

Implementasi *Internet of Things (IoT)* Untuk Sistem Kendali Pada *Smart Lamp*

Dinar Priambodo Utomo^{*1}, Nur Latifah Dwi Mutiara Sari²

^{1,2} Prodi Informatika, Universitas PGRI Semarang, Kota Semarang

*Email : ¹dinarutomo00@gmail.com, ²nurlatifah@upgris.ac.id

Abstract

This research aims to develop and implement an Internet of Things (IoT)-based control system for smart lamps. Smart lamp is designed to provide convenience in lighting management by utilizing modern technology. In this system, users can control lights through a smartphone-based application, which is connected to the internet network. The system involves the use of microcontrollers, sensors, and wireless communication modules to enable remote control and automation of lighting. Tests show that the system can respond to user commands quickly and accurately. The implementation of IoT in this smart lamp not only improves energy efficiency, but also provides convenience and flexibility for users in managing lighting at home or work. The results of this study show the great potential of using IoT in improving the quality of life through smart home devices.

Keywords: Internet of Things (IoT), Control system, Smart lamp, Sensor, Mobile application.

Abstrak

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem kendali berbasis *Internet of Things (IoT)* pada *smart lamp*. *Smart lamp* dirancang untuk memberikan kemudahan dalam pengelolaan pencahayaan dengan memanfaatkan teknologi modern. Dalam sistem ini, pengguna dapat mengontrol lampu melalui aplikasi berbasis smartphone, yang terhubung ke jaringan internet. Sistem ini melibatkan penggunaan mikrokontroler, sensor, dan modul komunikasi nirkabel untuk memungkinkan kendali jarak jauh dan otomatisasi pencahayaan. Pengujian menunjukkan bahwa sistem dapat merespons perintah pengguna dengan cepat dan akurat. Implementasi *IoT* pada *smart lamp* ini tidak hanya meningkatkan efisiensi energi, tetapi juga memberikan kenyamanan dan fleksibilitas bagi pengguna dalam mengatur pencahayaan di rumah atau tempat kerja. Hasil penelitian ini menunjukkan potensi besar penggunaan *IoT* dalam meningkatkan kualitas hidup melalui perangkat rumah pintar.

1. Pendahuluan

Pengelolaan pencahayaan melalui teknologi cerdas telah menjadi fokus utama dalam pengembangan teknologi rumah pintar. Tantangan yang dihadapi adalah meningkatkan efisiensi energi dan kenyamanan dalam pengelolaan pencahayaan di rumah atau tempat kerja[1]. Dengan meningkatnya kebutuhan akan sistem otomatisasi yang lebih pintar dan efisien, solusi berbasis *Internet of Things (IoT)* untuk *smart lamp* menawarkan potensi besar untuk diinvestigasi.

Berbagai teknologi pencahayaan sebelumnya, seperti sakelar manual, timer analog, dan sistem otomatis dengan sensor gerak, telah mencoba mengatasi masalah ini. Namun, meskipun teknologi tersebut memberikan kemudahan dan efisiensi, mereka masih terbatas dalam hal fleksibilitas dan kemampuan kendali jarak jauh *real-time*[2]. Teknologi komunikasi nirkabel seperti *ZigBee* dan *Z-Wave* telah diperkenalkan untuk mengatasi beberapa keterbatasan ini, tetapi belum sepenuhnya optimal dalam hal interoperabilitas dan skalabilitas.

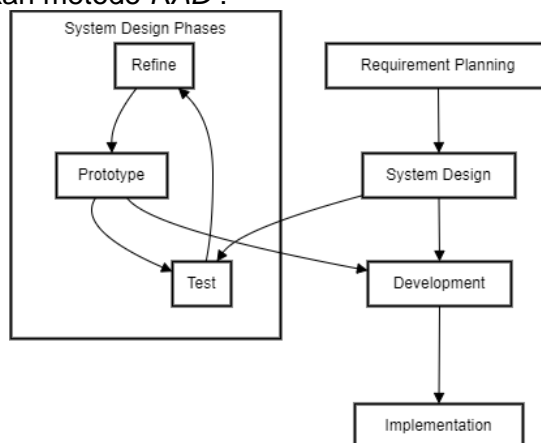
Penelitian sebelumnya telah menyarankan penggunaan *IoT* untuk meningkatkan fleksibilitas dan kendali pencahayaan. Namun, masih ada kekurangan dalam hal respon waktu dan keandalan sistem. Banyak solusi yang ada belum sepenuhnya mempertimbangkan

kemudahan penggunaan dan integrasi dengan perangkat pintar lainnya. Oleh karena itu, ada kebutuhan mendesak untuk mengembangkan sistem yang lebih andal dan *user-friendly*.

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem kendali smart lamp berbasis *IoT* yang memungkinkan pengguna mengontrol pencahayaan melalui aplikasi smartphone. Dengan menggunakan *mikrokontroler*, *sensor*, dan modul komunikasi nirkabel, sistem ini memungkinkan kendali jarak jauh dan otomatisasi pencahayaan, memberikan efisiensi energi dan kenyamanan yang lebih baik. Hasil penelitian diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan kualitas hidup melalui perangkat rumah pintar yang lebih efisien dan fleksibel.

