

**PROYEK IEE2022**  
**MIKROKONTROLER DAN SENSOR I**



Disusun oleh:

|                      |             |
|----------------------|-------------|
| Cleve Giosia Adryana | (212100125) |
| Vincent Waiman       | (212100113) |
| Yohan Margo Widodo   | (212100207) |

**CALVIN INSTITUTE OF TECHNOLOGY**

**2023**

## DAFTAR ISI

|  |           |
|--|-----------|
| <b>HALAMAN JUDUL.....</b>                        | <b>1</b>  |
| <b>DAFTAR ISI.....</b>                           | <b>2</b>  |
| <b>BAB I PENDAHULUAN.....</b>                    | <b>3</b>  |
| 1. Latar Belakang Masalah.....                   | 3         |
| 2. Tujuan.....                                   | 3         |
| 3. Manfaat.....                                  | 4         |
| <b>BAB II PERANCANGAN.....</b>                   | <b>5</b>  |
| 1. Identifikasi Kebutuhan & Cara Kerja Alat..... | 5         |
| 2. Cara Perakitan Rangkaian.....                 | 7         |
| 3. Cara Pemrograman.....                         | 7         |
| a. Arduino Sensor.....                           | 7         |
| b. Arduino Aktuator.....                         | 12        |
| c. Basis Data & Antarmuka Pengguna.....          | 13        |
| 4. Perangkat Lunak.....                          | 16        |
| a. TinkerCAD.....                                | 16        |
| b. Software Arduino UNO.....                     | 16        |
| c. Visual Studio Code.....                       | 17        |
| <b>BAB III KESIMPULAN DAN SARAN.....</b>         | <b>18</b> |
| 1. Kesimpulan.....                               | 18        |
| 2. Saran.....                                    | 18        |
| <b>LAMPIRAN.....</b>                             | <b>19</b> |
| 1. Bill of Material.....                         | 19        |

# **BAB I PENDAHULUAN**

## **1. Latar Belakang Masalah**

Budidaya tanaman menjadi tren yang semakin populer di kalangan masyarakat saat ini. Beberapa jenis tanaman yang dibudidayakan membutuhkan kondisi iklim tertentu untuk tumbuh optimal, seperti suhu, kelembaban, dan intensitas cahaya yang tepat. Oleh karena itu, greenhouse menjadi solusi yang tepat untuk mengatasi masalah tersebut. Greenhouse adalah bangunan atau struktur yang dirancang khusus dengan sistem pengendalian iklim buatan yang memungkinkan tumbuhnya tanaman dalam kondisi yang ideal.

Namun, pengendalian iklim di dalam greenhouse secara manual dapat menjadi tugas yang sulit dan melelahkan bagi perawat tanaman, terutama jika harus dilakukan dengan perhatian yang besar dan terus-menerus. Kebanyakan orang memiliki kesibukan lain yang membutuhkan perhatian mereka, sehingga seringkali tidak memiliki waktu dan energi yang cukup untuk mengurus tanaman dengan baik. Oleh karena itu, diperlukan suatu sistem yang dapat membantu mengendalikan pengairan tanaman, suhu ruang greenhouse, dan pencahayaan secara otomatis sehingga tanaman dapat tumbuh dengan baik meskipun perawat memberikan perhatian minimum.

Sistem pengendalian iklim otomatis seperti ini akan sangat berguna dan efektif dalam memastikan bahwa tanaman mendapatkan kondisi ideal yang dibutuhkan untuk tumbuh dengan baik. Selain itu, sistem otomatis tersebut juga dapat membantu perawat tanaman untuk menghemat waktu dan tenaga dalam merawat tanaman di dalam greenhouse.

## **2. Tujuan**

Tujuan proyek ini adalah untuk merancang dan mengimplementasikan sistem pengendalian iklim otomatis yang dapat membantu perawat tanaman dalam mengendalikan pengairan tanaman, suhu ruang greenhouse, dan pencahayaan secara otomatis. Dengan sistem otomatis ini, perawat tanaman dapat menghemat waktu dan tenaga dalam merawat tanaman di dalam greenhouse. Selain itu, tujuan lain dari proyek ini adalah untuk memastikan bahwa kondisi ideal yang dibutuhkan oleh tanaman untuk tumbuh dengan baik dapat terpenuhi secara konsisten, sehingga memungkinkan tanaman tumbuh optimal dan menghasilkan hasil panen yang berkualitas.

