

IMPLEMENTASI SISTEM KONTROL STOPKONTAK ELEKTRONIK CERDAS MENGGUNAKAN AKTIVASI SUARA DAN INTEGRASI PENGENALAN WAJAH

Muhammad Amiruddin; Mohammad Nasrul Mubin, S.T., M.T

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Surakarta

Abstrak

Penelitian ini bertujuan untuk mengembangkan sebuah stopkontak elektronik cerdas dengan mengintegrasikan pengenalan wajah (*face recognition*) sebagai sistem keamanan dan pengenalan suara (*speech recognition*) sebagai sistem pengendalian. Sistem terdiri dari dua perangkat utama, yaitu perangkat pengendali dan perangkat terkendali. Perangkat pengendali menggunakan Orange Pi 3B untuk menjalankan *face recognition* guna mengidentifikasi pengguna. Jika wajah pengguna teridentifikasi sebagai orang yang terdaftar pada dataset sistem, sistem akan membuka akses dan mengaktifkan *speech recognition*. Pada *speech recognition*, perangkat akan merespons jika pengguna mengucapkan kata kunci yang telah ditentukan. Jika kalimat yang diucapkan sesuai dengan kata kunci, perangkat akan mengirim perintah ke *Firestore Real-time Database* untuk mengubah parameter tertentu. Selanjutnya, perangkat terkendali yang menggunakan ESP32 akan membaca nilai parameter dari *Firestore*. Jika terjadi perubahan nilai, ESP32 akan mengendalikan kondisi relay sehingga mengubah daya listrik pada perangkat yang terhubung dengan stopkontak. Pada pengujian pertama, diperoleh hasil bahwa sistem dapat beroperasi secara optimal menggunakan waktu perekaman suara selama tiga detik dengan panjang kata kunci maksimal terdiri dari tiga kata. Selain itu, hasil pengujian lainnya menunjukkan hasil bahwa alat dan sistem ini dapat beroperasi dengan baik dengan ditunjukkannya waktu respon perangkat pengendali yang kurang dari lima detik, waktu respon perangkat terkendali yang kurang dari sembilan detik, tingkat akurasi sistem *face recognition* sebesar 100% pada wajah yang terdaftar dan 40-60% pada wajah yang tidak terdaftar, serta tidak adanya perubahan performa alat yang signifikan baik alat menggunakan *casing* maupun tidak.

Kata kunci: stopkontak elektronik cerdas, pengenalan wajah, pengenalan suara, Orange Pi 3B, ESP32, Firestore Real-time Database.

Abstract

This research aims to develop a smart electronic socket that integrating face recognition as a security system and speech recognition as control system. The system consists of two main devices, namely the control device and the controlled device. The control device utilizes an Orange Pi 3B to perform face recognition for user identification. Upon successful identification, the system grants access and activates speech recognition. During speech recognition, the device responds to specific keywords spoken by the users. If the spoken words and the keywords matches, the device sends commands to the Firestore real-time database to modify specific parameters. Subsequently, the controlled device using ESP32 reads parameter values from Firestore. Upon value changes, ESP32 controls the relay condition to alter the electrical power of devices plugged into the socket. In the initial test, it was found that the system operates optimally with a three-second voice recording time with a maximum keyword length of three words. Additionally, other test results indicated that the device and system perform well with the response time of the controlling device is less than five seconds, the response time of the controlled device is

less than nine seconds, the face recognition system achieves 100% accuracy on registered faces and 40-60% accuracy on unregistered faces, and there is no significant change in device performance whether using a casing or not.

Keywords: smart electronic socket, face recognition, speech recognition, Orange Pi 3B, ESP32, Firebase Real-time Database.

1. PENDAHULUAN

Pengoperasian peralatan elektronik seperti menyalakan dan mematikan lampu, kipas angin, obat nyamuk elektrik, atau peralatan elektronik lainnya merupakan kegiatan yang sangat umum untuk dilakukan setiap harinya, terutama pada masyarakat di daerah perkotaan. Pengendalian beberapa peralatan elektronik tersebut umumnya masih dilakukan secara tradisional dengan cara melakukan kontak langsung terhadap objek. Proses pengendalian perangkat elektronik yang dilakukan secara tradisional ini terkadang membuat ketidaknyamanan terhadap pengguna peralatan elektronik karena harus selalu mendekati perangkat elektronik dan melakukan kontak langsung agar dapat mengoperasikan perangkat elektronik yang diinginkan (Rahman, 2019).

Di sisi lain, perkembangan teknologi yang terjadi pada beberapa dekade terakhir mengakibatkan adanya peningkatan kebutuhan akan kenyamanan dalam pengoperasian perangkat elektronik, terutama pada skala rumah tangga. Terdapat banyak perangkat elektronik cerdas yang dibuat dan dirancang guna membantu manusia dalam menjalankan aktivitasnya sehari - hari dimana sebagian besar perangkat elektronik cerdas tersebut telah memiliki kemampuan komunikasi yang baik sehingga memungkinkan proses pengendalian perangkat elektronik secara lebih mudah dan efisien (Maharani, 2023). Kemampuan interaksi antar perangkat elektronik cerdas ini lah yang kemudian menjadi asal terciptanya sistem Smart Home Assistant.

Smart Home Assistant merupakan sebuah sistem perangkat lunak yang digunakan untuk melakukan otomatisasi terhadap perangkat elektronik yang berada pada suatu rumah dengan memanfaatkan jaringan lokal maupun jaringan internet (Pratmanto, Fandhilah, & Saputra, 2019). Dalam pengaplikasiannya, sistem *smart home* biasa dijalankan menggunakan sebuah asisten AI (*artificial intelligence*) yang berbeda-beda sesuai dengan vendor penyedia *smart home*. Asisten AI ini memungkinkan terjadinya proses pengendalian perangkat elektronik dari jarak jauh menggunakan aplikasi smartphone, panel kontrol atau suara. Dengan adanya asisten ini, proses pengendalian perangkat elektronik oleh pengguna akan menjadi lebih nyaman karena tidak diperlukannya lagi kontak langsung terhadap objek seperti proses pengendalian perangkat elektronik secara tradisional.

Pada umumnya, proses pemasangan *smart home* dilakukan pada tahap pembangunan rumah karena sistem *smart home* membutuhkan infrastruktur yang stabil dan kuat. Jika sistem *smart home*

dibangun setelah rumah dibangun, maka perlu dilakukan modifikasi dan perubahan yang mungkin mengakibatkan biaya tambahan (Eliza, 2015). Oleh karena itu, alat-alat elektronik yang ingin dipasang pada rumah tersebut harus direncanakan secara teliti sebelum memulai pembangunan rumah.

Meskipun penggunaan sistem *Smart Home Assistant* dapat memberikan kemudahan dan kenyamanan, sistem ini memiliki beberapa kekurangan dalam pengimplementasiannya. Kelemahan tersebut adalah mahalnya biaya instalasi sistem. Berdasarkan data yang dirilis oleh Badan Pusat Statistik di Indonesia (BPS) pada tahun 2022, tingkat upah standar dan upah minimum rata-rata di hampir semua provinsi di Indonesia memiliki nilai yang tergolong rendah (Theodora, 2023). Pendapatan rata-rata masyarakat Indonesia yang rendah juga mengakibatkan kurangnya daya beli masyarakat terhadap perangkat elektronik yang mendukung dijalkannya sistem ini. Kekurangan lainnya dari sistem ini adalah terbatasnya perangkat elektronik yang dapat dikendalikan menggunakan sistem *smart home* sehingga mengakibatkan kebanyakan perangkat elektronik rumah tangga konvensional tidak dapat dijalankan menggunakan sistem tersebut.

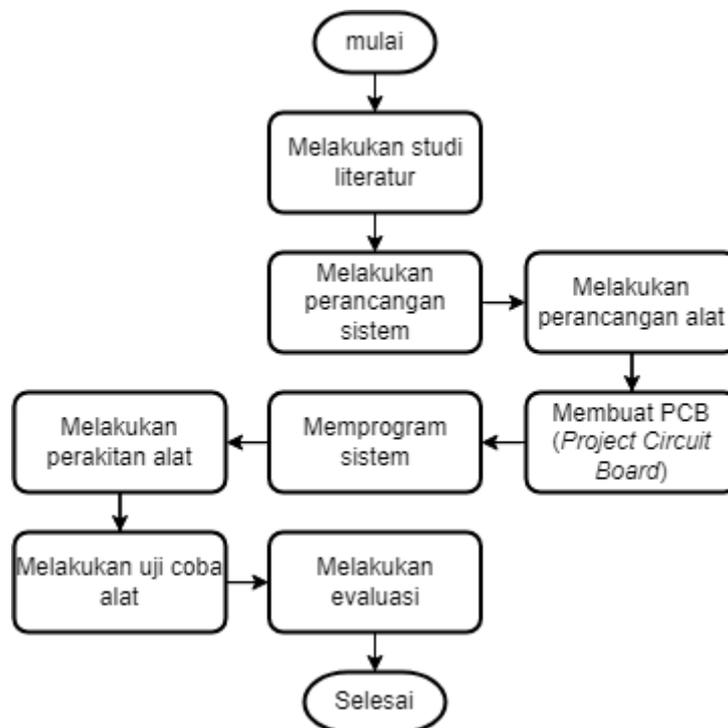
Dalam konteks kemajuan teknologi saat ini, penting untuk menciptakan solusi yang memudahkan pengguna dalam mengoperasikan perangkat elektronik rumah tangga dengan lebih efisien dan nyaman, namun tetap mempertimbangkan daya beli masyarakat. Oleh karena itu, dirancanglah sebuah sistem stopkontak elektronik cerdas yang mampu mengendalikan seluruh peralatan elektronik rumah tangga menggunakan perintah suara. Dengan menggunakan *single board computer* Orange Pi 3B sebagai perangkat kendali utama, sistem ini memungkinkan pengguna untuk memberikan perintah melalui suara dan mengirimkan perintah kendali ke stopkontak cerdas yang dituju melalui jaringan internet secara terpusat. Penggunaan teknologi ini tidak hanya memungkinkan pengendalian peralatan elektronik dengan lebih mudah dan nyaman, tetapi juga lebih terjangkau karena tidak memerlukan modifikasi pada struktur kabel yang ada di rumah. Dengan demikian, sistem ini diharapkan dapat memberikan kontribusi positif dalam memenuhi kebutuhan akan kenyamanan dan efisiensi dalam pengoperasian perangkat elektronik di rumah tangga.

2. METODE

Dalam penelitian ini, digunakan metode R&D (*Research and Development*), suatu pendekatan yang terfokus pada perancangan, pengembangan, dan pengujian secara bertahap terhadap prototipe sistem kontrol stopkontak elektronik cerdas (Yuliani & Banjarnahor, 2021). Metode penelitian mengacu pada proses ilmiah yang digunakan oleh peneliti untuk mengumpulkan data dengan tujuan dan kegunaan tertentu (Putri, 2020). Beberapa pertimbangan dalam penggunaan metode R&D mencakup:

- a. Metode R&D menghasilkan produk atau model yang memiliki validitas tinggi karena melibatkan serangkaian uji coba lapangan dan validasi oleh para ahli.

- b. Metode R&D mendorong inovasi yang berkelanjutan pada produk atau model, menghasilkan produk yang sesuai dengan kebutuhan zaman dan memiliki nilai berkelanjutan yang baik.
- c. Metode R&D menghubungkan penelitian teoritis dengan praktis, memungkinkan pengembangan solusi yang relevan dan teruji secara empiris.
- d. Metode penelitian dalam R&D mencakup berbagai pendekatan, termasuk deskriptif, evaluatif, dan eksperimental, yang memungkinkan analisis yang komprehensif terhadap fenomena yang diteliti.



