

Sistem pengendali saklar berbasis Nodemcu ESP8266 dengan aplikasi MQTT dan Google Assistant

Bekti Maryuni Susanto*, Moh Khafil Albab, Agus Hariyanto, Denny Wijanarko
Department of Information Technology, Politeknik Negeri Jember, Indonesia

Article Info

Article history:

Received November 10, 2022
Accepted November 15, 2022
Published November 25, 2022

Keywords:

MQTT
Google Assistant
Nodemcu
Saklar otomatis

ABSTRACT

The Internet of Things (IoT) provides the ability to connect many things or devices over the internet. Message Queue Telemetry Transport (MQTT) is a Client Server published/subscribed message transport protocol. The protocol runs on top of TCP/IP, or on top of other network protocols that provide sequential, lossless bi-directional connections. The system developed in this study consists of three layers based on the perception layer, network layer and application layer. The perception layer consists of relays that are used to control the screen based on user input. The network layer consists of the MQTT server and Google assistant connected to the IFTTT service. This system was developed with the aim of making it easier for humans to monitor home electrical devices remotely. The test results show that this system can be used to control the switch quickly even though it really depends on the quality of the internet network.



Corresponding Author:

Bekti Maryuni Susanto,,
Department of Information Technology,
Politeknik Negeri Jember,
Jl. Mastrip Kotak Pos 164 Jember Jawa Timur Indonesia.
Email: *bekti@polije.ac.id

1. PENGANTAR

Internet of Things (IoT) menyediakan kemampuan untuk menghubungkan banyak hal atau perangkat melalui internet. Benda ini atau perangkat memiliki identitas unik. IoT menciptakan lingkungan yang cerdas dengan menghubungkan perangkat dengan internet dan melengkapinya dengan kemampuan untuk mengumpulkan dan bertukar data. Perangkat atau gadget ini biasanya terhubung dengan mikrokontroler, sensor, aktuator dan konektivitas internet. Gadget tersebut dapat mencakup barang-barang rumah tangga biasa seperti mesin cuci, lemari es, sistem suara, pembuat kopi, jam alarm, dll. Selain itu, aplikasi IoT di kota pintar mencakup pemantauan lalu lintas, pemantauan polusi udara dan air, pemantauan konsumsi energi listrik, dan lain-lain. IoT menyediakan platform ke objek yang berbeda di mana mereka dapat berkomunikasi satu sama lain sambil memberi mereka kemampuan untuk mengatur diri sendiri [1].

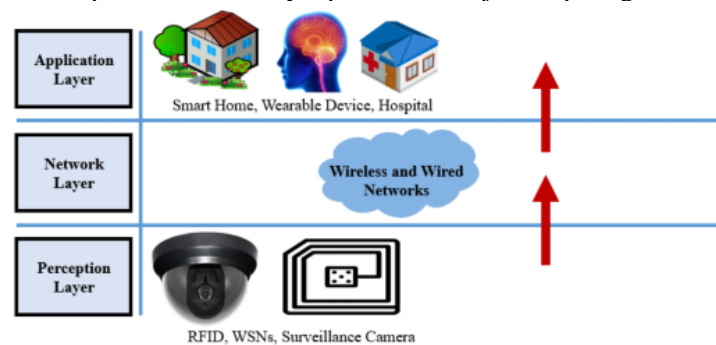
Message Queue Telemetry Transport (MQTT) adalah protokol transport pesan yang diterbitkan/berlangganan Server Klien. Ini ringan, terbuka, sederhana, dan dirancang agar mudah diimplementasikan. Karakteristik ini membuatnya ideal untuk digunakan dalam banyak situasi, termasuk lingkungan terbatas seperti untuk komunikasi dalam konteks Machine to Machine (M2M) dan Internet of Things (IoT) di mana jejak kode kecil diperlukan dan/atau bandwidth jaringan sangat mahal. Protokol berjalan di atas TCP/IP, atau di atas protokol jaringan lain yang menyediakan koneksi dua arah yang berurutan, tanpa kehilangan [2]. Penelitian yang dilakukan oleh [3] membuat sistem otomatisasi rumah yang aman menggunakan protokol MQTT dan juga ACL (akses kontrol daftar) untuk enkripsi data sensor. Penelitian ini mengimplementasikan SMQTT dan MQTT-SN yang menyediakan keamanan yang lebih baik dibandingkan dengan MQTT yang ada sebelumnya. Sistem dibangun menggunakan ESP8266 Nodemcu, sensor DHT11 yang mengukur suhu dan kelembapan serta sensor LDR yang mengukur intensitas cahaya. Diagram blok sistem yang dibangun ditunjukkan pada gambar. Server MQTT menggunakan layanan server publik yang diberikan oleh adafruit.com. Penelitian ini hanya menampilkan hasil pembacaan sensor DHT11 dan LDR dan tidak dilengkapi dengan aktuator.

