

# **Laporan Proyek Akhir**

## **Sistem Tertanam**

Rancang Bangun *Smart Green House Regulator* dengan Arduino Uno

<b>Nama Praktikan</b>	<b>NPM</b>
Gesha Mahendra C	1906348214
Yonathan	1906288316
Umar Abdul Aziz	1906248220



**Laboratorium Komputer - Departemen Fisika**  
**Fakultas Matematika dan Ilmu Pengetahuan Alam**  
**Universitas Indonesia**  
**2021**

## **JUDUL PROYEK**

Rancang Bangun Smart Green House Regulator dengan Arduino Uno

## **TUJUAN**

- Merancang Sistem *Smart Green House Regulator* dengan Arduino Uno,
- Melakukan simulasi untuk analisis sistem dengan memanfaatkan Arduino Uno, dan
- Menyelesaikan Tugas Akhir Praktikum Sistem Tertanam.

## **MANFAAT**

- Mengaplikasikan teori yang telah dipelajari selama Praktikum Sistem Tertanam, dan
- Otomatisasi deteksi dan kontrol system *Smart Green House Regulator*.

## **ALAT DAN BAHAN**

- Arduino Uno R3
- Breadboard
- LCD I2C
- Resistor 10K  $\Omega$
- Resistor 220  $\Omega$
- LED
- Relay
- Sensor TMP36
- Sensor Soil Moisture FC-28
- Light Dependent Resistor
- Servo
- Baterai AA  $\times 4$
- Kabel jumper male to male
- Kapasitor 0.33  $\mu F$
- Buzzer

## PENGELUARAN

No	Barang	Harga
1	Arduino Uno R3	Rp68.000,00
2	Breadboard	Rp18.000,00
3	LCD I2C	Rp35.000,00
4	Resistor, LDR, LED, dan Kapasitor	Rp20.000,00
5	Relay	Rp10.000,00
6	Sensor TMP36	Rp25.000,00
7	Sensor Soil Moisture FC-28	Rp10.000,00
8	Light Dependent Resistor	Rp5.000,00
9	Servo	Rp25.000,00
10	Baterai AA × 4	Rp10.000,00
11	Kabel jumper male to male	Rp25.000,00
12	Buzzer	Rp9.000,00
<b>Total</b>		<b>Rp260.000,00</b>

## TEORI DASAR

### Arduino R3

Arduino merupakan project dari Massimo Banzi dan rekan-rekannya dari Italia pada tahun 2005 yang bertujuan menjadi solusi murah dan mudah pelajar untuk mempelajari, dan menciptakan perangkat yang dapat berinteraksi dengan lingkungan menggunakan sensor dan actuator

Arduino UNO R3 adalah microcontroller yang berbasis ATmega 328P, memiliki 14 digital input/output pin, 6 analog pin input, 16MHz ceramic resonator, port USB, Power jack, pin ICSP (In Circuit Serial Programming) dan tombol reset.

ATmega328p yang menjadi microprocessor dari Arduino Uno R3 ini memiliki pin I/O, EEPROM dan SRAM yang lebih seikit dibandingkan dengan ATmega128. Tetapi untuk *instructions* yang dapat dijalankan oleh kedua microprosessor ini relative sama.

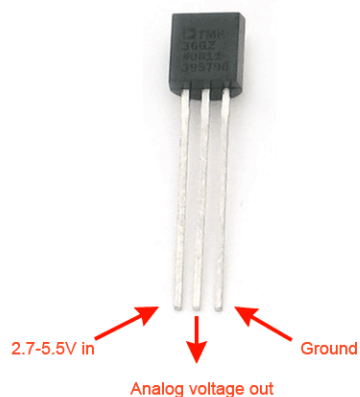
## LDR

LDR atau *Light Dependent Resistor* adalah jenis Resistor yang nilai hambatan atau nilai resistansinya dipengaruhi oleh intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (Light Dependent Resistor) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap.

Nilai Hambatan akan sebanding dengan jumlah cahaya yang diterimanya. Rentang Nilai Hambatan LDR akan mencapai 200 Kilo Ohm ( $k\Omega$ ) pada kondisi gelap dan menurun menjadi 500 Ohm ( $\Omega$ ) pada Kondisi Cahaya Terang.

Untuk menjadikan LDR sebagai sensor, diperlukan fungsi dari ADC yang mengubah data Analog menjadi digital, hal ini disebabkan LDR merupakan komponen Analog.

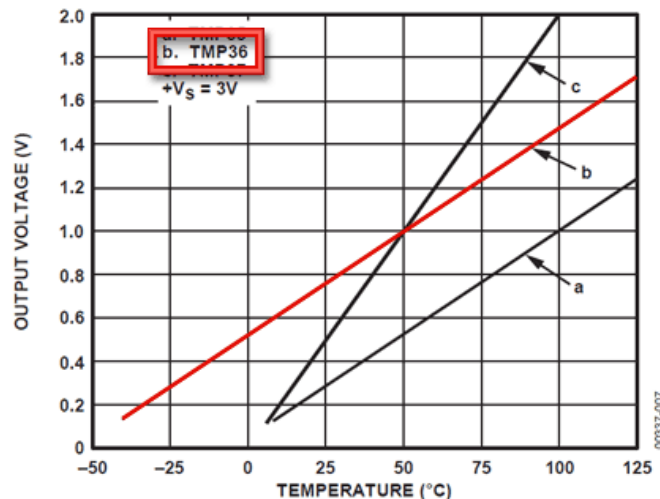
## TMP36



Gambar 1. Sensor Temperatur TMP36

[\[https://learn.adafruit.com/tmp36-temperature-sensor\]](https://learn.adafruit.com/tmp36-temperature-sensor)

TMP36 merupakan sensor yang dapat mengukur suhu pada rentang  $-40$  hingga  $150^{\circ}\text{C}$ , Sensor ini dapat beroperasi pada tegangan  $2.7 - 5.5\text{V}$  dengan arus  $0,05\text{mA}$ . TMP36 ini mengukur suhu tidak menggunakan mercury atau thermistor, melainkan dengan mengukur tegangan pada diode yang terdapat pada sensor, Perubahan tegangan inilah menimbulkan sinyal analog yang memiliki nilai proporsional dengan suhu.



