

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Hasil dari penelitian dan analisis yang terdapat dalam skripsi ini menggambarkan secara rinci mengenai mekanisme yang digunakan untuk menghidupkan lampu menggunakan mikrokontroler. Dalam hal ini, lampu konvensional berhasil diubah menjadi lampu LED, yang diimplementasikan dalam tahap uji coba alat.

Melalui serangkaian pengujian yang dilakukan pada seluruh sistem, tujuan utamanya adalah untuk memahami dengan jelas bagaimana kinerja sistem berjalan serta tingkat keberhasilannya. Dengan demikian, informasi berharga mengenai hasil perancangan dan pembuatan alat.

Penelitian ini memberikan gambaran menyeluruh tentang langkah-langkah yang diperlukan untuk mencapai tujuan menghidupkan lampu melalui mikrokontroler, dengan pengembangan khusus pada penggunaan lampu LED sebagai contoh konkret. Seluruh eksperimen yang dilakukan bertujuan untuk memberikan wawasan mendalam tentang bagaimana sistem ini berfungsi secara praktis, sambil mengamati perkembangan dari tahap desain hingga implementasi yang dapat divisualisasikan melalui gambar yang telah disediakan.

#### 4.1 Pengembangan ADDIE pada Sistem Rancangan

Metode ADDIE (Analysis, Design, Development, Implementation, Evaluation) adalah pendekatan sistematis yang digunakan dalam desain instruksional untuk mengembangkan materi pembelajaran. Meskipun biasanya digunakan dalam konteks pendidikan, prinsip ADDIE dapat diterapkan dalam rancang bangun proyek teknologi seperti alat pengontrol lampu berbasis IoT. Berikut adalah tahapan-tahapan metode ADDIE yang dapat diterapkan dalam rancangan ini:

1. **Analysis (Analisis):** Pada tahap analisis, Anda perlu memahami secara mendalam tentang apa yang sebenarnya dibutuhkan dalam alat pengontrol lampu berbasis IoT ini. Anda harus mengidentifikasi kebutuhan dan tujuan utama dari alat ini. Beberapa pertanyaan yang dapat Anda ajukan adalah: Apa fungsi utama alat ini? Bagaimana alat ini akan berinteraksi dengan lampu dan perangkat IoT lainnya? Apa jenis konektivitas yang diperlukan? Apa kendala atau batasan yang mungkin ada?
2. **Design (Desain):** Setelah Anda memiliki pemahaman yang kuat tentang kebutuhan, langkah selanjutnya adalah merancang alat secara keseluruhan. Ini termasuk merencanakan antarmuka pengguna, mengidentifikasi komponen perangkat keras yang diperlukan (seperti mikrokontroler, modul IoT, relay, dll.), serta merencanakan bagaimana alat ini akan berinteraksi dengan lampu dan platform IoT (seperti aplikasi seluler atau platform web). Anda juga perlu memikirkan aspek energi dan konsumsi daya.

3. **Development (Pengembangan):** Setelah desain telah disusun, Anda dapat mulai mengembangkan perangkat keras dan perangkat lunak yang diperlukan. Ini mungkin melibatkan pemrograman mikrokontroler atau perangkat lunak yang berjalan di perangkat penyambung lainnya. Anda akan mengintegrasikan berbagai komponen dan mengimplementasikan logika pengontrolan lampu berbasis IoT sesuai dengan desain yang telah Anda buat.
4. **Implementation (Implementasi):** Tahap implementasi melibatkan pemasangan dan pengujian alat yang telah Anda kembangkan. Anda akan menghubungkan alat dengan lampu dan menguji fungsi pengontrolan melalui konektivitas IoT. Pastikan bahwa semua fitur dan fungsionalitas yang Anda rencanakan berjalan dengan baik dalam skenario nyata.
5. **Evaluation (Evaluasi):** Setelah alat dikembangkan dan diimplementasikan, tahap evaluasi akan menguji apakah alat memenuhi tujuan dan kebutuhan yang telah ditetapkan pada tahap analisis. Anda dapat melibatkan pengguna untuk menguji alat dan mendapatkan umpan balik tentang kinerjanya. Berdasarkan umpan balik ini, Anda dapat melakukan perbaikan atau penyesuaian yang diperlukan.

Dengan menggunakan metode ADDE ini, Anda dapat memastikan bahwa pengembangan alat pengontrol lampu berbasis IoT dilakukan secara sistematis dan efektif, dari pemahaman awal hingga pengujian dan peningkatan berkelanjutan.

## 4.2 Implementasi Sistem

Implementasi merupakan tahapan penerapan perangkat keras dan perangkat lunak agar alat dapat bekerja sesuai kebutuhan dan fungsinya. Perangkat keras nantinya akan dirakit dan disusun sedemikian rupa agar dapat dihubungkan dengan perangkat lunak. Pada perangkat keras terbagi menjadi beberapa modul dan komponen-komponen dasar elektronika, sedangkan pada perangkat lunak penulis menggunakan aplikasi Blynk IoT sebagai editor penulisan program dan Kontrol Lampu sebagai alat untuk mengontrol lampu didalam pada tiap ruangan.

### 4.2.1 Implementasi Perangkat Keras (Hardware)

Pada bab ini, akan dijelaskan secara mendetail mengenai implementasi dari alat pengontrol lampu berbasis IoT yang melibatkan komponen-komponen utama, yaitu *LED*, *LCD*, Node MCU ESP8266, dan *relay Chanel 2*. Setiap komponen perangkat keras akan diuraikan secara rinci, termasuk konfigurasi perangkat lunak yang diperlukan serta interaksi antar komponen dalam sistem. Setiap komponen perangkat keras harus saling terhubung dan berkolaborasi dengan harmoni/berintegrasi sesuai dengan tuntutan sistem dan tujuan rancang bangun kontrol lampu ruangan dengan pemanfaatan teknologi wifi dan internet. Spesifikasi perangkat keras yang diterapkan dalam mengembangkan sistem ini mencakup:"

#### 1. Module

- a. LCD
- b. Relay 2 Channel
- c. ESP8266

## 2. Komponen Prototype

- a. Kabel Jumper
- b. LED

## 3. Komponen Penunjang

- a. a. Android Smartphone
- b. Aplikasi Blynk IoT

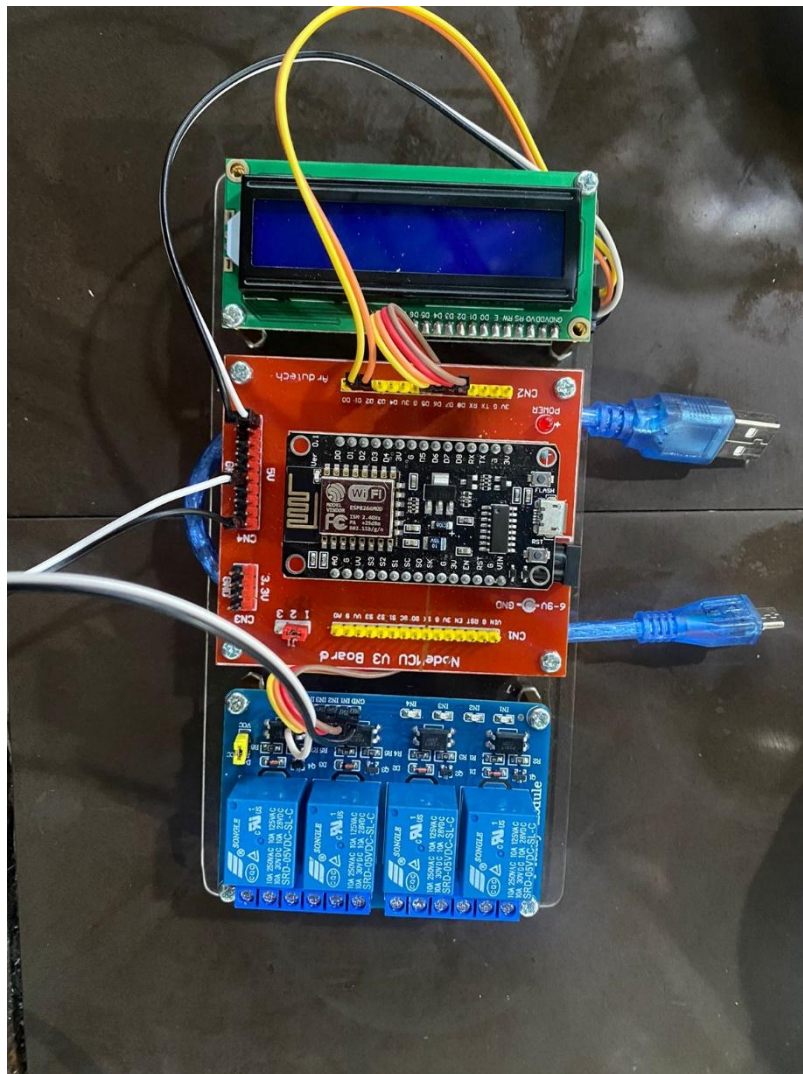
### 4.2.2 Perakitan Perangkat Keras

Pada proses perakitan, rangkaian perangkat keras ini melibatkan Arduino Uno, modul Wifi, modul relay, dan modul ESP8266. Setiap komponen perangkat keras harus dihubungkan dan beroperasi secara bersinergi, saling melengkapi sesuai kebutuhan alat, serta tujuan rancang bangun pengendalian pencahayaan rumah di tiap ruangan melalui koneksi wifi dan internet.

