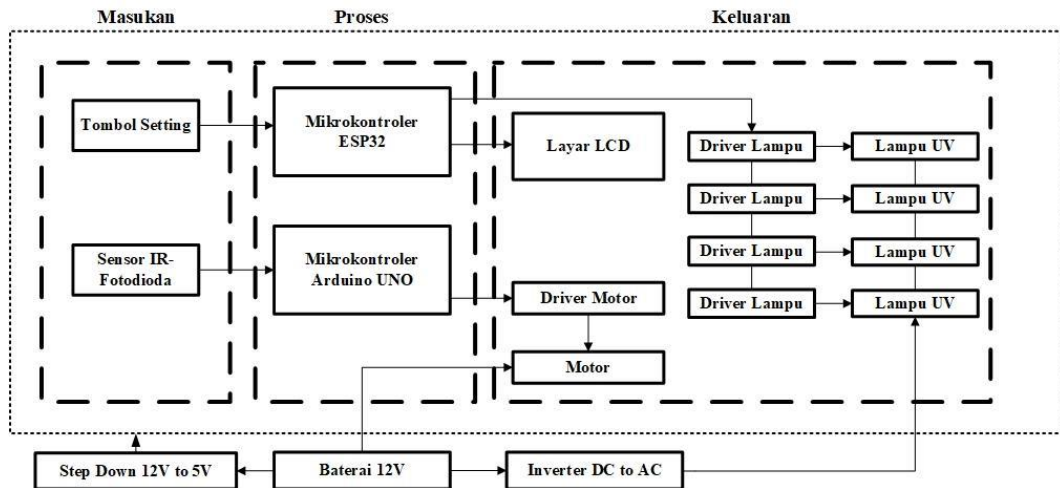


BAB III METODOLOGI PENELITIAN

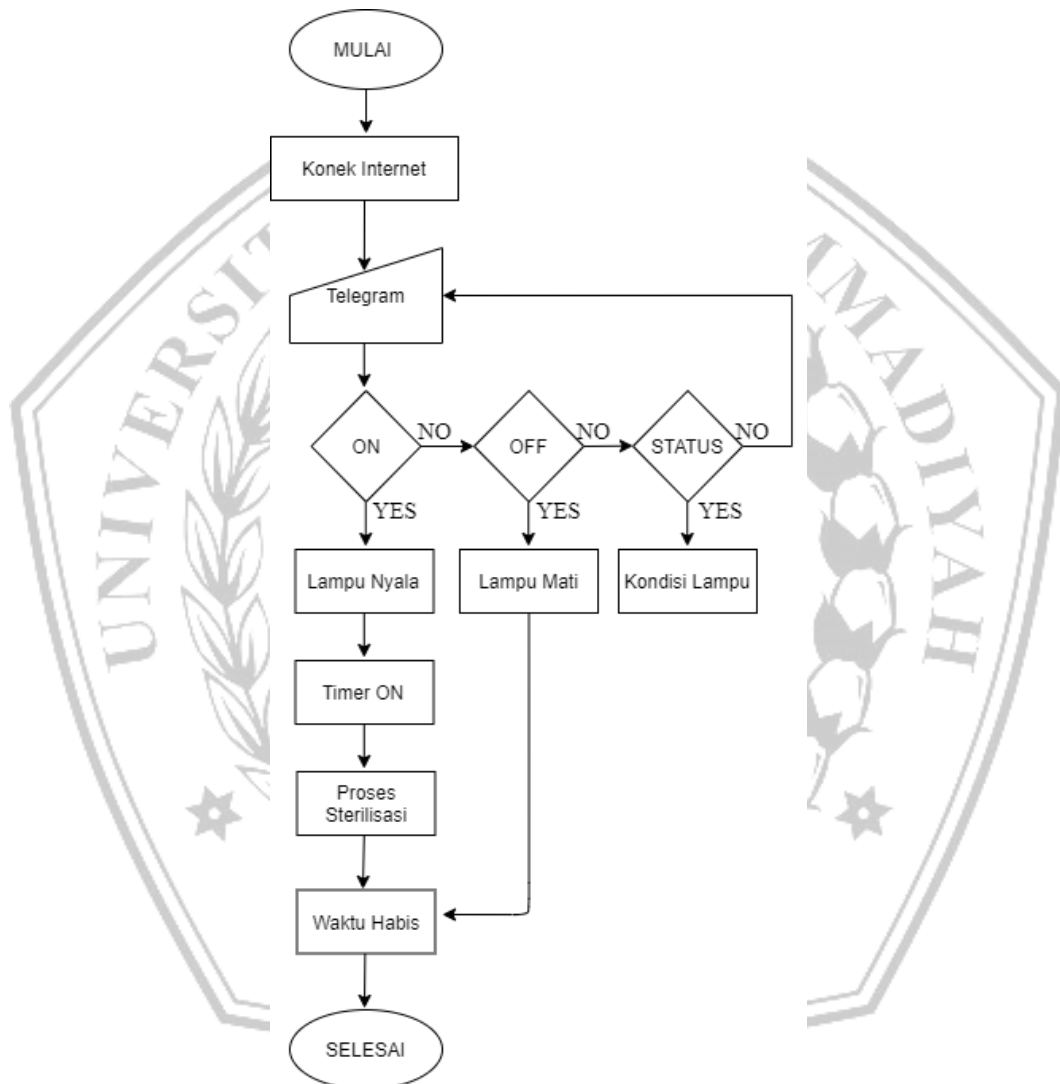
Pada BAB III ini akan di jelaskan tentang metode penelitian yang di lakukan pada penelitian kali ini. Adapun blok diagram untuk metode penelitian ini sebagai berikut:



Dari Gambar 1 Rangkaian dibagi menjadi 3 bagian yaitu masukan, proses, dan keluaran. Pada bagian masukan terdapat tombol setting yang berfungsi untuk memasukan nilai kecepatan dalam satuan *Revolution per Minute* (RPM) dan waktu lamanya sterilisasi. Sedangkan sensor IR-Fotodioda berfungsi untuk mendeteksi garis yang akan menghasilkan sinyal yang akan di olah pada mikrokontroler yang dapat mengatur driver motor *Direct Current* (DC). Pada bagian sistem proses terdapat mikrokontroler sebagai pusat proses yang berfungsi untuk mengontrol bagian masukan dan keluaran. Pada bagian sistem keluaran terdapat layar, driver motor, motor DC, driver lampu, lampu ultraviolet, dan modul ESP-32. Layar disini menggunakan *Liquid Crystal Display* (LCD) karakter 16x2 yang berfungsi untuk menampilkan nilai luas ruangan yang akan disterilkan dan waktu sterilisasi. Untuk dapat mengontrol lampu ultraviolet dari mikrokontroler diperlukan driver lampu agar dapat menyalakan lampu ultraviolet. Untuk menyalakan lampu ultraviolet menggunakan sumber daya *Alternating Current* (AC). Sedangkan modul ESP-32

berfungsi sebagai kontrol *on/off* alat secara nirkabel menghubungkan rangkaian inti dengan aplikasi telegram yang terdapat pada *smartphone*.