2. Metode

Penelitian ini menggunakan metode *Rapid Application Development (RAD)*. Metode RAD adalah sebuah metode proses pengembangan perangkat lunak yang menekankan siklus pengembangan dengan waktu singkat. Kelebihan *Rapid Application Development (RAD)* adalah metode pengembangan perangkat lunak yang sangat responsif terhadap perubahan persyaratan aplikasi yang dapat terjadi sewaktu-waktu. RAD memungkinkan aplikasi dikembangkan secara iteratif, berdasarkan kebutuhan dan keinginan pengguna yang dapat teridentifikasi dan diimplementasikan secara cepat. Pendekatan ini membantu dalam meminimalkan kemungkinan kesalahan dan masalah lainnya dengan melibatkan pengguna secara aktif selama proses pengembangan. Dengan fokus pada pengembangan yang lebih cepat dan efisien, RAD juga mendukung integrasi yang lebih mudah dengan sistem yang ada, mempercepat waktu peluncuran aplikasi secara keseluruhan. Penulis memilih metode ini karena mampu membantu membangun sistem dengan efisien dan efektif. Tahapan ini terdiri dari *Requirement Planning, Design, Contruction* [8]. Berikut adalah tahapan proses pengembangan menggunakan metode *RAD* :



Gambar 1. Metode RAD

2.1. Requirement Planning

Pada tahapan ini dilakukan analisis pengumpulan. Pengumpulan data dilakukan dengan cara observasi, informasi yang diperoleh akan menjadi data dan dianalisis untuk menentukan kebutuhan sehingga dapat dibangun rancangan alat dan sistem dengan kebutuhan.

1) Analisis Kebutuhan Fungsional

Dengan perancangan sistem ini pengguna dapat mengendalikan perangkat *smart lamp* walaupun tidak sedang berada dirumah, pengguna bisa mengontrol perangkat *smart lamp* melalui *voice asisstant*. Berikut adalah analisis kebutuhan pada sistem pengendali perangkat *smart lamp* ini : Sistem dapat mengendalikan On/Off perangkat *smart lamp* melalui *voice asisstant*

2) Analisis Kebutuhan Non-Fungsional

Terdapat dua kebutuhan nonfungsional yang digunakan pada sistem ini, yaitu kebutuhan *hardware* dan kebutuhan *software*.

a. Analisis Kebutuhan Perangkat Keras

Kebutuhan *hardware* yang diperlukan untuk sistem ini dapat dilihat pada Tabel I.

Tabel I. Analisis kebutuhan perangkat keras

No.	Nama Hardware	Fungsi
1	Realy	saklar otomatis
2	Modul ESP32	mikrokonroler dan juga koneksi internet (wifi)
3	Kabel Jumper	Menghubungkan relay dengan modul ESP32
4	Kabel data tipe USB Mikro B	Menghubungkan modul ESP32 dengan sumber listrik
5	Smartphone	Untuk mengakses voice assistant
6	Laptop	Untuk memprogram sistemdi aplikasi Arduino IDE

b. Analisis Kebutuhan Perangkat Lunak

Kebutuhan *software* yang diperlukan untuk sistem ini dapat dilihat pada Tabel II.

Tabel II. Analisis kebutuhan perangkat lunak

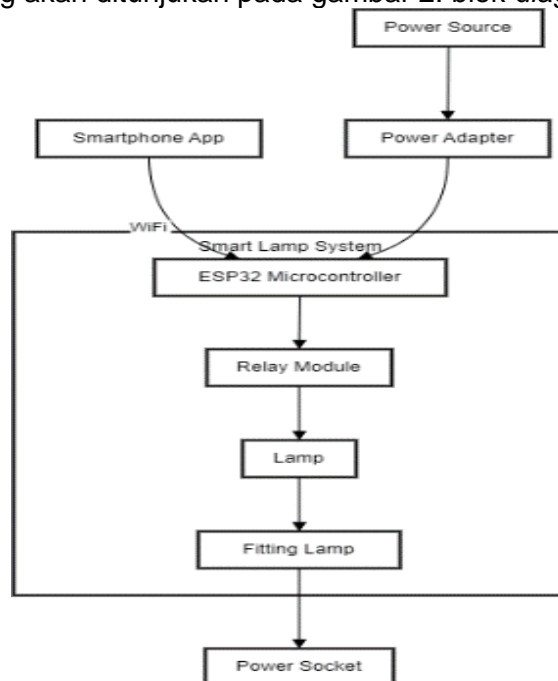
No.	Nama Software	Fungsi
1	Voice assistant	Penerima perintah suara
2	Android 13	OS smartphone
3	Arduino IDE	Untuk membuat kodeprogram

2.2. Design

Pada tahap design ini dilakukan perancangan sistem dan alat berdasarkan tahapan sebelumnya. Tahapan ini menghasilkan perancangan sistem dan alat yang lainnya yaitu rancangan mekanis, rancangan elektronis, dan rancangan *flowchart*.

2.2.1. Rancangan mekanis

Rancangan mekanis merupakan tahapan implementasi dari rancangan implementasi *IoT* untuk sistem kendali pada *smart lamp*. Berikut rancangan mekanis cara kerja perangkat yang akan ditunjukkan pada gambar 2. blok *diagram* berikut :

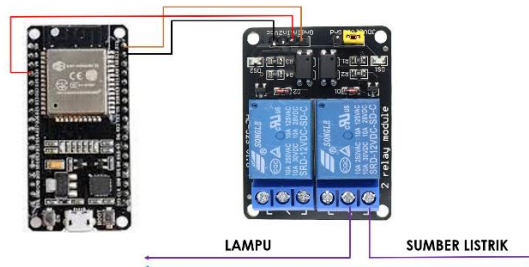


Gambar 2. blok *diagram* smart lamp berbasis *IoT*

Sistem *smart lamp* ini dirancang untuk memungkinkan kontrol pencahayaan secara otomatis dan manual melalui aplikasi *mobile*. Mikrokontroler *ESP32* digunakan sebagai pengendali utama yang menghubungkan sensor dan lampu *LED* serta memungkinkan komunikasi nirkabel dengan aplikasi *mobile*.

