



**CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT  
DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITAS INDONESIA**

**AUTOMATED GREENHOUSE**

**GROUP B2**

|                                       |                   |
|---------------------------------------|-------------------|
| <b>Handaneswari Pramudhyta Imanda</b> | <b>2106731346</b> |
| <b>Luthfi Misbachul Munir</b>         | <b>2106631961</b> |
| <b>Prima Shalih</b>                   | <b>2106636862</b> |
| <b>Zefanya Christira Deardo</b>       | <b>2106637214</b> |

## **Kata Pengantar**

Puji dan syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan proyek akhir Praktikum. Tidak lupa kami ucapkan terima kasih kepada para asisten Laboratorium Digital yang membantu kami dalam mengerjakan proyek akhir ini, terutama kepada bang Naufal Faza sebagai pendamping kami dalam mengerjakan proyek Sistem Siber Fisik yang berjudul “*Automated Greenhouse*”.

Laporan ini berisi tentang hasil pengimplementasian Assembly language pada alat yang kami buat. Selain itu, laporan ini juga mencakup cara kerja, pengujian, dan analisis selama percobaan pengimplementasiannya.

Perlu disadari bahwa laporan proyek akhir ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran dari berbagai pihak sangat kami harapkan agar laporan ini dapat menjadi lebih baik.

Terakhir, kami berharap laporan proyek akhir yang kami buat dapat bermanfaat bagi banyak orang, serta juga bisa menjadi referensi yang berguna bagi kemajuan teknologi di masa mendatang.

Depok , 16 May 2023

Kelompok B2

## TABLE OF CONTENTS

|  |           |
|--|-----------|
| <b>BAB 1.....</b>                          | <b>4</b>  |
| <b>PENDAHULUAN.....</b>                    | <b>4</b>  |
| 1.1 LATAR BELAKANG.....                    | 4         |
| 1.2 PROPOSED SOLUTION.....                 | 5         |
| 1.3 ACCEPTANCE CRITERIA.....               | 6         |
| 1.4 ROLES AND RESPONSIBILITIES.....        | 6         |
| 1.5 TIMELINE AND MILESTONES.....           | 7         |
| <b>CHAPTER 2.....</b>                      | <b>8</b>  |
| <b>IMPLEMENTATION.....</b>                 | <b>8</b>  |
| 2.1 HARDWARE DESIGN AND SCHEMATIC.....     | 8         |
| 2.2 SOFTWARE DEVELOPMENT.....              | 9         |
| 2.3 HARDWARE AND SOFTWARE INTEGRATION..... | 24        |
| <b>CHAPTER 3.....</b>                      | <b>29</b> |
| <b>TESTING AND EVALUATION.....</b>         | <b>29</b> |
| 3.1 TESTING.....                           | 29        |
| 3.2 RESULT.....                            | 31        |
| 3.3 EVALUATION.....                        | 32        |
| <b>CHAPTER 4.....</b>                      | <b>33</b> |
| <b>CONCLUSION.....</b>                     | <b>33</b> |

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 LATAR BELAKANG**

Greenhouse atau rumah kaca telah menjadi solusi yang populer dalam pertanian modern. Greenhouse menyediakan lingkungan terkendali yang memungkinkan pertumbuhan tanaman yang optimal sepanjang tahun. Salah satu faktor yang mempengaruhi pertumbuhan tanaman di dalam rumah kaca adalah temperatur. Tanaman adalah organisme yang termoreaktif, artinya mereka merespons temperatur lingkungan dengan cara tertentu. Tanaman membutuhkan temperatur yang tepat agar dapat tumbuh dengan baik dan mencapai potensi pertumbuhan yang maksimal. Temperatur mempengaruhi aktivitas metabolismik dalam tanaman. Enzim-enzim yang terlibat dalam proses fotosintesis, respirasi, dan sintesis protein beroperasi pada temperatur tertentu yang optimal. Temperatur yang terlalu rendah atau terlalu tinggi dapat menghambat aktivitas enzim dan mempengaruhi proses metabolismik, yang pada gilirannya mempengaruhi pertumbuhan dan perkembangan tanaman. Oleh karena itu, perhatian yang cermat terhadap pengendalian temperatur dalam rumah kaca sangat penting.

Pengaturan temperatur yang baik dalam greenhouse memiliki beberapa manfaat penting. Pertama, temperatur yang tepat memungkinkan pengendalian yang lebih presisi terhadap pertumbuhan tanaman. Tanaman membutuhkan temperatur yang sesuai dengan kebutuhan fisiologis mereka untuk mencapai pertumbuhan yang optimal. Dengan mengatur temperatur dengan cermat, tanaman dapat tumbuh lebih cepat, menghasilkan buah dan bunga lebih baik, serta mencapai kualitas yang lebih baik. Kedua, pengaturan temperatur yang baik membantu mengurangi risiko penyakit dan hama. Beberapa patogen dan serangga memiliki preferensi temperatur tertentu untuk berkembang biak dan menyerang tanaman sehingga, dengan menjaga temperatur di dalam greenhouse pada tingkat yang tidak menguntungkan bagi patogen dan serangga ini, risiko investasi dan penyebaran penyakit dapat dikurangi. Ketiga, pengaturan temperatur yang baik juga membantu mengoptimalkan penggunaan sumber daya seperti air dan energi. Dengan menjaga temperatur di dalam greenhouse pada tingkat yang optimal, kehilangan air melalui penguapan dapat dikurangi. Selain itu, penggunaan perangkat pengatur temperatur yang efisien dan sistem pengendalian energi

dapat mengoptimalkan penggunaan energi, mengurangi biaya operasional, dan meningkatkan efisiensi energi.

## 1.2 PROPOSED SOLUTION

Dalam greenhouse tradisional, pengendalian temperatur sering kali menjadi tantangan. Faktor lingkungan eksternal seperti temperatur udara, sinar matahari, kelembaban, dan angin dapat memiliki efek langsung pada temperatur di dalam greenhouse. Selain itu, variasi temperatur yang signifikan dapat terjadi antara siang dan malam hari, serta musim panas dan musim dingin. Fluktuasi temperatur yang tidak terkendali dapat mengganggu pertumbuhan tanaman dan mengakibatkan hasil panen yang kurang optimal.

Untuk mengatasi permasalahan tersebut, kelompok kami membuat sebuah alat untuk mengatasi masalah pengendalian temperatur dalam rumah kaca. Automated greenhouse atau greenhouse otomatis adalah suatu sistem yang menggunakan teknologi sensor dan pengendalian otomatis untuk menjaga temperatur yang ideal di dalam rumah kaca. Alat ini menggunakan sensor temperatur untuk terus memantau temperatur di dalam greenhouse secara real-time. Setelah mendapatkan data temperatur aktual, alat automated greenhouse akan membandingkannya dengan temperatur yang telah ditentukan sebelumnya sebagai temperatur yang diinginkan. Jika temperatur di dalam greenhouse melebihi temperatur yang diinginkan, alat automated greenhouse akan mengambil tindakan untuk menyalakan kipas. Jika temperatur di dalam greenhouse terlalu rendah, alat automated greenhouse akan memberikan peringatan melalui lampu sebagai indikator bahwa temperatur sudah terlalu rendah. Hal ini menjadi peringatan bagi petani atau pengelola greenhouse bahwa temperatur di dalam greenhouse perlu ditindaklanjuti dengan tindakan penghangat seperti mengaktifkan sistem pemanas atau pemanas tambahan di dalam greenhouse.