Gambar 1. Flowchart penelitian

Gambar 1 diatas menunjukkan diagram alur dari melaksanakan penelitian, dimana dilakukan serangkaian tahapan sebagai acuan dalam pelaksanaan penelitian dari awal hingga selesai. Terdapat delapan tahapan pada penelitian ini, dimulai dari studi literatur kemudian dilanjutkan perancangan sistem, perancangan alat, pembuatan PCB (*Project Circuit Board*), pemrograman sistem, perakitan alat, uji coba alat, dan diakhiri dengan evaluasi. Rincian tahapan-tahapan tersebut, yaitu:

a. Studi literatur

Pada tahap ini akan diuraikan beberapa kebutuhan terkait pencarian referensi yang berkaitan dalam proses penelitian, mulai dari perancangan alat, pengembangan alat, pengujian alat, serta evaluasi keseluruhan sistem yang diteliti.

b. Perancangan sistem

Perancangan sistem dan alat dilakukan sebelum tahap implementasi dengan tujuan untuk memberikan ilustrasi tahapan yang dikerjakan. Pada tahap ini, diilustrasikan rancangan sistem secara menyeluruh, antara lain bagaimana cara sistem beroperasi, fitur apa saja yang akan digunakan pada sistem, serta bagaimana uji coba sistem dijalankan nantinya.

c. Perancangan alat

Pada tahap ini, diilustrasikan rancangan alat secara menyeluruh, antara lain apa saja komponen elektronik yang akan digunakan, bagaimana spesifikasi komponen yang diperlukan, serta seberapa besar batasan kemampuan dari alat yang akan diteliti.

d. Pembuatan PCB

Pada tahap ini, diilustrasikan rancangan PCB secara menyeluruh, antara lain bagaimana penempatan komponen elektronik yang akan digunakan, bagaimana jalur rangkaian elektronik yang dibuat nantinya, serta bagaimana bentuk dari PCB.

e. Programming sistem

Pada tahap ini, dilakukan pembuatan program untuk sistem agar dapat beroperasi dengan optimal, antara lain bahasa pemrograman apa saja yang digunakan, *tools* apa saja yang digunakan, serta bagaimana alur kerja sistem beroperasi.

f. Perakitan alat

Pada tahap ini, dilakukan perakitan seluruh komponen elektronik yang diperlukan pada PCB serta dilakukannya pengujian sederhana untuk mengetahui bahwa alat dapat berfungsi dengan sesuai dan dapat digunakan untuk menjalankan sistem.

g. Uji coba alat

Pada tahap ini, dilakukan pengujian secara menyeluruh untuk mengetahui bahwa sistem keseluruhan dapat beroperasi dengan sesuai, baik dari sisi alat maupun dari sisi program.

h. Evaluasi

Setelah pengujian alat selesai dilaksanakan, dilakukanlah tahapan evaluasi sebagai tahapan terakhir guna menarik kesimpulan berdasarkan seluruh data pengujian yang telah didapat selama pengujian alat.

3. HASIL DAN PEMBAHASAN

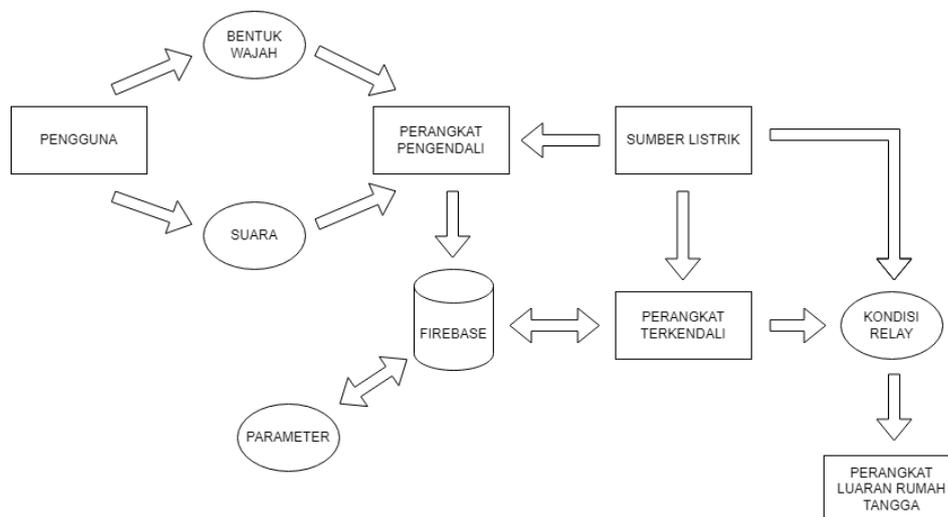
Pembuatan prototipe alat stopkontak elektronik cerdas ini ditujukan untuk membantu pengguna dalam mengendalikan kondisi mati dan nyala serangkaian perangkat elektronik rumah tangga. Alat ini akan

menerima sinyal suara yang diucapkan oleh pengguna melalui suatu mikrofon khusus dan diproses menggunakan *speech recognition*. Sinyal suara yang telah diproses akan diubah menjadi bentuk teks agar dapat diteruskan pada program berikutnya. Apabila teks yang telah diolah pada *speech recognition* sesuai dengan kata kunci tertentu pada sistem, maka perintah untuk menjalankan sistem akan dieksekusi dan alat akan dapat beroperasi.

Selain itu, prototipe alat ini juga dilengkapi dengan fitur pengenalan wajah (*face recognition*) yang memungkinkan sistem untuk dikunci dan dibuka aksesnya sesuai keperluan. Ketika sistem dibuka, alat akan merekam seluruh masukan suara yang terdengar pada mikrofon dan ketika suara pengguna sesuai dengan kata kunci yang diatur pada sistem maka perintah untuk menjalankan sistem akan dieksekusi. Sedangkan ketika sistem dikunci, alat tetap akan menerima semua masukan suara yang terdengar pada mikrofon namun perintah untuk menjalankan sistem tidak akan dieksekusi meskipun ucapan pengguna sesuai dengan kata kunci.

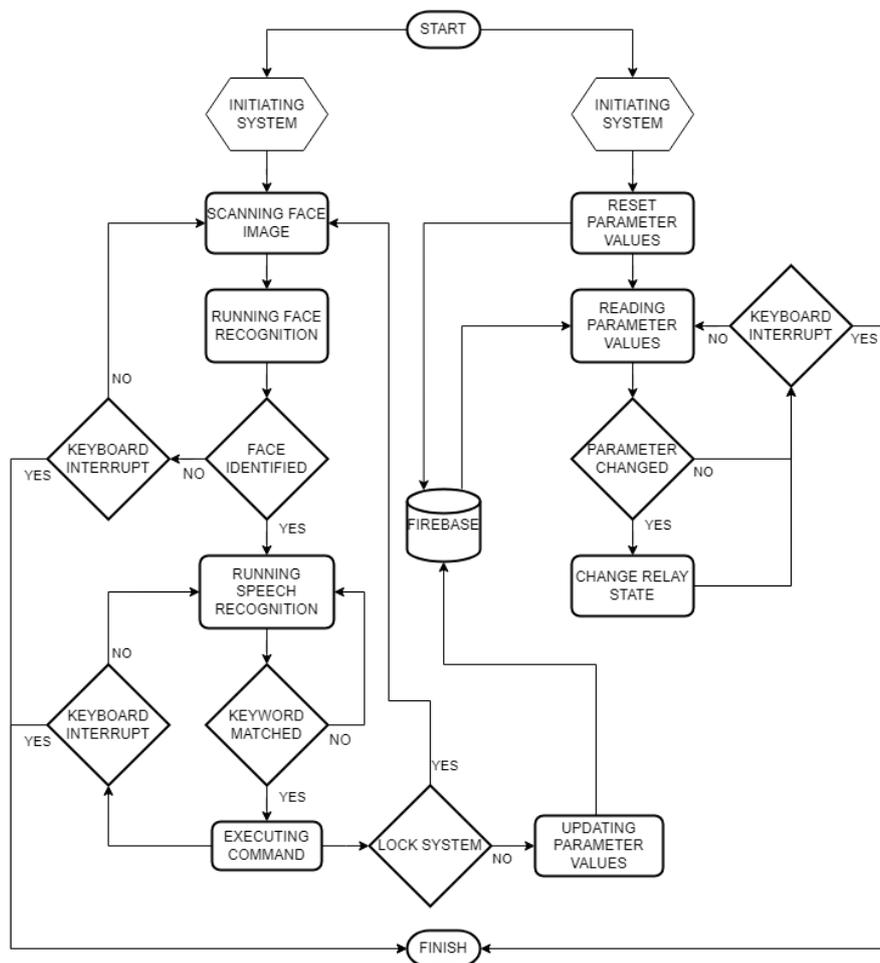
3.1 Blok Diagram Sistem

Rancangan sistem stopkontak elektronik cerdas yang telah diteliti terdiri dari dua perangkat utama, yaitu perangkat pengendali dan perangkat terkendali yang dihubungkan secara nirkabel menggunakan jaringan internet pada *firebase realtime-database*. Perangkat pengendali yang merupakan sebuah perangkat tunggal yang berfungsi untuk menjalankan pemrograman utama dari sistem, diantaranya *face recognition*, *speech recognition*, dan pengelolaan *database* sistem. Sedangkan perangkat terkendali merupakan perangkat stopkontak elektronik cerdas yang telah diprogram menggunakan mikrokontroler ESP32 dan dapat diduplikasi jumlahnya sesuai kebutuhan pengguna. Perangkat terkendali ini berfungsi untuk menyalurkan daya listrik menuju perangkat elektronik rumah tangga yang terpasang pada stopkontak elektronik cerdas sesuai dengan perintah yang diterima dari perangkat pengendali.



