# LAPORAN AKHIR TUGAS BESAR PERANCANGAN SISTEM TERTANAM



**SISTEM *GROW TENT* UNTUK PERAWATAN TANAMAN HIAS *INDOOR***

***Diajukan untuk memenuhi kelulusan mata kuliah Perancangan Sistem Tertanam di Prodi S1 Teknik Elektro.***

**Oleh:**

**Rangga Vallian Sayoga/1102204490**

**Hanif A Rahadiansyah/1102202584**

**Putra Wisnu Adjie/** **1102204517**

**Muhammad Fauzan Adzhima/1102204423**

**Muhammad Gibran Al Kafi Katami Arief/** **1102200432**

**PRODI S1 TEKNIK ELEKTRO**

**FAKULTAS TEKNIK ELEKTRO**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2023**

# DAFTAR ISI

[LAPORAN AKHIR TUGAS BESAR PERANCANGAN SISTEM TERTANAM 1](#_Toc154194761)

[DAFTAR ISI 2](#_Toc154194762)

[DAFTAR GAMBAR 3](#_Toc154194763)

[BAB 1 PENDAHULUAN 4](#_Toc154194764)

[1.1 Latar Belakang 5](#_Toc154194765)

[1.2 Rumusan Masalah 6](#_Toc154194766)

[1.2.1 Bagaimana model *embedded system* yang dirancang untuk melakukan perawatan tanaman hias secara otomatis? 6](#_Toc154194767)

[1.2.2 Sensor apa saja yang diperlukan untuk mewujudkan *embedded system* yang dapat melakukan perawatan tanaman hias secara otomatis? 6](#_Toc154194768)

[1.3 Tujuan Penelitian 6](#_Toc154194769)

[1.3.1 Merancang sebuah model *embedded system* yang dapat melakukan perawatan tanaman hias secara otomatis. 6](#_Toc154194770)

[1.3.2 Menentukan jenis sensor yang dapat mengambil data analog berupa intensitas cahaya, suhu, dan kelembaban secara *real time* dan akurat 6](#_Toc154194771)

[BAB 2 ALAT DAN BAHAN 6](#_Toc154194772)

[2.1.1 Arduino UNO 6](#_Toc154194773)

[2.1.2 BH1750 *Lux Sensor* 6](#_Toc154194774)

[2.1.3 DHT11 *Temperature & Humidity Sensor* 6](#_Toc154194775)

[2.1.4 DC 12 Volt *Exhaust Fan* 7](#_Toc154194776)

[2.1.5 LED Light Bar 220 V 7](#_Toc154194777)

[2.1.6 *Jumper* 8](#_Toc154194778)

[2.1.7 *Breadboard* 8](#_Toc154194779)

[2.1.8 *Relay Module* 9](#_Toc154194780)

[BAB 3 METODE PENELITIAN 10](#_Toc154194781)

[3.1 Langkah-langkah Metode penelitian *Trial and Error* 10](#_Toc154194782)

[3.1.1 Percobaan Awal 10](#_Toc154194783)

[3.1.2 Integrasi Sistem 15](#_Toc154194784)

[BAB 4 HASIL PENELITIAN 19](#_Toc154194785)

[BAB 5 EVALUASI 21](#_Toc154194786)

[BAB 6 KESIMPULAN 22](#_Toc154194787)

[REFERENSI 23](#_Toc154194788)

# DAFTAR GAMBAR

[**Gambar 2.1: Arduino UNO** 7](#_Toc154194804)

[**Gambar 2.2: BH1750 Lux Sensor** 7](#_Toc154194805)

[**Gambar 2.3: DHT11 Humidity & Temperature Sensor** 8](#_Toc154194806)

[**Gambar 2.4: DC Exhaust Fan** 8](#_Toc154194807)

[**Gambar 2.5: LED Light Bar** 9](#_Toc154194808)

[**Gambar 2.6: Jumper** 9](#_Toc154194809)

[**Gambar 2.7: Breadboard** 10](#_Toc154194810)

[**Gambar 2.8: AC to DC Relay Module** 10](#_Toc154194811)

