

## PERANCANGAN DAN IMPLEMENTASI SMART HOME MENGGUNAKAN RASPBERRY PI DAN ESP8266

Alif Fakhri Muhammad Hafidz<sup>1</sup>, Muhammad Ikhsan Sani<sup>2</sup>, Lisda Meisaroh<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

alifhafidz@student.telkomuniversity.ac.id<sup>1</sup>, ikhsansani@tass.telkomuniversity.ac.id<sup>2</sup>,

lisdameisaroh@tass.telkomuniversity.ac.id<sup>3</sup>

---

### Abstrak

Sistem Otomatisasi rumah dapat diterapkan ke banyak bidang termasuk keamanan rumah, *control* pencahayaan, sakelar pintar, dan *control* pintu memberikan kenyamanan, keamanan, efisiensi energi (biaya operasi rendah) dan kemudahan pemilik rumah setiap saat. *Internet of Things* adalahantisipasi untuk mengaktifkan berbagai layanan rumah pintar dimana setiap layanan menyediakan serangkaian solusi otomatisasi rumah. Proyek Akhir yang diusulkan ini terdiri dari pengembangan pemantauan rumah otomatis menggunakan raspberry Pi yang menyediakan platform yang dapat disesuaikan dan hemat biaya. Dalam pengerjaan ini, penulis mengembangkan dan mengimplementasikan konsep *smart home* menggunakan sensor dan aktuator yang terhubung ke *gateway* melalui protokol Wi-Fi. Raspberry Pi dan ESP8266 adalah kekuatan di balik proyek ini karena menyediakan antarmuka antara sensor dan aktuator.

Kata Kunci: Smart Home, Internet of Things

---

### Abstract

*The home automation system can be applied to many areas including home security, lighting control, smart switch, door control to provides its homeowner's comfort, security, energy efficiency (low operating costs) and convenience at all times. The Internet of Things (IoT) is anticipated to enable a variety of smart home services in which each service provides a set of home automation solutions. This proposed project consists of developing an automated home monitoring using Raspberry Pi that provides a customizable and cost efficient platform for a smart home. In this project, writer develop and implement a smart home concept using sensor and actuators which are connected to gateway via Wi-Fi protocol. Raspberry Pi and ESP8266 is the power behind this project as it provides an interface between the sensors and actuators.*

*Keywords: Smart Home, Internet of Things*

---

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Di zaman yang serba modern seperti sekarang, rumah bisa menjadi sangat bersahabat dengan penghuninya. Misalnya rumah cerdas atau biasa disebut *Smart Home* yang menurut *Home Living Indonesia* (2012), merupakan rumah yang dilengkapi dengan sistem pengoperasian terkontrol untuk banyak hal seperti pencahayaan lampu, barang-barang elektronik, serta benda-benda yang bisa diberi motor penggerak seperti pintu garasi, pintu pagar dan sebagainya. Dengan pengaturan *on* dan *off* atau buka-tutup melalui sebuah

atau beberapa *outlet* semacam *remote control*. *Internet Of Things* adalah konsep yang muncul dimana semua alat dan layanan terhubung satu dengan yang lain dengan mengumpulkan, bertukar, dan memproses data untuk beradaptasi secara dinamis. Teknologi *Internet Of Things* akan membuat sebuah rumah konvensional menjadi *Smart Home*, secara efektif semua *device* saling terhubung dengan *device* yang lain. Perangkat elektronik khususnya sensor mengalami perkembangan menjadi lebih baik untuk bekerja sesuai dengan kebutuhan manusia. Proyek tentang *Smart Home* sendiri telah banyak dilakukan diantaranya Detektor Keamanan Rumah melalui

*Telegram Messenger*, merancang sistem *Smart Home* yang dapat diakses dimana saja berbasis IoT. Dan pada proyek kali ini akan membahas tentang perancangan dan implementasi *Smart Home* menggunakan Raspberry Pi dan ESP8266. Pada proyek kali ini akan merancang dan mengimplementasikan sistem *Smart Home* dengan memanfaatkan IoT berbasis komunikasi *wireless*, sistem yang dirancang memiliki beberapa fitur diantaranya yaitu: pengontrolan lampu, pengontrolan lampu RGB, sensor pintu dan mengunci atau membuka pintu dengan dikontrol melalui aplikasi NodeRed. Pada perancangan *Smart Home* ini menggunakan NodeMCU 2 ESP8266 sebagai mikrokontroler sekaligus modul wifi dalam satu board, Raspberry Pi sebagai server MQTT *broker*, menggabungkan fitur pengontrolan kemudian *Client* menggunakan aplikasi NodeRed yang telah dilakukan sinkronisasi.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, permasalahan yang akan di jadikan bahasan pada proyek akhir ini, diantaranya:

1. Bagaimana membangun sistem client-server yang akan diterapkan pada proyek *Smart Home* ini?
2. Bagaimana merancang sistem komunikasi yang handal sehingga paket data yang dikirim oleh client diterima dengan baik oleh server?
3. Bagaimana merancang *Smart Home* dengan sistem yang mudah di implementasikan dengan harga yang relative terjangkau dan mampu berinterkoneksi satu sama lainnya?

### 1.3 Tujuan

Adapun tujuan dari proyek ini adalah merancang dan mengimplementasikan *Smart Home* untuk mengendalikan alat-alat listrik, fasilitas keamanan dan penerangan menjadi satu sistem.

### 1.4 Batasan Masalah

Agar penelitian ini dapat lebih sempurna dan mendalami. Maka perlu membatasi permasalahan dalam penelitiannya. Permasalahan yang dibatasi yaitu:

1. Fitur yang akan di sediakan yaitu pengontrolan lampu, pengontrolan lampu RGB, membuka kunci pintu, dan akses kamera secara online.

2. Membuat aplikasi menggunakan NodeRed berbasis *web browser*.
3. Komunikasi *client-server* dilakukan via WiFi.
4. *Client-server* menggunakan IP dalam satu jaringan.