Penelitian yang dilakukan oleh [4] mengimplementasikan protokol MQTT pada *smart home* berbasis web. Server MQTT diinstal secara lokal pada *single board computer* Raspberry Pi. Beberapa sensor dihubungkan dengan ESP8266 Nodemcu, yaitu sensor DHT11, Sensor PIR, sensor MQ5 dan sensor flame. Ambang batas setiap sensor ditentukan sehingga ketika nilai sensor melebihi ambang batas tersebut ESP8266 Nodemcu akan membunyikan *buzzer*. Penelitian ini hanya diterapkan dalam lingkup *local area network* yang berarti web monitoring hanya bisa diakses dalam jaringan lokal yang sama dengan server MQTT. Penelitian lain yang dilakukan oleh [5] mengimplementasikan MQTT pada *wearable device* yang dapat mengkomunikasikan datanya melalui serangkaian perangkat untuk dikirim ke sistem *cloud*. Sistem yang dikembangkan terdiri dari empat layer, yaitu sensing layer, gateway, MQTT broker dan dashboard. *Sensing layer* bertujuan untuk mengumpulkan data seperti misalnya informasi kesehatan penggunaan pada *wearable device*. Selain informasi kesehatan, sensing layer juga mengumpulkan informasi lingkungan dalam bentuk suhu dan kelembapan. Meskipun penelitian ini bertujuan untuk mengintegrasikan sistem IoT pada smart city namun lingkup sistem yang dikembangkan masih bersifat lokal sehingga dashboard web hanya bisa diakses pada jaringan lokal yang sama.

Penelitian ini mengimplementasikan protokol MQTT *public server* yang disediakan oleh adafruit.com pada pengendalian saklar rumah. Adafruit IO membuat *Internet of Things* dengan mudah dan sedikit pemrograman. Ini adalah platform IO open-source menyediakan dukungan server gratis[6]. Dashboard pengguna dibuat menggunakan aplikasi android. Selain dikendalikan menggunakan aplikasi android, sistem yang dikembangkan juga dapat dikendalikan menggunakan google assistant. Penerapan google assistant menggunakan layanan yang tersedia pada *If This Then That* (ifttt) yang menyediakan layanan untuk terhubung dengan berbagai macam aplikasi, perangkat dari berbagai macam pembang untuk membuatnya otomatis[7]. IFTTT dapat digunakan untuk menghubungkan dua layanan, Adafruit MQTT dan Google Assistant[8]. Penelitian ini bertujuan untuk membuat Perancangan Sistem Kontrol Pengendali Saklar Berbasis Nodemcu Esp8266 Dengan Aplikasi Mqtt Client Dan Google Assistant yang dapat mengontrol penggunaan perangkat elektronik yang telah terintegrasi dengan jaringan internet dan smartphone dengan aplikasi dan input perintah suara dari google assistant. Nodemcu ESP8266 dipilih karena bentuknya yang kecil yang dilengkapi dengan modul wifi dan kemampuannya yang mampu mengolah data baik analog maupun digital, sedangkan Google Assistant memberikan kemudahan dalam memberikan perintah melalui suara. Protokol MQTT dipilih karena kecepatannya yang lebih baik dibandingkan menggunakan http seperti yang dilakukan oleh[9]. Data dari Nodemcu ESP8266 selanjutnya dikirim ke layanan public MQTT yang disediakan oleh adafruit dan selanjutnya disinkronkan dengan google assistant melalui layanan IFTTT. Kebaruan dari penelitian ini adalah kemampuan dashboard aplikasi android yang dikembangkan dapat menambah board, ruangan dan perangkat tanpa harus membuka project code terlebih dahulu melainkan dengan memasukkan beberapa parameter yang dibutuhkan.