Gambar 2. Grafik perbandingan Suhu dan Tegangan Output

[<https://learn.adafruit.com/tmp36-temperature-sensor>]

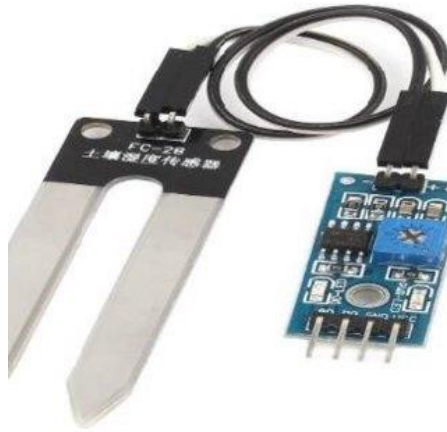
Sensor ini memiliki 3 input yaitu *voltage input*, *Analog voltage output*, dan *ground*. Kelebihan sensor ini dibandingkan dengan TMP34 dan TMP35 adalah sensor TMP36 memiliki rentang tegangan yang besar, hal ini menyebabkan sensor ini dapat mendeteksi suhu dibawah 0°C, Kekurangan dari sensor ini adalah kurangnya akurasi saat mengukur suhu diatas 125°C hal ini disebabkan oleh pengaruh suhu terhadap tegangan.

Simplifikasi dari perhitungan yang dilakukan sensor ini dapat dituliskan dengan persamaan berikut:  $\text{Temp in } ^\circ\text{C} = [(\text{Vout in mV}) - 500] / 10$

## FC-28

FC-28 adalah sensor pendeteksi kelembapan tanah, sensor ini memiliki dua batang terbuka yang berfungsi sebagai probe yang berperilaku layaknya variable resistor, jika terdapat air/kelembapan maka konduktivitas dari kedua batang akan naik yang dan resistansi, kedua nilai ini yang kemudian diproses oleh comparator dan menjadi sinyal output.

Sensor ini dapat beroperasi pada tegangan 3.3 – 5V, Sensor ini memiliki kelebihan dibandingkan sensor kelembapan lainnya yaitu sangat sensitif dengan kelembapan, untuk kekurangannya jika diletakan dalam tanah berongga dan terdapat air yang menempel pada batang pengukur sensor, maka akan terdapat beberapa error pengukuran.

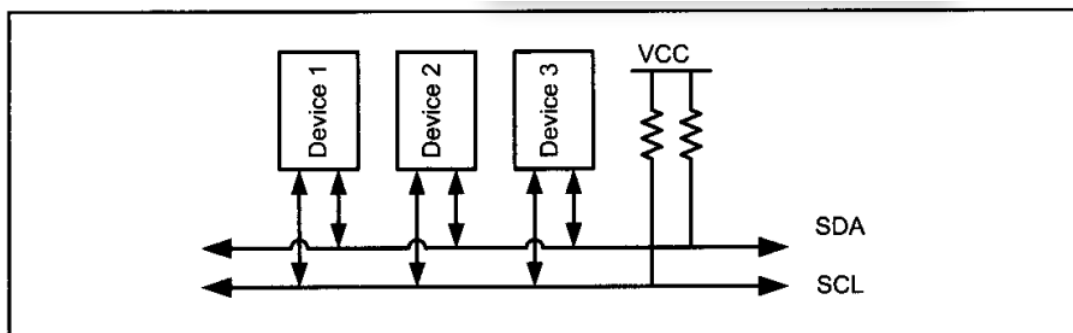


Gambar 3. Sensor Kelembapan FC-28

[[https://www.researchgate.net/figure/FC-28-Soil-Moisture-Sensor-DHT11-humidity-and-temperature-sensor-DHT11-humidity-and\\_fig2\\_342452977](https://www.researchgate.net/figure/FC-28-Soil-Moisture-Sensor-DHT11-humidity-and-temperature-sensor-DHT11-humidity-and_fig2_342452977)]

## Protokol Komunikasi I<sup>2</sup>C

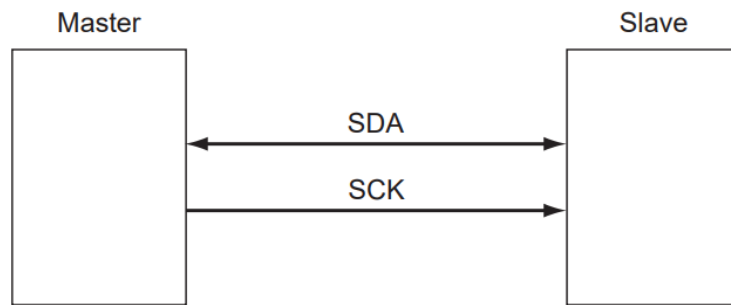
Komunikasi *Inter-Integrated Circuit* (I<sup>2</sup>C) merupakan system yang dikembangkan oleh Philips Semiconductor untuk menyediakan komunikasi antara perangkat IC yang kemudian digunakan secara luas dan diadopsi oleh perusahaan semikonduktor lainnya. I<sup>2</sup>C ideal digunakan untuk menyambungkan peripheral ke sebuah *motherboard* atau system tertanam atau apapun yang menggunakan komunikasi pada jarak pendek. Perangkat I<sup>2</sup>C hanya menggunakan 2 pin yaitu SCL (Serial Clock) yang mensinkronisasikan transfer data antara 2 chip dan SDA (Serial Data).



Gambar 4. Data Bus I<sup>2</sup>C

[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

Dengan hanya menggunakan 2 kabel seperti ini maka data tidak dapat mengalir secara bersamaan dari *master* ke *slave* dan dari *slave* ke *master* seperti pada SPI. Melainkan data akan mengalir dari *master* ke *slave* (ketika komunikasi telah selesai) barulah antara *slave* ke *master* jika memang diperlukan. Data akan mengalir kembali ke *master* jika *slave* merespon data sesuai permintaan *master*.

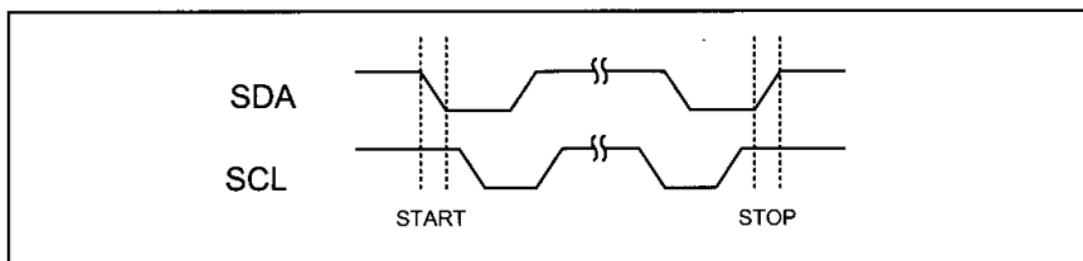


Gambar 5. Data Flow I<sup>2</sup>C

[Barnett, R. H., Cox, S., & O'Cull, L. (2006). *Embedded C programming and the Atmel AVR*. Thomson Delmar Learning.]

