3rd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
30 Agustus 2023 – Jakarta, Indonesia
Volume 2, Nomor 2, September 2023 - ISSN 2962-8628 (*online*)

IMPLEMENTASI TEKNOLOGI *INTERNET OF THINGS* PADA PROTOTIPE RUMAH PINTAR BERBASIS APLIKASI ANDROID

Abdul Rosyid[1*], Gunawan Pria Utama[2]

[1,2]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: [1*]rosyid.215.ar@gmail.com, [2]gunawan.priautama@budiluhur.ac.id
(* : *corresponding author*)

Dalam era teknologi yang maju seperti saat ini, masih banyak orang yang harus mengendalikan perangkat rumah elektronik secara manual dengan menyentuh tombol atau sakelar yang terhubung ke aliran listrik. Pengaturan manual ini dianggap kurang memberikan kenyamanan bagi sebagian orang, terutama bagi orang yang bekerja dengan mobilitas tinggi dan hidup sendiri karena perangkat elektronik tersebut tidak bisa dikendalikan tanpa menyentuh sakelar atau tombol. Untuk mengatasi masalah ini, hadir solusi *smart home* atau rumah pintar dari teknologi *Internet of Things* (IoT). Rumah pintar memungkinkan penghuni rumah untuk mengendalikan perangkat rumah elektronik dengan mudah, baik secara jarak jauh maupun berdasarkan jadwal yang telah ditentukan menggunakan fitur *timer*. Dengan menggunakan metode *prototyping*, dalam penelitian ini peneliti membuat sebuah prototipe Rumah Pintar berbentuk kamar studio dengan teras berdasarkan kondisi salah satu unit indekos di kawasan Ciledug, Tangerang. Rancangan prototipe ini menggunakan NodeMCU ESP8266, *relay* 4 *channel*, dan sensor DHT11. Manfaat penelitian ini pengguna dapat mengendalikan perangkat lampu dan kipas secara jarak jauh maupun sesuai jadwal yang ditentukan melalui aplikasi Android. Selain itu, pengguna dapat memantau kondisi suhu dan kelembapan udara dalam kamar secara *real-time* dengan integrasi sensor DHT11. Hasil dari penelitian ini menunjukan pengujian *delay* nyala atau mati lampu dan kipas berkisar sekitar 2-6 detik, menggunakan fitur *timer* berkisarsekitar 3-13 detik, dan menghapus kode blok pada NodeMCU ESP8266 menjadi sekitar 0-2 detik.

Kata Kunci: *internet of things*, rumah pintar, NodeMCU ESP8266, sensor DHT11, aplikasi Android

IMPLEMENTATION OF INTERNET OF THINGS TECHNOLOGY WITH ANDROID-BASED APP ON SMART HOME PROTOTYPE

In today's technologically advanced age, there are still many people who have to manually control electronic home appliances by touching buttons or switches that are connected to electricity. This manual setting is considered inconvenient for some people, especially those with high mobility and those who live alone, because these electronic devices cannot be controlled without touching the switch or button. To solve this problem, there is a smart home solution from the Internet of Things (IoT) technology. Smart homes allow homeowners to easily control home electronic devices, either remotely or based on a predetermined schedule using the timer function. In this study, with prototyping method, researchers created a smart home prototype based on the conditions of an apartment studio in Ciledug, Tangerang. This prototype design uses NodeMCU ESP8266, 4 channel relay and DHT11 sensor. From the results of this study, the benefits are that users can control the lights and a fan remotely or according to a predetermined schedule applied through the Android application. Users can also monitor the temperature and humidity conditions in the room in real-time by integrating the DHT11 sensor. The results of this study show that testing the delay of turning on or off the fan and lights ranges from about 2-6 seconds. When using the timer feature, the delay ranges from about 3-13 seconds, and when the block code on the NodeMCU ESP8266 removed the delay ranges from about 0-2 seconds.

Keywords*: internet of things, smart home, NodeMCU ESP8266, DHT11 sensor, Android app*

1. PENDAHULUAN

Dalam era teknologi yang maju seperti saat ini, perangkat rumah elektronik memainkan peran yang sangat penting dalam kehidupan sehari-hari. Namun masih banyak orang yang harus mengendalikan perangkat rumah elektronik secara manual dengan menyentuh tombol atau sakelar yang terhubung ke aliran listrik. Bagi sebagian orang, pengaturan manual ini dianggap kurang memberikan kenyamanan, terutama bagi orang yang bekerja dengan mobilitas tinggi dan hidup sendiri karena perangkat elektronik tersebut tidak bisa dikendalikan tanpa menyentuh sakelar atau tombol. Keadaan ini mengakibatkan penggunaan perangkat listrik yang tidak optimal karena tingkat mobilitas tinggi membuat orang yang hidup sendiri cenderung tidak memiliki cukup waktu untuk mengatur perangkat rumah elektronik, seperti lampu dan kipas, yang membuat perangkat-perangkat tersebut tetap

menyala meskipun tidak sedang digunakan. Selain itu, orang dengan mobilitas terbatas seperti penyandang disabilitas juga memiliki kemungkinan untuk sulit mengendalikan lampu dan kipas yang sakelarnya jauh dari jangkauan.