Dengan memadukan berbagai modul yang telah disebutkan, tahap selanjutnya melibatkan penggabungan komponen-komponen ini menjadi satu kesatuan yang terintegrasi, membentuk alat yang mampu beroperasi secara bersamaan. Integrasi ini memastikan bahwa semua aspek sistem dapat berinteraksi

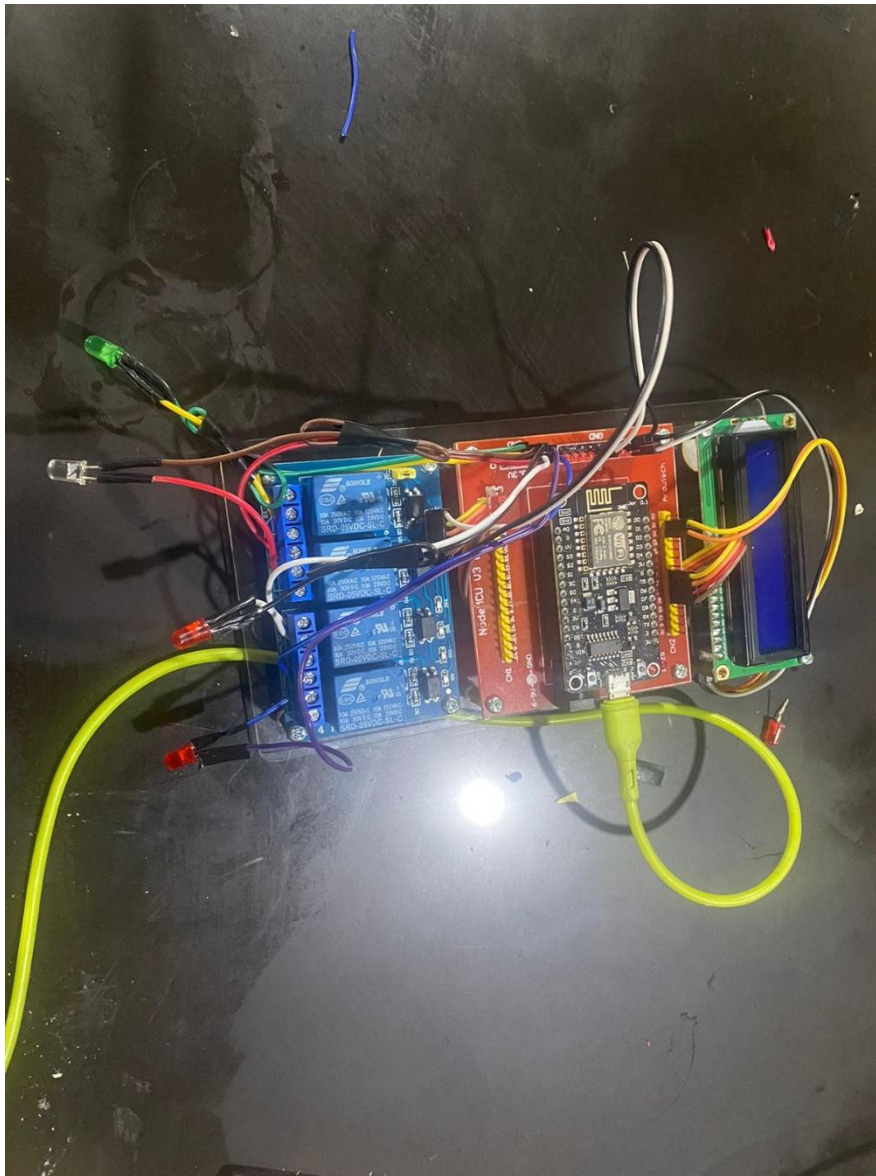
dan berfungsi secara efektif, sesuai dengan visi pengembangan perangkat kontrol lampu rumah yang telah ditetapkan.

### 1. Rangkaian keseluruhan Tanpa Lampu



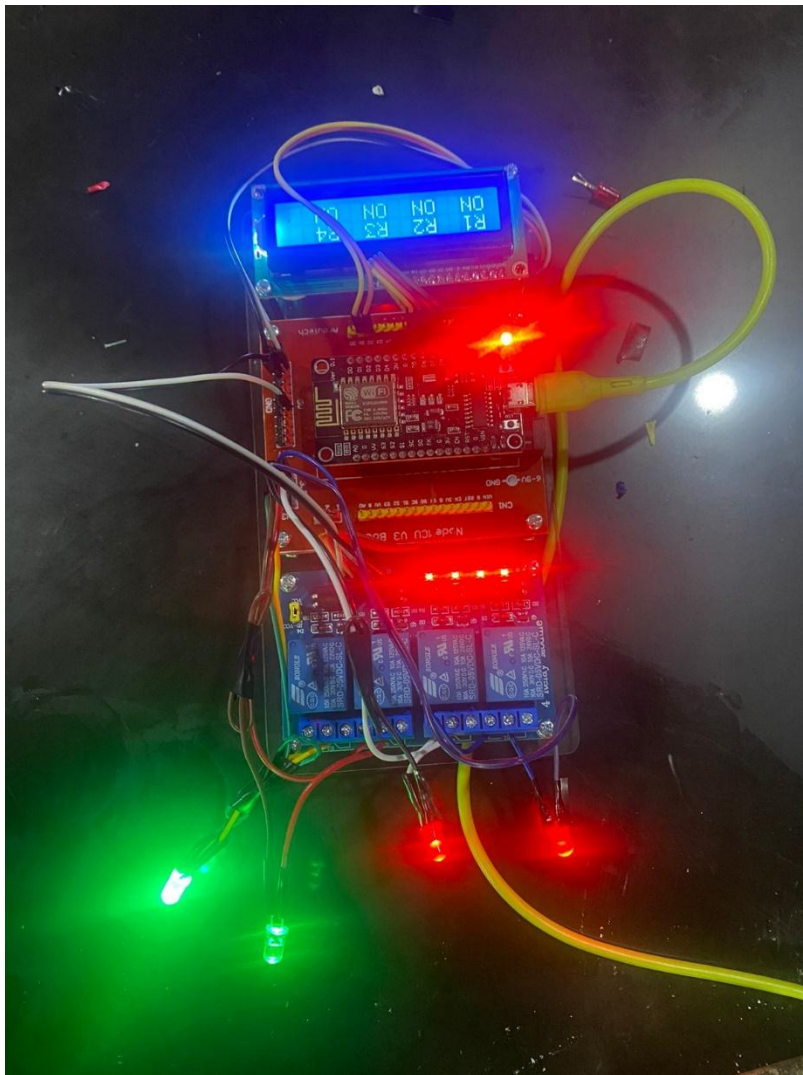
**Gambar 4.1. Rangkaian Alat Tanpa Lampu**

## 2.Rangkain Alat Dengan Lampu



**Gambar 4.2 Rangkaian Alat Dengan Lampu**

### 3. Hasil Rangkaian Keseluruhan



**Gambar 4.3 Hasil Rangkain Keseluruhan**

Keterangan:



1. **NodeMCU ESP8266:** NodeMCU ESP8266 adalah modul mikrokontroler yang memiliki kemampuan Wi-Fi, yang memungkinkan alat terhubung ke jaringan Internet. Modul ini bertindak sebagai otak dari alat pengontrol, menerima perintah dari aplikasi pengguna melalui jaringan Wi-Fi, dan mengontrol komponen lainnya sesuai instruksi yang diterima.
2. **LCD (Liquid Crystal Display):** LCD digunakan sebagai antarmuka visual antara pengguna dan alat. LCD menampilkan informasi seperti status lampu, perintah yang diterima, atau pesan lain yang relevan. Pengguna dapat melihat informasi ini secara langsung pada layar LCD, memungkinkan pemantauan dan interaksi yang lebih intuitif.
3. **Relay Chanel 2:** Relay Chanel 2 digunakan untuk mengendalikan pasokan listrik ke lampu. Relay ini memiliki dua saluran yang dapat dikontrol secara terpisah. Ketika NodeMCU menerima perintah untuk menghidupkan atau mematikan lampu, relay akan diaktifkan atau dinonaktifkan sesuai instruksi, sehingga mengendalikan sumber daya listrik menuju lampu.
4. **LED (Light Emitting Diode):** LED digunakan sebagai indikator visual untuk menunjukkan status operasional alat. LED dapat menyala atau mati sesuai dengan kondisi alat, seperti status koneksi Wi-Fi, status operasi lampu, atau pesan kesalahan. Hal ini membantu pengguna memahami keadaan alat secara cepat.

Alat ini bekerja dengan menghubungkan semua komponen di atas dalam satu sistem terintegrasi. NodeMCU mengatur koneksi Wi-Fi, menerima perintah dari pengguna melalui aplikasi, mengatur relay untuk mengontrol lampu, serta mengirimkan informasi status ke LCD dan LED. Aplikasi pengguna pada perangkat Android digunakan untuk mengirimkan perintah ke alat melalui jaringan Wi-Fi.

Dengan menggabungkan teknologi seperti NodeMCU ESP8266, LCD, Relay Chanel 2, dan LED, alat pengontrol lampu berbasis IoT ini memberikan kemudahan dan fleksibilitas dalam mengontrol pencahayaan, menjadikannya solusi yang efisien, praktis, dan sesuai dengan tren teknologi masa kini

#### 4.2.3 Implementasi Perangkat Lunak (Software)

Perangkat lunak yang mendukung alat pengontrol lampu berbasis IoT ini memiliki peran penting dalam menghubungkan komponen perangkat keras, menerima perintah dari pengguna, dan memantau status operasional. Perangkat lunak ini berinteraksi dengan NodeMCU ESP8266, mengontrol Relay Chanel 2, serta berkomunikasi dengan aplikasi pengguna melalui jaringan Wi-Fi. Berikut adalah keterangan mengenai perangkat lunak yang digunakan:

1. **Firmware pada NodeMCU ESP8266:** Firmware pada NodeMCU ESP8266 bertanggung jawab atas berbagai tugas, termasuk menghubungkan alat ke jaringan Wi-Fi, menerima perintah dari aplikasi pengguna, mengatur pengaktifan relay untuk mengontrol lampu, serta mengirimkan informasi status ke LCD dan LED. Firmware ini

dikembangkan dengan bahasa pemrograman seperti C atau Lua, dan diunggah ke NodeMCU menggunakan perangkat lunak seperti Arduino IDE atau platform pemrograman yang sesuai.

