

RANCANG BANGUN SISTEM SMARTHOME BERBASIS INTERNET OF THINGS DENGAN NODE MCU DAN GOOGLE ASSISTANT DI SMARTPHONE ANDROID

Muhamad Suryanto ¹,

Program Studi Teknik Elektro Universitas Muhammadiyah Palembang
Email: muhamadsuryanto@gmail.com

Feby Ardianto ¹

Program Studi Teknik Elektro Universitas Muhammadiyah Palembang
Email: ardianto.feby@gmail.com

Bengawan Alfaresi ¹

Program Studi Teknik Elektro Universitas Muhammadiyah Palembang
Email: bengawan_alfaresi@um.palembang.ac.id

ABSTRACTS : *Science and technology that continues to develop have a major influence in completing work. By using Internet of things (IoT) technology, electronic devices will have the ability to communicate with each other, send and receive data through an internet network connection. The science of internet of things can be applied to monitoring or controlling systems in certain environments such as Smarthomes. Smarthome systems are currently mostly made only based on applications, even if there is a Smarthome with voice commands such as Google assistant and even then the price will be expensive and one device can only control one device, therefore in this research, Smarthome Design Based on the Internet of Things Using NodeMcu is carried out. With Google Assistant on Android Smartphone. In this study there are several stages, namely the stage of determining hardware and software, designing, programming, and finally testing. In this research, NodeMcu is needed as a microcontroller, which functions to control electrical equipment, process data, send and receive data to the Blynk server. IFTTT itself acts as a liaison between 2 platforms, namely Blynk and Google Assistant. The final result of this research is a Smarthome system that can be controlled by voice commands or through an application on an Android phone*
Keywords: *Smarthome, Voice Command, IoT, NodeMCU..*

ABSTRAK: Ilmu pengetahuan dan teknologi yang terus berkembang memberikan pengaruh besar dalam menyelesaikan pekerjaan. Dengan menggunakan teknologi *Internet of things (IoT)* perangkat-perangkat elektronik akan memiliki kemampuan untuk saling berkomunikasi, saling mengirim dan menerima data melalui koneksi jaringan internet. Ilmu internet of things dapat diterapkan pada sistem monitoring atau pengendali pada lingkungan tertentu seperti Smarthome. Sistem Smarthome pada saat ini kebanyakan dibuat hanya berbasis aplikasi saja, walaupun ada Smarthome dengan voice command seperti Google assistant itupun harganya akan mahal dan satu perangkat hanya bisa Mengontrol satu alat saja, oleh karena itu dalam penelitian ini dilakukan Perancangan Smarthome Berbasis *Internet Of Things* Menggunakan NodeMCU Dengan *Google Assistant* Di *Smartphone Android*. Pada penelitian ini terdapat beberapa tahapan, yaitu tahap penentuan hardware dan Software, perancangan, pemrograman, dan yang terakhir pengujian. Pada penelitian ini, dibutuhkan NodeMcu sebagai mikrokontroler, yang berfungsi untuk mengendalikan peralatan listrik, mengolah data, mengirim dan menerima data ke server Blynk. IFTTT sendiri sebagai penghubung antara 2 buah platform yaitu Blynk dan *Google Assistant*. Hasil akhir dari penelitian ini adalah sistem Smarthome yang dapat dikendalikan Perintah suara maupun melalui aplikasi di handphone Android

Kata Kunci: Smarthome, Perintah Suara, IoT, NodeMcu.

PENDAHULUAN

Berkembangnya ilmu pengetahuan dan teknologi saat ini memberikan pengaruh besar dalam menyelesaikan pekerjaan. Ilmu yang diterapkan pada mesin dan elektronika dapat diselesaikan dengan lebih efektif dan efisien. Berkembangnya ilmu pengetahuan dan teknologi juga mendorong manusia untuk terus berpikir kreatif tidak hanya menemukan sesuatu yang baru, tapi juga memaksimalkan kinerja suatu teknologi [1]. Perkembangan teknologi yang makin pesat memungkinkan terbentuknya sistem yang saling terhubung melalui koneksi internet sebagai medianya.

Internet of things (IoT) ialah teknologi yang memungkinkan perangkat-perangkat elektronik dapat memiliki kemampuan untuk saling berkomunikasi, saling mengirim dan menerima data melalui jaringan internet, IoT dapat diterapkan pada sistem *monitoring* atau pengendali pada lingkungan tertentu seperti *smarthome* [2].

Konsep *smarthome* dapat memungkinkan pengendalian peralatan elektronik dengan menggunakan perintah dari mana saja, dimana manusia tidak perlu bergerak mendekati sebuah peralatan rumah tangga hanya untuk menghidupkan atau mematikan peralatan tersebut, melainkan dapat dikendalikan dari jarak jauh salah satunya dapat melalui perintah suara dari penghuni rumah tersebut [3].

¹ Program Studi Teknik Elektro Universitas Muhammadiyah Palembang

Smartphone yang banyak dipakai oleh masyarakat adalah *smartphone* dengan sistem operasi *Android*. Salah satu fitur yang disediakan *smartphone Android* adalah *speech command*. *Speech command* merupakan proses identifikasi untuk mengenali kata yang diucapkan oleh seseorang, contohnya *Google Assistant*. Selain itu *smartphone* dengan koneksi ke jaringan internet dapat digunakan untuk mengendalikan mikrokontroler yang juga terhubung dengan internet [4].

NodeMCU merupakan modul mikrokontroler yang didesain dengan konektivitas jaringan Wifi didalamnya. NodeMCU berbasis bahasa pemrograman Lua namun dalam penerapannya dapat menggunakan Arduino IDE untuk memprogramnya, sehingga menjadi lebih *simple* dalam penggunaannya [5] selain itu NodeMCU lebih murah dibandingkan mikrokontroler yang lain dengan fungsi yang relatif sama.