Gambar 2. Block Diagram Sistem

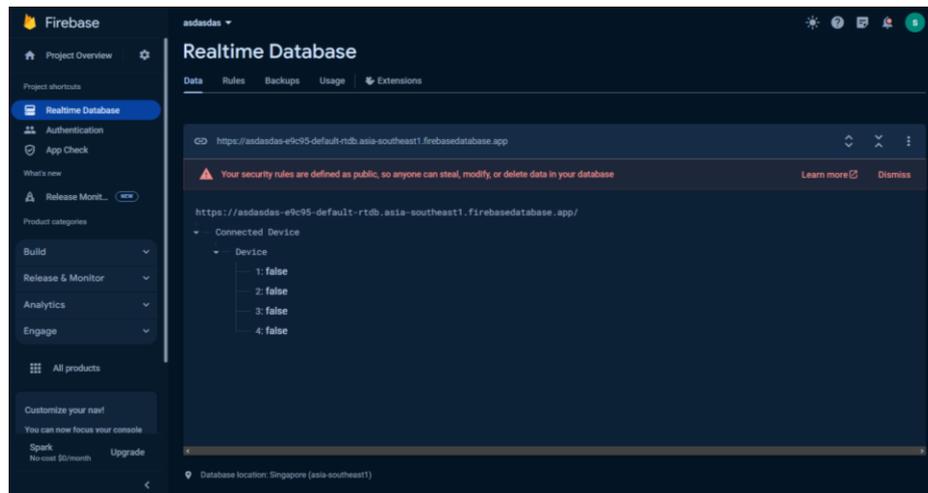
Gambar 2 diatas menunjukkan *block diagram* dari sistem stopkontak elektronik cerdas, dimana sistem berjalan pada dua perangkat yang terpisah. Perangkat pengendali memerlukan *input* berupa bentuk wajah pengguna yang didapat melalui *face recognition* serta kalimat yang diucapkan pengguna melalui *speech recognition*. Data input tersebut nantinya akan digunakan oleh perangkat pengendali untuk menentukan perubahan nilai parameter pada *firebase real-time database* menggunakan jaringan internet. Perubahan parameter yang dilakukan oleh perangkat pengendali hanya bersifat satu arah dimana perangkat pengendali dapat mempengaruhi nilai parameter pada *database* tetapi perubahan nilai parameter pada *database* tidak dapat mempengaruhi kinerja dari perangkat pengendali. Di sisi lain, perangkat terkendali tidak memerlukan *input* secara langsung dari pengguna melainkan perangkat in hanya akan membaca perubahan nilai parameter yang terjadi pada *firebase real-time database*. Perubahan parameter yang terjadi pada *database* mengakibatkan perangkat terkendali mengubah kondisi dari relay pada perangkat. Perubahan kondisi relay ini yang nantinya menyebabkan perangkat elektronik luaran yang terhubung ke perangkat terkendali dapat dihidupkan dan dimatikan sesuai dengan keperluan tanpa diperlukannya kontak secara langsung ke perangkat elektronik luaran. Alur Kerja Sistem



Gambar 3. Diagram alur kerja sistem

Gambar 3 diatas menunjukkan diagram alur kerja sistem stopkontak elektronik cerdas, dimana sistem berjalan pada dua perangkat yang terpisah. Perangkat tersebut adalah perangkat pengendali dan perangkat terkendali yang dihubungkan secara nirkabel menggunakan jaringan internet pada *firebase realtime-database*. Perangkat pengendali yang ditunjukkan pada diagram sebelah kiri beroperasi dengan melakukan pemindaian wajah terlebih dahulu menggunakan kamera sebagai fitur proteksi. Data gambar yang diperoleh kemudian dimasukkan pada sistem *face recognition* untuk mengidentifikasi apakah wajah pengguna terdaftar di dataset atau tidak. Apabila wajah pengguna teridentifikasi sebagai pengguna terdaftar maka sistem akan mulai menjalankan *speech recognition* untuk mengidentifikasi kata-kata yang diucapkan pengguna. Apabila kata yang diucapkan pengguna sesuai dengan kata kunci yang terdaftar pada sistem maka sistem akan mengeksekusi perintah dan mengganti parameter pada database. Perangkat terkendali yang ditunjukkan pada diagram sebelah kanan beroperasi dengan melakukan *reset* parameter di database terlebih dahulu. Selanjutnya perangkat akan membaca perubahan nilai parameter pada database terus menerus secara berkala. Apabila terjadi perubahan parameter pada database maka perangkat akan mengeksekusi perintah untuk mengubah kondisi relay pada stopkontak. Kedua perangkat akan beroperasi secara terus menerus hingga dilakukannya *keyboard interrupt* atau ketika perangkat keras dimatikan secara paksa. Dalam konteks pemrograman python, *keyboard interrupt* merupakan fitur bawaan bahasa python yang berfungsi menjalankan suatu perintah ketika pengguna menekan tombol kunci pada *keyboard* seperti kombinasi tombol *Ctrl + C* (Goyal, 2023). Pada sistem ini, *keyboard interrupt* digunakan menghentikan eksekusi program secara paksa tanpa perlu mematikan perangkat keras.

Ketika seluruh perangkat, baik perangkat pengendali maupun perangkat terkendali diaktifkan, perangkat tersebut akan mencoba menghubungkan diri ke *firebase database* yang telah dirancang khusus untuk berjalan pada sistem yang diteliti. Ketika perangkat tidak dapat terhubung dengan *database*, maka perangkat akan terus mencoba menghubungkan ke *database* secara berulang-ulang hingga perangkat benar-benar terhubung dengan *database*. Setelah perangkat terhubung dengan *database*, barulah perangkat dapat dioperasikan sesuai dengan program yang telah dibuat untuk menjalanka sistem.



Gambar 4. Penggunaan firebase realtime-database

Gambar 4 diatas menunjukkan penggunaan *firebase realtime-database* sebagai penghubung antara perangkat pengendali dengan perangkat terkendali. Penggunaan *firebase realtime-database* dilakukan untuk mengatasi batasan jarak yang dialami jika perangkat dihubungkan menggunakan kabel serta batasan penempatan alat oleh struktur bangunan jika perangkat dihubungkan menggunakan jaringan lokal (*local network*). Dengan digunakannya *firebase realtime-database* sebagai penghubung, seluruh perangkat pada sistem dapat saling berkomunikasi bahkan dari jarak yang jauh sekalipun serta dapat ditempatkan dimana saja pada rumah selama lokasi tersebut memiliki jaringan internet dan perangkat dapat terhubung dengan jaringan internet yang tersedia pada lokasi tersebut (Payara & Tanone, 2018). Hal ini membuat penempatan alat menjadi lebih fleksibel jika dibandingkan dengan penggunaan kabel atau jaringan lokal sebagai penghubung komunikasi antar perangkat.

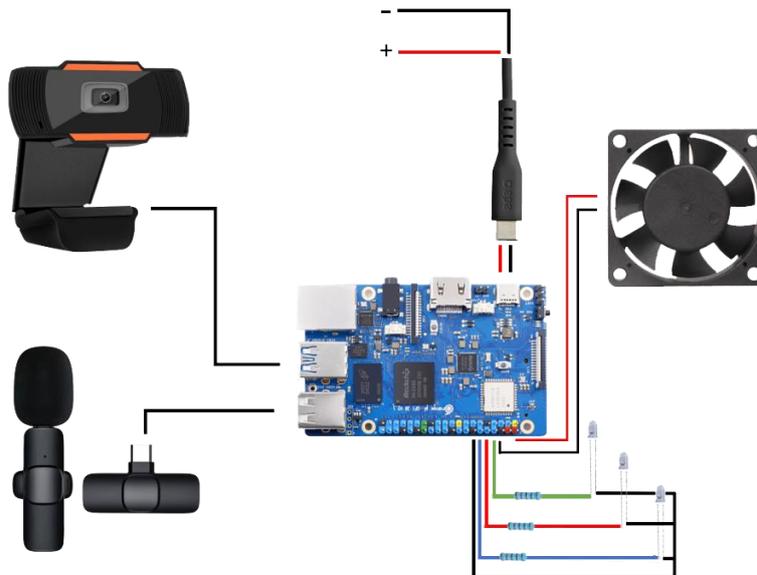
Sebagai pendukung dari fitur *face recognition*, sistem ini juga diintegrasikan dengan *google drive cloud storage*. Integrasi *google drive* ini akan mempermudah manajemen dataset wajah yang dilakukan oleh sistem sehingga pengguna tidak perlu lagi untuk mengunggah dan mengunduh ke penyimpanan *cloud* secara manual. Integrasi dilakukan pada saat pengambilan dataset wajah yang nantinya digunakan pada *face recognition* untuk dikenali oleh sistem. Setelah dataset wajah berhasil direkam dan dikumpulkan, dataset akan secara otomatis tersimpan secara lokal pada memori penyimpanan Orange Pi serta tersimpan secara *cloud* pada *google drive*. Pengguna juga dapat memberikan perintah kepada sistem untuk memperbarui dataset yang telah tersimpan secara lokal dari *google drive* yang terintegrasikan apabila pengguna melakukan perubahan data seperti mengubah nama folder foto atau menambahkan dataset wajah baru dari perangkat yang berbeda dari perangkat pengendali.

3.2 Rancangan Elektronika

Rancangan sistem stopkontak elektronik cerdas yang telah diteliti terdiri dari dua perangkat utama, yaitu perangkat pengendali dan perangkat terkendali yang masing-masing dihidupkan secara terpisah menggunakan sumber listrik yang berbeda. Pada perangkat pengendali, tidak dibuat PCB khusus dikarenakan seluruh komponen yang terpasang pada perangkat ini bersifat *plug and play*. Proses perancangan elektronika pada perangkat pengendali hanya berfokus pada pemilihan komponen serta penempatan posisi komponen pada *casing* alat. Namun, pada perangkat terkendali dibuat PCB khusus dikarenakan terdapat banyak komponen yang perlu disolder agar penempatannya rapih dan manajemen *port* mikrokontroller menjadi lebih efisien.



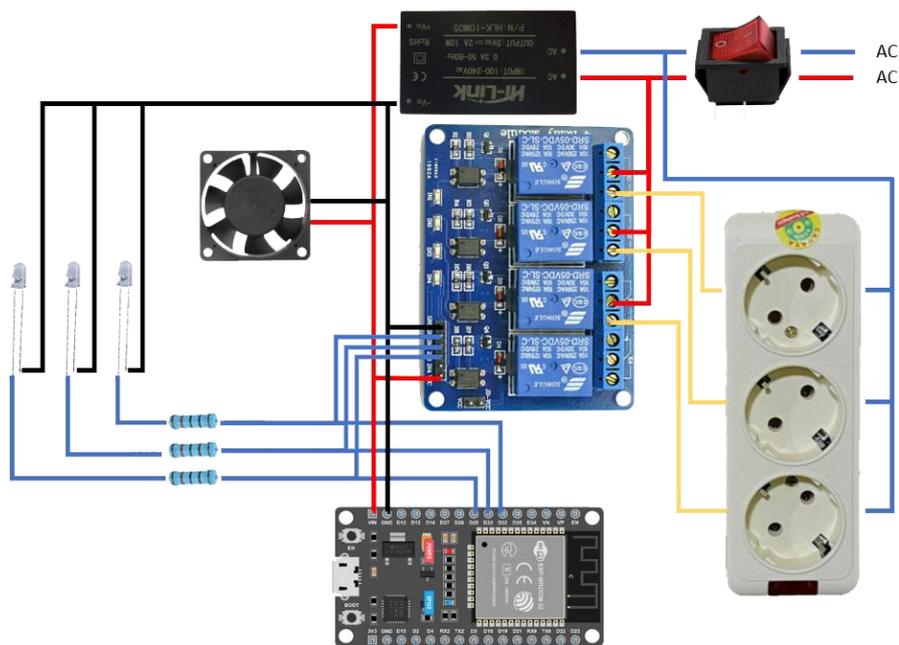
Gambar 5. Perangkat Pengendali



Gambar 6. *Wiring Diagram* Perangkat Pengendali

Gambar 5 diatas menunjukkan tampilan luar perangkat pengendali dan Gambar 6 diatas menunjukkan *wiring diagram* perangkat yang terdiri dari Orange Pi 3b, modul kamera yang dihubungkan menggunakan serial USB 2.0, mikrofon nirkabel yang dihubungkan menggunakan *port*

type-C, serta beberapa lampu led indikator yang dihubungkan menggunakan *port* GPIO dari Orange Pi. Untuk mendapatkan daya listrik, digunakan perangkat adaptor dc 5volt 15watt yang dihubungkan ke Orange Pi menggunakan *port* usb tipe-c. Sistem pendingin pada perangkat ini menggunakan dua buah *heatsink* tembaga yang diberikan aliran angin dari satu buah kipas dc 1watt berukuran 30 x 30 mm. OS (*Operating System*) dari Orange Pi disimpan dan dijalankan pada sebuah SSD (*Solid State Drive*) M.2 NVME berukuran 22 x 30 mm dengan jalur PCIe 2.0 x 1. Penggunaan SSD dibandingkan SD card (*Secure Digital Card*) sebagai penyimpanan pada Orange Pi ditujukan untuk meningkatkan performa dari Orange Pi dikarenakan SSD yang dijalankan pada jalur PCIe 2.0 x 1 memiliki kecepatan baca hingga 500 MB/s atau sekitar lima kali lebih cepat dibandingkan rata-rata SD card yang hanya memiliki kecepatan baca 100 MB/s (Orange Pi 3B, 2024).



