

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. Analisis Sistem**

Bab ini membahas mengenai masalah yang dihadapi, strategi penyelesaian masalah yang dicapai. Pada bab ini juga diajarkan mengenai implementasi serta kekurangan dan kelebihan dari perangkat lunak yang dikembangkan.

#### **4.2. Analisis Masalah**

Adapun analisis masalah pada penelitian ini adalah penyajian informasi mengenai implementasi *SmartHome* dengan Konsep *Internet Of Things* (IOT) berbasis *SmartPhone* dimana sebuah rumah bisa dikendalikan alat elektroniknya dengan *SmartPhone* sehingga memudahkan manusia. Maka memerlukan suatu penelitian dan teknologi yang tepat untuk mengetahui informasi yang sedang terjadi danb peralatan yang sedang aktif. Pada kasus seperti ini memerlukan suatu pemecahan masalah yang tepat seperti implementasi *SmartHome* dengan konsep *Internet Of Things* (IOT) berbasis *SmartPhone*.

#### **4.3. Analisis Kebutuhan**

Dari hasil analisa masalah manusia yang saya dapat adalah perancangan *SmartHome* dengan konsep *Internet Of Things* (IOT) berbasis *SmartPhone* dengan sistem yang efisien mempermudah pengguna rumah untuk mengontrol alat elektronik menggunakan *SmartPhone*. Oleh karena itu memerlukan perancangan *SmartHome* dengan konsep *Internet Of Things* (IOT) berbasis *SmartPhone* sangat membutuhkan perangkat lunak dan perangkat keras yang mendukung.

##### **4.3.1. Kebutuhan Perangkat Keras (Hardware)**

Untuk melakukan perancangan *SmartHome* dengan konsep *Internet Of Things* (IOT) berbasi *SmartPhone* disarankan untuk menggunakan perangkat keras dengan spesifikasi dibawah ini:

**Tabel 4.1** Analisis Perangkat Keras

No	Alat	Keterangan
1	NodeMCU ESP8266	Perangkat mikrokontroler seperti Arduino yang dapat terhubung langsung dengan <i>WiFi</i>
2	Nodem Board Lolin	Papan <i>board</i> penghubung agar memudahkan untuk menyambungkan kabel jumper
3	Relay	Pengatur arus listrik
4	Kabel Jumper	digunakan untuk menghubungkan dua titik atau lebih dan dapat juga untuk menghubungkan 2 komponen elektronika.
5	Lampu	Penghasil Cahaya
6	Kontak Colokan	mendistribusikan energi listrik dari instalasi rumah ke televisi, radio, rice cooker, mesin cuci dan alat elektronik lainnya.
7	Power Suply 12V 3 3A	Penyuplai power
8	Power Adaptor	penyuplai power

#### 4.3.2. Kebutuhan Perangkat Lunak (*Software*)

Adapun dari identifikasi masalah yang ditemui dapat dikemukakan dan didaftarkan kebutuhan akan perangkat lunak yang digunakan dalam implemtasi *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone* yaitu:

**Tabel 4.2** Analisis Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	<i>Windows 10 (ten)</i> 64bit
2	Arduino	IDE 1.8.19
3	Blynk	Pengontrol/remot
4	Fritzing	Untuk mendesain Rangkaian

#### 4.4. Analisis Kelayakan Sistem

Analisis kelayakan sistem bertujuan untuk menilai suatu tahapan pembuatan sistem yang akan dibuat, sekaligus untuk mengetahui apakah sistem tersebut layak atau tidak untuk digunakan. Pada proses penilaian tersebut dibagi

menjadi beberapa aspek yang membutuhkan pertimbangan tertentu dalam memutuskannya. Agar bisa mempermudah mengontrol atau mengendalikan peralatan elektronik menggunakan *SmartPhone*.

#### 4.4.1. Kelayakan Teknologi

Berdasarkan analisa kelayakan teknologi maka secara implemntasi *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone* ini memenuhi syarat aspek teknologi yang cukup baik. Sehingga layak untuk di implementasikan penilaian secara teknik dapat dibuat dalam bentuk tabel dibawah ini:

**Tabel 4.3** Analisis Kelayakan Teknologi

No	Item Penilaian	Keterangan
1	Teknologi Mikrokontroler	Untuk kelayakan teknologi ini cukup baik untuk digunakan. Karena teknologi ini bisa mempermudah pekerjaan manusia dalam melakukan pekerjaan rumah seperti menghidup atau mematikan alat elektronik yang ada dirumah.
2	Teknologi <i>SmartHome</i> berbasis <i>SmartPhone</i>	Dengan adanya teknologi <i>SmartHome</i> berbasis <i>SmartPhone</i> mempermudah mengontrol peralatan elektronik dirumah yang sudah kita implementasikan.

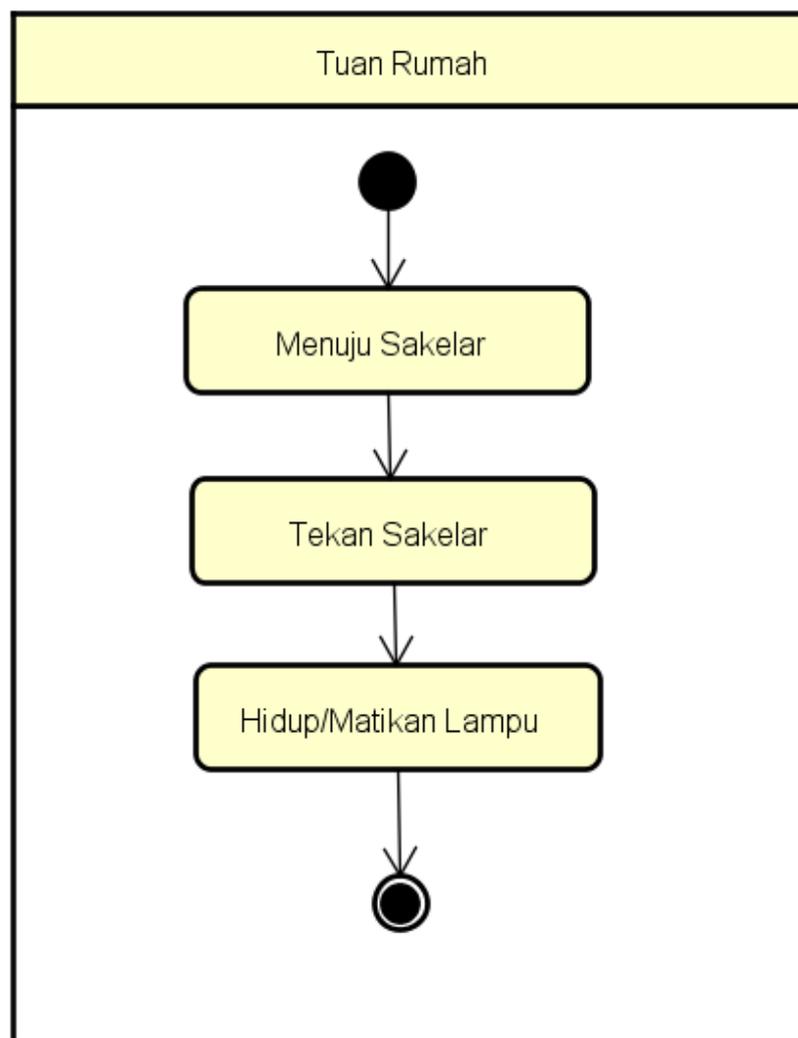
#### 4.4.2. Kelayakan Operasional

Dari segi kelayakan operasional, alat yang dibuat ini mempunyai teknik operasional yang sangat mudah. Sehingga dapat membantu pengguna dalam memonitoring peralatan elektronik yang sudah kita rangkai. Semua itu agar mempermudah tuan rumah untuk memonitoring peralata elektronik dengan mudah dan pemaiaakaian listrik sedikit lebih hemat.

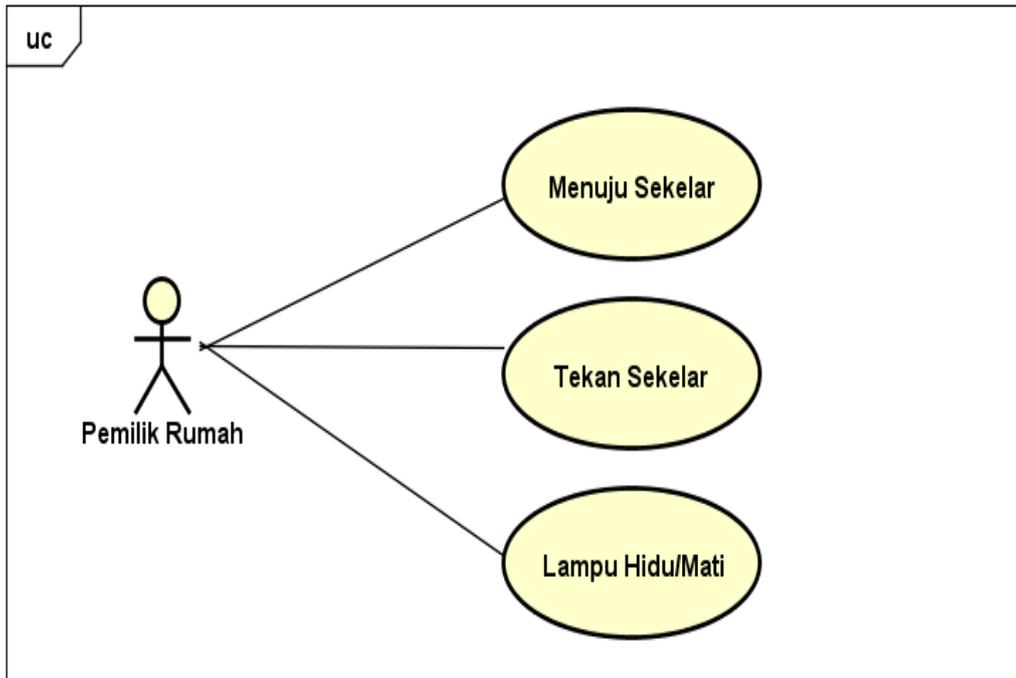
#### 4.5. Analisis Sistem Berjalan

Dalam Perancangan *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone* pada sistem ini akan menjelaskan proses menghidupkan atau mematikan alat elektronik secara manual, inilah gambaran yang akan dilakukan oleh manusia itu sendiri.