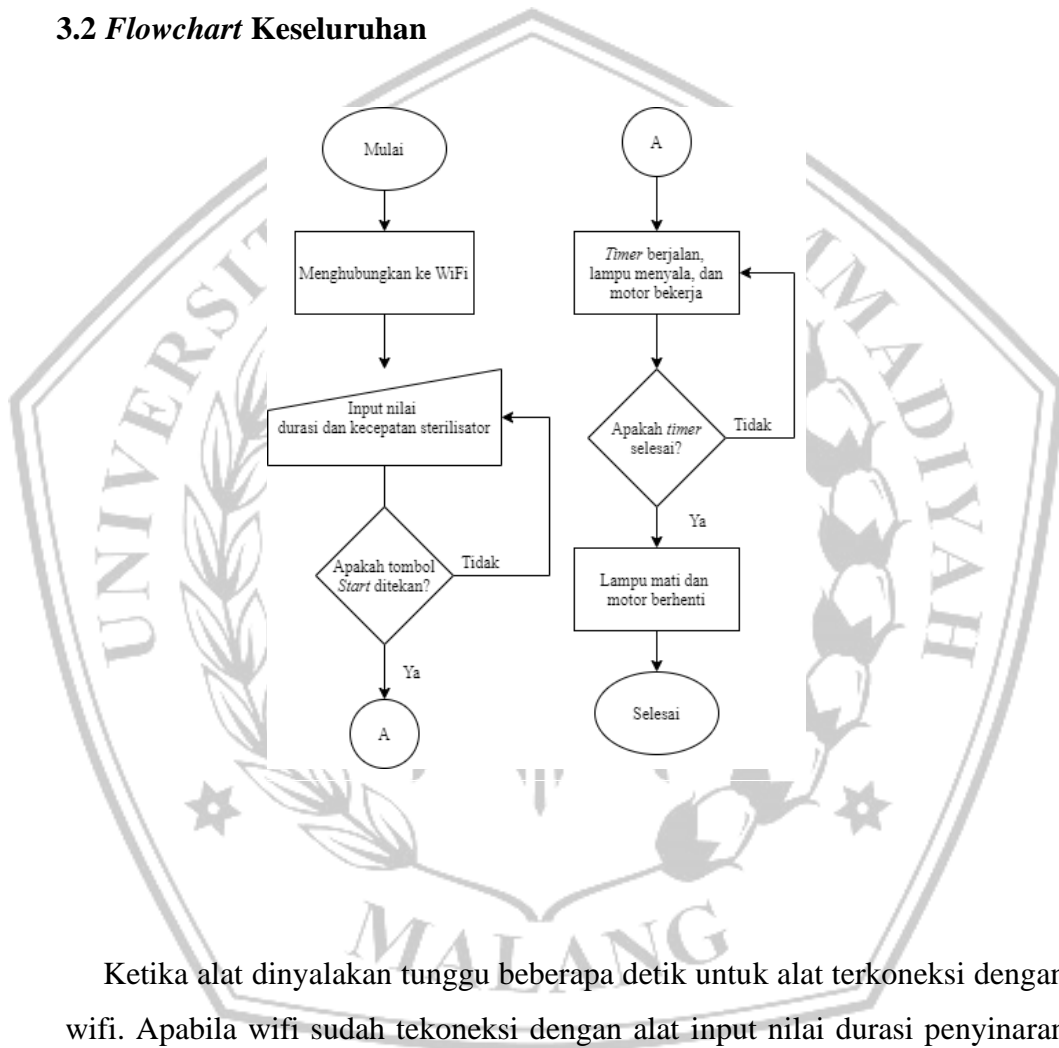
3.1 Flowchart Telegram



Dari *flowchart* yang terdapat pada Gambar 3.2, Ketika alat akan dinyalakan tunggu beberapa detik sempat terkoneksi ke internet, Apabila sudah terkoneksi dengan internet buka aplikasi telegram. Setelah telegram terbuka maka akan tertampil 3 mode yaitu *ON*, *OFF*, dan Status. Jika mode *ON* dipilih maka lampu

akan menyala bersamaan dengan timer akan mencacah mundur. Alat akan memulai sterilisasi dengan mengelilingi ruangan hingga waktu habis dan proses sterilisasi selesai. Jika mode *OFF* dipilih lampu langsung mati tidak menunggu waktu habis dan proses selesai. Sedangkan mode status berfungsi untuk memonitoring lampu UV.

3.2 Flowchart Keseluruhan



Ketika alat dinyalakan tunggu beberapa detik untuk alat terkoneksi dengan wifi. Apabila wifi sudah terkoneksi dengan alat input nilai durasi penyinaran dan kecepatan pergerakan alat. Jika tombol start ditekan pada smartphone maka lampu UV menyala, motor bekerja, dan waktu mulai mencacah mundur. Ketika waktu habis maka waktu berhenti mencacah, lampu UV mati dan motor berhenti. Apabila waktu belum habis maka lampu UV masih menyala dan motor masih bekerja.

