

USDB22Naskah

by Augustinus Bayu Primawan

Submission date: 21-Nov-2023 08:03AM (UTC+0700)

Submission ID: 2226476686

File name: Yoel_USDB_Artikel_2022.pdf (502.44K)

Word count: 4714

Character count: 25030

PROTOTYPE SMART HOME MENGGUNAKAN VOICE CONTROL DAN BLYNK

Yoel Aldo Moga¹, Augustinus Bayu Primawan²

¹*Department of Electrical Engineering University of Sanata Dharma
Paingan, Maguwoharjo, Depok, Sleman, Yogyakarta
Telp. (0274) 883037, 883968 Fax. (0274) 886529*

²*Department of Electrical Engineering University of Sanata Dharma
Paingan, Maguwoharjo, Depok, Sleman, Yogyakarta, Indonesia
Telp. (0274) 883037, 883968 Fax. (0274) 886529*

**Email : yoelmoga918@gmail.com*

Abstrak

Kemajuan teknologi di masa sekarang ini berkembang dengan sangat cepat, sehingga banyak perusahaan-perusahaan besar berlomba-lomba untuk menciptakan teknologi yang dapat memudahkan aktifitas manusia (Nur Afiyat, 2021). Salah satunya adalah perangkat-perangkat elektronik yang hemat energi seperti Lampu hemat energi, Air Conditioner (AC) hemat energi, Television (TV) hemat energi, dan masih banyak lagi. Namun pada penerapannya, kerap terdapat kasus di mana pengguna sering kelupaan mematikan perangkat-perangkat tersebut ketika sedang bepergian. Hal ini tentu saja akan tetap menjadi pemborosan energi walaupun sudah memakai perangkat-perangkat tersebut. Perancangan yang dibuat adalah sistem prototipe smart home menggunakan voice control dan Blynk. Perancangan ini dibuat untuk memudahkan aktivitas manusia dalam melakukan pengendalian dan pemantauan alat elektronik di rumah. Pengendalian dan pemantauan menggunakan NodeMCU esp32 dengan Blynk sebagai server dan databasenya. Hasil pengujian sistem prototipe smart home menggunakan voice control dan Blynk menghasilkan nilai delay di setiap pengendaliannya yang berada di tingkat kurang memuaskan. Nilai rata-rata delay yang dihasilkan berada di kisaran 1,4 detik hingga 3,8 detik, hasil ini berada pada indeks sangat buruk menurut standarisasi delay TIPHON. Hasil pengujian sistem pembacaan sensor tegangan dapat mengukur dengan baik dan juga pada bagian fungsional kalimat perintah suara dapat bekerja dengan baik.

Kata Kunci: *NodeMCU, Blynk, Voice Control, Internet of Things.*

SMART HOME PROTOTYPE USING VOICE CONTROL AND BLYNK

Yoel Aldo Moga¹, Augustinus Bayu Primawan²

¹*Department of Electrical Engineering University of Sanata Dharma
Paingan, Maguwoharjo, Depok, Sleman, Yogyakarta
Telp. (0274) 883037, 883968 Fax. (0274) 886529*

²*Department of Electrical Engineering University of Sanata Dharma
Paingan, Maguwoharjo, Depok, Sleman, Yogyakarta, Indonesia
Telp. (0274) 883037, 883968 Fax. (0274) 886529*

**Email : yoelmoga918@gmail.com*

Abstract

Advances in technology today are developing very quickly, so many large companies are competing to create technology that can facilitate human activities

(Nur Afiyat, 2021). One of them is energy-efficient electronic devices such as energy-saving lamps, energy-efficient Air Conditioners (AC), energy-efficient Television (TV), and many more. However, in its application, there are often cases where users often forget to turn off these devices while traveling. This of course will still be a waste of energy even though you have used these devices. The design made is a smart home prototype system using voice control and Blynk. This design is made to facilitate human activities in controlling and monitoring electronic devices at home. Control and monitoring using NodeMCU esp32 with Blynk as the server and database. The results of testing the smart home prototype system using voice control and Blynk resulted in a delay value in each control which was at an unsatisfactory level. The average value of the resulting delay is in the range of 1.4 seconds to 3.8 seconds, this result is in a very bad index according to the TIPHON delay standard. The test results of the voltage sensor reading system can measure well and also the functional part of the voice command sentence can work well.

Keywords: NodeMCU, Blynk, Voice Control, Internet of Things.

Pendahuluan

Kemajuan teknologi pada masa sekarang ini berkembang dengan sangat cepat, sehingga banyak perusahaan-perusahaan besar berlomba-lomba untuk menciptakan teknologi yang dapat memudahkan aktivitas manusia (Nur Afiyat, 2021). Salah satunya adalah perangkat-perangkat elektronik yang hemat energi seperti Lampu hemat energi, *Air Conditioner* (AC) hemat energi, *Television* (TV) hemat energi, dan masih banyak lagi. Namun pada penerapannya, kerap terdapat kasus di mana pengguna sering kelupaan mematikan perangkat-perangkat tersebut ketika sedang bepergian. Hal ini tentu saja akan tetap menjadi pemborosan energi walaupun sudah memakai perangkat-perangkat tersebut.

