

Pengembangan Sistem Dialog Interaksi Manusia dan Mesin Menggunakan Finite State Machine

1st Muhammad Hasbi Juri Utama

Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

muhammadhasbiju@student.telkomuni-
versity.ac.id

2th Ahmad Musnansyah

Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

ahmadanc@telkomuniversity.ac.id

3rd Hanif Fakhurroja

Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

haniff@telkomuniversity.ac.id

Abstrak— Penerapan sistem dialog di dalam kendaraan tidak hanya untuk memudahkan aktifitas pengguna, namun untuk menjaga pengguna agar tetap fokus selama mengemudi, sehingga dapat menghindari kecelakaan. Sistem dialog pada penelitian ini memanfaatkan pengenalan ucapan sebagai data input dari pengguna untuk mengontrol peralatan kendaraan mobil seperti menyalakan lampu, mematikan AC, menyalakan radio, dll. Penelitian ini menyajikan pengembangan sistem dialog berbasis Internet of Things (IoT) dengan menggunakan algoritma model Finite State Machine (FSM) untuk mengelola sistem dialog. Sistem ini diimplementasikan pada Raspberry Pi yang merupakan komputer berukuran kecil sebagai alat kendali utama, kemudian menggunakan Firebase real-time database sebagai penerima dan pengirim data berdasarkan input sebuah sistem dialog. Data pada Firebase akan tersinkronisasi dengan mikrokontroler ESP32 yang akan menghidupkan atau mematikan perangkat kendaraan. Metode perancangan pada penelitian ini menggunakan model prototyping karena akan sering dilakukan percobaan sebelum mencapai tahap akhir. Hasil dari penelitian ini yaitu, berhasil menciptakan sebuah prototipe sistem dialog interaksi manusia dan mesin berbasis IoT pada kendaraan mobil dengan mencapai rata-rata akurasi perintah berhasil 89,5% untuk menyalakan dan mematikan aliran listrik yang terhubung pada ESP32.

Kata kunci— finite state machine, internet of things, sistem dialog.

I. PENDAHULUAN

Gangguan mengemudi merupakan salah satu ancaman kecelakaan yang sering terjadi saat berkendara. Hal ini dikarenakan pengemudi kehilangan fokus dimana ada sesuatu yang dapat mengalihkan perhatian mereka saat mengemudi [1]. Selanjutnya, pengemudi umumnya dapat mengurangi gangguan mengemudi dengan mematikan telepon saat mengemudi. Gangguan berkendara dapat merugikan diri sendiri atau orang di sekitarnya, sehingga hal ini harus kita antisipasi dengan serius [14].

Penelitian teknologi berbasis Internet of Things (IoT) telah berkembang dari tahun ke tahun karena akan menjadi solusi yang bagus untuk membantu masalah aktivitas manusia dengan memberikan lebih banyak kenyamanan dan kemudahan kepada penggunanya. IoT bertujuan untuk membangun lingkungan yang cerdas dengan mengontrol objek melalui internet untuk pengambilan keputusan [12][2]. Ini dapat digunakan untuk diterapkan ke dalam peralatan mobil seperti lampu dekat, lampu jauh, AC, dll.

Pengenalan ucapan adalah teknik untuk mengenali kata-kata perintah dari suara manusia dan menerjemahkannya

menjadi data yang dapat dimengerti oleh sistem. Hal tersebut memungkinkan dapat menjadi solusi yang lebih efektif dan alami karena pengguna tidak perlu terganggu oleh mesin saat ingin menggunakannya [21][6].

Sistem dialog dapat memberikan informasi dengan mudah kepada pengguna karena sistem dialog akan membuat interaksi pengguna dan mesin, dimana sistem dapat membuat percakapan seperti pertanyaan tanya jawab [9] [18] [11]. Data dari pengenalan suara dapat digunakan oleh sistem dialog setelah diolah menggunakan library SpeechRecognition python untuk text-to-speech dan library gTTS untuk speech-to-text untuk mengirimkan informasi dari sistem ke pengguna [20].

Sastrawi dapat digunakan untuk pra-pemrosesan teks, data dari pengenalan ucapan akan diubah menjadi beberapa tahap dengan tujuan mendapatkan kata kunci [23]. Untuk mengelola kata kunci, terdapat algoritma model Finite State Machine untuk membuat keputusan dan mengelola setiap data berdasarkan input pengguna. Ini akan memberikan pemicu yang memungkinkan perubahan status berdasarkan kondisi status baru [16][10][5].

Berdasarkan penejelasan sebeulmya, penelitian ini mengusulkan prototipe sistem dialog dalam bahasa Indonesia untuk mengendalikan peralatan mobil menggunakan algoritma finite state machine. Sistem ini dapat membantu pengemudi memberikan perintah suara untuk mengontrol peralatan mobil. Prototipe ini dapat diintegrasikan dengan beberapa perangkat keras. Alat utama akan menggunakan Raspberry Pi, yang terhubung ke dua perangkat seperti mikrofon & speaker untuk interaksi pengguna. Selain itu, ada juga ESP32 untuk mengontrol peralatan mobil dengan menerima data dari basis data real-time berbasis sistem dialog.

II. KAJIAN TEORI

A. Pengenalan Ucapan

Pengenalan ucapan adalah proses identifikasi ucapan berdasarkan kata yang diucapkan dengan melakukan sebuah konversi sinyal yang diterima oleh perangkat perekam ucapan. Speech Recognition merupakan salah satu upaya yang digunakan untuk mengenali perintah kata dari ucapan manusia, kemudian menerjemahkannya menjadi sebuah data yang dapat dipahami dan dimanfaatkan oleh komputer [6].