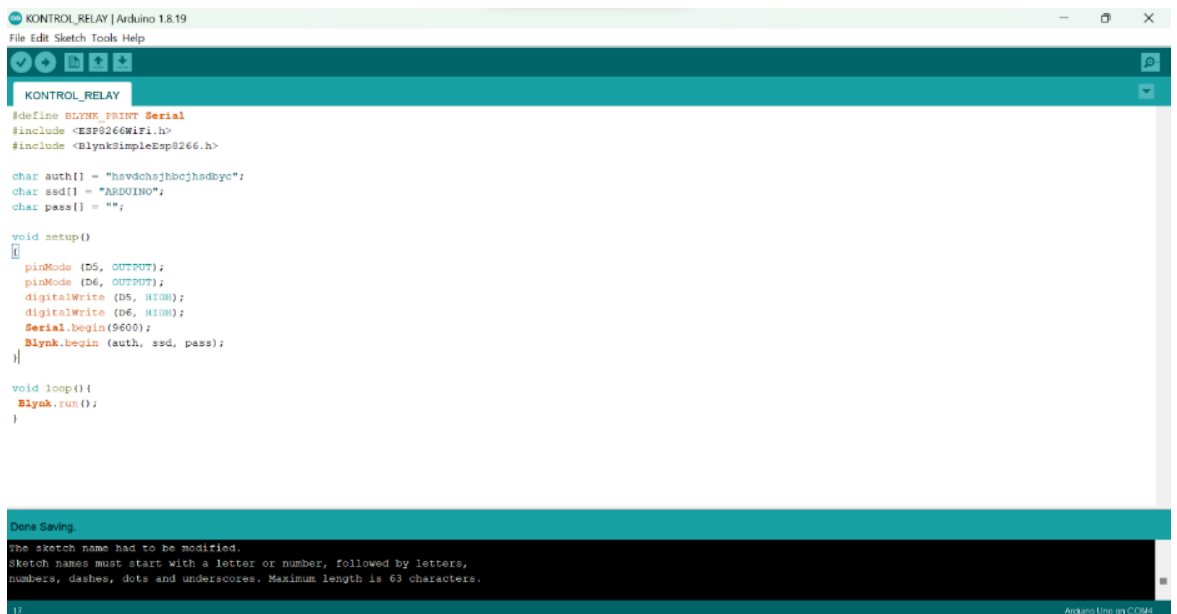
2. **Antarmuka Aplikasi Android:** Aplikasi pengguna pada perangkat Android berfungsi sebagai antarmuka untuk berkomunikasi dengan alat pengontrol melalui jaringan Wi-Fi. Aplikasi ini memungkinkan pengguna untuk mengirimkan perintah seperti menghidupkan atau mematikan lampu, serta memonitor status lampu melalui tampilan yang intuitif. Interaksi antara pengguna dan alat dilakukan melalui antarmuka grafis yang mudah dipahami.
3. **Protokol Komunikasi IoT:** Perangkat lunak pada alat ini menerapkan protokol komunikasi khusus untuk berinteraksi dengan aplikasi Android dan mengirim perintah melalui jaringan Wi-Fi. Protokol ini memungkinkan pertukaran data yang aman dan efisien antara alat dan aplikasi pengguna, serta memastikan bahwa perintah pengguna diterima dan dijalankan oleh alat.
4. **Algoritma Kontrol Lampu:** Perangkat lunak di dalam NodeMCU mengimplementasikan algoritma kontrol untuk mengatur operasi lampu sesuai perintah pengguna. Ini termasuk proses pengaktifan atau pemadaman relay sesuai instruksi, serta memastikan informasi status lampu yang akurat diteruskan ke antarmuka aplikasi Android.
5. **Pemrosesan Data LCD dan LED:** Perangkat lunak di NodeMCU juga mengelola data yang ditampilkan di LCD dan LED. Informasi status,

pesan, dan indikator visual dikendalikan oleh perangkat lunak ini untuk memberikan pemantauan yang tepat dan mudah dipahami oleh pengguna.

Perangkat lunak ini bekerja dalam keselarasan dengan perangkat keras untuk menciptakan alat pengontrol lampu berbasis IoT yang responsif dan efisien. Dengan menggabungkan antarmuka aplikasi pengguna, protokol komunikasi, dan algoritma kontrol lampu, perangkat lunak ini memastikan bahwa alat berfungsi sesuai dengan tujuan dan memberikan pengalaman pengguna yang nyaman dan efektif dalam mengendalikan pencahayaan secara jarak jauh.

#### 4.2.4 Firmware Node MCU

Dalam pengembangan sistem menggunakan Node MCU, Anda akan membutuhkan aplikasi bernama Blynk IoT. Fungsinya adalah memberikan instruksi atau kode program ke mikrokontroler . Kode program yang Anda tulis bertujuan untuk mengendalikan perintah relay on dan relay off yang terhubung melalui koneksi Wifi, internet, dan Arduino. Berikut adalah coding yang di upload kedalam Mikrokontroler :



```
KONTROL_RELAY | Arduino 1.8.19
File Edit Sketch Tools Help

KONTROL_RELAY
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "hsvdchsjhbcjhsdbyc";
char ssid[] = "ARDUINO";
char pass[] = "";

void setup()
{
  pinMode (D5, OUTPUT);
  pinMode (D6, OUTPUT);
  digitalWrite (D5, HIGH);
  digitalWrite (D6, HIGH);
  Serial.begin(9600);
  Blynk.begin (auth, ssid, pass);
}

void loop() {
  Blynk.run();
}
```

Done Saving.

The sketch name had to be modified.  
Sketch names must start with a letter or number, followed by letters, numbers, dashes, dots and underscores. Maximum length is 63 characters.

17

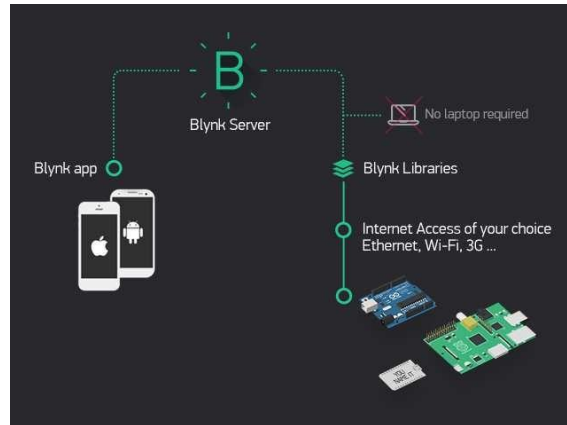
### **Gambar 4.4 source code program**

#### 4.2.5 Smart Relay dengan aplikasi Android Blynk IoT

Blynk adalah platform Internet of Things (IoT) yang memungkinkan pengguna untuk menghubungkan perangkat fisik dengan aplikasi ponsel atau perangkat lainnya melalui koneksi internet. Dengan Blynk, Anda dapat mengendalikan dan memonitor perangkat seperti mikrokontroler (seperti Arduino, Raspberry Pi) dan berbagai sensor atau perangkat lainnya dari jarak jauh. Platform Blynk memungkinkan Anda membuat antarmuka pengguna (UI) yang disesuaikan di aplikasi ponsel Anda, yang memungkinkan Anda berinteraksi dengan perangkat yang terhubung. Anda dapat membuat tombol, slider, grafik, tampilan status, dan elemen UI lainnya untuk mengontrol perangkat atau menerima data dari mereka. Selain itu, Blynk juga mendukung integrasi dengan layanan pihak ketiga seperti Twitter, Email, dan banyak lagi, untuk membuat proyek IoT yang lebih kompleks.

Secara umum, Blynk menyederhanakan proses mengembangkan proyek-proyek IoT dengan menyediakan antarmuka yang mudah digunakan dan dukungan untuk berbagai jenis perangkat keras. Anda dapat menghubungkan perangkat Anda ke server Blynk menggunakan protokol komunikasi seperti Wi-Fi dan Internet, tergantung pada kemampuan perangkat keras yang digunakan.

Penggunaannya juga mudah, tinggal *drag and drop* sesuai kebutuhan sistem yang akan dibuat. Blynk dapat dijalankan di Android maupun iOS.



Tidak perlu pemrograman yang rumit dengan komputer/laptop, kita hanya menginstal Blynk di Android kemudian mulai membuat aplikasinya. Cukup banyak komponen/widget yang disediakan oleh Blynk, mulai dari komponen kontrol seperti tombol (button), slider, joystick. Ada juga komponen untuk display seperti LCD, LED dll



Aplikasi ini gratis tetapi ada batasan penggunaan komponennya (widget). Untuk versi gratis dibatasi maksimal 2000 (nilai akumulasi widget).

#### Kebutuhan Jaringan Internet

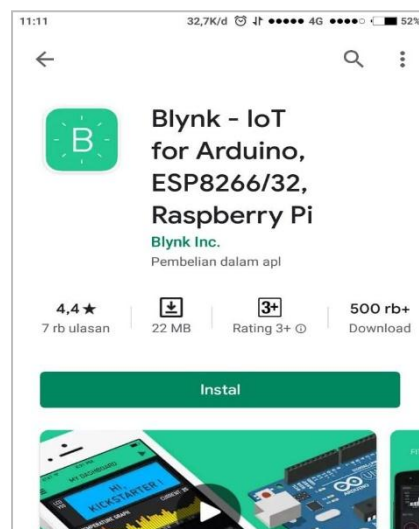
Siapkan sebuah WiFi (hotspot) dengan nama WiFi (ssid) dan password. Anda dapat membuatnya melalui hotspot dari internet HP (tethering).

