



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

**ROOM SECURITY SYSTEM**

**GROUP A1**

<b>AZZAH AZKIYAH ANGELI SYAHWA</b>	<b>2106731390</b>
<b>CECILIA INEZ REVA MANURUNG</b>	<b>2106636994</b>
<b>MIKHAEL MORRIS HAPATARAN SIALLAGAN</b>	<b>2106731491</b>
<b>MUHAMMAD CAVAN NAUFAL AZIZI</b>	<b>2106702730</b>

## KATA PENGANTAR

Dengan segala rasa puji dan syukur dan puji kami panjatkan kepada Tuhan Yang Maha Esa karena karunia-Nya, kami berhasil menyelesaikan laporan praktikum akhir untuk kelas Sistem Siber-Fisik, yang berjudul "Room Security System", tepat pada waktunya. Laporan ini disusun untuk memberikan pemahaman kepada pembaca tentang bagaimana cara kerja dari sistem keamanan ruangan yang efektif dan efisien dengan memanfaatkan teknologi modern.

Proyek ini merupakan implementasi dari pengetahuan dan keterampilan yang kami peroleh selama mengikuti mata kuliah Sistem Siber-Fisik. Kami juga ingin menyampaikan terima kasih yang sebesar-besarnya kepada dosen pembimbing, F.Astha Ekadiyanto, S.T., M.Sc., serta asisten laboratorium dan teman-teman yang telah membantu dalam proses pembuatan proyek ini.

Meskipun kami telah berusaha menyusun laporan dengan cermat, kami menyadari bahwa masih terdapat kekurangan dan kelemahan yang perlu diperbaiki pada laporan proyek akhir "Room Security System" ini. Oleh karena itu, kami mengharapkan masukan dan saran yang konstruktif dari pembaca sebagai bentuk kritik yang akan membantu kami memperbaiki laporan ini secara optimal. Kami berharap laporan ini dapat memberikan manfaat dalam pengembangan ilmu pengetahuan dan teknologi, khususnya dalam bidang sistem keamanan ruangan. Terima kasih atas perhatian dan waktu yang diberikan.

Depok, 16 Mei, 2023

Group A1

## **TABLE OF CONTENTS**

<b>CHAPTER 1</b>	<b>4</b>
<b>INTRODUCTION</b>	<b>4</b>
1.1    PROBLEM STATEMENT	4
1.3    ACCEPTANCE CRITERIA	5
1.4    ROLES AND RESPONSIBILITIES	5
1.5    TIMELINE AND MILESTONES	5
<b>CHAPTER 2</b>	<b>7</b>
<b>IMPLEMENTATION</b>	<b>7</b>
2.1    HARDWARE DESIGN AND SCHEMATIC	7
2.2    SOFTWARE DEVELOPMENT	7
2.3    HARDWARE AND SOFTWARE INTEGRATION	8
<b>CHAPTER 3</b>	<b>9</b>
<b>TESTING AND EVALUATION</b>	<b>9</b>
3.1    TESTING	9
3.2    RESULT	9
3.3    EVALUATION	10
<b>CHAPTER 4</b>	<b>11</b>
<b>CONCLUSION</b>	<b>11</b>

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 PROBLEM STATEMENT**

Room Security System adalah alat keamanan ruangan yang dirancang dengan teknologi canggih dan dilengkapi dengan sensor gerak yaitu PIR Sensor (Passive Infra Red) dan Sensor HC-SR04 yang hasilnya ditampilkan melalui seven-segment MAX7219. Alat ini telah menjadi pilihan yang lebih populer saat ini daripada menggunakan kunci konvensional untuk mengunci pintu. Hal ini terjadi karena sistem keamanan ini memiliki kemampuan untuk mengoptimalkan keamanan dan kenyamanan pengguna secara lebih efektif dan efisien.