2.2.2. Rancangan Elektronika

Untuk menghubungkan semua komponen *hardware*, diperlukan rangkaian elektronika yang tepat[3]. Gambar 3 menunjukkan rancangan elektronika *smart lamp* berbasis *IoT*.

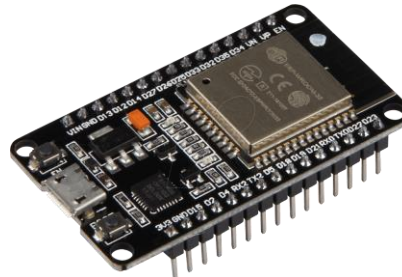


Gambar 3. Rancangan elektronika

Pada rangkaian ini, keseluruhan perangkat yang digunakan ditunjukkan. *Power* menggunakan 5 volt ditunjukkan oleh kabel berwarna merah yang tersambung pada setiap perangkat, sedangkan *ground* ditunjukkan oleh kabel berwarna hitam.

Pemilihan komponen perangkat keras merupakan langkah yang penting dalam pengembangan sistem kendali pada *smart lamp* dibawah ini adalah deskripsi mengenai komponen utama yang digunakan:

2.2.2.1. ESP32



Gambar 4. ESP32

ESP32 adalah mikrokontroler dari *Espressif Systems* yang populer untuk aplikasi *IoT*. Ini memiliki prosesor *dual-core*, modul *Wi-Fi* dan *Bluetooth* terintegrasi, banyak pin *GPIO*, *RAM* hingga 520 KB, dan memori *flash* hingga 16 MB. Fitur keamanan termasuk enkripsi *AES*, *RSA*, dan *SHA*. *ESP32* sering digunakan dalam proyek elektronik *DIY* dan produk komersial karena performanya yang tinggi dan kemampuan komunikasi nirkabel.

2.2.2.2. Relay Module



Gambar 5. Relay Module

Relay module adalah komponen elektronik yang berfungsi sebagai saklar elektromekanis yang dikendalikan oleh sinyal listrik. Modul ini memungkinkan perangkat berdaya rendah (seperti *mikrokontroler ESP32*) untuk mengontrol sirkuit berdaya tinggi (seperti lampu atau motor).

2.2.2.3. Kabel *Female to Female*



Gambar 6. Kabel *Female to Female*

Kabel *female to female* adalah jenis kabel konektor yang memiliki konektor *female* (soket) di kedua ujungnya. Jenis kabel ini digunakan untuk menghubungkan dua perangkat atau komponen yang masing-masing memiliki konektor *male* (pin atau colokan).

2.2.2.4. Adapter



Gambar 7. *Adapter*

Adapter adalah perangkat yang digunakan untuk mengubah antarmuka atau koneksi satu perangkat elektronik ke perangkat lain yang memiliki antarmuka atau konektor yang berbeda. Tujuannya adalah untuk memungkinkan dua perangkat yang sebelumnya tidak kompatibel berhubungan atau beroperasi bersama.

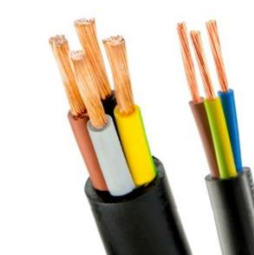
2.2.2.5. *USB ke USB Mikro B*



Gambar 8. *USB ke USB Mikro B*

USB ke USB mikro B adalah sebuah kabel atau *adapter* yang digunakan untuk menghubungkan dua perangkat yang memiliki konektor *USB mikro B*. Konektor *USB mikro B* umumnya digunakan pada perangkat seperti *smartphone*, *tablet*, atau perangkat kecil lainnya. Ini memungkinkan transfer data atau pengisian daya antara perangkat-perangkat tersebut.

2.2.2.6. Kabel



Gambar 9. Kabel

Kabel adalah media penghantar yang digunakan untuk mentransmisikan listrik, data, atau sinyal dari satu titik ke titik lain. Kabel terdiri dari satu atau lebih konduktor (biasanya tembaga atau aluminium) yang dilindungi oleh isolator untuk mencegah kontak antar konduktor dan melindungi dari kerusakan fisik.

2.2.2.7. Lampu



Gambar 10. Lampu

Lampu adalah perangkat yang digunakan untuk menghasilkan cahaya. Biasanya digunakan untuk penerangan, baik di dalam ruangan maupun di luar ruangan.

2.2.2.8. Colokan



Colokan adalah perangkat keras yang digunakan untuk menghubungkan perangkat listrik atau elektronik ke sumber daya listrik atau ke perangkat lain. Ini terdiri dari pin atau terminal yang dirancang untuk masuk ke soket atau *port* yang sesuai, memungkinkan aliran listrik atau transfer data.