### **3. Manfaat**

Manfaat dari proyek ini adalah meningkatkan efisiensi dan produktivitas dalam budidaya tanaman di greenhouse. Dengan menggunakan sistem pengendalian iklim otomatis, perawat tanaman dapat memantau kondisi tanaman dengan lebih mudah dan memastikan bahwa tanaman selalu mendapatkan kondisi yang tepat untuk tumbuh. Hal ini dapat mengurangi risiko kesalahan dalam mengendalikan pengairan, suhu, dan pencahayaan yang mungkin terjadi ketika dilakukan secara manual. Selain itu, penggunaan sistem otomatis ini juga dapat membantu mengurangi biaya operasional dalam jangka panjang karena mengurangi kebutuhan untuk tenaga kerja manusia. Akhirnya, manfaat penting lain dari sistem pengendalian iklim otomatis adalah dapat meningkatkan hasil panen dan kualitas produk karena tanaman dapat tumbuh optimal dalam kondisi ideal.

## **BAB II PERANCANGAN**

### **1. Identifikasi Kebutuhan & Cara Kerja Alat**

Alat yang kami hasilkan terdiri dari beberapa komponen seperti Arduino, sensor soil moisture, sensor suhu, sensor photoresistor LDR, relay, micro servo, LED pengganti AC, LED, dan resistor. Berikut adalah penjelasan cara kerja masing-masing komponen:

#### **a. Arduino**

Arduino berfungsi sebagai otak dari alat yang kami hasilkan. Arduino memproses data dari sensor dan memberikan perintah pada komponen lain seperti relay dan micro servo. Arduino juga dapat berkomunikasi dengan komputer melalui USB untuk memantau data sensor atau mengubah program alat.

#### **b. Sensor Soil Moisture**

Sensor soil moisture digunakan untuk memantau kelembaban tanah di dalam greenhouse. Sensor ini akan memberikan data kelembaban tanah pada Arduino. Arduino kemudian akan memproses data tersebut dan memberikan perintah pada relay untuk menghidupkan atau mematikan pompa air pada sistem irigasi tanaman.

#### **c. Sensor Suhu**

Sensor suhu digunakan untuk memantau suhu di dalam greenhouse. Sensor ini akan memberikan data suhu pada Arduino. Arduino kemudian akan memproses data tersebut dan memberikan perintah menyalakan atau mematikan AC.

#### **d. Sensor Photoresistor LDR**

Sensor photoresistor LDR digunakan untuk memantau cahaya di dalam greenhouse. Sensor ini akan memberikan data cahaya pada Arduino. Arduino kemudian akan memproses data tersebut dan memberikan perintah pada LED untuk menyalakan atau mematikan pencahayaan pada sistem pencahayaan greenhouse.

#### **e. Relay**

Relay digunakan sebagai saklar untuk menghidupkan atau mematikan AC dan LED pada sistem irigasi tanaman. Perintah untuk menghidupkan atau mematikan relay

diberikan oleh Arduino berdasarkan data suhu dan cahaya yang diterima dari sensor suhu dan LDR.

f. Micro Servo

Micro servo bekerja seperti sebuah kran air. Ketika sensor kelembaban tanah mendeteksi tanah kering, maka kran air akan terbuka sehingga dapat secara otomatis menyiram tanaman. Sedangkan ketika sensor kelembaban tanah mendeteksi tanah lembab, maka kran air akan tertutup.

g. LED (pengganti AC)

AC digunakan untuk memberikan pendinginan pada tanaman di dalam greenhouse. Perintah untuk menyalakan atau mematikan AC diberikan oleh Arduino berdasarkan data suhu yang diterima dari sensor suhu.