Gambar 7. Wiring Diagram Perangkat Terkendali



Gambar 8. Perangkat Terkendali

Gambar 7 diatas menunjukkan *wiring diagram* dari perangkat terkendali sedangkan Gambar 8 menunjukkan tampilan luar dari perangkat terkendali. Perangkat ini terdiri dari serangkaian komponen elektronika dirancang secara manual dan disusun pada sebuah PCB berukuran 10 x 15 cm. Komponen utama dari perangkat ini adalah sebuah stopkontak yang telah dimodifikasi serta mikrokontroler ESP32 yang digunakan untuk menerima perintah pengguna dari *real-time database* kemudian mengatur kendali stopkontak secara elektrik. Pengendalian stopkontak menggunakan ESP32 dilakukan menggunakan modul relay 5v yang dilengkapi dengan komponen *optocoupler* pada setiap *channel*-nya. Penggunaan optocoupler pada tiap *channel* relay dilakukan untuk melindungi dan mengisolasi komponen elektronik dari *feedback* sinyal listrik yang tidak diinginkan. *Optocoupler* bekerja dengan cara memisahkan sirkuit kontrol dari sirkuit beban yang berdaya tinggi menggunakan cahaya sehingga sirkuit masukan dan sirkuit luaran dapat beroperasi secara terpisah tanpa menintervensi sirkuit satu sama lain (Kiljo, 2020). Untuk mendapatkan daya listrik, perangkat mengambil masukan listrik ac dari kabel stopkontak secara paralel kemudian diubah menjadi listrik dc 5volt 10watt menggunakan komponen konverter HLK-10M05. Daya listrik 5volt ini kemudian digunakan untuk menghidupkan mikrokontroler, kipas pendingin, serta relay pada perangkat. Sistem pendingin pada perangkat ini menggunakan hanya menggunakan aliran angin dari satu buah kipas dc 1watt berukuran 30 x 30 mm tanpa adanya *heatsink* sama sekali.

Dengan rancangan elektronika yang telah disebutkan, kedua perangkat baik perangkat pengendali maupun perangkat terkendali dapat beroperasi dengan baik selama dilakukannya pengujian tanpa adanya indikasi panas berlebih yang diakibatkan oleh suhu ruang disekitar perangkat dan suhu panas yang diakibatkan oleh beban listrik yang berlebih. Selain itu, selama pengujian berlangsung juga tidak didapati satu pun komponen elektronik yang mengalami kerusakan baik secara fisik maupun fungsinya. Hal ini menunjukkan bahwa rancangan elektronik dari alat telah sesuai dan dapat dioperasikan dengan baik.

3.3 Rancangan Program

Rancangan program stopkontak elektronik cerdas yang telah diteliti terdiri dari dua perangkat utama, yaitu perangkat pengendali dan perangkat terkendali yang masing-masing diprogram menggunakan bahasa pemrograman yang berbeda. Perangkat pengendali dioperasikan pada Orange Pi dan diprogram menggunakan bahasa pemrograman python versi 3.9 yang dijalankan pada OS ubuntu 20.04 (*Focal Fossa*). Program pada perangkat ini dibagi menjadi 4 file utama, yaitu main, gdrive, control, dan command_config. File main berisikan algoritma utama yang mengatur jalannya proses pengendalian sistem stopkontak elektronik cerdas. File gdrive berisikan algoritma untuk menghubungkan sistem dengan API *google drive* dan fungsi-fungsi untuk mengatur proses perpindahan berkas antara data

lokal yang terdapat di Orange Pi dengan data *cloud* yang tersimpan di *google drive*. File control berisikan fungsi-fungsi untuk mengendalikan sistem secara keseluruhan meliputi *face recognition*, *speech recognition*, kontrol *firebase realtime-database*, pengoperasian kamera, serta pengoperasian mikrofon. Fungsi-fungsi yang terdapat pada file control ini yang dijalankan oleh algoritma utama pada file main. File *command_config* merupakan sebuah file text yang berfungsi menyimpan kata kunci dari sistem yang digunakan oleh *speech recognition*. File *command_config* ini merupakan file yang boleh dimodifikasi oleh pengguna untuk mengkustomisasi kalimat yang digunakan untuk mengendalikan sistem.

```

14 #connect to webcam 0, with 720x720 px
15 cap = cv2.VideoCapture(0)
16 cap.set(3, 720)
17 cap.set(4, 720)
18
19 #load frontal-face classifier
20 facedetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
21
22 #scaling other features from main command class and enable it to run by keyboard
23 command = main_function(cap)
24 command.run_controller()
25 command.listen_in_background()
26
27 while True:
28     if not command.idle:
29         #record video webcam
30         frame = cap.read()
31         faces = facedetect.detectMultiScale(frame, 1.3, 5)
32         try:
33             for x, y, w, h in faces:
34                 #check either the face is real or just an image
35                 # result = command.verify_img(frame, verify_model, verify_label)
36                 # if result == "Live":
37                 result = command.recognize_face(frame[y:y+h+10, x:x+w+10], database)
38                 if result in command.folders:
39                     command.verif_count += 1
40                     print(command.verif_count)
41                     if command.verif_count > 20:
42                         command.run_system = True
43                         command.life = True
44                         print('run system')
45                         for xx, hh in faces:
46                             # do something here
47
48 while True:
49     if not command.idle:
50         try:
51             for xx, hh in faces:
52                 # do something here
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Gambar 9. Program main.py

Pada file main seperti yang ditunjukkan pada Gambar 9 diatas, algoritma yang dijalankan adalah algoritma utama dari sistem. Ketika program dijalankan, beberapa file dan variabel seperti model *machine learning*, alamat *port* usb, ukuran gambar yang ditangkap kamera, dan direktori folder dari dataset akan diinisiasi. Setelah semuanya terinisiasi, sistem akan dijalankan secara iterasi. Kamera akan diaktifkan untuk melakukan pemindaian terhadap wajah pengguna dan ketika wajah pengguna berhasil dikenali maka sistem akan memasuki kondisi *idle*, yaitu kondisi dimana sistem tidak lagi mengaktifkan pemindaian kamera namun sistem akan hanya mendengarkan suara yang ditangkap oleh mikrofon untuk dikenali. Apabila suara yang ditangkap oleh sistem sesuai dengan kata kunci yang telah dirancang maka sistem akan menjalankan perintah yang diinginkan, tetapi apabila suara yang ditangkap tidak sesuai dengan kata kunci yang telah dirancang maka sistem tidak akan melakukan apapun.

```

1 from googleapiclient import httplib2, MediaFileUpload
2 from googleapiclient import create_service
3 import os
4 from gdrive import download_folder
5
6
7 CLIENT_SECRET_FILE = 'client_secret.json'
8 API_NAME = 'drive'
9 API_VERSION = 'v3'
10 SCOPES = ['https://www.googleapis.com/auth/drive']
11 service = create_service(CLIENT_SECRET_FILE, API_NAME, API_VERSION, SCOPES)
12
13 except Exception as e:
14     print(e)
15
16 def delete_contents_in_folder(folder_path):
17     pass
18
19 def create_folder(folder_url, folder_name=None):
20     pass
21
22 def create_new_file(source_path, folder_url, file_name):
23     pass
24
25 def upload_datasets(source_url):
26     pass
27
28 def download_folders(url, new_path=None):
29     pass
30
31 def download_and_overwrite(url, new_path=None):
32     pass
33
34 def update_current_dataset():
35     pass
36
37 # It is essential to update at least one file, otherwise, the system will fail.
38 #
39 source = "Dummies"

```

Gambar 10. Program gdrive.py

Pada file gdrive seperti yang ditunjukkan pada gambar 10 diatas, algoritma yang dijalankan adalah konektivitas antara sistem dengan *google drive*. Koneksi antara sistem dengan *google drive* dilakukan menggunakan API yang harus diaktifkan terlebih dahulu pada pengaturan akun *google*. Setelah pengaturan API diaktifkan, file *client_secret.json* dari *google drive* dapat diunduh dan dijalankan pada program. Dengan adanya file tersebut, koneksi antara program dengan *google drive* dapat dilakukan dengan menggunakan beberapa fungsi tambahan yang telah dimodifikasi agar sistem dapat terhubung dengan folder pada *google drive* dengan hanya menggunakan URL (*Uniform Resource Locator*) nya saja. Fungsi tambahan tersebut antara lain fungsi untuk membuat folder baru, menghapus folder, menambahkan file baru kedalam folder, serta mengunduh folder dari *google drive*.