Dengan adanya alat automated greenhouse, pengendalian temperatur di dalam greenhouse menjadi lebih efisien dan terkendali. Alat ini mengurangi ketergantungan pada intervensi manual dalam menjaga temperatur, sehingga petani atau pengelola greenhouse dapat fokus pada aspek lain dari budidaya tanaman. Penggunaan teknologi otomatis dalam automated greenhouse memberikan efisiensi dalam pengendalian temperatur, meningkatkan produktivitas pertanian, dan membantu tanaman tumbuh dengan maksimal tanpa harus terpengaruh oleh fluktuasi temperatur yang tidak diinginkan.

### **1.3 ACCEPTANCE CRITERIA**

Kriteria keberhasilan dari *Automated Greenhouse* :

1. DHT11 mampu mengambil data dari temperatur dan kelembapan yang ada
2. Temperatur yang diterima dapat ditampilkan pada module MAX7219
3. Ketika temperatur yang diterima oleh DHT11 lebih besar daripada yang diinginkan, maka servo sg90 akan berputar serta menyalaikan buzzer
4. Ketika temperatur yang diterima oleh DHT11 lebih kecil daripada yang diinginkan, maka LED akan menyala

### **1.4 ROLES AND RESPONSIBILITIES**

The roles and responsibilities assigned to the group members are as follows:

| Roles  | Responsibilities  | Person                            |
|--------|---|-----------------------------------|
| Role 1 | Mengerjakan kode servo, membantu membuat laporan, membuat readme, membuat flowchart, membantu troubleshooting kode  | Handaneswari<br>Pramudhyta Imanda |
| Role 2 | Membagi tugas tiap anggota kelompok untuk tiap komponen, menggabungkan seluruh kode baik secara logical maupun di github, melakukan testing serta troubleshooting serta | Luthfi Misbacul Munir             |

|        |   |                          |
|--------|---|--------------------------|
|        | membantu membuat laporan  |                          |
| Role 3 | Mempersiapkan peralatan yang dibutuhkan, melakukan testing proyek, dan mempersiapkan segala jenis berkas yang dibutuhkan. | Prima Shalih             |
| Role 4 | Mengerjakan kode dan rangkaian DHT + MAX7219, membantu menggabungkan keseluruhan kode                                     | Zefanya Christira Deardo |

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC

Dalam pengembangan desain *hardware* ini, kami menggunakan peralatan, seperti :

- Breadboard
- Sensor DHT11
- Arduino Uno
- Servo SG90
- Resistor
- Kabel Jumper
- LED
- Buzzer Active
- Layar MAX7219

Dengan alat-alat ini, kami mampu menciptakan suatu alat *Automated Greenhouse* sederhana. Berikut adalah sketsa dari rangkaian kami yang dibuat dengan memakai aplikasi Proteus dan rangkaian asli:

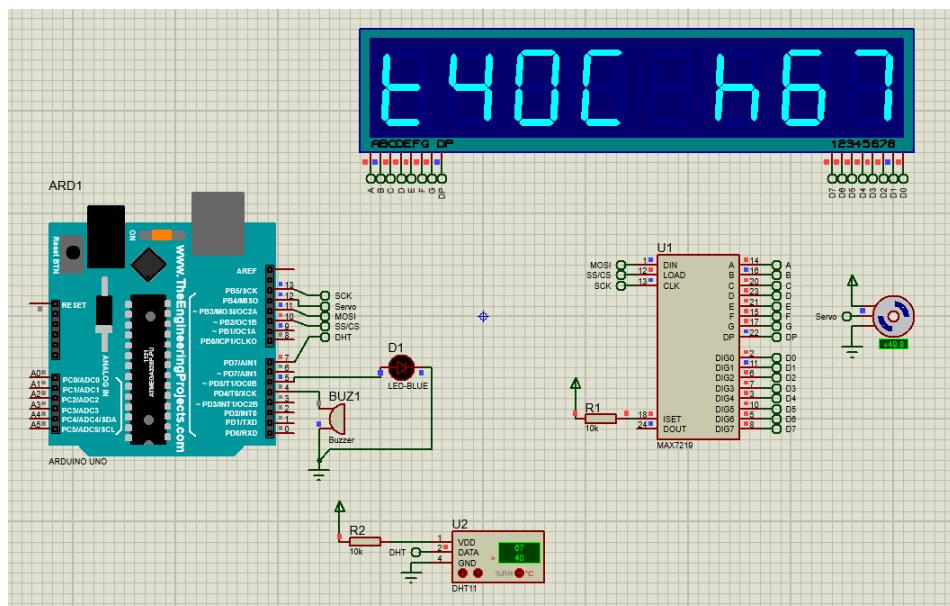


Figure 1. Rangkaian Proteus

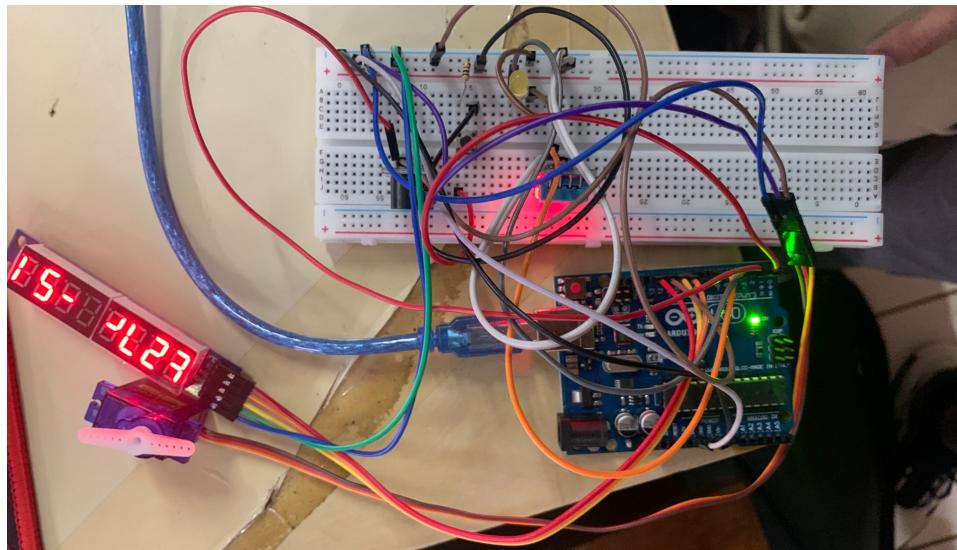


Figure 2. Rangkaian asli

## 2.2 SOFTWARE DEVELOPMENT

Pada pengembangan *software* ini, kami membuat kode yang sesuai dengan materi yang telah kami pelajari. Terdapat kode sensor untuk DHT11 dalam membaca temperatur, terdapat kode MAX7219 untuk menampilkan nilai data yang dibaca sensor baik berupa suhu maupun kelembaban. Terdapat kode servo sebagai asumsi kipas ketika suhu diatas batas tidak normal. Serta terdapat buzzer untuk suhu yang sangat tidak normal. Sebelum membuat kode program kami, pertama-tama kami membuat flowchart untuk menjadi gambaran dari jalannya program kami :