[**Gambar 4.1: Hasil Pembacaan Lux Sensor dan DHT11 Humidity & Temperature Sensor** 20](#_Toc154194812)

[**Gambar 4.2: Kipas mati saat kelembaban mencapai lebih dari 80% RH** 21](#_Toc154194813)

[**Gambar 4.3: Lampu bar mati saat lux melebihi 60** 21](#_Toc154194814)

# PENDAHULUAN

## Latar Belakang

Merawat tanaman hias merupakan salah satu hobi yang telah ada sejak dahulu dan dilakukan oleh berbagai kalangan masyarakat. Hobi ini digemari karena beberapa alasan, diantaranya membantu seseorang untuk berkonsentrasi, membuat tubuh bergerak aktif, hingga menjernihkan pikiran. Tanaman hias yang dirawat pun cukup beragam, mulai dari yang dari yang tahan terhadap berbagai kondisi cuaca hingga yang hanya dapat hidup di kondisi lingkungan tertentu. Terlepas dari jenisnya, tanaman hias memiliki kesamaan apabila ditinjau dari segi perawatannya, yakni memerlukan alokasi waktu perawatannya, mulai dari penyiraman, pemupukan, hingga pengaturan kondisi lingkungan.

Persoalan yang disinggung pada kalimat terakhir paragraf pertama tentunya bukan suatu masalah apabila pemilik tanaman hias memiliki banyak waktu sehingga dapat mengalokasikan waktu tersebut untuk melakukan perawatan pada tanaman hias. Namun, hal ini bertolak-belakang apabila pemilik tanaman hias memiliki banyak kesibukan sehingga tidak sempat untuk melakukan perawatan terhadap tanaman hias. Hal ini tentunya berisiko membuat kondisi tanaman hias menjadi tidak sehat, bahkan mati.

Oleh karena itu, penulis menyimpulkan bahwa diperlukan suatu solusi yang memungkinkan pemilik tanaman hias untuk merawat tanaman hias tanpa harus terlibat langsung dalam proses perawatan tersebut. Untuk memperoleh solusi ini, penulis menyarankan untuk membuat sebuah perangkat berbentuk *embedded system* yang dapat melakukan perawatan tanaman hias secara otomatis. Perawatan dilakukan dengan cara melakukan pengontrolan terhadap intensitas cahaya, pengairan, kelembaban, dan suhu pada lingungan tempat tanaman hias disimpan. Melalui perancangan sistem ini, penulis berharap solusi yang telah disinggung pada kalimat pertama paragraf ini dapat tercapai sehingga memudahkan pemilik tanaman hias untuk melakukan perawatan terhadap tanaman hias.

## Rumusan Masalah

### Bagaimana model *embedded system* yang dirancang untuk melakukan perawatan tanaman hias secara otomatis?

### Sensor apa saja yang diperlukan untuk mewujudkan *embedded system* yang dapat melakukan perawatan tanaman hias secara otomatis?

### 

## Tujuan Penelitian

### Merancang sebuah model *embedded system* yang dapat melakukan perawatan tanaman hias secara otomatis.

### Menentukan jenis sensor yang dapat mengambil data analog berupa intensitas cahaya, suhu, dan kelembaban secara *real time* dan akurat

# ALAT DAN BAHAN

### Arduino UNO

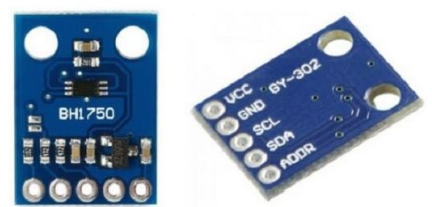
Arduino UNO pada penelitian ini berfungsi sebagai mikrokontroler yang memberikan perintah terhadap aktuator (*exhaust fan* dan *LED light bar*) berdasakan input atau masukan data yang diperoleh oleh sensor.