- Nama jaringan WiFi/Hotspot
- Password

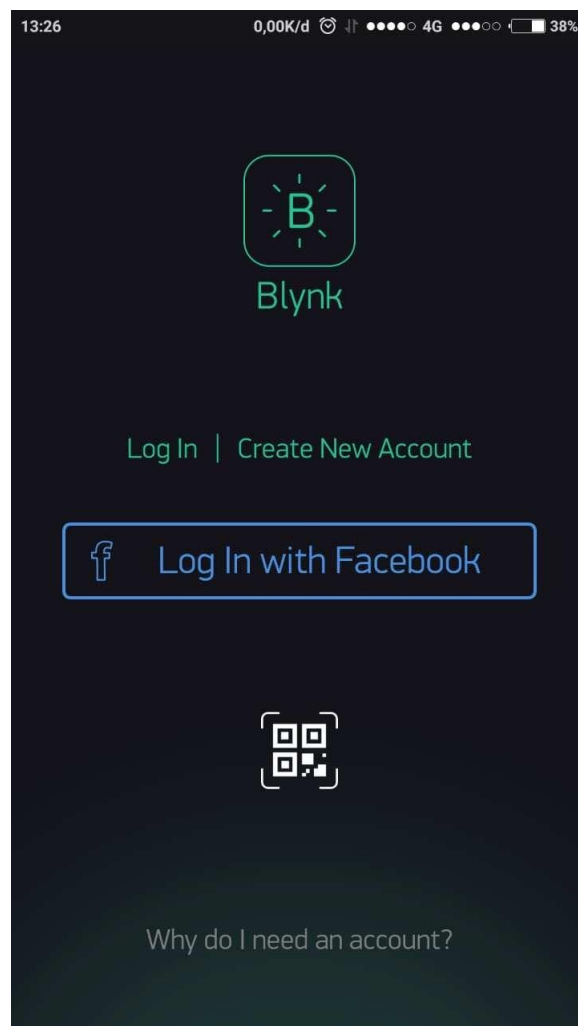
#### Kebutuhan Software

- Arduino IDE
- Blynk

Sekarang silahkan instal terlebih dahulu aplikasi Blynk. Pada panduan ini kita memakai Android. Buka **Play Store** dari smartphone Android anda kemudian cari “Blynk”.

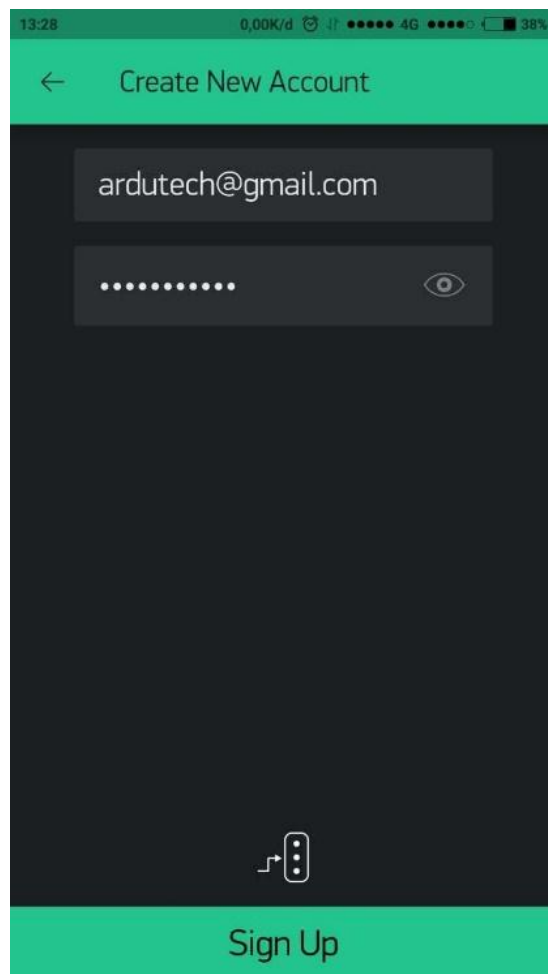


Install sampai selesai kemudian buka/jalankan Blynk dari HP Android anda. Pertamakali akan diminta untuk registrasi, siapkan sebuah email aktif dan password.





Pilih **“Create New Account”** jika anda belum punya akun Blynk.

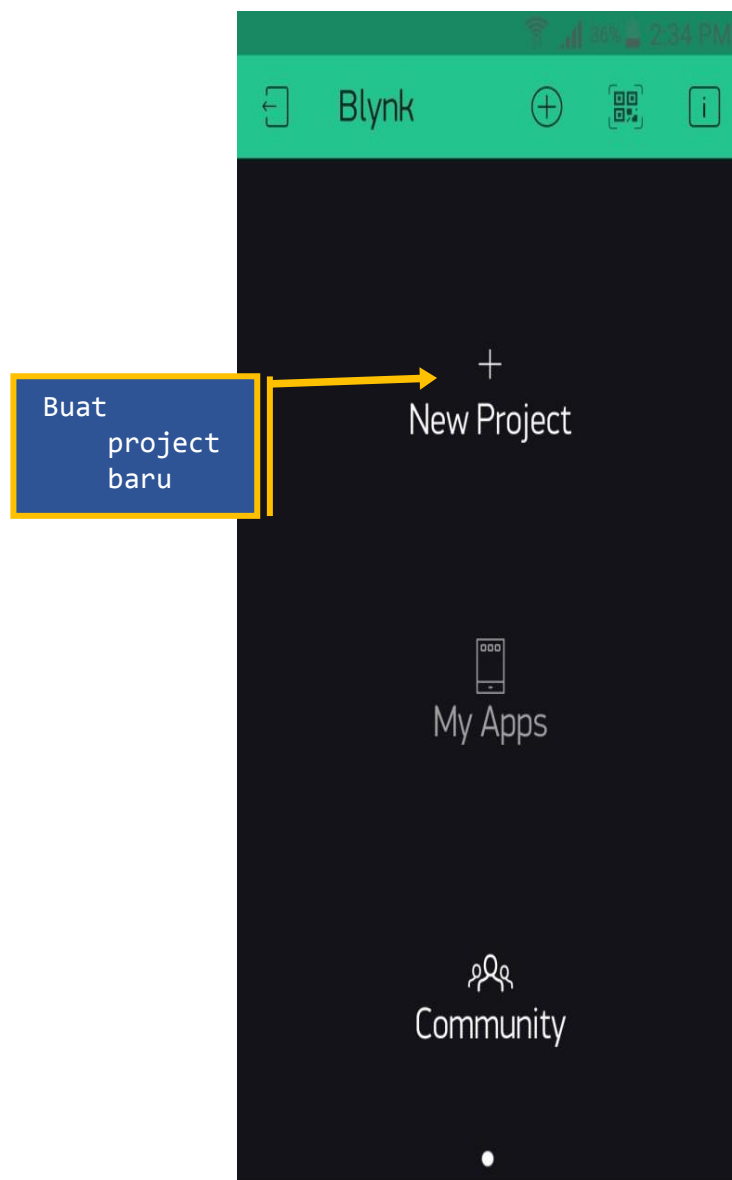


Isi alamat email dengan email aktif dan juga password kemudian klik **“Sign Up”**. Buka email anda untuk verifikasi proses pembuatan akun.

Email ini nanti juga akan dipakai untuk menerima kode (token) ketika

kita membuat sebuah project di Blynk.

Jika proses pembuatan akun sudah selesai berikutnya kita akan membuat sebuah project “**SmartRelay**”

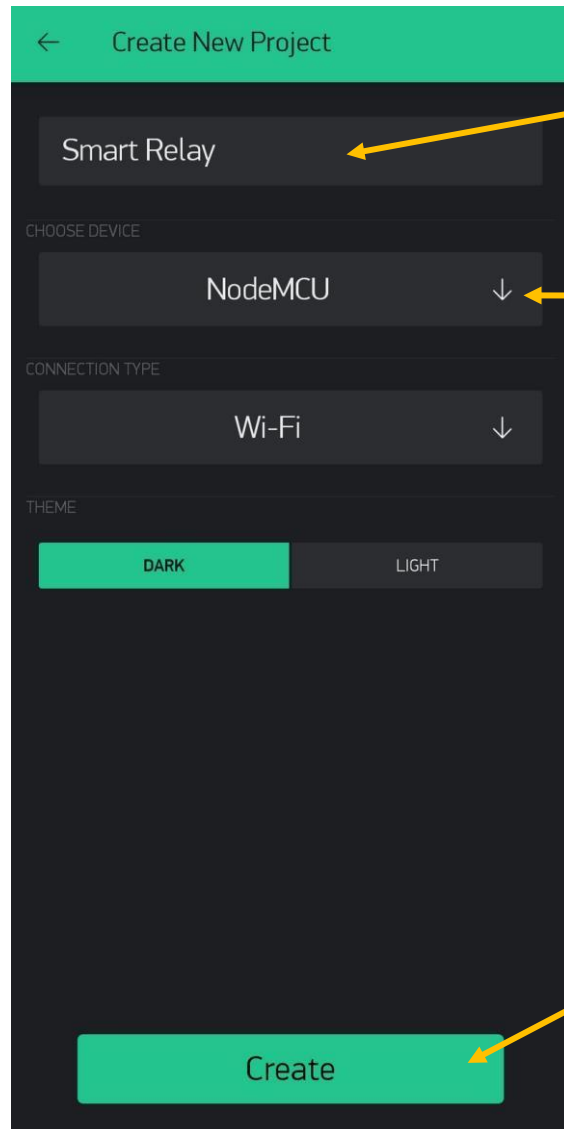


Klik “+ **New Project** “ untuk membuat project baru.