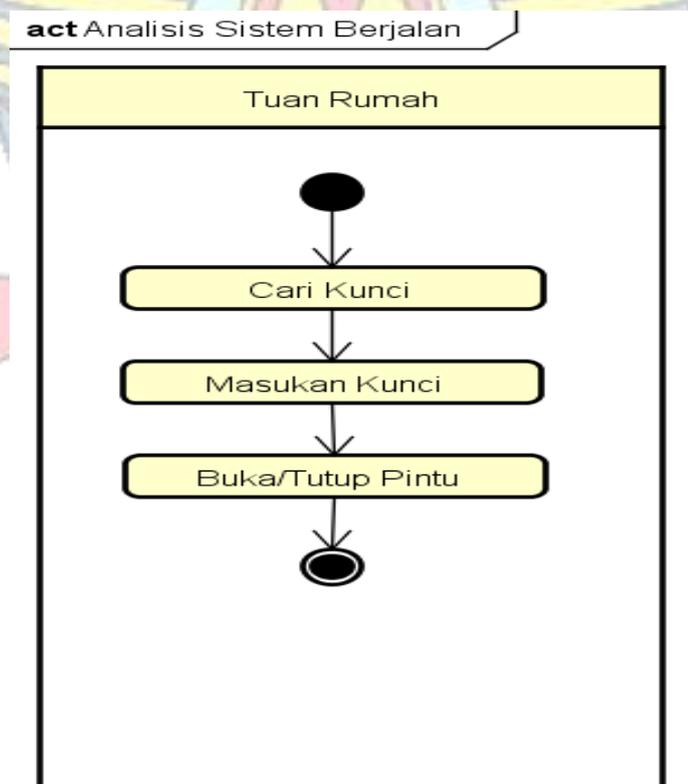
act Analisis Sistem Berjalan



**Gambar 4.1** Activity Diagram Sistem Berjalan Lampu

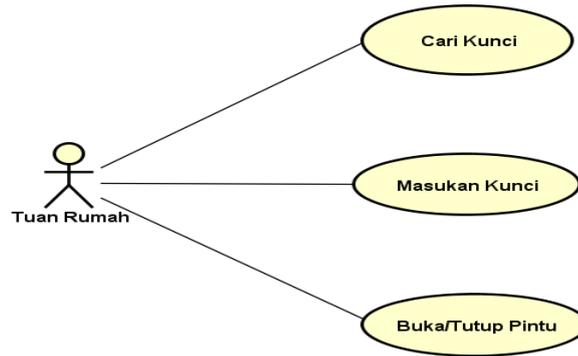


**Gambar 4.2** Use Case Diagram Sistem berjalan Nyalakan Lampu

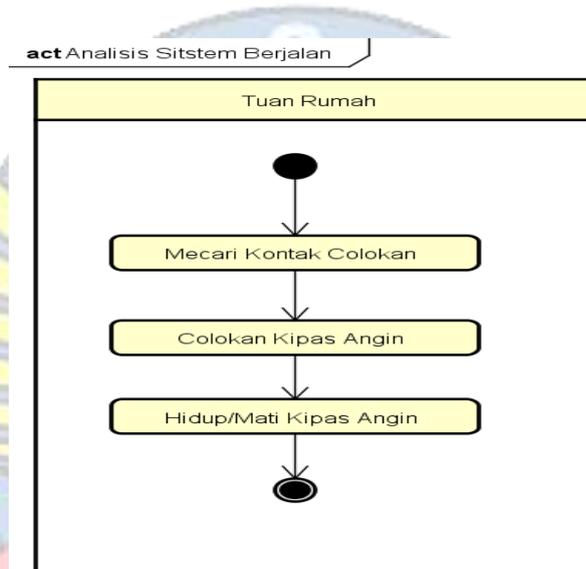


**Gambar 4.3** Activity Diagram Sistem Berjalan Pintu

uc

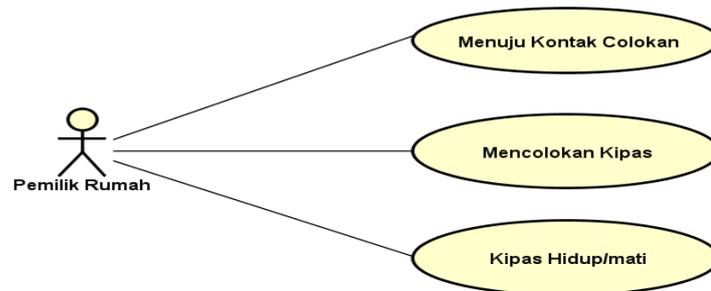


Gambar 4.4 Use Case Diagram Sistem berjalan Pintu



Gambar 4.5 Activity Diagram Sistem Berjalan kontak Colokan

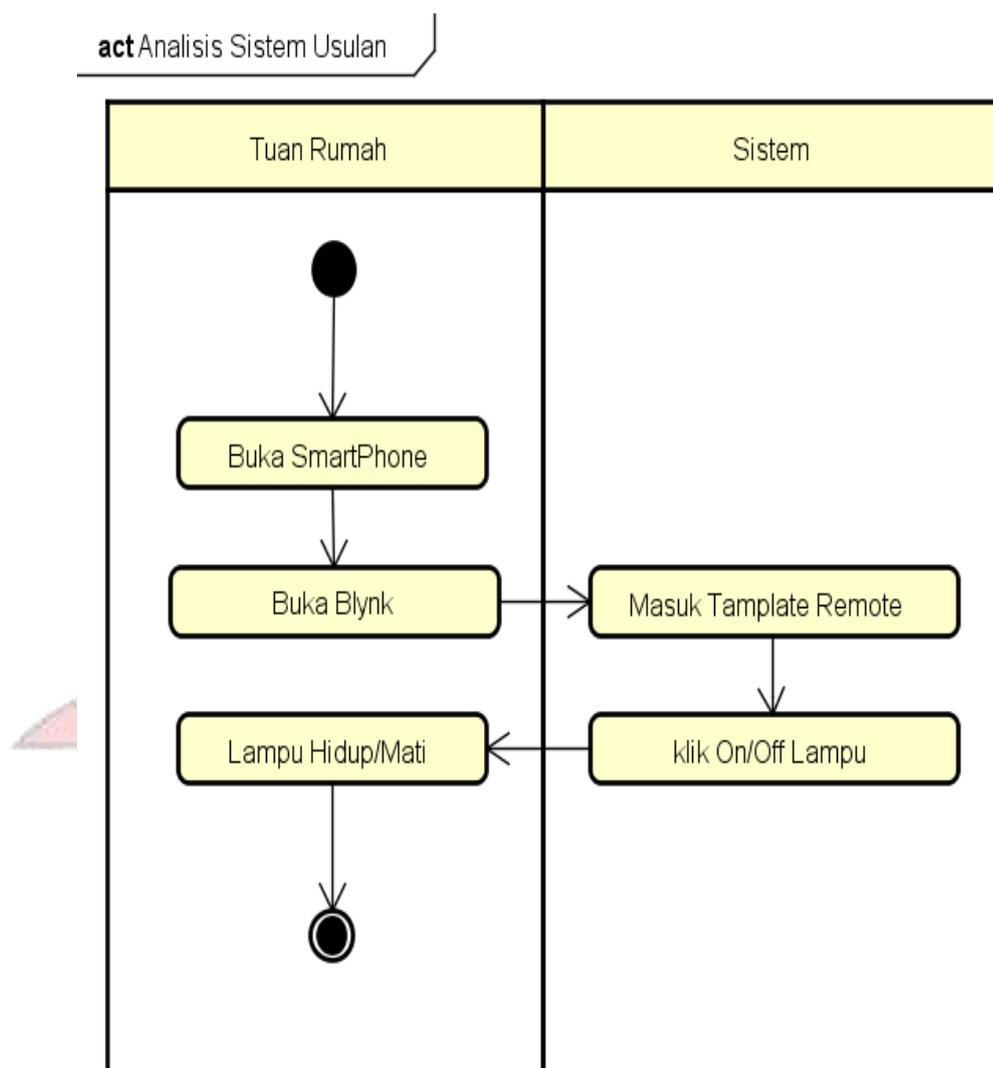
uc



Gambar 4.6 Use Case Diagram Sistem berjalan Kontak Colokan

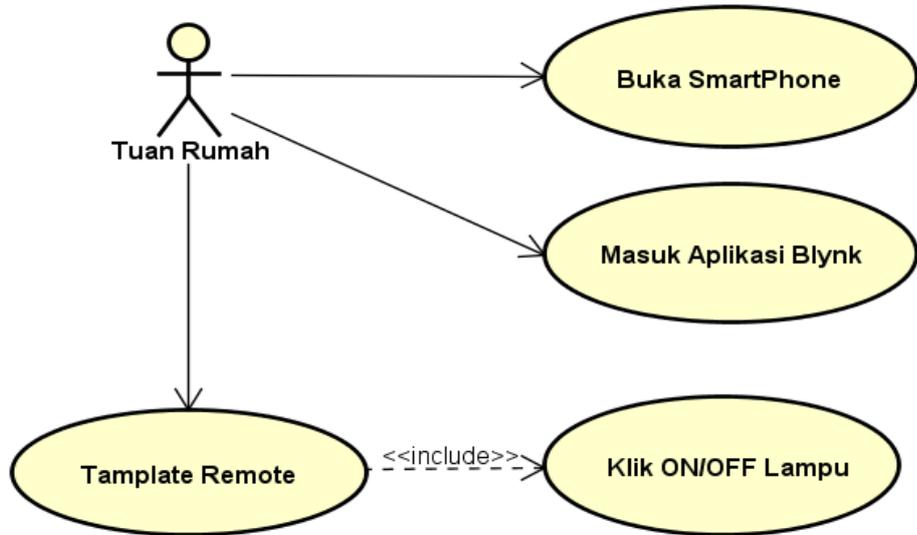
#### 4.6. Analisis Sistem Usulan

Dalam perancangan *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone* pada sistem sebelumnya dimana pemilik rumah biasanya menghidupkan atau mematikan alat elektronik secara manual kini dilakukan dengan hanya menggunakan *SmartPhone* kita untuk memudahkan menghidupkan atau mematikan alat elektronik dirumah kita. Bisa dilihat gambar dibawah ini:



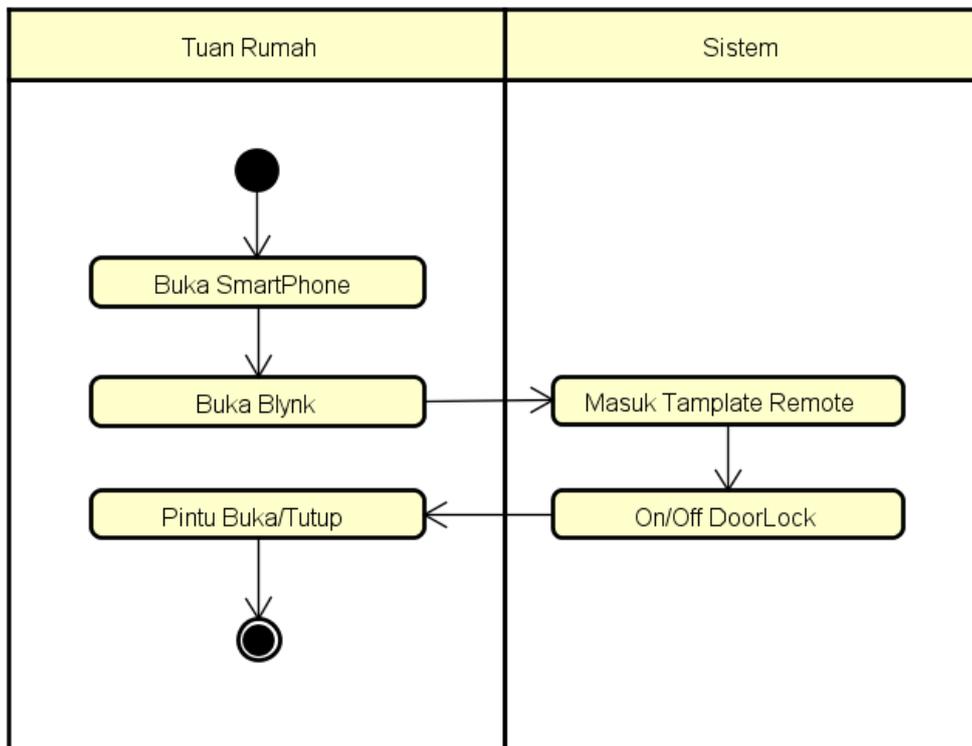
**Gambar 4.7** Activity Diagram Sistem Usulan Lampu

