

## BAB III METODE PENELITIAN

### 3.1 ALAT DAN BAHAN YANG DIGUNAKAN

Pada penelitian ini, tentu saja terdapat beberapa alat dan bahan yang digunakan untuk membuat sistem sterilisasi *Minim* sentuh menggunakan *ultraviolet-C* berbasis *internet of things*. Alat dan bahan yang akan digunakan yaitu berupa Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*) dimana perangkat keras itu sendiri terdiri dari Wemos D1 *Mini* sebagai mikropengendali utama dan UVC *LED Strips* sebagai perangkat yang bertugas untuk mensterilkan setiap benda yang akan di sterilkan oleh pengguna, selain itu ada juga *Obstacle Infrared Sensor* sebagai tombol yang *Minim* sentuh saat system akan digunakan. Kebutuhan Perangkat Keras dan Perangkat Lunak yang digunakan dalam penelitian ini dapat dilihat pada tabel dibawah, sebagai berikut.

**Tabel 3.1 Kebutuhan Perangkat Keras (*Hardware*)**

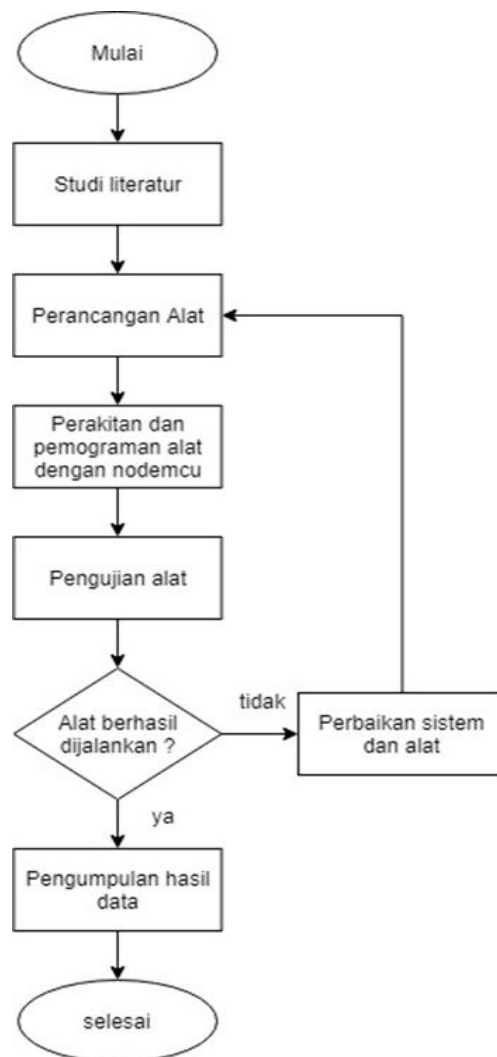
NO	Alat & Bahan	Keterangan
1	Laptop	Merk ASUS
2	<i>Power Supply Board</i>	3.3V-5V
3	Wemos D1 <i>Mini</i>	ESP-8266
4	Kabel <i>micro usb</i>	TYPE-B
5	<i>Obstacle Infrared Sensor</i>	FC-51
6	<i>Limit switch Module</i>	FYSETC
7	<i>Relay 1 Channel</i>	10A 5V
8	<i>Servo Motor Gear</i> Logam	MG99R
9	UVC <i>LED Strips</i>	254nm

**Tabel 3.2 Kebutuhan Perangkat Lunak (*Software*)**

NO	Nama <i>Software</i>
1	Arduino IDE
2	<i>Web Browser</i>

### 3.2 ALUR PENELITIAN

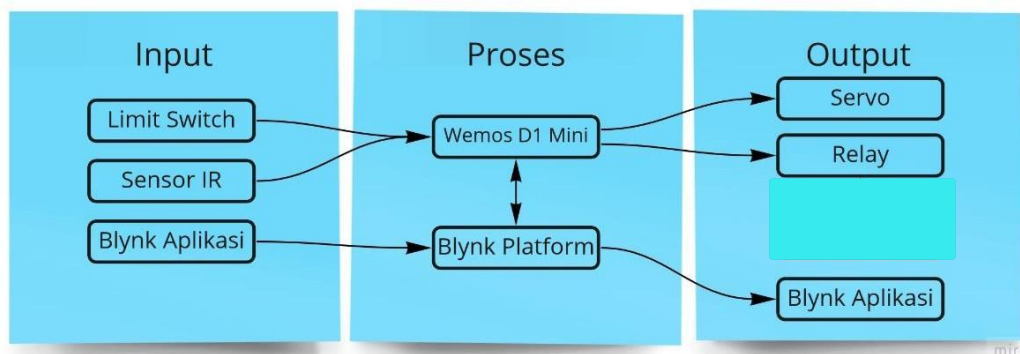
Agar penelitian ini cepat terselesaikan dan tidak menyimpang dari tujuan awal, maka dibuatlah sebuah alur penelitian mengenai implementasi sistem sterilisasi *Minim* sentuh menggunakan *Ultraviolet-C* berbasis *internet of things*. Dengan dibuatnya alur penelitian maka penelitian ini dilakukan dengan lebih terstruktur dan terarah, dimulai dari perancangan *hardware* kemudian perancangan sistem, setelah itu dilakukan pengujian fungsionalitas *hardware* dan sistem, jika terjadi masalah atau pengujian tidak sesuai dengan tujuan yang diharapkan maka akan langsung dilakukan perbaikan sistem dan dilakukan pengujian ulang sampai hasil pengujian sesuai dengan yang diharapkan, setelah itu dilakukan analisis hasil pengukuran perangkat berdasarkan pengujian sebelumnya untuk dibuat kesimpulan mengenai penelitian yang dilakukan.



