat "Tolong nyalakan kipas angin" tidak mengandung partikel, jika ada partikel dalam kalimat lain, algoritma *stemming* akan menghapusnya.
4. Penghapusan Sisipan (*Infix*): *Infix* adalah imbuhan yang disisipkan di dalam kata dasar. Pada kalimat “Tolong nyalakan kipas angin”, tidak terdapat sisipan. Namun, pada kata lain seperti "gemetar", *infix* "-em-" akan dihilangkan sehingga menghasilkan kata dasar "getar". *Infix* jarang digunakan dalam bahasa

Indonesia, namun pada beberapa kata, ini penting untuk dikenali oleh algoritma.

3.1.3 Algoritma *Colussi*

Algoritma *Colussi* adalah algoritma pencocokan string yang menggabungkan konsep dari beberapa algoritma lain, seperti *Knuth-Morris-Pratt* (KMP) dan *Boyer-Moore*. Algoritma ini dirancang untuk mempercepat pencarian pola dalam teks dengan mengurangi jumlah perbandingan karakter yang diperlukan. Algoritma ini terdiri dari dua tahap utama: *preprocessing* dan pencocokan.



Gambar 3. 4 Proses Algoritma Colussi

Diagram alir pada Gambar 3.4 menunjukkan tahapan kerja algoritma Colussi dalam mengendalikan kipas angin berdasarkan kalimat perintah. Proses diawali dengan memasukkan teks perintah dan pola, kemudian dilakukan preprocessing pola untuk membentuk tabel $h[k]$, $shift[k]$, dan $next[k]$. Selanjutnya, algoritma membandingkan karakter pola dengan teks sesuai urutan indeks pada $h[k]$. Jika terjadi ketidakcocokan, pola digeser berdasarkan nilai $shift[k]$ dan pemeriksaan dilanjutkan dari indeks $next[k]$. Apabila seluruh karakter pola cocok, sistem mengeksekusi keluaran berupa perintah ON jika pola yang ditemukan termasuk kategori nyalakan kipas, atau OFF jika termasuk kategori matikan kipas. Jika sampai akhir teks tidak ditemukan pola yang sesuai, sistem memberikan keluaran berupa “perintah tidak dikenali” dan tidak melakukan perubahan status pada kipas.

Untuk memastikan algoritma bekerja sesuai tujuan, langkah pertama adalah mengidentifikasi pola perintah suara menjadi kategori positif atau negatif. Pola Positif yaitu Pola yang menunjukkan tindakan "aktif", seperti "nyalakan" atau "hidupkan," di mana pengguna menginginkan kipas untuk menyala. Pola Negatif yaitu Pola yang menunjukkan tindakan pencegahan atau penolakan, seperti "jangan" atau "tidak," di mana pengguna tidak ingin kipas dinyalakan atau malah meminta kipas dimatikan. Dengan pemetaan ini, sistem bisa mengelompokkan dan memahami konteks perintah sesuai dengan intensi pengguna.

1. Tahap *Preprocessing* Pola

Pada tahap ini, pola P yang akan dicari diolah terlebih dahulu untuk membentuk tiga buah tabel utama, yaitu:

A. Tabel *h[k]*.

Tabel ini menyimpan urutan indeks karakter pola yang akan diperiksa. Artinya, algoritma tidak lagi memeriksa karakter pola secara berurutan dari indeks 0, 1, 2, dan seterusnya, tetapi mengikuti urutan indeks yang telah dioptimalkan di dalam tabel *h*. Secara konseptual, inti pemeriksaan pola terhadap teks dinyatakan sebagai (Charras & Lecroq, n.d.):

$$\boldsymbol{P[h[k]] == T[i + h[k]]} \qquad \textbf{(3.1)}$$

dengan:

i adalah posisi awal jendela pola pada teks,
h[k] adalah indeks karakter pola yang diperiksa pada langkah ke-*k*

B. Tabel *shift*

Tabel ini menyimpan besar pergeseran pola ketika terjadi ketidakcocokan (*mismatch*) atau setelah ditemukan kecocokan penuh. Nilai *shift[k]*, menyatakan berapa banyak karakter teks yang boleh "dilewati" ketika terjadi mismatch pada pemeriksaan ke-*k*. Pergeseran posisi jendela pada teks dapat dituliskan sebagai (Charras & Lecroq, n.d.):

$$\boldsymbol{i = i + shift[k]} \qquad \textbf{(3.2)}$$

C. Tabel *next[k]*

Selain menggeser pola, algoritma juga menentukan dari indeks urutan mana dalam tabel *h[k]* pemeriksaan berikutnya harus dilanjutkan. Nilai ini disimpan dalam tabel *next[k]*. Setelah jendela digeser, indeks urutan pemeriksaan diperbarui sebagai (Charras & Lecroq, n.d.):

$$\boldsymbol{k = next[k]} \qquad \textbf{(3.3)}$$

Dengan adanya tabel ini, algoritma tidak perlu kembali memulai pemeriksaan dari karakter pola pertama setiap kali terjadi *mismatch*, tetapi langsung melompat ke posisi yang masih berpotensi menghasilkan kecocokan.

2. Tahap Pencocokan

Setelah tahap *preprocessing* selesai, algoritma *Colussi* memasuki tahap pencocokan pola dengan teks hasil *speech recognition* dan *stemming*. Tahap ini dilakukan dengan langkah-langkah berikut:

A. Inisialisasi

Variabel posisi jendela pada teks diinisialisasi dengan $i = 0$: Menunjukkan posisi awal dalam teks. sedangkan indeks urutan pemeriksaan pola diinisialisasi dengan $k = 0$. Selama $i \leq n - m$ (di mana n adalah panjang teks dan m adalah panjang pola), algoritma akan terus mencoba mencocokkan pola pada posisi tersebut.

B. Proses pemeriksaan karakter sesuai urutan *h*

Pada setiap langkah, algoritma mengambil indeks pola $j = h[k]$ dan membandingkan karakter pola dengan teks dengan rumus (3.1) P[h[k]]=T[i+h[k]]. Jika sama, maka indeks urutan pemeriksaan dinaikkan menjadi $k = k + 1$. Jika k telah mencapai m, berarti seluruh karakter pola sudah cocok dan pola dinyatakan ditemukan pada posisi i.

C. Penanganan mismatch

Jika terjadi mismatch pada perbandingan $P[h[k]]$ dan $T[i+h[k]]$, algoritma tidak langsung kembali ke awal, melainkan menggeser jendela pola dengan menggunakan rumus (3.2): $i = i+shift\ [k]$, mengatur ulang indeks urutan

pemeriksaan dengan rumus (3.3): $k = next[k]$. Dengan mekanisme ini, karakter teks yang pasti tidak mungkin menjadi awal kecocokan tidak akan diperiksa ulang, sehingga menghemat jumlah perbandingan.

D. Penanganan kecocokan penuh

Jika seluruh karakter pola telah cocok (misalnya ketika $k = m$), posisi i saat itu dicatat sebagai lokasi di mana pola ditemukan. Setelah itu, algoritma tetap dapat mencari kecocokan berikutnya dengan: menggeser pola menggunakan nilai $shift[m]$, mengatur indeks pemeriksaan ke $next[m]$.

Dalam sistem kontrol kipas angin yang dikembangkan, hasil akhir dari proses pencocokan ini adalah keputusan apakah kalimat yang diucapkan pengguna mengandung pola perintah “nyala kipas/hidup kipas” (yang akan diterjemahkan menjadi perintah ON) atau “mati kipas/padam kipas” (yang akan diterjemahkan menjadi perintah OFF). Jika terdapat kata negasi seperti “jangan” atau “tidak”, sistem menambahkan logika terpisah untuk membalik atau meniadakan aksi yang semula diusulkan oleh hasil pencocokan pola.

Untuk mempermudah pemahaman, pada bagian ini diberikan contoh perhitungan tabel algoritma *Colussi* menggunakan pola sederhana yang relevan dengan sistem, yaitu P="NYALAKIPAS", Pola tersebut memiliki panjang $m = 10$ dengan indeks karakter seperti pada Tabel 3.1.