Untuk mengatasi masalah ini, hadir sebuah solusi menarik dari teknologi *Internet of Things* (IoT) yaitu *smart home* atau rumah pintar. Rumah pintar adalah hunian yang dilengkapi dengan teknologi pintar yang bertujuan untuk menyediakan layanan yang disesuaikan untuk pengguna. Teknologi rumah pintar memungkinkan untuk memantau, mengontrol, dan mendukung kebutuhan penghuni rumah, yang dapat meningkatkan kualitas hidup dan mendorong kehidupan yang mandiri [1]. Manfaat dari konektivitas perangkat pintar dengan internet adalah untuk memudahkan pemantauan dan pengaturan keadaan perangkat tersebut [2]. *Internet of Things* (IoT) adalah sebuah paradigma komunikasi terbaru yang dibayangkan sebagai masa depan, teknologi ini menambahkan perangkat sehari-hari dengan mikrokontroler, perangkat ini memungkinkan komunikasi digital pengirim-penerima dan terhubung dengan protokol komunikasi yang memungkinkan setiap perangkat terhubung dengan perangkat lainnya [3]. Teknologi rumah pintar memungkinkan untuk memantau, mengontrol, dan mendukung kebutuhan penghuni rumah, yang dapat meningkatkan kualitas hidup dan mendorong kehidupan yang mandiri [1]. Manfaat dari konektivitas perangkat pintar dengan internet adalah untuk memudahkan pemantauan dan pengaturan keadaan perangkat [2].

Penelitian tentang rumah pintar pernah dilakukan oleh Bagus Eryawan, Ari Endang Jayati, dan Sri Heranurweni [4], penelitian ini menggunakan *Raspberry Pi* 3 model B sebagai mikrokontoler. Hasil dari penelitian ini adalah sebuah prototipe berbasis web untuk mengatur lampu, mengatur motor *stepper* untuk pintu garasi, mengatur motor *servo* untuk kunci pintu, dan mengatur motor *brushless* (kipas angin). Penelitian lain juga pernah dilakukan oleh Lila Setiyani, Karya Suhada, dan Yulindawati [5], penelitian ini menggunakan *Raspberry Pi* 3 model B sebagai mikrokontoler dan sensor api/*flame* sensor. Hasil dari penelitian ini adalah sebuah prototipe berbasis aplikasi Android untuk mengatur lampu dan mendeteksi kebakaran. Penelitian lainnya pernah dilakukan oleh Nurul Hidayati Lusita Dewi, Mimin F. Rohmah, dan Soffa Zahara [6], penelitian ini menggunakan NodeMCU ESP8266 sebagai mikrokontroler dan sensor LM35, sensor PIR, serta sensor MQ2. Hasil dari penelitian ini adalah sebuah prototipe menggunakan aplikasi Blynk untuk mengatur lampu dan kipas, memantau suhu ruang, pendeteksi gerakan, dan pendeteksi kebocoran gas. Dari penelitian terdahulu yang telah dipaparkan di atas, penelitian ini kali ini memiliki perbedaan yaitu menggunakan fitur *timer*. Fitur *timer* memungkinkan penghuni rumah dapat meninggalkan rumah tanpa mematikan lampu dan kipas secara manual karena perangkat akan mengikuti pengaturan yang telah diatur oleh pengguna untuk mengendalikan perangkat rumah elektronik.

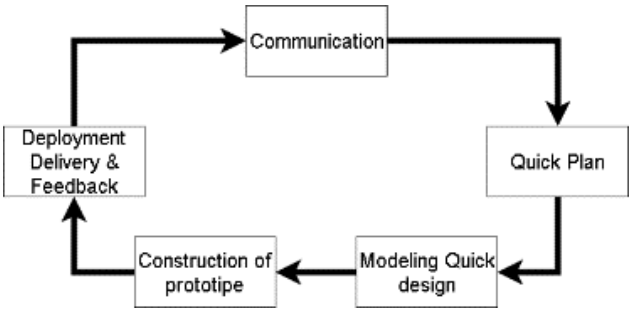
Penelitian ini berfokus pada pengembangan prototipe rumah pintar berbasis IoT dengan menggunakan NodeMCU ESP8266, *relay* 4 *channel*, dan sensor DHT11. Prototipe ini dirancang dalam bentuk kamar studio dengan teras sesuai dengan kondisi salah satu unit indekos di kawasan Ciledug, Tangerang. Pengendalian perangkat dilakukan melalui aplikasi Android. Manfaat penelitian ini adalah pengguna dapat mengendalikan perangkat rumah elektronik secara jarak jauh maupun berdasarkan jadwal yang telah ditentukan menggunakan fitur *timer*. Dengan tambahan integrasi sensor DHT11, pengguna juga dapat memantau kondisi suhu dan kelembapan dalam kamar secara *real-time*. Penelitian ini bertujuan untuk memanfaatkan potensi teknologi IoT secara optimal dengan merancang prototipe rumah pintar berbasis aplikasi Android untuk memberikan kemudahan dan kenyamanan yangsignifikan bagi penghuni rumah.