### Kondisi START dan STOP

Seperti yang telah diketahui bahwa I<sup>2</sup>C merupakan protocol komunikasi dengan orientasi koneksi, sehingga setiap transmisi data akan dimulai dengan kondisi START dan diakhiri dengan kondisi STOP. Kondisi START dan STOP dibuat berdasarkan perubahan level logika antara SCL dan SDA. Untuk kondisi START dibangkitkan oleh perubahan logika *high* ke *low* pada SDA ketika SCL berlogika *high*. Kondisi STOP dibangkitkan oleh perubahan logika *low* ke *high* pada SDA ketika SCL berlogika *low*.

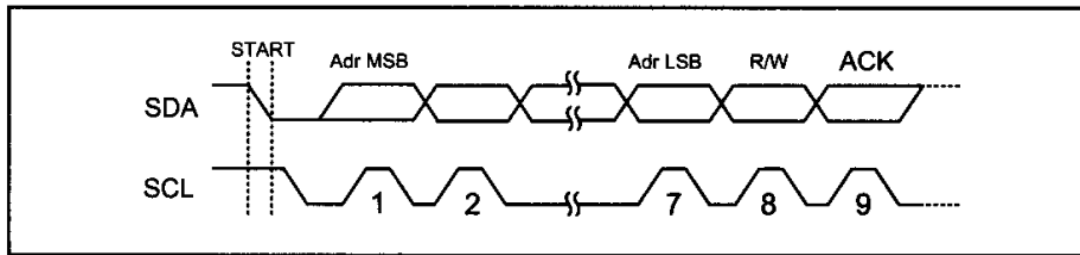


Gambar 6. Kondisi START dan STOP

[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

### Format Paket Alamat

Sama seperti pada paket lainnya, semua paket alamat yang ditransmisikan pada I<sup>2</sup>C memiliki panjang 9-bit dengan 7-bit sebagai alamat paket, satu kontrol READ/WRITE, dan satu untuk *Acknowledge Bit*.



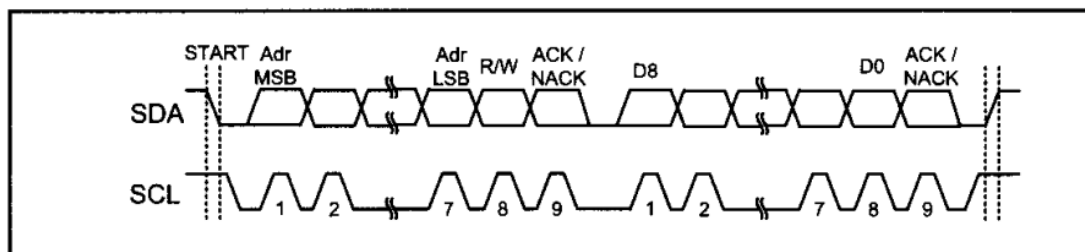
Gambar 7. Format Paket Alamat pada I<sup>2</sup>C

[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

### Format Paket Data

Seperti pada paket data menggunakan metode lainnya, paket data pada I<sup>2</sup>C memiliki panjang 9-bit, dimana 8-bit pertama merupakan data yang akan ditransmisikan dan bit ke 9 merupakan ACK (Acknowledge Bit). Jika *receiver* telah menerima bit data terakhir dan tidak ada data lagi yang akan diterima atau *receiver* tidak dapat menerima atau memproses data lagi, maka *receiver* akan mengirimkan NACK (Not Acknowledge Bit) dengan menyalakan SDA pada logika *high*.

### Gabungan Alamat dan Format Paket Data Menjadi Sebuah Transmisi



Gambar 8. Contoh Transmisi Data

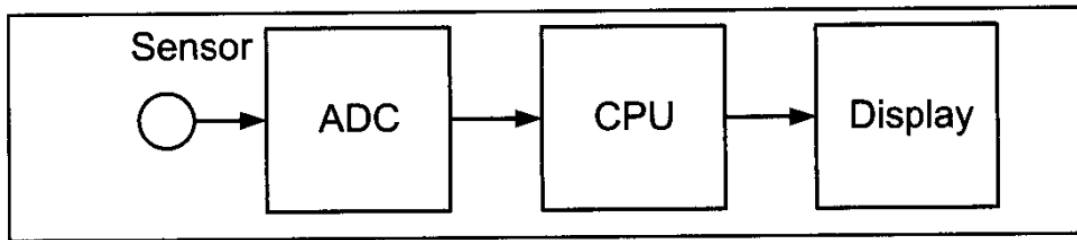
[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

### ADC Converter

Konverter analog ke digital merupakan perangkat yang sangat luas dan penting penggunaannya dalam bidang akuisisi data. Komputer digital menggunakan bilangan biner (nilai diskrit) dalam pengolahan datanya, sedangkan pada dunia fisik yang nyata seluruh besaran yang digunakan oleh alam merupakan analog (nilai kontinu) seperti temperatur, perubahan kecepatan, gelombang suara, dan



kuantitas fisis lainnya. Untuk menangkap segala macam bentuk data ini, maka sinyal dari alam dikonversi menuju sinyal elektrik dengan bantuan *transducer* atau sensor.