**Gambar 2.1: Arduino UNO**

### BH1750 *Lux Sensor*

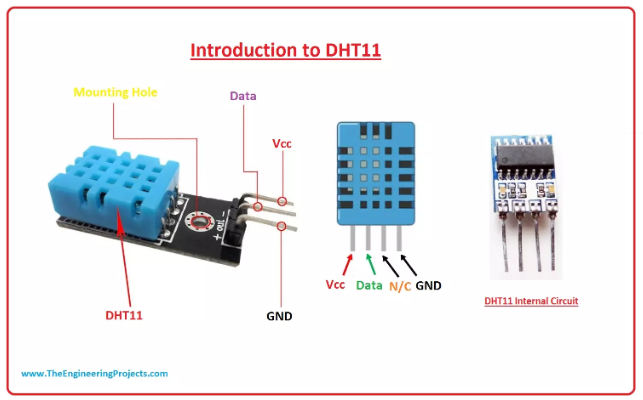
BH1750 lux sensor berfungsi untuk mendapatkan data berupa nilai lux yang akan masuk ke mikrokontroler yang kemudian menghasilkan perintah untuk mengaktifkan *LED light bar.*



**Gambar 2.2: BH1750 Lux Sensor**

### DHT11 *Temperature & Humidity Sensor*

DHT11 *temperature & humidity sensor* berfungsi untuk mendapatkan data berupa nilai lux yang akan masuk ke mikrokontroler yang kemudian menghasilkan perintah untuk mengaktifkan *exhaust fan.*



**Gambar 2.3: DHT11 Humidity & Temperature Sensor**

### DC 12 Volt *Exhaust Fan*

*Exhaust fan* berfungsi menarik udara kotor sehingga kondisi udara di dalam *grow box* lebih baik dan berkualitas sehingga kemungkinan tanaman untuk rusak dan mati pun berkurang.



**Gambar 2.4: DC Exhaust Fan**

### LED Light Bar 220 V

Light bar akan aktif ketika nilai lux yang didapatkan oleh *lux sensor* kurang dari *threshold yang* telah ditentukan. Fungsinya adalah untuk menjaga tanaman sehingga tetap mendapatkan pencahayaan ideal untuk tetap hidup dan sehat.



**Gambar 2.5: LED Light Bar**

### *Jumper*

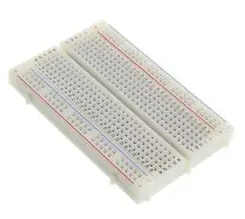
Fungsi jumper adalah untuk menghubungkan tiap komponen sesuai sistem perkabelan yang telah dirancang.



**Gambar 2.6: Jumper**

### *Breadboard*

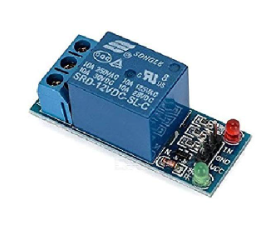
Breadboard difungsikan sebagai wadah untuk menghubungkan komponen dengan jumper yang kemudian dihubungkan ke komponen lainnya.



**Gambar 2.7: Breadboard**

### *Relay Module*

Tipe relay module yang kami gunakan adalah dengan yang memiliki kapasitas *switching* sebesar 10A dan didesain untuk *mounting* terhadap PC *board* sehingga memiliki kemampuan untuk mengonversi arus AC ke DC 5-12 Volt.



**Gambar 2.8: AC to DC Relay Module**

# METODE PENELITIAN

Metode penelitian yang digunakan pada tugas besar ini yaitu *trial and error* yang merupakan pendekatan di mana peneliti melakukan serangkaian percobaan atau upaya percobaan dengan tujuan menemukan solusi atau mencapai tujuan tertentu. Pendekatan ini melibatkan iterasi, di mana setiap percobaan memberikan umpan balik yang digunakan untuk menyesuaikan atau mengubah pendekatan selanjutnya. Metode *trial and error* melibatkan pengulangan percobaan atau tindakan tanpa mengandalkan perencanaan terperinci sebelumnya. Setiap percobaan memberikan kesempatan untuk belajar dari kesalahan atau mencapai pemahaman lebih baik.