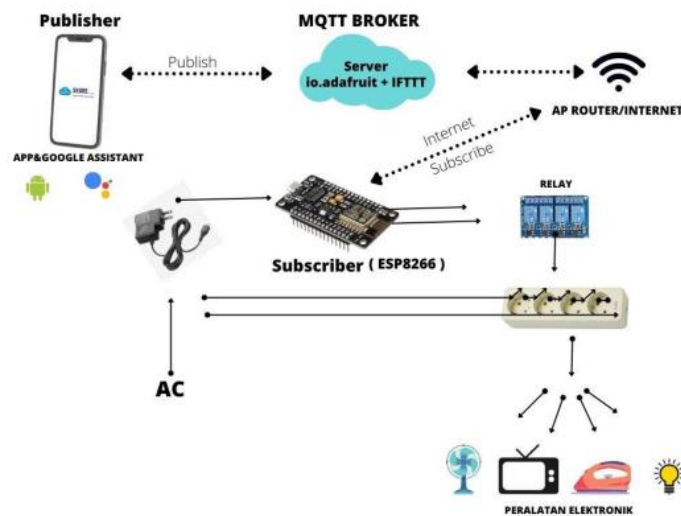
2. METODE PENELITIAN

Sistem yang dikembangkan pada penelitian ini terdiri dari tiga layer berdasarkan [10] yaitu *perception layer*, *network layer* dan *application layer*. Perception layer berfungsi seperti mata dan telinga manusia yang bertanggungjawab untuk mengidentifikasi benda dan mengumpulkan informasi dari benda tersebut. Pada penelitian ini perception layer terdiri dari aktuator berupa relay yang digunakan untuk mengendalikan layar berdasarkan input pengguna. Pada network layer terdiri dari server MQTT dan Google assistant yang terhubung pada layanan IFTTT. Pada application layer terdapat sebuah aplikasi dashbard yang digunakan untuk memberikan input ke dalam sistem oleh pengguna. Selain itu terdapat aplikasi google yang merupakan aplikasi bawaan pada smartphone android. Layer pada IoT ditunjukkan pada gambar 1.



Gambar 1. Layer pada IoT

Komponen-komponen yang digunakan untuk mengembangkan sistem ini yaitu, ESP8266 nodemcu, relay, stop kontak dan peralatan elektronik pada perception layer. Relay digunakan sebagai aktuator yang akan mengendalikan saklar on dan off. ESP8266 nodemcu terhubung pada jaringan wifi menggunakan library wifi manager untuk terhubung pada layanan server publik MQTT yang disediakan oleh adafruit dan IFTTT. Aplikasi android yang dikembangkan selanjutnya mengambil data dari server MQTT publik ini untuk ditampilkan pada dashboard. Aplikasi android dikembangkan menggunakan software Kodular, sebuah platform gratis untuk membuat aplikasi android dengan mudah. Selain itu aplikasi google dapat digunakan untuk memebrikan perintah suara memanfaatkan google assistant. Digram blok sistem yang dikembangkan ditunjukkan pada gambar 2. Alat ini dibuat untuk mempermudah pekerjaan yaitu mengontrol nyala peralatan elektronik yang tertancap di stop kontak dengan media aplikasi di smartphone dan perintah berupa suara. Relay dapat memutus dan menyambungkan arus listrik sesuai program yang tertanam di ESP8266 yang berfungsi sebagai Subscriber terhubung dengan internet, Subscriber sendiri disini berfungsi menerima data dari Broker data yang masuk sebelumnya di kirim oleh Publisher berupa data 0 yang di terjemahkan sebagai off (mati) dan data 1 sebagai on (menyala) di eksekusi dengan relay.untuk mematikan dan menyalakan perangkat yang terhubung ke stopkontak Io.Adafruit sebagai Server Broker yang berfungsi mendistribusikan data dari Publisher.