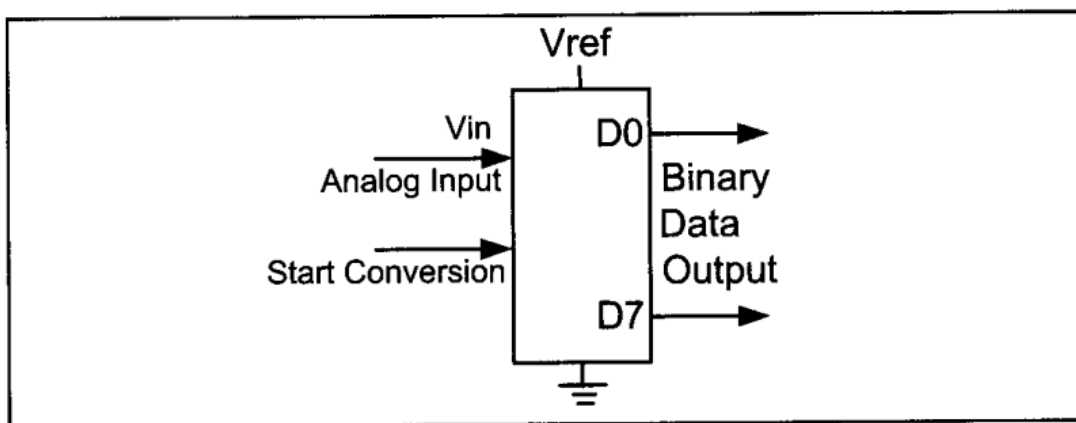


Gambar 9. Koneksi Mikrokontroller ke Sensor melalui ADC

[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

- ***Sampling dan Resolution***

Proses pengonversian dari sinyal analog ke digital disebut dengan *sampling* atau *quantization*. Dimana ADC memiliki dua karakteristik utama, yaitu *sample rate* dan *resolution*. *Sample rate* dinyatakan dengan besaran *sample per seconds* (SPS) dan merepresentasikan seberapa banyak sebuah input sinyal analog dikonversikan kedalam kode digital. Sedangkan *resolution* menyatakan akurasi dari setiap *sample* yang diambil. Sebuah konverter 8-bit dapat mengonversi output data dari 0 hingga  $2^8 - 1$  atau sebanyak 255 data.



Gambar 10. Block Diagram ADC 8-bit

[Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall]

ADC memiliki variasi rentang tegangan input dan rentang tegangan output. Dimana pada rentang tegangan output biasanya dinyatakan dengan bit. Nilai bit pada output menentukan rentang nilai yang dapat dibaca dari output konverter. Nilai faktor perbandingan atau faktor konversi dinyatakan dengan persamaan berikut:

$$\frac{V_{in}}{V_{fullscale}} = \frac{x}{2^n - 1}$$

Dimana dari persamaan diatas nilai “x” merupakan output digital dan “n” merupakan nilai bit pada output digital. Sedangkan nilai *resolution* dinyatakan dengan persamaan:

$$V_{resolution} = \frac{V_{fullscale}}{2^n - 1}$$

- **$V_{ref}$**

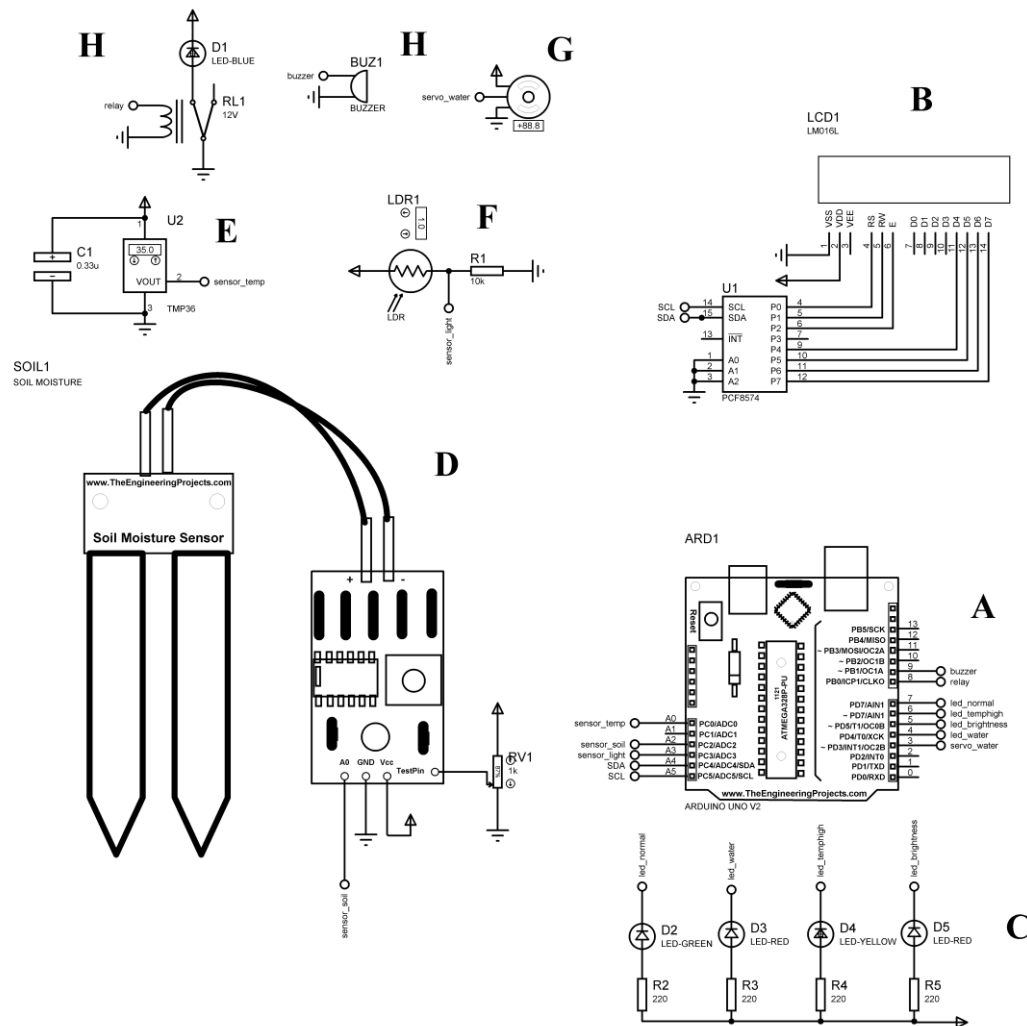
$V_{ref}$  atau tegangan referensi merupakan tegangan input yang digunakan untuk mereferensikan tegangan output yang biasanya terdapat pada lembar fabrikasi yang telah tertera. Dimana tegangan yang terhubung dengan pin ini bersama dengan *resolution* dari chip ADC akan menyatakan *step size*. Sebagai contoh untuk ADC 8-bit, maka *step size*-nya adalah  $V_{ref}/256$  dikarenakan  $2^8=256$ .