Tabel 3.1 Contoh Pola

Indeks (i)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Karakter (P[i])	N	Y	A	L	A	K	I	P	A	S

Berdasarkan pola ini, tahap preprocessing algoritma *Colussi* akan menyusun urutan pemeriksaan pola ke dalam array $h[k]$. Urutan h $[k]$ menentukan karakter pola mana yang akan diperiksa terlebih dahulu, kedua, dan seterusnya, sesuai dengan rumus umum (3.1) *P[h[k]]==T[i+h[k].* Hasil penyusunan urutan pemeriksaan $h[k]$ serta tabel pergeseran $shift$ $[k]$ dan tabel lanjutan $next$ $[k]$ untuk pola "NYALAKIPAS" dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh tabel h[k], shift[k], dan next[k]

k	**h[k]**	**Karakter (P[h[k]])**	**shift[k]**	**next[k]**
0	1	Y	1	0
1	2	A	2	0
2	3	L	3	0
3	4	A	4	0
4	5	K	5	0
5	6	I	6	0
6	7	P	7	0
7	8	A	8	0
8	9	S	9	0
9	0	N	0	9

Dari Tabel 3.2 dapat dijelaskan bahwa:

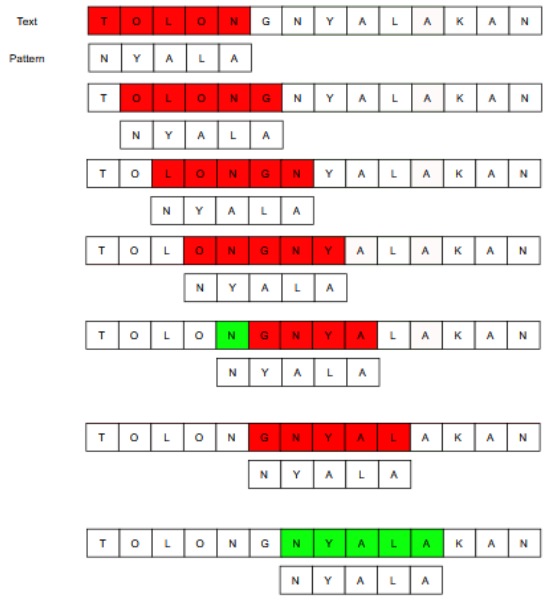
1. Kolom $h[k]$ menunjukkan indeks karakter pola yang akan diperiksa pada langkah ke-k . Misalnya, saat k = 0, algoritma membandingkan karakter P [h[0]] = P[1] = $'$ Y $'$ dengan karakter teks pada posisi T [i + 1].

2. Kolom $shift[k]$ menyatakan seberapa jauh pola digeser ketika terjadi mismatch pada langkah ke- k. Misalnya, jika terjadi mismatch saat $k = 2$, posisi jendela pola di teks diperbarui menjadi: $i = i + shift[2]$.
3. Kolom $next[k]$ menyatakan indeks urutan baru yang digunakan setelah pergeseran. Pada contoh ilustratif ini, semua nilai $next[k] = 0$, yang berarti setelah pergeseran pola, pemeriksaan kembali dimulai dari urutan pertama ($k = 0$).

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 gambar 3.3

Teks : TOLONG NYALAKAN

Pattern : NYALA



Gambar 3. 5 Proses Pencocokan

Pencocokan di atas merupakan proses pencarian dengan teks “TOLONG NYALAKAN” dan Pattern “NYALA”, yang mana ketika kata “NYALA” pada teks ditemukan yaitu dalam indeks 6, sehingga akan memerintahkan kipas angin untuk menyala, begitu juga ketika yang ditemukan pada teks kata “MATI” maka kipas angin akan mati.

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 3.4

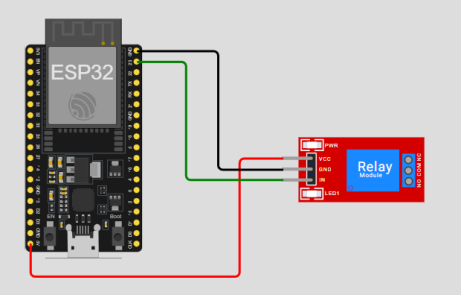
Gambar 3. 6 Rancangan Aplikasi

Penjelasan Gambar:

A. Status kipas angin.
B. Hasil suara diubah menjadi teks
C. Tombol rekam suara

3.3 Rancangan Komponen

Rangkaian komponen yang digunakan dalam penelitian dapat dilihat pada gambar 3.5. antara lain: NodeMCU ESP32, Relay 5v, konektor dan Motor kipas angin.



Gambar 3. 7 Rancangan Hardware

NodeMCU ESP32 digunakan sebagai otak yang mengatur Relay 5v, kemudian pada masing- masing port Relay 5v akan disambungkan pada kabel yang terdapat pada motor kipas angin.

3.4 Pengujian Sistem

Pengujian ini bertujuan untuk mengevaluasi seberapa besar error sistem yang dihasilkan dalam menjalankan fungsi-fungsi yang diharapkan. Berikut rumus perhitungan rata-rata error dalam setiap pengujian sistem yang ditunjukkan pada persamaan 3.4.

$$\boldsymbol{\mu = \frac{\sum Ei}{n} * 100\%} \qquad \textbf{(3.4)}$$

Keterangan:
μ : rata-rata error dalam bentuk persen (%)
n : jumlah percobaan
ΣEi : jumlah kesalahan dari semua percobaan

Rumus ini pada dasarnya merupakan bentuk umum dari metrik *Mean Percentage Error* (MPE) atau sering juga disebut *Mean Absolute Percentage Error* (MAPE) dalam evaluasi sistem. Konsep dasar dari MPE/MAPE adalah menghitung rata-rata kesalahan relatif pada setiap percobaan, kemudian dinyatakan dalam bentuk persentase(Ahmar, 2023). Dengan demikian, peneliti dapat mengetahui seberapa besar tingkat ketidaksesuaian hasil pengenalan sistem terhadap perintah yang diberikan pengguna dalam bentuk kalimat.

Proses pengujian pada penelitian ini dilakukan sebanyak 60 kali dengan merekap keberhasilan atau kegagalan di setiap percobaan. Dalam setiap uji coba, sistem dianggap berhasil jika perintah suara dikenali dan dijalankan dengan benar. Dan sisten dianggap gagal jika terjadi kesalahan dalam pengenalan atau pelaksanaan perintah.

BAB IV

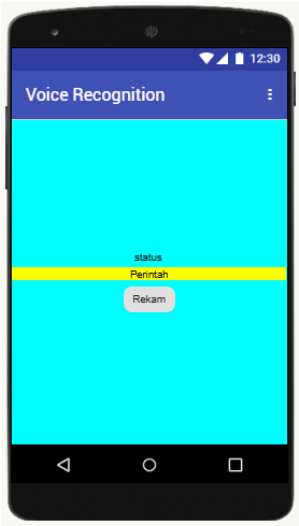
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada tahap implementasi sistem yang dibuat adalah hasil dari perancangan yang telah dibuat sebelumnya. Sistem kontrol ini terdiri dari dua utama, yaitu *software* yang mencakup aplikasi berbasis Android yang dilengkapi dengan fitur *speech recognition* untuk menerima perintah suara dari pengguna serta *hardware* yang mencangkup kipas angin dan beberapa komponen lainya.

4.1.1. Halaman Aplikasi

Pada halaman aplikasi terdapat *button* rekam yang memiliki fungsi untuk merekam perintah, serta terdapat teks hasil suara yang berfungsi untuk menampilkan teks perintah yang telah direkam. Tapilan halaman aplikasi dapat dilihat di gambar 4.1.