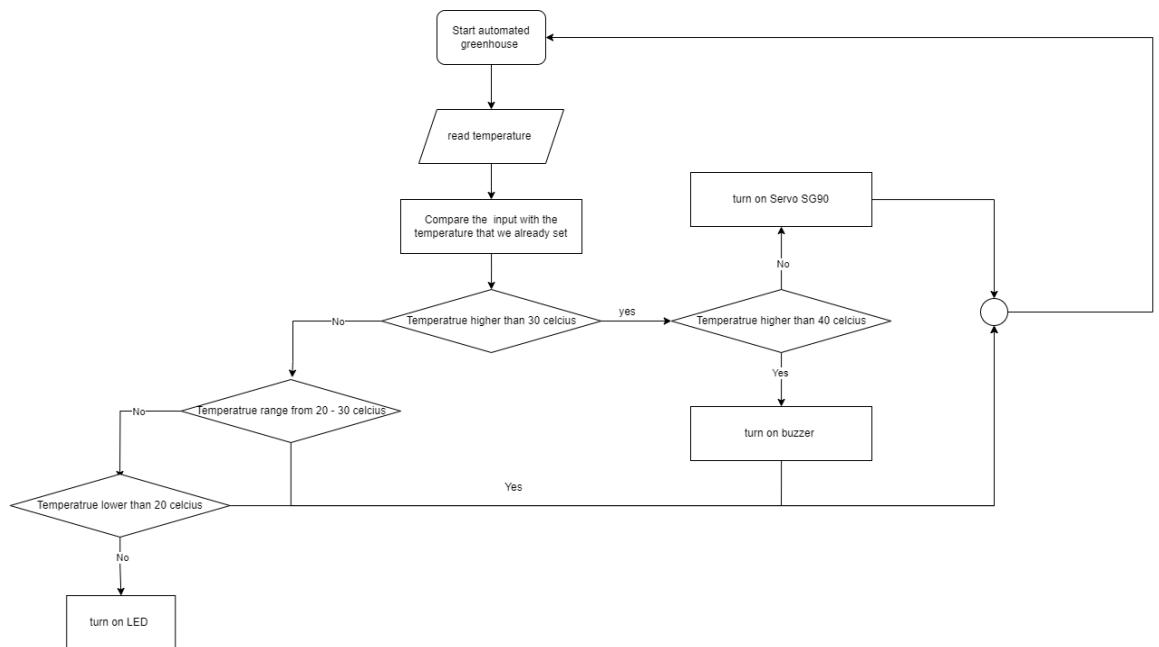


Figure 3. Flowchart

Pengujian pada alat dilakukan dengan merakit komponen dengan desain yang tertera di Figure 1 dan Figure 2 dan dijalankan dengan kode berikut :

```

;-----;
; Assembly Code
;-----;

#define __SFR_OFFSET 0x00

#include "avr/io.h"

;-----;

.global main

main:
    RCALL SPI_MAX7219_init
    RCALL MAX7219_disp_text
    agn:
        RCALL DHT11_sensor
        RCALL check
  
```

```

RJMP agn

;-----

DHT11_sensor:
;-----

RCALL delay_2s      ;wait 2s for DHT11 to get ready
;-----


;start_signal
;-----


    SBI    DDRD, 7          ;pin PD7 as o/p
    CBI    PORTD, 7         ;first, send low pulse
    RCALL delay_20ms       ;for 20ms
    SBI    PORTD, 7         ;then send high pulse
;-----


;response signal
;-----


    CBI    DDRD, 7          ;pin PD7 as i/p
w1: SBIC  PIND, 7
    RJMP  w1                 ;wait for DHT11 low pulse
w2: SBIS  PIND, 7
    RJMP  w2                 ;wait for DHT11 high pulse
w3: SBIC  PIND, 7
    RJMP  w3                 ;wait for DHT11 low pulse
;-----


    RCALL DHT11_reading ;read humidity (1st byte of 40-bit data)
    MOV    R25, R24
    RCALL DHT11_reading
    RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
;-----
```

```

;convert temp & humidity bytes to decimal & display on MAX7219

;-----

        MOV      R28,  R24          ;menyimpan nilai suhu di R28 untuk
;didisplay

        MOV      R31,  R24          ;menyimpan nilai suhu di R31 sebagai
;kondisi servo dan led

        LDI      R29,  0x07          ;MSD temp akan didisplay pada digit 6

        LDI      R30,  0x06          ;LSD temp akan didisplay pada digit 5

;-----

        RCALL binary2decimal    ;temp in decimal

;-----

        MOV      R28,  R25

        LDI      R29,  0x02          ;MSD humidity akan didisplay pada digit 1

        LDI      R30,  0x01          ;LSD humidity akan didisplay pada digit 0

;-----

        RCALL binary2decimal    ;humidity in decimal

        RET           ;kembali ke loop

;-----

;COMPARE THE OUTPUT

;-----

check:

        CPI      R31,  30          ;membandingkan nilai suhu pada R31
;dengan 30

        BRLO   skip_servo         ;jika suhu lebih kecil daripada 30,
;maka skip servo

        CBI      PORTD,  5          ;mematikan led

        RCALL servo_motor        ;menjalankan servo


skip_servo:

        CBI      PORTD,  4          ;mematikan led

```

```

        CPI    R31, 20          ;membandingkan nilai suhu R31 dengan 20

        BRGE   skip_led         ;jika suhu lebih besar daripada 20,
; maka skip led

        SBI    PORTD, 5          ;menyalakan led

        RET                 ;kembali ke looping

skip_led:

        CBI    PORTD, 5          ;mematikan led

        RET                 ;kembali ke looping

;=====

DHT11_reading:

        LDI    R16, 8           ;set counter for receiving 8 bits

        CLR    R24              ;clear data register

        ;-----

w4: SBIS  PIND, 7           ;if received bit=1, skip next inst

        RJMP   w4               ;detect data bit (high pulse)

        RCALL delay_timer0     ;wait 50us & then check bit value

        ;-----

        SBIS  PIND, 7           ;if received bit=1, skip next inst

        RJMP   skp              ;else, received bit=0, jump to skp

        SEC                ;set carry flag (C=1)

        ROL    R24              ;shift in 1 into LSB data register

        RJMP   w5               ;jump & wait for low pulse

skp:LSL   R24              ;shift in 0 into LSB data register

        ;-----

w5: SBIC  PIND, 7           ;wait for DHT11 low pulse

        RJMP   w5               ;jump & wait for low pulse

        ;-----

```

```
    DEC    R16          ;decrement counter
    BRNE   w4          ;go back & detect next bit
    RET           ;return to calling subroutine
;
;=====

;delay subroutines
;
;=====

delay_20ms:           ;delay 20ms
    LDI    R21, 255
    13: LDI    R22, 210
    14: LDI    R23, 2
    15: DEC    R23
    BRNE   15
    DEC    R22
    BRNE   14
    DEC    R21
    BRNE   13
    RET
;

delay_2s:             ;delay 2s
    LDI    R21, 255
    16: LDI    R22, 255
    17: LDI    R23, 164
    18: DEC    R23
    BRNE   18
    DEC    R22
    BRNE   17
    DEC    R21
    BRNE   16
    RET
```

```

;=====

delay_timer0:           ;50 usec delay via Timer 0

;-----

CLR    R20

OUT    TCNT0, R20        ;initialize timer0 with count=0

LDI    R20, 100

OUT    OCR0A, R20        ;OCR0 = 100

LDI    R20, 0b00001010

OUT    TCCR0B, R20        ;timer0: CTC mode, prescaler 64

;-----

12: IN     R20, TIFR0      ;get TIFR0 byte & check

SBRS   R20, OCF0A        ;if OCF0=1, skip next instruction

RJMP   12                  ;else, loop back & check OCF0 flag

;-----

CLR    R20

OUT    TCCR0B, R20        ;stop timer0

;-----

LDI    R20, (1<<OCF0A)

OUT    TIFR0, R20         ;clear OCF0 flag

RET

;=====

;MAX7219 subroutines

;=====

SPI_MAX7219_init:

;-----

SBI    DDRD, 4            ;pin PD4 as o/p buzzer

SBI    DDRD, 5            ;pin PD5 as o/p LED

.equ   SCK, 5

.equ   MOSI, 3