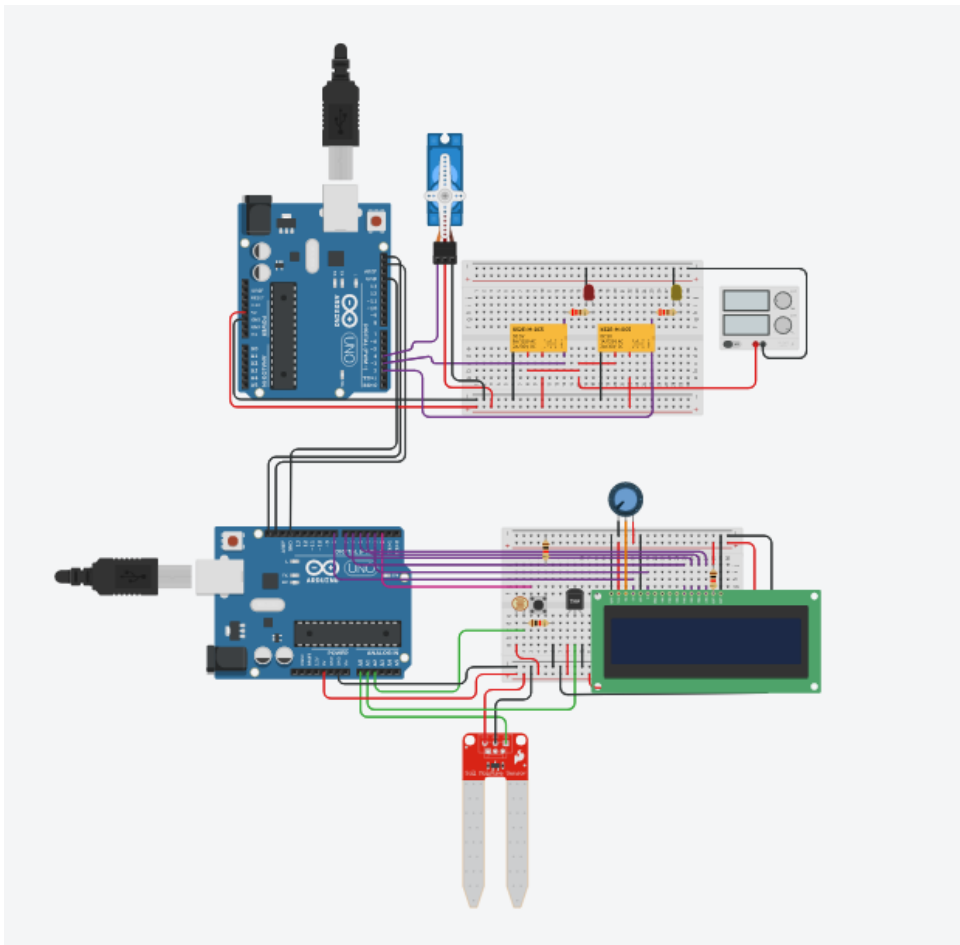
h. LED

LED bekerja sebagai penerang greenhouse. Led akan menyala jika cahaya pada greenhouse redup atau gelap.

i. Resistor

Resistor digunakan untuk menstabilkan arus listrik yang ada pada rangkaian.

## 2. Cara Perakitan Rangkaian



## 3. Cara Pemrograman

### a. Arduino Sensor

```
// C++ code
//
#include <Wire.h>
#include <LiquidCrystal.h>
#include <Servo.h>

int temp, light, moist;
const int m_sens = A0;
const int l_sens = A2;
const int t_sens = A1;
const int btt_pin = 2;
const int l_act = 12;
const int ac_act = 13;
const int rs = 8, en = 3, d4 = 6, d5 = 7, d6 = 4, d7 = 5;
bool pressed = false;
String tap = "Off", lamps = "Off", ac = "Off";
Servo servo5;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup()
```

```

{
  Wire.begin();
  Serial.begin(9600);
  lcd.begin(20, 4);
  servo5.attach(11);
  pinMode(btt_pin, INPUT_PULLUP);
  pinMode(l_act, OUTPUT);
  pinMode(ac_act, OUTPUT);
  lcd.print("Auto Greenhouse");
  lcd.setCursor(0, 1);
  lcd.print("Env. System");
  delay(500);
  lcd.clear();
  delay(500);
  attachInterrupt(digitalPinToInterrupt(2), lcd_showact, FALLING);
}

```

Kode di atas adalah program C++ untuk mengontrol sistem lingkungan otomatis pada rumah kaca. Program ini menggunakan tiga sensor analog (sensor suhu, cahaya, dan kelembaban tanah), sebuah tombol, dan dua aktuator (lampu dan AC) untuk mengatur lingkungan di dalam rumah kaca. Program menggunakan pustaka-pustaka Wire, LiquidCrystal, dan Servo untuk berkomunikasi dengan sensor dan aktuator, serta untuk menampilkan informasi di layar LCD. Fungsi setup() digunakan untuk menginisialisasi sensor, aktuator, dan layar LCD, serta untuk menetapkan interrupt pada tombol. Program kemudian menggunakan loop() untuk membaca nilai sensor dan mengatur aktuator berdasarkan nilai yang dibaca. Program juga menampilkan nilai sensor dan status aktuator pada layar LCD.

```

void lcd_showsens(int temp, int light, int moist)
{
  lcd.clear();
  lcd.print("S.M. : ");
  lcd.setCursor(6, 0);
  lcd.print(moist);
  lcd.setCursor(0, 1);
  lcd.print("Light: ");
  lcd.setCursor(6, 1);
  lcd.print(light);
  lcd.setCursor(-4, 2);
  lcd.print("Temp :");
  lcd.setCursor(2, 2);
  lcd.print(temp);
}