Sistem *Smarthome* pada saat ini kebanyakan dibuat hanya berbasis aplikasi saja, walaupun ada *Smarthome* dengan *voice command* seperti *Google assistant* itupun harganya akan mahal dan satu perangkat hanya bisa mengontrol satu alat saja, oleh karena itu dalam penelitian ini dilakukan Perancangan *Smarthome* Berbasis *Internet Of Things* Menggunakan *NodeMcu* Dengan *Google Assistant* Di *Smartphone Android*.

TINJAUAN PUSTAKA

A. *Internet of Things*

Internet of Things (IoT) adalah konsep yang memanfaatkan konektivitas internet, dengan berkembangnya infrastruktur internet, bukan hanya *smartphone* atau komputer saja yang bisa terhubung dengan internet. Namun semua benda elektronik akan dapat terkoneksi dengan internet [6].

Teknologi IoT untuk saling terhubung berkomunikasi ini membuat IoT dapat diterapkan pada segala bidang. Contohnya di bidang pertanian, bidang kesehatan, bidang smart building, dan sebagainya, IoT dalam bidang kelistrikan dapat digunakan untuk memonitor penggunaan energi listrik. IoT juga dapat digunakan di bidang *Smarthome*, *automation*, industri, transportasi lainnya. [7]

B. *Blynk*

Blynk adalah *platform* yang memungkinkan membuat *Interface* untuk proyek IoT baik *Hardware* dari *Smartphone* seperti *Android* dengan kinerja yang sangat cepat. Aplikasi *Blynk* memiliki banyak fitur pada *Widget* seperti mengatur tombol, grafik, *timer*, terminal, dan lainnya. *Widget* dapat mengendalikan pin pada mikrokontroler atau menampilkan data dari sensor. *Blynk* sangat cocok untuk proyek yang sederhana, karena *Blynk* dapat menghemat waktu dan sumber daya yang diperlukan ketika membangun aplikasi IoT. *Blynk* memudahkan dalam pengodingan sehingga tidak diperlukan *coding* yang sangat panjang [6].

Blynk memiliki tiga komponen utama, yaitu Aplikasi, *Server*, dan *Libraries*. *Blynk server* berfungsi untuk menangani semua komunikasi diantara aplikasi di *smartphone*. Jenis *server* yang bisa menggunakan *Blynk* yaitu *Cloud* atau *server* sendiri (*private*) [5].

C. *Smarthome*

Smarthome adalah sebuah konsep dimana di dalam rumah terdapat sistem yang berjalan secara otomatis dan dapat dikendalikan dari jauh untuk meningkatkan kenyamanan dan keamanan penghuni hunian. *Smarthome* merujuk pada rumah yang memiliki sistem otomatis yang canggih. *Smarthome* menghubungkan sistem tersebut kedalam satu jaringan yang terintegrasi dengan kontrol yang terpusat, salah satu tujuan dari *Smarthome* adalah meningkatkan keamanan dan kenyamanan penghuni rumah dengan kebutuhan masing masing. Melalui sistem *Smarthome* penghuni rumah dapat mengontrol sistem – sistem yang terhubung sesuai dengan kebutuhannya, perangkat yang ada di dalam *Smarthome* dapat terhubung dengan menggunakan berbagai cara salah satunya menggunakan teknologi seperti WiFi [2].

Secara umum *smarthome* memerlukan 3 syarat agar bisa disebut *smart*, yaitu 1. *Internal Network* : berupa kabel, *wireless* 2. *Intelligent Control* : berupa *gateway* untuk mengelola sistem 3. *Home Automation* : mengatur dan mengelola [7].

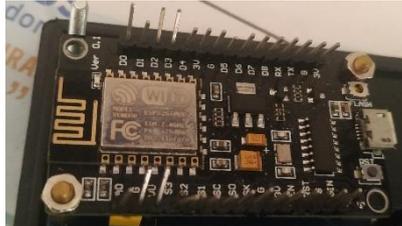
D. *NodeMCU*

NodeMCU adalah sebuah *platform* IoT yang bersifat *open source*. Terdiri dari perangkat keras dengan *System On Chip* (SoC) ESP8266-12, juga *firmware* yang menggunakan bahasa pemrograman *scripting Lua*. NodeMCU sebenarnya mengacu pada *firmware* yang digunakan dari pada perangkat kerasnya. NodeMCU dapat dianalogikan sebagai *board* Arduino-nya ESP8266. NodeMCU telah menggabungkan ESP8266 ke dalam sebuah *board* dengan berbagai fungsi seperti mikrokontroler umumnya, ditambah juga dengan kemampuan akses terhadap Wifi juga fitur komunikasi *USB to Serial* sehingga untuk memprogramnya hanya diperlukan sebuah kabel data mikro USB. Secara umum ada tiga produsen NodeMCU yang produknya banyak beredar yaitu: DOIT, Amica, dan Lolin/WeMos. Dengan beberapa jenis varian *board* yang diproduksi yakni Versi1,

Versi2 dan yang terbaru saat ini adalah Versi3 [8].

NodeMCU adalah sebuah pengembangan dari ESP8266 dengan *firmware* yang berbasis e-Lua, pada NodeMCU dilengkapi dengan micro usb port yang berfungsi untuk memasukan program ke NodeMcu maupun sebagai sumber listrik untuk menyalakan NodeMCU. Salain itu pada NodeMCU mempunyai dua buah tombol yaitu tombol reset dan flash. NodeMCUpada dasarnya menggunakan bahasa pemrograman Lua yang merupakan bahasa *default* dari ESP8266. Bahasa Lua memiliki logika dan susunan pemrograman yang mirip dengan bahasa C hanya ada perbedaan pada sintak. Jika ingin menggunakan bahasa Lua maka dapat menggunakan Lua *uploader* maupun Lua *loader*.