Terdapat istilah Automatic Speech Recogniton (ASR) yang pada umumnya digunakan untuk mengubah ucapan menjadi

teks (*Speech-to-text*). Selain itu, ASR dapat digunakan untuk otentikasi biometrik yang mengotentikasi pengguna dari ucapan. Proses identifikasi atau pengenalan ASR dapat digunakan untuk melakukan tugas berdasarkan intruksi yang dipahami oleh komputer[7].

Manfaat dari pengenalan ucapan ini adalah dapat menangkap data ucapan yang diberikan oleh manusia lalu mengubahnya menjadi teks dalam waktu yang cepat. Data yang ditangkap dan diketahui dapat dimanfaatkan untuk sebuah aplikasi seperti sistem dialog dengan cara mencocokkan data tersebut berdasarkan kamus kata yang sudah ditentukan[4].

B. Sistem Dialog

Sistem Dialog adalah sebuah sistem yang dapat melakukan interaksi hubungan antara manusia dan komputer dalam bahasa alami dengan tujuan untuk menghilangkan perintah atau prosedur yang kompleks. Pada umumnya, sistem dialog digunakan untuk berkomunikasi dengan pengguna dan memberikan tanggapan yang relevan untuk memenuhi kebutuhan pengguna. Selain itu terdapat sistem dialog berorientasi tugas yang memiliki tujuan untuk mencapai tujuan tertentu seperti menemukan sebuah produk, lokasi atau sebuah informasi [8][9].

C. Internet of Things

Internet of Things (IoT) adalah perangkat yang dapat melakukan transfer data secara melalui jaringan internet. Tujuan dari IoT ini untuk membangun lingkungan digital (smart environment) seperti smart home atau smart car. Dengan memanfaatkan benda atau perangkat yang dibekali kemampuan sensorik dan komunikasi, sehingga dapat memperoleh sebuah data dan mengirimkannya melalui internet untuk pengambilan keputusan. Hal tersebut dapat membantu untuk mengatasi masalah kehidupan manusia seperti manajemen energi listrik, transportasi, kesehatan, bisnis dan lain-lain [19][3].

D. Text preprocessing

Text preprocessing adalah bagian dari Natural language processing (NLP) yang merupakan ilmu komputer yang menggunakan machine learning dan komputasi linguistik dengan pemrosesan bahasa alami manusia. Sistem komputasional linguistik memiliki berbagai macam tujuan, salah satu nya yaitu dapat membantu komunikasi antar manusia dan mesin, Metode *text preprocessing* merupakan proses yang penting untuk mengolah kata [23]. Terdapat empat langkah yang penting dalam preprocessing yaitu *Case Folding*, *Tokenizing*, *Stop-word removal* dan *Stemming*.



GAMBAR 1
TEXT PREPROCESSING

1. Case Folding

Pada tahap case folding yaitu, terdapat perubahan huruf apabila terdapat huruf yang kapital, misalnya kata "Halo" menjadi "halo".

2. Tokenizing

Pada tahap tokenizing yaitu digunakan untuk mengubah kalimat yang diberikan menjadi tokens, misalnya kalimat "halo siti" menjadi ["halo", "siti"].

3. Stop-Word Removal

Pada tahap Stop-Word Removal yaitu berfungsi sebagai menghilangkan kata-kata yang tidak penting, maka dengan tahapan ini dapat mengurangi kata-kata yang sama sekali tidak dibutuhkan oleh sistem. Contoh kata "yang", "dong" dan "aduh" tidak akan digunakan karena bukan sebuah kata baku.

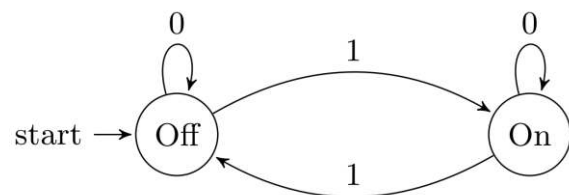
4. Stemming

Pada tahap Stemming yaitu berfungsi sebagai mengubah kata menjadi kata dasar. Contoh apabila terdapat kata "nyalain", maka akan diubah menjadi "nyala".

E. Finite State Machine

Finite State Machine (FSM) dalam matematika merupakan teori yang pertama kali dipelajari pada kuartal kedua abad ke-20 dengan kebutuhan objek yang terbatas dan tidak terbatas. Ide ini telah dikembangkan lebih lanjut untuk membentuk serutuh cabang teori formal hingga automata.

FSM terdiri dari keadaan diskrit yang direpresentasikan oleh *graph node* yang berbentuk lingkaran dan transisi yang direpresentasikan oleh *graph edges* yang berbentuk arah panah. FSM bekerja ketika transisi dipicu, maka mesin akan mengubah *state* (status) aktualnya dan membuat *transition* (transisi) ke keadaan lain yang sesuai dengan fungsi transisinya[16].



GAMBAR 2
CONTOH FSM

Pada Gambar 2 terdapat FSM yang memiliki dua status, kedua status tersebut merepresentasikan menyalakan perangkat dan mematikan perangkat. Transisi akan terjadi ketika tombol ditekan, apabila keadaan sedang mati lalu menekan tombol on, maka transisi akan terjadi dan status berubah menjadi on dan apabila keadaan dalam menyala lalu tombol off ditekan, maka transisi akan terjadi dan status berubah menjadi off.

F. Firebase

Firebase Real-time Database menggunakan sinkronisasi data setiap kali terjadi perubahan pada data, maka perangkat dan aplikasi yang terhubung dengan firebase akan menerima pembaruan data tersebut dengan cepat bahkan hingga milidetik. Firebase memiliki layanan berupa: *Firebase Analytics*, *Firebase Cloud Messaging*, *Firebase Auth*, *Real-time Database*, *Firebase Storage*, *Firebase Test Lab for Android*, dan *Firebase Notification* [17].