- **Output Data Digital**

Pada ADC 8-bit output data digital dinyatakan dengan pin output D0-D7, dan untuk ADC 10-bit dinyatakan dengan pin output D0-D9. Untuk menghitung tegangan output maka digunakan persamaan sebagai berikut:

$$D_{out} = \frac{V_{in}}{step\ size}$$

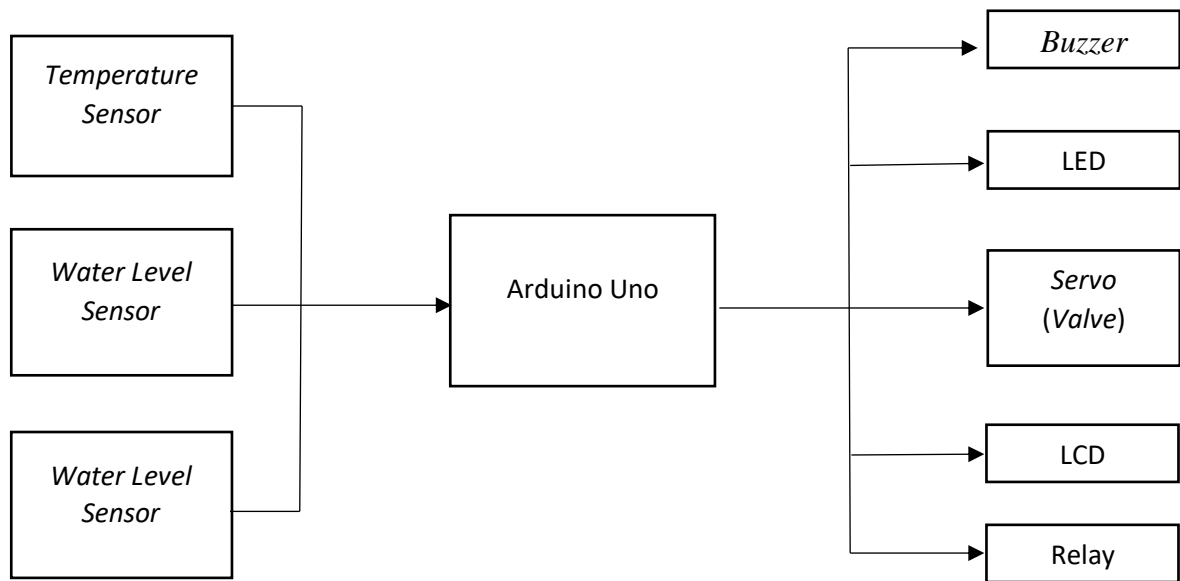
## MODEL RANGKAIAN



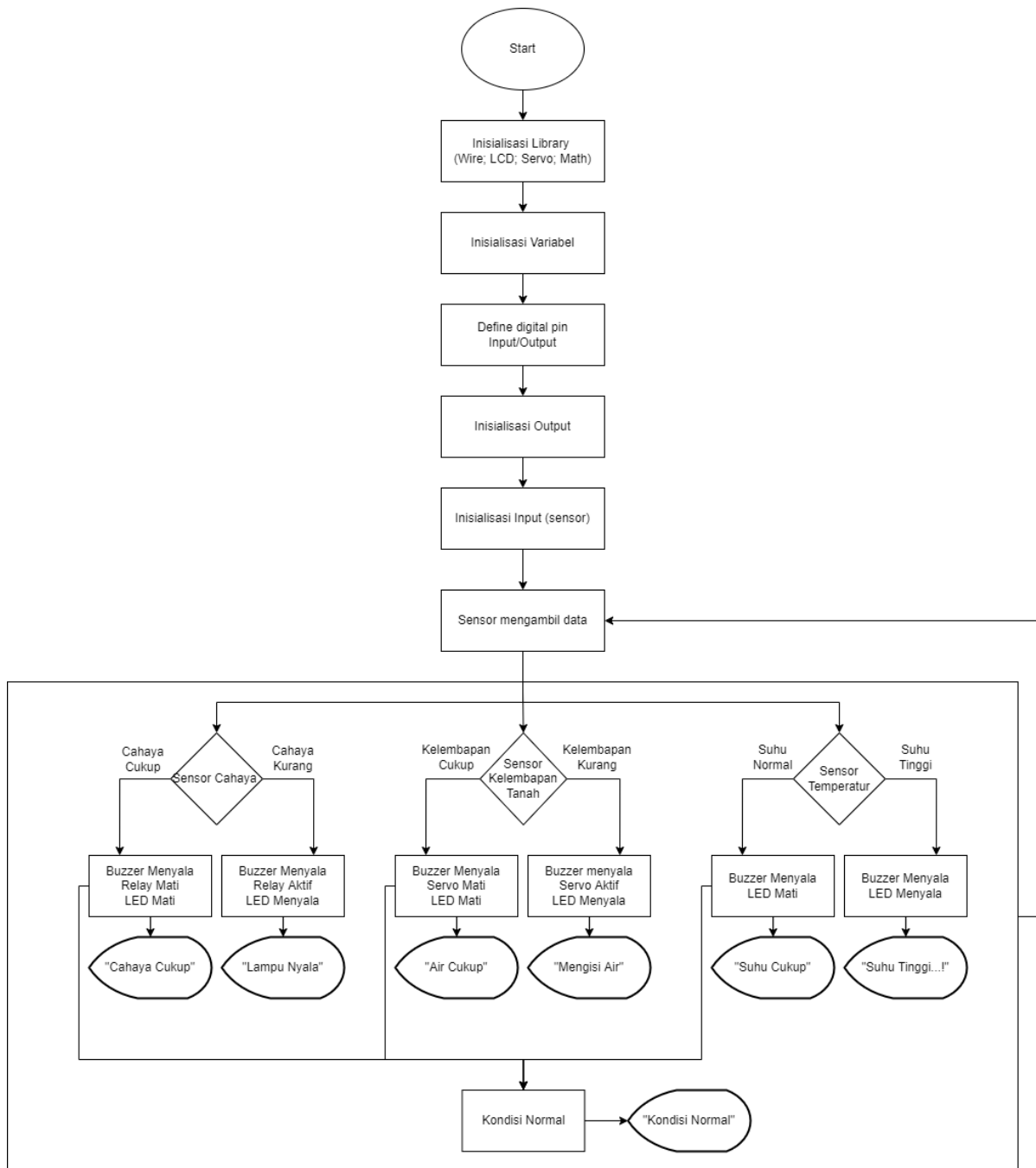
Keterangan:

- A. Arduino Uno R3
- B. LCD I2C
- C. LED Monitoring
- D. Soil Moisture Sensor FC-28
- E. Temperature Sensor TMP36
- F. LDR
- G. Servo
- H. Buzzer
- I. Relay
- J. Baterai 5V