2.2.2.9. Fitting Lampu

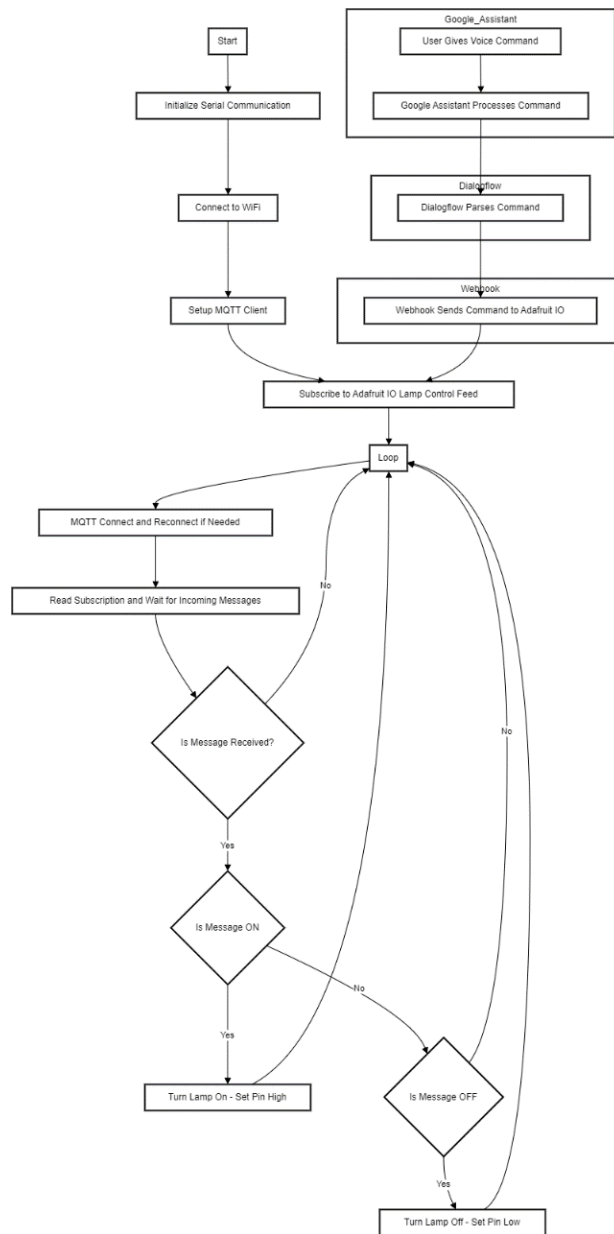


Gambar 12. Fitting Lampu

Fitting lampu adalah komponen yang digunakan untuk memasang dan menghubungkan bohlam lampu ke sumber listrik. Fitting ini menyediakan kontak listrik yang aman dan stabil, serta biasanya memiliki mekanisme untuk mengunci bohlam di tempatnya.

2.2.3. Rancangan *Flowchart*

Flowchart adalah diagram alir yang menjelaskan proses dari suatu perancangan program. *Flowchart* memiliki peran penting dalam memperjelas berjalannya suatu program agar lebih mudah untuk dipahami oleh pembaca. *Flowchart Smart Lamp* ini akan diimplementasikan kedalam aplikasi *Arduino IDE*[9]. Adapun proses perancangan dalam penyusunan program sistem yang akan disajikan pada gambar dibawah ini :

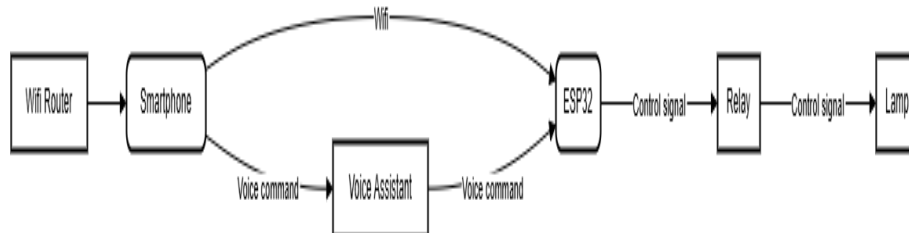


Gambar 13. Flowchart Smart Lamp berbasis IoT dengan perintah suara

Alur kerja sistem pengendali perangkat *smart lamp* digambarkan pada Gambar 13. Dari Gambar dapat kita lihat bahwa sistem pengendali perangkat *smart lamp* akan mulai berjalan ketika kita menyalakan ESP32 dan *Relay Module*. Ketika ESP32 menyala ESP32 akan secara otomatis terhubung ke internet, sehingga kita bisa memasukkan perintah suara berupa On dan Off perangkat *smart lamp* melalui *voice asisstant*, perintah suara tersebut akan diproses oleh sistem pengendali perangkat *smart lamp*. Relay akan menerima data dan mengeksekusi data yang diterima itu dengan cara mengaliri atau memutus arus listrik ke perangkat *smart lamp*.

Sedangkan untuk perancangan alat sistem pengendali *perangkat smart lamp* ini menggunakan module ESP32, Relay 2 chanel, dan beberapa perangkat elektronik sebagai contoh yaitu lampu. Untuk mengetahui prinsip kerja dari alat maka diperlukan rancangan arsitektur sistem yang akan dibuat. Gambar 14 adalah rancangan arsitektur sistem pengendali perangkat elektronik.

Gambar 14. Perancangan alat sistem



Dari gambar dapat dilihat bahwa smartphone merupakan device yang digunakan untuk mengakses voice asisstant dan untuk memberikan perintah suara agar dapat mengontrol on/off pada perangkat elektronik. Module ESP32 yang digunakan pada penelitian ini sudah di lengkapi dengan fitur wifi, sehingga module ESP32 dapat menerima perintah yang dikirimkan melalui *voice asisstant*. Relay dihubungkan ke module ESP32 melalui pin GPIO (*General Purpose Input-Output*) . kemudian, perangkat elektronik dihubungkan ke Relay sesuai dengan channel yang akan digunakan. Pada sistem ini relay digunakan sebagai saklar otomatis yang mana relay dapat mengendalikan dan mengaliri arus listrik ke perangkat elektronik. Ketika relay dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet sehingga kontak saklar akan atau bernilai 1 dan perangkat elektronik akan menyala.