3.3 Diagram Mekanis



Pada tahap pembuatan alat sterilisasi perlu dilakukan tahap desain atau perancangan alat. Langkah merancang alat ini dilakukan untuk melakukan perkiraan terkait bentuk maupun susunan komponen pada alat yang akan dibuat. Adapun bentuk alat yang akan dibuat, dirancang akan sesuai dengan Gambar 3.5

Alat sterilisasi ultraviolet (UV) pada ruangan mempunyai 4 buah lampu dengan tipe UV-C disertai dengan dua buah roda bebas di bagian depan dan dua buah motor DC di bagian belakang. Pada bagian depan alat terdapat LCD karakter 16x2 untuk menampilkan luas ruangan yang akan di sterilkan dan lama waktu sterilisasi dan tombol setting berfungsi untuk mengatur kecepatan pergerakan alat dan lama waktu sterilisasi.

3.4 Alat dan Bahan

Pada Tabel 3.1 menunjukkan alat yang digunakan untuk melakukan pembuatan alat sterilisator UV

Tabel 3.1 Alat

No	Nama Alat	Jumlah
1	Laptop	1
2	<i>Tool Set</i>	1
3	Bor	1
4	Gerinda	1
5	Solder	1
6	Lakban hitam	1
7	<i>Stopwatch</i>	1

Tabel 3.2 menunjukkan bahan yang diperlukan untuk melakukan pembuatan sterilisator UV.

Tabel 3.2 Bahan

No	Nama Alat	Jumlah
1	Arduino Uno	1
2	ESP-32	1
3	Modul sensor IR	4
4	Motor DC	2
5	Driver motor	2
6	Lampu UV-C	4
7	Trafo Ballast	4
8	Relay DC 5 Volt	1
9	<i>Power Supply 5-12 Volt</i>	1

10	Baterai	1
11	LCD	1
12	PCB	Seperlunya
13	Timah	Seperlunya
14	<i>Jumper Male/Female</i>	Seperlunya

3.5 Rancangan Perangkat Keras

3.5.1 Rangkaian Minimum Sistem ESP32

Rangkaian minimum sistem berfungsi sebagai penghubung antara beberapa jenis masukan tombol dan sensor untuk mengolah program agar dapat mengendalikan aktuator dan menampilkan karakter pada LCD sebagai keluaran alat.

3.5.2 Rangkaian Tombol Setting dan LCD display

Tombol setting rangkaian diatas digunakan untuk mengatur kecepatan pergerakan alat dan mengatur waktu sterilisasi. Sedangkan LCD karakter 16x2 sebagai display yang menampilkan waktu sterilisasi dan kecepatan alat. Untuk dapat menyalakan LCD membutuhkan modul 12C sebagai penghubung antara LCD dengan ESP32.

3.5.3 Rangkaian Kontrol Lampu Ultraviolet

Diperlukan tegangan AC untuk menyalakan lampu ultraviolet. Untuk menghubungkan mikrokontroler dengan lampu UV maka diperlukan sebuah komponen relay yang berfungsi untuk memutus dan menghubungkan tegangan yang masuk ke lampu UV.

3.5.4 Rangkaian Penggerak Motor

Diperlukan sebuah modul driver BTS7960 agar dapat menggerakkan motor DC. Karena motor DC yang digunakan memerlukan input tegangan sebesar 12 Volt DC. Skematik modul BTS7960 Rangkaian Modul Sensor FC-51



Untuk mendeteksi garis, alat ini menggunakan modul sensor FC-51. Sensor ini memiliki 3 buah pin yaitu VCC, ground, dan sinyal keluaran yang berupa tegangan high atau low. Pada alat ini menggunakan 4 buah sensor FC-51 yang dipasang di tengah-tengah alat.