## 2. LATAR BELAKANG

### 2.1 Tinjauan Pustaka

Pada bab ini akan dijelaskan mengenai tinjauan Pustaka yang menjadi acuan dalam melakukan penulisan mengenai “Perancangan dan Implementasi *Smart Home*”.

Berikut ini merupakan beberapa referensi yang berkaitan dengan proyek/penelitian tentang perangkat listrik dengan Internet Of Things (IoT) yang pernah dilakukan beberapa peneliti lain mengenai sistem ini adalah sebagai berikut:

1. (Rachman, 2017), didalam penelitian yang berjudul “*Smart Home* berbasis *Internet Of Things* (IoT)” ia membahas tentang bagaimana membuat sistem *smart home* yang dapat mendukung *multiple platform* melalui smartphone dan komputer, dan memanfaatkan modul ESP8266 sebagai web server untuk menjalankan web panel sistem. Hasil pengujian, di dalam rumah dapat dilakukan langsung dengan mengakses pada Arduino server secara intranet dan bila diakses dari luar dapat dilakukan melalui server Thinkspk melalui internet, akan tetapi diperlukan delay dalam setiap pengiriman data ke server Thinkspk.
2. (Aditya, Hafidudin, & Permana, 2015), didalam penelitian yang berjudul “Analisis dan Perancangan *Prototype Smart Home* dengan Sistem *Client Server* berbasis *Platform Android* melalui Komunikasi *Wireless*” mereka merancang sebuah prototype smart home dengan sistem client-server. Metode server yang dibuat adalah metode CGI (Control Gateway Interface) sehingga sistem dapat di akses oleh multi client.
3. (Irawan, 2019), didalam penelitian yang berjudul “Rancang Bangun Prototype *Smart Home* Menggunakan NodeMCU berbasis *Internet Of Things* (IoT)” ia membahas tentang perancangan sistem *smart home* dengan metode *prototype*. Hasil yang ia dapat adalah mengendalikan peralatan rumah seperti lampu, kipas, pintu dan kunci pintu dan dapat dikendalikan 4 secara jarak

jauh oleh pengguna dengan cara memberikan perintah atau kondisi melalui google assistant dan terkoneksi oleh mikrokontroler NodeMCU V3.

## 2.2 Dasar Teori

### 2.2.1 Smart Home

Smart Home atau juga disebut rumah pintar adalah rumah yang berbasis teknologi. Dimana teknologi dapat berfungsi untuk mengontrol atau mengatur rumah secara otomatis dari jarak jauh. Pengaturan dapat dilakukan menggunakan koneksi internet dengan smartphone maupun laptop.[1]. Sistem smart home menghubungkan seluruh alat-alat yang berada dirumah, sehingga memungkinkan penghuninya dapat mengontrol beragam fungsi seperti pencahayaan, keamanan, dan sebagainya.

### 2.2.2 Modul GSM SIM900A

Raspberry Pi (juga dikenal sebagai RasPi) adalah perangkat keras SBC (Single Board Computer) yang seukuran kartu. Raspberry Pi dapat digunakan untuk menjalankan program (Ms Office, dan sebagainya). Raspberry Pi memiliki dua model yaitu: Model A dan Model B, perbedaannya adalah di penyimpanan, model A mempunyai penyimpanan sebesar 256 MB sedangkan model B mempunyai penyimpanan sebesar 512 MB. Raspberry Pi menggunakan system on a chip (SoC) dari Broadcom BCM2835 hingga BCM 2837 (Raspberry Pi 3), juga sudah termasuk processor ARM1176JZF-S MHz bahkan 1.2 GHz 64-bit quad-core ARMv8 CPU untuk raspberry Pi 3, GPU VideoCore IV dan kapasitas RAM hingga 1 GB.[6]. Raspberry Pi tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka panjang.



**Gambar 2. 1 Raspberry Pi**

### 2.2.3 ESP8266

ESP8266 adalah modul WiFi serbaguna bersifat System on Chip (SoC). Kelebihan dari ESP8266 ini adalah dapat menjalankan peran sebagai adhoc akses poin maupun client sekaligus.[3] ESP8266 memiliki banyak varian diantaranya: ESP-01, ESP-02, ESP-03, ESP-12E dan WeMos D1 Mini. ESP8266 memiliki deep sleep mode, sehingga pengguna daya akan relatif jauh lebih efisien dibandingkan dengan modul WiFi.



**Gambar 2. 2 ESP8266**

### 2.2.4 Arduino IDE

Arduino IDE (Integrated Development Environment) adalah perangkat lunak opensource yang memungkinkan kita untuk memprogram Bahasa Arduino dalam bahasa C dan memungkinkan kita untuk menulis sebuah program secara step by step kemudian instruksi tersebut di upload ke papan Arduino.

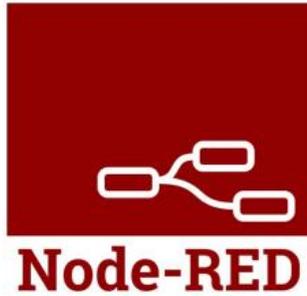


**Gambar 2. 3 Logo Arduino IDE**

### 2.2.5 NodeRed

Node-RED adalah alat pemrograman untuk menghubungkan perangkat-perangkat keras dengan cara yang baru. Node-RED memberikan editor berbasis browser yang memudahkan pengguna untuk menghubungkan flow dengan penggunaan node yang luas didalam palette yang dapat langsung diterapkan dengan satu klik.[3]. Node-RED digunakan Bersama

dengan Antares, MQTT dan platform atau protokol lainnya.