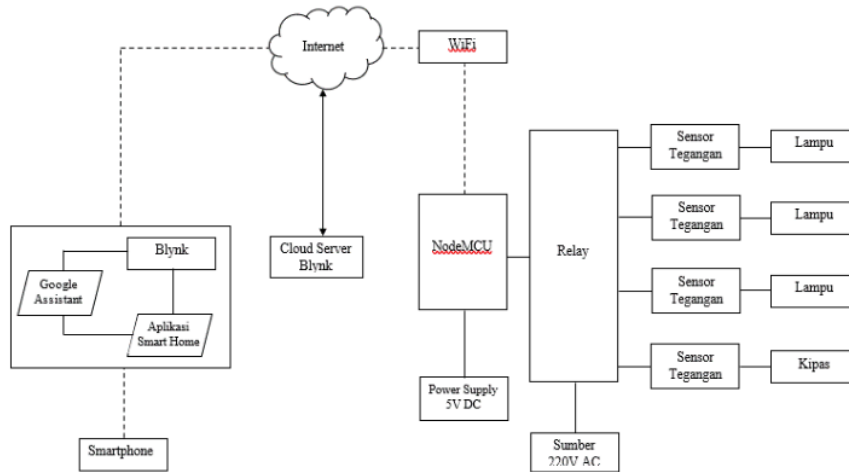
Ada beberapa solusi yang dapat diambil dari masalah tersebut, salah satunya adalah teknologi pengendali jarak jauh yang cukup efektif untuk membantu pengguna dalam mengendalikan perangkat-perangkat elektronik bila lupa dimatikan saat bepergian. Dalam hal ini sudah banyak penelitian yang mengangkat topik tentang pengontrolan jarak jauh, seperti pada penelitian (Reski Damayanti, 2020) Penelitian ini bertujuan untuk melakukan pengontrolan alat-alat elektronik dari jarak jauh dengan NodeMCU sebagai mikrokontrolernya dan menggunakan rancangan aplikasi dari Mit App Inventor sebagai sarana pengendaliannya. Selanjutnya ada penelitian (Florantina Cherli I. L. Herin, 2019) Penelitian ini bertujuan untuk melakukan pengembangan pada aspek pengontrolan perangkat elektronik jarak jauh dengan menggunakan perintah suara sebagai masukannya. Selanjutnya ada penelitian (Abdul Halim Mukti Nasution, 2019) Pada penelitian ini bertujuan untuk mengendalikan perangkat elektronik dari jarak jauh menggunakan aplikasi Blynk pada perangkat Android yang berbasis IoT.

Dari beberapa penelitian yang sudah dilakukan terdapat kelebihan dan kekurangan masing-masing. Salah satu kelebihannya adalah penggunaan Google Assistant dan Blynk yang sudah sangat familier di kalangan masyarakat luas. Penggunaan Google Assistant juga sangat memudahkan orang-orang ketika ingin melakukan pengontrolan, karena hampir di semua *Smartphone* Android sudah memiliki Google Assistant. Adanya Blynk dapat melengkapi kekurangan Google Assistant dengan menyediakan pilihan untuk melakukan pengontrolan perangkat

elektronik secara bersamaan, yang mana hal ini tidak bisa dilakukan bila menggunakan Google Assistant karena harus memasukkan datanya satu per satu. Dari segi kekurangannya, penelitian-penelitian tersebut hanya memfokuskan pada pengontrolan, sehingga bila ada pengguna yang kelupaan mematikan perangkat elektronik di rumahnya saat bepergian tidak akan sadar bila masih ada perangkat elektronik di rumahnya yang masih menyala. Pada penelitian ini penulis ingin mengembangkan penelitian-penelitian sebelumnya dengan menambahkan fitur pemantau dalam skema prototipe yang sudah diteliti sebelumnya. Sensor Tegangan zmp101b ditambahkan sebagai pemantau kondisi *on/off* pada beban.

Metode

Pemodelan Sistem

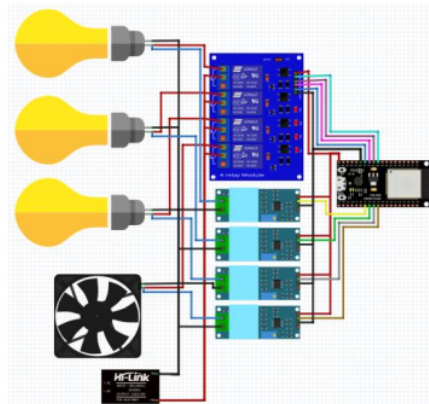


Gambar 1 Diagram Blok Sistem

Pada Gambar 1 menunjukkan diagram blok sistem yang mana data akan mulai diolah pada *smartphone* pengguna yang sudah terhubung dengan Blynk, aplikasi Smart Home, dan Google Assistant. Masukan berupa perintah akan dikirim ke NodeMCU ESP32 lewat jaringan internet. Pada tahap ini, NodeMCU ESP32 akan mengeksekusi perintah tersebut dengan menggunakan relay yang sudah terhubung dengan beban. Sensor tegangan berfungsi untuk memantau kondisi *on/off* pada beban.