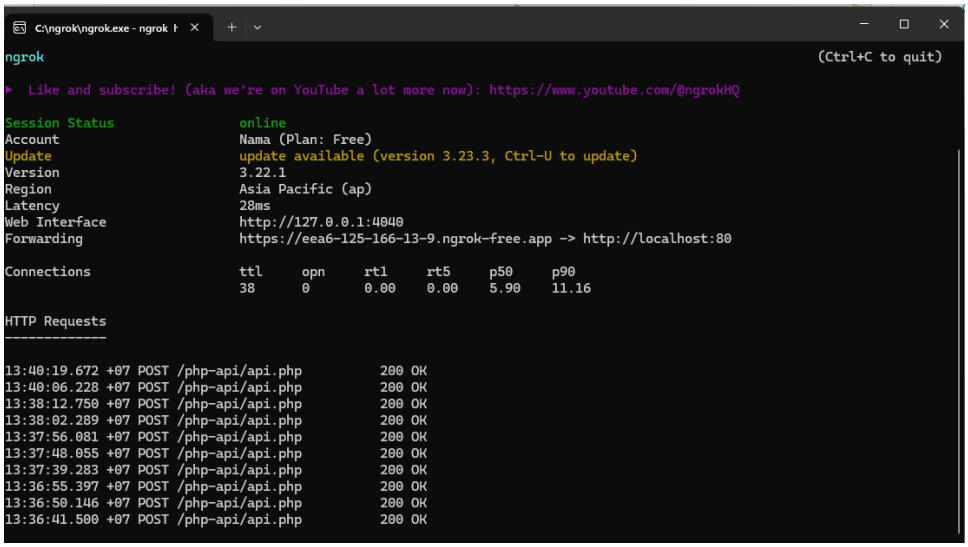
Gambar 4. 1 Tampilan aplikasi

Gambar di atas menampilkan tampilan antarmuka pengguna (*user interface*) dari aplikasi mobile berbasis MIT App Inventor. Aplikasi ini berfungsi sebagai media input perintah suara dari pengguna untuk mengontrol kipas angin. Tampilan sederhana ini menunjukkan adanya beberapa elemen penting, yaitu label status, label perintah, dan sebuah tombol "Rekam" di tengah layar. Warna latar belakang cerah memberikan kontras yang baik untuk keterbacaan teks, dan tampilan minimalis memudahkan penggunaan oleh pengguna.

Ketika pengguna menekan tombol "Rekam", aplikasi akan mengaktifkan fitur pengenalan suara (*speech recognition*) yang secara otomatis menangkap suara pengguna. Hasil dari pengenalan suara ini kemudian diubah menjadi teks dan ditampilkan pada bagian "Perintah". Teks tersebut selanjutnya dikirimkan ke server atau API backend untuk diproses menggunakan algoritma pencocokan *string Colussi*, guna mengenali maksud perintah untuk menyalakan atau mematikan kipas angin.

4.1.2. Implementasi Komunikasi IoT

Pada tahap ini, implementasi komunikasi *Internet of Things* (IoT) dilakukan untuk menjembatani pertukaran data antara aplikasi pengendali berbasis MIT App Inventor, server API yang memproses perintah kalimat, dan perangkat keras ESP32 yang berfungsi sebagai pengendali kipas angin. Proses komunikasi ini mengandalkan broker MQTT sebagai protokol utama dan platform ngrok untuk memberikan akses server secara publik.



Gambar 4. 2 Koneksi Ngrok

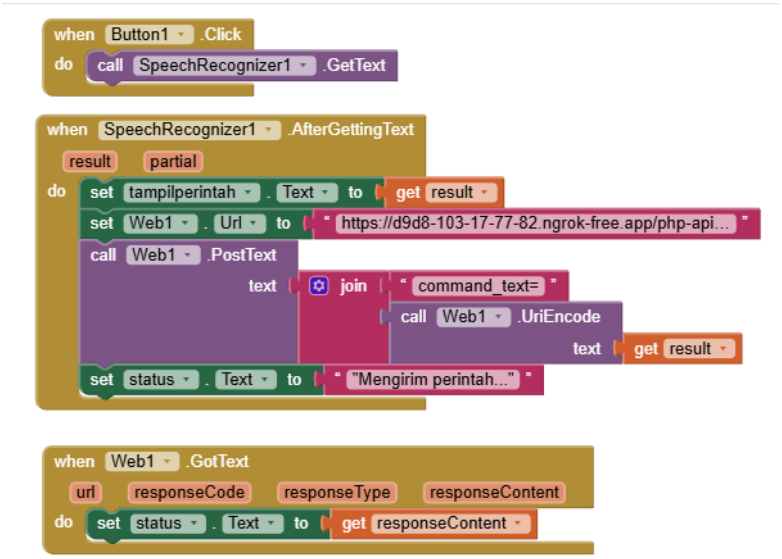
Gambar di atas menunjukkan tampilan debug terminal ngrok yang menampilkan status koneksi tunneling. Pada bagian informasi session, terlihat bahwa ngrok sedang dalam kondisi online dengan region server Asia Pacific (ap). URL publik yang diberikan, yaitu "https://ead1-126-165-13-9.ngrok-free.app," digunakan oleh aplikasi MIT App Inventor untuk mengakses API server. URL tersebut secara otomatis melakukan *forwarding* dari "http://localhost:80" ke alamat publik sehingga semua *request* dari aplikasi mobile dapat diteruskan ke server lokal tanpa hambatan konfigurasi jaringan tambahan.

Setelah aplikasi MIT App Inventor mengirimkan input suara yang dikonversi menjadi teks, kalimat tersebut diproses melalui API menggunakan library Sastrawi untuk stemming kata dasar serta algoritma *Colussi* untuk pencocokan string. Hasil pengolahan kemudian menentukan aksi yang sesuai, misalnya perintah ON untuk menyalakan kipas atau OFF untuk mematikannya. Aksi ini kemudian dikirimkan ke broker MQTT HiveMQ, yang berfungsi sebagai

penghubung antara server dan perangkat ESP32. ESP32 yang berperan sebagai *subscriber* akan menerima pesan tersebut secara real-time dan mengaktifkan atau menonaktifkan kipas sesuai instruksi.

4.1.3. Kontrol Kipas Angin dengan *Speech Recognition*

Pada tahap ini, sistem kontrol kipas angin mengimplementasikan teknologi *speech recognition* untuk memungkinkan pengguna memberikan perintah secara lisan. Fitur ini dikembangkan melalui aplikasi MIT App Inventor yang dipasang pada perangkat *smartphone*. Dengan menekan tombol Rekam pada aplikasi, pengguna dapat mengucapkan perintah seperti “nyalakan kipas” atau “matikan kipas”. Suara yang ditangkap kemudian diubah menjadi teks oleh modul *speech recognition* bawaan MIT App Inventor. Hasil teks ini dikirimkan ke server API melalui protokol HTTP POST dengan bantuan ngrok yang menyediakan jalur komunikasi publik.



Gambar 4. 3 Logika Speech Recognizer di MIT APP

Gambar di atas merupakan desain blok program pada MIT App Inventor yang digunakan untuk mengimplementasikan fitur *speech recognition* sebagai media input perintah dalam sistem kontrol kipas angin. Proses dimulai ketika pengguna menekan tombol Rekam (Button1.Click). Pada saat tombol ditekan, aplikasi memanggil komponen “SpeechRecognizer1.GetText” untuk menangkap suara pengguna dan mengubahnya menjadi teks.

Gambar 4. 4 Button mic diklik

Setelah sistem berhasil mengenali suara maka blok “SpeechRecognizer1. AfterGettingText” akan aktif dan menyimpan hasil pengenalan suara ke variabel “result”. Teks hasil pengenalan ini kemudian ditampilkan pada label tampilperintah agar pengguna dapat melihat perintah yang sudah diucapkan. Selanjutnya, aplikasi membentuk sebuah request POST ke server menggunakan komponen “Web1.PostText.” Pada blok ini, teks perintah dikirim ke alamat server yang telah disediakan oleh ngrok.

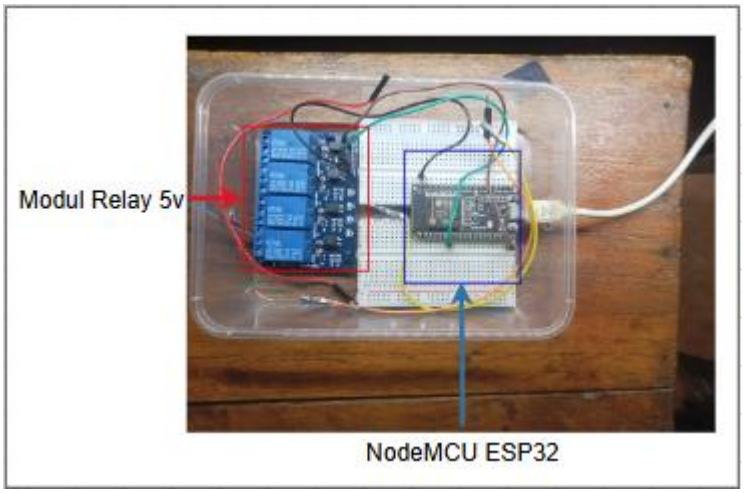
Data yang dikirim memiliki format berupa pasangan kunci-nilai dengan parameter *command_text*, yang berisi kalimat hasil pengenalan suara. Sebelum

dikirim, teks hasil pengenalan suara di-encode menggunakan fungsi "Web1.UriEncode" agar dapat terbaca dengan baik oleh server. Saat proses pengiriman sedang berlangsung, label status akan diperbarui dengan pesan "Mengirim perintah…" untuk memberikan informasi kepada pengguna bahwa sistem sedang memproses instruksi.