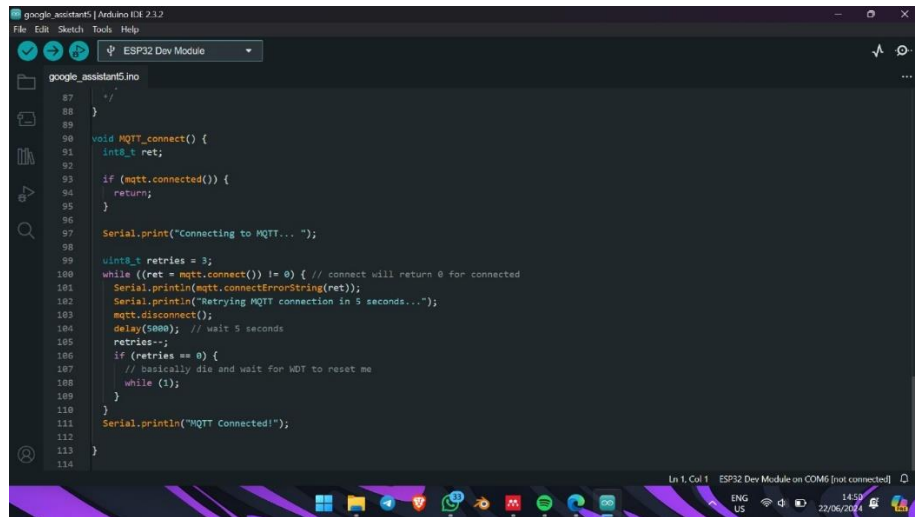
2.3. Construction

Pada tahap ini dilakukan pengembangan perangkat lunak menggunakan *Arduino IDE* dengan bahasa pemrograman C++. Hal pertama yang dilakukan adalah menginstal *library* yang diperlukan untuk mengendalikan *smart lamp*, seperti *library WiFi*, *Adafruit IO*, dan *Google Assistant* di *Arduino IDE*. Kemudian, untuk bisa mengontrol lampu, tuliskan kode untuk menghubungkan *ESP32* ke jaringan *WiFi* dan platform *Adafruit IO*. Kontrol lampu dilakukan dengan menuliskan kode berdasarkan perintah yang diterima dari *Google Assistant* melalui *Adafruit IO*. Berikut adalah kode program untuk menjalankan *smart lamp* berbasis *IoT* yang terintegrasi dengan sistem suara *Google Assistant*[4].

```

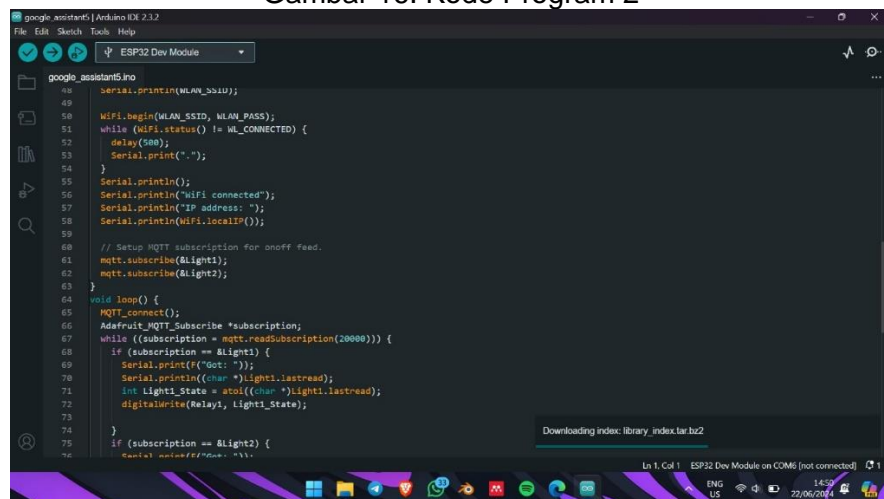
1 // google_assistant5.ino
2 // *****
3 //for esp32 use <wifi.h> ,for esp8266 use <esp8266wifi.h>
4 // this code is for esp32 for using esp8266 you just need to change <wifi.h> into <esp8266wifi.h> and also change the relay pin
5 #include <WiFi.h>
6 #include "Adafruit_HTTP.h"
7 #include "Adafruit_HTTP_Client.h"
8 #include <ArduinoHttpClient.h>
9 //***** Pin Definition *****
10
11 //Relays for switching appliances
12 #define Relay1 2
13 #define Relay2 12
14
15 //***** WiFi Access Point *****
16
17 #define WLAN_SSID "Xiaomi 11T"
18 #define WLAN_PASS "soehartono12"
19
20 //***** Adafruit.io Setup *****
21
22 #define AIO_SERVER "io.adafruit.com"
23 #define AIO_SERVERPORT 1883 // use 8883 for SSL
24 #define AIO_USERNAME "dostoyevsky"
25 #define AIO_KEY "aio_amo18870wqzysiyVeeGwHtZ0hngH"
26 //***** Global State (you don't need to change this!) *****
27
28 WiFiClient client;
    
```