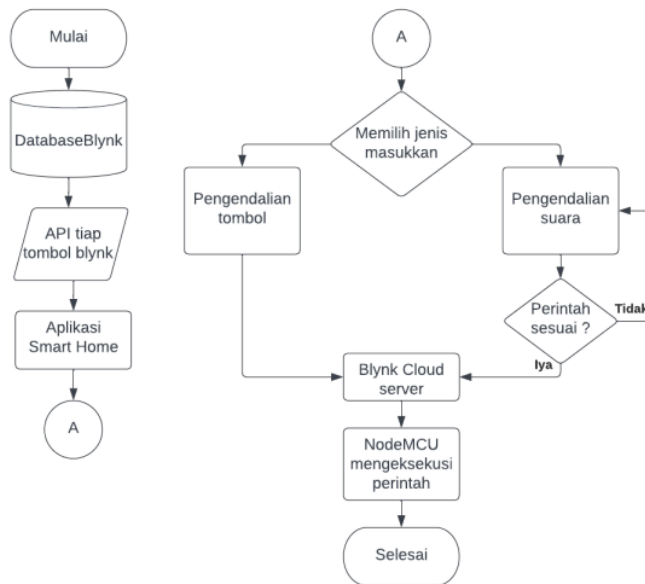
Rancangan wiring NodeMCU

NodeMCU esp32 dihubungkan ke komponen-komponen pengendali dan pemantau. Pada bagian pengendalian, NodeMCU esp32 dihubungkan dengan relay 4-*module*, dimana relay kemudian dihubungkan dengan output (lampu). Pada sistem pemantauan, NodeMCU esp32 dihubungkan dengan sensor tegangan zmp101b, dimana sensor kemudian dihubungkan dengan output (lampu).



Gambar 3 Rangkain wiring NodeMCU

Rancangan perangkat lunak



Gambar 4 Diagram alir sistem pengendalian

Pada perancangan sistem prototipe ini Blynk digunakan sebagai database sistem, di mana program utama akan dibuat pada Blynk. Sedangkan untuk aplikasi yang didesain pada Kodular hanya akan mengambil API dari masing-masing tombol yang sudah dibuat terlebih dahulu di Blynk.

Sedangkan untuk sistem pemantau masih tetap menggunakan aplikasi Blynk dengan desain tampilan yang lebih difokuskan untuk sistem pemantauan tanpa adanya tombol atau fitur pengendalian lagi di dalamnya. Untuk lebih jelasnya lagi dapat dilihat pada gambar 5.



Gambar 5 Diagram alir sistem pemantau

Sistem pengendalian tombol maupun suara yang dibuat di dalam aplikasi Smart Home, masing-masing memiliki cara kerja yang hampir sama, yaitu mengambil data dari setiap tombol di database Blynk yang sudah di program sebelumnya. Data tersebut akan dikirimkan melalui API tiap tombol Blynk menuju masing-masing sistem pengendalian. Pada sistem pengendalian tombol, data dari Blynk dimasukkan ke delapan (8) tombol yang sudah didesain menggunakan Kodular untuk menyalakan dan mematikan perangkat. Pada sistem pengendalian suara, data dari Blynk dimasukkan ke database aplikasi berupa kalimat-kalimat perintah yang akan aktif bila dipanggil melalui fitur *speech recognition*. Untuk lebih jelasnya, dapat dilihat pada gambar 4 dan gambar 5.

Metode Analisis Data

Pada tahap analisis sistem ini dilakukan uji coba dengan membandingkan waktu yang dibutuhkan sistem untuk mengirim perintah dan menerima perintah menggunakan *database* Blynk dan program pada Arduino IDE. Pengujian dilakukan dengan memberikan perintah terhadap masing-masing perangkat elektronik yang ada pada prototipe sistem ini. Pengujian penggunaan pengendalian tombol maupun suara dilakukan pada masing-masing perangkat elektronik sebanyak sepuluh (10) kali percobaan pada jarak 3 meter, 9 meter, dan 15 meter.

Pada bagian Google Assistant dilakukan juga pengujian terhadap kalimat perintah yang sudah di atur sebelumnya dengan mencoba masing-masing kalimat perintah.

Pada bagian pemantau dilakukan pengujian dengan cara yang sama seperti pada pengujian pengendalian. Data yang diambil adalah *delay* dari pembacaan hingga pengiriman data *sensor*. Pengujian dilakukan sebanyak sepuluh (10) kali pada jarak 3 meter, 9 meter, dan 15 meter. Setelah itu semua data hasil pengujian pada masing-masing perangkat tersebut akan di cara nilai rata-rata nya. Data rata-rata nilai *delay* yang didapat akan di bandingkan dengan standarisasi nilai *delay* versi TIPHON untuk mengetahui baik-buruknya *delay* yang dihasilkan prototipe sistem.