**Gambar 2. 4 Logo Node-RED**

### 2.2.6 MQTT

MQTT (Message Queuing Telemetry Transport) protokol merupakan protokol yang berjalan diatas stack TCP/IP. MQTT dirancang khusus untuk machine to machine yang tidak memiliki alamat khusus seperti Raspberry Pi, Arduino, dan ESP8266. Cara sistem kerja MQTT adalah menerapkan Publish dan Subscribe data. Dan pada penerapannya, device akan terhubung pada sebuah Broker dan mempunyai suatu 7 Topic tertentu. Broker MQTT berfungsi untuk menghandle data publish dan subscribe dari berbagai device. Beberapa contoh dari Broker yang ada seperti Mosquitto, HiveMQ, dan Mosca. Publish berfungsi untuk mengirimkan data ke subscribe, pada publisher biasanya ini adalah sebuah device yang terhubung dengan sensor tertentu. Subscribe adalah cara device untuk menerima data dari publish dan subscribe yang akan meminta data dari publish. Topic seperti halnya pengelompokan data disuatu kategori tertentu.[5]. Pada sistem kerja MQTT protokol ini, topic bersifat wajib. Setiap transaksi data antara Publish dan Subscribe harus memiliki suatu topic tertentu.

### 2.2.7 Relay

Relay adalah output yang berfungsi sebagai saklar untuk perangkat lain. Relay dikontrol dengan tegangan dari pin Arduino atau ESP8266 sehingga dapat melakukan switch. Ada 3 koneksi utama yaitu COM sebagai input dari perangkat lain, NC (Normally Close) pada keadaan biasa COM akan terhubung ke pin NC, dan NO (Normally Open) pada keadaan biasa tidak terhubung, namun saat relay mendapat tegangan dari arduino maka COM akan berpindah dari NC dan terhubung dengan NO.



**Gambar 2. 5 Relay 1 Channel**

### 2.2.8 Solenoid Door Lock

Solenoid Door Lock merupakan perangkat elektronik yang prinsip kerjanya menggunakan elektromagnetik. Solenoid door lock ini pada umumnya menggunakan 8 tegangan kerja 12 volt. Pada kondisi normal perangkat ini dalam kondisi terkunci, Ketika diberi tegangan 12 volt maka kunci akan terbuka.



**Gambar 2. 6 Solenoid Door Lock**

### 2.2.9 LED RGB Neopixel

Neopixel merupakan miniatur LED yang penuh warna, LED ini mempunyai 3 warna utama yaitu Red, Green dan Blue yang saling mengkombinasikan satu sama lain sehingga menghasilkan banyak warna. Neopixel mempunyai berbagai macam jenis diantaranya adalah strips, board dan ring.



**Gambar 2. 7 LED RGB Neopixel 7bit**

## 3. ANALISIS DAN PERANCANGAN

### 3.1 Gambaran Sistem Saat Ini

Pada sistem saat ini memiliki prinsip kerja sebagai berikut: 1. Alat sudah dalam keadaan menyala. 2. Semua alat atau komponen harus dalam satu jaringan. 3. Smartphone atau laptop memberikan perintah ke Server (Raspberry Pi) melalui aplikasi NodeRED. 4. Kemudian server akan meminta data ke client

(ESP8266) melalui MQTT protokol. 5. Dan setelah data diberikan dan diterima, maka sensor akan menyala.

**3.2 Identifikasi Kebutuhan Sistem (atau Produk)**

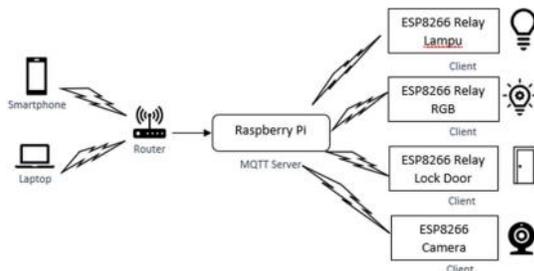
**3.2.1 Kebutuhan Fungsionalitas**

1. Sistem dapat menyalakan atau mematikan alat dengan mengirimkan perintah melalui aplikasi Node-RED.
2. Sistem dapat berfungsi ketika client dan server dalam satu jaringan.
3. Client dan server terhubung dengan protocol MQTT.

**3.2.2 Kebutuhan Non-Fungsionalitas**

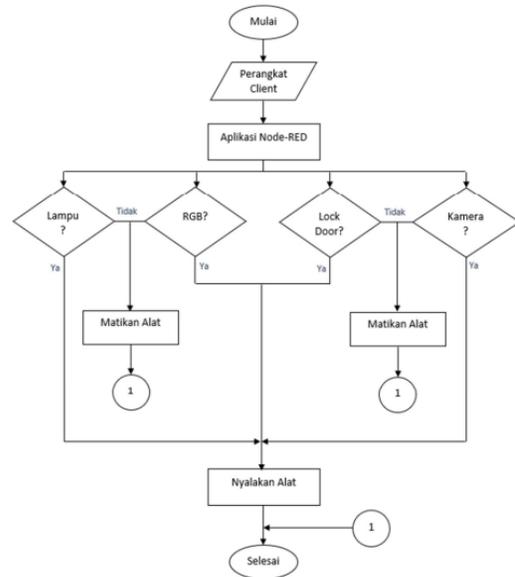
1. Raspberry Pi sebagai server MQTT broker.
2. Relay 1 channel untuk menjadi saklar pada alat.
3. ESP8266 sebagai client, pemrosesan system dan pengiriman data.

**3.3 Identifikasi Kebutuhan Sistem (atau Produk)**



**Gambar 3. 1 Model Perancangan Sistem Smart Home**

Cara kerja sistem yang dibuat pada proyek akhir ini adalah smartphone yang mengirimkan perintah kepada server melalui wireless, lalu dari server akan meminta data ke client menggunakan MQTT protokol, setelah menerima data maka lampu akan menyala sesuai perintah dari smartphone. Smartphone dan laptop menggunakan aplikasi NodeRED dengan platform web browser. Dalam pengerjaan proyek akhir ini di buat sebuah flowchart:



**Gambar 3. 2 Flowchart Pengerjaan Proyek Akhir**

**3.4 Kebutuhan Perangkat Keras dan Perangkat Lunak**

**3.4.1 Perangkat Keras**

**Table 3- 1 Kebutuhan Perangkat Keras**

No	Hardware	Fungsi	Jumlah
1	NodeMCU ESP8266	Pemrosesan sistem dan pengiriman data	4
2	Raspberry Pi	Server MQTT Broker	1
3	Relay	Menyalakan dan mematikan saklar	3