```

1 from keyboard import KeyboardEvent, KeyboardEvent, KeyboardEvent, KeyboardEvent
2 import numpy as np
3 from PIL import Image, ImageOps
4 import cv2
5 from DeepFace import DeepFace
6 from natsort import natsorted
7 import speech_recognition as sr
8 import firebase_admin
9 from firebase_admin import db
10 import json
11 import gdrive
12
13 class main_function(KeyboardEvent):
14     def __init__(self, camera):
15         self.camera = camera
16         self.speech_text = ''
17         self.speech_recognizer = sr.Recognizer()
18         self.speech_listener = sr.Recognizer()
19         self.speech_listener.listen_for_speech()
20         self.speech_listener.listen_for_speech()
21         self.speech_listener.listen_for_speech()
22         self.speech_listener.listen_for_speech()
23         self.speech_listener.listen_for_speech()
24         self.speech_listener.listen_for_speech()
25         self.speech_listener.listen_for_speech()
26         self.speech_listener.listen_for_speech()
27         self.speech_listener.listen_for_speech()
28         self.speech_listener.listen_for_speech()
29         self.speech_listener.listen_for_speech()
30         self.speech_listener.listen_for_speech()
31         self.speech_listener.listen_for_speech()
32         self.speech_listener.listen_for_speech()
33         self.speech_listener.listen_for_speech()
34         self.speech_listener.listen_for_speech()
35         self.speech_listener.listen_for_speech()
36         self.speech_listener.listen_for_speech()
37         self.speech_listener.listen_for_speech()
38         self.speech_listener.listen_for_speech()
39         self.speech_listener.listen_for_speech()
40         self.speech_listener.listen_for_speech()
41         self.speech_listener.listen_for_speech()
42         self.speech_listener.listen_for_speech()
43         self.speech_listener.listen_for_speech()
44         self.speech_listener.listen_for_speech()
45         self.speech_listener.listen_for_speech()
46         self.speech_listener.listen_for_speech()
47         self.speech_listener.listen_for_speech()
48         self.speech_listener.listen_for_speech()
49         self.speech_listener.listen_for_speech()
50         self.speech_listener.listen_for_speech()
51         self.speech_listener.listen_for_speech()
52         self.speech_listener.listen_for_speech()
53         self.speech_listener.listen_for_speech()
54         self.speech_listener.listen_for_speech()
55         self.speech_listener.listen_for_speech()
56         self.speech_listener.listen_for_speech()
57         self.speech_listener.listen_for_speech()
58         self.speech_listener.listen_for_speech()
59         self.speech_listener.listen_for_speech()
60         self.speech_listener.listen_for_speech()
61         self.speech_listener.listen_for_speech()
62         self.speech_listener.listen_for_speech()
63         self.speech_listener.listen_for_speech()
64         self.speech_listener.listen_for_speech()
65         self.speech_listener.listen_for_speech()
66         self.speech_listener.listen_for_speech()
67         self.speech_listener.listen_for_speech()
68         self.speech_listener.listen_for_speech()
69         self.speech_listener.listen_for_speech()
70         self.speech_listener.listen_for_speech()
71         self.speech_listener.listen_for_speech()
72         self.speech_listener.listen_for_speech()
73         self.speech_listener.listen_for_speech()
74         self.speech_listener.listen_for_speech()
75         self.speech_listener.listen_for_speech()
76         self.speech_listener.listen_for_speech()
77         self.speech_listener.listen_for_speech()
78         self.speech_listener.listen_for_speech()
79         self.speech_listener.listen_for_speech()
80         self.speech_listener.listen_for_speech()
81         self.speech_listener.listen_for_speech()
82         self.speech_listener.listen_for_speech()
83         self.speech_listener.listen_for_speech()
84         self.speech_listener.listen_for_speech()
85         self.speech_listener.listen_for_speech()
86         self.speech_listener.listen_for_speech()
87         self.speech_listener.listen_for_speech()
88         self.speech_listener.listen_for_speech()
89         self.speech_listener.listen_for_speech()
90         self.speech_listener.listen_for_speech()
91         self.speech_listener.listen_for_speech()
92         self.speech_listener.listen_for_speech()
93         self.speech_listener.listen_for_speech()
94         self.speech_listener.listen_for_speech()
95         self.speech_listener.listen_for_speech()
96         self.speech_listener.listen_for_speech()
97         self.speech_listener.listen_for_speech()
98         self.speech_listener.listen_for_speech()
99         self.speech_listener.listen_for_speech()
100        self.speech_listener.listen_for_speech()

```

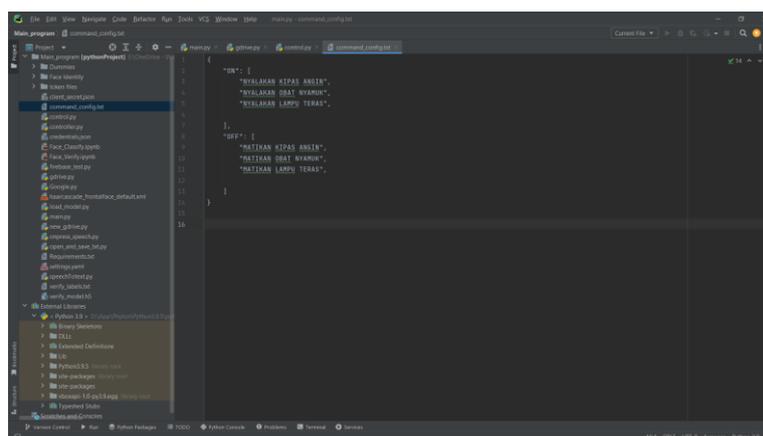
Gambar 11. Program control.py

Pada file control seperti yang ditunjukkan pada Gambar 11 diatas, algoritma yang dijalankan adalah pembuatan fungsi pengendalian sistem secara keseluruhan. Ketika program dijalankan, beberapa file dan variabel seperti model *machine learning*, alamat *port* mikrofon, direktori folder,

URL *database*, dan status sistem akan diinisiasi. Setelah semuanya terinisiasi, seluruh fungsi tidak dijalankan secara langsung melainkan fungsi akan diteruskan dan dieksekusi pada file main. Pada file ini terdapat dua fungsi yang sangat penting, yaitu *face recognition* dan *speech recognition*.

Program *face recognition* dijalankan dengan cara sistem mengambil *frame* gambar dari kamera secara *realtime* dan kontinu. *Frame* yang telah diambil tersebut kemudian akan dijalankan pada model *machine learning* pada library DeepFace dan dicocokkan dengan seluruh file gambar yang tersimpan di *dataset*. File yang tersimpan pada *dataset* dikumpulkan dan dipersiapkan pada tahapan *pre-processing*. Pada tahapan ini, dilakukan pengumpulan 49 file foto wajah yang disimpan pada sistem lokal dan *google drive*. Seluruh foto tersebut di potong dan hanya disisakan gambar wajahnya saja dengan skala persegi. Hal ini dilakukan untuk mengurangi gangguan dalam mendeteksi wajah dikarenakan pengaruh dari gambar latar belakang pengguna. Apabila program mendeteksi ada gambar dari *frame* dari kamera yang menyerupai gambar di *dataset* maka program akan menampilkan nama folder dimana gambar tersebut tersimpan. Sedangkan jika program tidak mendeteksi ada gambar dari *frame* kamera yang menyerupai gambar di *dataset* maka program akan menampilkan bahwa wajah tidak dikenali.

Program *speech recognition* dijalankan dengan cara sistem menangkap sinyal suara yang menggunakan mikrofon secara *realtime* dan kontinu. Sinyal suara yang didapat kemudian akan diidentifikasi dan diterjemahkan menjadi sebuah teks menggunakan model *machine learning* dari *google*. Dikarenakan proses ini menggunakan model *machine learning* dari *google* secara *online* maka dibutuhkan jaringan internet yang stabil agar proses *speech recognition* dapat berjalan dengan lancar. Setelah hasil teks didapatkan, program akan mencocokkan hasil teks yang didapat dari *speech recognition* dengan kata kunci yang terdapat pada file *command_config*. Agar proses perekaman suara dapat berjalan dengan optimal, proses *speech recognition* ini dijalankan secara *multi-threading* atau dijalankan pada *thread* program yang berbeda dari program utama.



Gambar 12. Program *command_config.txt*

Pada file `command_config` seperti yang ditunjukkan pada Gambar 12 diatas, tidak ada algoritma yang dijalankan. File ini hanya merupakan file tambahan dengan struktur penulisan berbentuk *dictionary* yang berfungsi untuk menyimpan seluruh kata kunci dari sistem yang digunakan oleh program *speech recognition*. File ini merupakan file yang boleh dimodifikasi oleh pengguna dengan cara menambah, menghapus, atau mengganti kata kunci yang terdapat pada *array* ON dan OFF. Jumlah kata kunci yang ditulis pada file ini dapat ditambahkan sesuai dengan jumlah perangkat terkendali serta jumlah masing-masing lubang stopkontak yang tersedia di masing-masing perangkat. Apabila jumlah kata kunci melebihi jumlah ketersediaan perangkat, tidak akan terjadi masalah perangkat. Namun, hanya beberapa kata kunci teratas sesuai dengan jumlah ketersediaan perangkat yang ketika dijalankan akan dapat mengendalikan perangkat.

Pada perangkat terkendali, program dijalankan pada perangkat ESP32 menggunakan bahasa pemrograman C++ Arduino yang diprogram menggunakan VSCode (*Visual Studio Code*) dengan *framework Platform IO*. Program pada perangkat ini dibagi menjadi dua file utama, yaitu `main` dan `config`. File `main` berisikan algoritma utama yang mengatur jalannya proses pengendalian sistem stopkontak elektronik cerdas. File `config` merupakan sebuah file yang berfungsi menyimpan parameter perangkat dari program yang digunakan oleh ESP32. File `config` ini merupakan file yang boleh dimodifikasi oleh pengguna untuk mengkustomisasi parameter perangkat yang digunakan untuk mengendalikan sistem.