Hasil dan Pembahasan

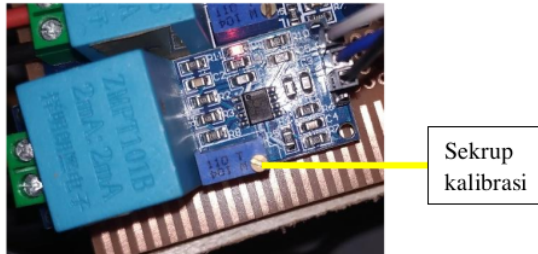
Pengujian Sensor Tegangan ZMPT101B

Pengujian terhadap sensor disini bertujuan untuk menguji ketepatan pembacaan sensor. Sensor yang digunakan adalah sensor tegangan *zmpt101b*. Kalibrasi dilakukan dengan dua (2) cara, yaitu dengan membuat program di dalam Arduino IDE dan mengatur sekrup yang ada pada sensor.

```
emon1.voltage(34, 170.26, 1.7);  
emon2.voltage(35, 180.26, 1.7);  
emon3.voltage(32, 180.26, 1.7);  
emon4.voltage(33, 180.26, 1.7);
```

Gambar 6 Program kalibrasi sensor

Pada gambar 6 merupakan program untuk kalibrasi sensor agar pengukuran sesuai dengan yang ditampilkan pada multimeter. Program untuk membaca hasil sensor menggunakan library dari *emon.lib*. Angka pertama yang berada di dalam kurung adalah input pin yang digunakan, angka yang kedua adalah nilai kalibrasi dari library *emon.lib* yang dapat disesuaikan penggunaannya dan angka yang ketiga adalah *phase shift*.



Gambar 7 Sekrup kalibrasi manual pada sensor

Pada gambar 7 merupakan sekrup yang digunakan untuk menyesuaikan hasil pembacaan data sensor secara manual. Cara penggunaannya adalah dengan memutar sekrup searah jarum jam untuk menaikkan hasil data pembacaan dan ke arah sebaliknya untuk menurunkan hasil data pembacaan.

Kalibrasi sensor dilakukan dengan membandingkan nilai yang keluar dari output program di serial monitor Arduino IDE dan hasil yang ditampilkan pada multimeter. Pengujian dilakukan sebanyak (3) pada masing-masing sensor dan mendapatkan hasil seperti pada tabel 1.

Pada pengujian ini dilakukan perbandingan nilai yang didapat dari pembacaan sensor tegangan dengan nilai yang didapat dari pembacaan multimeter menggunakan rumus persentase galat. Berikut merupakan rumus dan contoh yang digunakan untuk mendapatkan nilai galat pada pengujian ini.

$$\frac{|nilai\ perkiraan - nilai\ tepat|}{|nilai\ tepat|} \times 100 = error\% \quad (i)$$

$$\frac{|223 - 223,08|}{|223,08|} \times 100 = 0,03\% \quad (ii)$$

Tabel 1 Nilai rata-rata dan galat karakteristik sensor

No.	Sensor	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3	Rata-rata	Nilai Multimeter	Galat %
1.	Sensor kamar tidur utama	226,36 V	222,87 V	220,03 V	223,08 V	223 V	0,03 %
2.	Sensor kamar tidur tamu	219,45 V	222,14 V	217,06 V	219,55 V	223 V	1,57 %
3.	Sensor ruang tamu	224,34 V	222,88 V	222,55 V	223,25 V	224 V	0,33 %
4.	Sensor kipas angin	220,56 V	222,41 V	225,05 V	222,67 V	223 V	0,14 %

Berdasarkan hasil pengujian karakteristik sensor, nilai yang ditampilkan pada serial monitor dan nilai yang ditampilkan pada multimeter tidak terlalu jauh berbeda, dengan persentase galat yang berada diantara 0,03% hingga 1,57%.

Pengujian Fungsional Kalimat Perintah

Pengujian dilakukan dengan mengucapkan kalimat-kalimat perintah yang sudah diatur sebelumnya pada *block* program Kodular. Pengujian ini bertujuan untuk mengetahui fungsi dari masing-masing kalimat perintah apakah dapat berjalan dengan baik atau tidak. Terdapat dua (2) variasi kalimat perintah yang dapat digunakan untuk menyalakan atau mematikan perangkat.