Selain dengan Bahasa Lua NodeMCU juga *support* dengan *Software* Arduino IDE dengan melakukan sedikit perubahan pada *board manager* pada Arduino IDE. Sebelum digunakan *board* ini harus *di-flash* terlebih dahulu agar dapat mendukung *tool* yang digunakan. Jika menggunakan Arduino IDE dapat menggunakan *firmware* yang cocok seperti *firmware* dari *Ai-thinker* yang mendukung *AT Command*. Untuk penggunaan *tool* Lua *loader firmware* yang digunakan adalah *firmware* NodeMCU [9]



■ Gambar 1. NodeMCU

E. Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang disediakan oleh Arduino untuk melakukan berbagai proses yang berkaitan pemrograman Arduino. Arduino IDE adalah perangkat lunak *open source* sehingga dapat digunakan pula untuk pemrograman NodeMCU seperti membuat perintah atau *source code*, melakukan pengecekan kesalahan, kompilasi, *upload* program, dan menguji hasil kerja NodeMCU melalui *serial monitor*. Program yang ditulis dengan *Software* Arduino IDE disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam *file* dengan *ekstensi* .ino. IDE ini juga sudah didukung oleh berbagai sistem operasi saat ini seperti Windows, Mac, dan Linux [1].



■ Gambar 2. Arduino IDE

F. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat layar sentuh seperti *Smartphone* dan tablet. *Android* pertama dikembangkan oleh sebuah perusahaan *Android.Inc.*, yang dukungan oleh *Google*. Kemudian *Google* membeli *Android.Inc.* pada tahun 2005. Sejak tahun 2008, *Google* terus meneruskan dan secara bertahap melakukan pembaruan pada sistem *Android* untuk meningkatkan kinerja dari sistem operasi *Android* tersebut, *Google* juga menambahkan fitur-fitur baru, memperbaiki *bug* yang ada pada versi *Android* sebelumnya. Setiap versi yang *Android* dirilis akan dinamakan berurutan secara alfabet dengan berdasarkan nama sebuah makanan penutup atau pencuci mulut. Berikut adalah versi sistem operasi *Android* : *Apple Pie*, *Banana Bread*, *Cupcake*, *Donut*, *Éclair*, *Froyo*, *Gingerbread*, *Honey Comb*, *Ice Cream Sandwich*, *Jelly Bean*, *Kitkat*, *Lollipop*, *Marshmallow*, *Nougat*, *Oreo*, *Pie* [10].

G. Google Assistant

Google Assistant atau Asisten *Google* adalah asisten virtual dengan *Artificial Intelligent* (kecerdasan buatan) yang dikembangkan oleh *Google*. *Google Assistant* saat ini tersedia untuk *smarphone* berbasis *Android* dan perangkat rumah pintar seperti *Google Home*. *Google Assistant* dapat berinteraksi dalam percakapan dua arah. Pengguna dapat berinteraksi dengan *Google Assistant* melalui suara, *input keyboard* juga didukung dan dapat digunakan untuk disabilitas atau saat penangkapan suara kurang baik. *Google Assistant* dapat diperintahkan banyak hal [1], seperti mencari sesuatu di Internet, menyetel alarm, menjadwalkan acara,

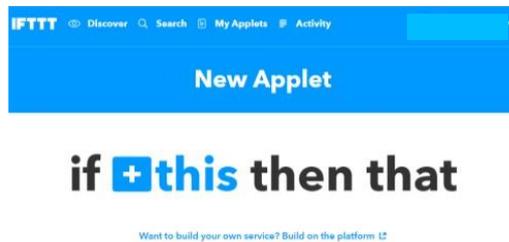
mencarikan musik dan video, hingga mengendalikan perangkat elektronik yang sudah tersambung dengan sistemnya.



■ Gambar 3. Tampilan Google Asisstant

H. *If This Then That (IFTTT)*

IFTTT atau *If This Then That* adalah sebuah aplikasi gratis untuk menghubungkan dua buah *platform*. Contohnya pengguna mendapatkan pesan singkat (sms) setiap ada *email* baru yang masuk, dapat dilakukan dengan menggunakan IFTTT. Pada penelitian ini IFTTT (*If This Then That*) digunakan untuk menghubungkan antara *Blynk* dengan *Google Assistant*. IFTTT mengambil data masukkan dari *Google Assistant* dan dikirim ke *Blynk*. [1].



■ Gambar 4. Tampilan depan *If This Then That*

I. Relay

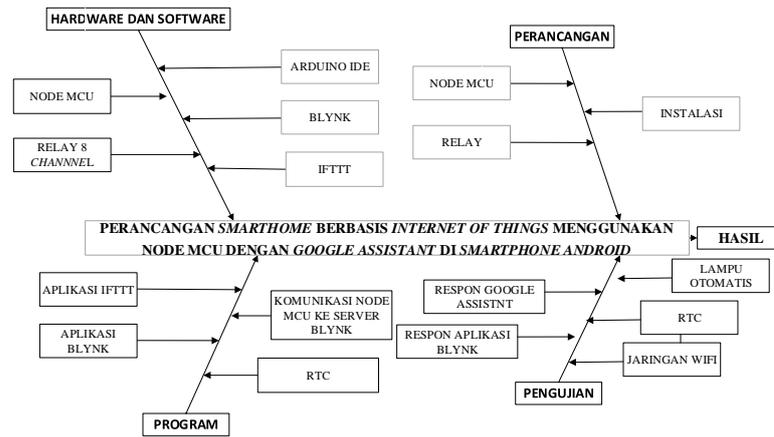
Relay adalah sakelar mekanik yang dikendalikan secara elektronik (elektro magnetik). Sakelar pada relay dapat terjadi perubahan posisi off ke on pada saat arus listrik pada kumparan relay tersebut. Relay mempunyai dua bagian utama yaitu sakelar kontak mekanik dan pembangkit elektromagnetik (kumparan). Sakelar pada relay dikendalikan menggunakan tegangan listrik yang diberikan ke kumparan yang menghasilkan elektro magnetik untuk menarik tuas sakelar *relay*. Kumparan adalah gulungan kawat yang dapat arus listrik, sedang saklar kontak mekanik adalah sakelar yang posisi pergerakannya tergantung pada ada tidaknya arus listrik yang mengalir di kumparan. [8]