Muncul tampilan baru kemudian isi nama project  
bagian CHOOSE DEVICE kemudian pilih **NodeMCU**. Untuk CONNECTION TYPE : **Wi-Fi**.

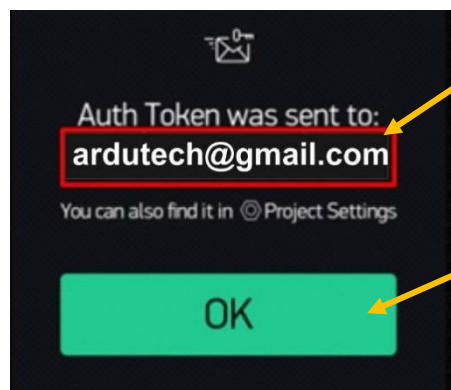
Tulis Judul Project

Klik. Pilih NodeMCU



Klik "Create"

Klik tombol **Create** sehingga kode (**token**) Blynk akan dikirim ke email akun anda.



Token dikirim ke email

Klik "OK"

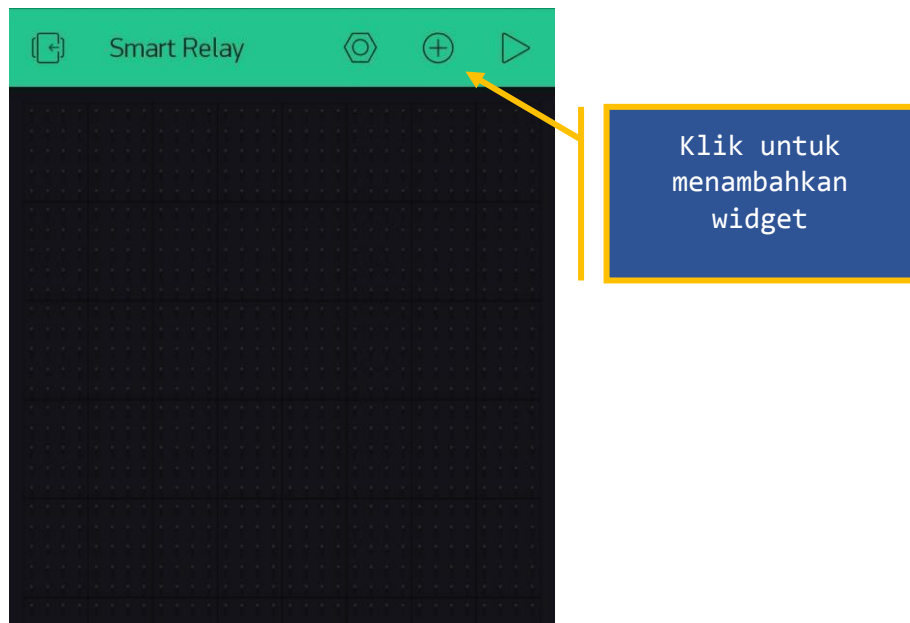
Silakan buka email-nya kemudian cek karena nanti akan dipakai pada pemrograman dengan Arduino IDE. Catat kode (*token*) tersebut.

```
Blynk <dispatcher@blynk.io> Berhenti berlangganan
Kepada saya ▾
Auth Token : 3-pngJHEZ0kVIEL1w0Br1pev-4_38bJk
Happy Blynking!
-
Getting Started Guide -> https://www.blynk.cc/getting-started
Documentation -> http://docs.blynk.cc/
Sketch generator -> https://examples.blynk.cc/
```

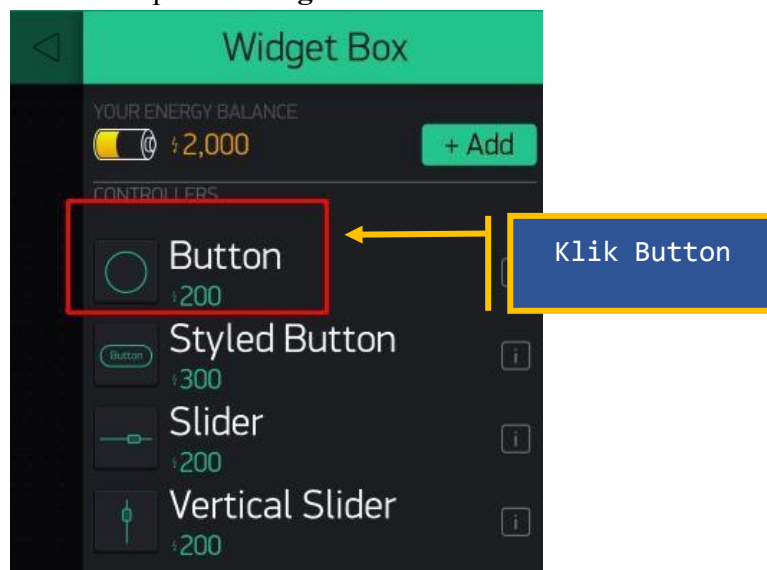
TOKEN

Kode/token. Catat atau Copy paste ke source code Arduino

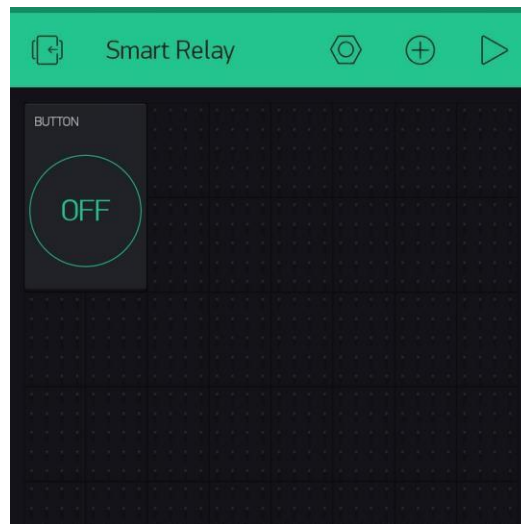
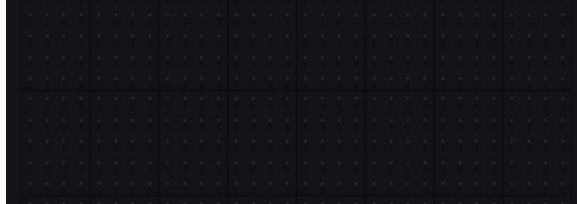
Berikutnya pada lembar kerja project "**Smart Relay**", tambahkan widget dengan klik tombol '+' :



Selanjutnya muncul tampilan “**Widget Box**”. Pilih “**Button**”.



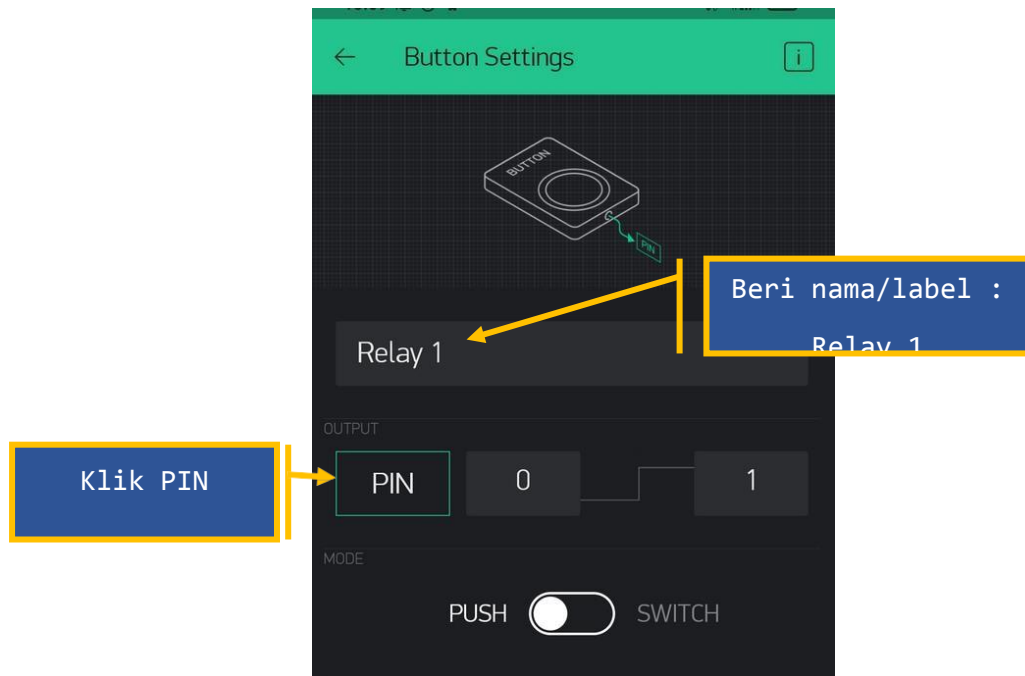
Selanjutnya akan muncul sebuah tombol (Button) di lembar kerja Blynk.