**Gambar 3.1 Flowchart Alur Penelitian**

### 3.2.1 PERANCANGAN SISTEM KESELURUHAN

Setelah menentukan alur penelitian serta mengumpulkan alat dan bahan yang akan digunakan pada penelitian ini. Maka tahap selanjutnya yaitu melakukan perancangan sistem keseluruhan baik itu *hardware* maupun *software*. Perancangan ini dibagi menjadi 2 (dua) bagian yaitu perancangan *hardware* dan perancangan *software*. Alur kerja sistem keseluruhan sendiri dapat dilihat pada Gambar 3.2 dibawah ini.



**Gambar 3.2 Blok Diagram Alur Kerja Sistem**

Berdasarkan gambar di atas, sama seperti sebuah sistem pada umumnya, sistem yang akan dibuat ini memiliki 3 (tiga) proses utama yaitu *input*, proses dan *output*. Dimana bagian *input* adalah perangkat antarmuka utama yang digunakan oleh pengguna untuk memberikan perintah. Bagian proses yaitu dimana setiap perintah yang diberikan oleh pengguna di proses oleh mikrokontroler Wemos D1 Mini dan Blynk Platform yang akan menyimpan setiap perintah dari pengguna secara real time. Bagian terakhir yaitu *output* dimana setiap perangkat keluaran akan mengeksekusi perintah yang diberikan oleh pengguna melalui perangkat masukan/*input* yang kemudian perintah tersebut dikirimkan oleh mikroprosesor Wemos D1 Mini untuk diteruskan ke perangkat *output*.

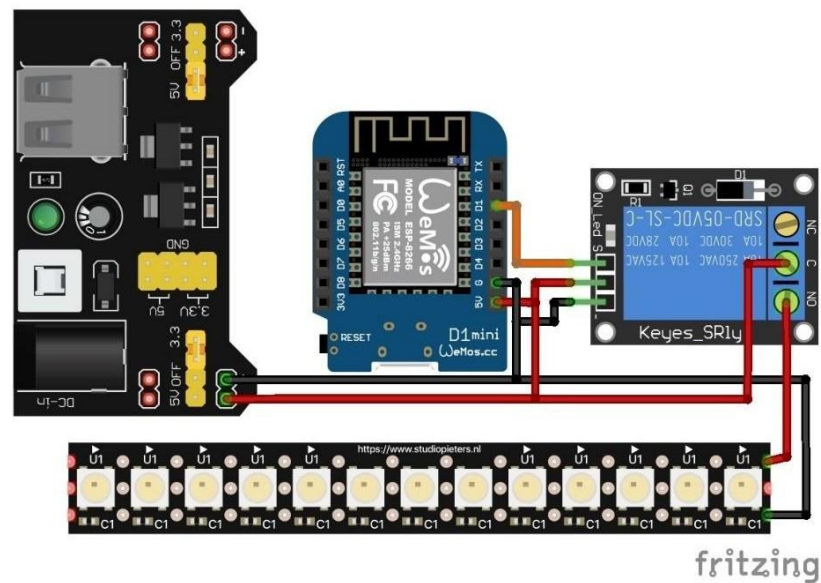
### 3.2.2 PERANCANGAN HARDWARE

Sesuai dengan susunan pada alur penelitian pada poin sebelumnya, tahap pertama yang dilakukan pada penelitian ini yaitu perancangan *hardware*. Tahap ini akan menjelaskan mengenai perangkat keras apa saja yang

digunakan serta rangkaian-rangkaian dari setiap perangkat yang saling terhubung satu sama lain.

### 1. Relay & LED UVC

Perangkat pertama yang dirancang adalah perangkat yang paling utama dalam sistem ini, yaitu *LED UVC (Ultraviolet-C)* dimana perangkat ini merupakan perangkat utama dalam sistem yang akan dibuat, berfungsi sebagai alat yang akan melakukan sterilisasi setiap benda yang dimasukkan ke dalam perangkat. Cara kerja perangkat ini yaitu melakukan sterilisasi dengan menggunakan sinar *Ultraviolet*. Perangkat ini juga langsung dihubungkan dengan *relay 1 channel* yang berfungsi sebagai saklar ketika sistem ini dihidupkan/dimatikan. Rancangan perangkat lebih jelasnya dapat dilihat pada gambar dibawah ini.



**Gambar 3.3 Perancangan Relay & LED UVC**

**Tabel 3.3 Pin terhubung power supply board dengan Wemos D1 Mini**

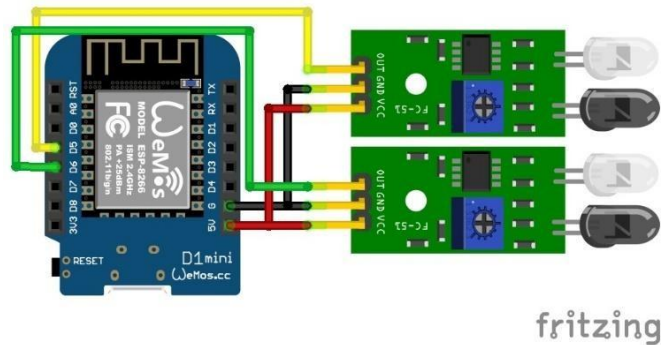
<i>Power Supply Board</i>	<i>Wemos D1 Mini</i>
+	5V
-	G

**Tabel 3.4 Pin terhubung *relay* dengan *Wemos D1 Mini***

<i>Relay</i>	<i>Wemos D1 Mini</i>
VCC	5V
GND	G
IN	D1

**2. *Infrared Sensor***

Selanjutnya yaitu *Infrared Sensor* dimana perangkat ini berperan sebagai pengganti tombol ketika pengguna ingin menggunakan sistem ini. Cara kerja *Infrared Sensor* sendiri yaitu perangkat ini dapat mengidentifikasi penghalang sehingga jika ada penghalang di depan sensor maka sensor akan memberitahu bahwa penghalang terdeteksi. Berdasarkan cara kerja tersebut maka *Infrared Sensor* pada sistem ini digunakan sebagai pengganti tombol, sehingga pengguna tidak perlu menempelkan jarinya ke tombol ketika akan menggunakan sistem ini. Rancangan perangkat lebih jelasnya dapat dilihat pada gambar dibawah ini.