uc

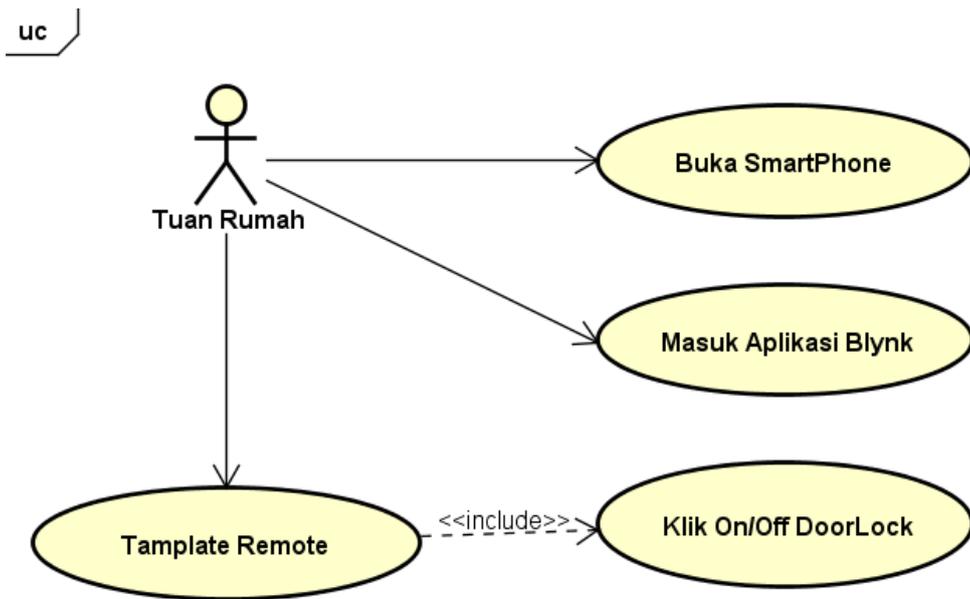


Gambar 4.8 Use Case Diagram Sistem Usulan Nyalakan Lampu

act Analisis Sistem Usulan

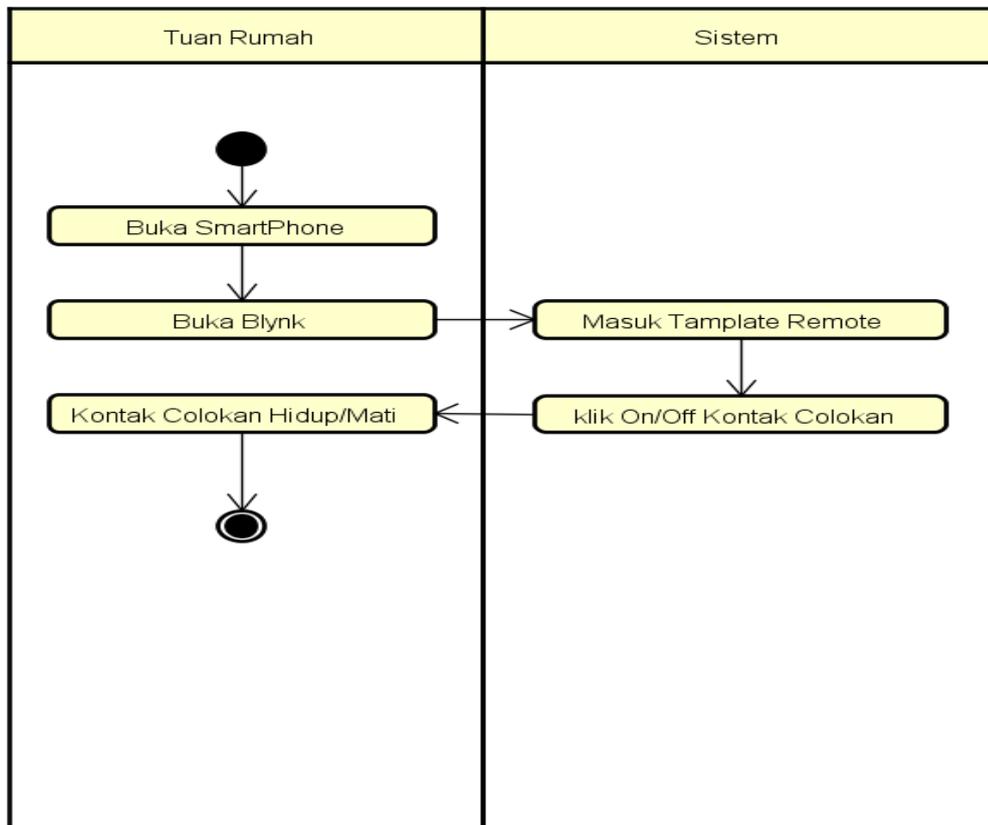


Gambar 4.9 Activity Diagram Sistem Usulan Pintu

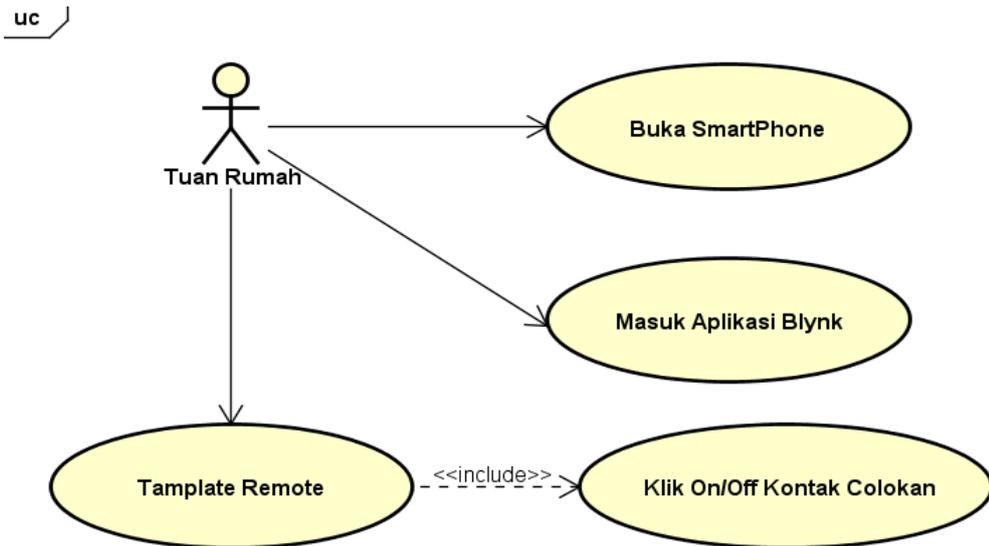


**Gambar 4.10** Use Case Diagram Sistem Usulan Pintu

act Analisis Sistem usulan



**Gambar 4.11** Activity Diagram Sistem Usulan Kontak Colokan



**Gambar 4.12** Use Case Diagram Sistem Usulan Kontak Colokan

#### 4.7. Rancangan

Tetap yang paling penting dalam perancangan adalah membuat didalam rangkaian, kemudian memilih komponen dengan karakteristik yang sesuai dengan kebutuhan. Untuk implemntasi *SmartHome* dengan konsep *Internet Of Things (IOT) berbasis SmartPhone* ini diperlukan data referensi serta petunjuk lain yang dapat membantu dalam mengetahui spesifikasi dari komponen tersebut sehingga komponen yang digunakan merupakan pilihan yang tepat bagi alat yang akan dibuat.

##### 4.7.1. Rancangan Perangkat Keras

Perancangan perangkat keras berfungsi memberikan gambaran perangkat yang kita gunakan untuk mengimplementasikan sistem mikrokontroler ini. Perangkat yang dimaksud dibagi menjadi beberapa tahapan yaitu:

a. Diagram *Block Proses* Download Data

Dari gambar dibawah menjelaskan bahwa pembuatan program dilakukan dikomputer dengan membuat *script* pada *Software* yang telah didownload. Setelah itu program melwati proses transfer data dari program ke mikrokontroler melalui kabel downloader. Mikrokontroler

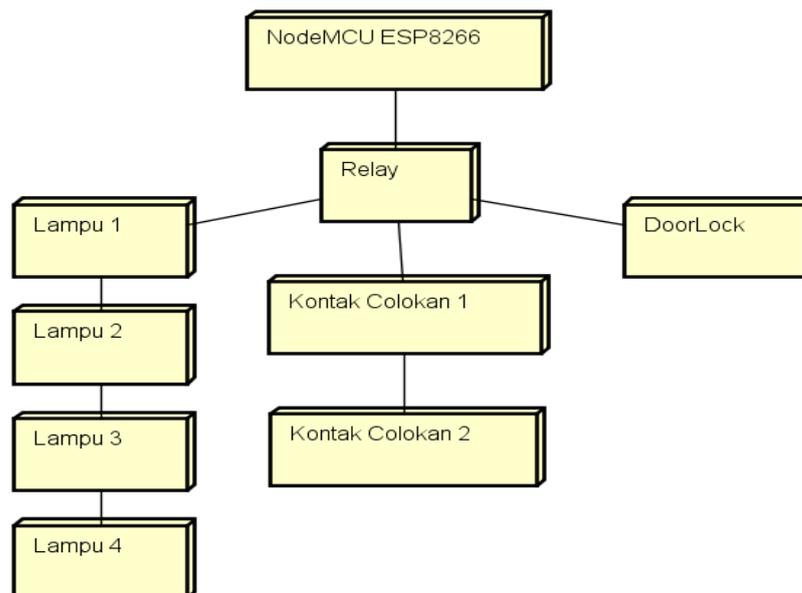
akan memproses data sesuai perintah *script* yang dibuat. Data atau perintah akan diteruskan ke perangkat mikrokontroler lainnya



**Gambar 4.13** Diagram Proses Download

b. Rancangan Keseluruhan Rangkaian

Rancangan keseluruhan rangkaian adalah rancangan keseluruhan alat yang terhubung dengan mikrokontroler yang dibuat dalam bentuk rangkaian ini terdiri dari module NodeMCU ESP8266, modul relay, lampu, Kontak Colokan dan Doorlock

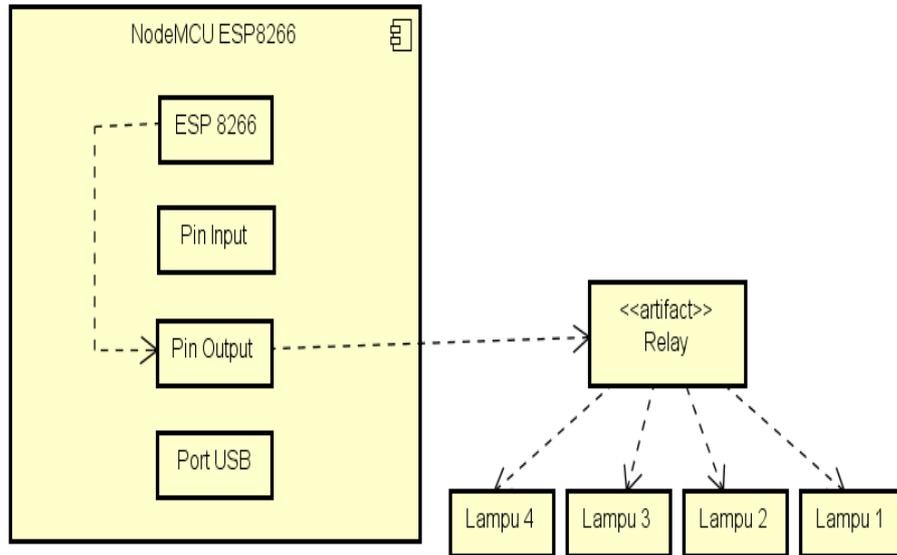


**Gambar 4.14** Komponen keseluruhan rangkaian

c. Rancangan Lampu

Rancangan lampu yang dihubungkan dengan modul NodeMCU ESP8266 dan Relay.

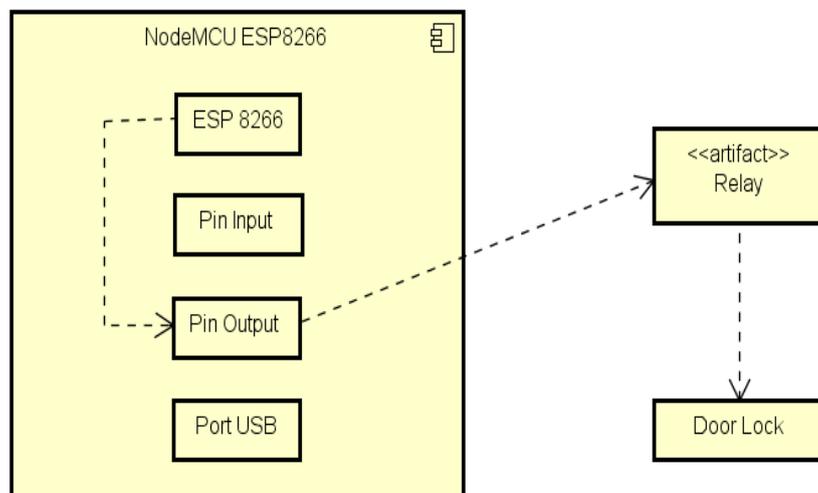
cmp



**Gambar 4.15** Komponen Diagram lampu

- d. Rancangan *DoorLock*  
Rangkaian *DoorLock* yang dihubungkan dengan modul NodeMCU ESP8266 dan Relay