G. Software Development Life Cycle (SDLC)

1. Prototyping

Model Prototyping disebut model perancangan awal yang dibuat untuk menguji sebuah konsep proyek. Model Prototyping ini memungkinkan untuk mengevaluasi pemahaman langsung mengenai seorang developer dan aplikasi dapat dilakukan uji coba secara berulang-ulang sebelum fitur aplikasi pada tahap akhir. Selain itu, model Prototyping dapat membuat proses pembuatan sistem atau perangkat mejadi lebih terstruktur. Hasil dari model prototyping dapat berupa mockup yang akan menjadi rujukan model desain, hasil tersebut diimplementasikan ketika akan melakukan uji coba, presentasi, penilaian sebuah desain atau kebutuhan lain nya

2. Incremental

Model incremental sangat baik diterapkan apabila ketika pengguna menginginkan banyak perubahan para suatu aplikasi. Proses pada model memiliki beberapa arah, pendekatan ini sangat baik untuk proyek yang inovatif karena waktu iterasi yang singkat dapat membuat tim untuk menyelesaikan tugas nya dengan cepat.

3. Spiral

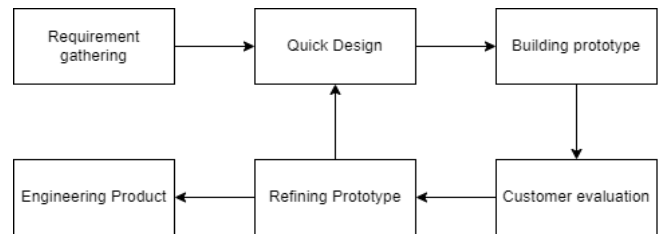
Model *spiral* ini memungkinkan untuk menggabungkan kemajuan eksekusi yang berulang dengan beberapa bagian yang sistematis dan tertata dengan baik yang ditampilkan dalam representasi berurutan. Model ini memungkinkan peluncuran produk yang dibuat akan diproses secara bertahap atau penyempurnaan bertahap melalui siklus dalam kurva heliks

4. Rapid Application Development (RAD)

Model RAD menggabungkan model *Prototyping* dan *Iterative Advancements* tanpa melibatkan perancangan spesifik. Pada model ini, seluruh tim di awal proses menentukan persyaratan dan desain proyek, Selama tahap perkembangan, kebutuhan yang telah diperoleh dapat digunakan untuk menghasilkan solusi yang lebih baik. Model RAD ini sangat baik digunakan untuk proyek yang sangat dinamis.

III. METODE

Perancangan sistem pada penelitian ini akan menggunakan model *prototyping*. Alasan pemilihan ini karena sistem yang dibangun akan sering dilakukan beberapa kali percobaan sebelum mencapai tahap akhir dan akan memiliki sedikit perubahan yang besar dikarenakan fitur dan fungsi yang dibuat akan dibatasi berdasarkan kebutuhan yang sudah ditetapkan pada tahap awal perancangan atau perubahan terjadi apabila terdapat saran dari klien. Sehingga model ini akan sangat mendukung untuk perancangan proyek penelitian.



GAMBAR 3
MODEL PROTOTYPING

1. Requirement Gathering

Dalam tahap ini mencari dan mendapatkan kebutuhan informasi sistem lalu didefinisikan dengan rinci. Pengumpulan kebutuhan dapat berupa informasi yang relevan dari jurnal penelitian sebelumnya dan kebutuhan perangkat dan aplikasi yang mendukung untuk merancang sistem penelitian.

2. Quick Design

Setelah kebutuhan pada tahapan sebelumnya terpenuhi, maka pada tahap *Quick Design* yaitu dilakukan integrasi persyaratan dari hasil tahapan sebelumnya dengan pembuatan desain sederhana, dengan tujuan untuk memberikan gambaran singkat tentang sistem yang akan dibuat untuk dijadikan sebuah *prototype*.

3. Building Prototype

Setelah *Quick Design* telah berhasil dibuat, maka selanjutnya adalah perancangan *prototype* yang akan dijadikan rujukan untuk pembuatan proyek.

4. Customer Evaluation

Setelah *prototype* selesai dibuat, maka selanjutnya dilakukan pengujian lebih lanjut dengan melakukan evaluasi bersama klien/pembimbing untuk dilakukan tinjauan dan pemberian saran.

5. Refining Prototype

Jika klien tidak memiliki perubahan atau perbaikan dari *prototype* yang telah dibuat, maka penelitian dapat dilanjutkan pada tahap berikutnya. Namun, jika terdapat catatan untuk perubahan atau perbaikan maka tahap 4 dan 5 akan terus dilakukan berulang sampai kebutuhan klien terpenuhi.

6. Engineering product

Pada tahap *Engineering Product* proyek yang dirancang sudah dapat digunakan, dilanjutkan dengan proses pengembangan dan perawatan perangkat atau aplikasi secara berkala agar tetap berjalan dengan baik.

IV. HASIL DAN PEMBAHASAN

A. Analisa Kebutuhan Sistem

Analisa kebutuhan pada penelitian ini, memiliki spesifikasi sistem perangkat keras dan perangkat lunak yang disajikan pada Tabel 1 berikut.