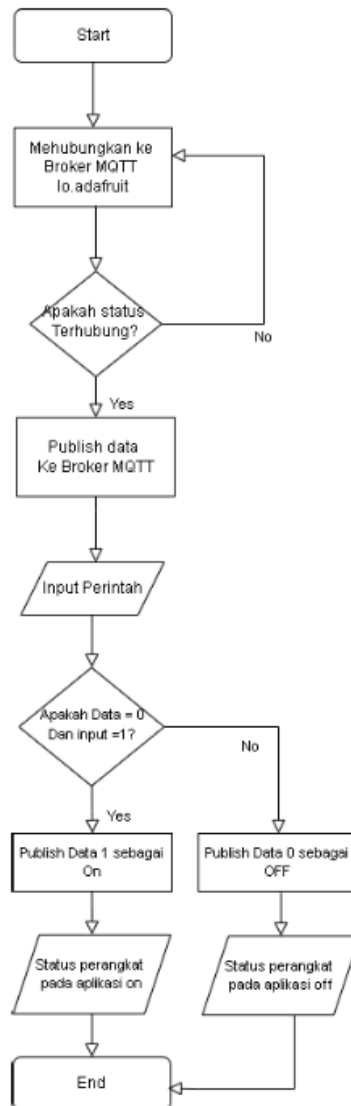
Gambar 2. Diagram blok sistem yang dikembangkan

Aplikasi Akses berfungsi sebagai Publisher data mengirimkan data 0 sebagai off (mati) dan 1 sebagai on (menyala) selanjutnya di kirimkan ke broker Mqtt Io.adafruit.com menggunakan Retained Message Sehingga Broker akan menyimpan salinan pesan yang di publikasi untuk di distribusikan kepada Subscriber (ESP8266) sehingga ketika terdapat Subscriber baru broker akan mengirimkan data salinan yang telah di pertahankan. Google assistant memiliki fungsi yang sama dengan aplikasi Akses yaitu sebagai Publisher namun penggunaannya menggunakan perintah atau kalimat yang telah di daftarkan sebelumnya di IFTT memungkinkan pengguna dapat mengontrol peralatan elektronik dengan smartphone. Proses pengiriman dan penerimaan data sangat bergantung kepada koneksi internet. Garis putus-putus pada gambar menunjukkan proses pertukaran data dari Publisher, Broker dan Subscriber melalui jaringan internet.

Alur dari cara kerja aplikasi Akses setelah mengisi data profile pada menu koneksi aplikasi akan terhubung ke broker mqtt io.adafruit.com, tombol on off yang di buat oleh pengguna akan melakukan proses publish terhadap broker akan di kirimkan data berupa 1 sebagai on dan 0 sebagai off, jika data yang di kirim berupa 1 maka status perangkat akan on, jika yang di publish merupakan 0 maka status perangkat akan off, Pesan yang dikirimkan terhadap broker mengaktifkan Retained Message sehingga pesan di duplikas yang nantinya akan di ambil datanya untuk memperbaharui kondisi tombol memastikan status tombol di aplikasi telah sesuai dengan di broker.

Tampilan dashboard aplikasi Akses, terdapat menu – menu berupa: Ucapan selamat pagi, siang, sore dan malam. Menu Scan QR Code sebagai fitur tambahan untuk scan key akun adafruit. Menu Profile sebagai profile koneksi, Home sebagai tombol pindah ke halaman home, Book sebagai buku panduan aplikasi dan in fo sebagai info aplikasi. Halaman Scan berisi scanner dan menampilkan hasil scan. Halaman Profile berisi profile mqtt yang telah di tambahkan dengan broker io.adafruit di halaman Home berisi nama ruangan dan peralatan – peralatan apa saja yang akan di kontrol nyala dan matinya serta terdapat menu log aktivitas untuk melihat status perangkat, Fitur Alarm untuk mengingatkan pengguna agar tidak lupa mematikan peralatan elektroniknya.

Pengujian alat dilakukan untuk menguji perangkat utama alat apakah sudah benar sesuai dengan keinginan ataupun ada kekurangan, pengujian ini juga menghindari kesalahan saat penerapan alat yang berupa on dan off menggunakan aplikasi yang telah di buat dan perintah suara dalam kehidupan yang nyata dan dikembangkan dengan komponen-komponen yang lebih tepat ataupun sesuai dengan lokasi dan lingkungan. Alat ini akan di uji secara bertahap, dari koneksi ESP8266 terhadap koneksi wireless pengujian aplikasi dan perintah suara pada smartphone android, rangkaian ESP8266, dan rangkaian relay. Maksud dari pengujian alat secara bertahap ini agar dapat mengetahui bagian-bagian yang tidak bekerja, kemudian dapat diperbaiki maupun di kembangkan secara satu-persatu bagian agar dapat diketahui apakah alat ini bekerja sesuai keinginan atau tidak. Flowchart sistem yang dibangun ditunjukkan pada gambar 3.