Relay biasanya memiliki tiga kontak. Kontak biasanya diberi label *Common (COM)*, *Normally Open (NO)*, dan *Normally Close (NC)*. Pada NC, kontak akan terhubung ke kontak COM ketika coil tidak diberi daya. Pada NO, kontak akan terhubung hanya jika ketika ada daya yang diberikan pada coil. Ketika daya listrik diberikan, maka COM akan terhubung dengan kontak NO dan kontak NC dibiarkan terputus. [9]

METODE PENELITIAN

A. Diagram *Fishbone*

Berikut gambar di bawah ini merupakan diagram *Fishbone* dimana ada empat proses yaitu : *Hardware* dan *Software*, Perancangan, Program, dan Pengujian.



■ Gambar 5. Diagram Fishbone

B. Alat dan bahan

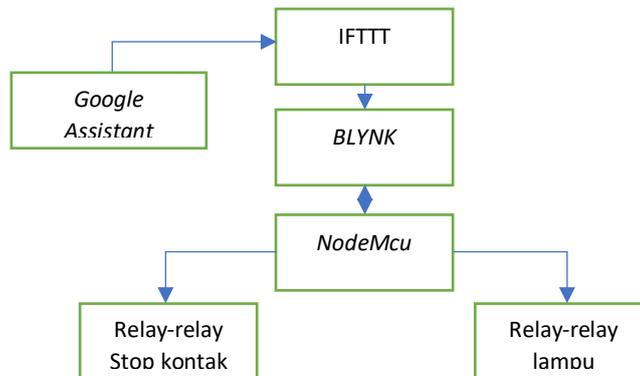
Penelitian ini membutuhkan beberapa alat dan bahan yang akan digunakan dalam perancangan *smarthome* berbasis *Internet of Things* Menggunakan *NodeMCU* dengan *Google Assistant* di *smartphone android*. Alat dan bahan yang diperlukan adalah sebagai berikut:

■ Tabel 1. Tabel Alat dan Bahan

No	Alat dan Bahan	Jumlah
1	<i>NodeMCU</i>	1
2	<i>Relay 8 Channel</i>	1
3	<i>Kabel Jumper</i>	Secukupnya
4	<i>Fitting Lampu</i>	2
5	Stop Kontak	1
6	<i>Box</i>	1
7	Obeng	1
8	Tang potong	1
9	Akrilik	34cm x 24cm

C. Diagram Blok Sistem

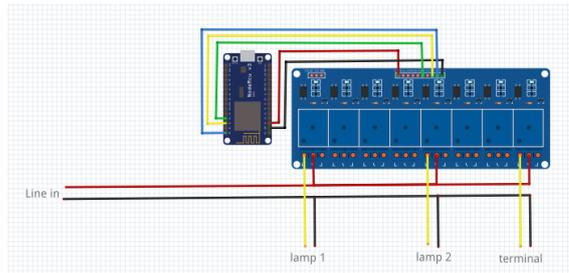
Pada penelitian ini, dibutuhkan *NodeMCU* sebagai mikrokontroler, yang berfungsi untuk mengendalikan peralatan listrik, mengolah data, mengirim dan menerima data ke *server Blynk*. *IFTTT* sendiri sebagai penghubung antara 2 buah *platform* yaitu *Blynk* dan *Google Assistant*.



■ Gambar 6. Diagram Blok Sistem

Diagram Blok sistem pada penelitian ini dibuat berdasarkan kebutuhan dari sistem *smarthome* itu sendiri, semua blok-blok dalam sistem akan saling terhubung secara dua arah maupun satu arah, hubungan dua arah hanya terdapat pada blok *Blynk* dan *NodeMCU*, dikarenakan disana akan terjadi pertukaran data secara sinkron antara keduanya, sedangkan yang lainnya hanya terhubung satu arah. *Google Assistant* hanya bisa

mengirim data ke *IFTTT* dan tidak bisa sebaliknya, begitu pula *IFTTT* hanya bisa mengirim data ke *Blynk*, tidak bisa sebaliknya.



■ **Gambar 7.** Rangkaian Peralatan

Gambar 3.3 menunjukkan rangkaian yang digunakan pada peralatan *Smarthome*. *Relay* dan *NodeMCU* akan terhubung secara langsung melalui pin-pin yang ada pada *NodeMCU* dan *Relay*. Pada *NodeMCU* dan *relay* terhubung dengan listrik DC, sedangkan pada bagian kontak relay terhubung dengan listrik AC. Listrik AC diambil langsung dari sumber listrik bisa berupa sumber listrik milik PLN dengan tegangan 220V. Listrik AC tersebut masuk ke kontak-kontak *Relay* dan kemudian akan disalurkan ke beban.