```



Kode di atas adalah sebuah fungsi yang menampilkan tiga variabel (suhu, cahaya, kelembaban) ke layar LCD. Pertama, layar LCD dikosongkan dengan memanggil fungsi `lcd.clear()`. Kemudian, tiga variabel yang dimasukkan sebagai parameter fungsi (`temp`, `light`, dan `moist`) ditampilkan di layar LCD menggunakan fungsi `lcd.print()`. `SetCursor` digunakan untuk mengatur posisi kursor di layar LCD. Pada baris pertama, variabel `moist` ditampilkan dengan posisi kursor pada kolom ke-6. Pada baris kedua, variabel `light` ditampilkan dengan posisi kursor pada kolom ke-6. Pada baris ketiga, variabel `temp` ditampilkan dengan posisi kursor pada kolom ke-2. Secara keseluruhan, fungsi ini menampilkan nilai suhu, cahaya, dan kelembaban ke layar LCD dengan format yang terstruktur.

```
void lcd_showact()
{
    pressed = !pressed;
    if (pressed) {
        lcd.clear();
        lcd.print("Water:");
        lcd.setCursor(6, 0);
        lcd.print(temp);
        lcd.setCursor(0, 1);
        lcd.print("Lamps:");
        lcd.setCursor(6, 1);
        lcd.print(lamps);
        lcd.setCursor(-4, 2);
        lcd.print("AC  :");
        lcd.setCursor(2, 2);
        lcd.print(ac);
    }
}
```

Kode di atas adalah sebuah fungsi yang menampilkan status tiga perangkat (keran, lampu, AC) ke layar LCD secara bergantian ketika fungsi dipanggil.

Pertama, variabel `pressed` diubah nilainya menjadi kebalikan dari nilai sebelumnya menggunakan operator `!`. Kemudian, menggunakan percabangan `if`, jika `pressed` bernilai `true`, maka layar LCD dikosongkan dengan memanggil fungsi `lcd.clear()`. Selanjutnya, tiga variabel yang merepresentasikan status perangkat (`tap`, `lamps`, dan `ac`) ditampilkan di layar LCD menggunakan fungsi `lcd.print()`.

SetCursor digunakan untuk mengatur posisi kursor di layar LCD. Pada baris pertama, variabel tap ditampilkan dengan posisi kursor pada kolom ke-6. Pada baris kedua, variabel lamps ditampilkan dengan posisi kursor pada kolom ke-6. Pada baris ketiga, variabel ac ditampilkan dengan posisi kursor pada kolom ke-2.

Secara keseluruhan, fungsi ini menampilkan status tiga perangkat ke layar LCD dengan format yang terstruktur dan bergantian setiap kali fungsi dipanggil.

```
void loop()
{
    Wire.beginTransmission(8);

    int x = analogRead(l_sens);
    temp = analogRead(t_sens);
    light = 0.0048*pow(x,2) + 5.1035*x - 116.11;
    moist = analogRead(m_sens);

    byte dataArray[3];

    if (temp > 1020) {
        ac = "On";
        dataArray[0] = 1;
    } else {
        ac = "Off";
        dataArray[0] = 0;
    }
    if (light < 15) {
        lamps = "On";
        dataArray[1] = 1;
    } else {
        lamps = "Off";
        dataArray[1] = 0;
    }
    if (moist < 1000) {
        tap = "On";
        dataArray[2] = 1;
    } else {
        tap = "Off";
        dataArray[2] = 0;
    }
    digitalWrite(13, HIGH);
    Wire.write(dataArray, 3);
    digitalWrite(13, LOW);

    Wire.endTransmission();

    Serial.print("Temp:");
```

```

    Serial.print(temp);
    Serial.print(", Light:");
    Serial.print(light);
    Serial.print(", Moist:");
    Serial.print(moist);
    Serial.print(", ac_status:");
    Serial.print(ac);
    Serial.print(", lamps_status:");
    Serial.print(lamps);
    Serial.print(", tap_status:");
    Serial.println(tap);

    if (!pressed) lcd_showsens(temp, light, moist);

    delay(1000);
}

```

Fungsi loop() digunakan untuk membaca nilai sensor dan mengatur aktuator untuk mengontrol lingkungan di dalam rumah kaca. Pertama, program memulai transmisi data menggunakan protokol I2C untuk berkomunikasi dengan modul lain (ditunjukkan oleh alamat slave 8). Kemudian, program membaca nilai sensor suhu, cahaya, dan kelembaban tanah menggunakan fungsi analogRead(), dan menyimpan nilai-nilai tersebut ke dalam variabel temp, light, dan moist. Program juga membuat array byte dataArray[3] yang digunakan untuk menyimpan status aktuator dalam bentuk biner (1 untuk ON dan 0 untuk OFF).