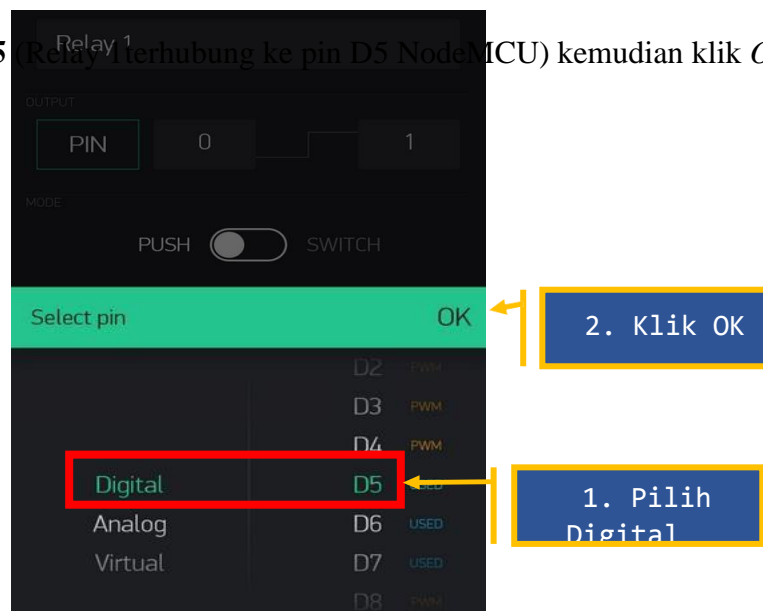
Berikutnya pada lembar kerja, dengan cara yang sama tambahkan 3 buah widget **Button** lagisehingga total ada 4 buah Button.

Selanjutnya kita seting untuk masing – masing Button, kita mulai dari **BUTTON 1**, klik **BUTTON 1**.



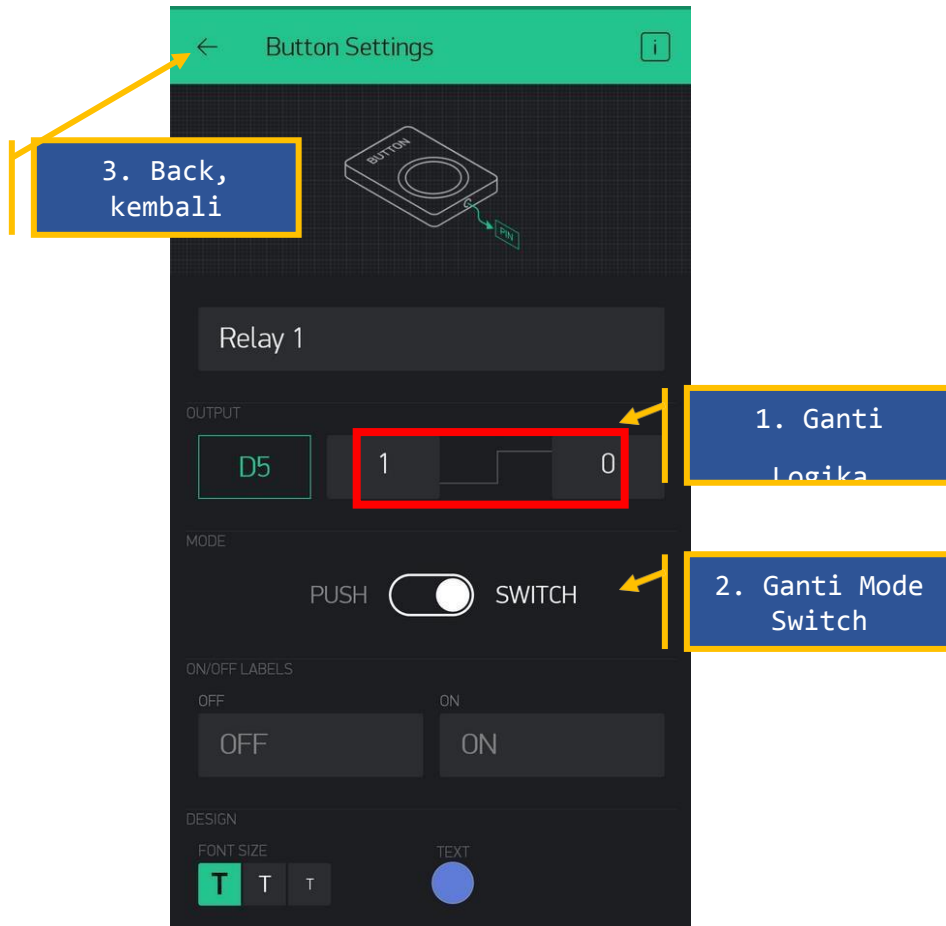


Beri nama/label “**Relay 1**”, kemudian pada bagian OUTPUT, klik **PIN** . Pilih **Digital D5** (Relay 1 terhubung ke pin D5 NodeMCU) kemudian klik **OK**.



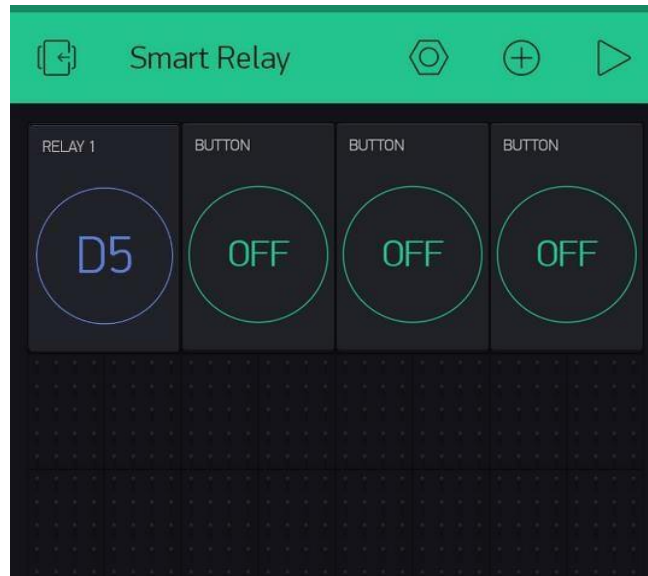
Logika juga diganti karena relay-nya aktif LOW (jika input diberi logika “LOW” relay akan “ON” dan

sebaliknya ). MODE diganti ke SWITCH.



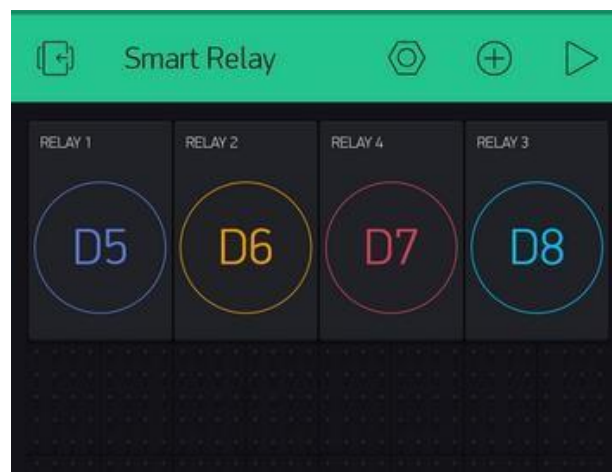
Klik tombol '*back*' di pojok kiri atas untuk kembali ke tampilan utama.





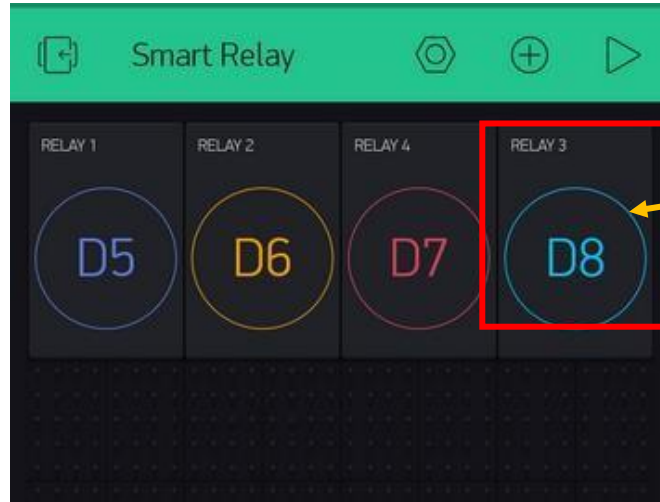
Lakukan untuk 3 BUTTON yang lain dengan parameter yang sama (logika diganti 1 □ 0 dan MODESWITCH).

BUTTON	Label	OUTPUT
2	Relay 2	D6
3	Relay 3	D7
4	Relay 4	D8

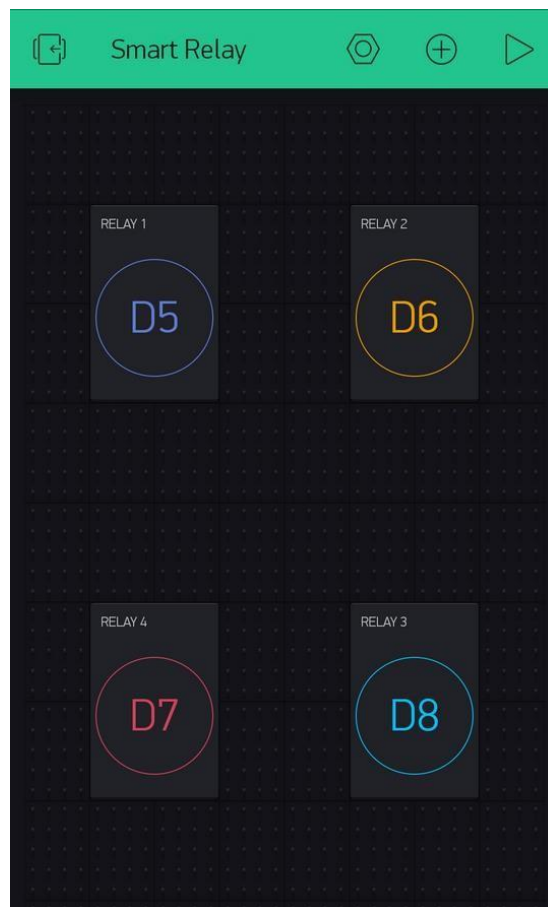


Tata letak dan ukuran silakan diatur sendiri. Untuk menggeser letak/posisi button, **klik dan tahan**

button kemudian geser ke tempat yang dikehendaki.