Dalam penelitian ini, diperlukan beberapa komponen untuk dirangkai menjadi sebuah prototipe. NodeMCU ESP8266 adalah platform sumber terbuka (*open source*), desain perangkat kerasnya terbuka untuk dimodifikasi. NodeMCU *Dev Kit*/*board* terdiri dari *chip* berkemampuan *WiFi* ESP8266 [7]. NodeMCU pada dasarnya adalah pengembangan dari ESP8266 dengan *firmware* berbasis e-Lua. Pada NodeMCU dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun sebagai *power supply* untuk menyalakan NodeMCU. Selain dengan Bahasa Lua, NodeMCU juga *support* dengan *software* Arduino IDE dengan melakukan sedikit pengubahan pada *board manager* pada Arduino IDE [8]. *Relay* adalah sakelar elektrik yang menggunakan elektromagnet untuk memindahkan sakelar dari posisi *off* ke posisi *on* [8]. *Relay* pada Arduino dasarnya mempunyai 3 buah *input* yang masing–masing berfungsi sebagai kontrol untuk menghidupkan *Relay*, pin tersebut adalah GND, VCC dan IN. GND dihubungkan ke Arduino yang bernilai *Ground* atau tegangan 0-*volt* (-), VCC untuk tegangan positif (+) yang bernilai +5-*volt* dan IN dapat dihubungkan langsung ke sensor untuk dikontrol melalui *Relay* [9]. Sensor DHT11 pada dasarnya memiliki pengaturan suhu dan kelembapan udara yang terkalibrasi dan dengan keluaran sinyal digital. Sensor DHT11 memiliki resolusi temperatur sebesar 8 bit, dengan akurasi minimum ±10°C dan akurasi maksimum ±20°C dan rentang pengukuran suhu dari 00°C sampai dengan 500°C [10]. Kabel *jumper* adalah kabel elektrik penghubung antar komponen tanpa memerlukan *solder*. Kabel *jumper* pada umumnya memiliki pin konektor pada setiap ujung sisinya [11]. Jenis kabel *jumper* yaitu *Male to Female*, *Male to Male*, dan *Female to Female*. Lampu yang digunakan untuk penelitian ini merupakan lampu LED *(Light Emitting Diode).* Cara kerja LED adalah menghasilkan cahaya dengan mengubah energi listrik menjadi energi cahaya (*transduser*). Dengan cara kerja ini, maka lampu LED dapat langsung memancarkan cahaya secara maksimal, tidak memerlukan waktu pemanasan seperti bohlam, dan juga tidak menimbulkan panas seperti pada bohlam ataupun neon [12].

Kipas DC 12Vdigunakan untuk prototipe penelitian ini karena ukurannya kecil mudah dibawa untuk kebutuhan presentasi dalam sidang.

Metode yang digunakan dalam penelitian ini adalah metode prototipe. Metode prototipe merupakan metode proses pembuatan sistem yang dibuat secara terstruktur serta memiliki beberapa tahap yang harus dilalui pada pembuatannya, namun jika pada tahap final dinyatakan bahwa sistem yang telah dibuat masih memiliki kekurangan, maka sistem akan di evaluasi kembali dan akan melalui proses dari awal. Berikut merupakan langkah-langkah atau tahapan dalam metode prototipe [11] beserta dengan ilustrasi diagram gambar 1 di bawah:

a. *Communication* atau komunikasi dan pengumpulan data awal, yaitu analisis terhadap kebutuhan pengguna
b. *Quick plan*, yaitu tahapan perencanaan kebutuhan
c. *Modelling Quick Design*, yaitu tahapan pembuatan desain
d. Pembentukan prototipe, yaitu pembuatan perangkat prototipe termasuk pengujian



Gambar 1. Diagram Alur Metode Prototipe

Berdasarkan diagram gambar 1 di atas, prototipe harus dikembangkan berdasarkan kebutuhan yang diperoleh dari pengguna. Prototipe kemudian dibuat berdasarkan berbagai data yang telah diperoleh. Prototipe yang telah dibuat selanjutnya diuji dan dipresentasikan kepada pengguna. Prototipe yang dibuat harus berfokus pada representasi aspek-aspek aplikasi yang akan terlihat oleh pengguna. Dengan kata lain, pengguna dapat mengharapkan program atau aplikasi yang nantinya digunakan sesuai dengan prototipe yang telah dirancang. Dalam metode *prototyping* terdapat beberapa keunggulan, yaitu [13]:

a. Pengembang sistem dan pengguna saling berkomunikasi khususnya dalam hal penyamaan persepsi terhadap pemodelan sistem yang akan menjadi dasar pengembangan sistem operasionalnya
b. Pengguna ikut terlibat secara aktif dan berpartisipasi dalam menentukan model sistem dan sistem operasionalnya sehingga pelanggan/pengguna akan puas karena sistem yang dibuat sesuai dengan keinginan dan harapannya
c. Sistem yang dibangun memiliki kualitas yang diinginkan karena sesuai dengan kebutuhan yang ada