```

```

.eqv  SS, 2

;-----

LDI    R17, (1<<MOSI) | (1<<SCK) | (1<<SS)
OUT   DDRB, R17          ;set MOSI, SCK, SS as o/p
;-----


LDI    R17, (1<<SPIE) | (1<<SPE) | (1<<MSTR) | (1<<SPR0)
OUT   SPCR, R17          ;enable SPI as master, fsck=fosc/16
;-----


LDI    R17, 0x0A          ;set segment intensity (0 to 15)
LDI    R18, 8              ;intensity level = 8
RCALL send_bytes          ;send command & data to MAX7219
;-----


LDI    R17, 0x09          ;set decoding mode command
LDI    R18, 0b01100011 ;decoding byte
RCALL send_bytes          ;send command & data to MAX7219
;-----


LDI    R17, 0x0B          ;set scan limit command
LDI    R18, 0x07          ;8 digits connected to MAX7219
RCALL send_bytes          ;send command & data to MAX7219
;-----


LDI    R17, 0x0C          ;set turn ON/OFF command
LDI    R18, 0x01          ;turn ON MAX7219
RCALL send_bytes          ;send command & data to MAX7219
;-----


RET

;=====

MAX7219_disp_text:
;-----

```

```

LDI    R17, 0x08      ;select digit 7
LDI    R18, 0x0F      ;data = t (defg) (00001111)
RCALL send_bytes      ;send command & data to MAX7219
;-----
LDI    R17, 0x05      ;select digit 4
LDI    R18, 0x4E      ;data = C (adef) (01001110)
RCALL send_bytes      ;send command & data to MAX7219
;-----
LDI    R17, 0x04      ;select digit 3
LDI    R18, 0x00      ;data = space
RCALL send_bytes      ;send command & data to MAX7219
;-----
LDI    R17, 0x03      ;select digit 2
LDI    R18, 0x17      ;data = h (cefg) (00010111)
RCALL send_bytes      ;send command & data to MAX7219
;-----
RET
;=====
send_bytes:
CBI    PORTB, SS      ;enable slave device MAX7219
OUT    SPDR, R17      ;transmit command
;-----
l12:  IN    R19, SPSR
SBRS  R19, SPIF       ;wait for byte transmission
RJMP  l12             ;to complete
;-----
OUT    SPDR, R18      ;transmit data
;-----
l13:  IN    R19, SPSR

```

```

        SBRS  R19, SPIF          ;wait for byte transmission
        RJMP  113                 ;to complete
        ;-----
        SBI    PORTB, SS          ;disable slave device MAX7219
        RET
        ;=====
binary2decimal:
        ;-----
        CLR   R26                  ;set counter1, initial value 0
        CLR   R27                  ;set counter2, initial value 0
        ;-----
170:  CPI    R28, 100           ;compare R28 with 100
Ret:  BRMI  180                ;jump when R28 < 100
        INC   R26                  ;increment counter1 by 1
        SUBI  R28, 100             ;R28 = R28 - 100
        RJMP  170
        ;-----
180:  CPI    R28, 10            ;compare R28 with 10
        BRMI  dsp                 ;jump when R28 < 10
        INC   R27                  ;increment counter2 by 1
        SUBI  R28, 10              ;R28 = R28 - 10
        RJMP  180
        ;-----
dsp:  MOV   R18, R27
        MOV   R17, R29             ;select digit
        RCALL send_bytes          ;send command & data to MAX7219
        ;-----
        MOV   R18, R28
        MOV   R17, R30             ;select digit

```

```
        RCALL send_bytes      ;send command & data to MAX7219
;
;-----RET

servo_motor:
    CPI    R31, 40          ;membandingkan nilai suhu pada R31 dengan 40
    BRLO  skip_buzzer ;jika suhu lebih kecil daripada 40, maka skip
;buzzer
    SBI    PORTD, 4         ;menyalakan buzzer

skip_buzzer:
    LDI    R17, 0
    OUT   SPCR, R17         ;enable SPI as master, fsck=fosc/16
    SBI   DDRB, 4            ;pin PB4 o/p for servo control
;
again_servo:
    LDI    R26, 8           ;counter for # of rotation pos
    LDI    ZL, lo8(rotate_pos)
    LDI    ZH, hi8(rotate_pos)
;
ls1: LPM   R24, Z+          ;load rotation pos
    RCALL rotate_servo     ;& rotate servo
    DEC   R26
    BRNE  ls1              ;go back & get another rotate pos
;
RJMP  SPI_MAX7219_init      ;go back & repeat
;
rotate_pos:
```

```

.byte 40,70,90,110,180,110,90,70
;=====

rotate_servo:
;-----

LDI    R20, 10          ;count to give enough cycles of PWM
SBI  DDRB, 1

ls2: SBI    PORTB, 4
      RCALL delay_timer0_servo
      CBI    PORTB, 4          ;send msec pulse to rotate servo
      RCALL delay_20ms_servo    ;wait 20ms before re-sending pulse
      DEC    R20
      BRNE   ls2              ;go back & repeat PWM signal
;-----


bak:RCALL delay_ms          ;0.5s delay
      RET                  ;& return to main subroutine
;-----


;delay subroutines
;=====

delay_timer0_servo:           ;delay via Timer0
;-----


CLR    R21
OUT    TCNT0, R21          ;initialize timer0 with count=0
MOV    R21, R24
OUT    OCR0A, R21
LDI    R21, 0b00001100
OUT    TCCR0B, R21          ;timer0: CTC mode, prescaler 256
;-----


ls3: IN     R21, TIFR0        ;get TIFR0 byte & check

```

```

        SBRS  R21, OCF0A      ;if OCF0=1, skip next instruction
        RJMP  ls3           ;else, loop back & check OCF0 flag
        ;-----
        CLR   R21
        OUT   TCCR0B, R21    ;stop timer0
        ;-----
        LDI   R21, (1<<OCF0A)
        OUT   TIFR0, R21     ;clear OCF0 flag
        RET
        ;=====

delay_20ms_servo:          ;delay 20ms (255*210*2*3/16000000)
        LDI   R21, 255
ls4: LDI   R22, 210
ls5: LDI   R23, 2
ls6: DEC   R23
        BRNE  ls6
        DEC   R22
        BRNE  ls5
        DEC   R21
        BRNE  ls4
        RET
        ;=====

delay_ms:                  ;delay 0.5s (255*255*41*3/16000000)
        LDI   R21, 255
ls7 :LDI   R22, 255
ls8 :LDI   R23, 41
ls9 :DEC   R23
        BRNE  ls9
        DEC   R22

```

```
BRNE    ls8
DEC     R21
BRNE    ls7
RET
```

Kode diatas adalah kode dalam bahasa Assembly yang digunakan untuk mengatur mikrokontroler AVR. Pada bagian awal, terdapat beberapa perintah yang mengatur pengaturan register dan pin mikrokontroler. Pertama-tama kita akan mengirim signal kepada dht11. Ketika dht11 mengirim sinyal response, maka microcontroller akan mulai menyimpan data berupa 40 bit yang diterima oleh sensor dht11;*;start\_signal*

```
;-----
        SBI    DDRD, 7           ;pin PD7 as o/p
        CBI    PORTD, 7          ;first, send low pulse
        RCALL delay_20ms        ;for 20ms
        SBI    PORTD, 7          ;then send high pulse
;-----
;
;response signal
;-----
        CBI    DDRD, 7           ;pin PD7 as i/p
w1: SBIC  PIND, 7
        RJMP  w1                 ;wait for DHT11 low pulse
w2: SBIS  PIND, 7
        RJMP  w2                 ;wait for DHT11 high pulse
w3: SBIC  PIND, 7
        RJMP  w3                 ;wait for DHT11 low pulse
```

Kemudian akan dilakukan pembacaan suhu melalui DHT11\_reading. Dht11 sendiri menerima 40 bit data dengan informasi berupa suhu, kelembaban, dan checksum. Namun,

hanya dua data yang kami perlukan yaitu temperature dalam bentuk integer dan humidity dalam bentuk integer. Oleh karena itu dilakukan pembacaan sebanyak 3 kali :

```
RCALL DHT11_reading ;read humidity (1st byte of 40-bit data)
MOV R25, R24
RCALL DHT11_reading
RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
```