Gambar 15. Kode Program 1



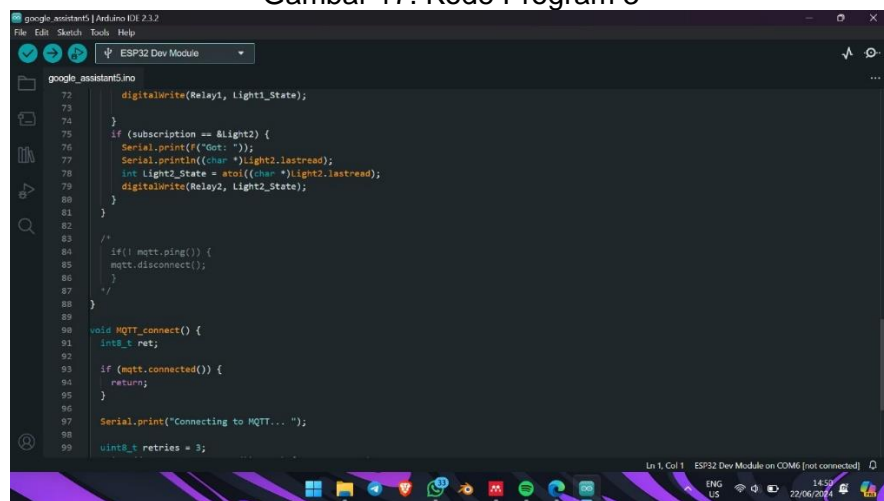
```
google_assistant5 | Arduino IDE 2.1.2
File Edit Sketch Tools Help
ESP32 Dev Module
google_assistant5.ino
87 /*
88 */
89
90 void MQTT_connect() {
91   int8_t ret;
92
93   if (mqtt.connected()) {
94     return;
95   }
96
97   Serial.print("Connecting to MQTT... ");
98
99   uint8_t retries = 3;
100  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
101    Serial.println(mqtt.connectErrorString(ret));
102    Serial.println("Retrying MQTT connection in 5 seconds...");
103    mqtt.disconnect();
104    delay(5000); // wait 5 seconds
105    retries--;
106    if (retries == 0) {
107      // basically die and wait for WDT to reset me
108      while (1);
109    }
110  }
111  Serial.println("MQTT Connected!");
112 }
113
114
```

Gambar 16. Kode Program 2



```
google_assistant5 | Arduino IDE 2.1.2
File Edit Sketch Tools Help
ESP32 Dev Module
google_assistant5.ino
49 Serial.println(WLAN_SSID);
50
51 WiFi.begin(WLAN_SSID, WLAN_PASS);
52 while (WiFi.status() != WL_CONNECTED) {
53   delay(500);
54   Serial.print(".");
55 }
56 Serial.println();
57 Serial.println("WiFi connected");
58 Serial.println("IP address: ");
59 Serial.println(WiFi.localIP());
60
61 // Setup MQTT subscription for onoff feed.
62 mqtt.subscribe(&light1);
63 mqtt.subscribe(&light2);
64
65 void loop() {
66   MQTT_connect();
67   Adafruit_MQTT_Subscribe *subscription;
68   while ((subscription = mqtt.readSubscription(20000))) {
69     if (subscription == &light1) {
70       Serial.print("Got: ");
71       Serial.println((char *)light1.lastread);
72       int Light1_State = atoi((char *)light1.lastread);
73       digitalWrite(Relay1, Light1_State);
74     }
75     if (subscription == &light2) {
76       Serial.print("Got: ");
77       Serial.println((char *)light2.lastread);
78       int Light2_State = atoi((char *)light2.lastread);
79       digitalWrite(Relay2, Light2_State);
80     }
81   }
82 }
83
84 /*
85 if (!mqtt.ping()) {
86   mqtt.disconnect();
87 }
88 */
89
90 void MQTT_connect() {
91   int8_t ret;
92
93   if (mqtt.connected()) {
94     return;
95   }
96
97   Serial.print("Connecting to MQTT... ");
98
99   uint8_t retries = 3;
100  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
101    Serial.println(mqtt.connectErrorString(ret));
102    Serial.println("Retrying MQTT connection in 5 seconds...");
103    mqtt.disconnect();
104    delay(5000); // wait 5 seconds
105    retries--;
106    if (retries == 0) {
107      // basically die and wait for WDT to reset me
108      while (1);
109    }
110  }
111  Serial.println("MQTT Connected!");
112 }
113
114
```

Gambar 17. Kode Program 3



```
google_assistant5 | Arduino IDE 2.1.2
File Edit Sketch Tools Help
ESP32 Dev Module
google_assistant5.ino
72 digitalWrite(Relay1, Light1_State);
73
74 }
75 if (subscription == &light2) {
76   Serial.print("Got: ");
77   Serial.println((char *)light2.lastread);
78   int Light2_State = atoi((char *)light2.lastread);
79   digitalWrite(Relay2, Light2_State);
80 }
81 }
82 }
83
84 /*
85 if (!mqtt.ping()) {
86   mqtt.disconnect();
87 }
88 */
89
90 void MQTT_connect() {
91   int8_t ret;
92
93   if (mqtt.connected()) {
94     return;
95   }
96
97   Serial.print("Connecting to MQTT... ");
98
99   uint8_t retries = 3;
100  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
101    Serial.println(mqtt.connectErrorString(ret));
102    Serial.println("Retrying MQTT connection in 5 seconds...");
103    mqtt.disconnect();
104    delay(5000); // wait 5 seconds
105    retries--;
106    if (retries == 0) {
107      // basically die and wait for WDT to reset me
108      while (1);
109    }
110  }
111  Serial.println("MQTT Connected!");
112 }
113
114
```

Gambar 18. Kode Program 4

```

25 #define AIO_KEY "aio_xmo188dToxqzyiyVemQvTWI0hgk"
26 /***** Global State (you don't need to change this!) *****/
27
28 WiFiClient client;
29 ADFRUIT_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
30 ADFRUIT_MQTT_Publish Light = ADFRUIT_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/light");
31
32 // Setup a feed called "onoff" for subscribing to changes.
33 ADFRUIT_MQTT_Subscribe Light1 = ADFRUIT_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/relay1");
34 ADFRUIT_MQTT_Subscribe Light2 = ADFRUIT_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/relay2");
35
36 /***** Sketch Code *****/
37 void MQTT_connect();
38 void setup() {
39   Serial.begin(115200);
40   delay(10);
41   pinMode(Relay1, OUTPUT);
42   pinMode(Relay2, OUTPUT);
43
44   Serial.println("Adafruit MQTT demo");
45   // Connect to WiFi access point.
46   Serial.println(); Serial.println();
47   Serial.print("Connecting to ");
48   Serial.println(WLAN_SSID);
49
50   WiFi.begin(WLAN_SSID, WLAN_PASS);
51   while (WiFi.status() != WL_CONNECTED) {
52     delay(500);