## Langkah-langkah Metode penelitian *Trial and Error*

### Percobaan Awal

1. Melakukan *testing* terhadap aktuator yang ada pada rangkaian alat dengan *source code* khusus.

* *Source Code Fan & Relay*

|  |
| --- |
| #define RELAY\_PIN 9 // Pin yang terhubung ke relay  #define FAN\_PIN 11 // Pin yang terhubung ke kipas  void setup() {  pinMode(RELAY\_PIN, OUTPUT);  pinMode(FAN\_PIN, OUTPUT);  // Inisialisasi relay dan kipas dalam keadaan mati  digitalWrite(RELAY\_PIN, LOW);  digitalWrite(FAN\_PIN, LOW);  Serial.begin(9600);  Serial.println("Fan Control");  }  void loop() {  Serial.println("Fan is ON");  turnFanOn();  delay(5000); // Tunggu 5 detik  Serial.println("Fan is OFF");  turnFanOff();  delay(5000); // Tunggu 5 detik  }  void turnFanOn() {  digitalWrite(RELAY\_PIN, HIGH); // Nyalakan relay  digitalWrite(FAN\_PIN, HIGH); // Nyalakan kipas  }  void turnFanOff() {  digitalWrite(RELAY\_PIN, LOW); // Matikan relay  digitalWrite(FAN\_PIN, LOW); // Matikan kipas  } |

* *Source Code Light Bar* LED

|  |
| --- |
| #define LAMP\_PIN 9 // Pin yang terhubung ke transistor  void setup() {  pinMode(LAMP\_PIN, OUTPUT);    // Inisialisasi lampu dalam keadaan mati  digitalWrite(LAMP\_PIN, LOW);  Serial.begin(9600);  Serial.println("Lamp Control");  }  void loop() {  Serial.println("Lamp is ON");  turnLampOn();  delay(5000); // Tunggu 5 detik  Serial.println("Lamp is OFF");  turnLampOff();  delay(5000); // Tunggu 5 detik  }  void turnLampOn() {  digitalWrite(LAMP\_PIN, HIGH); // Nyalakan lampu  }  void turnLampOff() {  digitalWrite(LAMP\_PIN, LOW); // Matikan lampu  } |

* *Source Code* LCD I2C 16×2

|  |
| --- |
| #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  // Inisialisasi objek LCD  LiquidCrystal\_I2C lcd(0x27, 16, 2); // Alamat I2C dan ukuran LCD (16 kolom, 2 baris)  void setup() {  // Inisialisasi LCD  lcd.begin(16,2);    // Menampilkan pesan selamat datang  lcd.print("LCD I2C Testing");  delay(2000); // Tunggu 2 detik  // Bersihkan layar  lcd.clear();  }  void loop() {  // Menampilkan pesan bergerak dari kiri ke kanan  lcd.setCursor(0,0);  lcd.print("Hello, World!");  delay(1000); // Tunggu 1 detik  lcd.clear(); // Bersihkan layar  delay(500); // Tunggu setengah detik  // Menampilkan pesan bergerak dari kanan ke kiri  lcd.setCursor(15,1);  lcd.print("Hello!");  delay(1000); // Tunggu 1 detik  lcd.clear(); // Bersihkan layar  delay(500); // Tunggu setengah detik  } |

1. Melakukan *testing* terhadap sensor yang ada pada rangkaian alat dengan *source code* khusus.

* *Source Code* DHT11

|  |
| --- |
| #include <DHT.h>  #define DHTPIN 2 // Pin yang terhubung ke sensor DHT11  #define DHTTYPE DHT11 // Tipe sensor yang digunakan (DHT11 atau DHT22)  DHT dht(DHTPIN, DHTTYPE);  void setup() {  Serial.begin(9600);  Serial.println("DHT11 Testing");  dht.begin();  }  void loop() {  delay(2000); // Tunggu 2 detik antara pembacaan  float humidity = dht.readHumidity(); // Baca kelembaban  float temperature = dht.readTemperature(); // Baca suhu dalam derajat Celsius (default)  // Periksa jika bacaan suhu dan kelembaban berhasil  if (isnan(humidity) || isnan(temperature)) {  Serial.println("Gagal membaca dari sensor DHT11");  return;  }  // Tampilkan hasil pembacaan suhu dan kelembaban  Serial.print("Kelembaban: ");  Serial.print(humidity);  Serial.print("%\t");  Serial.print("Suhu: ");  Serial.print(temperature);  Serial.println("°C");  } |