```

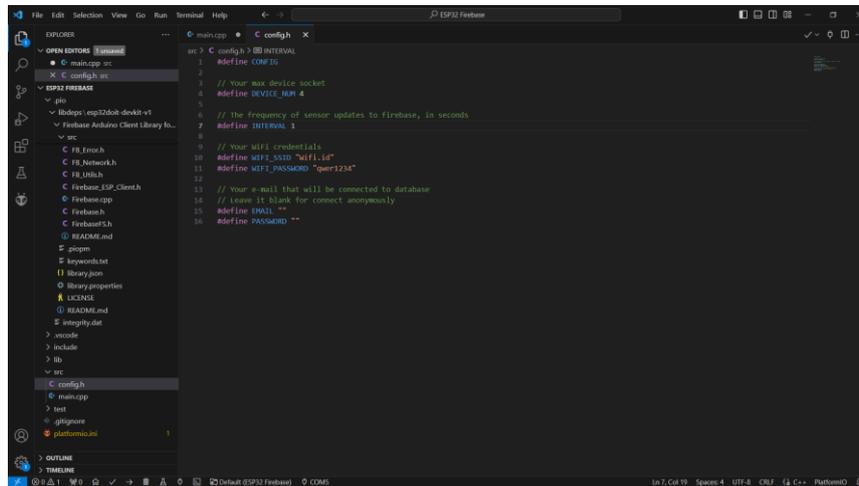
1 // main.cpp
2 #include <Arduino.h>
3 #include <Firebase.h>
4 #include <FirebaseAuth.h>
5 #include <FirebaseConfig.h>
6
7 // Stores the elapsed time from device start up
8 unsigned long elapsedMillis = 0;
9 // The frequency of sensor updates to firebase, in 0.01 seconds
10 unsigned long update_interval = INTERVAL * 100;
11
12 void WiFi_Init() {
13   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
14   Serial.println("Connecting to Wi-Fi");
15   while (WiFi.status() != WL_CONNECTED) {
16     Serial.print(".");
17     delay(300);
18   }
19   Serial.println();
20   Serial.println("Connected with IP: ");
21   Serial.println(WiFi.localIP());
22   Serial.println();
23 }
24
25 void firebase_init() {
26   // configure Firebase API key
27   config.api_key = API_KEY;
28   // configure Firebase realtime database url
29   config.database_url = DATABASE_URL;
30   // Enable API reconnection
31   Firebase.reconnectWiFi(true);
32   Serial.println("-----");
33   Serial.println("sign up new user...");
34   // Sign into Firebase anonymously
35   if (firebase.signin(config, EMAIL, PASSWORD)) {
36     Serial.println("Success");
37     isAuthenticated = true;
38   }
39   // Set the database path where updates will be loaded for this device
40 }
41
42 int main() {
43   WiFi_Init();
44   firebase_init();
45   while (true) {
46     // Your code here
47   }
48 }

```

Gambar 13. Program `main.cpp`

Pada file `main` seperti yang ditunjukkan pada Gambar 13 diatas, algoritma yang dijalankan adalah algoritma utama dari sistem perangkat terkendali. Ketika program dijalankan, variabel seperti model API *database*, URL *database*, *username*, *password*, dan konektivitas wifi akan diinisiasi. Setelah semuanya terinisiasi, sistem akan terlebih dahulu mematikan seluruh relay stopkontak pada perangkat lalu mengubah parameter di *database* menjadi *FALSE*. Setelah itu, dijalankan secara iterasi

untuk membaca adanya perubahan parameter pada *firebase real-time database* dimana jika parameter mengalami perubahan maka sistem akan memproses parameter tersebut menjadi luaran keadaan relay stopkontak. Ketika parameter pada *database* tidak mengalami perubahan maka sistem akan menjadi *idle* dan tidak melakukan apapun sampai terjadi perubahan parameter.

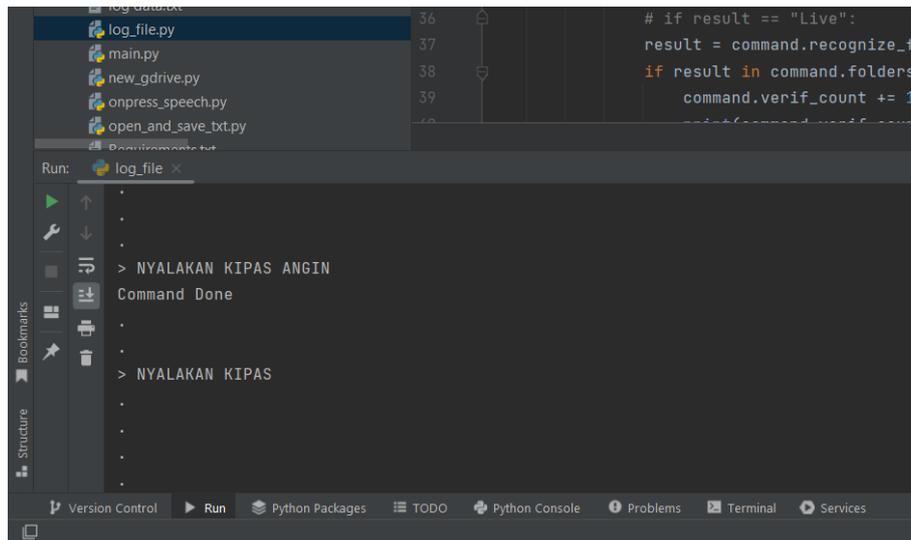


Gambar 14. Program config.h

Pada file config.h seperti yang ditunjukkan pada Gambar 14 diatas, tidak ada algoritma yang dijalankan. File ini hanya merupakan file tambahan yang berfungsi untuk menyimpan parameter sistem yang digunakan oleh perangkat terkendali. File ini merupakan file yang boleh dimodifikasi oleh pengguna dengan cara menambah, menghapus, atau mengganti variabel yang terdapat pada file. Nilai dari setiap parameter pada file ini harus disesuaikan dengan kondisi perangkat keras yang digunakan sesuai dengan instruksi yang tertulis pada file.

3.4 Pengujian Panjang Kata Kunci dan Durasi Perekaman Optimal dari *Speech Recognition*

Pengujian ini dilakukan untuk mengetahui berapa jumlah kata yang paling optimal untuk dijadikan kata kunci untuk menjalankan sistem serta mengetahui berapa durasi perekaman suara optimal sebelum suara diproses oleh sistem untuk diubah menjadi sebuah teks. Kondisi yang digunakan untuk pengujian adalah seluruh perangkat berupa perangkat pengendali, perangkat terkendali, serta laptop untuk memonitor sistem diletakkan pada satu ruangan yang sama dengan *router* wifi dan suhu udara yang dijaga menggunakan sebuah kipas angin berukuran 18 inci. Kondisi ini dipilih untuk mengetahui bagaimana hasil pengujian alat jika ditempatkan pada kondisi yang optimal baik dari sisi ketersediaan jaringan internet maupun suhu ruangnya.



Gambar 15. Pengujian panjang kata kunci

Gambar 15 diatas menunjukkan pengujian panjang kata kunci optimal yang dilakukan dengan cara mengamati *log* program yang tertulis. Proses pengujian dilakukan dengan cara mengucapkan lima kata kunci yang jumlah katanya berbeda dengan jumlah kata kunci paling sedikit satu kata dan paling banyak lima kata. Kata kunci yang diucapkan pada pengujian ini adalah KUNCI, NYALAKAN LAMPU, NYALAKAN KIPAS ANGIN, NYALAKAN LAMPU RUANG TAMU, dan NYALAKAN OBAT NYAMUK RUANG TENGAH. Kelima kata kunci tersebut diucapkan pada durasi perekaman yang berbeda, yaitu satu detik, dua detik, tiga detik, empat detik, dan auto. Masing-masing kata kunci pada satu durasi diuji sebanyak lima kali sehingga didapatkan data berapa kata kunci dan durasi perekaman kata yang paling optimal untuk diterapkan pada sistem. Berdasarkan pengujian yang dilakukan, diperoleh hasil seperti pada tabel berikut:

Tabel 1. Jumlah kalimat yang dikenali pada durasi perekaman dan kata kunci yang berbeda

Durasi (detik)	Panjang Kata Kunci	Jumlah Kalimat Dikenai	Jumlah Kalimat Tidak Dikenai
1	1	2	3
1	2	1	4
1	3	0	5
1	4	0	5
1	5	0	5
2	1	4	1
2	2	4	1
2	3	3	2
2	4	3	2

2	5	1	4
3	1	5	0
3	2	5	0
3	3	5	0
3	4	3	2
3	5	3	2
4	1	5	0
4	2	5	0
4	3	5	0
4	4	3	2
4	5	3	2
Auto	1	2	3
Auto	2	3	2
Auto	3	0	5
Auto	4	1	4
Auto	5	0	1

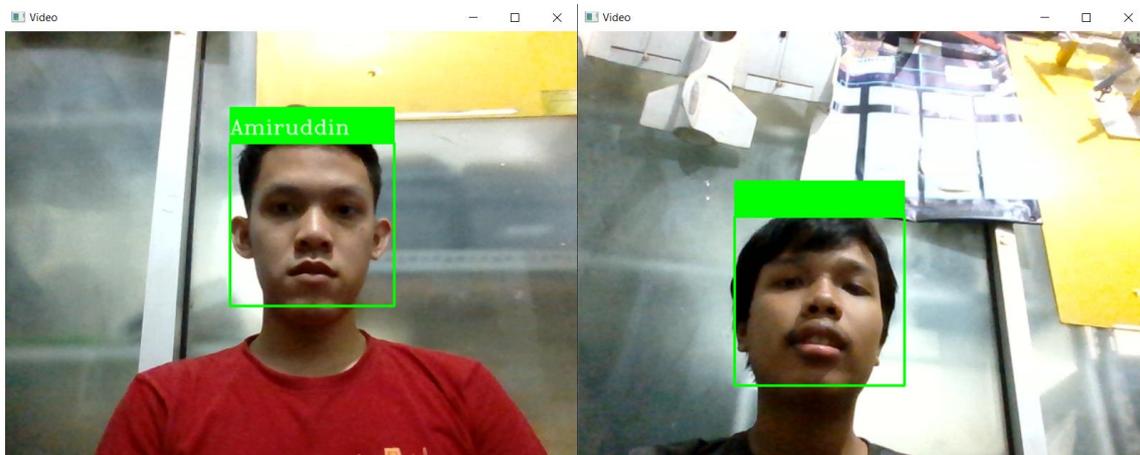
Berdasarkan Tabel 1 diatas, diperoleh hasil bahwa durasi perekaman selama satu detik dan auto memiliki kecenderungan kurang baik dalam mengenali kalimat, perekaman dengan durasi dua detik memiliki kecenderungan baik dalam mengenali kalimat, serta perekaman suara dengan durasi tiga dan empat detik memiliki kecenderungan sangat baik dalam mengenali kalimat. Pengujian dengan durasi satu detik memiliki hasil pengenalan kalimat paling rendah dimana perekaman pada durasi ini hanya bisa digunakan pada kata kunci dengan panjang satu sampai dua kata dengan jumlah kalimat yang dikenali kurang dari 50% per kata kuncinya. Pengujian dengan durasi auto memiliki hasil pengenalan kalimat yang rendah dimana perekaman pada durasi ini bersifat tidak pasti dimana terkadang sistem dapat mengenali kalimat dengan kata kunci panjang, namun terkadang juga tidak. Pengujian dengan durasi dua detik memiliki hasil pengenalan kalimat yang menengah dimana perekaman pada durasi ini bisa digunakan pada kata kunci dengan panjang satu sampai empat kata dengan jumlah kalimat yang dikenali lebih dari 50% per kata kuncinya, namun kurang dari 50% pada kata kunci dengan panjang lima kata. Pengujian dengan durasi tiga dan empat detik memiliki hasil pengenalan kalimat yang paling bagus dimana perekaman pada durasi ini bisa digunakan pada seluruh kata kunci dengan panjang satu sampai lima kata dengan jumlah kalimat yang dikenali lebih dari 50%.

Dikarenakan jumlah kalimat yang dikenali pada pengujian dengan durasi perekaman tiga dan empat detik tergolong identik, serta panjang kata kunci maksimal dengan persentase pengenalan 100%

berada pada kata kunci dengan panjang maksimal tiga kata maka diperoleh hasil pengujian bahwa durasi perekaman optimal pada sistem ini adalah tiga detik dengan panjang kata kunci optimal adalah maksimal tiga kata.

3.5 Pengujian Tingkat Akurasi dari *Face Recognition*

Pengujian ini dilakukan untuk mengetahui berapa tingkat akurasi dari sistem *face recognition* dalam mengenali wajah orang yang terdaftar pada dataset sistem serta memastikan bahwa sistem tidak salah dalam mengenali wajah orang yang tidak terdaftar pada dataset sebagai orang yang terdaftar pada dataset. Kondisi yang digunakan untuk pengujian adalah seluruh perangkat berupa perangkat pengendali, perangkat terkendali, serta laptop untuk memonitor sistem diletakkan pada satu ruangan yang sama dengan *router* wifi dan suhu udara yang dijaga menggunakan sebuah kipas angin berukuran 18 inci. Kondisi ini dipilih untuk mengetahui bagaimana hasil pengujian alat jika ditempatkan pada kondisi yang optimal baik dari sisi ketersediaan jaringan internet maupun suhu ruangnya.



Gambar 16. Pengujian akurasi sistem *face recognition*

Gambar 16 diatas menunjukkan pengujian akurasi sistem *face recognition* yang dilakukan dengan cara mengamati hasil gambar yang tampil pada program. Proses pengujian dilakukan dengan cara melakukan pemindaian wajah sebanyak lima kali pada kondisi pakaian dan pencahayaan yang sama pada tiap pemindaian. Pengujian dilakukan menggunakan wajah empat orang dimana dua orang telah didaftarkan pada dataset, sedangkan dua orang lainnya tidak didaftarkan pada *dataset*. Kondisi ini dibuat sehingga didapatkan data berapa kali sistem mengenali pemilik wajah terhadap dataset yang terdaftar dan berapa kali sistem salah dalam mengenali pemilik wajah yang tidak terdaftar pada dataset. Berdasarkan pengujian yang dilakukan, diperoleh hasil seperti pada tabel berikut:

Tabel 2. *Confusion Matrix* face recognition

Kondisi Dataset	Wajah Dikenali	Wajah Tidak Dikenali
Terdaftar	10	0
Tidak Terdaftar	5	5

Berdasarkan Tabel 2 diatas, dapat ditentukan nilai performa *Accuracy*, *Precision*, *Sensitivity*, dan *Specificity* dari sistem *face recognition*. Pada kasus ini :

- *True Positive* (TP) : Wajah terdaftar pada dataset dan wajah berhasil dikenali
- *True Negative* (TN) : Wajah tidak terdaftar pada dataset dan wajah berhasil tidak dikenali
- *False Positive* (FP) : Wajah tidak terdaftar pada dataset dan wajah berhasil dikenali
- *False Negative* (FN) : Wajah terdaftar pada dataset dan wajah berhasil tidak dikenali

Accuracy, yaitu rasio prediksi benar (positif dan negatif) dengan keseluruhan data

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100\%$$

$$Accuracy = \frac{(10 + 5)}{(10 + 5 + 0 + 5)} \times 100\%$$

$$Accuracy = 75\%$$

Precision, yaitu rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif

$$Precision = \frac{(TP)}{(TP + FP)} \times 100\%$$

$$Precision = \frac{(10)}{(10 + 5)} \times 100\%$$

$$Precision = 67\%$$

Sensitivity, yaitu rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif

$$Sensitivity = \frac{(TP)}{(TP + FN)} \times 100\%$$

$$Sensitivity = \frac{(10)}{(10 + 0)} \times 100\%$$

$$Sensitivity = 100\%$$

Specificity, yaitu rasio kebenaran memprediksi negatif dibandingkan dengan keseluruhan data negatif

$$Specificity = \frac{(TN)}{(FP + TN)} \times 100\%$$

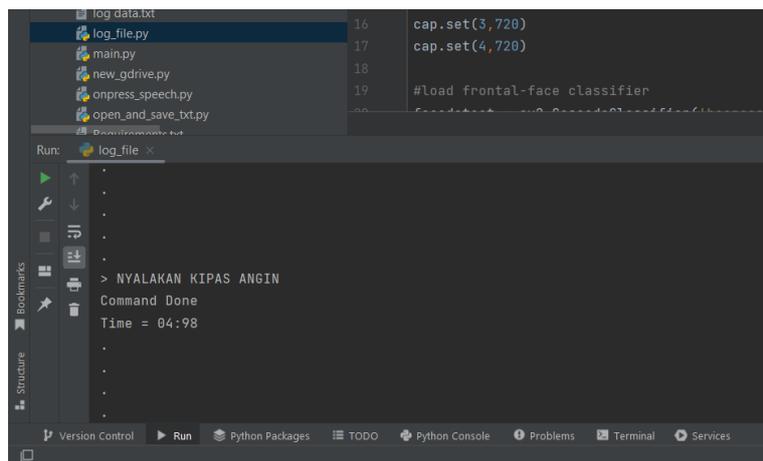
$$Specificity = \frac{(5)}{(5 + 5)} \times 100\%$$

$$Specificity = 50\%$$

Berdasarkan hasil perhitungan diperoleh hasil *Accuracy* sebesar 75%, *Precision* sebesar 67%, *Sensitivity* sebesar 100%, *Specificity* sebesar 50%. Dengan asumsi bahwa perangkat stopkontak elektronik cerdas ini digunakan di dalam rumah yang jarang terdapat orang asing di dalamnya, maka diperoleh hasil pengujian bahwa *face recognition* pada sistem ini tergolong layak untuk digunakan pada sistem sebagai bentuk proteksi agar sistem tidak diakses oleh orang asing secara bebas.

3.6 Pengujian Waktu Respon Perangkat Pengendali

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan oleh perangkat pengendali untuk menjalankan perintah *speech recognition* dan mengubah nilai parameter pada *firebase realtime-database*. Pada pengujian ini, digunakan hasil dari pengujian sebelumnya, yaitu pengujian panjang kata kunci dan durasi perekaman optimal dari *speech recognition* untuk menentukan variabel durasi perekaman suara dan panjang kata kunci *speech recognition*. Kondisi yang digunakan untuk pengujian adalah seluruh perangkat berupa perangkat pengendali serta laptop untuk memonitor sistem diletakkan pada satu ruangan yang sama dengan *router* wifi dan suhu udara yang dijaga menggunakan sebuah kipas angin berukuran 18 inci. Kondisi ini dipilih untuk mengetahui bagaimana hasil pengujian alat jika ditempatkan pada kondisi yang optimal baik dari sisi ketersediaan jaringan internet maupun suhu ruangnya.