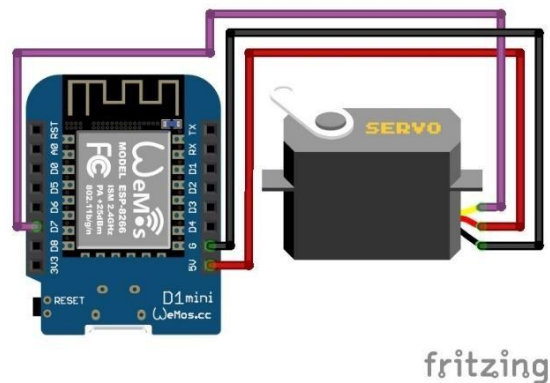
**Gambar 3.4 Perancangan *Infrared Sensor***

**Tabel 3.5 Pin terhubung *infrared sensor* dengan *Wemos D1 Mini***

<i>Infrared Sensor</i>	<i>Wemos D1 Mini</i>
VCC	5V
GND	G
OUT	D5 dan D6

### 3. Servo

Perangkat selanjutnya yang dirancang yaitu *servo*, dimana alat ini berfungsi sebagai alat penggerak untuk membuka dan menutup sistem ketika benda yang akan di sterilkan sudah dimasukkan. Rancangan perangkat lebih jelasnya dapat dilihat pada gambar dibawah ini.



**Gambar 3.5 Perancangan Servo**

**Tabel 3.6 Pin terhubung *servo* dengan Wemos D1 Mini**

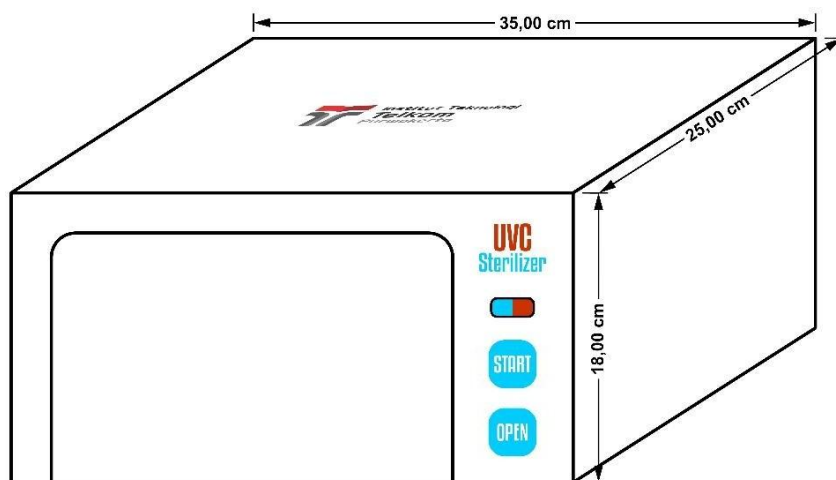
<i>Servo</i>	<i>Wemos D1 Mini</i>
VCC	5V
GND	G
<i>Pulse</i>	D7

### 4. Limit switch

Perancangan perangkat yang terakhir yaitu *Limit switch*, dimana cara kerja alat ini seperti magnet, yang mana bila pintu sudah tertutup rapat maka perintah akan dilanjutkan dengan *LED UVC* yang menyala dan proses sterilisasi benda dimulai. Jika *Limit switch* membaca bahwa pintu belum tertutup rapat maka lampu *LED UVC* tidak akan menyala dan sistem tidak akan bekerja sampai *Limit switch* membaca bahwa pintu tertutup rapat. Rancangan perangkat lebih jelasnya dapat dilihat pada gambar dibawah ini.