* *Source Code* BH1750

|  |
| --- |
| #include <Wire.h>  #include <BH1750.h>  BH1750 lightSensor;  void setup() {  Serial.begin(9600);  Serial.println("BH1750 Testing");  // Inisialisasi sensor cahaya  lightSensor.begin();  }  void loop() {  // Membaca intensitas cahaya dalam lux  uint16\_t lux = lightSensor.readLightLevel();  Serial.print("Intensitas Cahaya: ");  Serial.print(lux);  Serial.println(" lux");  delay(1000); // Tunggu 1 detik sebelum membaca lagi  } |

### Integrasi Sistem

1. Melakukan *wiring* terhadap sensor, aktuator, dan mikrokontroler yang telah mengalami proses *testing*.
2. Membuat *source code* berdasarkan *wiring* yang telah dibuat.
3. Melakukan *running program* pada *source code* yang telah dibuat.

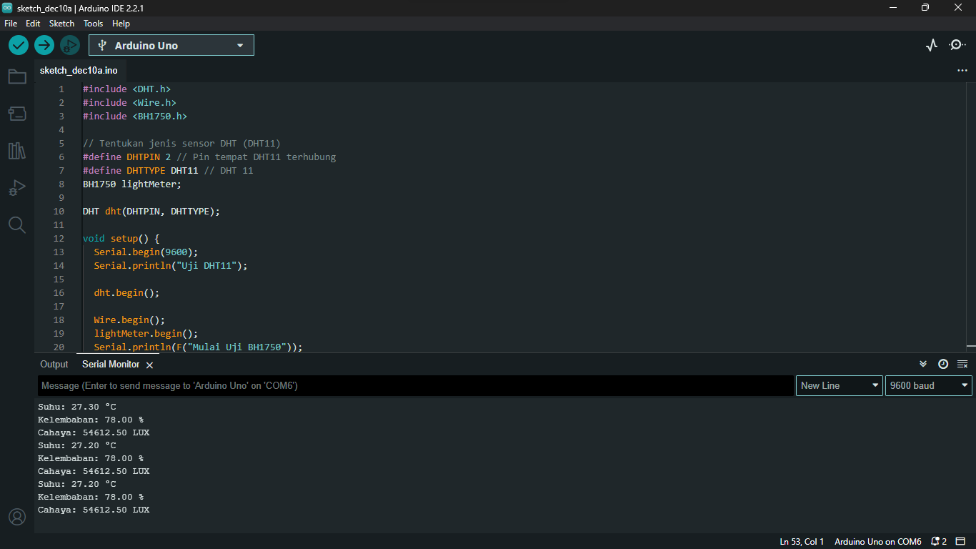
* *Source Code* Keseluruhan Sistem

|  |
| --- |
| #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  #include <DHT.h>  #include <BH1750.h>  #define DHTPIN 13  #define TIPEDHT DHT11  #define relay 3  #define fanPin 2 // Pin untuk mengontrol kipas  #define pumpPin 5 // Pin untuk mengontrol water pump  DHT dht(DHTPIN, TIPEDHT);  BH1750 Cahaya;  // Inisialisasi objek LCD menggunakan I2C  LiquidCrystal\_I2C lcd(0x27, 16, 2);  void setup() {  Serial.begin(9600);  lcd.begin(16, 2); // Inisialisasi LCD dengan 16 kolom dan 2 baris  lcd.backlight(); // Nyalakan backlight LCD  pinMode(relay, OUTPUT);  pinMode(fanPin, OUTPUT);  pinMode(pumpPin, OUTPUT);  dht.begin();  Cahaya.begin();  // Menampilkan judul di LCD  lcd.setCursor(3, 0);  lcd.print("Monitoring");  lcd.setCursor(4, 1);  lcd.print("GROW TENT");  delay(2000); // Tunggu 2 detik  // Bersihkan LCD  lcd.clear();  }  void loop() {  unsigned int Variabel\_1 = 0;  int IntensitasCahaya = Cahaya.readLightLevel();  float kelembaban = dht.readHumidity();  float temperatur = dht.readTemperature();  if (isnan(kelembaban) || isnan(temperatur) || isnan(IntensitasCahaya)) {  Serial.println(F("Gagal membaca sensor"));  return;  }  {  if (millis() - Variabel\_1 > 300) {  // Mengontrol kipas berdasarkan tingkat kelembaban  if (kelembaban < 80) {  digitalWrite(fanPin, HIGH); // Nyalakan kipas  } else {  digitalWrite(fanPin, LOW); // Matikan kipas  }  // Mengontrol water pump berdasarkan suhu  if (temperatur > 30) {  digitalWrite(pumpPin, HIGH); // Nyalakan water pump  } else {  digitalWrite(pumpPin, LOW); // Matikan water pump  }  // Mengontrol relay berdasarkan tingkat kelembaban    }  // Menampilkan data di LCD  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("H:");  lcd.print(kelembaban);  lcd.print("% ");  lcd.setCursor(0, 1);  lcd.print("T:");  lcd.print(temperatur);  lcd.print("C ");  lcd.setCursor(9, 1); // Pindah ke kolom ke-9 pada baris ke-1  lcd.print("Lux: ");  lcd.print(IntensitasCahaya);  }  Serial.print("Kelembaban: ");  Serial.print(kelembaban);  Serial.print("% Temperatur: ");  Serial.print(temperatur);  Serial.print("°C ");  Serial.print(" Lux: ");  Serial.print(IntensitasCahaya);  Serial.println();  delay(1000);  } |

1. Inisiasi *source code* pada *unit controller.*
2. Melakukan *testing* terhadap integrasi sistem.

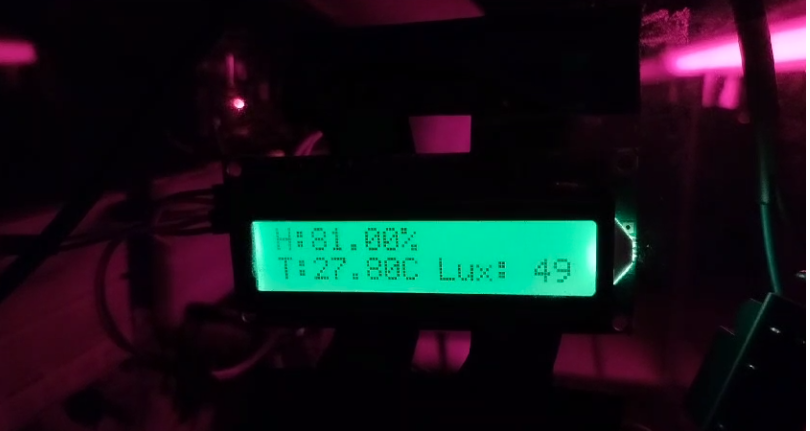
# HASIL DAN PEMBAHASAN

## Hasil Pengujian



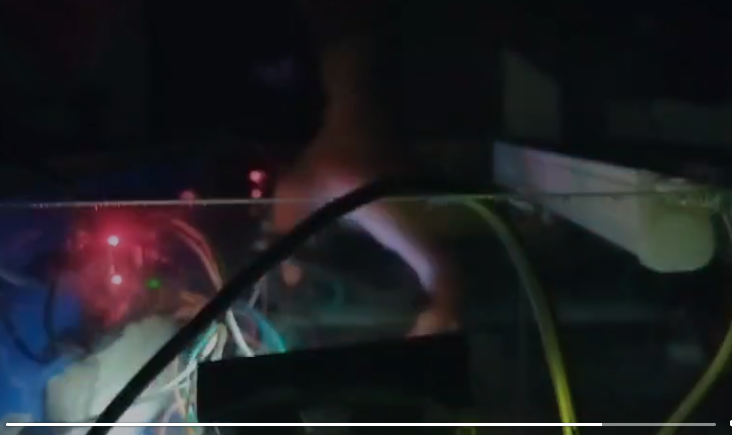
**Gambar 4.1: Hasil Pembacaan Lux Sensor dan DHT11 Humidity & Temperature Sensor**