```

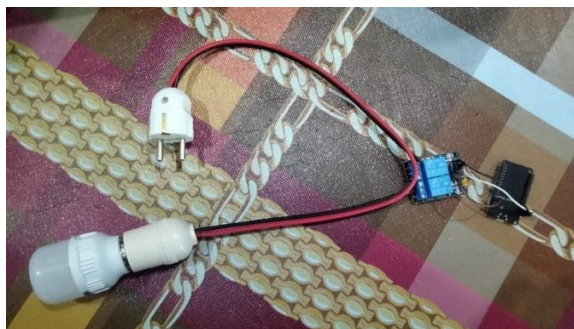
Gambar 19. Kode Program 5

3. Hasil dan Pembahasan

3.1 Penyajian Hasil

3.1.1. *Prototipe Desain*

Pada tahap penyajian hasil dari desain dan implementasi *iot* untuk sistem kendali pada *smart lamp*. Berikut adalah *prototipe desain smart lamp* yang dilengkapi dengan fitur *voice command*, ditampilkan untuk memberikan Gambaran menyeluruh mengenai komponen yang digunakan, seperti *ESP32* sebagai *mikrokontroler* utama, modul relay untuk mengontrol arus listrik ke lampu, serta integrasi dengan *Google Assistant* untuk memungkinkan kendali melalui perintah suara. Selain itu, skema rangkaian dan *diagram* alur kerja sistem turut disertakan untuk memperjelas proses komunikasi antara perangkat keras dan perangkat lunak, memastikan setiap komponen bekerja secara harmonis dalam menciptakan pengalaman pengguna yang optimal dan intuitif[10].



Gambar 20. *Prototipe* desain smart lamp.

Tampak *ESP32* yang tersambung dengan kabel *jumper* yang dihubungkan pada *relay module*. Tampak colokan yang tersambung dengan *relay module*. Tampak lampu yang juga tersambung pada *relay module*, memastikan suplai listrik yang diperlukan. Lampu juga tersambung pada *relay module*, menunjukkan bagaimana kontrol aliran listrik ke lampu dilakukan. Komponen-komponen ini bekerja bersama untuk memungkinkan kendali *smart lamp* melalui perintah suara, menunjukkan sinergi antara perangkat keras dan perangkat lunak dalam sistem kendali IoT ini[5].

3.1.2 *Pengujian*

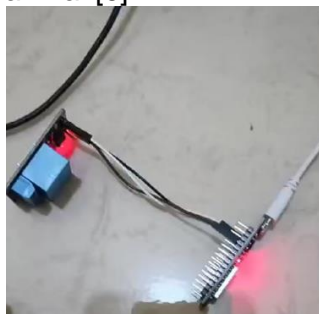
Pada tahap pengujian menyajikan hasil yang disajikan mengenai peforma dan efektivitas sistem yang telah dikembangkan, seperti menghidupkan dan mematikan lampu. berikut adalah hal yang diuji :

1. Menghidupkan dan Mematikan Lampu
 - *ESP32* :

Pengujian mematikan dan menyalakan lampu pada smart lamp menggunakan ESP32 telah berhasil dilakukan dengan sukses. Melalui pengaturan yang tepat menggunakan Arduino IDE, ESP32 dapat mengendalikan relay module yang berfungsi sebagai saklar untuk menghidupkan dan mematikan lampu. Hasil pengujian menunjukkan respons yang cepat dan konsisten dari ESP32 dalam merespons perintah dari pengguna atau melalui sistem otomatisasi yang terintegrasi. Keberhasilan pengujian ini menegaskan bahwa implementasi ESP32 dalam smart lamp berjalan dengan baik, memungkinkan kontrol yang efektif dan dapat diandalkan terhadap pencahayaan ruangan.

- Relay Module :

Pengujian pengendalian relay untuk mematikan dan menyalakan lampu pada smart lamp menggunakan ESP32 telah berhasil dilakukan dengan sukses. Dengan konfigurasi yang tepat melalui Arduino IDE, ESP32 dapat mengontrol relay module sebagai saklar untuk mengatur daya listrik yang diteruskan ke lampu. Hasil pengujian menunjukkan bahwa ESP32 mampu mengirimkan sinyal kontrol yang akurat dan responsif, sehingga dapat secara efektif menghidupkan dan mematikan lampu sesuai dengan perintah yang diberikan. Keberhasilan pengujian ini memvalidasi fungsi utama ESP32 dalam aplikasi smart lamp, memastikan operasional yang handal dan sesuai dengan kebutuhan penggunaan sehari-hari[6].



Gambar 21. Pengujian ESP32 dan Relay Module

Berikut adalah table penyajian hasil pengujian smart lamp berbasis IoT.