**Gambar 3.7 Rancangan *Hardware* Keseluruhan**



**Gambar 3.8 Rancangan *Case* Prototipe**

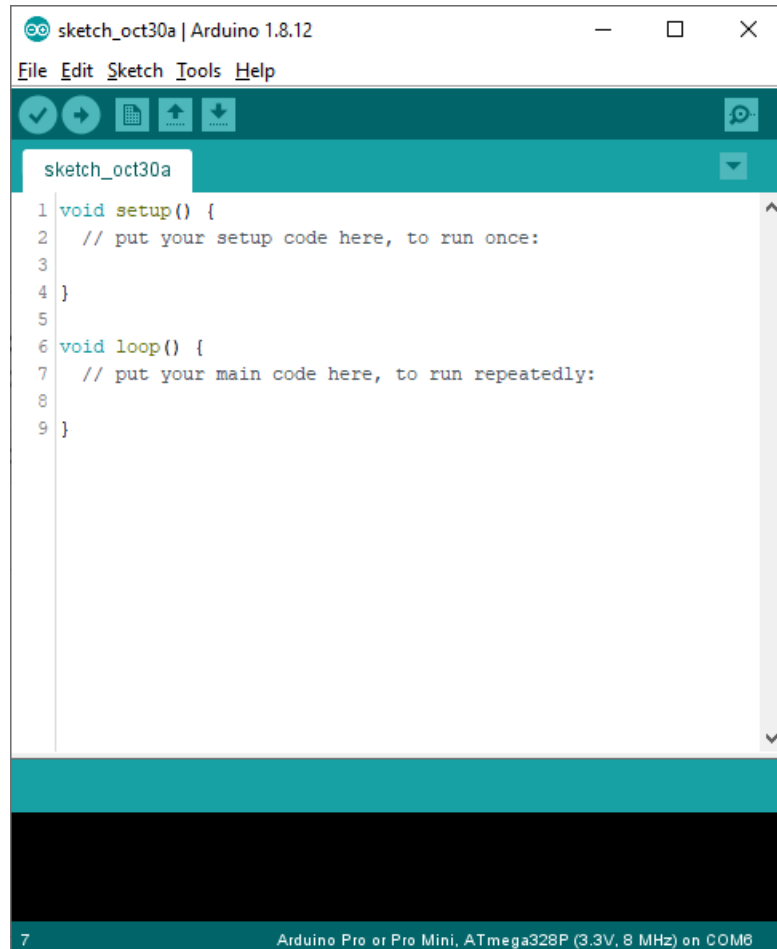
### **3.2.3 PERANCANGAN SOFTWARE**

Setelah perancangan masing-masing perangkat keras, maka tahap selanjutnya yaitu melakukan perancangan perangkat lunak atau (*Software*). Tahap ini akan menjelaskan mengenai setiap perangkat lunak yang digunakan sebagai bahan penelitian sistem yang akan dibuat.

#### **1. Arduino IDE**

Pada penelitian ini penulis menggunakan *software* Arduino IDE sebagai *tools* untuk menulis program yang nantinya akan dimasukkan ke mikrokontroler Wemos D1 *Mini* dengan bentuk perintah-perintah tertentu sehingga mikrokontroler dapat menghubungkan masing-masing perangkat keras yang dibutuhkan dalam penelitian ini.





**Gambar 3.9 Software Arduino IDE**

```
1 // ESP8266 Blynk
2 #define BLYNK_TEMPLATE_ID "TMPLpbQ5m5On"
3 #define BLYNK_DEVICE_NAME "UVC"
4 #define BLYNK_AUTH_TOKEN "1jK0tc5XJs1LbD0wSwdGHXnm8S6G6TfZ"
5 #include <BlynkSimpleEsp8266.h>
6 char auth[] = BLYNK_AUTH_TOKEN;
7 char ssid[] = "16101165"; // SSID WiFi
8 char pass[] = "12345678"; // Password WiFi
```

**Gambar 3.10 Library Blynk Arduino IDE**

Pada bagian ini, *library* Blynk disisipkan agar semua perintah yang akan dijalankan menggunakan perintah blynk dapat dieksekusi dengan baik termasuk SSID dan *password* WiFi yang akan digunakan.

```

10 // Servo Cover
11 #include <Servo.h>
12 Servo cover;
13 #define servoCover D7
14
15 // deklarasi pin
16 #define limitswitch D8 // pin limit switch cover
17 #define infraredOpen D5 // pin infrared tombol open
18 #define infraredStart D6 // pin infrared tombol start
19 #define relayUVC D1 // pin relay UVC
20 int terbuka = 0;
21
22 // pin LED RGB
23 #define Red D3
24 #define Green D2
25 #define Blue D4

```

**Gambar 3.11 Deklarasi pin Arduino IDE**

*Library servo* disisipkan disini agar *servo* dapat bekerja dengan baik dan benar serta pin digital yang digunakan oleh *servo* juga didefinisikan. Pendefinisian pin seperti *limit switch*, *infrared sensor*, *relay*, dan LED RGB juga dilakukan pada bagian ini agar pada badan program dapat dipanggil sesuai dengan pendefinisian yang sesuai.

```

27 long minumTimer = 60, // dalam detik
28     timer = minumTimer,
29     timerCover = 30000; // dalam milidetik
30
31 int waktu,
32     bacabuka,
33     bacacover,
34     bacamulai,
35     state = 1,
36     buttonbuka = 0,
37     buttonmulai = 0,
38     aktif = LOW; // jenis relay untuk mengaktifkan UVC
39 unsigned long count = 0,
40     timerCoverA = 0,
41     timerCoverB = 0;

```

**Gambar 3.12 Deklarasi variabel waktu pin Arduino IDE**

Variabel yang akan digunakan pada sebuah program harus dideklarasikan terlebih dahulu agar tidak terjadi *error* ketika program dijalankan. Variabel dapat dideklarasikan seperti pada bagian ini.

```

43 void cyan() {
44   analogWrite(Red, 1023);
45   analogWrite(Green, 0);
46   analogWrite(Blue, 0);
47 }
48
49 void orange() {
50   analogWrite(Red, 0);
51   analogWrite(Green, 512);
52   analogWrite(Blue, 1023);
53 }
54
55 void black() {
56   digitalWrite(Red, LOW);
57   digitalWrite(Green, LOW);
58   digitalWrite(Blue, LOW);
59 }

```

**Gambar 3.13 Deklarasi pewarnaan LED RGB**

LED RGB juga digunakan pada skripsi ini untuk menampilkan proses apa yang sedang terjadi pada prototipe. Penulis membuat subprogram pewarnaan LED RGB agar mudah dalam pemanggilan saat LED RGB perlu digunakan dalam program utama.

```

61 void wifi() {
62   orange();
63   WiFi.begin((char*)ssid, (char*)pass);
64   Serial.print("connecting");
65   while (WiFi.status() != WL_CONNECTED) {
66     Serial.print(".");
67     black();
68     delay(150);
69     orange();
70     delay(350);
71   }
72   cyan();
73   Serial.print("connected: ");
74   Serial.println(WiFi.localIP());
75   Blynk.begin(auth, ssid, pass);