## DIAGRAM BLOK



## DIAGRAM ALUR



## **CARA KERJA**

Dalam proyek *Smart Green House Regulator* mula-mula Arduino akan menjalankan pengaturan awal berupa memasukkan *library*, mendefinisikan port input dan output analog serta menampilkan animasi awal berisi nama praktikan dan judul proyek pada LCD, kemudian proses logika akan berjalan dengan membaca data dari tiga sensor, yaitu; sensor cahaya, sensor kelembapan tanah, dan sensor suhu yang ditampilkan secara terus-menerus dan diperbaharui secara berkala. Jika sensor temperatur mendeteksi suhu tinggi, maka akan menampilkan pesan pada LCD dan menyalakan indikator LED suhu tinggi beserta buzzer yang akan berbunyi. Jika sensor kelembapan tanah mendeteksi nilai kelembapan yang ditentukan, maka akan menampilkan pesan pada LCD dan menyalakan katup untuk mengalirkan air yang digerakkan oleh servo, kemudian indikator LED air menyala akan aktif. Jika sensor cahaya mendeteksi cahaya kurang dari yang telah ditentukan, maka akan menampilkan pesan pada LCD dan menyalakan saklar lampu yang dinyatakan dengan relay, kemudian LED lampu aktif akan menyala. Jika nilai pembacaan dari ketiga sensor pada keadaan normal, maka akan menampilkan pada LCD dan indikator LED normal akan menyala.

## **HASIL UJI**

Proyek *Smart Green House Regulator* dengan Arduino Uno mendapatkan hasil yang sesuai dengan pengaturan program yang dibuat, LDR yang digunakan sebagai sensor cahaya berkerja dengan baik, Saat kecerahan kurang maka buzzer akan berbunyi dan LCD menunjukan teks “Lampu Nyala”, Sensor lainnya yang digunakan adalah sensor Suhu TMP36 yang berkerja dengan baik dalam mengukur suhu ruangan, Saat suhu bernilai lebih dari yang diatur pada program, maka LCD akan menunjukan “Suhu Tinggi” dan buzzer akan berbunyi. Sensor lain yang digunakan adalah FC-28 yaitu sensor kelembapan tanah, saat sensor mendeteksi kurangnya kelembapan maka buzzer akan berbunyi dan LCD menunjukan “Mengisi Air”. Semua teks yang muncul pada LCD merupakan indikator proses yang sedang berjalan. Saat adanya dua atau lebih sensor yang mendeteksi masalah, maka tulisan pada LCD akan menunjukan teks secara bergantian. Saat semua sensor mendeteksi tidak adanya masalah, maka program akan masuk ke kondisi Normal yang juga tertulis pada LCD.

## ANALISIS

```
//Define Library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <math.h>

LiquidCrystal_I2C lcd(0x27,20,4);

//Analog pin input
const int sensor_temp = A0;
const int sensor_soil = A2;
const int sensor_light = A3;

//Digital pin output
#define servo_water 3
#define led_water 4
#define led_light 5
#define led_temphigh 6
#define led_normal 7
#define relay 8
#define buzzer 9
```

Pada baris kode ini merupakan definisi *library* yang digunakan untuk percobaan kali ini dengan *library* Wire.h untuk komunikasi I2C, LiquidCrystal\_I2C.h untuk menggerakkan LCD I2C, Servo.h untuk menggerakkan servo, math.h untuk melakukan operasi matematika. Kemudian mendefinisikan pin analog sebagai input pembacaan sensor dan pindigital sebagai output indikator dan input pembacaan indikator.

```
// Create a new servo object:
Servo myservo;

// Create a variable to store the servo position:
```

```
int angle = 0;
```

```
// Create char for LCD
```

```
char text_kelompok[] = "Kelompok: 1  ";
```

```
char nama1[] = "Gesha M.C.  ";
```

```
char nama2[] = "Yonathan L.  ";
```

```
char nama3[] = "Umar A. A.  ";
```

Pada baris kode ini mendefinisikan objek Servo dan inisialisasi sudut servo yang akan dijalankan ketika mikrokontroler baru menyala, kemudian mendefinisikan karakter text dan nama kelompok yang akan ditampilkan pada LCD sebagai pesan pertama.

```
void setup()
```

```
{
```

```
myservo.attach(servo_water);
```

```
Serial.begin(9600);
```

```
//initialize pin
```

```
pinMode(led_water, OUTPUT);
```

```
pinMode(led_light, OUTPUT);
```

```
pinMode(led_temphigh, OUTPUT);
```

```
pinMode(led_normal, OUTPUT);
```

```
pinMode(buzzer, OUTPUT);
```

```
pinMode(relay, OUTPUT);
```

```
//initialize output pin
```

```
digitalWrite(led_water, HIGH);
```

```
digitalWrite(led_light, HIGH);
```

```
digitalWrite(led_temphigh, HIGH);
```

```
digitalWrite(led_normal, HIGH);
```

```
digitalWrite(relay, LOW);
```

```
noTone(buzzer);
```



```
myservo.write(0);
```

```
// initialize the lcd
```

```
lcd.init();
```

```
lcd.backlight();
```

```
lcd.setCursor(0, 0);
```

```
lcd.print(text_kelompok);
```

```
delay(500);
```

```
//Display opening animation
```

```
lcd.setCursor(0, 1);
```

```
for ( int positionCounter = 0; positionCounter < 16; positionCounter++)
```

```
{
```

```
    lcd.print(nama1[positionCounter]);
```

```
    delay(150);
```

```
}
```

```
lcd.setCursor(0, 1);
```

```
for ( int positionCounter = 0; positionCounter < 16; positionCounter++)
```

```
{
```

```
    lcd.print(nama2[positionCounter]);
```

```
    delay(150);
```

```
}
```

```
lcd.setCursor(0, 1);
```

```
for ( int positionCounter = 0; positionCounter < 16; positionCounter++)
```

```
{
```

```
    lcd.print(nama3[positionCounter]);
```

```
    delay(150);
```

```
}
```

```
delay(500);
```

```
lcd.clear();
```

```
//display project tittle
```

```

lcd.setCursor(5,0);
lcd.print("Smart");
lcd.setCursor(2,1);
lcd.print("Green House");
delay(3000);
lcd.clear();
}