Gambar 4.1 merupakan tampilan *serial monitor* Arduino IDE ketika BH1750 *lux sensor* dan DHT11 *temperature & humidity sensor* telah berhasil membaca variabel yang telah dideklarasikan. Sensor BH1750 membaca variabel atau besaran *lux,* dimana *lux* merupakan besaran yang mengukur paparan intensitas cahaya terhadap suatu bidang atau permukaan. Selain itu seperti yang telah diketahui sebelumnya, sensor DHT11 mengukur kelembaban serta suhu yang terdapat pada suatu medium atau permukaan.

****

**Gambar 4.2: Kipas mati saat kelembaban mencapai lebih dari 80% RH**

Gambar 4.2 menampilkan bukti hasil percobaan bahwa kipas akan mati saat kelembaban ruangan lebih dari 80% RH. Hal ini bertujuan untuk menjaga kelembaban ideal bagi tanaman, yakni >80% RH. Selain itu, kipas yang menyala berfungsi untuk menarik udara kotor yang ada pada *box* tanaman sehingga kelembaban tetap terjaga.



**Gambar 4.3: Lampu bar mati saat lux melebihi 60**

Gambar 4.3 menunjukkan bahwa lampu bar mati ketika variabel lux telah melebihi *threshold* yang ditentukan, yakni 60. Perlu diketahui bahwa nilai lux yang ideal bagi tanaman hias adalah ≥60 lux, sehingga apabila nilai lux kurang dari 60, maka lampu *bar* pada *box* akan menyala. Hal ini bertujuan untuk menjaga tanaman tetap mendapatkan pencahayaan yang cukup untuk tetap hidup dan sehat

## Analisis

Dari percobaan yang dilakukan ada beberapa kendala yang muncul dari berbagai faktor, baik faktor kesalahan pemilihan komponen, kualitas komponen dan faktor lingkungan. Berikut beberapa kendala beserta evaluasi yang bisa dilakukan.

Masukan pada sensor intensitas cahaya yang tidak akurat.

Kendala ini bisa disebabkan karena kondisi lingkungan yang dinamis sehingga pembacaan masukan nilai intenistas cahaya tidak akurat dan juga bisa disebabkan karena kualitas komponen. Kendala ini bisa diatasi dengan melakukan pengujian sensor pada kondisi lingkungan yang ideal dan melakukan kalibrasi ulang.

Pecahnya box/casing alat.

Kendala ini disebabkan adanya kecelakan pada saat membawa alat dan karena faktor jenis bahan yang dipilih merupakan bahan yang rentan pecah yaitu kaca. Kendala ini bisa diatasi dengan pemilihan bahan yang lebih kuat dan flexibel seperti fiber glass atau akrilik.

# KESIMPULAN DAN SARAN

## Kesimpulan

Pembuatan alat ini merupakan pengembangan *grow tent* yang dimana penggunaannya untuk tanaman hias, Pengembangan yang dilakukan yaitu menambahkan sensor dan aktuator sehingga rekayasa kondisi lingkungan bisa di otomatisasi. Tujuan yang diharapkan dengan adanya alat ini yaitu bisa mempermudah merawat tanaman hias di berbagai kondis. Namun bukan tanpa rintangan, alat ini memiliki beberapa kendala dalam pem buatannya yaitu masalah akurasi sensor dan juga pemilihan bahan untuk box/casing alat yang masih kurang kuat.

## Saran

Adanya saran untuk perancangan *grow tent* ini dengan melakukan pengujian sensor pada kondisi lingkungan yang ideal dan melakukan kalibrasi ulang., untuk kendala sensor intensitas cahaya yang tidak akurat. Serta memilih bahan yang lebih kuat dan flexible untuk mengatasi alat yang rentan rusak.

# REFERENSI

<https://youtu.be/y5CvL1RKvR4?si=bfs4KABfk98HCoWC>

# LAMPIRAN

# 