3.6 Rancangan Software

3.6.1 Program Pada ESP32

Berikut program yang digunakan pada sterilisator UV sehingga mampu terhubung dengan WiFi dan terkoneksi dengan Telegram.

```
#include "CTBot.h"
CTBot myBot;
#include <WiFi.h>
String ssid = "freewifi" ; // REPLACE mySSID WITH YOUR WIFI
SSID
String pass = "12345678"; // REPLACE myPassword YOUR WIFI
PASSWORD, IF ANY
//String token =
"5318534770:AAEmgdYUXJ6OMyhLzSUYroxnHO0hvHmKR5A" ; // REPLACE
myToken WITH YOUR TELEGRAM BOT TOKEN
String token = "5947921992:AAHn_q1Q7Ac4ssKvpkHwAc5-
dmkubaQ3z0Q" ; // REPLACE myToken WITH YOUR TELEGRAM BOT TOKEN
//Library analog ESP32
#include <Arduino.h>
#include <analogWrite.h>

//LCD Library
#include <LiquidCrystal_I2C_Auto.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
```

```

LiquidCrystal_I2C_Auto lcd(0x27, 16, 2);

//Setting nilai PWM
int nilai_PWM, pwm1, pwm2, level;

//Setting tombol dan timer
unsigned long waktu_berjalan;
unsigned long interval;
unsigned long waktu_awal;
const int UP = 3, DOWN = 14, UpPWM = 12, DownPWM = 13, reset = 9,
LEDtbl = 17, Mulai = 16;
const int indikator = 2, DR_UV = 23; //indikator UV, Lampu UV
int ledStatus = 0;
bool UV = LOW;
bool p_up, p_don, i_up, i_don, mula;
String ledStts_str, pwm1_str, pwm2_str, intrvl_str,skled="0";
String dtkrsebelum,dtkrsekarang;
unsigned long pm_lcd,pm_bacaTele, pm_ledpwm, pm_tombol,pm_kirim;

TBMMessage msg;

void set_lcd() {
  // initialize the LCD
  lcd.init();

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("UV STERILISATOR");
  lcd.setCursor(0, 1);
  lcd.print(" LINE FOLLOWER");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Connecting Wifi");
}

void tampil_lcd() {
  lcd.setCursor(0, 0);
  lcd.print("Timer:");
  lcd.setCursor(0, 1);
  lcd.print(interval / 60000);
  lcd.setCursor(3, 1);
  lcd.print("Menit");

  lcd.setCursor(9, 0);
  lcd.print("PWM:");  lcd.setCursor(9, 1);
  lcd.print("evel");
  lcd.setCursor(15, 1);
  lcd.print(level);
}

void connectwifi() {
  myBot.wifiConnect(ssid, pass);

  // set the telegram bot token
  myBot.setTelegramToken(token);
}

```



```

while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  digitalWrite(19, LOW);
  delay(500);
  digitalWrite(19, HIGH);
  delay(500);
}
lcd.setCursor(0, 1);
lcd.print("Connected");
digitalWrite(19, LOW);
Serial.println("");
// check if all things are ok
if (myBot.testConnection())
  Serial.println("\ntestConnection OK");
else
  Serial.println("\ntestConnection NOK");
delay(2000);
lcd.clear();
}
void setTombol() {
  pinMode(UP, INPUT_PULLUP);
  pinMode(DOWN, INPUT_PULLUP);
  pinMode(UpPWM, INPUT_PULLUP);
  pinMode(DownPWM, INPUT_PULLUP);
  pinMode(reset, INPUT_PULLUP);
  pinMode(Mulai, INPUT_PULLUP);
  pinMode(LEDtbl, OUTPUT);
  pinMode(indikator, OUTPUT);
  pinMode(DR_UV, OUTPUT);
  pinMode(19, OUTPUT); //indikator WIFI terkoneksi
  digitalWrite(DR_UV, HIGH);
}
void bacaTelegram() {
  if (millis() - pm_bacaTele > 500) {
    if (CTBotMessageText == myBot.getNewMessage(msg)) {
      String text = msg.text;
      String ps_blk = "";
      String from_name = msg.sender.username;
      if (from_name == "") from_name = "Guest";
      if (text == "/UVon") {
        ledStatus = 1;
        digitalWrite(indikator, HIGH); // turn the LED on (HIGH
is the voltagelevel)
        digitalWrite(DR_UV, LOW); // turn the UV on (HIGH is the
voltage level)
        UV = !UV;
        waktu_awal = millis();
        ledStatus = 1;
        //      delay(20);
        ps_blk = "UV aktif";
        //      bot.sendMessage(chat_id, "UV aktif", "");
        Serial.println("\nUser menyalakan Lampu UV\n");
      }

      if (text == "/UVoff") {
        ledStatus = 0;
        digitalWrite(indikator, LOW); // turn the LED off (LOW
is the voltagelevel)