Setelah server menerima dan memproses perintah, balasan dari server akan diterima oleh blok "Web1.GotText". Pada tahap ini, respon dari server yang berisi status perintah atau hasil pemrosesan algoritma *Colussi* akan ditampilkan kembali pada label status. Dengan demikian, pengguna dapat mengetahui apakah instruksi yang diberikan telah diterima dan dijalankan dengan benar oleh sistem.

4.1.4. Rangkaian Hardware

Pada sistem kontrol kipas angin berbasis kalimat ini, rancangan hardware memiliki peran penting sebagai eksekutor dari perintah yang diberikan melalui aplikasi berbasis *speech recognition*. Rangkaian hardware dirancang untuk dapat menerima sinyal perintah dari server melalui komunikasi IoT, kemudian menerjemahkannya menjadi aksi nyata berupa pengendalian kipas angin.



Gambar 4. 5 Komponen Hardware

Komponen utama yang digunakan dalam rangkaian ini adalah mikrokontroler ESP32. Perangkat ini dipilih karena memiliki dukungan Wi-Fi bawaan yang memudahkan integrasi dengan broker MQTT, sehingga dapat menerima perintah dari server yang sudah dihubungkan melalui platform ngrok. ESP32 berfungsi sebagai penghubung antara sistem aplikasi dengan perangkat keras yang mengendalikan kipas angin.

Untuk mengatur arus dan tegangan pada kipas, digunakan modul *relay*. *Relay* ini bertindak sebagai saklar elektronik yang dikendalikan langsung oleh ESP32. Ketika ESP32 menerima pesan perintah, misalnya “nyalakan kipas”, maka pin output tertentu pada ESP32 akan memberikan sinyal logika HIGH yang mengaktifkan *relay*. Dengan demikian, *relay* akan menutup rangkaian listrik yang menghubungkan kipas ke sumber daya, sehingga kipas menyala. Sebaliknya, untuk perintah “matikan kipas”, ESP32 akan mengirimkan sinyal logika LOW yang memutuskan arus listrik pada *relay*, sehingga kipas berhenti berputar.

4.2 Hasil Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa sistem kontrol kipas angin berbasis kalimat menggunakan algoritma *Colussi* dapat berjalan sesuai dengan perancangan. Tahapan pengujian mencakup pengujian fungsi *speech recognition*, pengolahan perintah menggunakan algoritma *Colussi*, serta respons perangkat keras dalam mengeksekusi perintah melalui komunikasi IoT.

4.2.1 Proses Pencocokan Pola dengan Algoritma *Colussi*

Pada tahap ini, sistem menguji bagaimana algoritma *Colussi* digunakan untuk mengenali pola perintah pada kalimat hasil *speech recognition* dan *stemming*. Sebagaimana dijelaskan pada Bab III, kalimat perintah yang diucapkan pengguna terlebih dahulu dikonversi menjadi teks, kemudian diproses dengan *stemming* menggunakan library Sastrawi sehingga kata berimbuhan diubah menjadi kata dasar. Sebagai contoh, kalimat "tolong nyalakan kipasnya" diubah menjadi teks stem "tolong nyala kipas", "tolong matikan kipasnya" diubah menjadi teks stem "tolong mati kipas". Dari teks hasil stemming tersebut, sistem mencocokkan bagian kalimat setelah kata "tolong" dengan pola kata kunci, yaitu: nyala kipas dan hidup kipas dipetakan sebagai aksi ON, mati kipas dan padam kipas dipetakan sebagai aksi OFF.

Pencocokan antara teks dan pola dilakukan dengan algoritma *Colussi*. Algoritma ini tidak membandingkan karakter pola secara berurutan dari indeks 0 sampai $m-1$, melainkan mengikuti urutan indeks yang sudah dioptimalkan dalam tabel $h[k]$, serta memanfaatkan tabel $shift[k]$ dan $next[k]$ untuk mengatur pergeseran dan titik lanjutan pemeriksaan ketika terjadi ketidakcocokan pola.

A. Tahap pra-proses pola

Pada penelitian ini digunakan dua pola utama, yaitu: Pola ON : P = "NYALAKIPAS" Pola OFF : P = "MATIKIPAS". Masing-masing pola dipra-proses untuk membentuk tiga tabel utama:

1. Tabel $h[k]$ Menentukan urutan indeks karakter pola yang akan diperiksa. Pada saat pencocokan, algoritma akan membandingkan karakter sesuai dengan rumus (3.1) P[$h[k]$] == T[i+$h[k]$] dengan i adalah posisi awal jendela pola pada teks.
2. Tabel $shift[k]$ Menentukan seberapa jauh pola digeser ketika terjadi ketidakcocokan (mismatch) pada langkah ke-k. Pergeseran posisi jendela dapat dituliskan sebagaimana rumus (3.2) $i = i + shift\ [k]$.
3. Tabel $next[k]$ Menentukan dari urutan ke berapa dalam tabel h pemeriksaan akan dilanjutkan setelah pola digeser sebagaimana rumus (3.3) $k = next[k]$.

Contoh hasil pra-proses untuk pola "NYALAKIPAS" ditunjukkan pada Tabel 4.1, sedangkan contoh untuk pola "MATIKIPAS" ditunjukkan pada Tabel 4.2.

Tabel 4.1Tabel h[k], shift[k], dan next[k] untuk pola “NYALAKIPAS”

k	**h[k]**	**Karakter (P[h[k]])**	**shift[k]**	**next[k]**
0	1	Y	1	0
1	2	A	2	0
2	3	L	3	0
3	4	A	3	0
4	5	K	4	0
5	6	I	4	0

k	h[k]	Karakter (P[h[k]])	shift[k]	next[k]
6	7	P	5	0
7	8	A	6	0
8	9	S	7	0
9	0	N	1	0

Pada Tabel 4.1, Kolom $h[k]$ menunjukkan indeks karakter pola yang diperiksa pada langkah ke- k. Kolom $shift\ [k]$ menunjukkan seberapa jauh pola digeser ketika mismatch terjadi pada langkah ke- k . Kolom $next\ [k]$ menunjukkan indeks urutan baru setelah pergeseran. Pada contoh ini nilai $next[k]$ dibuat 0 (ilustratif), yang berarti setelah digeser, pemeriksaan kembali dimulai dari urutan pertama.

Tabel 4.2 Tabel h[k], shift[k], dan next[k] untuk pola "MATI KIPAS"

k	h[k]	Karakter (P[h[k]])	shift[k]	next[k]
0	1	A	1	0
1	2	T	2	0
2	3	I	2	0
3	4	K	3	0
4	5	I	3	0
5	6	P	4	0
6	7	A	5	0

k	h[k]	Karakter (P[h[k]])	shift[k]	next[k]
7	8	S	6	0
8	0	M	1	0

Tabel 4.2 memiliki makna yang sama: urutan pemeriksaan ditentukan oleh $h[k]$, sedangkan $shift[k]$ dan $next\ [k]$ mengatur pergeseran dan langkah lanjutan ketika terjadi ketidakcocokan.

B. Tahap pencocokan pola

Setelah pra-proses selesai, algoritma Colussi digunakan untuk mencocokkan pola dengan teks hasil speech recognition dan stemming. Secara garis besar, langkah pencocokan yang berhubungan langsung dengan kendali kipas adalah sebagai berikut:

1. Inisialisasi jendela dan urutan.

Posisi awal jendela pola pada teks diset i = 0. Indeks urutan pemeriksaan pola diset k = 0. Selama $i \leq n - m$ (dengan n n panjang teks dan m m panjang pola), algoritma akan mencoba mencocokkan pola pada posisi tersebut.

2. Membandingkan karakter pola dan teks.

Pada setiap langkah, algoritma mengambil indeks j = $h[k]$, kemudian membandingkan: $P[h[k]]$ == $T[i+h[k]]$ Jika sama, maka k dinaikkan menjadi k+1. Jika k sudah mencapai m, artinya seluruh karakter pola cocok dan pola dinyatakan ditemukan pada posisi i.