Tekan + tahan  
kemudian geser  
di tempat yg  
dikehendaki.

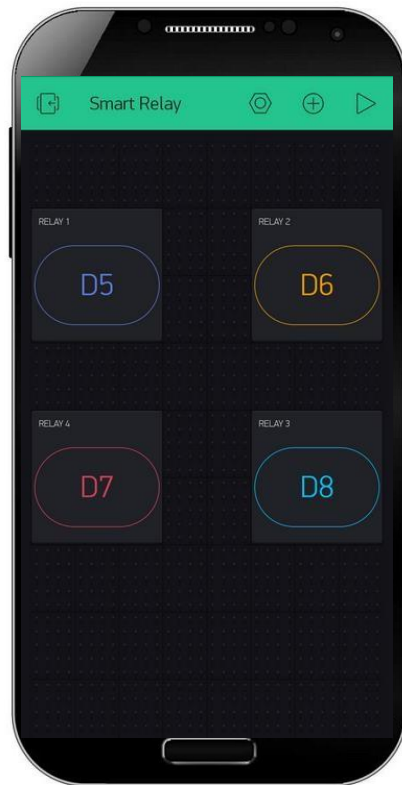


Untuk merubah ukuran widget (ini juga tidak wajib, hanya untuk tampilan saja) **klik/tekan + tahan sebentar** widgetnya kemudian **lepaskan** hingga

muncul “frame” dengan dua (2) titik disebelah kanan dan kiri. Tarik ke –kanan atau ke-kiri agar ukuran berubah.



Berikut ini contoh untuk ukuran yang sudah dirubah (sekali lagi ini tidak wajib, hanya untuktampilan saja).



Selanjutnya kita siapkan software Arduino IDE.

Program di Arduino IDE.

Program ini dibuat untuk ‘diisikan’ ke NodeMCU V3 agar dapat bekerja sesuai dengan sistem yang kita buat yaitu Smart Relay dengan Blynk. Pastikan anda sudah mempunyai Arduino IDE dan seting untuk NodeMCU juga sudah beres. Kalau belum silakan baca lagi “**Pemrograman Dasar NodeMCU dg Arduino IDE**” di awal panduan ini. Library juga jangan lupa untuk disertakan.

Buka/jalankan Arduino IDE kemudian buat lembar kerja baru. Tulis kode program berikut.

```
/*  
* SMART RELAY KONTROL VIA BLYNK
```

```

* Board   : NodeMCU V3
* Input   : Blynk
* Output  : Relay 4 Channel
* IoT Starter SMART RELAY
* www.ardutech.com

*****/

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>
//---GANTI SESUAI DENGAN TOKEN BLYNK ANDA
char auth[] = "CbS08agGfefffJ9GJ6b3G5kKE5Q4S";
//---GANTI SESUAI DENGAN JARINGAN WIFI
//---HOTSPOT ANDA
char ssid[] = "Ardutech"; // Nama Hotspot/WiFi
char pass[] = "12345678"; // Password
LiquidCrystal_I2C lcd(0x27, 16, 2);
boolean st;
int g;
//=====
void cek_koneksi(){
    st=Blynk.connected();
    if(st==true){
        lcd.setCursor(0, 1);

```

```

    lcd.print("Koneksi Sukses");
}
else{
    lcd.setCursor(0, 1);
    lcd.print("Koneksi Gagal");
}
}
//=====

void setup()
{
    pinMode(D5,OUTPUT);
    pinMode(D6,OUTPUT);
    pinMode(D7,OUTPUT);
    pinMode(D8,OUTPUT);
    digitalWrite(D5,HIGH);
    digitalWrite(D6,HIGH);
    digitalWrite(D7,HIGH);
    digitalWrite(D8,HIGH);
    lcd.begin(16,2);
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("Smart Relay 4Ch");

    lcd.setCursor(0, 1);
    lcd.print("Kontrol Blynk");
    delay(1000);
    lcd.setCursor(0, 1);
    lcd.print("Tunggu Koneksi..");
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
    cek_koneksi();
}

```

```
//=====
void loop()
{
  Blynk.run();
  g++;
  if(g>300){
    g=0;
    cek_koneksi();
  }
}
}
```

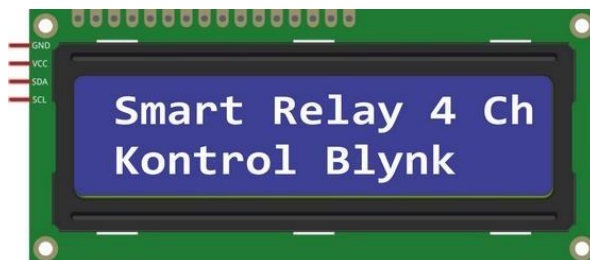
Ganti/sesuaikan variabel berikut :

- Nama jaringan WiFi/hotspot : **ssid [ ]**
- Password jaringan WiFi/hotspot : **pass [ ]**
- Kode token Blynk : **auth[ ]**

Simpan (*Save*) kemudian *Upload*. Pastikan tidak ada error, jika masih ada silakan cek penulisan dll kemudian perbaiki. (Program ini sudah diuji langsung dan sudah berjalan tanpa ada error)

Jalannya Alat

Setelah program berhasil di Upload maka pada LCD tampil :



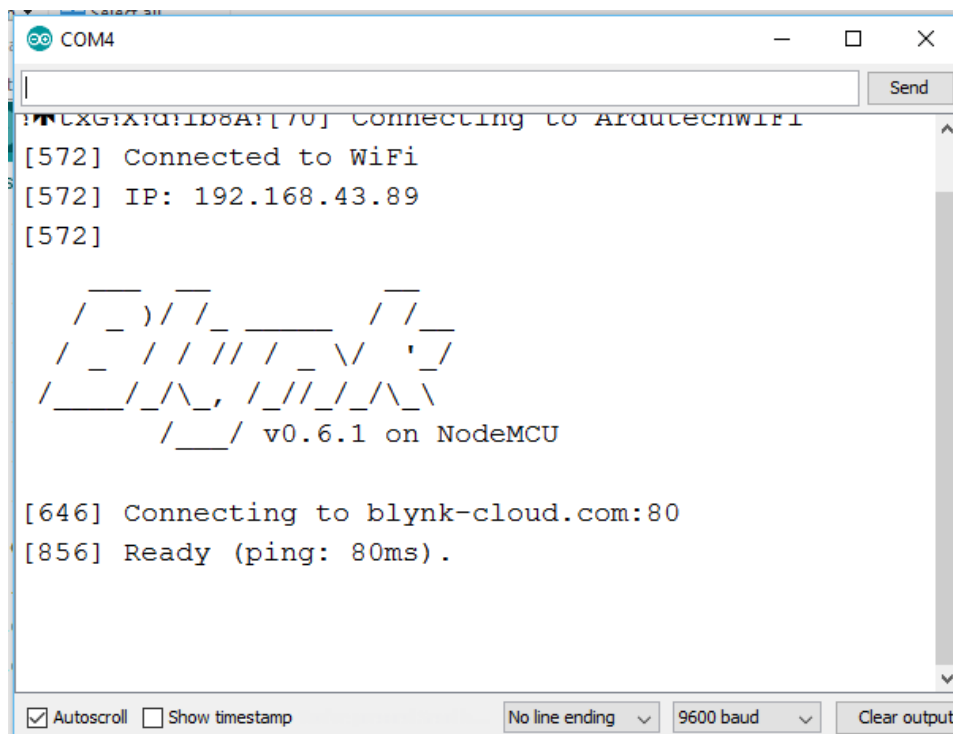
Setelah sekitar 1 detik menjadi :



Jika koneksi sukses maka tampilan LCD menjadi :

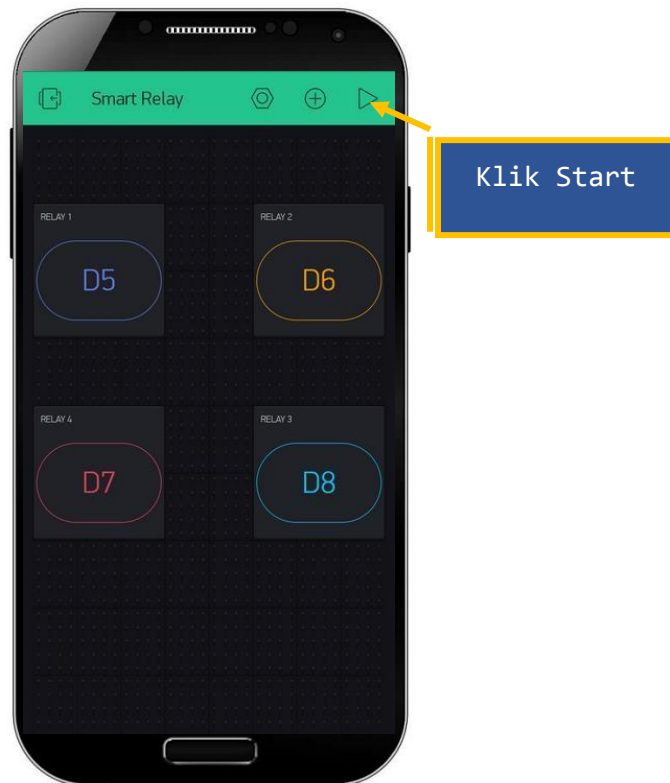


Silakan buka Serial Monitor dari menu **Tools**  **Serial Monitor**, seting baudrate pada 9600 :