**3.4.2 Perangkat Lunak**

**Table 3- 2 Kebutuhan Perangkat Lunak**

No	Software	Fungsi
1	Arduino IDE	Pemrograman Sistem
2	Node Red	Aplikasi Smart Home berbasis web browser

**4. IMPLEMENTASI DAN PENGUJIAN**

**4.1 Implementasi Sistem**

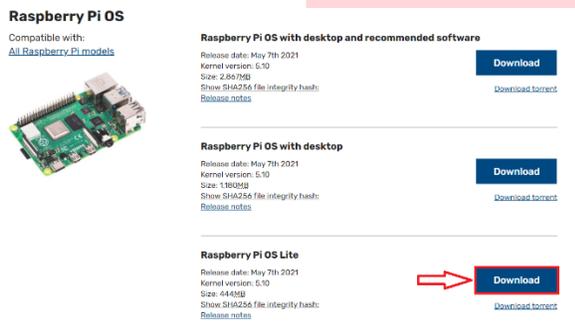
Dalam tahap implementasi terdapat beberapa hal yang harus disiapkan, diantaranya instalasi Raspberry Pi OS

(Operating System), instalasi NodeRED, instalasi Mosquitto pada Raspberry Pi, membuat kode program ESP8266, dan Mendesain grafis antarmuka pengguna. Berikut tahapan-tahapan yang harus dilakukan:

#### 4.1.1 Instalasi Raspberry Pi OS

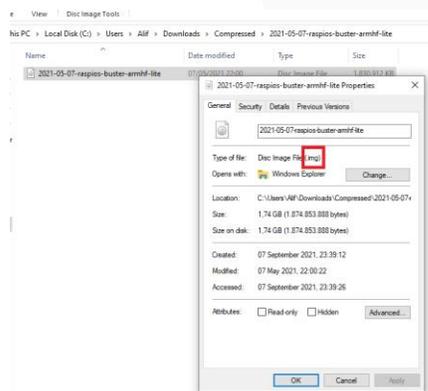
Raspberry Pi adalah sebuah komputer dan seperti komputer lainnya, ia membutuhkan system operasi yang harus di install. Raspberry Pi pada umumnya tidak memiliki memori internal, jadi sangat diperlukan kartu microSD untuk menginstal Sistem operasi.

1. Langkah pertama, yaitu download sistem operasi Raspbian lite.



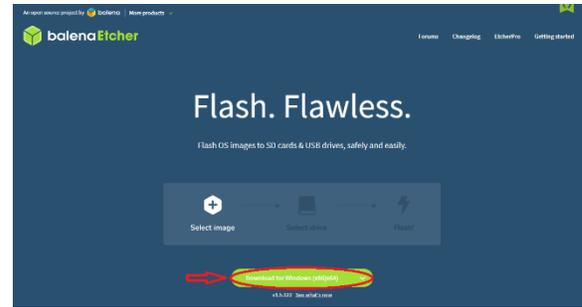
Gambar 4. 1 Raspberry Pi OS

2. Setelah download system operasi Raspbian lite, pengguna harus memiliki file .zip di folder download. Unzip dan akan menemukan file .img seperti pada gambar dibawah ini.



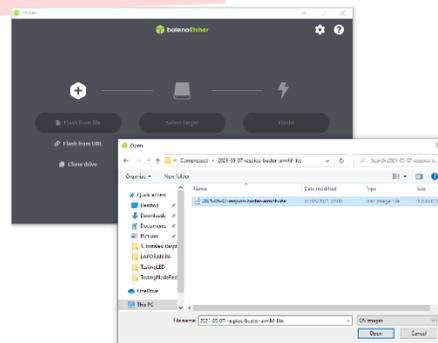
Gambar 4. 2 File Raspberry Pi OS

3. Untuk mem-flash kartu microSD dengan file .img, maka akan diperlukan aplikasi Etcher. Download terlebih dahulu aplikasi di <https://etcher.io/>



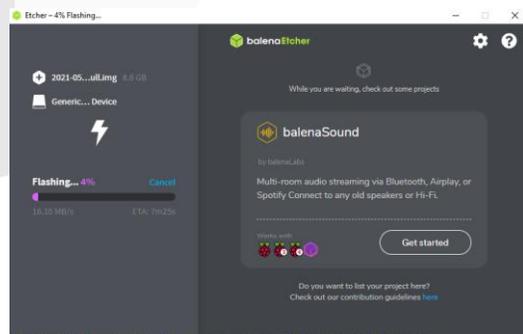
Gambar 4. 3 Website Etcher

4. Setelah download aplikasi Etcher, jalankan aplikasi tersebut untuk write file .img di microSD. Koneksikan microSD ke komputer, lalu pilih lah file Raspbian Lite .img, seperti pada gambar 4-4.



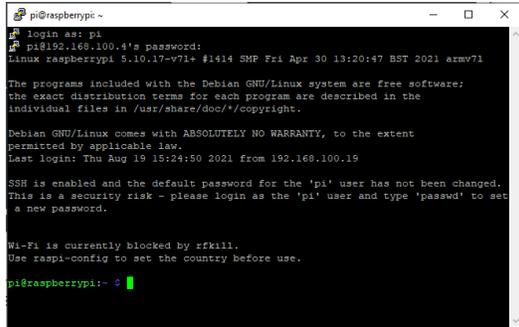
Gambar 4. 4 Masukkan File Raspbian OS

5. Pilihlah microSD sebagai perangkat. Kemudian klik Flash! untuk memulai writing file .img ke microSD. Pada tahap ini akan memakan 5 sampai 20 menit.



Gambar 4. 5 Proses Flashing Raspbian OS

6. Setelah proses flash selesai. Masukkan microSD ke Raspberry Pi, lalu pada komputer pengguna buka aplikasi putty.exe untuk meremote raspberry pi menggunakan SSH.