3. Menangani mismatch.

Jika terjadi mismatch pada perbandingan $P[h[k]]$ dan T $[i+h$ $[k]]$, algoritma Colussi tidak kembali ke awal pola, tetapi: Menggeser jendela pola dengan i = $i + shift[k]$ Mengubah indeks urutan menjadi k baru = $next[k]$ Dengan cara ini, algoritma melompati bagian teks yang dipastikan tidak mungkin mengandung kecocokan, sehingga mempercepat proses.

4. Kendali kipas berdasarkan hasil pencocokan

Jika pola "nyala kipas" atau "hidup kipas"' ditemukan pada teks, sistem mengirimkan perintah "ON" ke broker MQTT, kemudian diteruskan ke ESP32 untuk menyalakan kipas. Jika pola "mati kipas" atau "padam kipas" ditemukan, sistem mengirimkan perintah "OFF" untuk mematikan kipas.

Selain pencocokan pola perintah ON dan OFF, sistem juga memperhatikan keberadaan kata negasi dalam kalimat, seperti "tidak", "jangan", "bukan", "enggak", atau "nggak". Pendeteksian negasi ini dilakukan sebelum hasil akhir aksi dikirim ke modul kendali kipas. Secara logika, algoritma *Colussi* tetap digunakan untuk mencari pola utama seperti "nyala kipas" atau "mati kipas", namun jika di dalam kalimat yang sama ditemukan kata negasi, maka keputusan aksi dapat dibalik atau dibatalkan. Sebagai contoh, apabila kalimat yang diucapkan adalah "jangan nyalakan kipas", maka secara pola kalimat tersebut mengandung frasa nyala kipas (yang seharusnya memicu aksi ON), tetapi karena terdapat kata "jangan", sistem menafsirkan perintah tersebut sebagai tidak menyalakan kipas (aksi ON dibatalkan atau diterjemahkan sebagai OFF sesuai kebutuhan rancangan). Dengan cara ini, algoritma Colussi tetap berfungsi sebagai mesin pencocokan pola, sementara

penanganan negasi dilakukan sebagai lapisan logika tambahan di atas hasil pencocokan pola.

4.2.2 Pengujian Performa Sistem

Pengujian performa sistem dilakukan untuk mengetahui kinerja sistem kontrol kipas angin berbasis kalimat menggunakan algoritma *Colussi*. Uji coba dilakukan dengan memberikan beberapa variasi kalimat perintah yang memiliki arti sama, namun dengan struktur kata yang berbeda. Tujuan pengujian ini adalah untuk memastikan sistem dapat mengenali pola kata kunci pada kalimat perintah dan menjalankan aksi yang sesuai. Berikut adalah hasil dari pengujian:

Tabel 4.3 Pengujian Perintah "Nyalakan / Matikan Kipas"

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
1	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	972
2	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	2979
3	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	2222
4	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	1987
5	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	1988
6	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	3484
7	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	1909
8	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	684
9	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	611
10	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	585
11	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	2268
12	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	796
13	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	617
14	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	2674
15	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	623

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
16	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	585
17	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	664
18	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	1899
19	tolong nyalakan kipasnya	tolong nyala kipas	nyala kipas	ON	1167
20	tolong matikan kipasnya	tolong mati kipas	mati kipas	OFF	629
Rata-rata durasi eksekusi perintah					**1467,15 ms.**

Berdasarkan hasil pengujian pada Tabel 4.3, seluruh perintah "nyalakan kipas" dan "matikan kipas" berhasil dikenali dengan benar oleh sistem, di mana perintah "nyalakan" selalu dieksekusi menjadi aksi ON dan perintah "matikan" selalu dieksekusi menjadi OFF. Hal ini menunjukkan bahwa proses *stemming* menggunakan Sastrawi dan pencocokan pola menggunakan algoritma *Colussi* mampu bekerja dengan baik pada kalimat perintah sederhana tanpa negasi. Perbedaan waktu eksekusi yang terlihat pada tabel disebabkan oleh beberapa faktor seperti perbedaan kestabilan jaringan internet, waktu respon *broker* MQTT dan ESP32, serta beban proses pada server saat melakukan *stemming*, pencocokan pola, dan pengiriman pesan. Meskipun terdapat variasi waktu antara pengujian satu dengan lainnya, rata-rata durasi sebesar 1467,15 ms masih menunjukkan bahwa sistem dapat merespons perintah dengan cukup cepat.

Tabel 4.4 Pengujian menggunakan sinonim lain

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
1	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	674
2	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	673
3	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	2626
4	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	6812

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
5	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	5073
6	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	838
7	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	2615
8	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	9034
9	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	4378
10	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	641
11	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	705
12	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	647
13	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	4769
14	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	3314
15	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	4309
16	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	3179
17	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	3564
18	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	3859
19	tolong hidupkan kipasnya	tolong hidup kipas	hidup kipas	ON	1752
20	tolong padamkan kipasnya	tolong padam kipas	padam kipas	OFF	4012
Rata-rata durasi eksekusi perintah					**3173,7 ms**.

Berdasarkan hasil pengujian pada Tabel 4.4, sistem diuji menggunakan kalimat perintah dengan sinonim lain seperti “hidupkan” untuk menyalakan kipas dan “padamkan” untuk mematikan kipas. Hasil menunjukkan bahwa seluruh perintah berhasil dikenali dengan benar, di mana perintah “hidupkan” dieksekusi sebagai aksi ON dan “padamkan” dieksekusi sebagai OFF. Hal ini membuktikan bahwa proses stemming serta pencocokan pola menggunakan algoritma *Colussi* mampu mengenali variasi kata dengan makna yang sama. Namun, waktu eksekusi pada tabel menunjukkan variasi yang cukup besar, dengan durasi tercepat sekitar

641 ms dan terlama mencapai 9034 ms, serta rata-rata 3173,7 ms. Perbedaan ini dipengaruhi oleh kestabilan jaringan internet, waktu respon broker MQTT HiveMQ, proses pemrosesan kalimat pada server, serta beban kerja sistem saat pengujian. Meskipun demikian, sistem tetap mampu menjalankan perintah dengan baik sehingga dapat dikategorikan berjalan efektif meskipun dengan waktu respon yang bervariasi.

Tabel 4.5 Pengujian Perintah dengan Kata Larangan "Jangan"

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
1	tolong jangan matikan kipasnya	tolong jangan mati kipas	mati kipas	OFF	3461
2	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	759
3	tolong jangan matikan kipasnya	tolong jangan mati kipas	mati kipas	OFF	749
4	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	681
5	tolong jangan matikan kipasnya	tolong jangan mati kipas	mati kipas	OFF	686
6	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	3172
7	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	1134
8	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	1873
9	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	1899
10	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	1254
11	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	2566
12	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	578
13	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	649
14	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	697
15	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	4115

No	Kalimat Perintah	Stem	Pattern	Aksi	Durasi (ms)
16	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	3871
17	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	3877
18	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	3654
19	Tolong jangan matikan kipasnya	Tolong jangan mati kipas	mati kipas	OFF	1904
20	tolong jangan nyalakan kipasnya	tolong jangan nyala kipas	nyala kipas	ON	688
Rata-rata durasi eksekusi perintah					**1913,35 ms**.

Berdasarkan hasil pengujian pada Tabel 4.5, sistem diuji menggunakan kalimat perintah yang mengandung kata larangan “jangan”, baik pada perintah menyalakan maupun mematikan kipas angin. Dari hasil pengujian terlihat bahwa sistem tetap mengeksekusi aksi sesuai pola utama yang terdeteksi, yaitu “mati kipas” dieksekusi sebagai OFF dan “nyala kipas” dieksekusi sebagai ON, meskipun pada kalimat terdapat kata larangan. Hal ini menunjukkan bahwa sistem belum sepenuhnya memahami makna negasi secara semantik, karena pencocokan kalimat hanya berfokus pada pola utama kata kerja tanpa mempertimbangkan makna larangan yang seharusnya membalikkan perintah.