Pengujian kalimat perintah dilakukan sebanyak sepuluh (10) kali pada masing-masing variasi perintah. Berdasarkan pengujian tersebut didapat hasil sangat baik dan tidak terdapat kendala. Berikut merupakan data hasil pengujian kalimat-kalimat perintah suara :

Tabel 2 Hasil pengujian fungsional kalimat-kalimat perintah suara

Percobaan	Kamar tidur utama		Kamar tidur tamu		Ruang tamu		Kipas Angin	
	Var. 1	Var.2	Var.1	Var.2	Var.1	Var.2	Var.1	Var.2
1.	✓	✓	✓	✓	✓	✓	✓	✓
2.	✓	✓	✓	✓	✓	✓	✓	✓
3.	✓	✓	✓	✓	✓	✓	✓	✓
4.	✓	✓	✓	✓	✓	✓	✓	✓
5.	✓	✓	✓	✓	✓	✓	✓	✓
6.	✓	✓	✓	✓	✓	✓	✓	✓
7.	✓	✓	✓	✓	✓	✓	✓	✓
8.	✓	✓	✓	✓	✓	✓	✓	✓
9.	✓	✓	✓	✓	✓	✓	✓	✓
10.	✓	✓	✓	✓	✓	✓	✓	✓

Pengujian terhadap fungsional kalimat perintah dicoba lagi menggunakan orang yang berbeda dengan dialek bahasa Indonesia yang agak berbeda, hal ini dilakukan untuk mengetahui pengaruh dialek dalam pembacaan kalimat-kalimat perintah yang sudah diatur sebelumnya. Hasil yang didapat cukup baik, namun ada kendala pada pengucapan yang terlalu cepat. Pembacaan kalimat perintah menjadi tidak maksimal dan cenderung salah mengartikan perintah masukan dari pengguna ketika perintah masukan diucapkan dengan cepat. Berikut merupakan hasil dari percobaan kedua yang dilakukan dengan orang yang berbeda :

Tabel 3 Hasil pengujian fungsional kalimat-kalimat perintah suara dengan orang yang berbeda

Percobaan	Kamar tidur utama		Kamar tidur tamu		Ruang tamu		Kipas Angin	
	Var. 1	Var.2	Var.1	Var.2	Var.1	Var.2	Var.1	Var.2
1.	✗	✗	✓	✗	✓	✓	✓	✓
2.	✓	✓	✗	✓	✓	✓	✓	✓
3.	✓	✓	✓	✓	✓	✓	✓	✓
4.	✓	✓	✓	✓	✓	✓	✓	✓
5.	✓	✓	✓	✓	✓	✓	✓	✓
6.	✓	✓	✓	✓	✓	✓	✓	✓

7.	✓	✓	✓	✓	✓	✓	✓	✓
8.	✓	✓	✓	✓	✓	✓	✓	✓
9.	✓	✓	✓	✓	✓	✓	✓	✓
10.	✓	✓	✓	✓	✓	✓	✓	✓

Pengujian delay pengendalian tombol pada aplikasi Smart Home

Pada tahap ini, dilakukan pengujian terhadap *delay* yang dihasilkan sistem ketika menggunakan pengendalian tombol yang ada pada aplikasi Smart Home. Data yang diambil adalah waktu data diterima dan waktu data dikirim. Pengujian dilakukan sebanyak dua (2) kali, pada kondisi jaringan yang bagus dan pada kondisi jaringan yang jelek. Masing-masing pengujian dilakukan dalam sepuluh (10) kali percobaan pada tiga (3) jarak pengukuran, yaitu jarak 3 meter, 9 meter, dan 15 meter.

Tabel 4 Pengukuran delay pengendalian tombol pada jarak 3 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	03:49:05	03:49:06	1 detik
2.	03:49:09	03:49:10	1 detik
3.	03:49:13	03:49:16	3 detik
4.	03:49:18	03:49:19	1 detik
5.	03:49:22	03:49:23	1 detik
6.	03:49:26	03:49:27	1 detik
7.	03:49:31	03:49:32	1 detik
8.	03:49:35	03:49:36	1 detik
9.	03:49:48	03:49:51	3 detik
10.	03:49:53	03:49:54	1 detik
Rata-rata <i>delay</i>			1,4 detik

Tabel 5 Pengukuran delay pengendalian tombol pada jarak 9 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	04:00:41	04:00:42	1 detik
2.	04:00:42	04:00:45	3 detik
3.	04:00:46	04:00:47	1 detik
4.	04:00:49	04:00:50	1 detik

5.	04:00:52	04:00:54	2 detik
6.	04:00:55	04:00:57	2 detik
7.	04:00:59	04:01:00	1 detik
8.	04:01:02	04:01:03	1 detik
9.	04:01:06	04:01:07	1 detik
10.	04:01:09	04:01:11	2 detik
Rata-rata <i>delay</i>			1,5 detik

Tabel 6 Pengukuran delay pengendalian tombol pada jarak 15 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	04:09:50	04:09:52	2 detik
2.	04:09:56	04:09:58	2 detik
3.	04:10:00	04:10:03	3 detik
4.	04:10:04	04:10:05	1 detik
5.	04:10:08	04:10:09	1 detik
6.	04:10:11	04:10:14	3 detik
7.	04:10:15	04:10:17	2 detik
8.	04:10:19	04:10:20	1 detik
9.	04:10:23	04:10:24	1 detik
10.	04:10:26	04:10:27	1 detik
Rata-rata <i>delay</i>			1,7 detik