2. METODE PENELITIAN

2.1 Metode *Prototyping*

Berdasarkan metode *prototyping* yang sudah dijelaskan pada bab pendahuluan, tahapan yang pertama dilakukan dalam penelitian ini adalah melakukan pengumpulan kebutuhan dari calon pengguna aplikasi. Pada tahapan ini diketahui bahwa calon pengguna aplikasi membutuhkan sebuah aplikasi yang dapat mengontrol lampu dan kipas serta memantau suhu dari jarak jauh. Setelah memperoleh kebutuhan calon pengguna aplikasi, kemudian rancangan dasar dibuat dalam bentuk aplikasi Android. Setelah aplikasi Android dibuat, kemudian dilakukan pengujian terhadap aplikasi Android dengan mengubungkan aplikasi ke rangkaian komponen yang terdiri dari mikrokontroler, *relay*, serta sensor menggunakan Firebase. Setelah rangkaian komponen dan aplikasi Android terhubung, kemudian semua komponen dirangkai ke sebuah maket yang dibangun sesuai dengan kondisi tempat tinggal calon pengguna aplikasi. Rangkaian yang telah dibangun kemudian dipresentasikan ke calon pengguna aplikasi.

2.2 Rancangan Pengujian

Rancangan pengujian yang dilakukan adalah dengan membuat sebuah prototipe yang dapat terhubung dengan aplikasi Android. Pada aplikasi ini terdapat beberapa *button* yang memiliki berbagai fungsi perintah yang apabila *button* ini di tekan maka menghasilkan *output* yang sudah diprogram sebelumnya. Pengujian ini akan berjalan ketika adanya perintah yang diberikan melalui aplikasi Android.

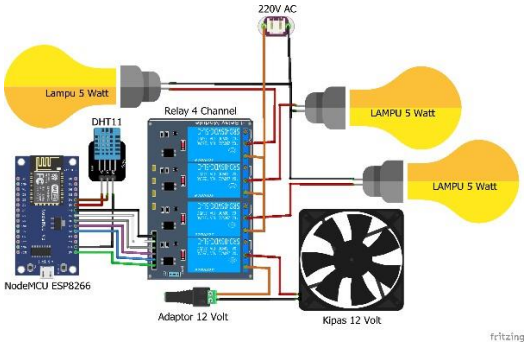
2.3 Rancangan Prototipe

Pada rancangan prototipe, peneliti akan mejelaskan alat atau komponen apa saja yang digunakan pada tahapan rancangan prototipe. Berikut tabel 1 daftar komponen yang digunakan.

Tabel 1. Komponen yang Digunakan

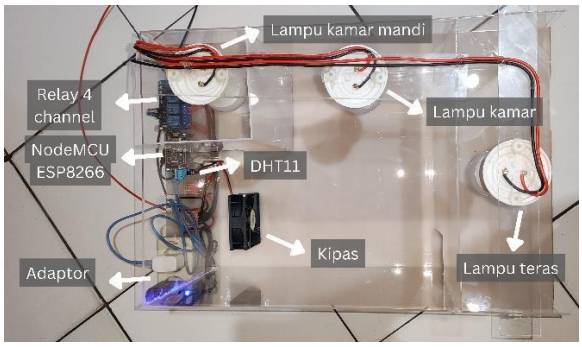
No.	Komponen	Kegunaan
1.	NodeMCU ESP8266	Berfungsi sebagai mikrokontroler
2.	*Relay*	Berfungsi sebagai penghubung dan pemutus arus
3.	Sensor DHT11	Digunakan sebagai pendeteksi suhu dan kelembapan ruangan
4.	Kabel *Jumper*	Digunakan sebagai pengubung antar komponen
5.	Lampu	Digunakan sebagai penerangan ruangan
6.	Kipas DC 12V	Digunakan sebagai pendingin ruangan

Komponen di atas kemudian dirangkai menjadi sebuah desain prototipe rumah pintar berbasis *Internet of Things* seperti gambar 2 di bawah ini.



Gambar 2. Desain Semua Perangkat

Desain prototipe pada gambar 2 di atas kemudian diaplikasikan menjadi sebuah maket dengan komponen yang telah dipasang seperti pada gambar 3 di bawah ini.