```

Pada baris kode ini merupakan berbagai macam perintah yang dijalankan pertama kalinya atau inisialisasi mikrokontroler yang berisi pengaktifan servo yang telah didefinisikan, mendefinisikan *baud rate* sebesar 9600 yang digunakan sebagai komunikasi serial, mendefinisikan pin digital dan menuliskan inout/output untuk pertama kali, dan menyetel lcd dengan mengaktifkan lcd dan mengatur backlightnya kemudian menampilkan animasi nama kelompok beserta nama-nama praktikan yang diakhiri dengan judul proyek yang dibuat.

```

void normal ()
{
  digitalWrite(led_normal, LOW);
  digitalWrite(led_water, HIGH);
  digitalWrite(led_light, HIGH);
  digitalWrite(led_temphigh, HIGH);
  noTone(buzzer);
  digitalWrite(relay, LOW);
  myservo.write(0);
}

```

```

float read_temperature (void)
{
  //temp sensor reading
  int reading_temp = analogRead(sensor_temp);
  float voltage_temp = reading_temp * 5.0;
  voltage_temp /= 1024.0;
}

```

```

float temperature = (voltage_temp - 0.5) * 100 ;

lcd.setCursor(0,0);
lcd.print(temperature);
lcd.setCursor(5,0);
lcd.print((char)223);
lcd.setCursor(6,0);
lcd.print("C");
return(temperature);
}

```

Pada baris kode ini merupakan fungsi pembacaan data dari sensor temperatur TMP36, dimana mula-mula tegangan analog yang dihasilkan oleh sensor akan dibaca dan dikonversikan menjadi suhu dengan satuan derajat celcius yang menggunakan pin analog dengan sumber tegangan 5,5 V dengan rentang data 1024 bit. Kemudian data temperatur yang diperoleh akan ditampilkan pada LCD baris 0.

```

int read_soil_moist()
{
  //Soil moisture sensor reading
  int reading_soil= analogRead(sensor_soil);
  int soil = map(reading_soil,550,250,0,100);

  lcd.setCursor(8,0);
  lcd.print(soil);
  lcd.setCursor(11,0);
  lcd.print("%");
  return(soil);
}

```

Pada baris kode ini merupakan fungsi pembacaan sensor kelembapan tanah dari sensor FC-28, mula-mula tegangan analog yang dihasilkan oleh sensor akan dibaca oleh mikrokontroller, kemudian tegangan analog yang dibaca akan dilakukan *mapping* atau

pengubahan dengan rasio yang telah didapatkan pada pengambilan pembacaan data menjadi data persentase kelembapan tanah. Nilai yang diperoleh akan ditampilkan pada LCD baris 0.

```
int read_brightness()
{
    int light= analogRead(sensor_light);

    lcd.setCursor(13,0);
    lcd.print(light);
    return(light);

}
```

Pada baris kode ini merupakan pembacaan cahaya yang didapat dari tegangan LDR dengan tanpa pengonversian lebih lanjut, hasil data yang diperoleh ditampilkan pada LCD pada baris ke 0.

```
void mode_normal()
{
    if (digitalRead(led_normal) == HIGH)
    {
        tone(buzzer, 1000);
        delay(250);
        noTone(buzzer);
        delay(250);
        tone(buzzer, 1000);
        delay(250);
        noTone(buzzer);
        tone(buzzer, 1000);
        delay(250);
        noTone(buzzer);
    }
}
```

```
    else
    {
        noTone(buzzer);
    }
}
```

```
void mode_water_on()
{
    if (digitalRead(led_water) == HIGH)
    {
        lcd.setCursor(0,1);
        lcd.print(""); // buat clear
        lcd.print(" Mengisi Air ");
        tone(buzzer, 630);
        delay(250);
        tone(buzzer, 800);
        delay(250);
        tone(buzzer, 750);
        delay(250);
        noTone(buzzer);
    }
    else
    {
        noTone(buzzer);
    }
}
```

```
void mode_water_off()
{
    if (digitalRead(led_water) == LOW)
    {
        lcd.setCursor(0,1);
```

```

    lcd.print("");
    lcd.print(" Air Cukup ");
    tone(buzzer, 698);
    delay(250);
    tone(buzzer, 830);
    delay(250);
    tone(buzzer, 783);
    delay(250);
    noTone(buzzer);
}
else
{
    noTone(buzzer);
}
}

void mode_light_on()
{
    if (digitalRead(led_light) == HIGH)
    {
        lcd.setCursor(0,1);
        lcd.print("");
        lcd.print(" Lampu Nyala ");
        tone(buzzer, 1000);
        delay(250);
        tone(buzzer, 600);
        delay(250);
        tone(buzzer, 1000);
        delay(250);
        noTone(buzzer);
    }
    else

```

```
{  
    noTone(buzzer);  
}  
}
```

```
void mode_light_off()  
{  
    if (digitalRead(led_light) == LOW)  
    {  
        lcd.setCursor(0,1);  
        lcd.print("");  
        lcd.print(" Cahaya Cukup ");  
        tone(buzzer, 987);  
        delay(250);  
        tone(buzzer, 783);  
        delay(250);  
        tone(buzzer, 880);  
        delay(250);  
        noTone(buzzer);  
    }  
    else  
    {  
        noTone(buzzer);  
    }  
}
```

```
void mode_temp_high()  
{  
    if (digitalRead(led_temphigh) == HIGH)  
    {  
        lcd.setCursor(0,1);  
        lcd.print("");
```

```

    lcd.print("Suhu Tinggi...! ");
    tone(buzzer, 900);
    delay(250);
    tone(buzzer, 600);
    delay(250);
    tone(buzzer, 800);
    delay(250);
    noTone(buzzer);
}
else
{
    noTone(buzzer);
}
}

void mode_temp_ok()
{
    if (digitalRead(led_temphigh) == LOW)
    {
        lcd.setCursor(0,1);
        lcd.print("");
        lcd.print(" Suhu Cukup ");
        tone(buzzer, 698);
        delay(250);
        tone(buzzer, 1396);
        delay(250);
        tone(buzzer, 880);
        delay(250);
        noTone(buzzer);
    }
    else
    {

```



```

    noTone(buzzer);
  }
}

```

Pada baris kode ini merupakan fungsi kondisional yang akan dijalankan ketika fungsi void loop() memanggil fungsi-fungsi ini yang berisikan indikator berupa pesan yang ditampilkan pada LCD dan suara buzzer yang bervariasi pada setiap kasus yang bergantung pada LED indikator. Jika LED indikator telah menyala, maka fungsi ini tidak perlu dikerjakan kembali dengan menambahkan fungsi else yang kosong.

```

void loop()
{
  lcd.clear();

  int soil = read_soil_moist();
  int light = read_brightness();
  float temperature = read_temperature();

  if (soil >= 40 && light >= 300 && temperature <= 35)    //normal
  {
    lcd.setCursor(0,1);
    lcd.print("");
    lcd.print("  NORMAL  ");
    mode_normal();
    normal();
    delay(1500);
  }
  else
  {
    digitalWrite(led_normal, HIGH);
    if (soil < 40)
    {