Tabel 7 Pengukuran delay pengendalian tombol pada jarak 3 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	10:11:08	10:11:09	1 detik
2.	10:11:14	10:11:15	1 detik
3.	10:11:18	10:11:19	1 detik
4.	10:11:22	10:11:23	1 detik
5.	10:11:27	10:11:28	1 detik
6.	10:11:32	10:11:33	1 detik
7.	10:11:36	10:11:37	1 detik

8.	10:11:40	10:11:41	1 detik
9.	10:11:45	10:11:46	1 detik
10.	10:11:48	10:11:49	1 detik
Rata-rata <i>delay</i>			1 detik

Tabel 8 Pengukuran delay pengendalian tombol pada jarak 9 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	10:22:45	10:22:47	2 detik
2.	10:22:48	10:22:49	1 detik
3.	10:22:51	10:22:52	1 detik
4.	10:22:54	10:22:55	1 detik
5.	10:22:56	10:22:57	1 detik
6.	10:23:01	10:23:02	1 detik
7.	10:23:05	10:23:06	1 detik
8.	10:23:10	10:23:11	1 detik
9.	10:23:14	10:23:15	1 detik
10.	10:23:17	10:23:18	1 detik
Rata-rata <i>delay</i>			1,1 detik

Tabel 9 Pengukuran delay pengendalian tombol pada jarak 15 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	10:38:12	10:38:13	1 detik
2.	10:38:15	10:38:17	2 detik
3.	10:38:20	10:38:21	1 detik
4.	10:38:24	10:38:25	1 detik
5.	10:38:28	10:38:30	2 detik
6.	10:38:34	10:38:36	2 detik
7.	10:38:37	10:38:38	1 detik
8.	10:38:40	10:38:41	1 detik
9.	10:38:44	10:38:45	1 detik
10.	10:38:48	10:38:49	1 detik

Rata-rata <i>delay</i>	1,3 detik
------------------------	-----------

Berdasarkan hasil pengujian pada tabel 4.7 hingga tabel 4.9, didapatkan hasil yang kurang memuaskan, di mana nilai rata-rata *delay* yang di hasilkan adalah 1,3 detik hingga 1,7 detik. Hal ini terjadi karena dua (2) faktor utama, yang pertama adalah nilai yang didapat pada pengujian, hanya dapat menggunakan nilai pada Blynk yang di perbarui tiap 1 detik sedangkan menurut standarisasi *delay* pada tabel 2.6, nilai *delay* yang ideal berada dikisaran milidetik. Hal tersebut membuat perbandingan nilai *delay* yang dihasilkan dengan standarisasi *delay* cukup sulit untuk dilakukan. Faktor yang kedua adalah masalah pada jaringan yang digunakan untuk melakukan pengendalian pada sistem ini, di mana kecepatan Wi-Fi yang digunakan pada saat pengujian yang pertama ini berkisar diantara 3 mbps hingga 5 mbps.

Berdasarkan hasil pengujian pada tabel 4.10 hingga tabel 4.12, didapatkan hasil yang cukup memuaskan, di mana nilai rata-rata *delay* yang dihasilkan adalah 1 detik hingga 1,3 detik. Hal ini memang masih berada dalam indeks “sangat buruk” bila dibandingkan dengan standarisasi *delay* pada tabel 2.6, namun bila dilihat berdasarkan faktor pertama pada paragraf sebelumnya hal ini disebabkan oleh pembacaan data dari Blynk yang hanya dapat diperbaharui tiap 1 detik, walaupun *delay* yang dihasilkan berada dikisaran milidetik, pembacaannya akan selalu berada dikisaran 1 detik. Pada pengujian yang kedua ini, kecepatan Wi-Fi yang digunakan berkisar antara 11 mbps hingga 18 mbps.

Pengujian terhadap *delay* yang dihasilkan pada sistem pengendalian menggunakan tombol mendapatkan nilai rata-rata *delay* yang bervariasi, di mana untuk pengujian kedua mendapatkan nilai rata-rata *delay* yang lebih bagus dibandingkan pengujian pertama. Hal ini dapat terjadi karena faktor jaringan yang cukup mempengaruhi sistem pengendalian ini, di mana untuk pengujian kedua dilakukan pada kecepatan jaringan Wi-Fi yang lebih baik dibandingkan pengujian pertama.

Pengujian delay pengendalian suara pada aplikasi Smart Home

Pada tahap ini, dilakukan pengujian terhadap *delay* yang dihasilkan sistem ketika menggunakan pengendalian suara yang ada pada aplikasi Smart Home. Data yang diambil adalah waktu data diterima dan waktu data dikirim. Pengujian dilakukan sebanyak dua (2) kali, pada kondisi jaringan yang bagus dan pada kondisi jaringan yang jelek. Masing-masing pengujian dilakukan dalam sepuluh (10) kali percobaan pada tiga (3) jarak pengukuran, yaitu jarak 3 meter, 9 meter, dan 15 meter.