Gambar 4. 6 Remote Raspberry Pi menggunakan SSH

### 4.1.2 Instalasi NodeRED

Menginstal NodeRED di Raspberry Pi akan memakan waktu cepat dan mudah. Hanya membutuh kan beberapa perintah.

1. Jalankan putty.exe untuk meremote raspberry pi, lalu ketikkan perintah untuk melakukan instalasi dan upgrade NodeRED:

```
pi@raspberrypi:~ $ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

2. Untuk menjalankan NodeRED secara otomatis saat raspberry pi booting, ketikkan perintah berikut:

```
pi@raspberrypi:~ $ sudo systemctl enable nodered.service
```

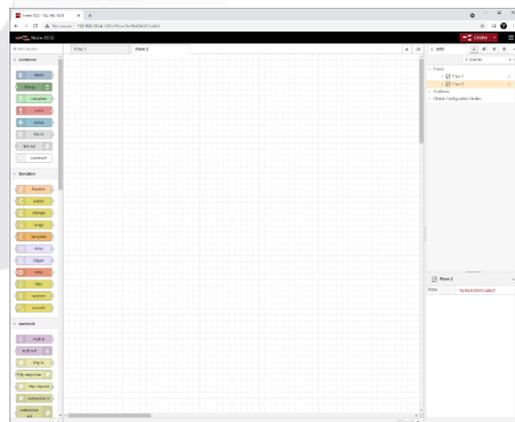
3. Kemudian restart raspberry pi sehingga NodeRED akan menjalankan otomatis.

```
pi@raspberrypi:~ $ sudo restart
```

4. Lalu test instalasi dengan mengetikkan IP address raspberry pi di web browser dan 1880 sebagai port.

```
http://192.168.100.4:1880
```

5. Akan terbuka tampilan NodeRED seperti gambar dibawah ini:



Gambar 4. 7 Tampilan NodeRED

### 4.1.3 Instalasi Mosquitto Broker

Mosquitto broker berfungsi untuk menerima semua pesan, menyaring pesan dan kemudian mempublikasikan pesan ke semua client (ESP8266).

1. Untuk instalasi mosquito broker dan membuat otomatis, ketikkan perintah berikut:

```
pi@raspberrypi:~$ sudo apt update
pi@raspberrypi:~$ sudo apt install -y mosquitto
mosquitto-clients
pi@raspberrypi:~$ sudo systemctl enable
mosquitto.service
```

Update raspberry pi terlebih dahulu, lalu lakukan instalasi seperti pada gambar dibawah ini.

```
pi@raspberrypi:~$ sudo apt update
Hit:1 https://deb.nodesource.com/node_14.x buster InRelease
Get:2 http://archive.raspberrypi.org/debian buster InRelease [32.6 kB]
Hit:3 http://raspbian.raspberrypi.org/raspbian buster InRelease
Get:4 http://archive.raspberrypi.org/debian buster/main armhf Packages [378 kB]
Fetched 410 kB in 4s (99.6 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
38 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@raspberrypi:~$ sudo apt install -y mosquitto mosquitto-clients
```

Gambar 4. 8 Instalasi Mosquitto Broker

2. Ketikkan perintah “mosquitto -v” untuk melihat atau mengetahui versi mosquitto broker.

```
pi@raspberrypi:~$ mosquitto -v
1631067941: mosquitto version 1.5.7 starting
1631067941: Using default config.
1631067941: Opening ipv4 listen socket on port 1883.
1631067941: Error: Address already in use
pi@raspberrypi:~$
```

Gambar 4. 9 Versi Mosquitto Broker

4.1.4 Desain Grafis Antarmuka Pengguna

Dalam subbab ini pengguna akan menginstal paket Node-RED yang di sebut NodeRED Dashboard. Paket Node-RED ini memungkinkan pengguna untuk menambahkan buttons, switches, charts dan sebagainya.

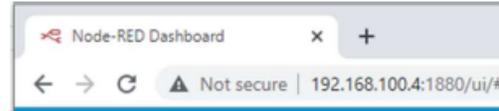
1. Langkah pertama untuk menginstal Node-RED dashboard, jalankan perintah tersebut di Raspberry Pi:

```
pi@raspberrypi:~$ node-red-stop
pi@raspberrypi:~$ cd ~/.node-red
pi@raspberrypi:~/.node-red$ npm install node-red-dashboard
```

2. Reboot raspberry pi:

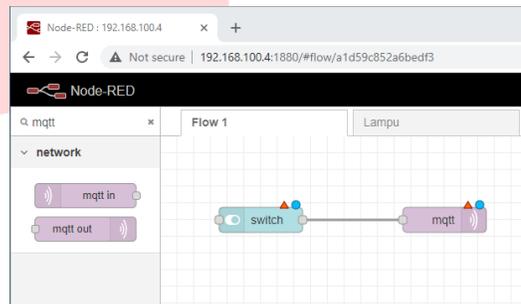
```
pi@raspberrypi:~$ sudo restart
```

3. Ketika raspberry pi kembali nyala, pengguna dapat membuka User Interface dengan memasukkan alamat IP raspberry pi di web browser diikuti “:1880/ui”



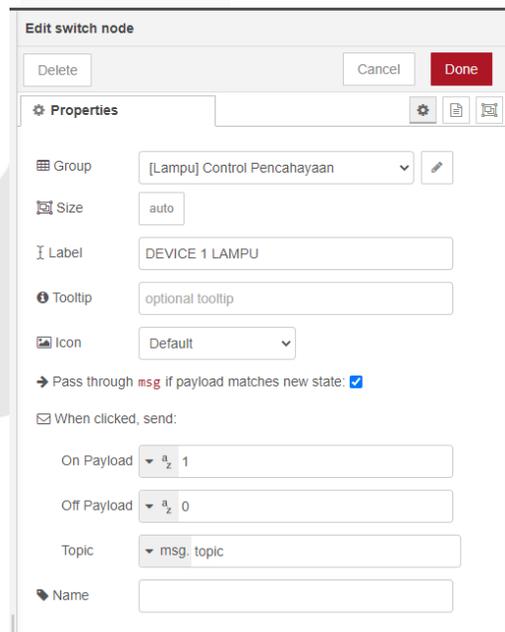
Gambar 4. 10Alamat Node-RED Dashboard

4. Setelah instalasi Node-RED dashboard selesai dan berhasil dijalankan, maka buka alamat IP raspberry pi untuk memulai membuat aplikasi smart home.
5. Cari “Switch & mqtt out” pada bagian filter nodes (Pojoek kiri atas). Pilih dan seret switch pada bagian dashboard ke area flow. Lalu hubungkan switch ke mqtt, seperti pada gambar dibawah ini:



Gambar 4. 11 Menghubungkan Switch ke MQTT

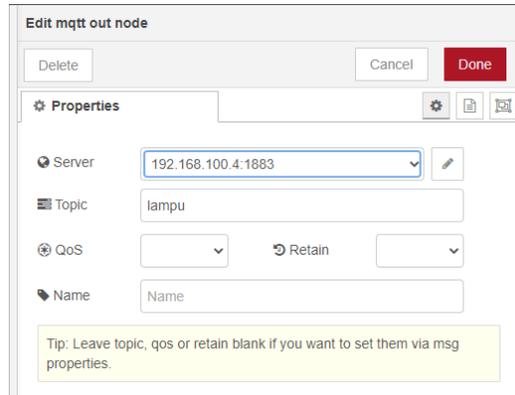
6. Kemudian klik switch dua kali untuk mengedit node switch.



Gambar 4. 12 Edit Node Switch

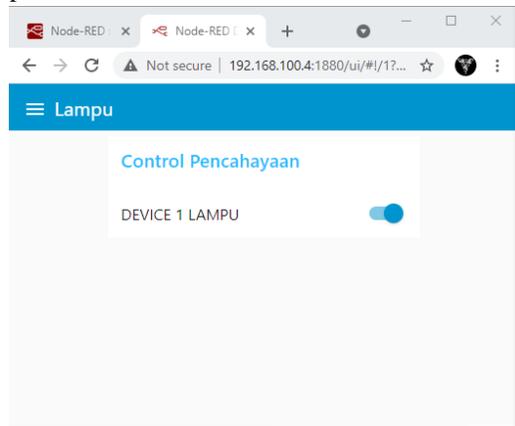
7. Klik mqtt dua kali untuk mengedit node mqtt. Pada bagian server, masukkan alamat IP raspberry pi ddiikuti “:1883”. Tahap ini adalah

cara membangun komunikasi antara raspberry pi dan client menggunakan MQTT protokol. Pastikan Topic harus sama dengan kode program ESP8266.



**Gambar 4. 13 Node MQTT Out**

8. Buka Kembali UI pada browser, untuk melihat tampilan nodered nya. Hasilnya akan seperti pada Gambar 4-14.

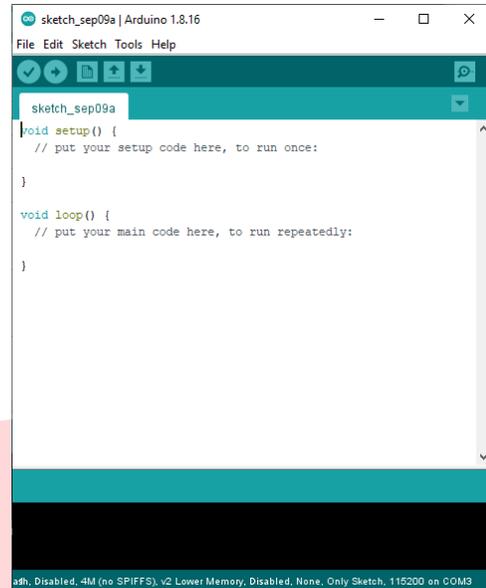


**Gambar 4. 14 Tampilan UI Node-RED**

#### 4.1.5 Membuat Kode Program ESP8266

Tahap terakhir adalah membuat kode program client atau ESP8266, didalam tahapan ini akan membahas program apa saja dan fungsi program nya.

1. Langkah pertama, buka software Arduino IDE.



**Gambar 4. 15 Tampilan Arduino IDE**

2. Sebelum membahas program nya, pastikan library dan board pubsubclient, dan ESP8266WiFi sudah terinstal.
3. Pertama, dimulai dengan loading ESP8266Wifi dan PubSubClient library.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
```

**Gambar 4. 16 ESP8266 dan PubSubClient Library**

4. Kemudian konfigurasi WiFi network (SSID, password dan MQTT broker IP address).

```
const char* ssid = "Pamoyanan(Test); //Nama SSID
const char* password = "EminDat12"; //Password WiFi
const char* mqtt_server = "192.168.100.4"; //Alamat IP Raspberry Pi
```

**Gambar 4. 17 Konfigurasi WiFi Network**

5. Mendefinisikan subscribe dan relay.

```
#define sub "lampu" //lampu sebagai topic subscribe
#define relay 0 //Relay berada di pin GPIO 0
```

**Gambar 4. 18 Definisi Sub dan Relay**

6. Menginisialisasi Client.

```
// Inisialisasi espClient
WiFiClient espClient;
PubSubClient client(espClient);
```

**Gambar 4. 19 Inisialisasi Client**

7. Menghubungkan ESP8266 ke jaringan WiFi.

```
//Mulai menghubungkan ke jaringan WiFi
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
  Serial.println(WiFi.localIP());
}
```

**Gambar 4. 20 Setup WiFi**

8. Fungsi ini dijalankan saat perangkat mempublish pesan ke topik subscribe ESP8266.

```
void callback(String topic, byte* message, unsigned int length) {
  Serial.print("Message arrived on topic: ");
  Serial.print(topic);
  Serial.print(". Message: ");
  String messageTemp;
  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
}
```