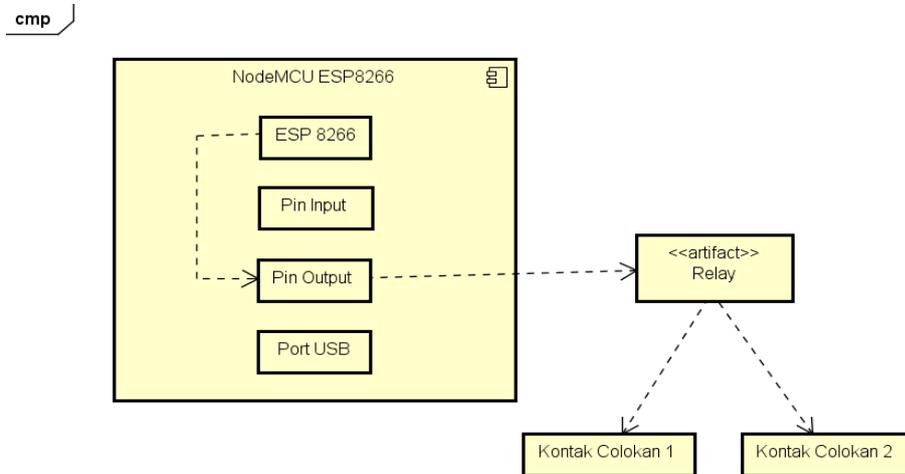
cmp



**Gambar 4.16** Komponen Diagram *DoorLock*

e. Rancangan Kontak Colokan

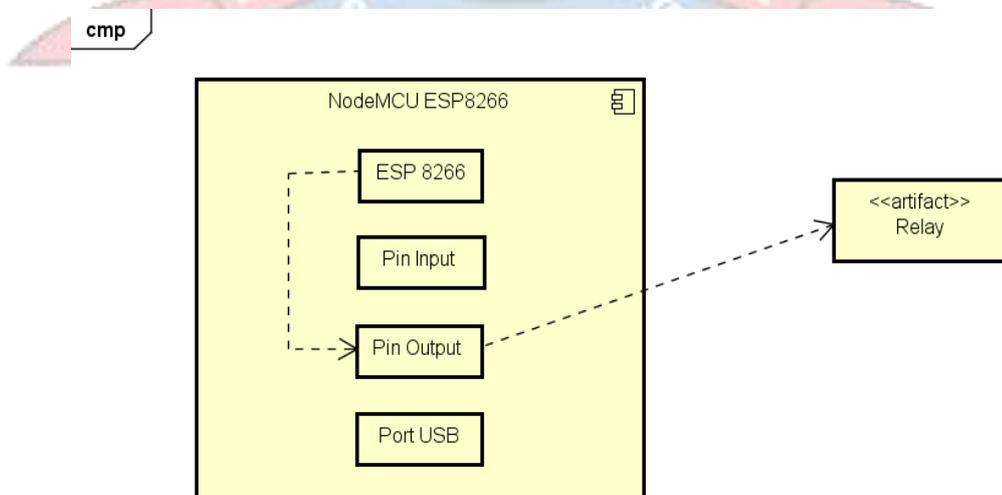
Rangkaian Kontak Colokan yang dihubungkan dengan modul NodeMCU ESP8266 dan Relay.



**Gambar 4.17** Komponen Diagram Kontak Colokan

f. Rancangan *Relay*

Rangkaian *relay* dengan NodeMCU ESP8266 digunakan untuk menghubungkan atau memutuskan arus listrik kepada semua yang terhubung. Rangkaian ini dirancang sesuai program mikrokontroler arduino, dimana terdapat sinyal control dari mikrokontroler.



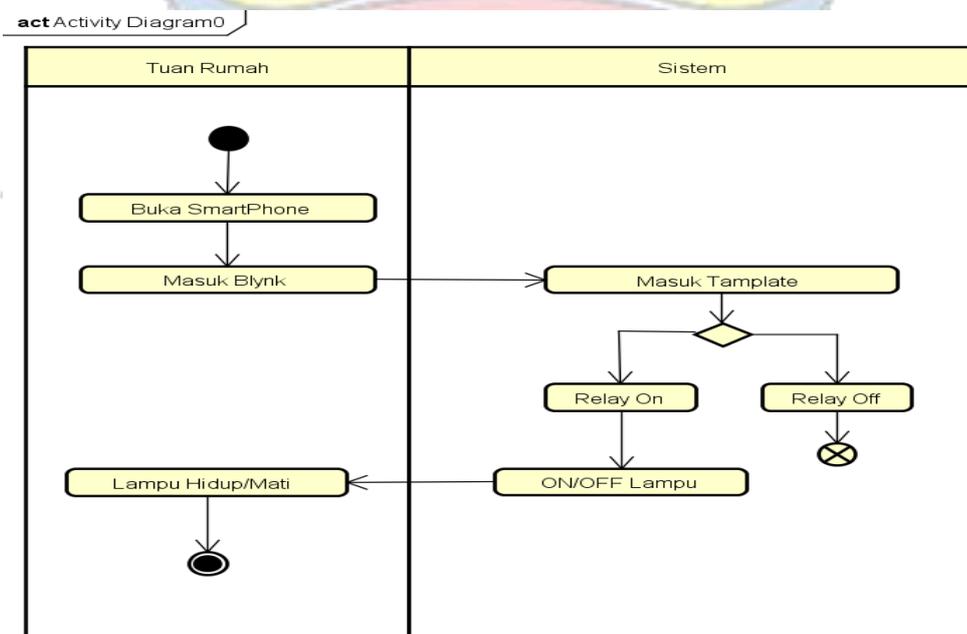
**Gambar 4.18** Komponen Diagram relay

#### 4.7.2. Rancangan Perangkat Lunak

Pada sistem ini implementasi perangkat lunak dapat dilakukan menjadi dua bagian yaitu implementasi perangkat lunak pada modul pengendali utama papan mikrokontroler dengan menggunakan program arduino IDE versi 1.8.19 windows berbasis bahasa C dan perangkat lunak BLYNK untuk memonitoring perangkat mikrokontroler yang sudah terhubung.

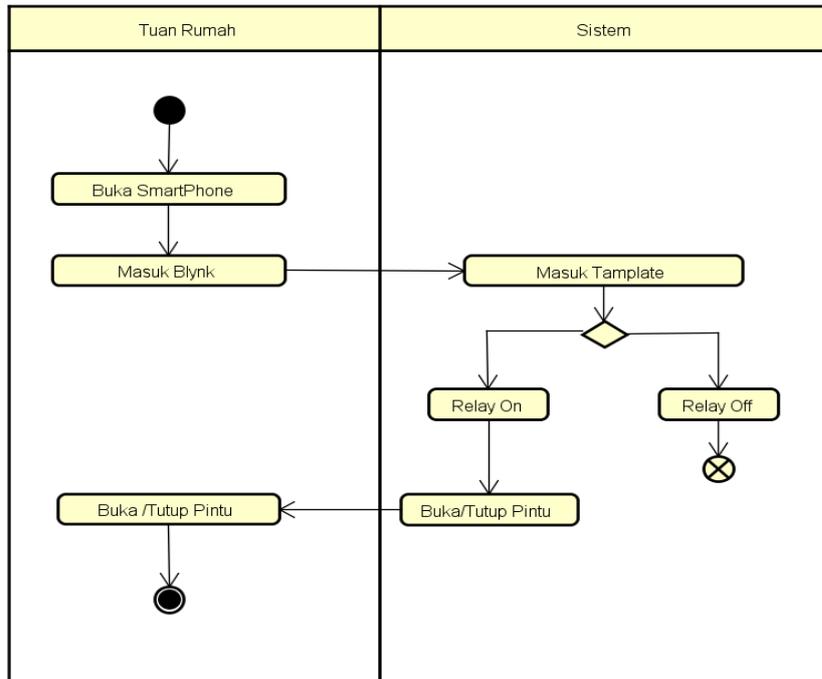
##### 4.7.2.1. Activity Diagram

Dalam implementasi *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone* dimulai dari remote pengontrol aplikasi BLYNK kemudian relay akan aktif dan mengirim perintah ke mikrokontroler NodeMCU ESP8266. NodeMCU ESP8266 sendiri adalah jantung dari sub sistem transmiter karena dari sinilah semua data yang masuk diolah dan diperintahkan untuk dikirimkan ke *SmartPhone User*. Semua data yang dikirimkan harus terlebih dahulu melewati rangkaian *interface*. Selanjutnya data yang telah disimpan pada mikroprosesor berupa perintah akan dikirimkan ke *SmartPhone user*. Berikut adalah *Activity diagram* pada sistem baru:



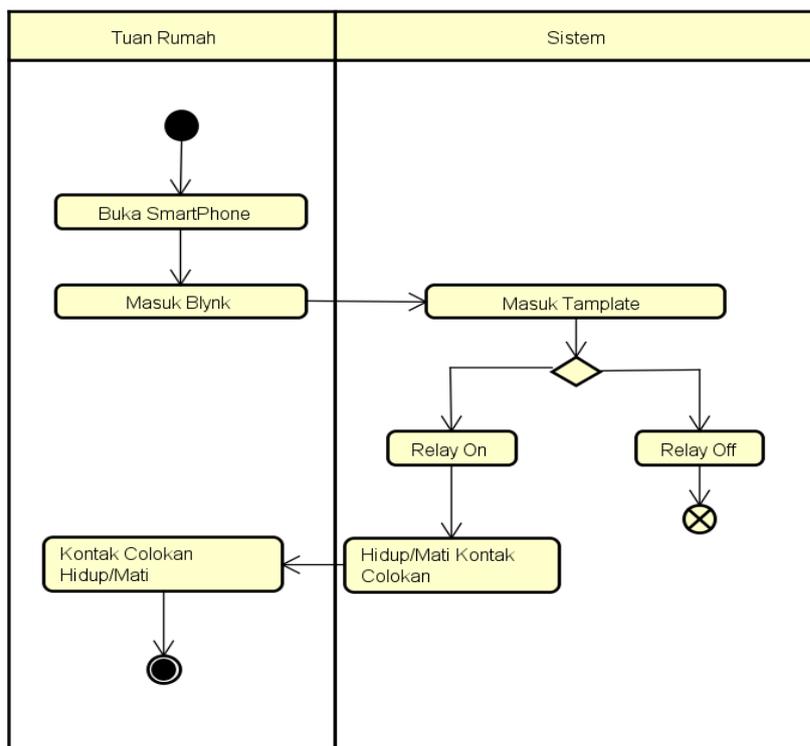
Gambar 4.19 Activity Diagram Proses Perancangan Lampu

act Activity Diagram Proses Perancangan Pintu



Gambar 4.20 Activity Diagram Proses Perancangan DoorLock

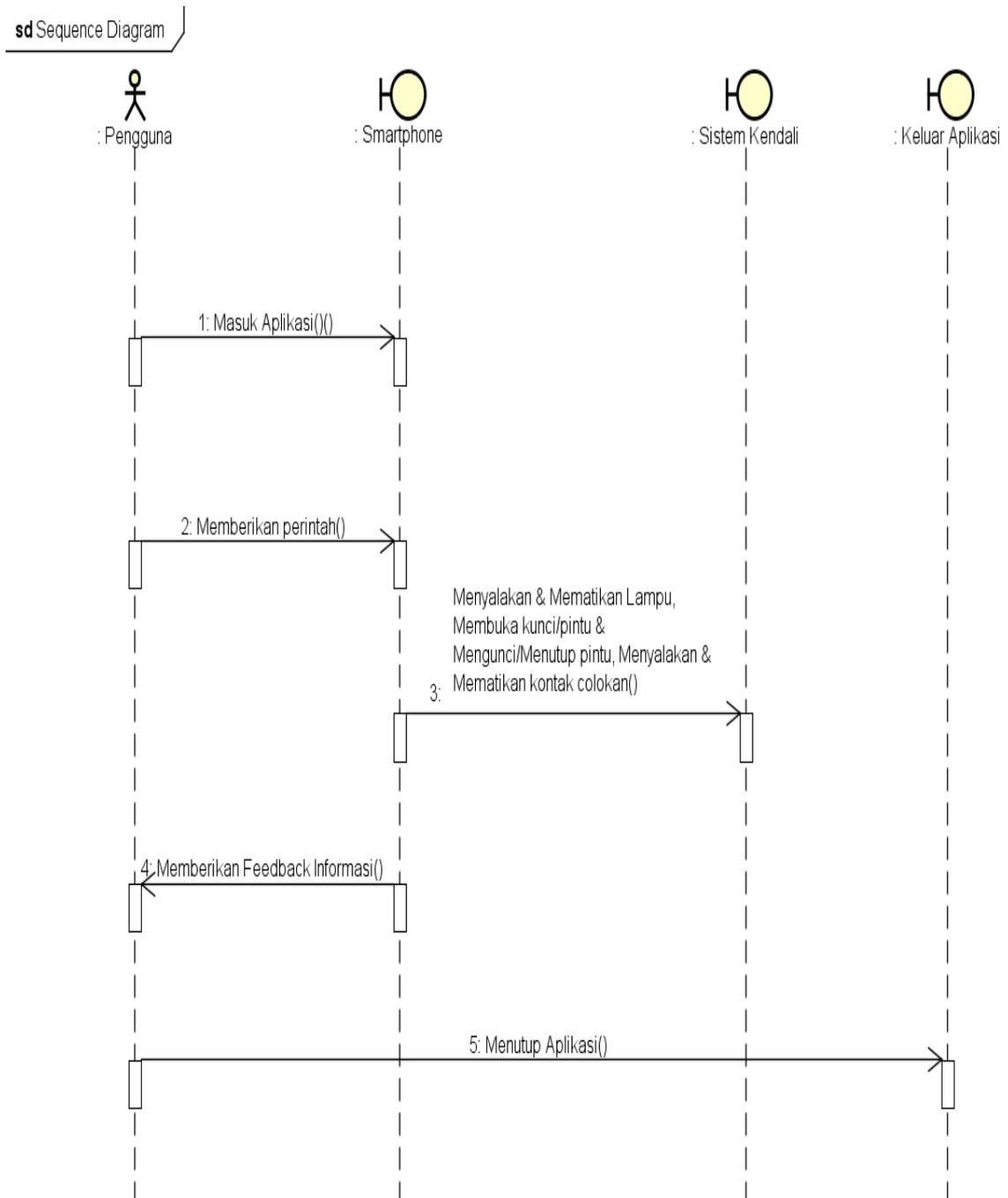
act Activity Diagram Perancangan Kontak Colokan