```
log_data.txt
log_file.py 16 cap.set(3,720)
main.py 17 cap.set(4,720)
new_gdrive.py 18
onpress_speech.py 19 #Load frontal-face classifier
open_and_save_txt.py 20

Run: log_file x
> NYALAKAN KIPAS ANGIN
Command Done
Time = 04:98
```

Gambar 17. Pengujian waktu respon perangkat pengendali

Gambar 17 diatas menunjukkan pengujian waktu respon perangkat pengendali yang dilakukan dengan memanfaatkan *library datetime* pada bahasa pemrograman python. Proses pengujian dilakukan dengan cara menjalankan sistem secara khusus pada perangkat pengendali kemudian mengucapkan kata kunci pada *speech recognition* dengan parameter durasi perekaman dan jumlah kata kunci yang telah ditetapkan sebanyak tujuh kali. Waktu eksekusi perintah dihitung sejak huruf pertama kata kunci diucapkan hingga parameter pada *firebase realtime-database* sukses terganti. Apabila ada kata yang gagal dikenali oleh sistem, maka kata kunci akan diucapkan berulang dengan waktu yang terus berjalan hingga parameter pada *firebase realtime-database* sukses terganti. Kondisi ini dibuat sehingga didapatkan data berapa lama waktu rata-rata yang dibutuhkan sistem untuk mengeksekusi satu perintah. Berdasarkan pengujian yang dilakukan, diperoleh hasil seperti pada tabel berikut:

Tabel 3. Waktu respon perangkat pengendali

Percobaan ke	Durasi Perekaman (detik)	Jumlah Kata Kunci	Waktu Eksekusi Perintah (detik)
1	3	3	4.98
2	3	3	4.62
3	3	3	4.50
4	3	3	4.18
5	3	3	4.88
6	3	3	4.55
7	3	3	4.77

Berdasarkan Tabel 3 diatas, diperoleh hasil bahwa waktu respon perangkat pengendali pada seluruh percobaan berada dibawah lima detik dengan waktu eksekusi perintah paling cepat adalah 4.18 detik dan waktu eksekusi perintah paling lambat adalah 4.98 detik. Pada seluruh pengujian, kalimat yang diucapkan dapat secara langsung dikenali oleh sistem sebagai kata kunci pada pengucapan pertama dan perintah langsung dieksekusi. Tidak ada satupun kata kunci yang pengucapannya harus diulangi sebanyak dua kali atau lebih. Parameter durasi perekaman suara yang digunakan pada percobaan adalah tiga detik, maka waktu yang secara spesifik diperlukan untuk eksekusi perintah oleh sistem hingga parameter pada *real-time database* sukses terganti berkisar kurang dari dua detik dengan alat berada pada kondisi optimalnya.

Dikarenakan hasil pengujian respon perangkat pengendali menunjukkan waktu yang secara spesifik digunakan oleh sistem untuk mengubah parameter pada *real-time database* berkisar kurang dari dua detik maka diperoleh hasil pengujian bahwa respon perangkat pengendali untuk mengeksekusi perintah tergolong cepat pada durasi eksekusi berkisar satu sampai dua detik.

3.7 Pengujian Waktu Respon Perangkat Terkendali

Pengujian ini dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan oleh perangkat terkendali untuk membaca nilai dari *firebase real-time database* dan mengeksekusi perintah berdasarkan parameter tersebut. Kondisi yang digunakan untuk pengujian adalah seluruh perangkat berupa perangkat terkendali serta laptop untuk mengubah parameter dan memonitor sistem diletakkan pada satu ruangan yang sama dengan *router* wifi dan suhu udara yang dijaga menggunakan sebuah kipas angin berukuran 18 inci. Kondisi ini dipilih untuk mengetahui bagaimana hasil pengujian alat

jika ditempatkan pada kondisi yang optimal baik dari sisi ketersediaan jaringan internet maupun suhu ruangnya.



Gambar 18. Pengujian waktu respon perangkat terkendali

Gambar 18 diatas menunjukkan pengujian waktu respon perangkat terkendali yang dilakukan dengan memanfaatkan penggunaan *stopwatch* pada saat perangkat dioperasikan. Proses pengujian dilakukan dengan cara menjalankan sistem secara khusus pada perangkat terkendali dengan dilakukannya pengubahan nilai parameter secara manual menggunakan perangkat laptop pada laman <https://console.firebase.google.com/> sebanyak tujuh kali. Waktu eksekusi perintah dihitung sejak parameter pada *firebase realtime-database* diganti hingga kondisi relay pada perangkat sukses berubah. Apabila pada saat pengujian dilakukan terjadi hilang respon pada perangkat yang disebabkan oleh gangguan sinyal atau gangguan kelistrikan maka waktu akan terus dianggap berjalan hingga kondisi relay sukses berubah. Kondisi ini dibuat sehingga didapatkan data berapa lama waktu rata-rata yang dibutuhkan sistem untuk mengeksekusi satu perintah. Berdasarkan pengujian yang dilakukan, diperoleh hasil seperti pada tabel berikut

Tabel 4. Waktu respon perangkat terkendali

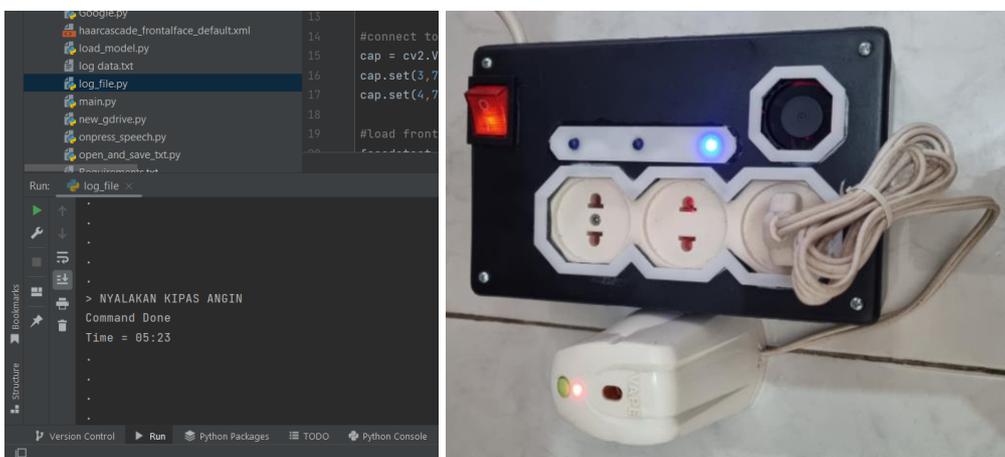
Percobaan ke	Waktu Eksekusi Perintah (detik)
1	4.75
2	8.99
3	3.57
4	7.10
5	1.21
6	2.71
7	6.87

Berdasarkan Tabel 4 diatas, diperoleh hasil bahwa waktu respon perangkat terkendali pada seluruh percobaan berada dibawah lima detik dengan waktu eksekusi perintah paling cepat adalah 1.21 detik dan waktu eksekusi perintah paling lambat adalah 8.99 detik. Pada seluruh pengujian, parameter yang diubah pada *real-time database* dapat secara langsung dikenali oleh perangkat dan perintah langsung dieksekusi. Selama pengujian berlangsung, tidak ada terjadi hilang respon pada perangkat yang disebabkan oleh gangguan sinyal atau gangguan kelistrikan dengan alat berada pada kondisi optimalnya.

Dikarenakan hasil pengujian respon perangkat terkendali menunjukkan waktu yang digunakan oleh sistem untuk mengubah parameter pada *real-time database* menjadi kondisi relay berkisar kurang dari sepuluh detik maka diperoleh hasil pengujian bahwa respon perangkat terkendali untuk mengeksekusi perintah tergolong layak pada durasi eksekusi berkisar satu sampai sembilan detik.

3.8 Pengujian Pengaruh Penggunaan *Casing* Terhadap Kinerja Alat

Pengujian ini dilakukan untuk mengetahui pengaruh dari penggunaan *casing* pada kinerja perangkat baik perangkat pengendali maupun perangkat terkendali. Pada pengujian ini, digunakan metode yang mirip seperti dua pengujian sebelumnya, yaitu pengujian waktu respon perangkat pengendali dan pengujian waktu respon perangkat terkendali hanya saja variabel yang digunakan lebih difokuskan terhadap pengaruh penggunaan *casing* pada alat. Kondisi yang digunakan untuk pengujian adalah seluruh perangkat berupa perangkat pengendali, perangkat terkendali, serta laptop untuk memonitor sistem diletakkan pada satu ruangan yang sama dengan *router* wifi dan suhu udara yang dijaga menggunakan sebuah kipas angin berukuran 18 inci. Kondisi ini dipilih untuk mengetahui bagaimana hasil pengujian alat jika ditempatkan pada kondisi yang optimal baik dari sisi ketersediaan jaringan internet maupun suhu ruangnya.



Gambar 19. Pengujian waktu respon kedua perangkat menggunakan *casing*

Gambar 19 diatas menunjukkan pengujian waktu respon kedua perangkat yang dilakukan sama seperti pengujian sebelumnya, hanya saja terdapat perbedaan pada kondisi *casing* perangkat. Proses pengujian dilakukan dengan cara menjalankan sistem pada perangkat pengendali dan perangkat terkendali secara terpisah dan bergantian. Pada perangkat pengendali, waktu eksekusi perintah dihitung sejak huruf pertama kata kunci diucapkan hingga parameter pada *firebase realtime-database* sukses terganti. Apabila ada kata yang gagal dikenali oleh sistem, maka kata kunci akan diucapkan berulang dengan waktu yang terus berjalan hingga parameter pada *firebase realtime-database* sukses terganti. Sedangkan pada perangkat terkendali, waktu eksekusi perintah dihitung sejak parameter pada *firebase realtime-database* diganti hingga kondisi relay pada perangkat sukses berubah. Apabila pada saat pengujian dilakukan terjadi hilang respon pada perangkat yang disebabkan oleh gangguan sinyal atau gangguan kelistrikan maka waktu akan terus dianggap berjalan hingga kondisi relay sukses berubah. Kondisi ini dibuat untuk mengetahui perbedaan waktu rata-rata yang dibutuhkan sistem untuk mengeksekusi perintah antara kondisi alat saat diberikan tambahan *casing* dengan kondisi alat saat tidak diberikan tambahan *casing*. Berdasarkan pengujian yang dilakukan, diperoleh hasil seperti pada tabel berikut:

Tabel 5. Waktu respon perangkat

Percobaan ke	Kondisi Alat	Waktu Eksekusi Perangkat Pengendali	Waktu Eksekusi Perangkat Terkendali
1	Menggunakan <i>casing</i>	4.92	1.67
2	Menggunakan <i>casing</i>	4.87	9.12
3	Menggunakan <i>casing</i>	5.23	3.22
4	Menggunakan <i>casing</i>	4.98	4.19
1	Tidak menggunakan <i>casing</i>	4.68	3.79
2	Tidak menggunakan <i>casing</i>	4.18	2.18
3	Tidak menggunakan <i>casing</i>	5.21	1.46
4	Tidak menggunakan <i>casing</i>	4.87	8.87

Berdasarkan Tabel 5 diatas, diperoleh hasil bahwa waktu respon dari perangkat, baik perangkat pengendali maupun perangkat terkendali tidak berbeda secara signifikan antara pengujian menggunakan *casing* dengan pengujian tidak menggunakan *casing*. Pada perangkat pengendali, waktu respon perangkat menunjukkan angka berkisar antara empat hingga lima detik baik pada kondisi *casing* terbuka maupun tertutup. Pada perangkat terkendali, waktu respon perangkat menunjukkan angka berkisar antara satu hingga sembilan detik.

Dikarenakan hasil pengujian pengaruh penggunaan *casing* terhadap kinerja alat menunjukkan perbedaan waktu eksekusi perintah yang tidak terlalu signifikan, dimana waktu respon perangkat pengendali berada di angka empat sampai lima detik dan waktu respon perangkat pengendali berada di angka satu sampai sembilan detik maka diperoleh hasil pengujian bahwa penggunaan *casing* tidak menyebabkan adanya penurunan performa yang signifikan dari seluruh perangkat.

4. PENUTUP

Dengan menggunakan perangkat pengendali berbasis Orange Pi 3B dan perangkat terkendali berbasis ESP32 sebagai perangkat utama sistem, pengguna dapat melakukan pengendalian perangkat elektronik rumah tangga menggunakan perintah suara tanpa perlu adanya kontak langsung terhadap perangkat elektronik rumah tangga tersebut secara efisien. Pada pengujian waktu respon perangkat pengendali dengan menggunakan *casing* diperoleh waktu respon tercepat pada 4.87 detik, sedangkan pada perangkat terkendali dengan menggunakan *casing* diperoleh waktu respon tercepat pada 1.67 detik. Hal ini menunjukkan bahwa sistem memiliki tingkat respon yang sangat baik dalam pengoperasiannya dalam mengendalikan perangkat elektronik rumah tangga.

Sebagai fitur proteksi agar sistem tidak diakses oleh orang asing secara bebas, penggunaan face recognition pada sistem ini memiliki hasil yang tergolong layak. Hal ini ditunjukkan pada pengujian tingkat akurasi dari *face recognition* yang berada pada nilai *Accuracy* sebesar 75%, *Precision* sebesar 67%, *Sensitivity* sebesar 100%, *Specificity* sebesar 50%. Persentase akurasi pada pengujian ini tergolong buruk jika diimplementasikan pada perangkat yang memerlukan tingkat keamanan yang tinggi dan akurat sehingga memerlukan pengkajian lebih lanjut dalam peningkatan akurasi sistem *face recognition*. Namun, dikarenakan perangkat stopkontak elektronik cerdas ini ditujukan untuk penggunaan di dalam rumah dengan asumsi jarang adanya orang asing di dalam rumah tersebut, persentase ini masih dapat ditolerir.

Penerapan *firebase real-time database* sebagai media komunikasi antarperangkat memungkinkan sistem untuk saling berkomunikasi bahkan dari jarak yang jauh sekalipun serta dapat ditempatkan dimana saja pada rumah selama lokasi tersebut memiliki jaringan internet dan perangkat dapat terhubung dengan jaringan internet yang tersedia pada lokasi tersebut. Penggunaan *firebase real-time database* juga membuat tampilan perangkat yang terpasang menjadi lebih bersih dikarenakan tidak diperlukannya kabel penghubung antar perangkat untuk berkomunikasi. Dengan adanya penerapan ini, sistem menjadi dapat menyinkronisasi data secara langsung dan responsif terhadap perubahan kondisi atau parameter yang dikendalikan.

Selain memanfaatkan *firebase real-time database* untuk komunikasi antarperangkat, integrasi dengan layanan *google drive* juga dapat menjadi solusi yang efisien dalam mengelola dataset untuk

sistem *face recognition*. Dengan memanfaatkan *speech recognition* sebagai antarmuka pengguna, pengguna dapat memberikan perintah suara untuk mengunggah atau mengunduh dataset langsung dari *google drive*. Penggunaan *google drive* sebagai penyimpanan dataset memberikan fleksibilitas dalam manajemen dan akses data. Melalui integrasi ini, pengguna dapat dengan mudah mengunggah dataset baru atau mengunduh dataset yang diperlukan hanya dengan menggunakan perintah suara yang dikenali oleh sistem.

Pengembangan selanjutnya dari sistem ini dapat mencakup peningkatan akurasi pengenalan wajah dan suara, serta pengembangan antarmuka pengguna yang lebih intuitif dan responsif. Integrasi dengan teknologi *Internet of Things* (IoT) juga dapat diperluas untuk mencakup lebih banyak perangkat rumah tangga dan menciptakan ekosistem rumah pintar yang lebih terintegrasi. Dengan demikian, sistem kontrol stopkontak elektronik cerdas ini merupakan langkah awal yang penting dalam menghadirkan teknologi rumah pintar yang lebih efisien, aman, dan dapat diandalkan untuk mendukung kehidupan modern di era digital saat ini. Diharapkan pengembangan teknologi ini dapat memberikan manfaat yang signifikan dalam mendukung gaya hidup yang lebih nyaman dan berkelanjutan bagi pengguna di seluruh dunia.

PERSANTUNAN

Penulis ingin menyampaikan ucapan terima kasih yang tulus kepada semua pihak yang telah memberikan bantuan, dukungan, dan inspirasi dalam penelitian ini. Terima kasih kepada dosen pembimbing, rekan sejawat, dan keluarga atas dorongan, panduan, dan dukungan moral yang diberikan selama proses penulisan skripsi ini. Selain itu, penulis juga mengucapkan terima kasih kepada partisipan yang telah bersedia berkontribusi dalam pengumpulan data serta semua pihak yang turut membantu secara langsung maupun tidak langsung dalam penyelesaian skripsi ini. Ucapan terima kasih juga disampaikan kepada institusi pendidikan yang telah memberikan fasilitas dan sumber daya yang mendukung penelitian ini. Semoga hasil dari penelitian ini dapat memberikan manfaat dan kontribusi yang positif bagi perkembangan ilmu pengetahuan dan aplikasinya di bidang terkait.

DAFTAR PUSTAKA

- Eliza, D. (2015). Aplikasi Bluetooth HC-06 Untuk Pengontrolan Pintu Dan Alat Elektronik Menggunakan Smartphone Android Pada Smart Home. Other thesis, Politeknik Negeri Sriwijaya.
- Goyal, Y. (2023). Python KeyboardInterrupt. In *Educba*, www.educba.com/python-keyboardinterrupt.
- Kiljo, E. (2020). Optocoupler Breakage in Frequency Converter Use. Other thesis, Aalto University.
- Maharani, T. (2023). Perkembangan Penggunaan Internet Of Things Untuk Masa Yang Akan Datang. Medan: Universitas Islam Negeri Sumatera Utara.
- Orange Pi 3B. (2024). In *WikiOrange Pi*. orangeipi.org/orangepiwiki/index.php/Orange_Pi_3B.

- Payara, G. R., & Tanone, R. (2018). Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android. *JuTISI*, 4(3) 397–406.
- Pratmanto, D., Fandhilah, F., & Saputra, S. (2019). Rancang Bangun Rumah Pintar Dengan Platform Home Assistant Berbasis Raspberry Pi 3. *Evolusi: Jurnal Sains dan Manajemen*, 81-85.
- Putri, R. D. (2020). Pengembangan Video Pembelajaran Openstreetmap Untuk Pembuatan Peta Digital Format Shapefile Menggunakan Global Mapper. Other thesis, Universitas Negeri Yogyakarta.
- Rahman, M. B. (2019). Sistem Kendali Peralatan Elektronik Rumah Tangga Berbasis Internet Of Things (IoT) Menggunakan Nodemcu. *Ubiquitous: Computers and its Applications Journal*, 2(2) 99-104.
- Theodora, A. (2023). Pendapatan Rendah, Kemiskinan Bertambah. In Kompas.id. <https://www.kompas.id/baca/ekonomi/2023/01/18/pendapatan-rendah-kemiskinan-bertambah>.
- Yuliani, W., & Banjarnahor, N. (2021). Metode Penelitian Pengembangan (RnD) Dalam Bimbingan Dan Konseling. *QUANTA: Jurnal Kajian Bimbingan Dan Konseling Dalam Pendidikan*, 5(3) 111-118.