Gambar 3. Maket Prototipe Rumah Pintar

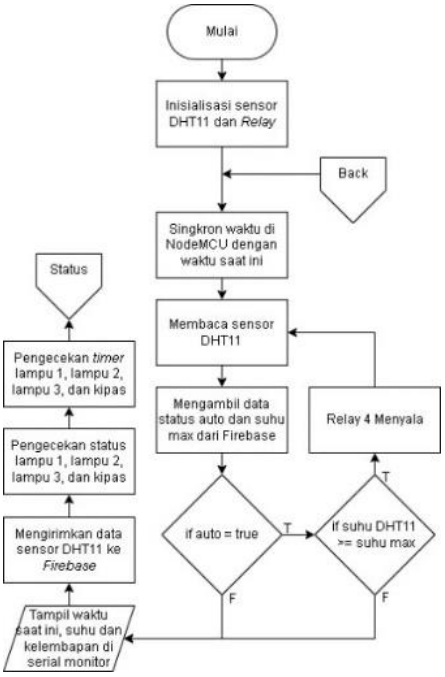
3. HASIL DAN PEMBAHASAN

3.1 Implementasi Metode

Pada pembuatan prototipe rumah pintar ini. Penulis menggunakan metode *prototyping*. Dimana metode *prototyping* ini memiliki beberapa tahapan untuk megguakan metode ini agar dapat berjalan sesuai dengan keinginan pengguna. Setelah melewati tahapan tersebut, untuk menjalankan perintah pengguna harus menggunakan aplikasi Android yang sudah dibuat dan terkoneksi dengan NodeMCU ESP8266, dan akan menghasilkan *output* sesuai dengan keinginan pengguna.

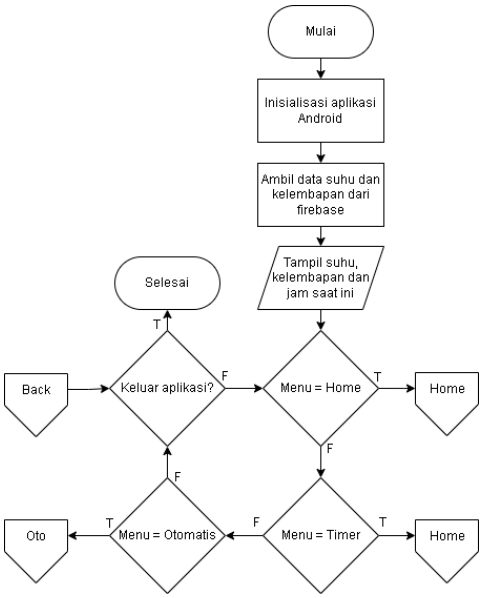
3.2 *Flowchart*

Flowchart ini menjelaskan alur kerja dari NodeMCU ESP8266. Gambar 4 di bawah merupakan *flowchart* NodeMCU ESP8266.



Gambar 4. *Flowchart* NodeMCU ESP8266

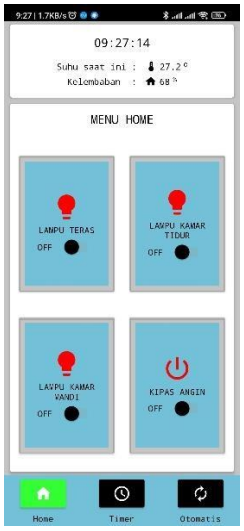
Sedangkan *flowchart* di bawah ini menjelaskan alur kerja dari aplikasi Android. Gambar 5 di bawah merupakan *flowchart* aplikasi Android.



Gambar 5. *Flowchart* Aplikasi Android

3.3 Tampilan Aplikasi Android

Tampilan aplikasi Andorid terdapat menu *home*, *timer*, dan otomatis. Tampilan menu *home* dimana pengguna akan mengatur nyala/mati lampu/kipas secara manual. Tampilan menu *timer* dimana pengguna akan mengatur nyala/mati lampu/kipas secara otomatis dengan cara menekan tombol dan memasukkan waktu yang diinginkan. Tampilan menu otomatis dimana pengguna akan mengatur suhu *max* untuk mengatur nyala/mati kipas secara otomatis berdasarkan suhu dengan cara input batas *max* dan menekan *check box* untuk mengaktifkan fitur secara otomatis. Tampilan ketiga menu dapat dilihat pada gambar 6, 7, dan 8 di bawah.



Gambar 6. Tampilan Menu *Home*

Gambar 7. Tampilan Menu *Timer*

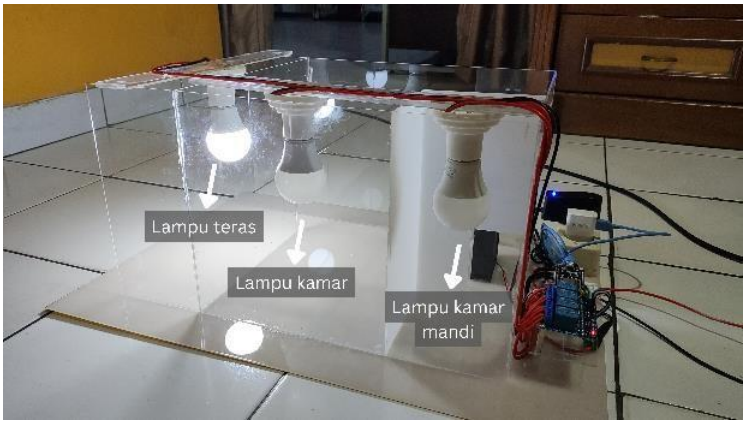
Gambar 8. Tampilan Menu Otomatis

3.4 Pengujian Alat

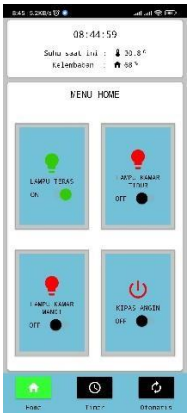
Pada pengujian ini akan melakukan pengujian terhadap sistem yang sudah dirangkai. Berikut adalah perincian pengujian yang dilakukan.