Gambar 4.21 Activity Diagram Proses Perancangan Kontak Colokan

### 4.7.2.2. Sequence Diagram

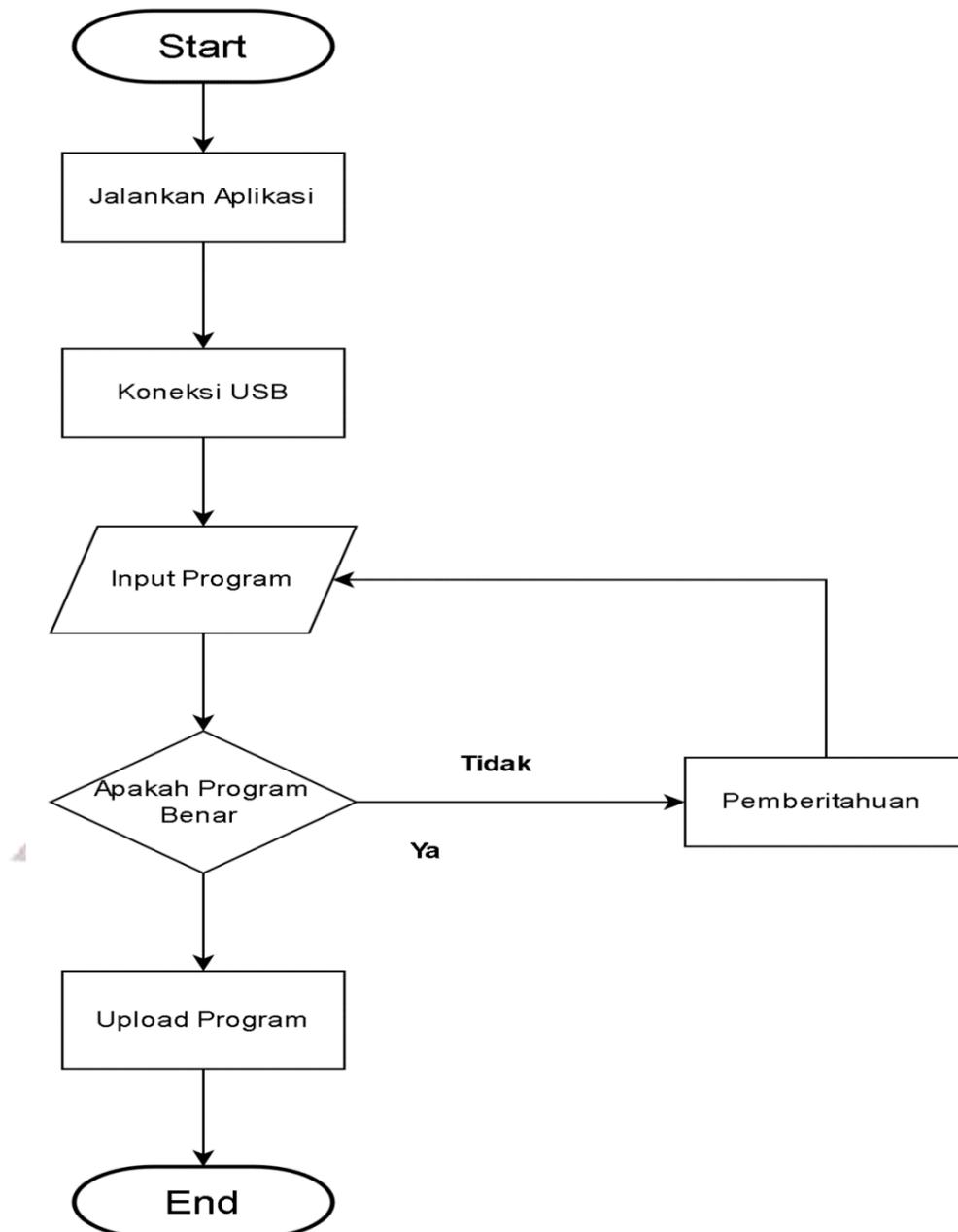
Berikut adalah proses sequence diagram pada rancangan proses monitoring Blynk



**Gambar 4.22** Sequence Diagram Proses rancangan Monitoring Blynk

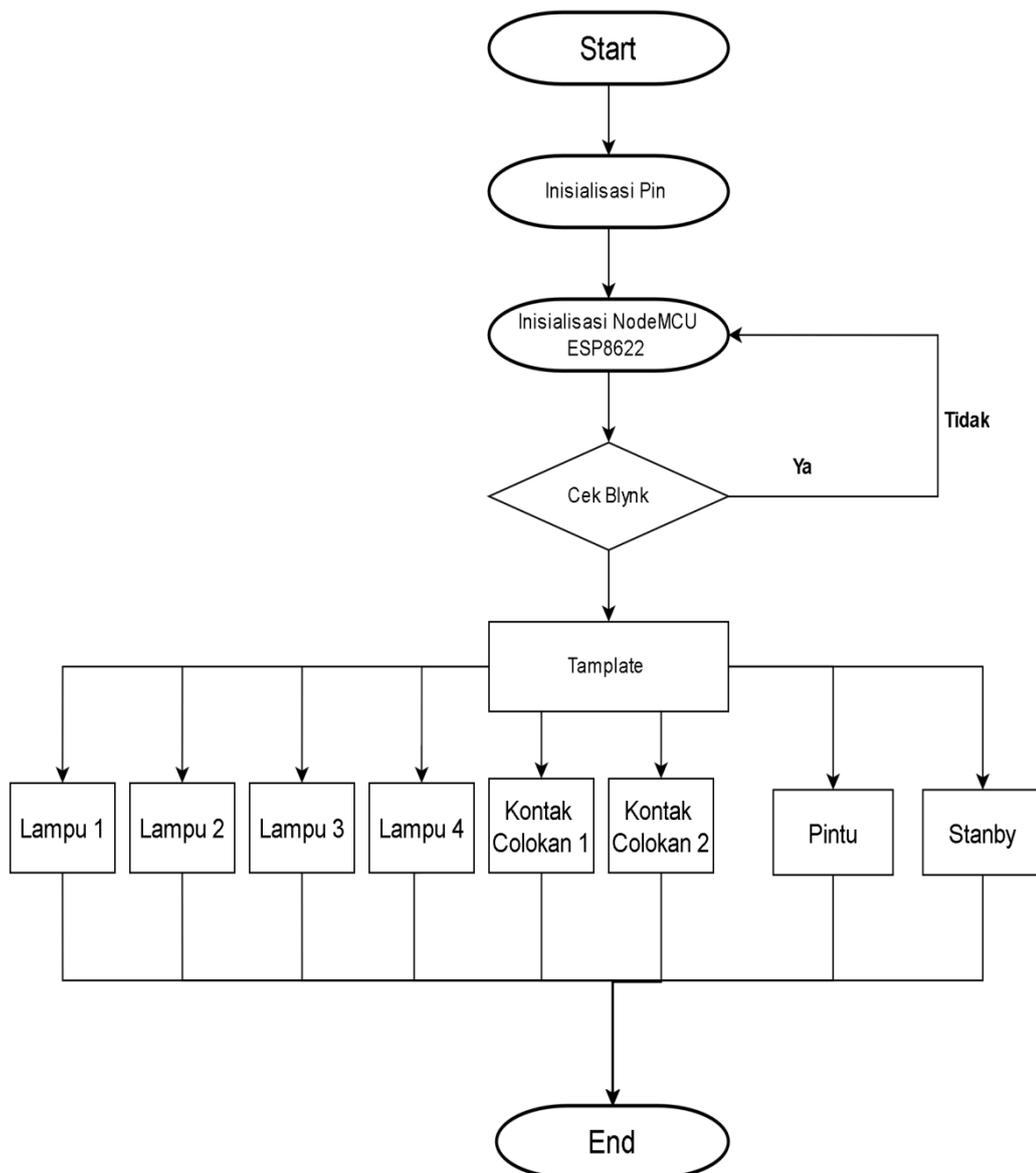
#### 4.7.2.3. Rancangan Perangkat Lunak pada NodeMCU ESP8266

Sebelum ketahapan berikutnya, maka terlebih dahulu dibuatlah *flowchart* proses upload kode program atau *sketch* papan NodeMCU ESP8266 dengan *flowchart* sebagai Berikut:



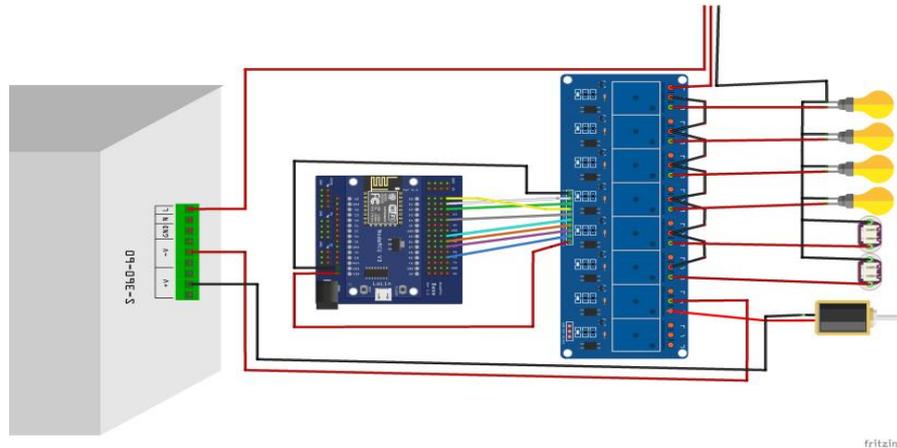
**Gambar 4.23** *Flowchart* upload kode Program NodeMCU ESP826

Langkah awal yang harus dilakukan adalah dengan menentukan logika yang akan diterapkan pada lampu, *DoorLock*, dan kontak colokan yang akan dikendalikan, kemudian membuat algoritmanya yang kemudian di implemntasikan menggunakan *Software ArduinoIDE* maka dibuatlah *flowchart* dan algoritma *input* perintah dari perangkat lunak yang akan ditanam didalam mikrokontroler, yaitu sebagai berikut:



**Gambar 4.24** *Flowchart input* perintah pada NodeMCU ESP8266

#### 4.7.2.4. Rancangan Fisik Perangkat Keras



**Gambar 4.25** Rancangan Fisik Perangkat Keras

Dari gambar 4.25 menunjukkan :

- kebel warna hitam adalah kabel netral. Kabel ini pada dasarnya adalah kabel acuan atau tegangan 0 pada instalasi listrik.
- Kabel warna merah kabel fasa merupakan kabel yang mengandung tegangan dan pada umumnya selalu dilambangkan dengan simbol sinusoida, kabel yang berupa kabel fasa ini jika dilakukan tes dengan tespen maka lampu indikator tespen tersebut akan menyala.
- Kabel Jumper warna Putih untuk menghubungkan NodeMCU pin D1 (GPIO 5) ke *relay* pin D1
- Kabel Jumper warna Hijau ke 1 untuk menghubungkan NodeMCU pin D2 (GPIO 4) ke *relay* pin D2
- Kabel Jumper warna putih untuk menghubungkan NodeMCU pin D0 (GPIO 16) ke *relay* pin D3
- Kabel Jumper warna abu – abu untuk menghubungkan NodeMCU pin D4 (GPIO 2) ke *relay* pin D4
- Kabel Jumper warna Hijau ke 2 untuk menghubungkan NodeMCU pin D5 (GPIO 14) ke *relay* pin D5