Dari sisi performa, waktu eksekusi menunjukkan variasi, dengan waktu tercepat sekitar 578 ms dan terlama mencapai 4115 ms, serta rata-rata 1913,35 ms. Variasi waktu tersebut dipengaruhi oleh faktor kondisi jaringan internet, respon broker MQTT HiveMQ, serta proses pemrosesan kalimat pada server. Namun jika dibandingkan dengan tabel sebelumnya, rata-rata waktu pada pengujian ini relatif lebih cepat karena struktur kalimat lebih sederhana dan proses pencocokan pola lebih mudah dikenali oleh sistem. Secara keseluruhan, pengujian ini menunjukkan bahwa meskipun sistem mampu mengenali perintah dengan kata larangan, tetapi

masih memiliki keterbatasan dalam memahami konteks negasi sehingga dapat menyebabkan interpretasi perintah yang tidak sesuai dengan maksud sebenarnya.Dari hasil pengujian, terlihat bahwa sistem mampu mengenali variasi kalimat perintah yang berbeda dengan pola kata kunci yang sama. Perintah dengan kata larangan “jangan” masih terdeteksi sesuai pola, meskipun makna semantik seharusnya membatalkan perintah. Hal ini menunjukkan bahwa sistem berbasis algoritma Colussi sangat bergantung pada pencocokan pola kata kunci, bukan pemahaman konteks penuh kalimat.

Secara keseluruhan, perintah “hidupkan/padamkan kipas” membutuhkan waktu eksekusi rata-rata lebih lama (3173,7 ms) dibandingkan dengan “nyalakan/matikan kipas” (1467,15 ms) maupun perintah dengan kata larangan (1913,35 ms). Hal ini menunjukkan bahwa pemilihan pola dan panjang kalimat mempengaruhi performa sistem.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa sistem mampu mengeksekusi perintah pengendalian kipas dengan variasi kalimat yang berbeda. Namun, sistem belum mampu memahami konteks penuh, sehingga perintah dengan kata larangan seperti “jangan” tetap diproses sebagai perintah utama. Dari segi performa, kalimat yang lebih panjang dan kompleks cenderung membutuhkan waktu pemrosesan yang lebih lama. Dengan demikian, sistem ini efektif untuk pengendalian sederhana berbasis kata kunci, namun masih memerlukan pengembangan lebih lanjut agar dapat mengenali konteks dan makna perintah secara lebih akurat.

4.2.3 Pengujian *Error* Sistem

Pengujian *error* sistem bertujuan untuk mengukur tingkat kesalahan sistem dalam memahami dan mengeksekusi perintah yang diberikan. Kesalahan yang dimaksud adalah ketika aksi yang dijalankan sistem tidak sesuai dengan makna sebenarnya dari kalimat perintah. Pada penelitian ini, pengujian dilakukan menggunakan suara penutur dewasa berbahasa Indonesia dengan artikulasi yang jelas, kecepatan bicara normal, dan intonasi wajar, diucapkan pada jarak kurang lebih 20–30 cm dari mikrofon smartphone di ruangan yang relatif tenang (minim kebisingan), sehingga variasi kesalahan yang muncul lebih merefleksikan kinerja sistem daripada gangguan kualitas suara.

Tabel 4.6 Pengujian Error Sistem

No	Kalimat Perintah	Aksi Sistem	Aksi Seharusnya	Error (Ei)
1	tolong nyalakan kipasnya	ON	ON	0
2	tolong matikan kipasnya	OFF	OFF	0
3	tolong nyalakan kipasnya	ON	ON	0
4	tolong matikan kipasnya	OFF	OFF	0
5	tolong nyalakan kipasnya	ON	ON	0
6	tolong matikan kipasnya	OFF	OFF	0
7	tolong nyalakan kipasnya	ON	ON	0
8	tolong matikan kipasnya	OFF	OFF	0
9	tolong nyalakan kipasnya	ON	ON	0
10	tolong matikan kipasnya	OFF	OFF	0
11	tolong nyalakan kipasnya	ON	ON	0
12	tolong matikan kipasnya	OFF	OFF	0
13	tolong nyalakan kipasnya	ON	ON	0
14	tolong matikan kipasnya	OFF	OFF	0
15	tolong nyalakan kipasnya	ON	ON	0
16	tolong matikan kipasnya	OFF	OFF	0
17	tolong nyalakan kipasnya	ON	ON	0

No	Kalimat Perintah	Aksi Sistem	Aksi Seharusnya	Error (Ei)
18	tolong matikan kipasnya	OFF	OFF	0
19	tolong nyalakan kipasnya	ON	ON	0
20	tolong matikan kipasnya	OFF	OFF	0
21	tolong hidupkan kipasnya	ON	ON	0
22	tolong padamkan kipasnya	OFF	OFF	0
23	tolong hidupkan kipasnya	ON	ON	0
24	tolong padamkan kipasnya	OFF	OFF	0
25	tolong hidupkan kipasnya	ON	ON	0
26	tolong padamkan kipasnya	OFF	OFF	0
27	tolong hidupkan kipasnya	ON	ON	0
28	tolong padamkan kipasnya	OFF	OFF	0
29	tolong hidupkan kipasnya	ON	ON	0
30	tolong padamkan kipasnya	OFF	OFF	0
31	tolong hidupkan kipasnya	ON	ON	0
32	tolong padamkan kipasnya	OFF	OFF	0
33	tolong hidupkan kipasnya	ON	ON	0
34	tolong padamkan kipasnya	OFF	OFF	0
35	tolong hidupkan kipasnya	ON	ON	0
36	tolong padamkan kipasnya	OFF	OFF	0
37	tolong hidupkan kipasnya	ON	ON	0
38	tolong padamkan kipasnya	OFF	OFF	0
39	tolong hidupkan kipasnya	ON	ON	0
40	tolong padamkan kipasnya	OFF	OFF	0
41	tolong jangan matikan kipasnya	OFF	ON	1
42	tolong jangan nyalakan kipasnya	ON	OFF	1
43	tolong jangan matikan kipasnya	OFF	ON	1
44	tolong jangan nyalakan kipasnya	ON	OFF	1
45	tolong jangan matikan kipasnya	OFF	ON	1

No	Kalimat Perintah	Aksi Sistem	Aksi Seharusnya	Error (Ei)
46	tolong jangan nyalakan kipasnya	ON	OFF	1
47	tolong jangan matikan kipasnya	OFF	ON	1
48	tolong jangan nyalakan kipasnya	ON	OFF	1
49	tolong jangan matikan kipasnya	OFF	ON	1
50	tolong jangan nyalakan kipasnya	ON	OFF	1
51	tolong jangan matikan kipasnya	OFF	ON	1
52	tolong jangan nyalakan kipasnya	ON	OFF	1
53	tolong jangan matikan kipasnya	OFF	ON	1
54	tolong jangan nyalakan kipasnya	ON	OFF	1
55	tolong jangan matikan kipasnya	OFF	ON	1
56	tolong jangan nyalakan kipasnya	ON	OFF	1
57	tolong jangan matikan kipasnya	OFF	ON	1
58	tolong jangan nyalakan kipasnya	ON	OFF	1
59	tolong jangan matikan kipasnya	OFF	ON	1
60	tolong jangan nyalakan kipasnya	ON	OFF	1

Pengujian *error* sistem bertujuan untuk mengukur tingkat kesalahan sistem dalam memahami dan mengeksekusi perintah yang diberikan. Kesalahan yang dimaksud adalah ketika aksi yang dijalankan sistem tidak sesuai dengan makna sebenarnya dari kalimat perintah. Pada pengujian ini, setiap kalimat perintah dibandingkan antara aksi hasil sistem dan aksi yang seharusnya dijalankan berdasarkan arti kalimat. Jika hasilnya berbeda, maka percobaan tersebut dihitung sebagai *error*. Berdasarkan data hasil pengujian sebelumnya, total percobaan yang

dilakukan sebanyak n = 60 kali, dengan jumlah kesalahan ΣEi = 20 kali. Perhitungan rata-rata *error* menggunakan Persamaan (3.4) adalah sebagai berikut:

$$\mu = \frac{20}{60} * 100\% = 33{,}33\%$$

Dari perhitungan tersebut, diperoleh bahwa rata-rata *error* sistem adalah 33,33%. Nilai *error* ini sebagian besar disebabkan oleh ketidakmampuan sistem untuk memahami konteks kata larangan seperti "jangan" yang seharusnya membatalkan aksi, namun tetap dikenali sebagai perintah utama berdasarkan pencocokan pola kata kunci.

Hasil ini menunjukkan bahwa meskipun algoritma *Colussi* mampu bekerja dengan baik dalam pencocokan pola, sistem masih memerlukan pengolahan bahasa alami yang lebih mendalam agar dapat memahami makna kalimat secara kontekstual, sehingga dapat mengurangi tingkat *error* pada kasus perintah dengan bentuk negasi.