HASIL DAN PEMBAHASAN

A. Hasil Perancangan



■ **Gambar 8.** Hasil Perancangan

Gambar 8 adalah hasil akhir perangkat keras dari Perancangan *Smarthome* Berbasis Internet of Things Menggunakan *NodeMcu*. Di dalam perangkat keras yang dibuat, terdapat 2 perangkat utama yaitu *NodeMcu* sebagai mikrokontroler yang memungkinkan untuk terkoneksi ke sebuah jaringan wifi yang tersambung ke internet, dan sebuah modul relay 8 *channel* yang digunakan untuk memutus maupun menyambungkan arus listrik ke beban yang akan dipasangkan ke dalam sistem *smarthome* ini. Sistem *Smarthome* Berbasis Internet of Things Menggunakan *NodeMcu* ini digunakan untuk mengendalikan beban berupa 2 buah lampu dan 1 buah stop kontak yang terhubung ke relay *channel* 1, 2, dan 4. Dengan konfigurasi *channel* 1 dan 2 sebagai pengendali untuk lampu, dan *channel* 4 sebagai pengendali untuk sebuah stop kontak. Relay 8 *channel* terhubung dengan *NodeMCU* dengan konfigurasi pin D0 terhubung ke relay *channel* 1, pin D3 terhubung dengan relay *channel* 2, dan pin D2 terhubung ke relay *channel* 4.

■ **Tabel 2.** Konfigurasi hubung perangkat

<i>NodeMcu</i> Pin	Relay Chanel	Beban
D0	Chanel 1	Lampu
D2	Chanel 1	Stop Kontak
D3	Chanel 1	Lampu

B. Hasil Perancangan Perangkat Lunak

Perangkat lunak pada penelitian ini terbagi menjadi beberapa macam yaitu perangkat lunak atau program yang di tulis pada *NodeMcu* sebagai kontroller dari *Smarthome*, perangkat lunak aplikasi *Android Blynk* sebagai *Interface* dari *Smarthome* dan sebagai *server* yang menghubungkan antara *NodeMcu* dan *IFTTT*, dan perangkat lunak *IFTTT* yang berfungsi untuk mengatur perintah dan respon *Google Assistant* dan perintah tersebut akan diteruskan ke aplikasi *Blynk*.

1. Implementasi Interface Smarthome Blynk



■ **Gambar 9.** Interface Smarthome aplikasi Blynk

Aplikasi *Blynk* digunakan sebagai *Interface* dari *Smarthome*, dengan beberapa kontrol diantaranya yaitu kontrol untuk lampu 1, lampu 2 dan untuk stop kontak, selain itu pada aplikasi *Blynk* dibuat pula sebuah tombol untuk mengaktifkan semua lampu secara sekaligus dan tombol untuk megaktifkan program penghidupan lampu otomatis. Kontrol pada aplikasi *Blynk* terhubung dengan peralatan melalui *Virtual* pin yang sudah tersinkron ke program untuk mengendalikan pin pada *NodeMcu* yang menuju ke relay untuk mengendalikan peralatan listrik rumah. *Server Blynk* memiliki *IP (internet Protocol) Address* publik yang dapat diakses langsung, *IP Address* nya adalah 188.166.206.43.

■ **Tabel 3.** Konfigurasi Interface Aplikasi Blynk

Interface aplikasi Blynk	Virtual pin Blynk	Peralatan yang dikendalikan
<i>Auto lamp</i>	V0	Lampu 1 dan 2
<i>Lamp 1</i>	V1	Lampu 1
<i>Lamp 2</i>	V2	Lampu 2
Terminal	V3	Terminal
<i>All lamp</i>	V4	Lampu 1 dan 2

2. Implementasi Perintah dan respon Google Assistant dengan IFTTT

IFTTT digunakan sebagai penghubung antara *server Blynk* dan *Google Assistant*, Pada *Platform IFTTT* dapat diatur perintah dan respon yang pada *Google Assistant*, perintah yang di atur terlihat pada halaman utama IFTTT



■ **Gambar 10.** Tampilan pada IFTTT

Perintah yang sudah dibuat pada IFTTT dapat Diakses melalui *Google Assitant*, dan direspon sesuai dengan yang sudah diatur pada *platform IFTTT*.



■ **Gambar 11.** Tampilan Perintah suara pada aplikasi *Google Assistant*

Perintah suara pada *Google Assistant* dapat diatur dengan sedemikian rupa sehingga mudah dilafalkan walaupun menggunakan bahasa Inggris dalam pengucapannya. Perintah suara pada IFTTT hanya bisa dalam bahasa Inggris sedangkan untuk respon dari perintah suara bisa bahasa Indonesia maupun bahasa Inggris.

■ **Tabel 4.** Perintah suara dan respon suara *Google Assistant*

Perintah suara	Respon suara	Perintah peralatan
Turn on lamp one	Lamp 1 on	Lampu 1 menyala
Turn off lamp one	Lamp 1 off	Lampu 1 mati
Turn on lamp two	Lamp 2 on	Lampu 2 menyala
Turn off lamp two	Lamp 2 off	Lampu 2 mati
Turn on the terminal	Terminal on	Terminal menyala
Turn off the terminal	Terminal on	Terminal mati
Turn on all light	All lamp on	Semua lampu menyala
Turn off all light	All lamp off	Semua lampu mati
Turn on auto lamp	Auto lamp on	Lampu otomatis diaktifkan
Turn off auto lamp	Auto lamp off	Lampu otomatis dimatikan

C. Analisa dan Pengujian

Pengujian penelitian ini dibagi menjadi lima pengujian yaitu, pengujian perintah suara dari *Google Assistant*, pengujian Wifi, pengujian penggunaan Aplikasi *Blynk*, pengujian RTC, dan pengujian program penghidupan lampu otomatis.