**Gambar 4. 21 Fungsi Callback**

9. Baris kode berikutnya adalah cara menghidupkan dan mematikan relay. Ketika data nya 1 maka relay akan hidup dan ketika data nya berubah menjadi angka 0, maka relay akan mati.

```
//Menghidupkan dan mematikan Relay
if(topic=="sub"){
  Serial.print("Berubah ke ");
  if(messageTemp == "1"){
    digitalWrite(relay, HIGH);
    Serial.print("On");
  }
  else if(messageTemp == "0"){
    digitalWrite(relay, LOW);
    Serial.print("Off");
  }
}
```

**Gambar 4. 22 Menghidupkan dan Mematikan Relay**

10. Fungsi dari “reconnect()” adalah menghubungkan Kembali ESP8266 ke MQTT broker. Ketika client berhasil terhubung MQTT, maka client akan memsubscribe ke topic. Jika tidak terhubung maka client akan mengulang kembali sampai client terhubung ke MQTT.

```
// Menghubungkan atau menghubungkan kembali ESP8266 ke MQTT Broker
void reconnect() {
  // Loop sampai terhubung kembali
  while (!client.connected()) {
    Serial.print("Mencoba koneksi MQTT...");
    if (client.connect("ESP8266Client")) {
      Serial.println("connected");
    }
    // Subscribe atau resubscribe ke topic
    client.subscribe(sub);
  }
  else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Tunggu 5 detik sebelum mengulang kembali
    delay(5000);
  }
}
```

**Gambar 4. 23 Menghubungkan ESP8266 ke MQTT Broker**

11. Fungsi “setup()” adalah menetapkan pin GPIO ESP8266 sebagai OUTPUT, memulai komunikasi serial, menghubungkan ESP8266 ke router, memasang MQTT broker, dan menyetel “callback”.

```
void setup() {
  pinMode(relay, OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}
```

**Gambar 4. 24 Program Setup**

12. Dan yang terakhir adalah “loop()”. Fungsi ini untuk memeriksa apakah ESP8266 terhubung ke MQTT broker. Jika tidak terhubung, maka ESP8266 akan mencoba untuk menghubungkan Kembali.