4.3 Integrasi Islam

Dalam pengembangan sistem kontrol kipas angin berbasis kalimat menggunakan algoritma *Colussi* ini, aspek integrasi Islam menjadi hal penting yang perlu diperhatikan agar teknologi yang dihasilkan tidak hanya memberikan manfaat praktis, tetapi juga memiliki nilai moral dan spiritual. Integrasi Islam di sini tidak terbatas pada aspek teori, tetapi juga menyangkut penerapan nilai-nilai keislaman dalam proses perancangan, implementasi, hingga pemanfaatan sistem.

4.3.1 Muamalah *Ma'a Allah*

Muamalah *Ma'a Allah* adalah interaksi seorang hamba dengan Allah yang mencakup segala bentuk ibadah. Salah satu wujud utama dari muamalah ini adalah sikap syukur, yaitu kesadaran bahwa seluruh nikmat baik yang bersifat material maupun non-material berasal dari Allah dan harus digunakan sesuai kehendak-Nya. Nikmat tersebut mencakup akal, ilmu pengetahuan, kemampuan berpikir logis, serta teknologi yang dikembangkan manusia. Dalam Al-Qur'an, Allah berfirman:

وَإِذْ تَأَذَّنَ رَبُّكُمْ لَئِن شَكَرْتُمْ لَأَزِيدَنَّكُمْ ۖ وَلَئِن كَفَرْتُمْ إِنَّ عَذَابِي لَشَدِيدٌ

"Dan (ingatlah juga), tatkala Tuhan-mu memaklumkan, "Sesungguhnya jika kamu bersyukur pasti Kami akan menambah (nikmat) kepadamu dan jika kamu mengingkari (nikmat-Ku), maka sesungguhnya azab-Ku sangat pedih" (QS. Ibrahim: 7)

Dalam Tafsir Ibn Katsir pada kitab *Tafsir al-Qur'an al-'Azhim* dijelaskan bahwa syukur tidak hanya diucapkan dengan lisan, tetapi diwujudkan melalui pemanfaatan nikmat Allah dalam ketaatan. Ibn Katsir menegaskan bahwa menggunakan nikmat Allah untuk kebaikan adalah bentuk syukur yang paling nyata, sedangkan menggunakan nikmat untuk kemaksiatan atau kesia-siaan merupakan bentuk kufur nikmat(Tafsir Al-Qur'an al-'Azhim Jilid 4.4,).

Dalam aplikasi sistem kontrol kipas ini, aspek muamalah ma'a Allah dapat dihadirkan dengan sikap syukur yang diwujudkan melalui perancangan sistem yang bertujuan memberi manfaat, menghindari pemborosan, serta menyadari bahwa keberhasilan sistem bukan semata hasil kemampuan manusia, melainkan atas izin Allah. Dengan demikian, pengembangan sistem ini merupakan bentuk syukur

amali, yaitu penggunaan ilmu dan teknologi sebagai ibadah dan amanah yang dipertanggungjawabkan di hadapan Allah SWT.

4.3.2 Muamalah *Ma'a An-Nas*

Muamalah *Ma'a An-Nas* mencakup segala hubungan dan interaksi antar manusia dalam bentuk kehidupan sosial. Dalam ranah teknologi, ini berarti bahwa inovasi harus diarahkan agar memberi manfaat nyata, tidak merugikan orang lain. Sistem kontrol kipas dengan perintah suara sangat membantu kelompok rentan seperti lansia atau penyandang disabilitas, memungkinkan mereka memanfaatkan teknologi tanpa hambatan. Sistem ini memudahkan manusia dalam melakukan aktivitas sehari-hari. Hal ini selaras dengan hadis:

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia."(HR. Ahmad)

Hadis ini menekankan bahwa setiap karya teknologi harus diarahkan untuk kemaslahatan, bukan sekadar kemewahan atau kesia-siaan. Dalam bukunya *Fiqh al-Qiyam wa al-Akhlaq fi al-Islam*, al-Qardhawi menjelaskan bahwa kemanfaatan adalah ukuran utama keutamaan manusia, karena Islam datang untuk mewujudkan *maslahah* (kebaikan) dan mencegah *mafsadah* (kerusakan). Hadis ini, menurutnya, menunjukkan bahwa manusia terbaik bukanlah yang paling banyak ibadah individualnya semata, tetapi yang keberadaannya membawa kebaikan, kemudahan, dan solusi bagi orang lain(Fiqh Al-Qiyam Wa al-Akhlaq Fi al-Islam,.). Dengan demikian, muamalah ma'a an-nas dalam penelitian ini adalah memastikan teknologi dapat membantu, memudahkan penggunanya.

4.3.3 Muamalah *Ma'a Alam*

Muamalah *Ma'a Alam* mencakup hubungan manusia dengan alam semesta. Islam mengajarkan manusia untuk menjaga keseimbangan alam dan tidak melakukan kerusakan. Sistem kontrol kipas angin yang dikembangkan juga berhubungan dengan aspek efisiensi energi. Dengan adanya kendali berbasis kalimat, penggunaan listrik dapat diatur sesuai kebutuhan, sehingga terhindar dari pemborosan yang termasuk perbuatan *israf*. Allah berfirman dalam Al-Qur'an:

وَلَا تُفْسِدُوا۟ فِى ٱلْأَرْضِ بَعْدَ إِصْلَٰحِهَا وَٱدْعُوهُ خَوْفًا وَطَمَعًا ۚ إِنَّ رَحْمَتَ ٱللَّهِ قَرِيبٌ مِّنَ ٱلْمُحْسِنِينَ

"Dan janganlah kamu membuat kerusakan di muka bumi, sesudah (Allah) memperbaikinya dan berdoalah kepada-Nya dengan rasa takut (tidak akan diterima) dan harapan (akan dikabulkan). Sesungguhnya rahmat Allah amat dekat kepada orang-orang yang berbuat baik"(QS. Al- A'raf: 56)

Ayat ini mengingatkan manusia untuk mengelola sumber daya dengan bijak. Dalam prinsip Islam, konsep *israf* (pemborosan) sangat dilarang, sehingga setiap penggunaan sumber daya harus disertai kehati-hatian. Dalam *Tafsir al-Qur'an al-'Azim*, Ibnu Katsir menjelaskan bahwa ayat ini melarang manusia melakukan kerusakan (*fasad*) di muka bumi setelah Allah menjadikannya baik dan teratur. Kerusakan termasuk perilaku maksiat, kezaliman, dan perbuatan yang mengacaukan keseimbangan sosial dan alam(Tafsir Al-Qur'an al-'Azhim Jilid 3.4,). Allah memerintahkan hamba-Nya untuk menjauhi kerusakan, berdoa dengan takut dan penuh harap kepada-Nya, karena rahmat Allah amat dekat bagi orang-orang yang berbuat baik (*al-muhsinin*). Pada sistem kendali kipas, nilai ini hadir sebagai dorongan efisiensi energi, desain yang hemat daya di sisi perangkat, dan operasi yang menghindari *israf* (pemborosan).

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan. Penelitian ini berhasil merancang dan mengimplementasikan sebuah sistem kontrol perangkat elektronik yang dapat merespons perintah suara dalam bentuk kalimat bahasa Indonesia. Sistem ini mengintegrasikan proses pengenalan suara, stemming menggunakan library Sastrawi, pencocokan pola dengan algoritma *Colussi*, serta pengendalian perangkat melalui modul ESP32 yang terhubung dengan protokol MQTT. Hasil implementasi menunjukkan bahwa algoritma *Colussi* dapat digunakan untuk melakukan pencocokan pola secara efisien. Pada pola kalimat sederhana seperti "nyalakan kipas" dan "matikan kipas", algoritma mampu menemukan kecocokan dengan cepat melalui proses pembentukan tabel *h[k]*, *shift[k]*, dan *next[k]*. Waktu eksekusi rata-rata berkisar antara 578 – 9034 milidetik tergantung panjang dan kompleksitas kalimat yang diucapkan. Pengujian sistem menunjukkan bahwa tingkat keberhasilan perintah mencapai 66,7%, sementara tingkat kesalahan sebesar 33,3%. Kesalahan terutama terjadi ketika perintah mengandung kata negasi seperti "jangan", "tidak", atau "nggak", serta pada kalimat yang menggunakan sinonim berbeda dari pola utama. Hal ini menandakan bahwa sistem sudah mampu mengenali perintah dasar dengan baik, namun masih terbatas dalam mengantisipasi variasi kalimat yang lebih kompleks.