Table 1. penyajian hasil pengujian

Komponen	Fungsi	Hasil Pengujian	Keterangan
ESP32	Otak dari sistem kontrol lampu yang terhubung ke jaringan WiFi	Menerima perintah dari mobile	Berfungsi dengan baik
Relay Module	pengontrol daya utama yang mengatur apakah lampu dinyalakan atau dimatikan	Mengontrol daya lampu	Berfungsi dengan baik

3.2 Pembahasan

Berdasarkan hasil penelitian maka pembahasannya adalah Implementasi Internet of Things (IoT) untuk sistem kendali pada smart lamp membawa transformasi signifikan dalam cara lampu dikelola dan dimanfaatkan dalam lingkungan rumah atau kantor. Dengan memanfaatkan konektivitas internet, smart lamp menjadi lebih dari sekadar sumber cahaya, tetapi juga alat yang dapat dikendalikan secara jarak jauh melalui perangkat mobile atau sistem terhubung lainnya. Kemampuan ini memungkinkan pengguna untuk menyalakan, mematikan dengan mudah tanpa harus berada di dekat lampu itu sendiri. Selain itu, otomatisasi dapat meningkatkan efisiensi energi. Namun, keberhasilan implementasi IoT pada smart lamp juga menuntut perhatian yang serius terhadap keamanan data dan privasi pengguna, serta memastikan kompatibilitas dengan berbagai platform dan perangkat pintar lainnya. Dengan manajemen yang tepat

atas tantangan ini, *IoT* dapat mengubah cara kita berinteraksi dengan pencahayaan dalam kehidupan sehari-hari, menciptakan lingkungan yang lebih cerdas, efisien, dan terhubung secara *seamless*.

4. Kesimpulan

Kesimpulan yang dapat diambil adalah Sistem Pengendali Perangkat *Smart Lamp* melalui *Voice Assisstant* telah berhasil dirancang dengan menggunakan modul ESP32 sebagai mikrokontroler, modul Relay sebagai saklar otomatis yang berfungsi untuk mengaliri dan memutuskan arus listrik pada perangkat elektronik. Untuk pengendalian perangkat elektronik dibuat dalam bentuk kontrol suara lewat *voice asisstant* yang dapat diakses melalui *smartphone*. Sistem ini memungkinkan pengguna untuk mengendalikan perangkat elektronik meski berada jauh dari rumah dengan memanfaatkan jaringan internet.

Penelitian ini menghadirkan kemudahan dan keterhubungan yang belum pernah terjadi sebelumnya. Dengan memanfaatkan teknologi ini, pengguna dapat mengontrol lampu secara langsung melalui perintah suara kepada asisten *virtual Google Assistant*, dari mana pun mereka berada. *ESP32* berperan sebagai perangkat utama yang memfasilitasi komunikasi antara lampu dan internet, memungkinkan pengendalian yang *real-time* dan responsif. Hal ini tidak hanya meningkatkan kenyamanan, tetapi juga efisiensi energi dengan memungkinkan pengguna untuk dengan mudah mematikan atau menghidupkan lampu secara tepat waktu, bahkan saat mereka tidak berada di tempat. Dengan demikian, implementasi *IoT* dalam *smart lamp* ini tidak hanya membawa kepraktisan dalam kehidupan sehari-hari, tetapi juga mendorong ke arah rumah pintar yang lebih terkoneksi dan cerdas.

5. Referensi

- [1] Suhardi, Rahmi Hidayati, Irma Nirmala. Smart Lamp: Kendali dan Monitor Lampu Berbasis Internet Of Things (IoT). J.
- [2] Dody Susilo, Churnia Sari, Galas Widya Krisna. Sistem Kendali Lampu pada Smart Home Berbasis IoT (Internet of Things). Jurnal ELECTRA : Electrical Engineering Articles. 2021;2:23–30.
- [3] Harun Kurniawan, Yudi Sutanto. Perancangan dan Pembuatan Smart Garden Lamp Berbasis Internet of Things (IoT) Menggunakan Mikrokontroler Untuk Efisiensi Penggunaan Energi Listrik. Jurnal Teknologi Informasi. 2022;XVII:62.
- [4] Yulisman, Ikhwanul Ikhsan, Anita Febriani, Rika Melyanti. PENERAPAN INTERNET OF THINGS (IoT) KONTROL LAMPU MENGGUNAKAN NODEMCU ESP8266 DAN SMARTPHONE. JURNAL ILMU KOMPUTER (COMPUTER SCIENCE JOURNAL). 2021;10:139.
- [5] Ramdan, Lasmadi, Paulus Setiawan. Sistem Pengendali On-Off Lampu dan Motor Servo sebagai Penggerak Gerendel Pintu Berbasis Internet Of Things (IoT). AVITEC. 2022;4:217.
- [6] Marina Artiyasa, Aidah Nita Rostini, Edwinanto, Anggy Pradifita Junfithrana. APLIKASI SMART HOMENODE MCU IOT UNTUK BLYNK. Jurnal Rekayasa Teknologi Nusa Putra. 2020;7:1–7.
- [7] Nurul Imamah, M.T., Dewa Sagara Andika. PERANCANGAN SISTEM MONITORING DAN PENGENDALIAN LAMPU MENGGUNAKAN SENSOR GERAK DAN SENSOR CAHAYA DILENGKAPI INTERNET OF THINGS (IOT). Jurnal Informatika – COMPUTING. 2021;08:14–21.
- [8] Dede Kurniadi, Lia Amelia. Sistem Kendali Perangkat Elektronik Rumah Berbasis Android dan Arduino. Jurnal Algoritma. 2018;15:38.
- [9] Sarmayanta Sembiring, Muhammad Ali Buchar. Perancangan dan Implementasi Sistem Pengendalian dan Monitoring Penggunaan Peralatan Elektronik Berbasis Internet of Thing (IoT). JURNAL MEDIA INFORMATIKA BUDIDARMA. 2021;5:1589.
- [10] Fredy Wardana, Mukhsin. PENGGUNAAN INTERNET OF THING (IOT) ARDUINO UNO PADA LAMPU PENGINAPAN BERBASIS WEBSITE. Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi. 2023;5:96.