```

**Gambar 3.14 Algoritma terhubung ke WiFi dan Blynk**

Subprogram wifi() dideklarasikan untuk memudahkan dalam pemanggilan. Subprogram wifi() berfungsi untuk menghubungkan Wemos D1 Mini dengan WiFi dan platform Blynk.

```

78 BLYNK_WRITE (V0) {
79   waktu = param.asInt(); // inputan waktu dari aplikasi
80   if (waktu != 0 && waktu > minumTimer / 60) timer = waktu * 60; // logika
81   if (waktu != 0 && waktu <= minumTimer / 60) timer = minumTimer; // logika
82 }
83 BLYNK_WRITE (V1) {
84   buttonbuka = param.asInt(); // inputan tombol buka/tutup dari aplikasi
85 }
86 BLYNK_WRITE (V2) {
87   buttonmulai = param.asInt(); // inputan tombol start/stop dari aplikasi

```

**Gambar 3.15** Pembacaan *virtual* variabel Blynk

BLYNK\_WRITE digunakan untuk membaca data *timer*, *button open*, dan *button start* pada aplikasi Blynk agar Wemos D1 Mini dapat mengikuti perintah yang dioperasikan melalui aplikasi Blynk.

```

90 void bukaCover() {
91   state = 0;
92   for (int pos = 150; pos >= 0; pos -= 1) { // dalam derajat
93     cover.write(pos);
94     delay(10);
95   }
96   timerCoverA = millis();
97   Serial.println("Cover Terbuka");
98   orange();
99   delay(250);
100  Blynk.virtualWrite (V3, "Tertutup");
101 }
102
103 void tutupCover() {
104   state = 1;
105   for (int pos = 0; pos <= 150; pos += 1) { // dalam derajat
106     cover.write(pos);
107     delay(10);
108   }
109   Serial.println("Cover Tertutup");
110   delay(250);
111   Blynk.virtualWrite (V3, "Tertutup");
112   cyan();
113 }

```

**Gambar 3.16** Algoritma buka-tutup cover

Subprogram `bukaCover()` dan `tutupCover()` dideklarasikan pada bagian luar `void setup()` dan `void loop()`. Subprogram ini menggunakan fungsi *servo* yang dapat bergeser sesuai derajat yang diperlukan agar dapat terbuka dan tertutup.

```

115 void aktifUVC() {
116   Serial.println("UVC Aktif");
117   Blynk.virtualWrite (V4, "Aktif");
118   digitalWrite(relayUVC, LOW);
119   while (count < timer) {
120     count++;
121     black();
122     delay(150);
123     cyan();
124     delay(350);
125     Serial.print(".");
126     black();
127     delay(150);
128     cyan();
129     delay(350);
130     if (digitalRead(infraredOpen) == LOW || digitalRead(infraredStart) == LOW) break;
131   }
132   count = 0;
133   digitalWrite(relayUVC, HIGH);
134   Blynk.virtualWrite (V4, "Non-Aktif");
135   Serial.println("");
136   Serial.println("UVC Non-Aktif");
137 }

```

**Gambar 3.17 Algoritma penyinaran UVC**

Algoritma penyinaran UVC dideklarasikan pada subprogram aktifUVC(). Algoritma secara berurutan berisi penulisan “UVC Aktif” pada serial monitor, kemudian pengiriman kata “Aktif” ke aplikasi Blynk, pengaktifan UVC dengan cara menyalakan *relay*, lalu *timer internal* Wemos D1 Mini bekerja sesuai waktu yang sudah ditentukan dengan memperhatikan adanya interupsi tombol untuk membatalkan, lalu terakhir ketika *timer* sudah selesai maka Wemos D1 Mini akan mematikan UVC, mengirimkan kata “Non-Aktif”, dan menuliskan “UVC Non-Aktif” pada serial monitor.

```

139 void tutupAuto() {
140   timerCoverB = millis();
141   if (timerCoverB - timerCoverA >= timerCover && bacacover == HIGH && state == 0) {
142     tutupCover();
143     timerCoverA = millis();
144   }
145 }

```

**Gambar 3.18 Algoritma tutup otomatis**

Algoritma tutup otomatis dibuat agar ketika prototipe UVC *sterilizer* tidak digunakan dalam kurun waktu yang ditentukan maka *cover* dari prototipe akan tertutup secara otomatis.

```

147 void setup() {
148   Serial.begin(115200);
149   pinMode(relayUVC, OUTPUT);
150   pinMode(Red, OUTPUT); // pin D3
151   pinMode(Green, OUTPUT); // pin D2
152   pinMode(Blue, OUTPUT); // pin D4
153   cover.attach(servoCover); // pin D7
154   cover.write(150); // kemiringan servo dalam derajat untuk menutup cover
155   digitalWrite(relayUVC, HIGH); // memastikan UVC non-aktif saat alat dinyalakan
156   orange();
157   wifi();
158 }