Selanjutnya, data yang sudah dibaca akan ditampilkan pada LCD MAX7219. Untuk bisa didisplay, maka data tersebut dikonversi kedalam bentuk decimal, kemudian baru ditampilkan pada LCD MAX7219. Setelah itu, akan dilakukan pengecekan melalui routine check, yang berfungsi untuk mengecek apakah suhu pada ruangan sudah sesuai dengan suhu yang ideal yaitu 20 - 30 derajat celcius. Jika diatas 40 derajat maka buzzer dan servo akan menyala, jika diantara 30 - 40 derajat maka servo akan menyala, dan jika dibawah 20 derajat maka lampu LED akan menyala ;

```
;-----
;COMPARE THE OUTPUT
;-----
check:
CPI R31, 30 ;membandingkan nilai suhu pada R31 dengan 30
BRLO skip_servo ;jika suhu lebih kecil daripada 30, maka skip servo
CBI PORTD, 5 ;mematikan led
RCALL servo_motor ;menjalankan servo

skip_servo:
CBI PORTD, 4 ;mematikan led
CPI R31, 20 ;membandingkan nilai suhu R31 dengan 20
BRGE skip_led ;jika suhu lebih besar daripada 20, maka skip led
```

```

SBI    PORTD, 5           ;menyalakan led
RET               ;kembali ke looping

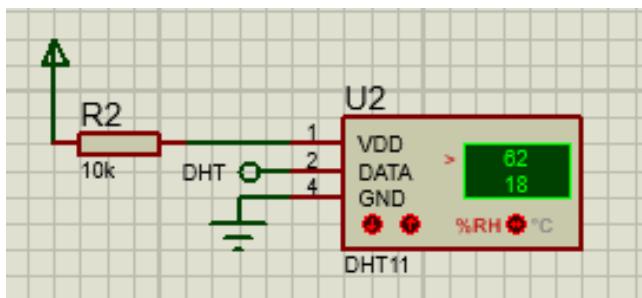
skip_led:
CBI    PORTD, 5           ;mematikan led
RET               ;kembali ke looping

```

Program akan terus menerus membaca data pada ruangan untuk menjaga suhu greenhouse agar tetap stabil dan ideal untuk tanaman.

### 2.3 HARDWARE AND SOFTWARE INTEGRATION

DHT11 adalah sensor suhu dan kelembaban yang bekerja dengan mengukur perubahan resistansi pada elemen sensor yang dipengaruhi oleh suhu dan kelembaban udara. DHT11 bekerja dengan cara mengubah nilai resistansi dari hasil pengukuran suhu dan kelembaban menjadi bentuk sinyal digital yang dapat dibaca oleh mikrokontroler. DHT11 :



DHT11 menerima data sebanyak 40 bit yang terdiri dari kelembaban dalam bentuk integer, kelembaban dalam bentuk decimal, temperature (integer), temperature (decimal), dan checksum. Namun, pada alat ini kami hanya menggunakan dua data yaitu kelembaban (integer) dan suhu (integer) :

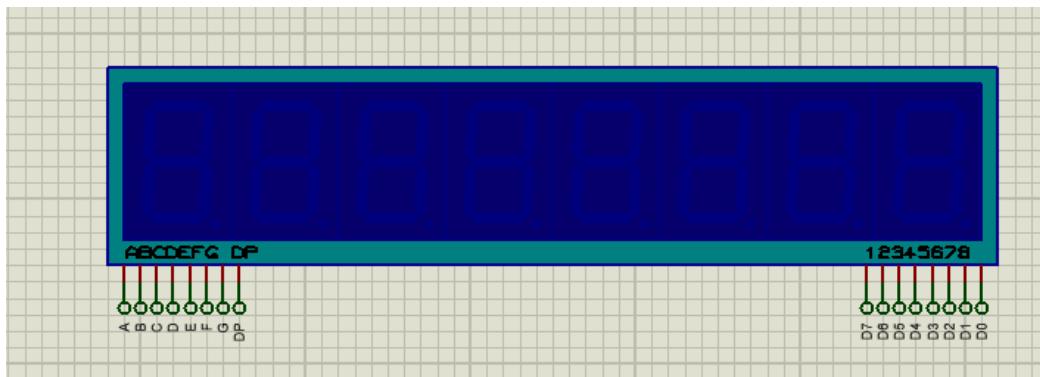
```

RCALL DHT11_reading ;read humidity (1st byte of 40-bit data)
MOV    R25, R24
RCALL DHT11_reading

```

```
RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
```

Kemudian data tersebut akan di display oleh LCD MAX721. LCD MAX7219 berkomunikasi dengan mikrokontroler melalui protokol serial seperti SPI (Serial Peripheral Interface) :



Suhu yang sudah dibaca oleh DHT11 kemudian akan digunakan oleh LCD MAX7219 untuk di display. Pertama-tama data tersebut akan dikonversi menjadi desimal, kemudian di display sesuai dengan suhu yang dibaca :

```
;=====
MAX7219_disp_text:
;-----

LDI    R17, 0x08      ;select digit 7
LDI    R18, 0xF        ;data = t (defg) (00001111)
RCALL send_bytes      ;send command & data to MAX7219
;-----

LDI    R17, 0x05      ;select digit 4
LDI    R18, 0x4E       ;data = C (adef) (01001110)
RCALL send_bytes      ;send command & data to MAX7219
;-----

LDI    R17, 0x04      ;select digit 3
LDI    R18, 0x00       ;data = space
RCALL send_bytes      ;send command & data to MAX7219
```

```

;-----

LDI    R17, 0x03      ;select digit 2
LDI    R18, 0x17      ;data = h (cefg) (00010111)
RCALL send_bytes      ;send command & data to MAX7219
;-----

RET

;=====

send_bytes:

CBI    PORTB, SS      ;enable slave device MAX7219
OUT    SPDR, R17      ;transmit command
;-----

l12:  IN    R19, SPSR
SBRS  R19, SPIF      ;wait for byte transmission
RJMP  l12            ;to complete
;-----

OUT    SPDR, R18      ;transmit data
;-----


l13:  IN    R19, SPSR
SBRS  R19, SPIF      ;wait for byte transmission
RJMP  l13            ;to complete
;-----

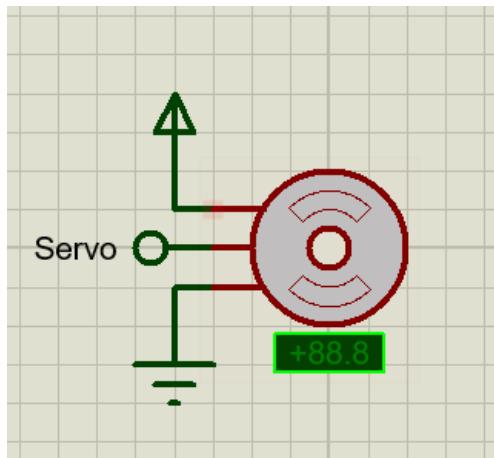

SBI    PORTB, SS      ;disable slave device MAX7219
RET

```

Tampilan dari LCD MAX7219 akan menjadi :



Kemudian terdapat servo yang berfungsi sebagai pendingin yang akan menyala ketika suhu diatas 30 derajat celcius. Servo SG90 dikendalikan oleh sinyal PWM untuk mengatur sudut rotasi :