Gambar 3. Flowchart sistem yang dibangun

3. HASIL DAN ANALISIS

Perangkat keras pada penelitian ini menggunakan beberapa komponen yang memiliki tugas sesuai fungsinya. Komponen tersebut terdiri dari ESP8266, modul relay 4 channel, modul relay 2 chanel, Buzzer, Led, Module charger 9v. ESP8266 berfungsi sebagai mikrokontroler untuk pemrosesan data. Module relay 4 channel berfungsi penghubung dan pemutus arus untuk menghidupkan dan mematikan peralatan elektronik. Led dan Buzzer berfungsi sebagai notifikasi dan isyarat. Charger berfungsi sebagai sumber tegangan ESP8266, Relay, Buzzer dan Led.

Wifi manager digunakan untuk mengelola jaringan wifi pada ESP8266 Nodemcu, menghubungkan ke server MQTT dan memasukkan topik subscribe yang disediakan oleh adafruit tanpa harus membuka project.

Selanjutnya ESP8266 akan memanggil subscribe MQTT dan menampilkannya pada dashboard aplikasi android berupa kondisi saklar hidup atau mati. Kondisi mati dinyatakan dengan nilai 0 dan kondisi nyala dinyatakan dengan nilai 1. Selanjutnya relay akan menerima input dari pengguna melalui dashboard aplikasi android. Nodemcu dihubungkan ke relay 4 channel 1, 4buah led hijau, buzzer, input tegangan. NodeMcu esp8266 berfungsi sebagai mikrokontroler untuk memproses data yang masuk melalui broker mqtt dan menjalankan program untuk menghidupkan dan mematikan relay beserta led indikator. Relay berfungsi sebagai module yang akan memutus dan menyambungkan tegangan kepada beberapa perangkat elektronik yang terhubung, Led akan mengikuti alur program di mana ketika relay menyala maka led akan menyala pula begitu sebaliknya sesuai urutan nomor relay. Buzzer di sini berfungsi memberi indikasi bahwasanya alat telah terhubung ke broker mqtt dan jika tidak maka akan berbunyi setiap 5 detik sekali.

```

Output Serial Monitor x
Message (Ctrl+Enter to send message to NodeMCU 1.0 (ESP-12E Module)

*wm:[2] Connection result: WL_CONNECTED
*wm:[1] AutoConnect: SUCCESS
*wm:[1] STA IP Address: 192.168.0.106

WiFi connected To:
Admin
IP address:
192.168.0.106
Ready!
=====
Akses.
Attempting MQTT connection...io.adafruit.com
connected
    
```

Gambar 4. Output serial monitor saat berhasil terhubung ke wifi dan server MQTT menggunakan wifi manager

Gambar 4 menunjukkan output serial monitor saat ESP8266 Nodemcu terhubung ke wifi dan server MQTT berdasarkan topik subscribe yang diinput pada wifi manager. Keberhasilan terhubung ke server MQTT ditandai dengan status connected pada Attempting MQTT connection. Untuk menghubungkan perangkat control Akses ke broker io.adafruit.com nyalakan terlebih dahulu perangkat Akses anda di awal konfigurasi akan muncul nada beb satu kali pertanda perangkat sedang booting. Jika belum mengkonfigurasi akan ada SSID untuk dihubungkan, pilih SSID yang diinginkan "silahkan hubungkan dengan kata sandi: " password " klik login page akan muncul wifi manager seperti gambar 5, untuk memasukkan koneksi wifi dan menghubungkan alat dengan internet dan Broker yang telah di buat sebelumnya. Parameter yang digunakan untuk terhubung dengan server MQTT yaitu, mqtt server: io . adafruit .com //alamat broker, mqtt port: 1883, mqtt password dan mqtt keyword: username/feeds/relay1.