1. Pengujian respon Perintah suara dari *Google Assistant*



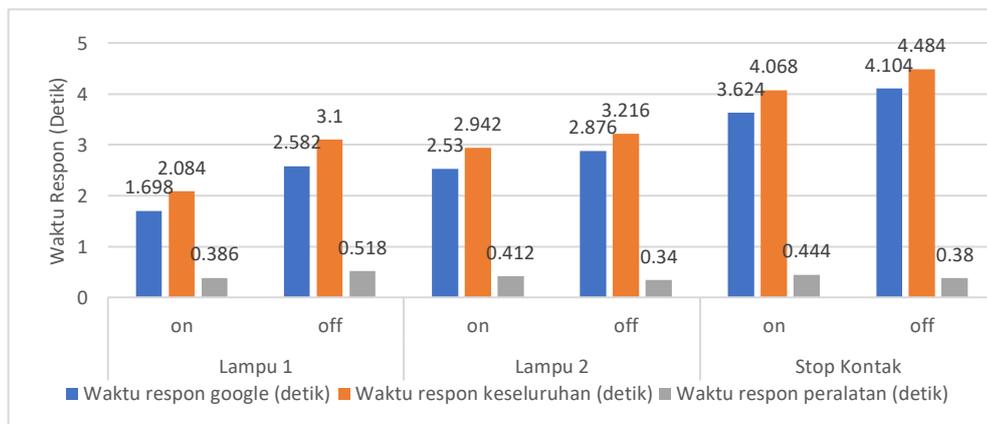
■ **Gambar 12.** Pengujian respon Perintah suara dari *Google Assistant*

Tujuan dari pengujian respon Perintah suara dari *Google Assistant* untuk melihat fungsionalitas pada sistem dalam proses input perintah suara pada *Google Assistant* di Smartphone Android yang kemudian perintah suara tersebut dikirim ke *Server Blynk* dan diteruskan ke *NodeMcu*.

■ **Tabel 5.** Hasil Pengujian respon Perintah suara dari *Google Assistant*

Peralatan yang dikendalikan	posisi	hasil	Waktu respon <i>Google</i> (detik)					Waktu respon keseluruhan (detik)					Waktu respon peralatan (detik)				
			1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Lampu 1	on	OK	1,85	1,71	1,71	0,67	2,55	2,36	2,09	2,18	0,95	2,84	0,51	0,38	0,47	0,28	0,29
	off	OK	1,71	2,03	1,96	3,75	3,46	2,10	2,69	2,84	4,21	3,66	0,39	0,66	0,88	0,46	0,20
Lampu 2	on	OK	2,82	1,86	3,00	2,81	2,16	3,11	2,08	3,22	3,70	2,60	0,29	0,22	0,22	0,89	0,44
	off	OK	2,11	3,50	3,13	2,29	3,35	2,34	4,12	3,42	2,65	3,55	0,23	0,62	0,29	0,36	0,2
Stop Kontak	on	OK	4,19	4,92	2,95	4,52	1,54	4,62	5,15	3,65	4,82	2,10	0,43	0,23	0,7	0,3	0,56
	off	OK	4,85	2,55	5,07	3,67	4,38	5,14	2,97	5,40	4,16	4,75	0,29	0,42	0,33	0,49	0,37

Data pengujian diatas menunjukkan bahwa waktu respon keseluruhan *Google Assistant* dapat mengirim perintah dengan paling lama 5,14 detik, dan paling cepat 0,95 detik. Tabel diatas menunjukkan waktu respon dari *Google* untuk mengetahui perintah yang diucapkan paling lama 5,05 detik dan paling cepat 0,67 detik. Data perintah dari *Google* dikirim ke perangkat melalui *server Blynk* dengan waktu respon rata rata 0,412 detik.



■ **Gambar 13.** Grafik Pengujian respon Perintah suara dari *Google Assistant*

Pengujian menunjukkan bahwa perintah suara dari *Google Assistant* yang dibuat melalui *Platform IFTTT* dapat bekerja dengan cepat dan akurat. variasi respon keseluruhan sangat dipengaruhi dari kecepatan respon *Google* dalam mengetahui perintah apa yang diucapkan. Lambatnya *Google* dalam mengetahui perintah yang diucapkan salah satunya dapat dipengaruhi oleh cara pengucapan perintah yang kurang pas.

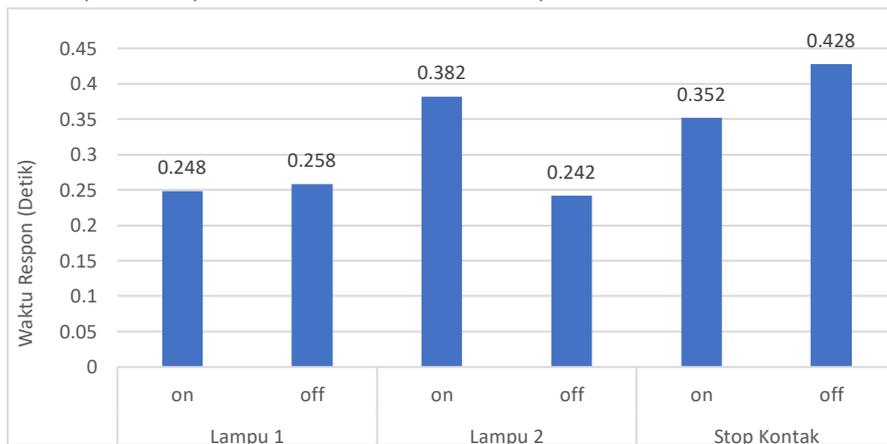
2. Pengujian respon kendali melalui Aplikasi *Blynk*

Tujuan dari pengujian respon kendali melalui Aplikasi *Blynk* untuk melihat fungsionalitas pada sistem dalam proses input kendali pada *Interface* pada Aplikasi *Blynk* yang yang disinkronkan dengan *NodeMcu*. Keberhasilan pengujian diberi tanda “OK” sedangkan pengujian yang tidak berhasil diberi “Gagal”, pengujian dilakukan sebanyak 5 kali.