Selanjutnya, program mengevaluasi nilai sensor dan mengatur status aktuator (lampu, AC, dan pompa air) berdasarkan nilai-nilai tersebut. Jika suhu di atas 1020, maka AC akan dihidupkan (ac = "On"), dan dataArray[0] akan diset menjadi 1. Jika cahaya kurang dari 15, maka lampu akan dihidupkan (lamps = "On"), dan dataArray[1] akan diset menjadi 1. Jika kelembaban tanah kurang dari 1000, maka pompa air akan dihidupkan (tap = "On"), dan dataArray[2] akan diset menjadi 1. Setelah itu, status aktuator dan nilai sensor ditampilkan pada Serial Monitor.

Selain itu, jika tombol tidak ditekan (pressed = false), program akan menampilkan nilai sensor suhu, cahaya, dan kelembaban tanah pada layar LCD menggunakan fungsi lcd\_showsens(). Program juga mengirimkan dataArray melalui protokol I2C dan mengakhiri transmisi.

Terakhir, program melakukan delay selama 1000 milidetik (1 detik) sebelum loop() diulang kembali.

## b. Arduino Aktuator

```
#include <Wire.h>
#include <Servo.h>

byte pinArray[3] = {2, 3, 4};
Servo servo5;

void setup() {
  Wire.begin(8);
  servo5.attach(4);
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);
}

void loop() {
  delay(1000);
}

void receiveEvent(int numBytes) {
  while(Wire.available() > 0) {
    for (int i=0; i<3; i++) {
      int data = Wire.read();
      Serial.println(data);
      if (i == 2) {
        servo5.write(90 * data);
      }
      digitalWrite(pinArray[i], data);
    }
  }
}
```

Kode di atas merupakan program untuk mengontrol tiga pin output (dalam hal ini dipakai sebagai kontrol servo dan LED) secara nirkabel melalui protokol I2C sebagai master. Program ini dijalankan oleh sebuah mikrokontroler (Arduino atau sejenisnya) yang bertindak sebagai slave dan menerima data dari perangkat master melalui koneksi I2C.

Pada bagian setup(), program menginisialisasi koneksi I2C dengan Wire.begin(8), dan menghubungkan servo ke pin 4 melalui servo5.attach(4).

Kemudian, program menyiapkan sebuah fungsi `receiveEvent()` sebagai penanganan terhadap data yang diterima.

Pada bagian `loop()`, program menunda eksekusi selama 100ms dengan menggunakan `delay(100)`, sehingga program dapat melakukan tugas lainnya.

Fungsi `receiveEvent(int numBytes)` digunakan untuk menangani data yang diterima melalui koneksi I2C. Ketika ada data yang tersedia, fungsi akan membaca semua data yang diterima dan menyimpannya dalam variabel `data`. Data kemudian akan digunakan untuk mengontrol pin output dan posisi servo. Data pada elemen terakhir dari array `pinArray` akan digunakan untuk mengontrol posisi servo, dan akan dikalikan dengan 90 untuk mendapatkan nilai sudut dalam derajat.

Selama program berjalan, data yang diterima dari master akan dicetak pada serial monitor dengan menggunakan `Serial.println(data)`.