Gambar 5. Login page untuk terhubung dengan wifi dan server MQTT

Sistem kendali pada alat ini adalah melakukan proses subscrib terhadap broker layanan mqtt server di platform io.adafruit.com proses pertukaran data berlangsung dengan konektivitas internet aplikasi Akses sendiri akan melakukan publish terhadap Broker Mqtt dimana pesan berupa 1 dan 0 dengan Retain Message untuk mengambil data terupdate di broker, Aplikasi android ini memudahkan pengguna dalam mengontrol peralatan di rumah tersebut . Gambar 6 merupakan tampilan Dashboard aplikasi android yang dibuat untuk menampilkan status saklar dalam keadaan on atau off.



Gambar 6. Tampilan dashboard aplikasi android yang dibuat

Pada halaman dashboard pengguna dapat melihat tombol-tombol yang nantinya akan membuka screen sesuai nama tombol tersebut, tampilan waktu selamat pagi, siang, sore, malam . Terdapat tombol scan perangkat yang dapat di gunakan untuk scan io key dari akun adafruit atau qr code yang dapat dibuat sendiri. Pengguna selanjutnya di arahkan menuju halaman profile untuk menambah data profile baru, di menu Dashboard klik profile selanjutnya di halaman profile akan muncul tombol tambah di sebelah kanan kemudian klik popup form penambahan profile baru akan muncul seperti gambar 7. Setelah diperoleh informasi profile sebagai berikut : mqtt address : io.adafruit.com/alamat broker, Username : Nama username saat mendaftar, Mqtt port : 1883, Adafruit IO Key : “ bisa di lihat di menu my key di io.adafruit.com ” kemudian Save untuk menyimpan akan muncul profile baru. Untuk mengedit dan menghapus pada tampilan profile tersedia tombol edit profile dan hapus profile.

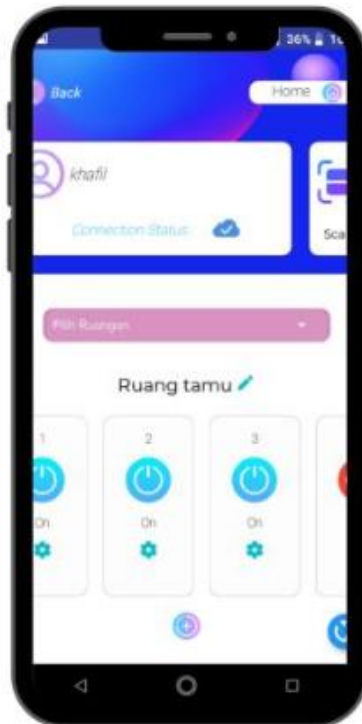
Parameter yang diinput pada aplikasi android sama dengan parameter yang diinputkan ke ESP8266 Nodemcu pada menu wifi manager. Langkah ini digunakan agar aplikasi android dapat menerima data yang dikirimkan oleh Nodemcu ke broker MQTT yang disediakan oleh adafruit. Broker MQTT adafruit bersifat publik yang dapat diakses dimana saja dengan koneksi internet. Hal ini berbeda dengan apabila kita menggunakan server MQTT sendiri yang bersifat lokal yang hanya bisa diakses pada jaringan lokal yang sama seperti yang dilakukan oleh [4]. Server MQTT adafruit akan menerima data dari ESP8266 Nodemcu, dalam hal ini Nodemcu berperan sebagai publisher dan server MQTT sebagai subscriber. Data akan disimpan pada server publik MQTT dan selanjutnya akan ditampilkan pada aplikasi android dengan memasukkan beberapa parameter agar data pada server MQTT dapat tampil pada aplikasi android. Salah satu parameter yang paling penting adalah adafruit IO key yang bersifat unik untuk setiap pengguna. Selain tentunya username, password dan mqtt port. Untuk implementasi protokol MQTT berbasis Publish-Subscribe, Adafruit IO mendukung

library penting di Arduino IDE SDK[11]. MQTT sendiri menggunakan protokol TCP/IP yang ringan sehingga bandwidth yang digunakan tetap kecil.