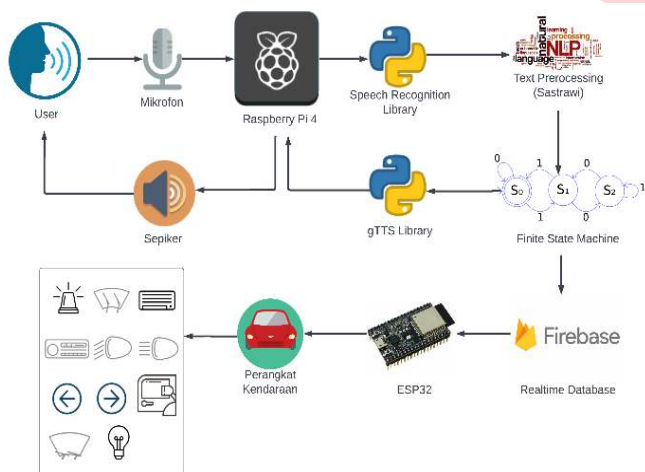
TABEL 1
KEBUTUHAN PERANGKAT KERAS

| NO | Kebutuhan Sistem | |
|----|-------------------|---|
| | Hardware/Software | Deskripsi |
| 1 | Raspberry Pi 4.0 | Komputer berukuran kecil untuk penyimpanan program dan mesin pengendali utama |

| NO | Kebutuhan Sistem | |
|----|-----------------------------------|---|
| | Hardware/Software | Deskripsi |
| 2 | ESP32 | Mikrokontroler untuk menerima data dan menyalakan atau mematikan arus listrik |
| 3 | Sepiker | Perangkat yang dapat memberikan informasi suara |
| 4 | Mikrofon | Perangkat yang dapat menerima ucapan <i>user</i> |
| 5 | LED | Lampu yang akan menjadi representasi perangkat kendaraan |
| 6 | Pustaka gTTS | Pustaka untuk mengubah teks menjadi ucapan |
| 7 | Pustaka <i>Speech Recognition</i> | Pustaka untuk mengubah ucapan menjadi teks |
| 8 | Pustaka Sastrawi | Pustaka untuk text preprocessing |
| 9 | Firebase | <i>Real-time</i> database untuk penyimpanan data. |

Berdasarkan hasil analisa tersebut, sistem yang akan dilakukan perancangan dapat menghasilkan sebuah produk prototipe yang dapat digunakan langsung oleh *user*.

B. Gambaran Umum Sistem



GAMBAR 4
GAMBARAN UMUM SISTEM

Proses sistem dialog dapat dimulai dengan pengenalan suara oleh mikrofon yang sudah terintegrasi dengan Raspberry Pi 4.0, kemudian akan mengubah data suara menjadi text oleh *library Speech Recognition*. Data yang telah diubah tersebut akan memasuki proses *text preprocessing* untuk mendapatkan kata kunci, kemudian dikelola oleh FSM untuk pengambilan keputusan, hasil dari keputusan tersebut akan dikirimkan langsung ke firebase *real-time* database dan di terima oleh ESP32 untuk mengontrol perangkat-perangkat kendaraan. *Output* dari sistem ini terdapat dua, yaitu respon audio yang diberikan oleh sistem kepada *user* dan sebuah aksi yang dapat menyalakan atau mematikan sebuah perangkat-perangkat kendaraan.

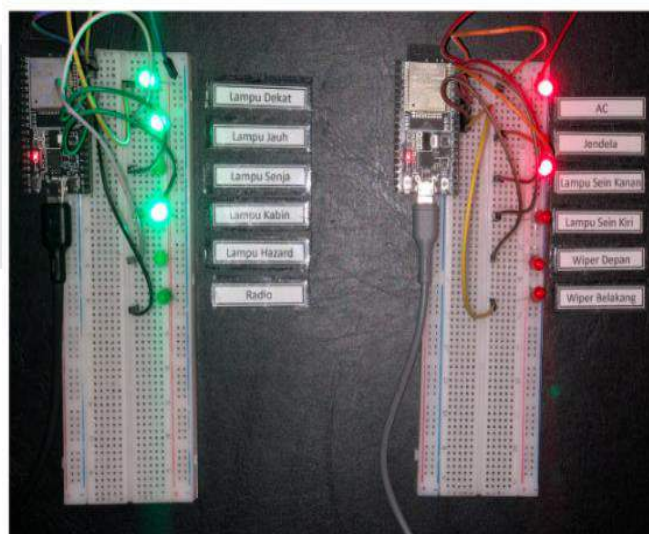
C. Implementasi Perangkat Keras

Kebutuhan perangkat pada perancangan prototipe penelitian akan menggunakan beberapa perangkat, keterangan kebutuhan perangkat akan disajikan dalam Tabel 2.

TABEL 2
KEBUTUHAN PERANGKAT KERAS

| NO | Kebutuhan Perangkat Keras | |
|----|---------------------------|--------|
| | Nama perangkat | Jumlah |
| 1 | Raspberry Pi 4.0 | 1 |
| 2 | ESP32 | 2 |
| 3 | Sepiker | 1 |
| 4 | Mikrofon | 1 |
| 5 | LED | 12 |

Setelah dilakukan pengumpulan perangkat kebutuhan. Selanjutnya, dilakukan implementasi perangkat-perangkat prototipe. Implementasi prototipe dibangun dengan pemasangan mikrofon dan sepiker telah dihubungkan dengan Raspberry Pi. Kemudian, ESP32 ditanamkan pada breadboard dan terhubung dengan 12 LED sebagai representasi perangkat kendaraan. Berikut hasil implementasi perancangan dapat disajikan dalam Gambar 5.