### c. Basis Data & Antarmuka Pengguna

```
import sqlite3
import threading
from tkinter import ttk

import serial
import datetime
import time
import tkinter as tk

def create_cursor(database_path):
    conn = sqlite3.connect(database_path, check_same_thread=False)
    cursor = conn.cursor()
    return cursor

def update_label(label, text):
    label.config(text=text)

def update_gui(tab1, tab2, cursor, ser):
    temp_text = tk.Label(tab1, text="Temperature", font=("Arial", 16),
        anchor="w", width=25)
    temp_text.grid(row=0, column=0)
    colon = tk.Label(tab1, text=":", font=("Arial", 16))
```

```

        colon.grid(row=0, column=1)
        temp_label = tk.Label(tab1, text="N/A", font=("Arial", 16),
                                anchor="w")
        temp_label.grid(row=0, column=2)

        hum_text = tk.Label(tab1, text="Humidity", font=("Arial", 16),
                                anchor="w", width=25)
        hum_text.grid(row=1, column=0)
        colon = tk.Label(tab1, text=":", font=("Arial", 16))
        colon.grid(row=1, column=1)
        hum_label = tk.Label(tab1, text="N/A", font=("Arial", 16), anchor="w")
        hum_label.grid(row=1, column=2)

        light_text = tk.Label(tab1, text="light", font=("Arial", 16),
                                anchor="w", width=25)
        light_text.grid(row=2, column=0)
        colon = tk.Label(tab1, text=":", font=("Arial", 16))
        colon.grid(row=2, column=1)
        light_label = tk.Label(tab1, text="N/A", font=("Arial", 16),
                                anchor="w")
        light_label.grid(row=2, column=2)

        # tab 2 Labels
        water_text = tk.Label(tab2, text="Water", font=("Arial", 16),
                                anchor="w", width=25)
        water_text.grid(row=0, column=0)
        colon = tk.Label(tab2, text=":", font=("Arial", 16))
        colon.grid(row=0, column=1)
        water_label = tk.Label(tab2, text="N/A", font=("Arial", 16),
                                anchor="w")
        water_label.grid(row=0, column=2)

        lamp_text = tk.Label(tab2, text="Lamps", font=("Arial", 16),
                                anchor="w", width=25)
        lamp_text.grid(row=1, column=0)
        colon = tk.Label(tab2, text=":", font=("Arial", 16))
        colon.grid(row=1, column=1)
        lamp_label = tk.Label(tab2, text="N/A", font=("Arial", 16),
                                anchor="w")
        lamp_label.grid(row=1, column=2)

        ac_text = tk.Label(tab2, text="Air Conditioner", font=("Arial", 16),
                                anchor="w", width=25)
        ac_text.grid(row=2, column=0)
        colon = tk.Label(tab2, text=":", font=("Arial", 16))
        colon.grid(row=2, column=1)
        ac_label = tk.Label(tab2, text="N/A", font=("Arial", 16), anchor="w")
        ac_label.grid(row=2, column=2)

        def update():
            data = ser.readline().decode('latin-1').strip()
            values = data.split(',')

            temp = (values[0].split(':')[1])
            light = (values[1].split(':')[1])
            moist = (values[2].split(':')[1])

```

```

lamps = (values[4].split(':')[1])
ac = (values[3].split(':')[1])
tap = (values[5].split(':')[1])

print(temp, light, moist, ac, lamps, tap)
now = datetime.datetime.now()

tanggal = now.date()
waktu = now.time()

jam = waktu.hour
menit = waktu.minute
detik = waktu.second

timenow = str(jam) + ":" + str(menit) + ":" + str(detik)

cursor.execute(
    "INSERT INTO monitoring (tanggal, time, kelembapan, suhu,
cahaya, status_keran, status_ac, "
    "status_lampu)"
    "VALUES (?, ?, ?, ?, ?, ?, ?, ?)",
    (tanggal, timenow, moist, temp, light, tap, ac, lamps))

cursor.connection.commit()

update_label(temp_label, "{}".format(temp))
update_label(hum_label, "{}".format(moist))
update_label(light_label, "{} lux".format(light))

update_label(water_label, "{}".format(tap))
update_label(lamp_label, "{}".format(lamps))
update_label(ac_label, "{}".format(ac))

tab1.after(1000, update)

tab1.after(1000, update)

def main():
    cursor = create_cursor('greenhouse_project')

    ser = serial.Serial('COM5', 9600)

    cursor.execute('''CREATE TABLE IF NOT EXISTS monitoring (tanggal DATE,
time TEXT, kelembapan INT, suhu INT,
cahaya INT, status_keran TEXT, status_ac TEXT, status_lampu TEXT)''')

    root = tk.Tk()
    root.title("Greenhouse Monitoring System")
    root.geometry("500x200")

    notebook = ttk.Notebook(root)

    greenhouse_label = tk.Label(root, text="Greenhouse System",
font=("Arial", 20), anchor="center")
    greenhouse_label.grid(row=0, columnspan=3)

```

```

tab1 = ttk.Frame(notebook, width=500, height=180)
tab2 = ttk.Frame(notebook, width=500, height=180)

notebook.add(tab1, text="Information")
notebook.add(tab2, text="Status")
notebook.grid(row=2, column=0, columnspan=2, sticky="nsew")

gui_thread = threading.Thread(target=update_gui, args=(tab1, tab2,
cursor, ser))
gui_thread.start()

try:
    root.mainloop()
except KeyboardInterrupt:
    print("KeyboardInterrupt received. Stopping...")
    cursor.close()
    ser.close()
    gui_thread.join()

if __name__ == '__main__':
    main()

```

## 4. Perangkat Lunak

### a. TinkerCAD

TinkerCAD merupakan perangkat lunak yang sangat membantu dalam proyek kami. Dengan TinkerCAD, kami dapat membuat prototipe rangkaian dengan mudah dan cepat. Kami dapat merancang dan mensimulasikan rangkaian elektronik, termasuk menggunakan komponen-komponen seperti sensor, lampu, dan relay, untuk memastikan bahwa ide yang kami miliki dapat direalisasikan dengan baik sebelum kami melakukan pembuatan fisiknya. Hal ini sangat membantu kami dalam menghemat waktu dan biaya dalam proses pembuatan fisik prototipe, serta memastikan bahwa prototipe yang kami hasilkan sesuai dengan ekspektasi dan tujuan kami.