```

**Gambar 3.19 Void setup pada Arduino IDE**

Void setup() berisi program inisiasi komunikasi serial antara Wemos D1 *Mini* dengan laptop/PC, pendefinisian pin *relay* dan RGB LED sebagai *output*, pendefinisian *servo*, mengaktifkan LED RGB sebagai status prototipe, dan menghubungkan prototipe dengan WiFi.

```

160 void loop() {
161   Blynk.run();
162
163   if (WiFi.status() != WL_CONNECTED) wifi();
164
165   bacabuka = digitalRead(infraredOpen); // membaca button open
166   bacamulai = digitalRead(infraredStart); // membaca button start
167   bacacover = digitalRead(limitswitch); // membaca limit switch cover
168
169   // logika tombol open close
170   if (bacabuka == LOW || buttonbuka == 1) { // jika button buka ditekan
171     if (state == terbuka) tutupCover();
172     else if (state == !terbuka) bukaCover();
173   }
174
175   // logika tombol start stop
176   else if (bacamulai == LOW || buttonmulai == 1) { // jika button mulai ditekan
177     if (bacacover == HIGH) tutupCover();
178     aktifUVC();
179     bukaCover();
180   }
181
182   if (bacacover == HIGH) { // jika cover terbuka
183     tutupAuto();
184     Blynk.virtualWrite (V3, "Terbuka");
185   }
186   else Blynk.virtualWrite (V3, "Tertutup");
187   Serial.println("Standby mode");
188   delay(250);
189 }

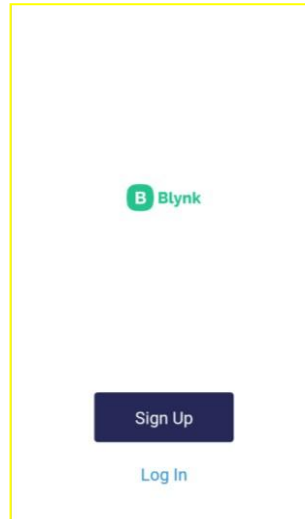
```

**Gambar 3.20 Void loop pada Arduino IDE**

Bagian terakhir dari program prototipe yaitu void loop(). Void loop() berisi pembaruan operasi yang ada pada aplikasi Blynk, pengecekan konektivitas WiFi, pembacaan tombol fisik, logika buka-tutup *cover*, logika mulai-berhenti penyinaran dan logika *cover* otomatis.

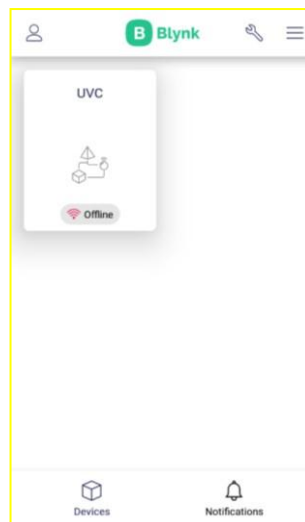
## 2. Blynk

Blynk pada penelitian ini akan digunakan sebagai *network server*, *database*, dan antamuka untuk menyimpan sekaligus menampilkan data dari *end-device* untuk mengedalikan *prototype*.



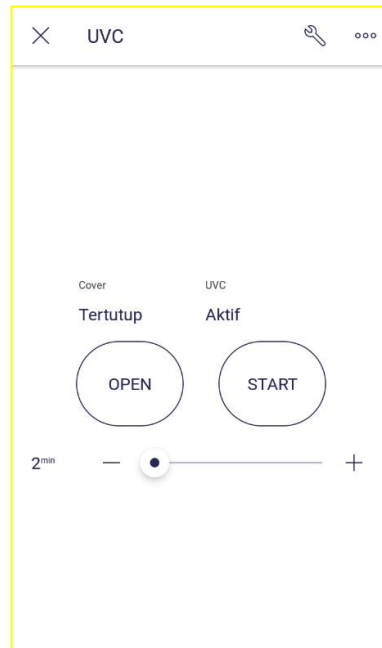
**Gambar 3.21 Tampilan Blynk**

Tampilan awal Blynk berisi menu untuk *Sign Up* atau *Log In*. Jika sudah memiliki akun maka hanya perlu melakukan *Log In* saja. Jika belum memiliki akun maka perlu melakukan *Sign Up*.



**Gambar 3.22 Tampilan proyek**

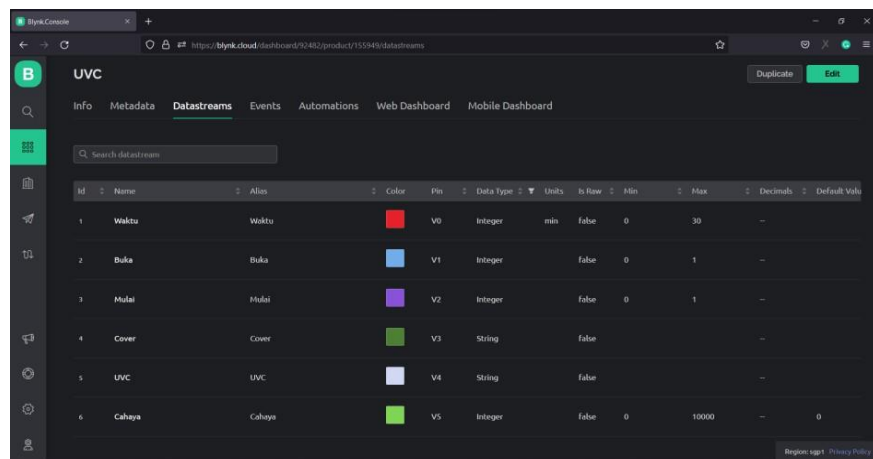
Proyek yang ditampilkan adalah proyek yang sudah dibuat pada aplikasi Blynk. Pada prototipe ini, penulis menamakan proyek sebagai UVC.