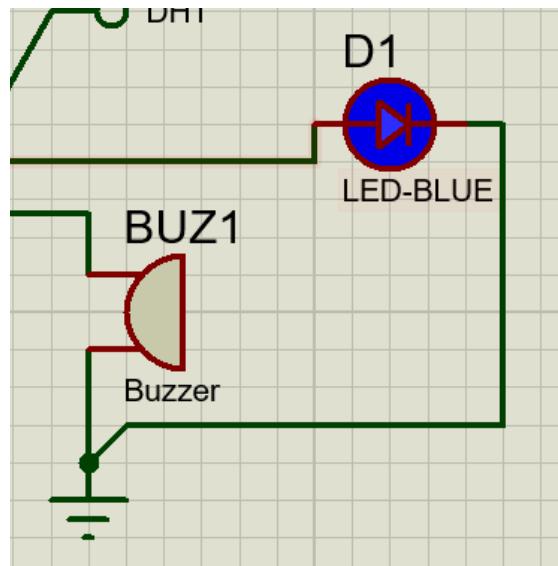
Kode untuk mengecek apakah suhunya diatas atau dibawah 30 derajat dan jika diatas maka akan call servo\_motor :

```

check:
    CPI    R31, 30          ; membandingkan nilai suhu pada R31 dengan 30
    BRLO  skip_servo       ; jika suhu lebih kecil daripada 30, maka
skip servo
    CBI    PORTD, 5         ; mematikan led
    RCALL servo_motor      ; menjalankan servo

```

Lalu, kami juga menggunakan buzzer dan LED :



Buzzer akan berbunyi jika suhu yang diterima diatas 40 derajat, sementara LED akan menyala ketika suhu yang diterima dibawah 20 derajat :

Buzzer :

```
servo_motor:
    CPI    R31, 40      ; membandingkan nilai suhu pada R31 dengan 40
    BRLO   skip_buzzer ; jika suhu lebih kecil daripada 40, maka skip
buzzer
    SBI    PORTD, 4     ; menyalakan buzzer
```

LED:

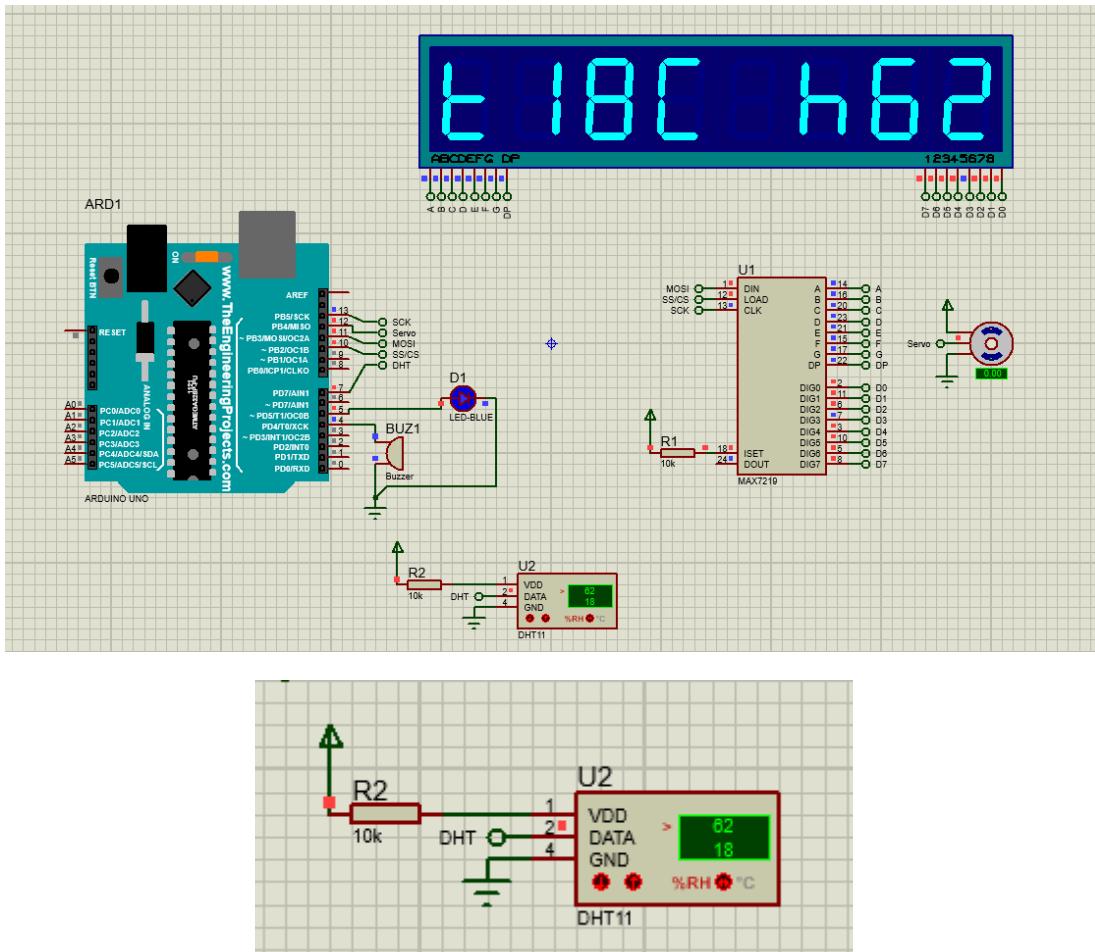
```
skip_servo:
    CBI    PORTD, 4      ; mematikan led
    CPI    R31, 20        ; membandingkan nilai suhu R31 dengan 20
    BRGE   skip_led       ; jika suhu lebih besar daripada 20, maka
skip led
    SBI    PORTD, 5        ; menyalakan led
    RET                 ; kembali ke looping
```

# CHAPTER 3

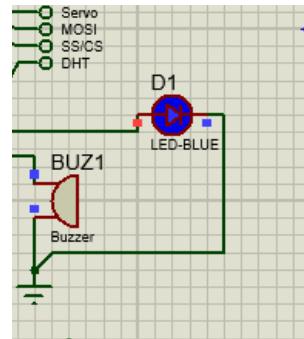
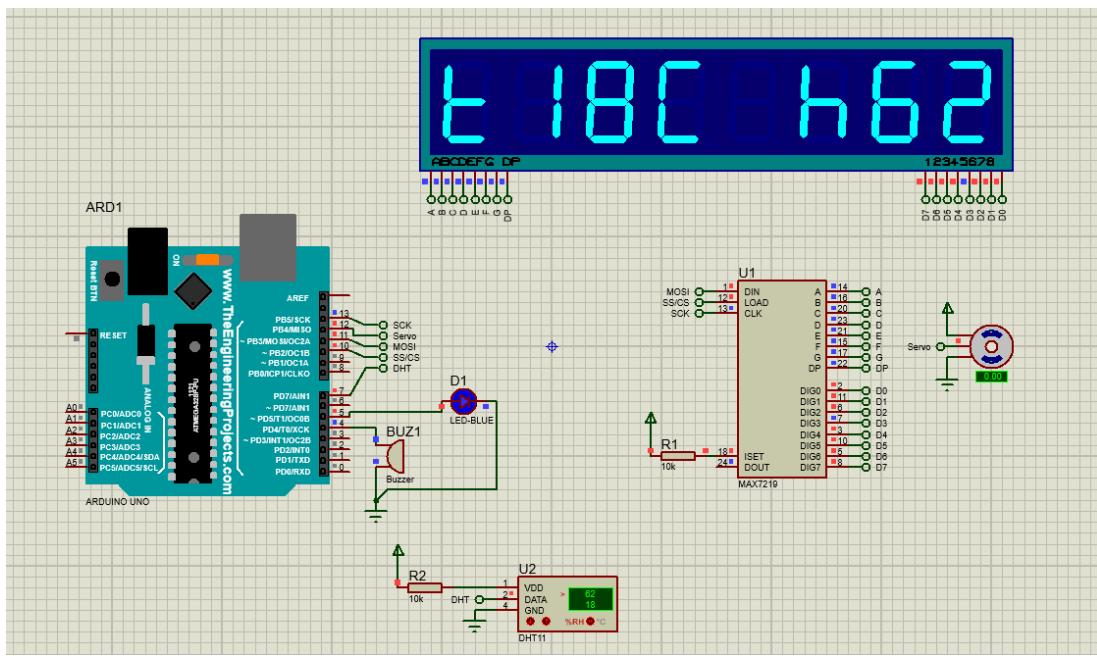
## TESTING AND EVALUATION