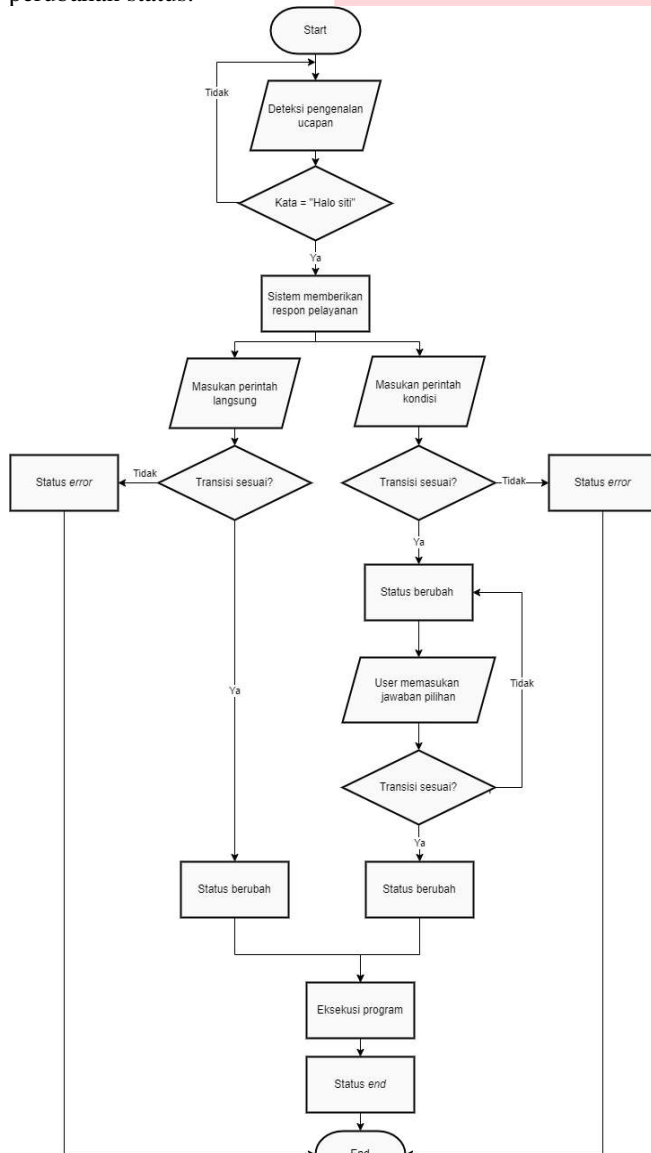


GAMBAR 1
IMPLEMENTASI PERANGKAT KERAS

D. Implementasi Sistem Dialog

Sistem dialog akan menerima pengenalan ucapan *user* berhasil yang telah berhasil diubah menjadi teks oleh pustaka Speech Recognition yang direkam oleh mikrofon pada Raspberry Pi. Data teks tersebut dilakukan *Text Preprocessing* yang berfungsi untuk memproses kata dari kalimat pengenalan ucapan yang user berikan. Kalimat tersebut akan dilakukan beberapa tahapan mulai dari *tokenization*, menghapus tanda baca, *case folding*, *stop word removal*, *stemming*, dan *sorting*. Hasil dari *text preprocessing* yaitu, mendapatkan kata kunci yang dapat dimanfaatkan oleh sistem dialog untuk menentukan keputusan.

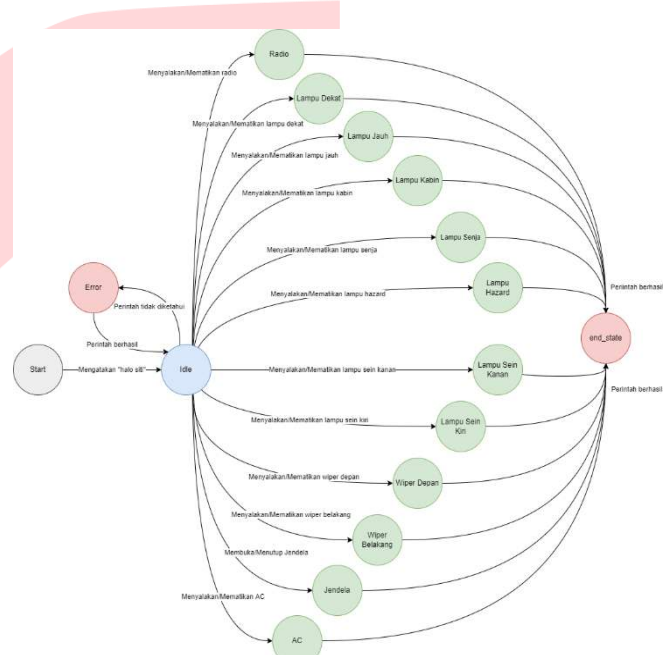
Perancangan diagram alir FSM disajikan dalam Gambar 6. Sistem dimulai dengan *user* mengucapkan perintah berupa perintah langsung atau perintah. Apabila perintah telah sesuai maka sistem dapat mengeksekusi perintah tersebut dan mengubah status pada FSM, namun apabila perintah tidak diketahui maka sistem akan mengubah status ke status *error* lalu sistem kembali ke tahap awal. Dalam perintah kondisi terdapat proses status yang tetap, sehingga tidak mengalami perubahan status.