Tabel 10 Pengukuran *delay* pengendalian suara pada jarak 3 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	23:26:23	23:26:27	4 detik
2.	23:26:32	23:26:35	3 detik
3.	23:26:41	23:26:45	4 detik

4.	23:26:54	23:26:58	4 detik
5.	23:27:15	23:27:18	3 detik
6.	23:27:24	23:27:27	3 detik
7.	23:27:34	23:27:38	4 detik
8.	23:27:43	23:27:47	4 detik
9.	23:27:55	23:27:59	4 detik
10.	23:28:07	23:28:11	4 detik
Rata-rata <i>delay</i>			3,7 detik

Tabel 11 Pengukuran delay pengendalian suara pada jarak 9 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	23:31:25	23:31:29	4 detik
2.	23:31:43	23:31:46	3 detik
3.	23:32:00	23:32:04	4 detik
4.	23:32:13	23:32:17	4 detik
5.	23:32:26	23:32:30	4 detik
6.	23:32:35	23:32:38	3 detik
7.	23:32:44	23:32:48	4 detik
8.	23:32:54	23:32:58	4 detik
9.	23:33:07	23:33:11	4 detik
10.	23:33:18	23:33:22	4 detik
Rata-rata <i>delay</i>			3,8 detik

Tabel 12 Pengukuran delay pengendalian suara pada jarak 15 meter (kondisi jaringan jelek)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	00:14:03	00:14:07	4 detik
2.	00:14:14	00:14:18	4 detik
3.	00:14:23	00:14:26	3 detik
4.	00:14:30	00:14:34	4 detik
5.	00:14:39	00:14:42	3 detik
6.	00:14:47	00:14:51	4 detik

7.	00:14:56	00:15:00	4 detik
8.	00:15:04	00:15:08	4 detik
9.	00:15:19	00:15:22	3 detik
10.	00:15:25	00:15:29	4 detik
Rata-rata <i>delay</i>			3,7 detik

Tabel 13 Pengukuran delay pengendalian suara pada jarak 3 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	19:40:22	19:40:23	1 detik
2.	19:40:29	19:40:30	1 detik
3.	19:40:36	19:40:37	1 detik
4.	19:40:41	19:40:42	1 detik
5.	19:40:49	19:40:50	1 detik
6.	19:40:55	19:40:56	1 detik
7.	19:41:03	19:41:05	2 detik
8.	19:41:09	19:41:10	1 detik
9.	19:41:16	19:41:17	1 detik
10.	19:41:22	19:41:23	1 detik
Rata-rata <i>delay</i>			1,1 detik

Tabel 14 Pengukuran delay pengendalian suara pada jarak 9 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	19:55:12	19:55:13	1 detik
2.	19:55:18	19:55:19	1 detik
3.	19:55:26	19:55:27	1 detik
4.	19:55:31	19:55:32	1 detik
5.	19:55:36	19:55:37	1 detik
6.	19:55:47	19:55:48	1 detik
7.	19:55:51	19:55:52	1 detik
8.	19:55:59	19:56:00	1 detik
9.	19:56:07	19:56:08	1 detik

10.	19:56:14	19:56:15	1 detik
Rata-rata <i>delay</i>			1 detik

Tabel 15 Pengukuran delay pengendalian suara pada jarak 15 meter (kondisi jaringan bagus)

No.	Waktu data dikirim	Waktu data diterima	Delay
1.	20:12:18	20:12:20	2 detik
2.	20:12:25	20:12:27	2 detik
3.	20:12:30	20:12:32	2 detik
4.	20:12:39	20:12:40	1 detik
5.	20:12:45	20:12:47	2 detik
6.	20:12:51	20:12:52	1 detik
7.	20:12:58	20:12:59	1 detik
8.	20:13:06	20:13:07	1 detik
9.	20:13:13	20:13:16	3 detik
10.	20:13:20	20:13:21	1 detik
Rata-rata <i>delay</i>			1,6 detik

Berdasarkan hasil pengujian pada tabel 4.13 hingga tabel 4.15, didapatkan hasil yang kurang memuaskan, di mana nilai rata-rata delay yang dihasilkan adalah 3,7 detik hingga 3,8 detik. Hal ini terjadi karena dua (2) faktor utama, yang pertama adalah nilai yang didapat pada pengujian, hanya dapat menggunakan nilai pada Blynk yang diperbarui tiap 1 detik sedangkan menurut standarisasi *delay* pada tabel 2.6, nilai *delay* yang ideal berada dikisaran milidetik. Hal tersebut membuat perbandingan nilai *delay* yang dihasilkan dengan standarisasi *delay* cukup sulit untuk dilakukan. Faktor yang kedua adalah masalah pada jaringan yang digunakan untuk melakukan pengendalian pada sistem ini, di mana kecepatan Wi-Fi yang digunakan pada saat pengujian yang pertama ini berkisar diantara 1 mbps hingga 3 mbps.