### 3.1 TESTING

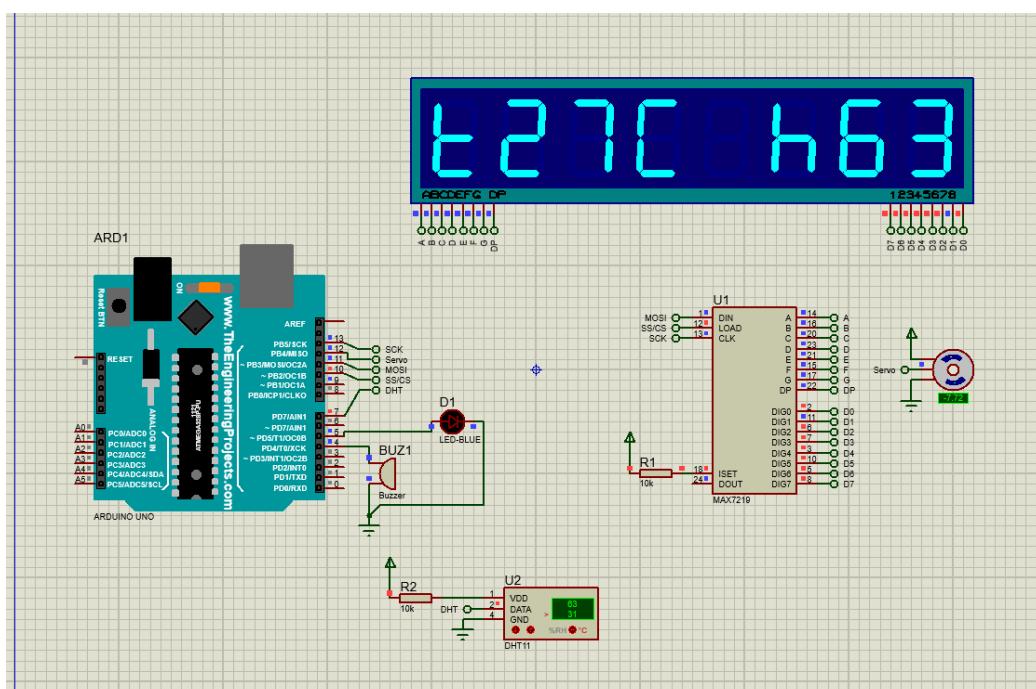
Pada rangkaian ini, kami mencoba perihal ketersesuaian fungsi dari alat ini. Pertama - tama kami mengetes dengan menggunakan sensor DHT11 untuk bisa membaca temperatur dan kelembaban yang ada pada dunia nyata. Setelah itu, sensor DHT11 ini juga akan terkoneksi dengan LCD MAX7219 untuk menampilkan besar temperatur dan kelembabannya:



Melalui rangkaian tersebut kita bisa melihat bahwa nilai suhu yang kita masukkan pada DHT11 dan output pada MAX7219 sudah sesuai. Kemudian, kami menguji fitur LED yang akan menyala ketika suhu dibawah 20 derajat :



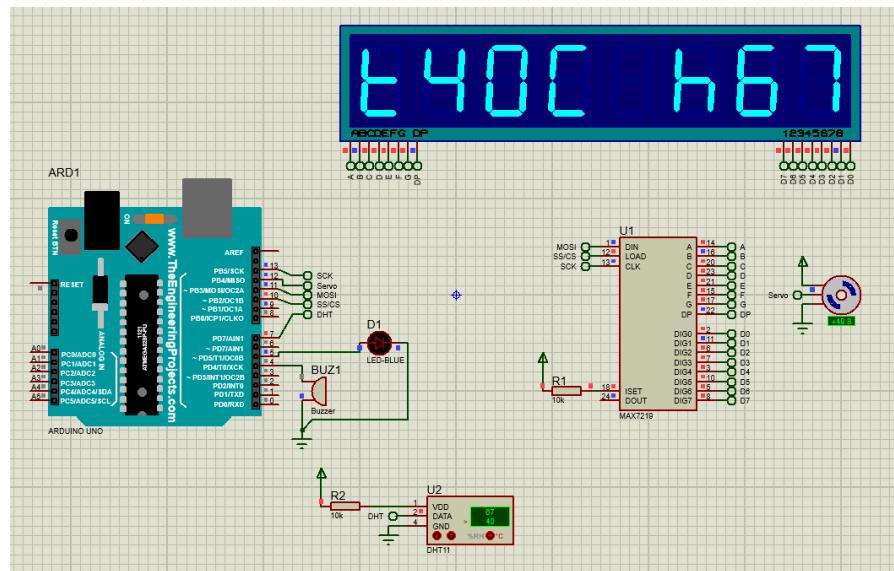
Selanjutnya kami mengetes apabila suhu diatas 30 derajat maka servo akan berputar :



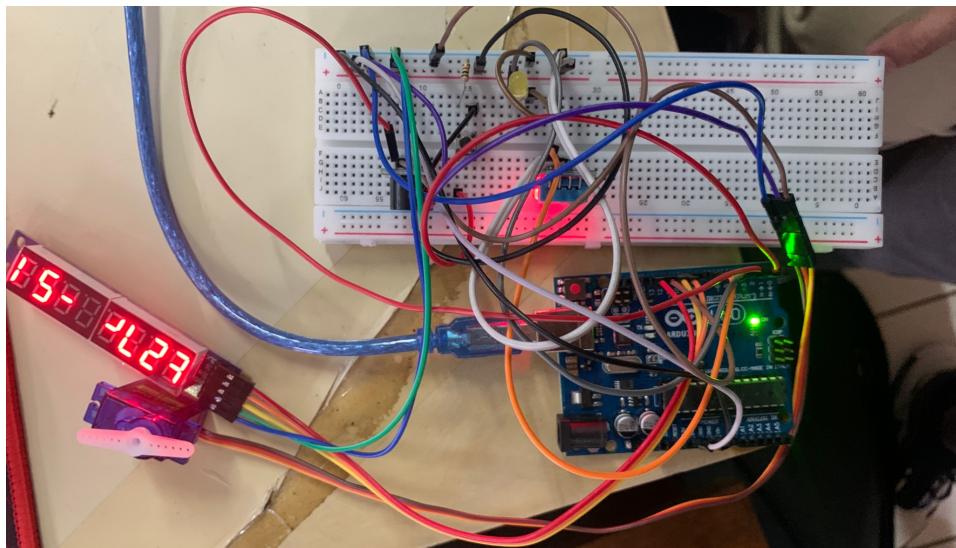
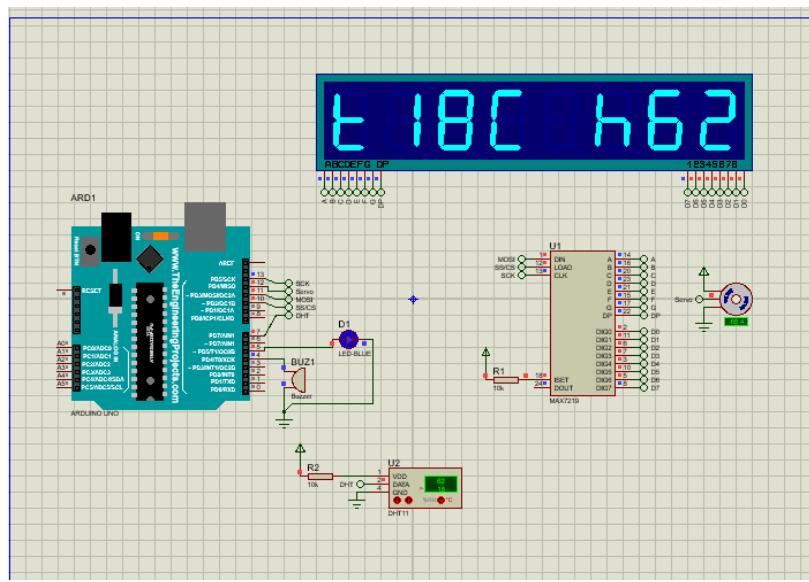
Bisa dilihat pada rangkaian tersebut servo sudah menyala. Sehingga, bisa disimpulkan bahwa semua fitur yang ada pada rangkaian kami sudah berjalan dengan semestinya.