```

```
mode_water_on();
myservo.write(90);
digitalWrite(led_water, LOW);
delay(1500);
}
else
{
myservo.write(0);
mode_water_off();
digitalWrite(led_water, HIGH);
delay(1500);
}

if (light < 300)
{
mode_light_on();
digitalWrite(relay, HIGH);
digitalWrite(led_light, LOW);
delay(1500);
}
else
{
mode_light_off();
digitalWrite(led_light, HIGH);
digitalWrite(relay, LOW);
delay(1500);
}

if (temperature > 35)
{
mode_temp_high();
digitalWrite(led_temphigh, LOW);
```

```

    delay(1500);
}
else
{
    mode_temp_ok();
    digitalWrite(led_temphigh, HIGH);
    delay(1500);
}
}
delay(1000);
}

```

Pada baris kode ini merupakan fungsi utama yang akan terus-menerus dijalankan yang akan mengecek pembacaan sensor dengan memanggil fungsi-fungsi konversi dan pembacaan nilai tegangan analog yang telah didefinisikan sebelumnya, ketika nilai pembacaan sensor tidak sesuai dengan batas yang ditentukan, maka dari fungsi ini akan memanggil fungsi-fungsi yang sesuai dengan kondisi fungsi yang didefinisikan sebelumnya kemudian menyalakan output yang sesuai. Ketika tanah kurang lembap, maka servo yang bertindak sebagai katup akan menyala kemudian mikrokontroler menampilkan pesan pada LCD serta menyalakan indikator LED dan sebaliknya. Kemudian ketika cahaya kurang, maka relay akan mengarahkan saklar pada posisi menyala yang bertindak untuk memnyalakan jaringan kelistrikan lampu kemudian mikrokontroler menampilkan pesan pada LCD serta menyalakan indikator LED dan sebaliknya. Ketika temperatur tinggi hanya akan dikeluarkan pesan pada LCD kemudian mikrokontroler menampilkan pesan pada LCD serta menyalakan indikator LED tanpa output apapun dan sebaliknya. Dan ketika semua kondisi terpenuhi yaitu mikrokontroler akan masuk mode normal dimana semua output indikator LED akan mati dan semua aktuator akan kembali pada keadaan awal.

Proyek Smart Green House Regulator dengan Arduino Uno mendapatkan hasil yang sesuai dengan pengaturan program yang dibuat, LDR yang digunakan sebagai sensor cahaya berkerja dengan baik, dengan menghubungkan komponen analog ini ke ADC hingga didapatkan nilai digital yang kemudian dimanfaatkan dalam perhitungan dan deteksi kecerahan, Sensor lainnya yang digunakan adalah sensor Suhu TMP36 yang berkerja dengan

baik dalam mengukur suhu ruangan, pada konfigurasi Sensor suhu ini terdapat masalah yaitu arus yang masuk tidak stabil yang menyebabkan pengukuran suhu yang salah, hal ini diselesaikan dengan memisahkan sumber tegangan komponen yang memerlukan tegangan yang besar seperti LCD, Relay, dan Servo dengan sumber tegangan eksternal sebesar 6V dengan komponen yang memerlukan tegangan kecil seperti sensor TMP36 dan LDR dengan sumber tegangan arduino pin 3.3V dan memasang kapasitor sebesar 0.33 $\mu$ F untuk mengantisipasi lonjakan arus khusus untuk sensor TMP36. Sensor lain yang digunakan adalah FC-28 yaitu sensor kelembapan tanah, sensor ini diuji cobakan tidak langsung pada tanah yang lembap, melainkan pada air, hasil dari sensor ini juga sesuai dengan pengaturan program yang dibuat dimana jika sensor tidak mendeteksi kelembapan, maka LED akan menyala dan servo akan berputar sebesar 90° yang menandakan kran air terbuka.

## KESIMPULAN

- Sistem *Smart Green House Regulator* dapat direalisasikan menggunakan modul Arduino UNO R3 dengan microprocessor Atmega328P, Sensor Temperatur TMP36, Sensor Kelembapan Tanah FC-28 dan LDR sebagai Sensor Cahaya.
- Diperlukan pemisahan sumber tegangan dan arus untuk komponen yang memerlukan tegangan dan arus tinggi dengan komponen yang memerlukan tegangan dan arus yang relatif rendah.
- Hasil percobaan sesuai dengan yang program yang dibuat

## REFERENSI

1. Mazidi, M. Ali; Naimi, Samad; Naimi, Sepehr (2011). *The AVR Microcontroller and Embedded Systems using Assembly and C*. Prentice Hall.
2. *In-Depth: How Soil Moisture Sensor Works and Interface it with Arduino*. (2019). Retrieved 21 December 2021, from <https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/>
3. *Using An LDR Sensor With Arduino*. (2021). Retrieved 21 December 2021, from <https://create.arduino.cc/projecthub/tarantula3/using-an-ldr-sensor-with-arduino-807b1c>
4. *TMP36 Temperature Sensor*. (2021). Retrieved 21 December 2021, from <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>

5. *Low Voltage Temperature Sensors TMP35/TMP36/TMP37* (2021). Retrieved 22 December 2021, from [https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35\\_36\\_37.pdf](https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf)
6. *FC-28 Soil Moisture Sensor* (2021). Retrieved 22 December 2021, from <https://cdn.hackaday.io/files/2734811700568>
7. *Arduino UNO R3* (2021). Retrieved 22 December 2021, from <https://docs.arduino.cc/hardware/uno-rev3>
8. *ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash* (2021). Retrieved 22 December 2021, from [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
9. *Light Dependent Resistor LDR: Photoresistor* (2021). Retrieved 22 December 2021, from [https://www.electronics-notes.com/articles/electronic\\_components/resistors/light-dependent-resistor-ldr.php](https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php)
10. *Pengertian LDR (Light Dependent Resistor) dan Cara Mengukurnya* (2021). Retrieved 22 December 2021, from <https://teknikelektronika.com/pengertian-ldr-light-dependent-resistor-cara-mengukur-ldr/>