GAMBAR 6
DIAGRAM ALIR FSM

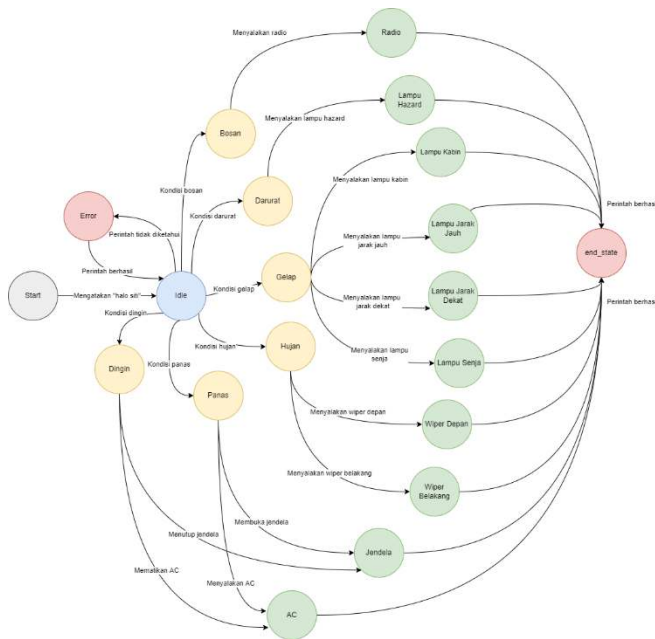
Pada Gambar 7 merupakan sekumpulan status pada algoritma FSM untuk sistem dialog. Seperti yang sudah dijelaskan sebelumnya, status dapat dipicu apabila terdapat perintah yang memacu sistem (transisi) yang merupakan input dari user sehingga status dapat berubah.

Pertama sistem memulai ketika user mengucapkan perintah ucapan, pengguna berada pada status *start* yang harus mengucapkan "Halo SITI" sehingga sistem akan memberikan respon perintah diterima, kemudian diikuti dengan perintah yang ingin dilakukan user, lalu status akan berubah menjadi *idle* dan mengubahnya lagi sesuai dengan perintah yang diberikan, namun apabila perintah user tidak diketahui oleh sistem, maka status sistem akan tetap pada status dimiliki pada saat itu sampai sistem dapat menerima perintah yang sesuai.



GAMBAR 7
FSM PERINTAH LANGSUNG

Perintah kondisi FSM disajikan pada Gambar 8. Perbedaan pada perintah kondisi yaitu, sistem dapat membangun komunikasi dua arah untuk memberikan pertanyaan kepada user untuk mendapatkan persetujuan. Transisi pada perintah kondisi berjumlah 20 transisi berdasarkan pada Tabel 1 mulai dari kode dialog D26 sampai D31. Contoh perintah kondisi, apabila user mengucapkan kondisi panas "aduh hari ini panas" kemudian sistem akan memberikan pertanyaan "Apakah ingin membuka jendela atau menyalakan AC?". Selanjutnya, user dapat menjawab sesuai dengan keinginan user, contoh user menjawab "AC" dan sistem akan memberikan respon serta aksi mengendalikan perangkat tersebut.



GAMBAR 8
FSM PERINTAH KONDISI

E. Pengujian

Pengujian dilakukan langsung dengan responden yang terdiri dari 20 orang dengan usia mulai dari 7 tahun sampai 70 tahun dan terdapat 30% perempuan serta 70% laki-laki. Pengujian dilakukan dengan cara responden mengucapkan setiap perintah ucapan yang tersedia pada Tabel 3 dengan jumlah 31 skenario dialog.

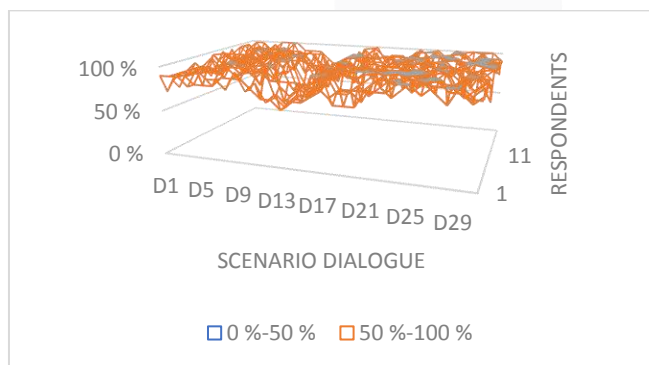
TABEL 3
SKENARIO DIALOG

| Kode | Skenario Deskripsi | Input pengenalan ucapan user | Output Sistem |
|------|-------------------------|------------------------------|--|
| D1 | Pembukaan | Halo SITI | Memberikan respon ucapan |
| D2 | Menyalakan lampu dekat | Tolong nyalakan lampu dekat | Memberikan respon ucapan dan menyalakan LED (Lampu Dekat) |
| D3 | Mematikan lampu dekat | Tolong matikan lampu dekat | Memberikan respon ucapan dan mematikan LED (Lampu Dekat) |
| D4 | Menyalakan lampu jauh | Tolong nyalakan lampu jauh | Memberikan respon ucapan dan menyalakan LED (Lampu Jauh) |
| D5 | Mematikan lampu jauh | Tolong matikan lampu jauh | Memberikan respon ucapan dan mematikan LED (Lampu Jauh) |
| D6 | Menyalakan lampu kabin | Nyalakan lampu kabin | Memberikan respon ucapan dan menyalakan LED (Lampu Kabin) |
| D7 | Mematikan lampu kabin | Matikan lampu kabin | Memberikan respon ucapan dan mematikan LED (Lampu Kabin) |
| D8 | Menyalakan lampu senja | Hidupkan lampu senja | Memberikan respon ucapan dan menyalakan LED (Lampu Senja) |
| D9 | Mematikan lampu senja | Matikan lampu senja | Memberikan respon ucapan dan mematikan LED (Lampu Senja) |
| D10 | Menyalakan lampu hazard | Hidupkan lampu hazard | Memberikan respon ucapan dan menyalakan LED (Lampu Hazard) |