**Gambar 3.23 Aplikasi**

Tampilan utama pada aplikasi Blynk seperti tertera pada gambar berisi tombol membuka-tutup, tombol mulai-berhenti, dan *slidebar* untuk mengatur lamanya waktu penyinaran

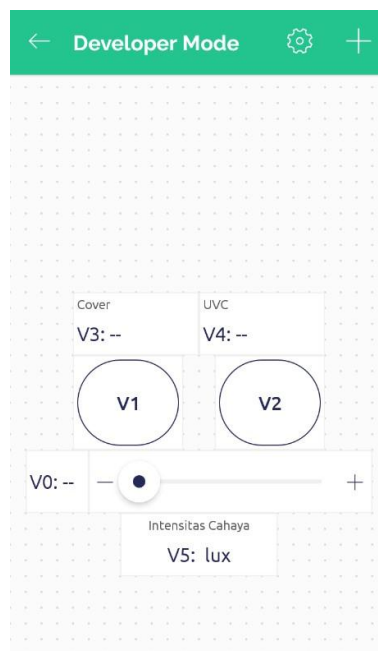
Perancangan perangkat lunak (*software*) berupa aplikasi *Blynk* yang dapat diunduh di *Play Store* maupun *Apps Store*. Pengguna dapat melakukan unduh selanjut *log in* dengan akun yang sudah dibuat pada *Blynk* versi *web* yaitu di alamat *blynk.cloud*. Sebelum melakukan konfigurasi pada *Blynk* versi *mobile*, pengguna perlu melakukan penambahan *datastream* pada *web* seperti gambar berikut.



**Gambar 3.24 Tampilan Datastream pada Blynk Web**

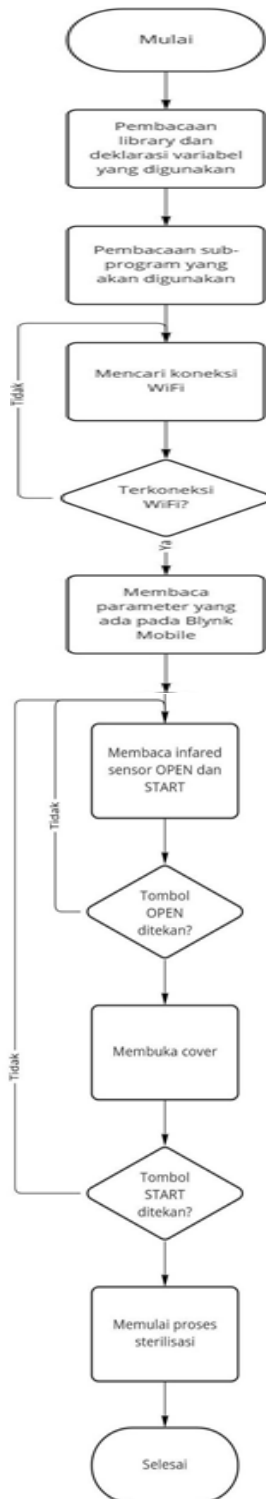


Setelah melakukan konfigurasi pada web, pengguna dapat melakukan konfigurasi pada Blynk mobile dengan menambah widget pada developer mode.



**Gambar 3.25 Konfigurasi Tampilan dan Data pada *Blynk Mobile***

Adapun algoritma yang beroperasi pada prototipe sesuai dengan gambar 3.25. Program dibuat menggunakan *software* Arduino IDE yang kemudian diterjemahkan langsung oleh *compiler* dari Arduino IDE untuk dioperasikan oleh Wemos D1 *Mini*. Program sesuai dengan gambar 3.10 sampai gambar 3.20.



**Gambar 3.26 Algoritma Program**

Sesaat setelah prototipe diaktifkan, Wemos D1 Mini akan membaca library dan deklarasi variabel. Kemudian sub-program akan dibaca sebelum mencari koneksi internet melalui WiFi. Jika belum terkoneksi dengan internet, maka prototipe akan mencari koneksi internet. Jika prototipe sudah terkoneksi dengan internet, maka prototipe akan membaca parameter yang ada pada Blynk mobile seperti durasi penyinaran.

Kemudian prototipe akan membaca tombol "OPEN" baik pada tombol infrared sensor maupun pada Blynk mobile. Lalu tombol "START" juga dibaca oleh mikropendali, jika ditekan maka proses sterilisasi akan dimulai dengan durasi sesuai yang diatur pada Blynk mobile hingga selesai..

### 3.3 PENGUJIAN *HARDWARE* DAN *SOFTWARE*

Setelah melakukan perancangan seluruh perangkat keras dan perangkat lunak, maka tahap selanjutnya yaitu melakukan pengujian terhadap *hardware* dan *software* dengan tujuan apakah setiap perangkat dan sistem sudah sesuai dengan standar yang ditetapkan.

#### 3.3.1 AKURASI WAKTU PENYINARAN

Pengujian yang pertama yaitu melakukan pengujian akurasi waktu penyinaran oleh *LED Ultraviolet-C* untuk proses sterilisasi benda yang akan di sterilkan. Pengujian ini sendiri dimulai dari perintah membuka-tutup dari perangkat sistem, kemudian perintah di proses oleh *Wemos D1 Mini* dan diteruskan ke *relay* untuk menyalakan komponen *LED Ultraviolet-C*. Blok diagram pengujian dapat dilihat pada gambar dibawah ini.