Berdasarkan hasil pengujian pada tabel 4.16 hingga tabel 4.18, didapatkan hasil yang cukup memuaskan, di mana nilai rata-rata delay yang dihasilkan adalah 1 detik hingga 1,6 detik. Hal ini memang masih berada dalam indeks “sangat buruk” bila dibandingkan dengan standarisasi delay pada tabel 2.6, namun bila dilihat berdasarkan faktor pertama pada paragraf sebelumnya hal ini disebabkan oleh pembacaan data dari Blynk yang hanya dapat diperbaharui tiap 1 detik, walaupun delay yang dihasilkan berada dikisaran milidetik, pembacaannya akan selalu berada dikisaran 1 detik. Pada pengujian yang kedua ini, kecepatan Wi-Fi yang digunakan berkisar antara 10 mbps hingga 18 mbps.

Pengujian terhadap *delay* yang dihasilkan pada sistem pengendalian menggunakan tombol mendapatkan nilai rata-rata *delay* yang bervariasi, di mana

untuk pengujian kedua mendapatkan nilai rata-rata *delay* yang lebih bagus dibandingkan pengujian pertama. Hal ini dapat terjadi karena faktor jaringan yang cukup mempengaruhi sistem pengendalian ini, di mana untuk pengujian kedua dilakukan pada kecepatan jaringan Wi-Fi yang lebih baik dibandingkan pengujian pertama.

Kesimpulan

Berdasarkan hasil percobaan dan pengujian dapat disimpulkan bahwa sistem prototipe smart home menggunakan *voice control* dan Blynk dapat bekerja dengan baik. Sistem pengendalian menggunakan tombol maupun suara dapat berfungsi dengan baik apabila digunakan pada koneksi jaringan yang bagus. Pengukuran sensor empat (4) buah tegangan memiliki persentase galat yang cukup kecil, berkisar antara 0,03% hingga 1,57%. Pengujian fungsional kalimat perintah suara memiliki angka keberhasilan 100% pada percobaan pertama dan 95% pada percobaan kedua dengan orang yang berbeda. Sistem dapat mengirimkan kondisi beban ke database Blynk.

Daftar Pustaka

- Abdul Halim Mukti Nasution, S. I. (2019). Pengontrolan Lampu Jarak Jauh Dengan NodeMCU Menggunakan Blynk. *Jurnal TEKINKOM*.
- Agus Wagyana, R. (2019). Prototipe Modul Praktik untuk Pengembangan . *JURNAL ILMIAH SETRUM, Vol. 8, No. 1*.
- ETSI. (2000). *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)*. France: European Telecommunications Standards Institute.
- Florantina Cherli I. L. Herin, H. P. (2019). Voice Control Sebagai Pengendali Peralatan Elektronik Berbasis NodeMCU. *Comasie Journal*.
- Google. (n.d.). *Bantuan Asisten Google*. Retrieved Februari 20, 2022, from <https://support.google.com/assistant/answer/7172657?hl=id&co=GENIE.Platform%3DiOS&oco=0#zippy=%2Cperiksa-apakah-perangkat-anda-memiliki-spesifikasi-yang-dibutuhkan>
- Lestari, N. (2016). Implementasi Perangkat WIFI Sebagai Pengendali Robot Mobil Pemantau. *JUSIKOM*.
- Mario, B. P. (2018). Rancang Bangun Sistem Proteksi dan Monitoring Penggunaan Daya Listrik . *PRISMA FISIKA*.
- Muhamad Suryanto, F. A. (2021). Rancang Bangun Sistem Smartphone Berbasis Internet of Things Dengan NodeMCU dan Google Assistant di Smartphone Android. *TESLA, Vol. 23*.
- NIAGAHOSTER. (n.d.). *Panduan Lengkap dan Cara Mudah Menggunakan IFTTT*. Retrieved Februari 22, 2022, from <https://www.niagahoster.co.id/blog/cara-menggunakan-ifttt/>

- Nur Afiyat, M. H. (2021). Prototype Sistem Pengendalian Perangkat Elektronik Berbasis IoT (Internet of Things) Menggunakan Voice Control dan Blynk. *JURNAL RESISTOR, Vol. 4 No. 1.*
- Rafiq hariri, M. A. (2019). Perancangan Aplikasi Blynk Untuk Monitoring dan Kendali Penyiraman Tanaman. *JURNAL ELEKTRIKAL, Vol. 6, No. 1.*
- Reski Damayanti, A. M. (2020). Rancang Bangun Smart Home Berbasis Internet of Things. *JOURNAL OF APPLIED SMART ELECTRICAL NETWORK AND SYSTEMS (JASENS).*
- Rita Dewi Risanty, L. A. (2009). Rancang Bangun Sistem Pengendalian Listrik Ruangan Dengan Menggunakan ATMEGA 328 dan SMS Gateway Sebagai Media Informasi. *Jurnal Sistem Informasi, Vol. 7, No. 2.*
- Wikipedia Ensiklopedia Bebas. (2022, Januari 10). *Google Assistant*. Retrieved Februari 2, 2022, from https://id.wikipedia.org/wiki/Google_Assistant

USDB22Naskah

ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes Off

Exclude matches < 5%

Exclude bibliography On