| | | | |
|-----|-----------------------------|---------------------------|--|
| D11 | Mematikan lampu hazard | Matikan lampu hazard | Memberikan respon ucapan dan mematikan LED (Lampu Hazard) |
| D12 | Menyalakan lampu sein kiri | Nyalakan lampu sein kiri | Memberikan respon ucapan dan menyalakan LED (Sein Kiri) |
| D13 | Mematikan lampu sein kiri | Matikan lampu sein kiri | Memberikan respon ucapan dan mematikan LED (Sein Kiri) |
| D14 | Menyalakan lampu sein kanan | Nyalakan lampu sein kanan | Memberikan respon ucapan dan menyalakan LED (Sein Kanan) |
| D15 | Mematikan lampu sein kanan | Matikan lampu sein kanan | Memberikan respon ucapan dan mematikan LED (Sein Kanan) |
| D16 | Menyalakan radio | Nyalain radio | Memberikan respon ucapan dan menyalakan LED (Radio) |
| D17 | Mematikan radio | Matikan radio | Memberikan respon ucapan dan mematikan LED (Radio) |
| D18 | Menyalakan wiper depan | Menyalakan wiper depan | Memberikan respon ucapan dan menyalakan LED (Wiper Depan) |
| D19 | Mematikan wiper depan | Matikan wiper depan | Memberikan respon ucapan dan mematikan LED (Wiper Depan) |
| D20 | Menyalakan wiper belakang | Nyalakan wiper belakang | Memberikan respon ucapan dan menyalakan LED (Wiper Belakang) |
| D21 | Mematikan wiper belakang | Matikan wiper belakang | Memberikan respon ucapan dan mematikan LED (Wiper Belakang) |
| D22 | Membuka jendela | Membuka jendela | Memberikan respon ucapan dan menyalakan LED (Jendela) |
| D23 | Menutup jendela | Menutup jendela | Memberikan respon ucapan dan mematikan LED (Jendela) |
| D24 | Menyalakan AC | Nyalakan AC | Memberikan respon ucapan dan menyalakan LED (AC) |
| D25 | Mematikan AC | Matikan AC | Memberikan respon ucapan dan mematikan LED (AC) |
| D26 | Kondisi panas | Aduh saya panas | Memberikan respon ucapan |
| | Menyalakan AC | Nyalakan AC aja | Memberikan respon ucapan dan menyalakan LED (AC) |
| | Membuka jendela | Buka jendela | Memberikan respon ucapan dan menyalakan LED (Jendela) |
| D27 | Kondisi dingin | Hari ini dingin | Memberikan respon ucapan |
| | Mematikan AC | Ya, matikan AC | Memberikan respon ucapan dan menyalakan LED (AC) |
| | Menutup Jendela | Ok, tutup jendela | Memberikan respon ucapan dan menyalakan LED (Jendela) |
| D28 | Kondisi Darurat | Ini darurat | Memberikan respon ucapan |
| | Menyalakan lampu hazard | Ya, nyalakan lampu hazard | Memberikan respon ucapan dan menyalakan LED (Lampu Hazard) |
| | Mematikan lampu hazard | Tidak perlu | Memberikan respon ucapan dan mematikan LED (Lampu Hazard) |
| D29 | Kondisi hujan | Hari ini hujan | Memberikan respon ucapan |
| | Menyalakan wiper depan | Nyalakan wiper depan | Memberikan respon ucapan dan menyalakan LED (Wiper Depan) |

| | | | |
|------------|---------------------------|-------------------------|--|
| | Menyalakan wiper belakang | Nyalakan wiper belakang | Memberikan respon ucapan dan menyalakan LED (Wiper Belakang) |
| D30 | Kondisi gelap | Ini gelap banget | Memberikan respon ucapan |
| | Menyalakan lampu jauh | Nyalakan lampu jauh | Memberikan respon ucapan dan menyalakan LED (Lampu Dekat) |
| | Menyalakan lampu dekat | Nyalakan lampu dekat | Memberikan respon ucapan dan menyalakan LED (Lampu Jauh) |
| | Menyalakan lampu kabin | Nyalakan lampu kabin | Memberikan respon ucapan dan menyalakan LED (Lampu Kabin) |
| | Menyalakan lampu senja | Nyalakan lampu senja | Memberikan respon ucapan dan menyalakan LED (Lampu Senja) |
| D31 | Kondisi bosan | Aku bosan | Memberikan respon ucapan |
| | Menyalakan radio | Ya, nyalakan radio | Memberikan respon ucapan dan menyalakan LED (Radio) |
| | Mematikan radio | Tidak perlu | Memberikan respon ucapan dan mematikan LED (Radio) |

Pada Gambar 9 menunjukan grafik akurasi dan hasil dari pengujian. Data hasil pengujian sistem dialog interaksi manusia dan mesin berjumlah 6.200 pengujian (20 orang x 31 dialog x 10 kali) dengan akurasi rata-rata mencapai 89%. Kemudian, akurasi pengujian dengan tingkat tertinggi terjadi pada skenario dialog dengan deskripsi kondisi panas (D26) yaitu mencapai 98,5%, namun nilai terendah terdapat pada deskripsi dengan skenario matikan sein kiri (D13) yaitu mencapai 70,5% karena ketika user mengucapkan “matikan sein kiri”, sistem memahami perintah tersebut menjadi “matikan sendiri”. Oleh karena itu, perintah tersebut menjadi tidak diketahui.



GAMBAR 9
GRAFIK HASIL PENGUJIAN

V. KESIMPULAN

Perancangan perangkat IoT pada sistem dialog menggunakan sensor ucapan dan aktuator sepiker yang dihubungkan pada Raspberry Pi dapat menerima ucapan user dan memberikan respon dari mesin dengan tepat. Kemudian, ESP32 dapat terkoneksi dengan jaringan internet dan mampu menerima data dari firebase *real time* database untuk mengelola perangkat yang terhubung. Pengembangan sistem dialog menggunakan Finite State Machine dapat berjalan dengan baik. FSM dapat membuat keputusan untuk mengelola setiap status dialog pada sistem. Saran yang dapat diberikan untuk pengembangan pada sistem dialog selanjutnya adalah perlu dilakukan pemilihan aplikasi pihak ketiga atau

pengembangan terkait pengenalan ucapan agar dapat menerima ucapan manusia lebih akurat sehingga mesin dapat memahami dengan jelas dan benar.