- h. Kabel Jumper warna oren untuk menghubungkan NodeMCU pin D6 (GPIO 12) ke *relay* pin D6
- i. Kabel Jumper warna ungu untuk menghubungkan NodeMCU pin D7 (GPIO 13) ke *relay* pin D7
- j. Kabel Jumper warna ungu untuk menghubungkan NodeMCU pin RX (GPIO 3) ke *relay* pin D8

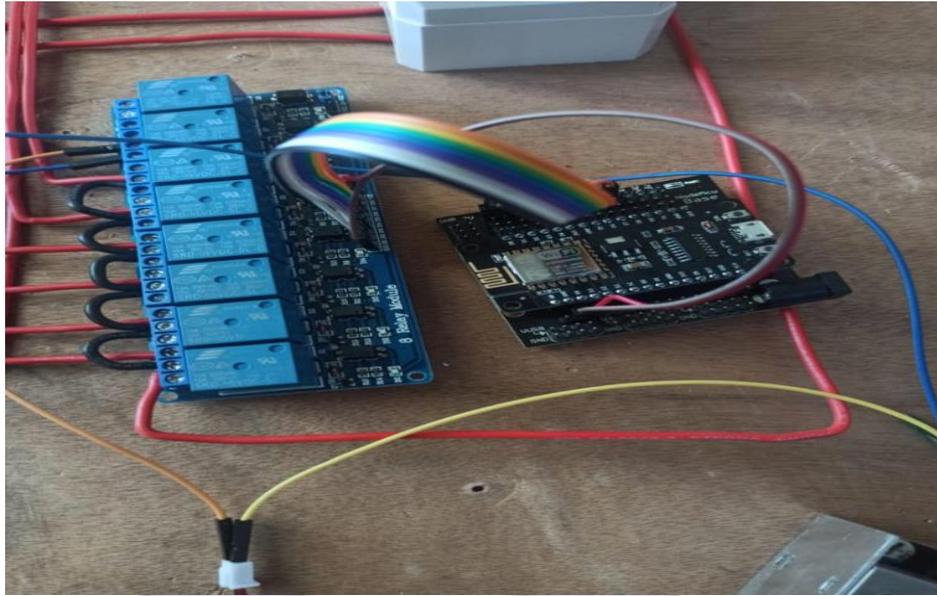
#### **4.8. Implementasi**

Implementasi merupakan kegiatan akhir dari proses penerapan sistem baru, dimana tahap ini merupakan tahap meletakkan sistem supaya siap untuk dioperasikan dan dapat dipandang sebagai usaha untuk mewujudkan sistem yang telah dirancang.

##### **4.8.1. Instalasi Perangkat keras**

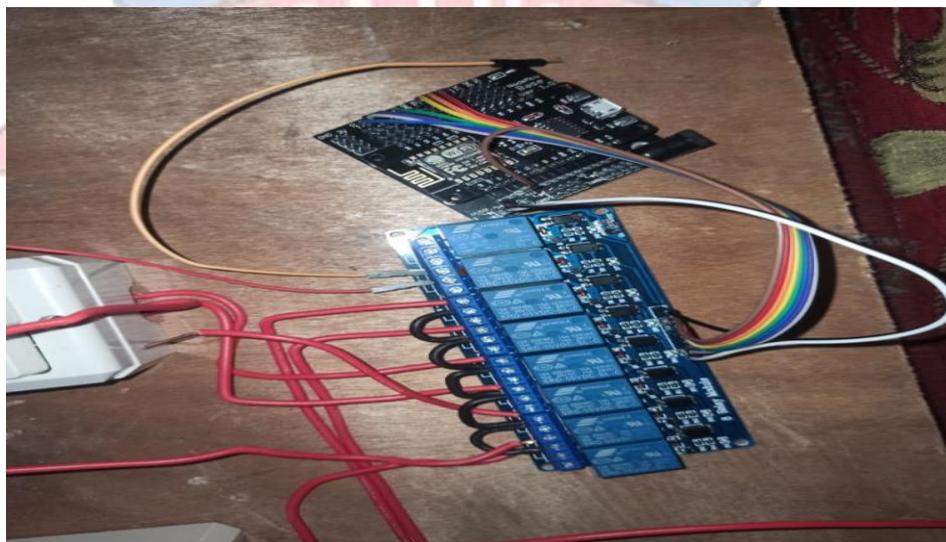
Instalasi perangkat keras merupakan suatu proses instalasi alat atau perakitan alat yang digunakan dalam membangun implementasi *SmartHome* dengan konsep *Internet Of Things (IOT)* berbasis *SmartPhone*.

- a. Rangkaian NodeMCU ESP8266  
Rangkaian ini merupakan untuk menjalankan alat mikrokontroler dan juga koneksi internet, terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun *controlling* pada *project IoT*. Bisa dilihat dari gambar 4.26



**Gambar 4.26** rangkaian NodeMCU ESP8266

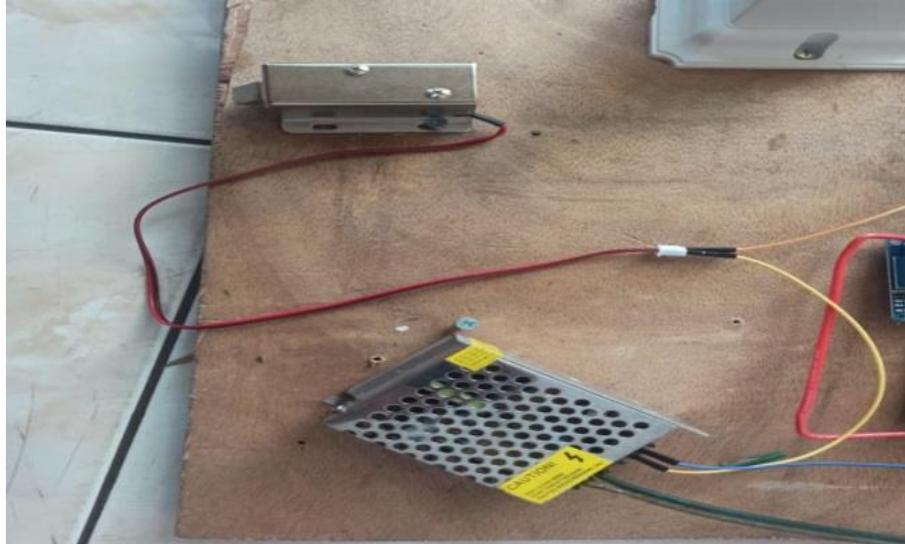
- b. Rangkaian *Relay*  
Rangkaian ini merupakan memutuskan atau menyambungkan aliran. Bisa dilihat dari gambar 4.27



**Gambar 4.27** Rangkaian *Relay*

c. Rangkaian *DoorLock*

*DoorLock* merupakan alat pengunci pintu elektrik. Bisa dilihat gambar 4.28



**Gambar 4.28** Rangkaian *DoorLock*

d. Rangkaian Lampu

Lampu merupakan alat penerangan ruangan bisa dilihat gambar 4.29



**Gambar 4.29** Rangkaian lampu

e. Rangkaian Kontak Colokan

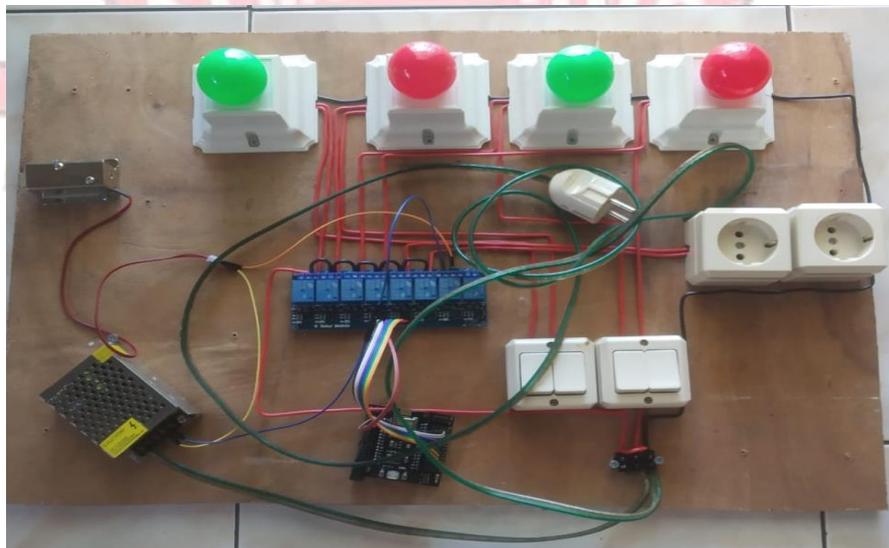
Kontak Colokan merupakan mendistribusikan energi listrik dari instalasi rumah ke alat elektronik. Bisa dilihat gambar 4.30



**Gambar 4.30** Rangkaian Kontak Colokan

f. Rangkaian Keseluruhan

Pada bagian ini merupakan hasil dari rangkaian alat secara keseluruhan dimana sudah menjadi suatu *prototype* alat yang siap di ujicoba. Bisa dilihat gambar 4.31

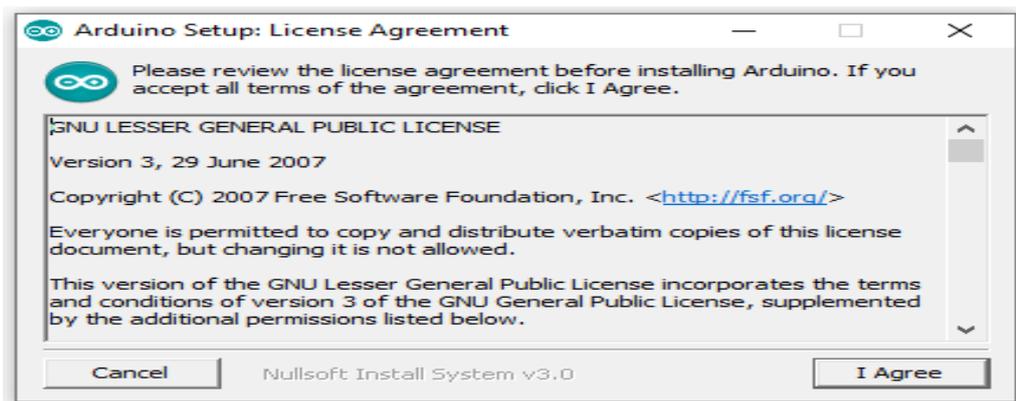


**Gambar 4.31** Rangkaian Keseluruhan

#### 4.8.2. Instalasi Perangkat Lunak

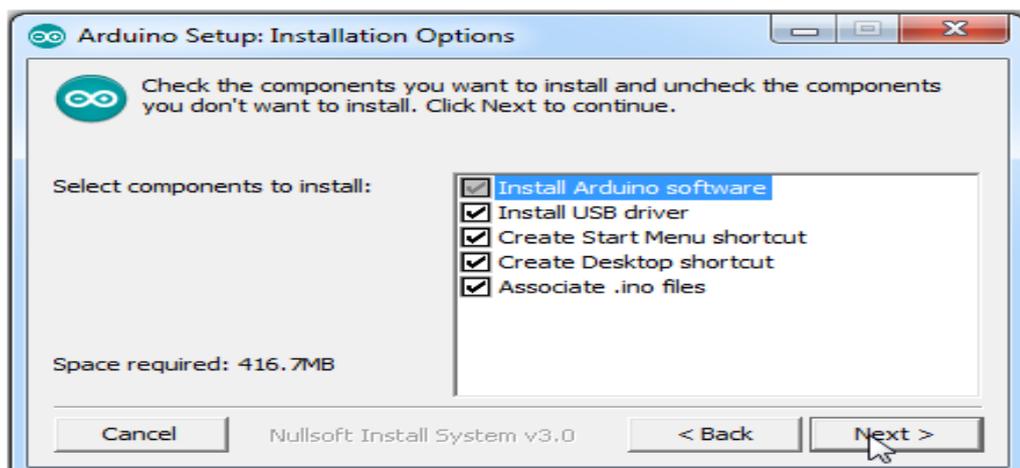
Pada topik ini dijelaskan langkah – langkah instalasi arduinoIDE adalah sebagai berikut:

- a. Klik dua kali *setup* arduino IDE 1.8.19 *windows*, sehingga muncul jendela *license Agreement*.
- b. Pada jendela *license Agreement* klik “*I Agree*” untuk melanjutkan. Bisa dilihat gambar 4.32



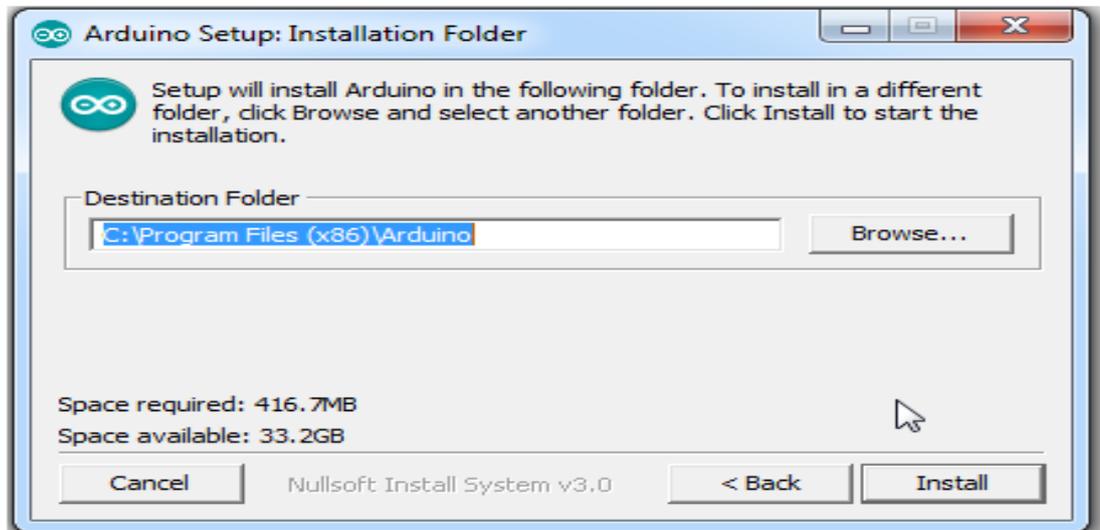
**Gambar 4.32** *License Agreement*

- c. Pada jendela *installation options* tentang semua komponen, kemudian klik “*Next*” untuk melanjutkan. Bisa dilihat gambar 4.33



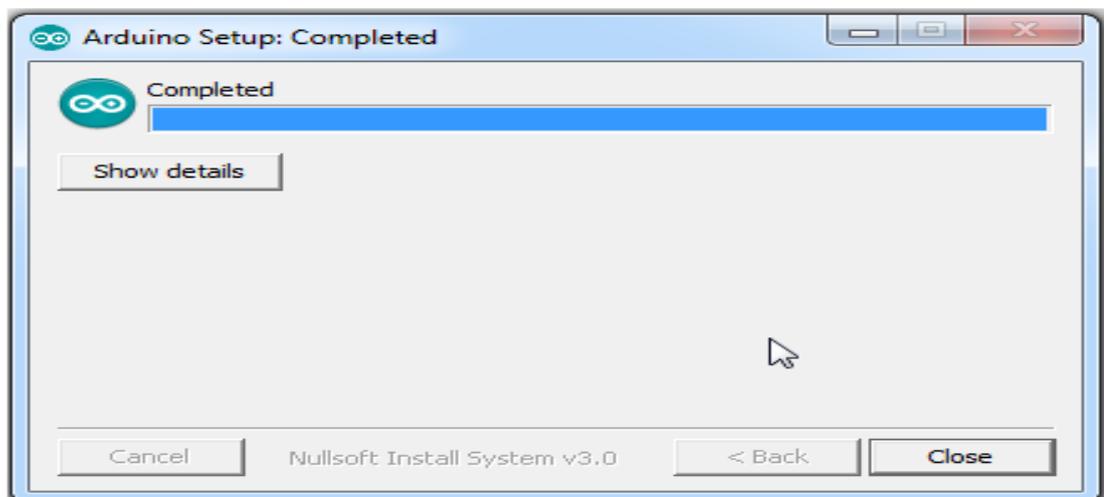
**Gambar 4.33** *Installation Options*

- d. Pada jendela *installation folder* untuk *destination folder* dibiarkan saja seperti standar, kemudian klik “*Instal*” untuk memulai instalasi *software*. Bisa dilihat gambar 4.34



**Gambar 4.34** *Installation Folder*

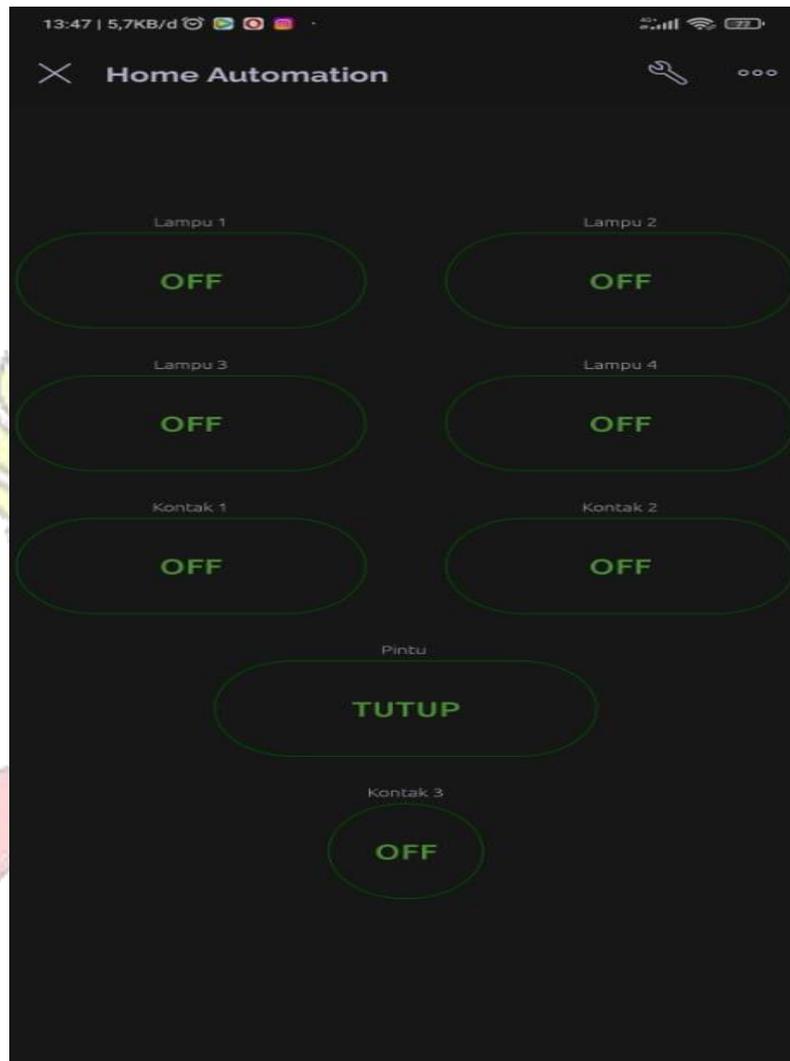
- e. Pada jendela *Completed* merupakan bahwa instalasi sudah kompleks kemudian klik “*Close*” untuk mengakhiri proses instalasi. Bisa diluhat gambar 4.35



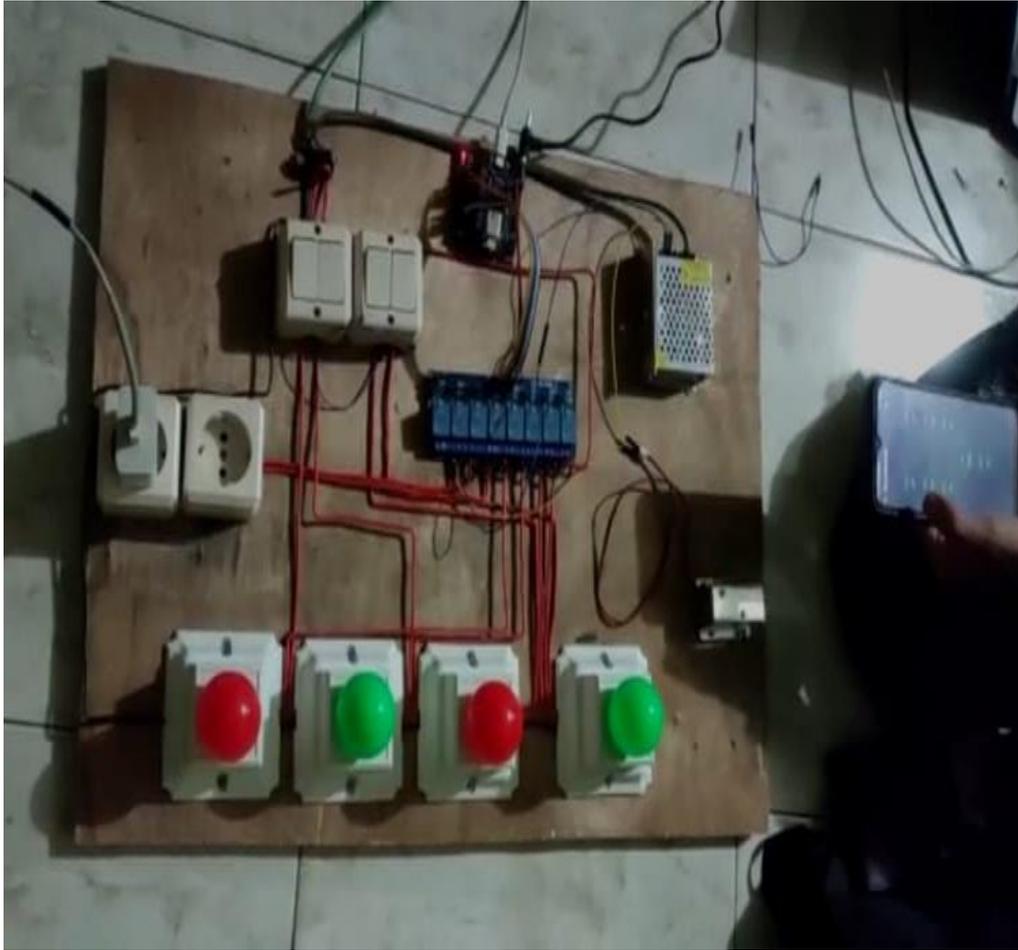
**Gambar 4.35** *Installing Completed*

#### 4.9. Pengujian Sistem

Setelah melakukan dan membuat perancangan, maka akan dilakukan pengujian terhadap sistem. Pengujian ini dilakukan agar sistem berjalan dengan sempurna. Pengujian sistem juga dimaksudkan untuk menguji semua elemen – elemen perangkat keras yang dibuat apakah sudah sesuai dengan yang diharapkan.

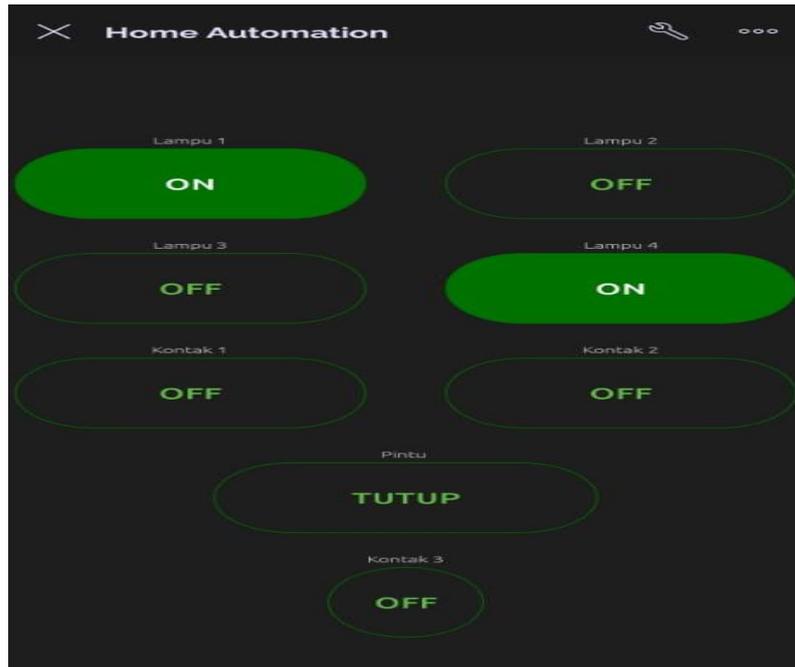


**Gambar 4.36** *Template Blynk* belum di tekan



**Gambar 4.37** Ketika lampu mati belum dihidupkan

Pada Gambar 4.36 dan 4.37 menunjukkan bahwa sistem sudah di aktifkan tapi belum menekan tombol ON/OFF pada *template* aplikasi Blynk. Maka dari itu ketika *user* ingin menyalakan atau mematikan lampu dan kontak colokan harus menekan ON/OFF pada layar *template* aplikasi Blynk.