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}
```

**Gambar 4. 25 Program Loop**

13. Setelah program selesai, Upload program tersebut ke ESP8266.

## 4.2 Pengujian

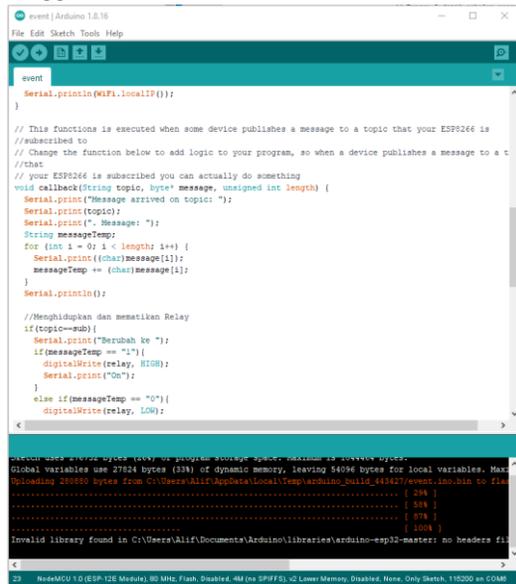
Tahap selanjutnya adalah tahap pengujian untuk mengetahui cara sistem kerja dari implementasi Smart Home ini. Pengujian dilakukan di dalam ruangan. Sistem yang akan di uji adalah mengkoneksikan atau menghubungkan ESP8266 ke jaringan WiFi dan menjalankan alat atau komponen menggunakan aplikasi Node-RED (Lampu LED, lampu RGB, Door lock dan kamera).

### 4.2.1 Menghubungkan ESP8266 ke Jaringan WiFi

Hal yang paling penting pada proyek ini adalah menghubungkan ESP8266 ke WiFi, jika ESP8266 tidak bisa dihubungkan ke WiFi, maka proyek smart home ini tidak dapat dijalankan. Tujuannya adalah untuk mengkoneksikan client ke server. Dapat disimpulkan client dan server terhubung menggunakan WiFi.

1. Pada Gambar 4-20 adalah contoh kode untuk menghubungkan ESP8266 ke jaringan WiFi.

2. Upload program ke ESP8266. Tunggu upload hingga 100%.



Gambar 4. 26 Upload Program

3. Ketika selesai upload, buka serial monitor untuk melihat apakah ESP8266 sudah terhubung. Jika terhubung, maka akan terlihat seperti pada gambar dibawah ini:



Gambar 4. 27 Pengujian Koneksi WiFi

#### 4.2.2 Menjalankan Alat menggunakan Aplikasi Node-RED

Tujuan dari pengujian ini adalah dapat menyalakan atau mematikan alat dengan menggunakan aplikasi node-RED pada smartphone.

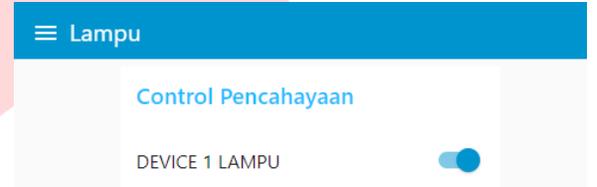
1. Lampu LED

Lampu LED dipasang di tembok kamar, lampu ini bertujuan untuk penerangan atau pencahayaan pada ruangan.



Gambar 4. 28 Pemasangan Lampu

Apabila tombol di tekan pada aplikasi Node-RED, maka data yang diberikan ke client atau ESP8266 akan berubah menjadi 1 yaitu lampu menyala.



Gambar 4. 29 Tombol Lampu pada Aplikasi



**Gambar 4. 30 Lampu Menyala**

2. Door Lock dan Kamera

Door lock dan kamera berfungsi untuk keamanan, ketika ada orang yang masuk maka bisa dilihat dari aplikasi Node-RED. Door lock berfungsi untuk mengunci atau membuka pintu apabila tombol di tekan.



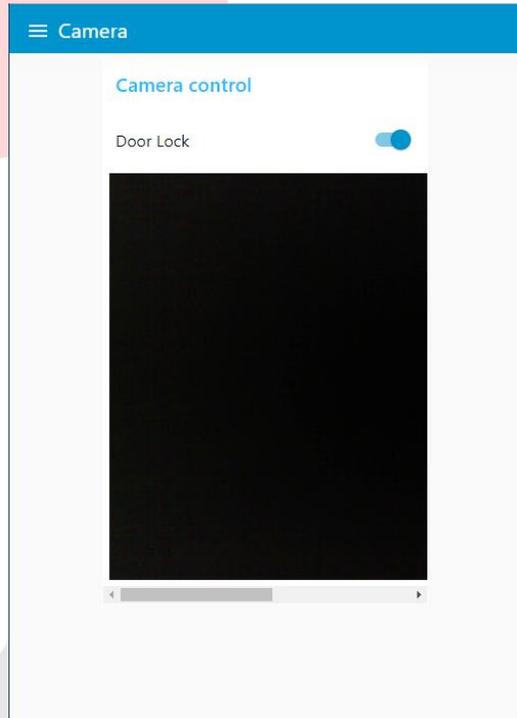
**Gambar 4. 31 Pemasangan Door Lock**



**Gambar 4. 32 Pemasangan Kamera**



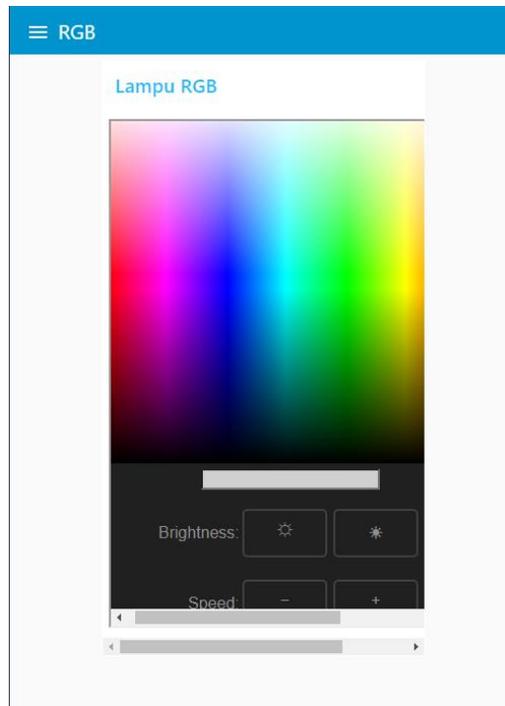
**Gambar 4. 33 Kondisi Door Lock Mengunci**



**Gambar 4. 34 Pengaturan Kamera dan Door Lock**

3. Lampu RGB

RGB ini digunakan sebagai lampu tidur, karena lampu RGB dapat mengatur penerangan atau warna yang sesuai. Untuk mengatur warna nya dapat diatur di aplikasi Node-RED.



**Gambar 4. 35 Pengaturan Lampu RGB**



**Gambar 4. 36 Kondisi Lampu RGB ketika Menyala**

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah melakukan pengujian terhadap implementasi Smart Home, maka dapat disimpulkan bahwa:

1. Komunikasi antara client dan server berjalan lancar tanpa adanya hambatan.
2. Alat bekerja berdasarkan kendali aplikasi Node-RED yang sudah dibuat dengan cara mengirimkan logika 1 dan 0 pada ESP8266 sehingga dapat mematikan dan menyalakan modul relay.
3. Aplikasi Node-RED yang dibuat bisa digunakan disemua smartphone.

### 5.2 Saran

Setelah melakukan perancangan dan penelitian, penulis dapat memberikan saran sebagai berikut:

1. Sebaiknya dilakukan pengembangan lagi terhadap implementasi smart home ini dengan menambahkan beberapa komponen, seperti motion sensor, sensor suara, dan lain sebagainya.
2. Dalam perancangan mekanik sebaiknya dilakukan dengan rapi dan teliti agar hasilnya akan lebih menarik perhatian yang melihatnya.
3. Dilakukan pengembangan lagi seperti menambahkan database untuk penyimpanan rekaman kamera.

## 6. REFERENSI

- [1] Rumah.com. (2020) Mengenal Smart Home System. [Online]. [HYPERLINK "https://www.rumah.com/panduan-properti/smart-home-37050"](https://www.rumah.com/panduan-properti/smart-home-37050)
- [2] SiniArduino. (2016) Modul WiFi ESP8266. [Online]. [HYPERLINK "https://www.sinauarduino.com/artikel/esp8266/"](https://www.sinauarduino.com/artikel/esp8266/).
- [3] Antares. (2018) Node-RED Pengiriman dan Penerimaan Data. [Online]. [HYPERLINK "https://antares.id/id/node-red.html"](https://antares.id/id/node-red.html)
- [4] Warriornux. (2021) ESP8266 Arduino IDE – Setting RGB Neopixel WS2812. [Online]. [HYPERLINK "https://www.warriornux.com/setting-neopixelws2812-esp8266/"](https://www.warriornux.com/setting-neopixelws2812-esp8266/)
- [5] RESLabSiskom. (2018). Mengenal MQTT Protokol untuk IoT. [Online]. [HYPERLINK "http://reslab.sk.fti.unand.ac.id/index.php?option=com\\_k2&view=item&id=229:mengenal-mqtt-protokol-untuk-iot&Itemid=303"](http://reslab.sk.fti.unand.ac.id/index.php?option=com_k2&view=item&id=229:mengenal-mqtt-protokol-untuk-iot&Itemid=303)
- [6] Bpptik. (2014). Mengenal Lebih Dekat Raspberry Pi. [Online]. [HYPERLINK "https://bpptik.kominfo.go.id/2014/04/14/410/mengenal-lebih-dekatraspberry-pi/"](https://bpptik.kominfo.go.id/2014/04/14/410/mengenal-lebih-dekatraspberry-pi/)