Gambar 7. Menambahlan profile baru pengguna

Sistem yang dikembangkan dapat juga idgunakan untuk menambah ruangan dan perangkat, dimana pada setiap ruangan terdapat beberapa perangkat. Setiap perangkat pada aplikasi android diidentifikasi nama feeds relay berupa nomor relay yang ada pada akun io.adafruit.com. Ketika penambahan ruangan dan perangkat berhasil akan tampil tombol on/off untuk mengendalikan perangkat tersebut. Sistem yang dikembangkan dapat di kontrol melalui perintah suara smartphone yaitu melalui google assistant. Namun, sebelum mengontrol perangkat terlebih dahulu menyiapkan akun ifttt di website ifttt.com. Setelah membuat akun pada ifttt.com selanjutnya ke menu My Applets pilih create selanjutnya pilih if This Choose a service google assistant kemudian pilih say a simple pharese, dibagian ini nantinya dapat ditampilkan apa yang di ucapkan. Gambar 8 menunjukkan tampilan aplikasi android yang sudah berhasil menambahkan ruangan dan perangkat.



Gambar 8. Tampilan aplikasi android saat berhasil menambahkan ruangan dan perangkat

Tabel 1. Hasil pengujian relay menggunakan aplikasi android

No.	Sensor	Instruksi Aplikasi	Kondisi Led pada relay	Jeda Respon Alat (s)	Keterangan
1.	Relay1	<i>Switch Button On</i>	Nyala	1,403 s	Berhasil
		<i>Switch Button Off</i>	Mati	1,480 s	Berhasil
2.	Relay2	<i>Switch Button On</i>	Nyala	1,188 s	Berhasil
		<i>Switch Button Off</i>	Mati	1,388 s	Berhasil
3.	Relay3	<i>Switch Button On</i>	Nyala	1.123 s	Berhasil
		<i>Switch Button Off</i>	Mati	0.848 s	Berhasil
4.	Relay4	<i>Switch Button On</i>	Nyala	0,847 s	Berhasil
		<i>Switch Button Off</i>	Mati	0.625 s	Berhasil

Tabel 2. Hasil pengujian relay menggunakan google assistant

No.	Sensor	Instruksi Google Assistant	Kondisi Led pada relay	Jeda Respon Alat (s)	Keterangan
1.	Relay1	<i>Instruksi On</i>	Nyala	2,703 s	Berhasil
		<i>Instruksi Off</i>	Mati	2,489 s	Berhasil
2.	Relay2	<i>Instruksi On</i>	Nyala	2,888 s	Berhasil
		<i>Instruksi Off</i>	Mati	1,987 s	Berhasil
3.	Relay3	<i>Instruksi On</i>	Nyala	2.123 s	Berhasil
		<i>Instruksi Off</i>	Mati	2.848 s	Berhasil
4.	Relay4	<i>Instruksi On</i>	Nyala	3,307 s	Berhasil
		<i>Instruksi Off</i>	Mati	2.928 s	Berhasil

Antarmuka sistem yang dikembangkan menggunakan aplikasi berbasis android yang dikembangkan menggunakan software kodular. Sistem ini dijalankan dengan memanfaatkan jaringan internet tanpa harus berada pada jaringan lokal yang sama seperti yang dilakukan oleh [12]. Antarmuka aplikasi menggunakan aplikasi android yang dikembangkan sendiri tidak menggunakan blynk seperti yang dilakukan oleh [13], [14] dan [15]. Kelebihan dari aplikasi ini adalah mampu memberikan perintah on ataupun off pada perangkat secara jarak jauh melalui koneksi internet. Sedangkan kekurangan dari aplikasi ini adalah ketergantungannya pada internet sehingga jika jaringan internet mengalami gangguan maka aplikasi ini tidak bisa bekerja sebagaimana mestinya.

4. KESIMPULAN

Sistem pengendali saklar berbasis MQTT dan google assistant dikembangkan dengan tujuan untuk mempermudah manusia dalam melakukan monitoring perangkat listrik rumah secara jarak jauh. Selain itu juga untuk mengurangi resiko bahaya yang diakibatkan oleh adanya arus listrik ketika kita menyentuh perangkat listrik secara langsung. Hasil pengujian menunjukkan bahwa server MQTT dan Google Assistant dapat

digunakan untuk mengendalikan saklar perangkat listrik rumah dengan cepat meskipun sangat tergantung pada kualitas jaringan internet. Sistem ini dapat dikembangkan lagi dengan menambahkan analisis konsumsi daya listrik dan timer pada setiap perangkat.