### b. *Software* Arduino UNO

Software Arduino UNO merupakan perangkat lunak yang sangat penting dalam proyek kami. Dalam konteks proyek kami, board Arduino UNO digunakan untuk mengontrol berbagai komponen elektronik, seperti sensor, lampu, dan relay, berdasarkan data yang diterima dari sensor dan kondisi lingkungan di dalam greenhouse. Dengan menggunakan software Arduino



UNO, kami dapat memprogram board Arduino UNO dan mengunggah kode yang telah dibuat ke board tersebut. Hal ini memungkinkan kami untuk mengontrol dan memantau kondisi lingkungan di dalam greenhouse dengan lebih efektif dan efisien.

**c. Visual Studio Code**

Visual Studio Code juga merupakan perangkat lunak yang sangat penting dalam proyek kami. Dengan menggunakan Visual Studio Code, kami dapat membuat program basis data untuk memasukkan status lampu, AC, dan kran air ke dalam database. Selain itu, kami juga dapat membuat kode user interface yang dapat digunakan untuk mengontrol aktuator dalam greenhouse dengan mudah.

## **BAB III KESIMPULAN DAN SARAN**

### **1. Kesimpulan**

Dalam kesimpulannya, dapat disimpulkan bahwa penggunaan teknologi mikrokontroler dan sensor dalam pertanian dapat memberikan solusi yang efektif dan efisien untuk mengoptimalkan pertumbuhan tanaman di dalam greenhouse. Dengan teknologi ini, petani dapat mengontrol lingkungan pertumbuhan tanaman dengan lebih tepat dan akurat, sehingga dapat meningkatkan kualitas dan kuantitas hasil pertanian, serta mengurangi penggunaan pupuk dan pestisida yang berlebihan. Selain itu, penggunaan teknologi ini juga dapat mencegah kerusakan tanah dan pencemaran air yang dapat disebabkan oleh kegiatan pertanian dalam skala besar. Oleh karena itu, dukungan dan pengembangan lebih lanjut dalam bidang ini sangat penting untuk memberikan dampak positif yang besar pada sektor pertanian dan lingkungan. Dengan adanya proyek ini, diharapkan dapat mendorong perkembangan teknologi di sektor pertanian dan memberikan kontribusi yang besar bagi keberlanjutan pertanian dan lingkungan.

### **2. Saran**

Dalam proyek ini, terdapat beberapa saran yang dapat diberikan untuk meningkatkan efektivitas dan efisiensi penggunaan teknologi mikrokontroler dan sensor dalam pertanian. Pertama, perlu dilakukan pengujian yang lebih lanjut untuk memastikan keakuratan dan keandalan sensor yang digunakan dalam proyek ini. Kedua, perlu dikembangkan algoritma yang lebih canggih untuk mengoptimalkan pengaturan lingkungan dalam greenhouse, sehingga dapat meminimalkan penggunaan sumber daya yang tidak perlu. Ketiga, perlu diberikan edukasi dan pelatihan kepada petani mengenai penggunaan teknologi ini, sehingga dapat digunakan secara maksimal dan memberikan manfaat yang optimal. Keempat, perlu dikembangkan solusi yang lebih terjangkau dan mudah diakses oleh petani, sehingga dapat meningkatkan aksesibilitas dan penggunaan teknologi ini di seluruh wilayah pertanian. Dengan penerapan saran-saran tersebut, diharapkan teknologi mikrokontroler dan sensor dapat memberikan dampak positif yang lebih besar pada sektor pertanian dan lingkungan secara keseluruhan.