3.4.1 Pengujian Manual Lampu Teras

Pengujian dilakukan untuk lampu teras. Lampu dapat menyala apabila dikontrol melalui aplikasi Android. Jika *switch* lampu teras pada posisi *on* maka lampu teras akan menyala seperti gambar 9. Tampilan *monitoring* pada aplikasi Android ketika lampu teras menyala, seperti pada gambar 10.



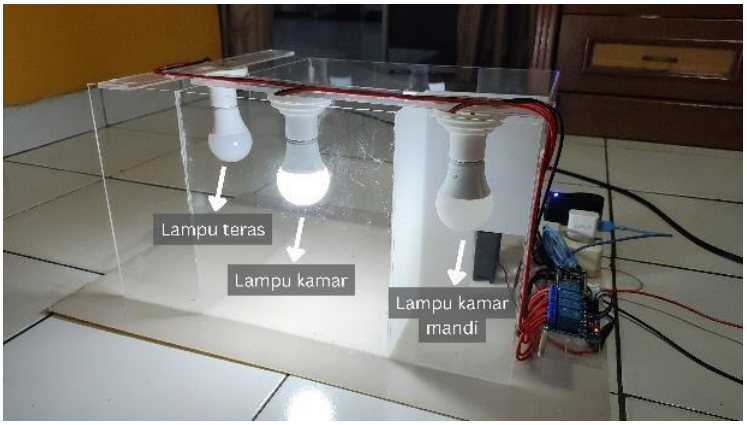
Gambar 9. Lampu Teras Nyala



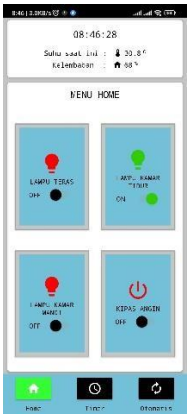
Gambar 10. Tampilan Aplikasi Android Lampu Teras *On*

3.4.2 Pengujian Manual Lampu Kamar

Pengujian dilakukan untuk lampu kamar. Lampu dapat menyala apabila dikontrol melalui aplikasi Android. Jika *switch* lampu kamar pada posisi *on* maka lampu kamar akan menyala seperti gambar 11. Tampilan *monitoring* pada aplikasi Android ketika lampu kamar menyala, seperti pada gambar 12.



Gambar 11. Lampu Kamar Nyala

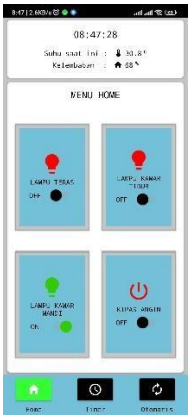


Gambar 12. Tampilan Aplikasi Android Lampu Kamar *On*

3.4.3 Pengujian Manual Lampu Kamar Mandi

Pengujian dilakukan untuk lampu kamar mandi. Lampu dapat menyala apabila dikontrol melalui aplikasi Android. Jika *switch* lampu kamar mandi pada posisi *on* maka lampu kamar mandi akan menyala seperti gambar 13. Tampilan *monitoring* pada aplikasi Android ketika lampu kamar mandi menyala, seperti pada gambar 14.

Gambar 13. Lampu Kamar Mandi Nyala

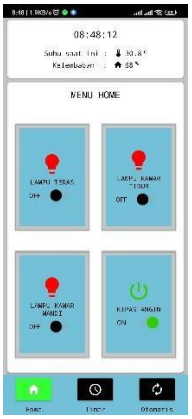


Gambar 14. Tampilan Aplikasi Android Lampu Kamar Mandi *On*

3.4.4 Pengujian Manual Kipas

Pengujian dilakukan untuk kipas. kipas dapat menyala apabila dikontrol melalui aplikasi Android. Jika *switch* kipas pada posisi *on* maka kipas akan menyala seperti gambar 15. Tampilan *monitoring* pada aplikasi Android ketika kipas menyala, seperti pada gambar 16.

Gambar 15. Kipas Nyala



Gambar 16. Tampilan Aplikasi Android Kipas *On*

3.4.5 Pengujian Otomatis Kipas

Pengujian dilakukan untuk mengetahui keberhasilan menyalakan kipas otomatis. Pada aplikasi Android diinput suhu max 31 sebagai batas suhu. Jika nilai suhu ruangan >= suhu max maka kipas akan menyala, jika nilai suhu ruangan < suhu max maka kipas akan mati secara otomatis. Pada gambar 17 menampilkan kondisi kipas saat nilai suhu ruangan >= suhu max kipas menyala secara otomatis. Pada gambar 18 menampilkan aplikasi Android saat nilai suhu >= suhu *max*.