```

```

        digitalWrite(DR_UV, HIGH); // turn the UV off (LOW is
the oltage level
        UV = LOW;
        ps_blk = "UV mati";
        //          bot.sendMessage(chat_id, "UV mati", "");
        Serial.println("\nUser mematikan Lampu UV\n");
    }

    if (text == "/status") {
        if (ledStatus) {
            ps_blk = "UV aktif";
            //          bot.sendMessage(chat_id, "UV aktif", "");
        } else {
            ps_blk = "UV mati";
            //          bot.sendMessage(chat_id, "UV mati", "");
        }
    }
    if (text == "/system_on") {
        skled="1";
        String welcome = "Sistem Dimulai";
        ps_blk = welcome;
        //          bot.sendMessage(chat_id, welcome, "Markdown");
    }
    if (text == "/system_off") {
        skled="0";
        String welcome = "Sistem berhenti";
        ps_blk = welcome;
        //          bot.sendMessage(chat_id, welcome, "Markdown");
    }
    if (text == "/start") {
        String welcome = "Sterilisator Ruangan dengan Ultraviolet
Berbasis LineFollower, " + msg.sender.firstName + ".\n";
        welcome += "Teknik Elektro - UMM\n\n";
        welcome += "/UVon : to switch the UV ON\n";
        welcome += "/UVoff : to switch the UV OFF\n";
        welcome += "/status : Cek kondisi UV\n";
        welcome += "/system_on : Memulai sistem\n";
        welcome += "/system_off : Menonaktifkan sistem\n";
        ps_blk = welcome;
        //          bot.sendMessage(chat_id, welcome, "Markdown");
    }
    myBot.sendMessage(msg.sender.id, ps_blk);
}
// ...forward it to the sender
pm_bacaTele = millis();
}
}

void setPwm() {
    //Set PWM
    if (level == 1) {
        pwm1 = 200;
        pwm2 = 170;
    } else if (level == 2) {
        pwm1 = 235;
        pwm2 = 190;
    } else if (level == 3) {
        pwm1 = 255;
    }
}

```

```

    pwm2 = 220;
}

// Set level PWM
if (level <= 1) {
    level = 1;
} else if (level >= 3) {
    level = 3;
}

// Set Timer
if (interval <= 0) {
    interval = 0;
} else if (interval >= 7206000) {
    interval = 0;
}

}

void UVL() {
    if (UV == HIGH) {
        if (millis() - waktu_awal >= interval) {
            UV = !UV;
            ledStatus = 1;
        }
    }

    if (UV == LOW) {
        digitalWrite(indikator, LOW);
        digitalWrite(DR_UV, HIGH);
        ledStatus = 0;
    }

    if (UV == HIGH) {
        digitalWrite(indikator, HIGH);
        digitalWrite(DR_UV, LOW);
        ledStatus = 1;
    }
}

void kirim() {
    if (millis() - pm_kirim > 1000) {
        dtkrsekarang=skled+"_"+String(level);
        if(dtkrsekarang != dtkrsebelum){
            String datakirim="-"+dtkrsekarang+"#";
            Serial.println(datakirim);
            dtkrsebelum=dtkrsekarang;
        }

        pm_kirim = millis();
    }
}

void nyala_ledtombol() {
    if (millis() - pm_ledpwm < 200) {
        digitalWrite(LEDtbl, HIGH);
    } else {
        digitalWrite(LEDtbl, LOW);
    }
}
}