REFERENSI

- [1] J. A. Horne and A. L. Reyner, "Driver sleepiness," *European Sleep Research Society*, pp. 23-29, 1995.
- [2] H. Fakhurroja, A. S. Prihatmanto, A. Purwarianti, C. Machbub and Riyanto, "Integration of Indonesian Speech and Hand Gesture Recognition for Controlling Humanoid Robot," *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1590-1595, 2018.
- [3] V. Pravalika and C. R. Prasad, "Internet of Things Based Home Monitoring and Device Control Using Esp32," *International Journal of Recent Technology and Engineering (IJRTE)*, pp. 58-62, 2019.
- [4] E. Fytrakis, I. Georgoulas, J. Part and Y. Zhu, "Speech-Based Home Automation System," *British HCI Conference*, pp. 271-272, 2015.
- [5] H. Fakhurroja, C. Machbub, A. S. Prihatmanto and A. Purwarianti, "Multimodal Fusion Algorithm and Reinforcement Learning-Based Dialog," *International Journal on Electrical Engineering and Informatics*, pp. 1016-1046, 2020.
- [6] N. Anggraini, A. Kurniawan, L. K. Wardhani and N. Hakiem, "Speech Recognition Application for the Speech Impaired using the Android-based Google Cloud Speech API," *TELKOMNIKA*, vol. 16, no. 6, p. 2733-2739, 2018.
- [7] W. Mustikarini, R. Hidayat and A. Bejo, "Real-Time Indonesian Language Speech Recognition with MFCC Algorithms and Python-Based SVM," *IJITEE*, Vol. 3, No. 2, pp. 55-60, 2019.
- [8] Y. Fan, X. Luo and P. Lin, "A Survey of Response Generation of Dialogue Systems," *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:14, No:12*, pp. 461-472, 2020.
- [9] Z. Zhang and M. Huang, "Memory-Augmented Dialogue Management for Task-Oriented Dialogue Systems," *ACM Transactions on Information Systems*, Vol. 37, No. 3, pp. 34:1-34:30, 2019.
- [10] H. Fakhurroja, A. S. Prihatmanto and C. Machbub, "Multimodal Interaction System for Home Appliances Control," *International Journal of Interactive Mobile Technologies (IJIM)* Vol.14, No.15, pp. 44-67, 2020.
- [11] Y. J. Zhao, Y. L. Li and M. Lin, "A Review of the Research on Dialogue Management of Task-Oriented Systems," *Australian International Academic Centre*, 2019.
- [12] P. Uma, P. Prasenjeet, B. Nilesh and A. Manvita, "Internet of Things based Integrated Smart Home Automation," *International Conference on Communication and Information Processing (ICCIP-2020)*, 2020.
- [13] B. S. Sarjerao and A. Prakasarao, "A Low Cost Smart Pollution Measurement System Using REST API and ESP32," *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018.
- [14] F. Alberto, U. Rubén, L. C. Juan and C. Rubén, "Driver Distraction Using Visual-Based Sensors and Algorithms," *Molecular Diversity Preservation International (MDPI)*, vol. 16, no. 1805, 2016.

- [15] G. Larionov, Z. Kaden, H. V. Dureddy, G. B. T. Kalejaiye, M. Kale, S. P. Potharaju, A. P. Shah and A. I. Rudnicky, "Tartan: A retrieval-based socialbot powered by a dynamic finite-state machine architecture," *2nd Proceedings of Alexa Prize*, 2018.
- [16] R. Balogh and D. Obdr̃z'alek, "Using Finite State Machines in Introductory Robotics," *International Conference on Robotics and Education RiE 2017*, pp. 85-91, 2018.
- [17] C. Khawas and P. Shah, "Application of Firebase in Android App Development-A Study," *International Journal of Computer Applications*, pp. 49-53, 2018.
- [18] M. Firdaus, H. Chauhan, A. Ekbal and P. Bhattacharyya, "EmoSen: Generating Sentiment and Emotion Controlled Responses in a Multimodal Dialogue System," *Journal of Latex Class Files*, Vol. 14, NO. 8, 2015.
- [19] M. Chernyshev, Z. Baig, O. Bello and S. Zeadally, "Internet of Things (IoT): Research, Simulators, and Testbeds," *IEEE Internet of Things Journal Volume: 5, Issue: 3*, pp. 1637-1647, 2018.
- [20] S. Khattar, A. Sachadeva, R. Kumar and R. Gupta, "Smart Home With Virtual Assistant Using Raspberry Pi," *9th International Conference on Cloud Computing, Data Science & Engineering*, pp. 576-579, 2019.
- [21] M. T. Tombeng, R. Najoan and N. Karel, "Smart Car: Digital Controlling System Using Android Smartwatch Voice Recognition," *The 6th International Conference on Cyber and IT Service Management (CITSM 2018)*, 2018.
- [22] I. S. Areni, A. D. Mufidah, I. S. Wahyuni, I. Nurtanio, I. R. Sahali and A. Bustamin, "Speech to Text in Indonesian Personal Assistant," *Journal of Computer Science*, vol. 15, no. 6, pp. 824-831, 2019.
- [23] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh and H. A. Gozali, "Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi," *IOP Conf. Series: Materials Science and Engineering*, 2019.