Gambar 17. Kipas Nyala Otomatis

Gambar 18. Tampilan Aplikasi Android Suhu >= Suhu Max

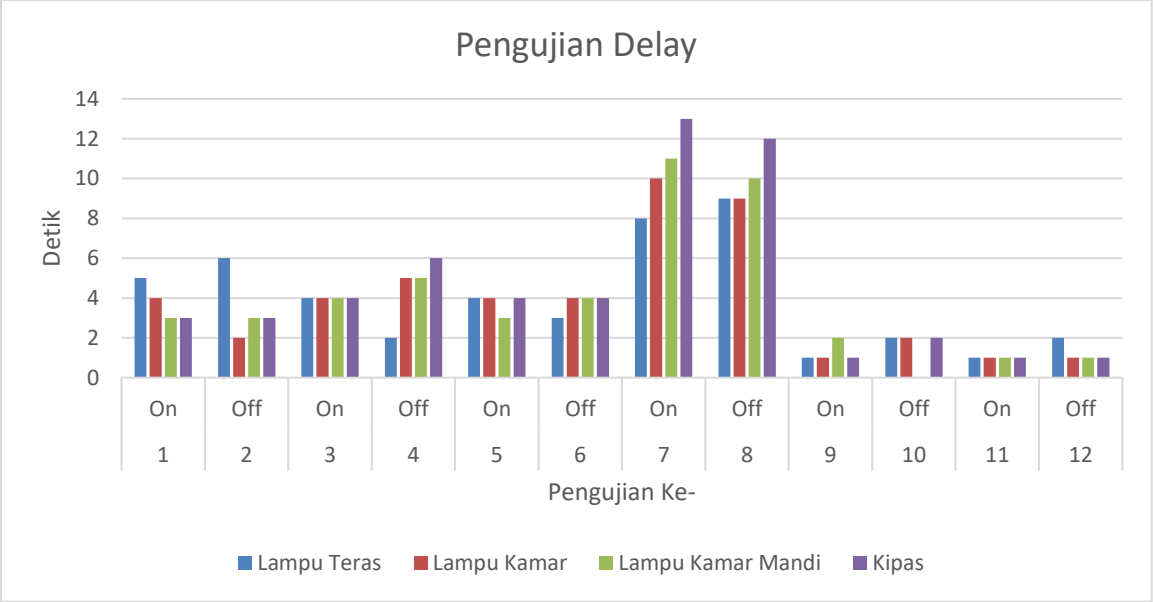
3.5 Hasil Pengujian

Pada tahap ini, pengujian beberapa kali dilakukan pada lampu dan kipas dengan fokus terhadap kecepatan respon dari perintah pada setiap perangkat. Hasil pengujian seperti pada tabel 2.

Tabel 2. Tabel Pengujian Delay

Pengujian ke-	Cara pengujian	Pengujian	*Delay* (Detik)			
			Lampu Teras	Lampu Kamar	Lampu Kamar Mandi	Kipas
1.	Manual	*On*	5	4	3	3
2.	Manual	*Off*	6	2	3	3
3.	Manual	*On*	4	4	4	4
4.	Manual	*Off*	2	5	5	6
5.	Timer (satu per satu)	*On*	4	4	3	4
6.	Timer (satu per satu)	*Off*	3	4	4	4
7.	Timer (semua alat bersamaan)	*On*	8	10	11	13
8.	Timer (semua alat bersamaan)	*Off*	9	9	10	12
9.	Fitur timer dihilangkan	*On*	1	1	2	1
10.	Fitur timer dihilangkan	*Off*	2	2	0	2
11.	Fitur timer dihilangkan	*On*	1	1	1	1
12.	Fitur timer dihilangkan	*Off*	2	1	1	1

Pengujian *delay* nyala atau mati lampu dan kipas berkisar sekitar 2-6 detik, menggunakan fitur *timer* berkisar sekitar 3-13 detik, dan menghapus kode blok pada NodeMCU ESP8266 menjadi sekitar 0-2 detik. Berikut ini pada gambar 19 merupakan visualisasi data pengujian *delay* lampu dan kipas dalam bentuk grafik batang.



Gambar 19. Grafik Visualisasi Pengujian *Delay*

4. KESIMPULAN

Berdasarkan pada perancangan, pembuatan, dan berbagai macam pengujian pada prototipe ini, maka dapat diambil kesimpulan yaitu aplikasi Android yang telah dibuat khusus untuk penelitian ini berhasil untuk mengendalikan lampu dan kipas secara jarak jauh maupun berdasarkan jadwal yang telah ditentukan. Namun adanya kode blok fitur *timer* membuat *delay* menjadi lebih besar karena terjadi proses yang lebih kompleks pada NodeMCU ESP8266 saat pengecekan kode blok fitur *timer*. Pengujian delay nyala atau mati lampu dan kipas berkisar sekitar 2-6 detik, menggunakan fitur timer berkisar sekitar 3-13 detik, dan menghapus kode blok pada NodeMCU ESP8266 menjadi sekitar 0-2 detik. Adapun saran yang dapat diajukan berdasarkan penelitian mengenai prototipe rumah pintar ini adalah menggunakan koneksi internet yang lebih stabil untuk mengontrol lampu dan kipas dalam upaya mengurangi waktu *delay*. Pengembangan lebih lanjut seperti menambahkan sensor dan alat pada rangkaian prototipe juga dapat dilakukan agar kegunaan NodeMCU ESP8266dapat dioptimalkann sesuai dengan kapasitas maksimalnya.