```

```

void baca_tombol() {
  if (millis() - pm_tombol > 100) {
    bool pu = digitalRead(UpPWM);
    if (pu == false && p_up == false) p_up = true;
    if (pu && p_up) {
      p_up = false;
      level += 1;
      pm_ledpwm = millis();
    }
    bool pd = digitalRead(DownPWM);
    if (pd == false && p_don == false) p_don = true;
    if (pd && p_don) {
      p_don = false;
      level -= 1;
      pm_ledpwm = millis();
    }
    bool iu = digitalRead(UP);
    if (iu && i_up == false) i_up = true;
    if (iu == false && i_up) {
      i_up = false;
      interval += 60000;
      pm_ledpwm = millis();
    }
    bool id = digitalRead(DOWN);
    if (id == false && i_don == false) i_don = true;
    if (id && i_don) {
      i_don = false;
      interval -= 60000;
      pm_ledpwm = millis();
    }
    bool ml = digitalRead(Mulai);
    if (ml == false && mula == false){
      mula = true;
    }
    if (ml && mula) {
      mula=false;
      UV = !UV;
      waktu_awal = millis();
    }
    pm_tombol=millis();
  }
}

void setup() {
  // initialize the Serial
  Serial.begin(115200);
  Serial.println("Starting TelegramBot...");
  set_lcd();
  setTombol();
  connectwifi();
}

void loop() {
  // a variable to store telegram message data
  // TBMessage msg;
  setPwm();
  UVL();
}

```

```

    kirim();
    tampil_lcd();
    baca_tombol();
    nyala_ledtombol();
    bacaTelegram();
    if(millis()-pm_lcd >5000){
lcd.init();
    pm_lcd=millis();
    }
}
}

```

3.6.2 Program Pada Arduino UNO

Berikut program yang digunakan pada sterilisator UV sehingga mampu berjalan dengan membaca sensor infrared dan fotodiode.

```

int R_PWM1 = 11;
int L_PWM1 = 10;
int R_PWM2 = 6;
int L_PWM2 = 9;
const int SensorProximty1 = 2;
const int SensorProximty2 = 3;
const int SensorProximty3 = 4;
const int SensorProximty4 = 5;
int data2=200,data3=150;
#define pintombol A0
int td,level;
// the setup routine runs once when you press reset:
//=====
//===== M E T H O D
//=====
//=====

void setSensor(){
    pinMode(SensorProximty1, INPUT_PULLUP);
    pinMode(SensorProximty2, INPUT_PULLUP);
    pinMode(SensorProximty3, INPUT_PULLUP);
    pinMode(SensorProximty4, INPUT_PULLUP);
}

void read_sensor() {

    bool knluar = digitalRead(SensorProximty1);
    bool kndalam = digitalRead(SensorProximty2);
    bool krdalam = digitalRead(SensorProximty3);
    bool krluar = digitalRead(SensorProximty4);
    Serial.print(knluar);
    Serial.print("_");
    Serial.print(kndalam);
    Serial.print("_");
    Serial.print(krdalam);
    Serial.print("_");
    Serial.println(krluar);
    cekhasilsensor(knluar,kndalam,krdalam,krluar);
}

```

```

}

void cekhasilsensor(bool knl,bool knd,bool krd,bool krl){
  if(knl && !knd && !krd && !krl){
    kekanan();
  }else if(knl && knd && !krd && !krl){
    kekanansedikit();
  }else if(!knl && knd && !krd && !krl){
    maju();
  }else if(!knl && knd && krd && !krl){
    maju();
  }else if(!knl && !knd && krd && !krl){
    maju();
  }else if(!knl && !knd && krd && krl){
    kekirisedikit();
  }else if(!knl && !knd && !krd && krl){
    kekiri();
  }
}

}

void kekanan(){
  Serial.println("kekanan");
  gerak(data2,LOW,LOW,LOW);
}
void kekanansedikit(){
  Serial.println("kekanansedikit");
  gerak(data2,LOW,data2-40,LOW);
}
}
void maju(){
  Serial.println("Maju");
  gerak(data2,LOW,data2,LOW);
}
}
void kekirisedikit(){
  Serial.println("kekiriSedikit");
  gerak(data2-40,LOW,data2,LOW);
}
}
void kekiri(){
  Serial.println("kekiri");
  gerak(LOW,LOW,data2,LOW);
}
}
void berhenti(){
  Serial.println("BERhenti");
  gerak(LOW,LOW,LOW,LOW);
}
}

void gerak(int pr1,int pl1,int pr2,int pl2){
  analogWrite(R_PWM1, pr1);
  analogWrite(L_PWM1, pl1);
  analogWrite(R_PWM2, pr2);
  analogWrite(L_PWM2, pl2);
}

}

void cekLevel(){
  if(level==1){
    data2=210;
  }else if(level==2){