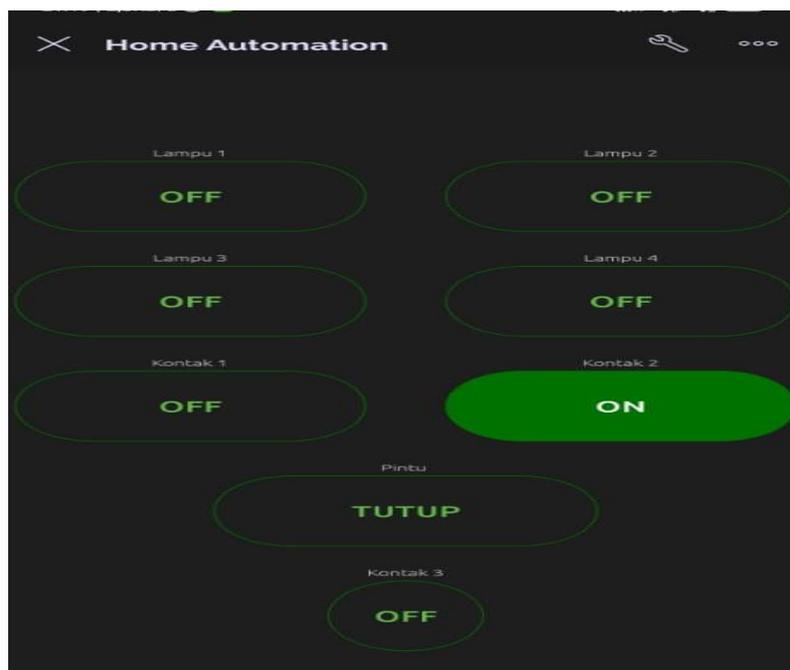


**Gambar 4.38** Blynk Lampu 1 dan 4 Menyala



**Gambar 4.39** Ketika Lampu 1 dan 4 dinyalakan

Pada gambar 4.38 dan 4.39 menunjukkan bahwa template ON pada lampu 1 dan lampu 4 di aktifkan, maka ke dua lampu itu menyala. Ketika ingin mematikan kembali lampu tersebut, maka *user* menekan tombol ON/OFF pada *template* aplikasi Blynk.

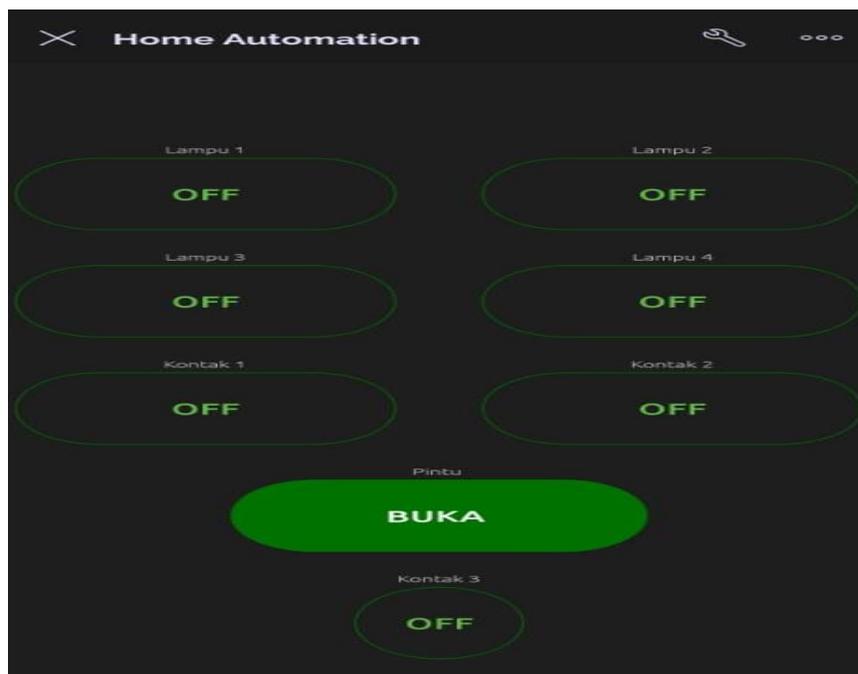


**Gambar 4.40** Tamplate Blynk ketika kontak colokan ON



**Gambar 4.41** ketika kontak colokan 2 ON

Pada gambar 4.40 dan 4.41 menunjukkan bahwa tamplate ON pada kontak colokan 2 di aktifkan maka kontak colokan tersebut akan menyala. Ketika ingin mematikan kembali kontak colokan tersebut, maka *user* menekan tombol ON/OFF pada *template* Kontak aplikasi Blynk.

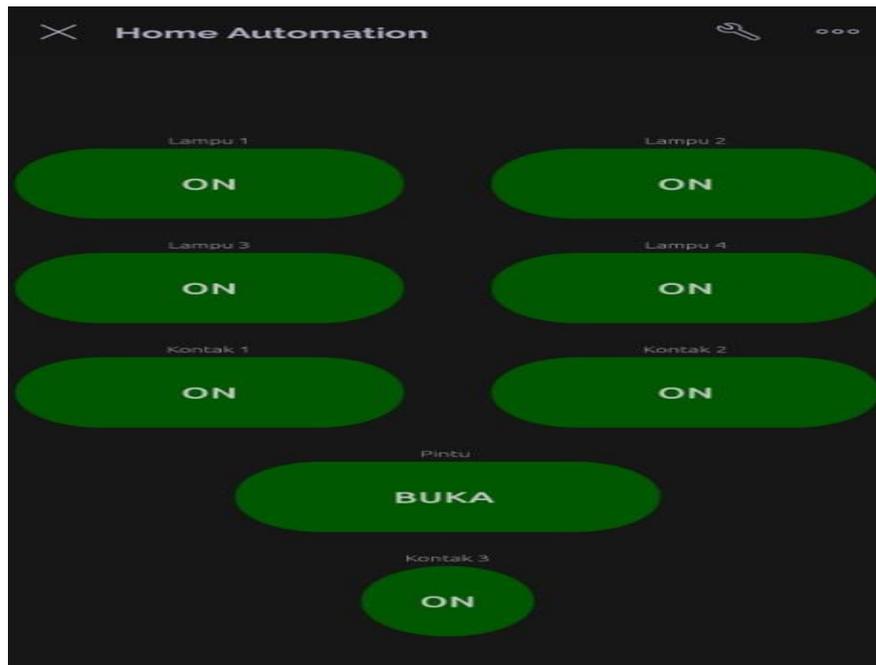


**Gambar 4.42** *Template Blynk Solenoid pintu terbuka*



**Gambar 4.43** *Solenoid Pintu terbuka*

Bisa dilihat dari gambar 4.42 dan 4.43 menunjukkan bahwa template buka pada pintu di aktifkan maka *solenoid* pada pintu akan menarik kutup pada ujung *solenoid* maka pintu akan terbuka. Ketika ingin menutup kembali pintu tersebut maka user menekan tombol Buka/Tutup pada *template* pintu aplikasi Blynk.



**Gambar 4.44** Tamplate Blynk Ketika dinyalakan semua



**Gambar 4.45** Ketika lampu dinyalakan semua

Pada gambar 4.44 dan 4.45 menunjukkan bahwa *template* ON dan Buka pada semua sistem di aktifkan maka 4 lampu menyala, 2 kontak colokan aktif dan pintu terbuka. Ketika ingin memamatkannya kembali maka user menekan tombol ON/OFF dan Buka/Tutup pada *template* aplikasi Blynk.

#### 4.10. Pengujian sistem pada perangkat

Berikut ini adalah hasil pengujian memonitoring alat menggunakan perangkat Redmi Note 8 pro dengan web browser Google Chrome pada laptop. Bisa dilihat dari tabel 4.4

**Tabel 4.4** Pengujian *Black Box* Monitoring Pertama semua perangkat

No	Jenis Peralatan	Percobaan	Kondisi	
			ON	OFF
1	Lampu 1	Pertama	Berhasil	Berhasil
2	Lampu 2	Pertama	Berhasil	Berhasil
3	Lampu 3	Pertama	Berhasil	Berhasil
4	Lampu 4	Pertama	Berhasil	Berhasil
5	Kontak Colokan 1	Pertama	Berhasil	Berhasil
6	Kontak Colokan 2	Pertama	Berhasil	Berhasil
7	Pintu	Pertama	Berhasil	Berhasil
8	Stanby	Pertama	Berhasil	Berhasil

**Tabel 4.5** Pengujian *Black Box* Monitoring kedua semua perangkat

No	Jenis Peralatan	Percobaan	Kondisi	
			ON	OFF
1	Lampu 1	Kedua	Berhasil	Berhasil
2	Lampu 2	Kedua	Berhasil	Berhasil
3	Lampu 3	Kedua	Berhasil	Berhasil
4	Lampu 4	Kedua	Berhasil	Berhasil
5	Kontak Colokan 1	Kedua	Berhasil	Berhasil
6	Kontak Colokan 2	Kedua	Berhasil	Berhasil
7	Pintu	Kedua	Berhasil	Berhasil
8	Standby	Kedua	Berhasil	Berhasil

**Tabel 4.6** Pengujian *Black Box* monitoring ketiga semua perangkat

No	Jenis Peralatan	Percobaan	Kondisi	
			ON	OFF
1	Lampu 1	Ketiga	Berhasil	Berhasil
2	Lampu 2	Ketiga	Berhasil	Berhasil
3	Lampu 3	Ketiga	Berhasil	Berhasil
4	Lampu 4	Ketiga	Berhasil	Berhasil
5	Kontak Colokan 1	Ketiga	Berhasil	Berhasil
6	Kontak Colokan 2	Ketiga	Berhasil	Berhasil
7	Pintu	Ketiga	Berhasil	Berhasil
8	Standby	Ketiga	Berhasil	Trouble

Pada percobaan ketiga ini sedikit ada masalah pada pin antara NodeMcu ESP8266 ke *Relay*. nomor 8 atau disebut dengan *stanby* yang hanya bisa dihidupkan dan tidak bisa dimatikan. Setelah dilakukan perbaikan pada perangkat semua bisa berjalan dengan baik.

**Tabel 4.7** Pengujian *Black Box* monitoring keempat semua perangkat

No	Jenis Peralatan	Percobaan	Kondisi	
			ON	OFF
1	Lampu 1	Keempat	Berhasil	Berhasil
2	Lampu 2	Keempat	Berhasil	Berhasil
3	Lampu 3	Keempat	Berhasil	Berhasil
4	Lampu 4	Keempat	Berhasil	Berhasil
5	Kontak Colokan 1	Keempat	Berhasil	Berhasil
6	Kontak Colokan 2	Keempat	Berhasil	Berhasil
7	Pintu	Keempat	Berhasil	Berhasil
8	Standby	Keempat	Berhasil	Berhasil

**Tabel 4.8** Pengujian *Black Box* monitoring kelima semua perangkat

No	Jenis Peralatan	Percobaan	Kondisi	
			ON	OFF
1	Lampu 1	kelima	Berhasil	Berhasil
2	Lampu 2	kelima	Berhasil	Berhasil
3	Lampu 3	kelima	Berhasil	Berhasil
4	Lampu 4	kelima	Berhasil	Berhasil
5	Kontak Colokan 1	kelima	Berhasil	Berhasil
6	Kontak Colokan 2	kelima	Berhasil	Berhasil
7	Pintu	kelima	Berhasil	Berhasil
8	Standby	kelima	Berhasil	Berhasil

Pada percobaan kelima ini semua perangkat bisa berjalan dengan baik dan dengan mudah kita memonitoring perangkat hanya dengan aplikasi Blynk saja.