DAFTAR PUSTAKA

- [1] M. Kashyap, V. Sharma, and N. Gupta, "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things," *Procedia Computer Science*, vol. 132, no. Iccids, pp. 1611–1618, 2018, doi: 10.1016/j.procs.2018.05.126.
- [2] OASIS, "MQTT Version 3.1.1," *OASIS Standard*, no. October, p. 81, 2014.
- [3] R. K. Kodali and K. S. Mahesh, "A low cost implementation of MQTT using ESP8266," *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics, IC3I 2016*, no. December 2016, pp. 404–408, 2016, doi: 10.1109/IC3I.2016.7917998.
- [4] B. M. Susanto, E. S. J. Atmadji, and W. L. Brenkman, "Implementasi Mqtt Protocol Pada Smart Home Security Berbasis Web," *Jurnal Informatika Polinema*, vol. 4, no. 3, p. 201, 2018, doi: 10.33795/jip.v4i3.207.
- [5] C. D'ortona, D. Tarchi, and C. Raffaelli, "Open-Source MQTT-Based End-to-End IoT System for Smart City Scenarios," *Future Internet*, vol. 14, no. 2, 2022, doi: 10.3390/fi14020057.
- [6] V. Vadi, A. Kotadiya, R. Solanki, and M. M. Khambalkar, "Low-Cost Asset Tracking and Inventory Management System (ATIM)," *Journal of Physics: Conference Series*, vol. 2007, no. 1, 2021, doi: 10.1088/1742-6596/2007/1/012072.
- [7] N. Goud and A. Sivakami, "Spectate home appliances by internet of things using mqtt and ifttt through google assistant," *International Journal of Scientific and Technology Research*, vol. 8, no. 10, pp. 1852–1857, 2019.
- [8] M. D. Agarwal and S. K. Srivastava, "Integration of ' Google Assistant ' and ' AdafruitIO ' via ' IFTTT ' to Implement an IoT based Wireless Multi-Sensor Network Surveillance and Multi-Appliance Integration of ' Google Assist ant ' and ' AdafruitIO ' via ' IFTTT ' to Implement an IoT based W," *International Research Journal of Engineering and Technology (IRJET)*, vol. 07, no. September, pp. 2226–2232, 2020.
- [9] C. F. Permatasari and H. Dhika, "JISA (Jurnal Informatika dan Sains) Optimasi Jalur Transfer Data dari HTTP menjadi MQTT pada IoT menggunakan Cloud Services," vol. 01, no. 02, 2018.
- [10] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors (Switzerland)*, vol. 18, no. 9, 2018, doi: 10.3390/s18092796.
- [11] D. Mishra, P. Yadav, V. Yadav, R. Srivastava, K. K. Agrawal, and R. S. Yadav, "Adafruit IO Based Smart Irrigation System using MQTT Protocol for Urban Farming," *Journal of Computer Science*, vol. 18, no. 5, pp. 374–381, 2022, doi: 10.3844/jcssp.2022.374.381.
- [12] A. D. Hidayat, B. Sudibya, and C. B. Waluyo, "Pendeteksi Tingkat Kebisingan berbasis Internet of Things sebagai Media Kontrol Kenyamanan Ruang Perustakaan," *AVITEC*, vol. 1, no. 1, Aug. 2019, doi: 10.28989/avitec.v1i1.497.
- [13] R. Sahtyawan and A. I. Wicaksono, "Application for Control of Distance Lights Using Microcontroller Nodemcu Esp 8266 Based on Internet of Things (IoT)," *Compiler*, vol. 9, no. 1, May 2020, doi: 10.28989/compiler.v9i1.627.
- [14] R. Sahtyawan, A. B. Saputra, and S. Arief, "NodeMcu Microcontroller Based Disinfectant Fluid Monitoring System Using Water Level Control Sensor and Ultrasonic Sensor," *Angkasa: Jurnal Ilmiah Bidang Teknologi*, vol. 12, no. 2, Oct. 2020, doi: 10.28989/angkasa.v12i2.770.
- [15] N. Kusumaningrum and R. Ananda, "STERILIZER CHAMBER DESIGN WITH TELEGRAM-BASED INTERNET OF THINGS (IoT) APPLICATIONS," *Compiler*, vol. 10, no. 2, p. 89, Nov. 2021, doi: 10.28989/compiler.v10i2.1111.