```

```

    data2=232;
  }else if(level==3){
    data2=255;
  }else{
    data2=210;
  }
}

String seplit(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length() - 1;
  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0], strIndex[1])
: "";
}

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(115200);
  setSensor();
}

// the loop routine runs over and over again forever:
void loop() {
  if(Serial.available()){
    if(Serial.find("=")){
      String dtser=Serial.readStringUntil('#');
      td=seplit(dtser, '_', 0).toInt();
      level=seplit(dtser, '_', 1).toInt();
      cekLevel();
      // data2=td;
      // maju();
      // cekgerak(td);
    }
  }
  if(td==1){
    read_sensor();
  }else{
    berhenti();
  }
  delay(100);
}

```

3.7 Metode Pengujian Alat

Pengujian alat ini bertujuan untuk mengetahui batas kemampuan alat sesuai dengan fungsinya. Dengan kata lain, metode pengujian alat berarti kegiatan penelitian, pencarian, penjelasan, juga pembuatan instrumen yang akan digunakan

oleh petugas. Adapun variabel yang menjadi objek penelitian dan pengamatan pada alat sterilisasi UV dengan remot kontrol adalah lama waktu sterilisasi.

3.7.1 Pengujian *timer*

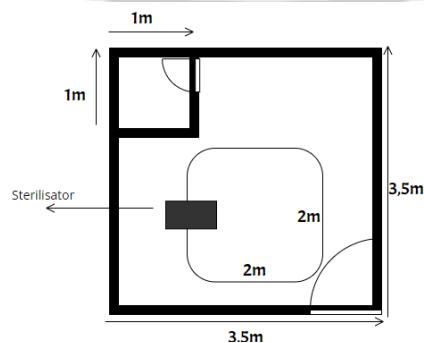
Pengujian *timer* dengan cara mengaturnya pada tombol *setting* yang terdapat pada sisi bagian depan alat. *Setting timer* pada alat menggunakan kelipatan satu menit. Pengujian ini dilakukan dengan membandingkan perbedaan waktu antara *stopwatch* dengan *timer* pada alat.

3.7.2 Pengujian kendali *on-off* lampu UV

Pengujian kendali on-off lampu UV bertujuan untuk mengetahui kemampuan koneksi antara aplikasi telegram di *smartphone* dengan *mobile sterilisator UV*. Indikator keberhasilan yaitu apabila diberikan perintah aktif maka booth akan memberikan *feedback* berupa notifikasi bahwasanya alat dapat berjalan sesuai perintah berupa lampu UV akan menyala. Sedangkan untuk indikator kegagalan pengujian dipengaruhi oleh koneksi internet yang kurang stabil. Indikasi kegagalan pengujian berupa notifikasi berulang yang di kirimkan oleh booth ke telegram. Pengujian ini dilakukan sebanyak 10 kali.

3.7.3 Pengujian Akurasi Sensor

Pengujian akurasi sensor bertujuan untuk mengetahui kemampuan sensor dalam membaca jalurnya. Indikator keberhasilannya yaitu apabila sensor membaca garis jalur yang diberikan dengan sesuai *setting* yang yang diberikan. Sedangkan indikator kegagalannya yaitu sensor tidak membaca sesuai jalur sehingga sterilisator keluar pada garisnya. Pengujian ini dilakukan sebanyak 10 kali pada setiap *setting levelnya* dengan ukuran lintasan 2x2m seperti pada gambar 3.6



Gambar 3.6 Denah *Line Follower*