■ **Tabel 6.** Pengujian respon kendali melalui Aplikasi *Blynk*

Peralatan yang dikendalikan	posisi	hasil	Waktu respon peralatan (detik)					Rata-rata
			1	2	3	4	5	
Lampu 1	on	OK	0,27	0,21	0,28	0,28	0,20	0,248
	off	OK	0,20	0,27	0,27	0,21	0,34	0,258
Lampu 2	on	OK	0,86	0,26	0,20	0,27	0,32	0,382
	off	OK	0,32	0,21	0,27	0,21	0,20	0,242
Stop Kontak	on	OK	0,40	0,33	0,20	0,40	0,43	0,352
	off	OK	0,33	0,27	0,47	0,77	0,30	0,428

Pengujian waktu respon dilakukan sebanyak 5 kali. Dapat dilihat bahwa Aplikasi *Blynk* dapat mengirim perintah menghidupkan lampu 1 dengan kecepatan rata-rata 0,284 detik, sedangkan untuk memaatkan nya membutuhkan waktu rata rata 0,258 detik. Lampu 2 juga mempunyai waktu respon yang sangat cepat yaitu rata-rata 0,382 detik untuk menghidupkan, dan 0,242 detik untuk mematikan. Waktu respon stop kontak saat dihidupkan rata-rata 0,352 detik, dan saat dimatikan rata-rata 0,428 detik.



■ **Gambar 14.** Grafik pengujian respon Perintah suara dari aplikasi *Blynk*

Dari gambar 14 dapat dilihat bahwa waktu respon rata rata berkisar 0,248 detik hingga 0,428 detik. Dari pengujian diatas dapat disimpulkan bahwa bahwa kendali dari aplikasi *Blynk* dapat bekerja dengan cepat dan akurat, dengan waktu respon yang sangat kecil dan cukup stabil.

3. Pengujian Jaringan Wifi

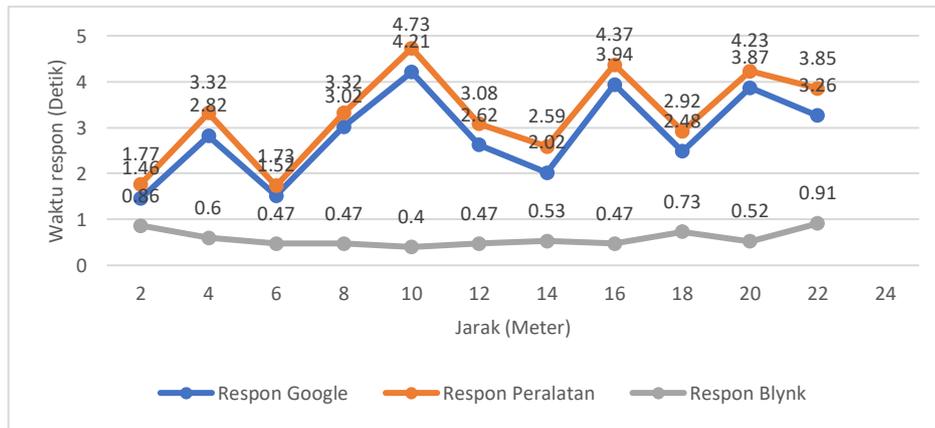
Tujuan dari pengujian Jaringan Wifi ini untuk melihat seberapa jauh jarak maksimal peralatan yang sudah dibuat dari pemancar jaringan Wifi agar dapat berfungsi dengan baik, dalam hal ini pengujian dilakukan dengan menggunakan koneksi Wifi dari *Smartphone*, dengan *IP Adres* lokal 192.168.43.102, dan *IP Address* publik 114.125.251.171. Pengujian ini dilakukan dengan kondisi *Line Of Sight* (LOS), yaitu antara pemancar jaringan wifi dan peralatan ini dalam posisi satu garis tanpa terhalang benda apapun. Pengujian dilakukan dengan mengirim satu perintah kepada peralatan dengan aplikasi *Blynk* maupun perintah suara *Google Assitant*. Keberhasilan pengujian diberi tanda “OK” sedangkan pengujian yang tidak berhasil diberi “Gagal”.

■ **Tabel 7.** Pengujian Jaringan Wifi

Jarak	hasil	Waktu respon <i>Google Assistan</i>		Waktu respon aplikasi <i>Blynk</i>
		Respon <i>Google</i>	Respon keseluruhan	
2 Meter	OK	1,46	1,77	0,86
4 Meter	OK	2,82	3,32	0,60
6 Meter	OK	1,52	1,73	0,47
8 Meter	OK	3,02	3,32	0,47
10 Meter	OK	4,21	4,73	0,40
12 Meter	OK	2,62	3,08	0,47
14 Meter	OK	2,02	2,59	0,53
16 Meter	OK	3,94	4,37	0,47

18 Meter	OK	2,48	2,92	0,73
20 Meter	OK	3,87	4,23	0,52
22 Meter	OK	3,26	3,85	0,91
24 Meter	Gagal	-	-	-

Pada hasil pengujian dapat dilihat bahwa peralatan hasil perancangan ini dapat bekerja dengan maksimal radius maksimal 22 meter dari sumber wifi. Dari hasil pengujian juga dapat dilihat bahwa jarak radius penerimaan wifi tidak terlalu berpengaruh terhadap waktu respon peralatan.



■ Gambar 15. Grafik Pengujian Jaringan Wifi

Dari gambar 15 dapat dilihat bahwa jarak tidak terlalu mempengaruhi lamanya waktu respon ke peralatan, tetapi jarak yang terlalu jauh dari sumber wifi akan memutuskan koneksi ke jaringan, sehingga peralatan smarthome tidak dapat terhubung ke Smartphone melalui jaringan Internet.

4. Pengujian RTC

Tujuan dari pengujian untuk mengetahui keakuratan dari RTC yang ada dalam peralatan yang dibuat, faktor keberhasilan dari pengujian ini adalah seberapa jauh perbedaan antara waktu RTC yang ada di peralatan dan waktu yang sebenarnya.