## LAMPIRAN

### 1. Bill of Material

| No. | Nama Komponen        | Tipe Komponen | Jumlah Unit | Harga/pc   | Link Pembelian  |
|-----|----------------------|---------------|-------------|------------|---|
| 1.  | Arduino + kabel      | UNO           | 2           | Rp129.900  | <a href="https://tokopedia.link/F3jVnND7lzb">https://tokopedia.link/F3jVnND7lzb</a>   |
| 2.  | Breadboard (30 rows) | -             | 1           | Rp. 8.900  | <a href="https://www.tokopedia.com/supreme/mb-102-400-hole-breadboard-arduino-prototyping">https://www.tokopedia.com/supreme/mb-102-400-hole-breadboard-arduino-prototyping</a> |
| 3.  | Breadboard (63 rows) | -             | 1           | Rp. 10.900 | <a href="https://www.tokopedia.com/supreme/mb-102-830-hole-breadboard-arduino-prototyping">https://www.tokopedia.com/supreme/mb-102-830-hole-breadboard-arduino-prototyping</a> |
| 4.  | Set kabel jumper     | -             | 1           | Rp. 12.500 | <a href="https://tokopedia.link/EB1eB7K7lzb">https://tokopedia.link/EB1eB7K7lzb</a>   |
| 5.  | Button               | -             | 1           | Rp. 350    | <a href="https://tokopedia.link/tw4VICX7lzb">https://tokopedia.link/tw4VICX7lzb</a>   |
| 6.  | LCD                  | 16x4          | 1           | Rp.65.000  | <a href="https://tokopedia.link/ynuVezK7lzb">https://tokopedia.link/ynuVezK7lzb</a>   |
| 7.  | Sensor Soil Moisture | YL-69         | 1           | Rp.46.000  | <a href="https://tokopedia.link/0ZZgQyS7lzb">https://tokopedia.link/0ZZgQyS7lzb</a>   |
| 8.  | Sensor Suhu          | DHT22         | 1           | Rp. 36.400 | <a href="https://tokopedia.link/pPKwdlS7lzb">https://tokopedia.link/pPKwdlS7lzb</a>   |

|     |                          |              |   |            |   |
|-----|--------------------------|--------------|---|------------|---|
| 9.  | Sensor Photoresistor LDR | Ukuran besar | 1 | Rp. 3.250  | <a href="https://www.tokopedia.com/jaya-sentosa-/ldr-besar-10mm-sensor-cahaya-photo-resistor-photoresistor">https://www.tokopedia.com/jaya-sentosa-/ldr-besar-10mm-sensor-cahaya-photo-resistor-photoresistor</a>       |
| 10. | Relay                    | HK19F        | 2 | Rp. 75.110 | <a href="https://www.tokopedia.com/bracuike-relay-hk19f-dc5v-shg-hk19f-dc12v-shg-hk19f-dc24v-shg-3v">https://www.tokopedia.com/bracuike-relay-hk19f-dc5v-shg-hk19f-dc12v-shg-hk19f-dc24v-shg-3v</a>                     |
| 11. | Micro Servo              | SG90         | 1 | Rp. 17.000 | <a href="https://www.tokopedia.com/starlectric/micro-servo-sg90">https://www.tokopedia.com/starlectric/micro-servo-sg90</a>   |
| 12. | LED Putih (AC)           | -            | 1 | Rp. 1.500  | <a href="https://www.tokopedia.com/ardushopid/led-rgb-4-pin-warna-merah-hijau-biru-5mm-arduino-common-cathode">https://www.tokopedia.com/ardushopid/led-rgb-4-pin-warna-merah-hijau-biru-5mm-arduino-common-cathode</a> |
| 13. | LED Kuning (Keran Air)   | -            | 1 | Rp. 1.500  | <a href="https://www.tokopedia.com/ardushopid/led-rgb-4-pin-warna-merah-hijau-biru-5mm-arduino-common-cathode">https://www.tokopedia.com/ardushopid/led-rgb-4-pin-warna-merah-hijau-biru-5mm-arduino-common-cathode</a> |

|     |          |         |   |           |   |
|-----|----------|---------|---|-----------|---|
| 14. | Resistor | 1k Ohm  | 1 | Rp.24.000 | <a href="https://tokopedia.lnk/3V8YPwd8lzb">https://tokopedia.lnk/3V8YPwd8lzb</a> |
|     |          | 10k Ohm | 1 |           |   |
|     |          | 220 Ohm | 1 |           |   |

Penilaian mandiri oleh masing-masing individu di dalam sebuah kelompok

Anggota kelompok 1: 212100125, Cleve Giosia Adryana

Anggota kelompok 2: 212100113, Vincent Waiman

Anggota kelompok 3: 212100207, Yohan Margo Widodo

|                    | Kontribusi menurut anggota kelompok 1 | Kontribusi menurut anggota kelompok 2 | Kontribusi menurut anggota kelompok 3 | Rata-rata kontribusi anggota kelompok |
|--------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| Anggota kelompok 1 | 33.3%                                 | 33.3%                                 | 33.3%                                 | 33.3%                                 |
| Anggota kelompok 2 | 33.3%                                 | 33.3%                                 | 33.3%                                 | 33.3%                                 |
| Anggota kelompok 3 | 33.3%                                 | 33.3%                                 | 33.3%                                 | 33.3%                                 |
|                    | 100%                                  | 100%                                  | 100%                                  | 100%                                  |