5.2. Saran

Berdasarkan hasil penelitian yang diperoleh, terdapat beberapa masih terdapat banyak kekurangan yang perlu dikembangkan lagi pada penelitian berikutnya:

1. Penambahan dataset kalimat perintah agar sistem mampu mengenali lebih banyak variasi bahasa, khususnya pada kalimat dengan negasi dan sinonim yang sering digunakan sehari-hari.
2. Pengembangan pada kontrol kecepatan kipas angin, supaya kipas angin dapat bekerja sesuai dengan kecepatan yang diinginkan pengguna.

Dengan mempertimbangkan saran-saran tersebut, diharapkan penelitian ini dapat menjadi landasan untuk pengembangan sistem kontrol berbasis suara yang lebih cerdas, inklusif, dan bermanfaat bagi masyarakat luas.

DAFTAR PUSTAKA

Aghenta, L. O., & Iqbal, M. T. (2019). Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol. *AIMS Electronics and Electrical Engineering*, *4*(1), 57–86. https://doi.org/10.3934/ElectrEng.2020.1.57

Ahmar, A. S. (2023). Forecast Error Calculation with Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). *JINAV: Journal of Information and Visualization*, *1*(2), 94–96. https://doi.org/10.35877/454RI.jinav303

Fiqh al-Qiyam wa al-Akhlāq fi al-Islām. (n.d.).

Anggoro, W. W. (2021). The Perancangan dan Penerapan Kendali Lampu Ruangan Berbasis IoT (Internet of Things) Android. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *8*(3), 1596–1606. https://doi.org/10.35957/jatisi.v8i3.1311

Arrosyidah, Q., Oktaviyati, F., Faridawati, D., Cahyani, V. D., Afni, L. N., Harijanto, A., & Abstrak, U. J. (n.d.). Pengembangan Sistem Kipas Otomatis Berbasis Internet Of Things (Iot) Menggunakan Sensor PIR Dan DHT11. *Jurnal Ilmiah Wahana Pendidikan*, *2023*(16), 647–654. https://doi.org/10.5281/zenodo.8263192

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Balta-Ozkan, N., Boteler, B., & Amerighi, O. (2014). European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy. *Energy Research and Social Science*, *3*(C), 65–77. https://doi.org/10.1016/j.erss.2014.07.007

Bardadi, A. (2019). KNOWLEDGE SHARING MENGGUNAKAN ALGORITMA COLLUSI. *JSI: Jurnal Sistem Informasi (E-Journal)*, *11*(1). https://doi.org/10.36706/jsi.v11i1.9054

Charras, C., & Lecroq, T. (n.d.). *Handbook of Exact String-Matching Algorithms*.

Filipe, L., Peres, R. S., & Tavares, R. M. (2021). Voice-Activated Smart Home Controller Using Machine Learning. *IEEE Access*, *9*, 66852–66863. https://doi.org/10.1109/ACCESS.2021.3076750

Hanani, A., & Hariyadi, M. A. (2020). Smart Home Berbasis IoT Menggunakan Suara Pada Google Assistant. *Jurnal Ilmiah Teknologi Informasi Asia, 14 (1). Pp. 49-56. ISSN 25808397*. https://repository.uin-malang.ac.id/12113/1/12113.pdf

Isyanto, H., Arifin, A. S., & Suryanegara, M. (2020). *Performance of Smart Personal Assistant Applications Based on Speech Recognition Technology using IoT-based Voice Commands*. IEEE.

Iyappan, P., Loganathan, J., Verma, M. K., Dumka, A., Singh, R., Gehlot, A., Akram, S. V., Kaur, S., & Joshi, K. (2022). A generic and smart automation system for home using internet of things. *Bulletin of Electrical Engineering and Informatics*, *11*(5), 2727–2736. https://doi.org/10.11591/eei.v11i5.3785

Khancome, C., & Boonjing, V. (2012). New hashing-based multiple string pattern matching algorithms. *Proceedings of the 9th International Conference on Information Technology, ITNG 2012*, 195–200. https://doi.org/10.1109/ITNG.2012.34

Mabrouk, N. El, Crochemore, M., & El-Mabrouk, N. (1996). *Boyer-Moore strategy to efficient approximate string matching Boyer-Moore strategy to eecient approximate string matching*. https://hal.science/hal-00620020

Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *IEEE Access*, *8*, 201071–201086. https://doi.org/10.1109/ACCESS.2020.3035849

Novia Intan Pratiwi, N. A. H. S. R. S. (2020). Aplikasi Pencarian Biografi Tokoh Politik Indonesia Berbasis Mobile Dengan Algoritma Colussi. *Novia Intan Pratiwi, Nelly Astuti Hasibuan, Saidi Ramadan Siregar*.

Raihan Azis, M., Fitri, I., Rahman, B., Teknologi Komunikasi dan Informatika, F., Nasional Ps Minggu, U., Jakarta Selatan, K., & Khusus Ibukota Jakarta, D. (n.d.). *PENGGUNAAN ALGORITMA BRUTE FORCE STRING MATCHING DALAM PENCARIAN ORANG HILANG PADA WEBSITE TEMUKANDIA.COM*.

Rifai, W., & Winarko, E. (2019). Modification of Stemming Algorithm Using A Non Deterministic Approach To Indonesian Text. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *13*(4), 379. https://doi.org/10.22146/ijccs.49072

Ritonga, M. R., Fadillah, N., & Fitria, L. (2019). Sistem Kendali Peralatan Elektronik Rumah Tangga Melalui Media Wireless Fidelity Menggunakan Voice Recognition Secara Real Time. *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, *3*(2), 1–7. https://doi.org/10.30743/infotekjar.v3i2.905

Riyadh as-Salihin. (n.d.).

Setyawan, Y., & Ceng Giap, Y. (2022). Implementasi Speech Recognition untuk Asisten Virtual dengan Python. *JURNAL ALGOR*, *1*. https://jurnal.buddhidharma.ac.id/index.php/algor/index

Sudrajat, R., & Rofifah, F. (2023). Rancang Bangun Sistem Kendali Kipas Angin dengan Sensor Suhu dan Sensor Ultrasonik Berbasis Arduino Uno. *Remik*, *7*(1), 555–564. https://doi.org/10.33395/remik.v7i1.12082

Tafsir al-Qur'an al-'Azhim Jilid 3.4. (n.d.).

Tafsir al-Qur'an al-'Azhim Jilid 4.4. (n.d.).

Tjandra Setiati, A., Kurniawati, N., Apriliani, I., Argya Wardani, N., Studi Telekomunikasi, P., Teknik Elektro, J., Negeri Jakarta, P., & Siwabessy, J. G. (2023a). Sistem Kendali Kipas Angin Otomatis Dengan Sistem Monitoring Berbasis IoT. In *Prosiding Seminar Nasional Teknik Elektro* (Vol. 8).

Tjandra Setiati, A., Kurniawati, N., Apriliani, I., Argya Wardani, N., Studi Telekomunikasi, P., Teknik Elektro, J., Negeri Jakarta, P., & Siwabessy, J. G. (2023b). Sistem Kendali Kipas Angin Otomatis Dengan Sistem Monitoring Berbasis IoT. In *Prosiding Seminar Nasional Teknik Elektro* (Vol. 8).

Utomo, D., Indriyanto, S., & Yuliantoro, P. (2024). Sistem Otomatisasi Peralatan Rumah Tangga Bagi Penyandang Disabilitas Menggunakan Google Assistant. *Jurnal SINTA: Sistem Informasi Dan Teknologi Komputasi*, *1*(2). https://doi.org/10.61124/sinta.v1i2.19

Vaidya, V. D., & Vishwakarma, P. (n.d.). *A Comparative Analysis on Smart Home System to Control, Monitor and Secure Home,based on technologies like GSM,IOT,Bluetooth and PIC Microcontroller with ZigBee Modulation*.

Zulkarnain, S. N., Ahmad Shauri, R. L., Saidin, M. H., & Zamanhuri, A. Z. A. (2024). IoT Monitoring of a Master-Slave Robot System using MIT App Inventor. *Journal of Electrical & Electronic Systems Research*, *24*(Apr2024), 33–39. https://doi.org/10.24191/jeesr.v24i1.005