DAFTAR PUSTAKA

[1] D. Marikyan, S. Papagiannidis, and E. Alamanos, "A Systematic Review of The Smart Home Literature: A User Perspective," *Technol. Forecast. Soc. Change*, vol. 138, no. 2019, pp. 139–154, Jan. 2019, doi: https://doi.org/10.1016/j.techfore.2018.08.015.

[2] S. Kashan Ali Shah and W. Mahmood, "Smart Home Automation Using IOT and its Low Cost Implementation," *Int. J. Eng. Manuf.*, vol. 10, no. 5, pp. 28–36, Oct. 2020, doi: https://doi.org/10.5815/ijem.2020.05.03.

[3] A. Mayub, Fahmizal, M. Shidiq, U. Y. Oktiawati, and N. R. Rosyid, "Implementation Smart Home Using Internet Of Things," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 6, pp. 3126–3136, Dec. 2019, doi: https://doi.org/10.12928/TELKOMNIKA.v17i6.11722.

[4] B. Eryawan, A. E. Jayati, and S. Heranurweni, "Rancang Bangun Prototype Smart Home Dengan Konsep Internet of Things (Iot) Menggunakan Raspberry Pi Berbasis Web," *eLEKTRIKAL*, vol. 11, no. 2, pp. 1–5, 2019, doi: 10.26623/elektrika.v11i2.1691.

[5] L. Setiyani, K. Suhada, and Yulindawati, "Perancangan dan Implementasi IoT (Internet of Things) pada Smarthome Menggunakan Raspberry Pi Berbasis Android," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 2, pp. 459–466, 2019.

[6] N. H. L. Dewi, M. F. Rohmah, and Z. Soffa, "Prototype Smart Home Dengan Nodemcu Esp8266 Berbasis Iot," Skripsi, Fakultas Teknik, Universitas Islam Majapahit, Mojokerto, Indonesia, 2019. [Online]. Available: http://repository.unim.ac.id/265/2/JURNAL 5.14.04.11.0.097 NURUL HIDAYATI LUSITA DEWI.pdf

[7] Y. S. Parihar, "Internet of Things and Nodemcu: A review of use of Nodemcu ESP8266 in IoT products," *J. Emerg. Technol. Innov. Res.*, vol. 6, no. 6, pp. 1085–1088, 2019, doi: http://doi.one/10.1729/Journal.22685.

[8] M. Aluh and L. Lidyawati, "IoT Berbasis Sistem Smart Home Menggunakan NodeMCU V3," *J. Kaji. Tek. Elektro*, vol. 3, no. 2, pp. 138–149, 2019, doi: https://doi.org/10.52447/jkte.v3i2.1225.

[9] M. A. Putratama, "Perancangan Alat Pendeteksi Asap serta Kualitas Udara dalam Ruangan Menggunakan Sensor Mq-2 dan Mq-135 Berbasis SMS Gateway," Skripsi, Fakultas Teknologi Informasi, Universitas Budiluhur, Jakarta, Indonesia, 2020. [Online]. Available: https://lib.budiluhur.ac.id/Koleksitesis/detail/MXwxNjExNTAxMzk0

[10] D. Hidayat and I. Sari, "Monitoring Suhu dan Kelembaban Berbasis Internet Of Things (IoT)," *J. Penelit. Tek. Inform. Univ. Prima Indones. Medan*, vol. 4, no. 1, pp. 525–530, 2021, doi: https://doi.org/10.34012/jutikomp.v4i1.1676.

[11] R. D. Ritonga, "Prototype Smart Home Berbasis Internet Of Things Dengan Menggunakan Website," Skripsi, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia, 2022. [Online]. Available: https://lib.budiluhur.ac.id/Koleksitesis/show/OHwxfDE4MTE1MTAzNTE

[12] I. A. Agustina, "Lampu LED, Bohlam, Neon : Perbedaan, Penggunaan, Kelebihan Serta Kekurangan," *Binus University*, 2020. https://binus.ac.id/malang/2020/06/lampu-led-bohlam-neon-perbedaan-penggunaan-kelebihan-serta-kekurangan/ (accessed Jun. 11, 2023).

[13] I. Salim, A. Muzakir, and H. Syaputra, "Perangkat Lunak Koreksi Penggunaan Sumber Kutipan dan Daftar Pustaka Pada Jurnal Mahasiswa," *J. Pengemb. Sist. Inf. dan Inform.*, vol. 1, no. 2, pp. 100–109, 2020, doi: https://doi.org/10.47747/jpsii.v1i2.169.