### 3.2 RESULT

Berdasarkan alat yang telah kami buat, kami telah berhasil mengintegrasikan antara kodingan dengan perangkat kerasnya. Sensor DHT11 berhasil membaca kondisi temperatur ruangan dan berhasil ditampilkan pada layar MAX7219. Selain itu, kerja dari Servo juga telah sesuai dengan keinginan kami dimana saat temperatur di atas 30 derajat Celcius, maka Servo akan menyala dan kipas akan menyala untuk mendinginkan ruangan. Hal ini dapat dilihat pada rangkaian proteus berikut :



Kami juga melakukan simulasi langsung pada alat kami yang dapat dilihat pada foto berikut :



### 3.3 EVALUATION

Kerja dari peralatan proyek ini telah sesuai dengan kriteria dan keinginan kami. Namun, terdapat kesalahan alat dimana pada Sensor DHT11 ini tidak selalu membacakan data temperatur dengan akurat sehingga bisa menimbulkan kesalahpahaman jika dibandingkan dengan kondisi temperatur aslinya. Selain itu, untuk penggunaan Servo sebaiknya bisa digunakan dengan alat lainnya karena pergerakan Servo hanya terbatas sehingga agak sulit jika ingin digunakan untuk menggerakkan kipas. Namun, Servo ini bisa dijadikan bukti contoh simulasi untuk menggerakkan kipas di kondisi tertentu. Terakhir, untuk penggunaan layar Layar MAX7219 telah sesuai dengan keinginan kami. Pada layar ini

telah ditampilkan data pembacaan temperatur (dengan simbol t) dan kelembaban (dengan simbol h).

## **CHAPTER 4**

### **CONCLUSION**

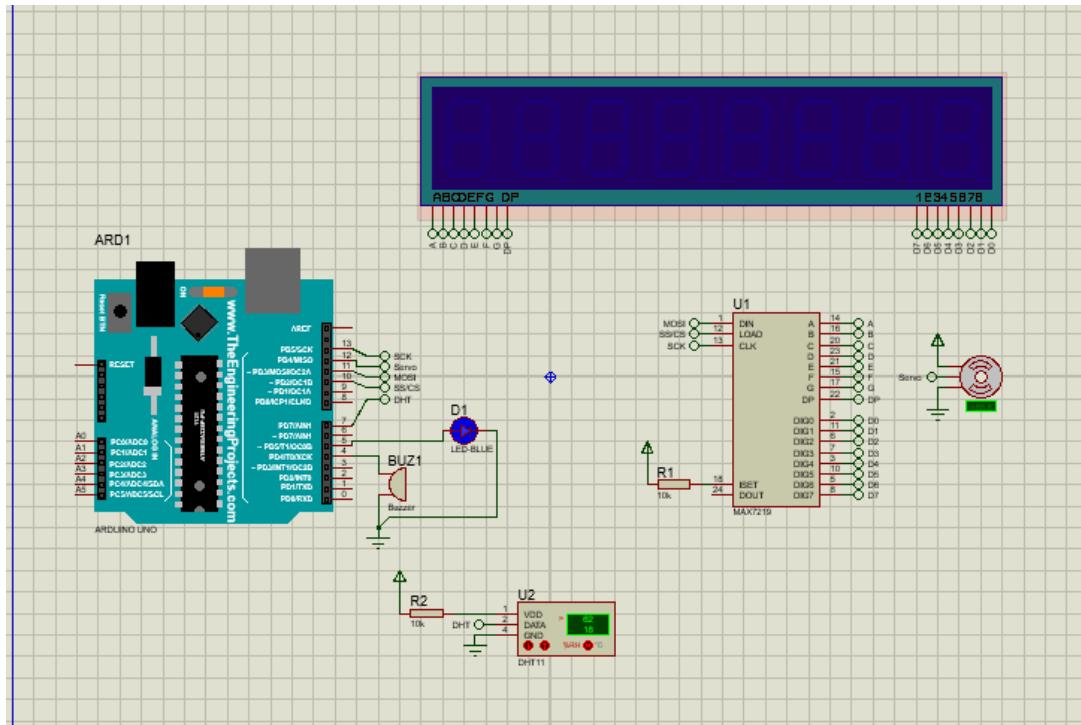
Alat *Automated Greenhouse* yang kami buat ini telah sesuai dengan kriteria dan keinginan kami. Alat ini mampu membantu pemilik *Greenhouse* untuk menjaga kestabilan temperaturnya. Alat ini mampu membaca temperatur dari kondisi ruangan secara riil, lalu terkoneksi dengan layar Layar MAX7219 untuk menampilkan data temperatur dan kelembabannya, dan terkoneksi dengan servo untuk menggerakkan kipas ketika kondisi temperatur telah di atas 30 derajat Celcius. Ketika temperatur telah di atas 40 derajat Celcius, maka buzzer akan menyala dan ketika temperatur berada di bawah 20 derajat Celcius, maka LED penghangat akan menyala. Dengan suksesnya alat ini, kami harap alat ini dapat diimplementasikan ke dunia nyata dengan peralatan-peralatan yang lebih besar dan lebih sesuai sehingga bisa membantu menjaga kestabilan temperatur *Greenhouse* secara nyata.

## REFERENCES

- [1] E. W. Support, “Environmental factors affecting plant growth,” OSU Extension Service,  
<https://extension.oregonstate.edu/gardening/techniques/environmental-factors-affecting-plant-growth#:~:text=Temperature%20influences%20most%20plant%20processes,photosynthesis%2C%20transpiration%20and%20respiration%20increase> (accessed May 5, 2023).
- [2] “How air temperature affects plants,” Air temperature for plants | CANNA UK,  
[https://www.canna-uk.com/how\\_air\\_temperature\\_affects\\_plants](https://www.canna-uk.com/how_air_temperature_affects_plants) (accessed May 5, 2023).
- [3] Admin@nutricontrol, “The importance of temperature for greenhouse cultivation,” Nutricontrol,  
<https://nutricontrol.com/en/the-importance-of-temperature-for-greenhouse-cultivation/> (accessed May 5, 2023).
- [4] *Assembly via Arduino (part 19) - dht11 sensor* (2021) YouTube. Available at:  
<https://www.youtube.com/watch?v=vnLpzvkCUq8> (Accessed: 16 May 2023).
- [5] Anas Kuzechie, “Assembly via Arduino (part 20) - DHT11 Data on MAX7219 Display,” YouTube. Dec. 02, 2021. [Online]. Available:  
<https://www.youtube.com/watch?v=UjWbE2Q75ms>

## APPENDICES

### Appendix A: Project Schematic



### Appendix B: Documentation