**Gambar 3.27 Blok Diagram Akurasi Waktu Penyinaran**

$$Akurasi = 100 - \left( \frac{|Waktu Penyinaran - Waktu Stopwatch|}{Waktu Penyinaran} \times 100\% \right) \dots\dots\dots (3)$$

1. Alat yang digunakan :
  - a. *Wemos D1 Mini*
  - b. *LED UVC (Ultraviolet-C)*
  - c. *Relay*
  - d. *Stopwatch*
  - e. Aplikasi yang sudah dibuat dari Blynk
2. Prosedur pengujian :
  - a. Jalankan aplikasi
  - b. Atur pewaktuan sterilisasi menjadi 5 detik
  - c. Tekan tombol *start* pada aplikasi
  - d. Lakukan penghitungan waktu penyinaran hingga selesai menggunakan *stopwatch*
  - e. Catat waktu yang tertera pada *stopwatch*

- f. Ulangi langkah c hingga e sebanyak 15 kali
- g. Ulangi langkah c hingga f dengan mengubah pewaktuan sterilisasi menjadi 15 detik dan 30 detik

### 3.3.2 AKURASI PENUTUP PINTU

Pengujian akurasi penutup pintu dilakukan untuk mengetahui akurasi pembacaan *infrared sensor* dan respon dari Wemos D1 Mini. Pengujian dilakukan dengan mendekatkan tangan dengan *infrared sensor* dan melihat respon dari Wemos D1 Mini *pada servo*. Jika *servo* bergerak maka percobaan dianggap berhasil, jika *servo* tidak bergerak percobaan dianggap gagal. Nilai diambil sebanyak 30 kali untuk dihitung rata-rata.



**Gambar 3.28 Blok Diagram Akurasi Penutup Pintu**

$$Akurasi = 100 - \left( \frac{Percobaan\ Gagal}{Banyaknya\ Percobaan} \times 100\% \right) \dots\dots\dots(4)$$

1. Alat yang digunakan :
  - a. *Infrared Sensor*
  - b. Wemos D1 Mini
  - c. *Servo*
2. Prosedur pengujian :
  - a. Dekatkan tangan ke *infrared sensor*
  - b. Jika *servo* bergerak maka percobaan dianggap berhasil
  - c. Ulangi langkah a dan b sebanyak 30 kali

### 3.3.3 AKURASI PEWAKTUAN OTOMATIS PINTU

Pengujian akurasi pewaktuan otomatis pintu dilakukan untuk mengetahui ketepatan waktu dalam menutup pintu otomatis ketika alat sterilisasi tidak digunakan. Pengujian dilakukan dengan membuka pintu dengan tombol “*OPEN*” dan menunggu pintu tertutup dengan sendirinya. Nilai diambil sebanyak 30 kali untuk dihitung rata-rata.



**Gambar 3.29 Blok Diagram Akurasi Pewaktuan Otomatis Pintu**

$$Akurasi = 100 - \left( \frac{|Waktu Tunggu - Waktu Terprogram|}{Waktu Terprogram} \times 100\% \right) \dots \dots \dots (5)$$

1. Alat yang digunakan :
  - a. *Infrared sensor*
  - b. *Wemos D1 Mini*
  - c. *Servo*
  - d. *Stopwatch*
2. Prosedur pengujian :
  - a. Dekatkan tangan ke tombol “*OPEN*” (*infrared sensor*) sehingga *cover* menjadi terbuka
  - b. Aktifkan *stopwatch* setelah *cover* terbuka sempurna lalu tunggu selama  $\pm 30$  detik
  - c. Hentikan *stopwatch* saat *cover* sudah mulai menutup dengan sendirinya.
  - d. Catat waktu yang tertera pada *stopwatch*
  - e. Ulangi langkah a hingga d sebanyak 30 kali

### 3.3.4 3AKURASI INTENSITAS CAHAYA

Pengujian akurasi intensitas cahaya dilakukan untuk mengetahui akurasi pembacaan sensor BH1750. Pengujian dilakukan dengan menyalakan LED UVC dan meletakkan sensor BH1750 dan Luxmeter pada satu titik yang berdekatan. Nilai diambil sebanyak 30 kali untuk dihitung rata-rata.



**Gambar 3.30 Blok Diagram Akurasi Intensitas Cahaya**

$$Akurasi = 100 - \left( \frac{|Pembacaan BH1750 - Pembacaan Luxmeter|}{Waktu Terprogram} \times 100\% \right) \dots \dots \dots (6)$$

1. Alat yang digunakan :
  - a. BH1750
  - b. NodeMCU ESP8266
  - c. Luxmeter
2. Prosedur pengujian :
  - a. Nyalakan LED UVC
  - b. Letakkan luxmeter dan BH1750 bersebelahan
  - c. Catat nilai pembacaan luxmeter dan BH1750 pada *Serial Monitor*.
  - d. Ulangi langkah a hingga c sebanyak 30 kali

### 3.3.5 QUALITY OF SERVICE (QOS)

Pengujian nilai *Quality of Service* (QOS) dilakukan untuk mengetahui kualitas transmisi dari *board* Wemos D1 Mini. Data *dummy* dari *board* Wemos D1 Mini dikirim ke *platform* Blynk setiap interval waktu 1 menit sekali selama 30 menit. Data QOS diambil untuk dihitung tiap pengujian, selanjutnya diambil nilai rata-rata *throughput*, *packet loss* dan *delay* secara keseluruhan.



**Gambar 3.31 Pengujian QOS (*Quality of Service*)**

1. Alat yang digunakan :
  - a. Wemos D1 Mini.
  - b. Kabel *micro* USB
  - c. 1 unit laptop.
  - d. *Software* Wireshark
  - e. Koneksi internet
2. Prosedur pengujian :
  - a. Menyiapkan koneksi internet
  - b. Menjalankan program Arduino IDE berisi data *dummy*

- c. Menjalankan *software Wireshark* dan melakukan *capture* pada data selama 1 menit
- d. Menghentikan *capture* lalu menambah jarak antara laptop dengan Wemos D1 *Mini* sejauh 1 meter
- e. Mengulangi langkah c dan d sebanyak 30 kali