■ Tabel 8. Pengujian RTC

Waktu RTC	Waktu sebenarnya	Selisih
13:36:47	13:36:48	1 Detik
13:36:57	13:36:58	1 Detik
13:37:07	13:37:08	1 Detik
09:35:51	09:35:52	1 Detik
09:36:01	09:36:02	1 Detik
09:36:11	09:36:12	1 Detik

Pada tabel hasil pengujian dapat dilihat bahwa RTC yang terdapat pada peralatan dapat bekerja dengan baik dengan selisih waktu dari waktu sebenarnya sebesar 1 detik. Selisih waktu 1 detik tersebut dapat diabaikan dan tidak mempengaruhi performansi pada peralatan *Smarthome* dalam penelitian ini, karena RTC hanya sebagai referensi waktu yang dipakai untuk peralatan agar dapat berjalan sebagaimana semestinya.

5. Pengujian program lampu otomatis

Tujuan dari pengujian ini adalah untuk melihat keoptimalan program lampu otomatis yang sudah ada pada peralatan *Smarthome* pada penelitian ini. Faktor keberhasilan dari pengujian ini adalah ketepatan waktu penghidupan lampu otomatis dengan waktu yang sudah ditentukan pada program. Pada program lampu otomatis, semua lampu akan menyala pada jam 17.00 yaitu pada sore hari, dan semua lampu akan mati pada jam 6.00 pada keesokan paginya.

■ Tabel 9. Pengujian program lampu otomatis

Program yang berjalan	Waktu program	Waktu saat program berjalan
-----------------------	---------------	-----------------------------

Lampu hidup otomatis	17.00	17.01
Lampu mati otomatis	06.00	06.01

Pada hasil pengujian dapat dilihat bahwa waktu program lampu otomatis saat lampu dihidupkan otomatis maupun saat dimatikan otomatis berjalan berbeda 1 detik dari waktu yang sudah ditentukan. Perbedaan waktu 1 detik ini dikarenakan keterlambatan 1 detik pada RTC pada peralatan yang sudah diuji sebelumnya. Keterlambatan waktu yang sama antara program lampu otomatis dan RTC dikarenakan program lampu otomatis ini mengambil waktu dari RTC sebagai referensi untuk menghidupkan maupun mematikan semua lampu.

KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah

1. *Android device* dengan *Google Assistant* dapat digunakan untuk pengolahan perintah suara untuk nantinya diproses menjadi perintah untuk mengendalikan peralatan listrik. Berdasarkan pengujian yang dilakukan, perintah suara dalam proses penerimaan suara oleh *Google Assistant*, dapat berjalan dengan baik selama perintah suara diucapkan dengan baik dan benar dalam bahasa Inggris. hasil pengujian fungsional dinyatakan berhasil dengan baik.
2. Berdasarkan pengujian respon aplikasi *Blynk* yang dilakukan, hasil pengujian fungsional dinyatakan berhasil dengan tingkat keberhasilan 100%, dengan waktu jeda rata-rata kurang dari satu detik
3. Berdasarkan pengujian Jaringan Wifi yang dilakukan, dengan *IP Address* lokal 192.168.43.102, dan *IP Address* publik 114.125.251.171. Hasil pengujian fungsional dinyatakan berhasil dengan kondisi LOS, jarak terjauh peralatan dan wifi agar dapat berfungsi dengan baik yaitu 22 Meter.
4. Berdasarkan pengujian yang dilakukan pada RTC, RTC bekerja untuk memberikan informasi tentang waktu saat ini yang kemudian digunakan sebagai referensi waktu untuk menjalankan perintah penghidupan lampu otomatis. Hasil pengujian fungsional dinyatakan berhasil dengan selisih waktu RTC dan waktu sebenarnya sebesar 1 detik.

B. Saran

Untuk penggunaan perintah suara dipenelitian selanjutnya dapat menggunakan bahasa Indonesia karena dengan menggunakan bahasa inggris terdapat beberapa kekeliruan dalam penangkapan perintah yang diucapkan oleh *Google Assistant*.

DAFTAR PUSTAKA

- [1] F. C. I. L. Herin and . H. Pangaribuan, "VOICE CONTROL SEBAGAI PENGENDALI PERALATAN ELEKTRONIK BERBASIS NODEMCU," 2019.
- [2] H. A. Rochman, . R. Primananda and . H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," 2017.
- [3] M. K. A. Putra, S. . R. Akbar and G. E. Setyawan, "Perancangan Sistem Keamanan Pada Smart Home Menggunakan Voice Command Dengan Konektivitas Bluetooth," 2018.
- [4] A. R. Azka, E. D. Marindani and R. . D. Nyoto, "Rancang Bangun Sistem Pengendali Smarthome menggunakan Mikrokontroler dengan Speech Command pada Smarthome Android," 2018.
- [5] A. D. Pangestu, . F. Ardianto and B. Alfaresi, "SISTEM MONITORING BEBAN LISTRIK BERBASIS ARDUINO NODEMCU ESP8266," 2019.
- [6] Arafat, "SISTEM PENGAMANAN PINTU RUMAH BERBASIS Internet Of Things (IoT) Dengan ESP8266," 2016.
- [7] A. Setiawan, W. W. Mustika and T. . B. Adji, "PERANCANGAN CONTEXT-AWARE SMART HOME DENGAN MENGGUNAKAN INTERNET OF THING," 2016.
- [8] A. Satriadi, Y. Christiyono and W. , "PERANCANGAN HOME AUTOMATION BERBASIS NodeMCU," 2019.
- [9] M. A. Ashari and L. Lidyawati, "IOT BERBASIS SISTEM SMART HOME MENGGUNAKAN NODEMCU V3," 2019.

- [10] E. Hesti and Adewasti, "APLIKASI ANDROID SEBAGAI PENGONTROL JARAK JAUH SMARTHOME DENGAN KONEKSI JARINGAN INTERNET," 2018.