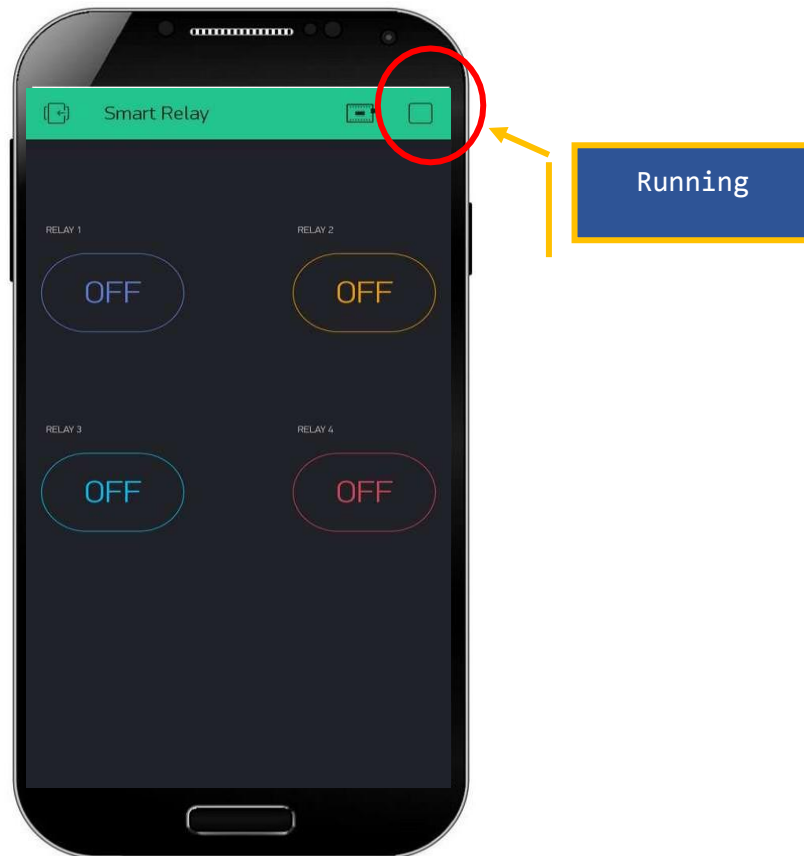




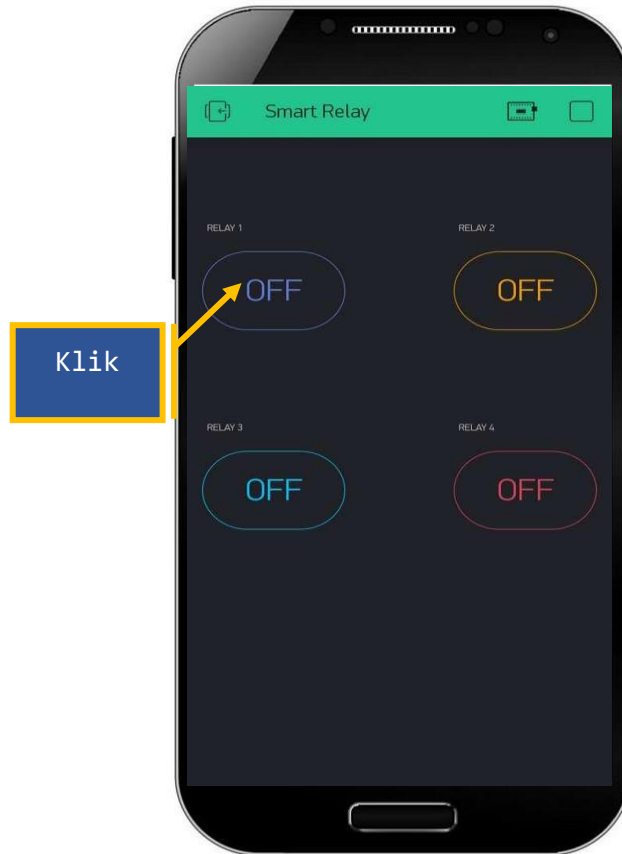
Jika sudah terhubung dengan server Blynk selanjutnya kita jalankan aplikasi Blynk di Android yang tadi sudah dibuat. Klik tombol Start (pojok kanan atas) sehingga tampil antarmuka untuk **Smart Relay**.

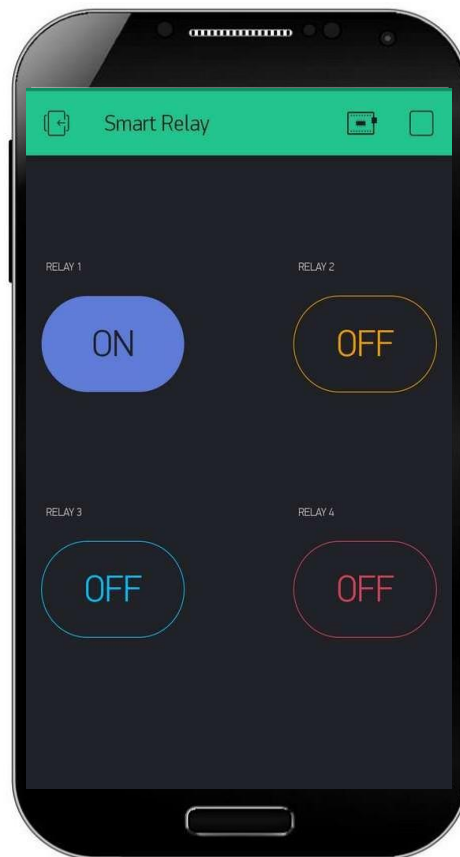


Jika aplikasi berjalan dengan normal maka akan tampil seperti gambar berikut

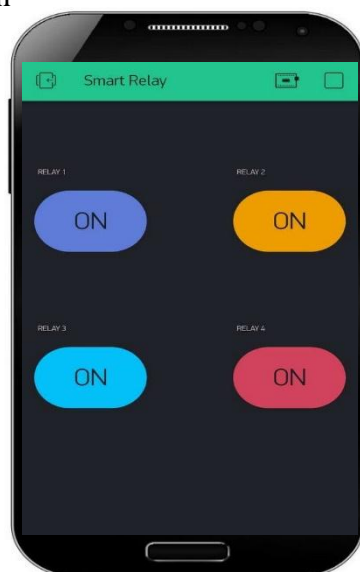


Kita coba menyalakan Relay 1, klik tombol RELAY 1 (BUTTON 1).





Perhatikan Relay 1 juga akan nyala. Coba juga untuk kontrol Relay 2 serta Relay yang lain



### 4.3 Pengujian Sistem

Rancang bangun kontrol lampu menggunakan IoT harus berfungsi dengan baik dan terbebas dari kesalahan atau error. Oleh karena itu diperlukan sebuah pengujian sistem untuk menghindari kesalahan pada saat diimplementasikan pada prototype kontrol lampu.

Pengujian pada penelitian ini dibagi menjadi 4 yaitu, pengujian delay menyalakan lampu, pengujian delay mematikan lampu, pengujian delay dan jaraknya hidupkan lampu dan pengujian delay dan jaraknya matikan lampu.

#### 4.3.1 Pengujian delay menghidupkan lampu

Pengujian delay menyalakan lampu ini bertujuan untuk mengetahui berapa jarak waktu yang di butuhkan untuk menyalakan lampu pada alat prototype kontrol lampu rumah, berikut tabel 4.2.1 adalah hasil pengujiannya

no	Lampu	Waktu		Delay(detik)	Keterangan
		Off	On		
1	Lampu 1	16.50	16.50	1.00	Lampu Hidup
2	Lampu 2	16.55	16.55	1.00	Lampu Hidup

*delay hidup lampu*

t

abel

4.3.1

Peng

ujian

#### 4.3.2 Pengujian delay mematikan lampu

Pengujian delay mematikan lampu ini bertujuan untuk mengetahui berapa jarak waktu yang di butuhkan untuk mematikan lampu pada alat prototype kontrol lampu, berikut tabel 4.2.2 adalah hasil pengujiannya:.

no	Lampu	Waktu		Delay(detik)	Keterangan
		Off	On		
1	Lampu 1	17.00	17.00	1.00	Lampu Mati
2	Lampu 2	17.10	17.10	1.00	Lampu Mati

**tabel 4.3.2** Pengujian *delay* matikan lampu

Pengujian delay dan jaraknya menghidupkan lampu ini bertujuan untuk mengetahui berapa jarak waktu yang di butuhkan untuk menghidupkan lampu pada alat prototype kontrol lampu, berikut tabel 4.3.3 adalah hasil pengujiannya:

**Tabel 4.3.3** Pengujian *delay* dan jaraknya hidupkan lampu

No	Lampu	Jarak	waktu		Delay(detik)	Keterangan
			on	off		
1	Lampu 1	2 meter	16.50	16.50	1.00	Lampu Hidup

2	Lampu 2	3 meter	17.00	17.00	1.00	Lampu Hidup
---	---------	---------	-------	-------	------	-------------

Pengujian delay dan jaraknya mematikan lampu ini bertujuan untuk mengetahui berapa jarak waktu yang di butuhkan untuk menghidupkan lampu pada alat prototype kontrol lampu, berikut tabel 4.3.4 adalah hasil pengujiannya:

**Tabel 4.3.4** Pengujian *delay* dan jaraknya matikan lampu

No	Lampu	Jarak	waktu		Delay(detik)	Keterangan
			on	off		
1	Lampu 1	2 meter	17.00	17.00	1.00	Lampu Mati
2	Lampu 2	3 meter	17.10	17.10	1.00	Lampu Mati