Alat pendeteksi keamanan ruangan dikembangkan untuk menjawab tantangan keamanan yang semakin kompleks di era digital. Dalam keadaan darurat seperti pencurian, kebakaran atau bencana alam, sistem keamanan ini dapat membantu menghindari kerugian yang besar dan melindungi orang-orang yang berada di dalam ruangan. Dalam kasus keamanan rumah tangga, alat ini dapat membantu mencegah pencurian dan memantau ruangan secara real-time melalui perangkat mobile atau perangkat lainnya.

Dalam rangka meningkatkan kualitas keamanan dan memberikan perlindungan terbaik bagi pengguna, pengembangan Room Security System menjadi hal yang penting. Oleh karena itu, proyek akhir ini bertujuan untuk merancang dan mengembangkan alat Room Security System yang terintegrasi dengan perangkat keras dan perangkat lunak yang dapat mendeteksi keadaan lingkungan sekitar dengan menggunakan sensor gerak dan memberikan tindakan yang tepat yaitu memberikan sinyal kepada pengguna berupa LED dan buzzer jika terdapat aktivitas yang terdeteksi di lingkungan sekitar. Diharapkan hasil dari proyek ini akan dapat memberikan manfaat bagi masyarakat dalam mengoptimalkan keamanan dan kenyamanan di lingkungan rumah atau kantor.

#### **1.2 PROPOSED SOLUTION**

Room Security System adalah sebuah perangkat keamanan ruangan yang dilengkapi dengan sensor gerak, yaitu sensor PIR, yang dapat mendeteksi gerakan pintu dalam radar dan sensitifitas tertentu sehingga dapat menjadi pemicu untuk mengaktifkan suara dari buzzer dan

LED sebagai tanda bahaya. Kemudian, sensor HC-SR04 mengukur jarak antara sensor dengan objek di depannya dan ditampilkan melalui LED seven-segment MAX7219 menggunakan protokol komunikasi serial, SPI. Alat ini ditujukan sebagai solusi untuk keamanan yang lebih ketat pada sebuah ruangan dengan menggunakan alat dan perangkat lunak yang terintegrasi.

### 1.3 ACCEPTANCE CRITERIA

Kriteria penerimaan proyek ini adalah sebagai berikut:

1. Menerapkan sebuah sistem siber-fisik, dalam bentuk mikrokontroler Arduino, sebagai platform dalam membuat prototype alat dan memprogram Sensor dan komponen lainnya.
2. Membuat rangkaian yang dilengkapi sensor gerak dan jarak yang dapat mendeteksi pergerakan pada pintu secara otomatis.
3. Mengimplementasikan protokol serial pada Arduino yaitu Serial Peripheral Interface (SPI).
4. Mengintegrasikan sensor dan Arduino agar dapat menjalankan program sehingga mampu mengaktifkan LED dan Buzzer.
5. Mendapatkan pemberitahuan kepada pengguna jika sensor berhasil mendeteksi gerakan pada pintu.

### 1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Programmer Arduino	Program Assembly	<ul style="list-style-type: none"><li>• Muhammad Cavan Naufal Azizi</li><li>• Cecilia Inez Reva Manurung</li></ul>

Pendesain skematik proteus	Membuat rangkaian schematic Arduino, sensor, LED, buzzer agar dapat terhubung	<ul style="list-style-type: none"> <li>• Muhammad Cavan Naufal Azizi</li> <li>• Azzah Azkiyah Angeli Syahwa</li> <li>• Mikhael Morris Hapataran Siallagan</li> </ul>
Mendesain Presentasi PPT	Membuat PPT	<ul style="list-style-type: none"> <li>• Cecilia Inez Reva Manurung</li> <li>• Mikhael Morris Hapataran Siallagan</li> </ul>
Pembuat Laporan	Membuat Laporan	<ul style="list-style-type: none"> <li>• Muhammad Cavan Naufal Azizi</li> <li>• Azzah Azkiyah Angeli Syahwa</li> <li>• Mikhael Morris Hapataran Siallagan</li> <li>• Cecilia Inez Reva Manurung</li> </ul>

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

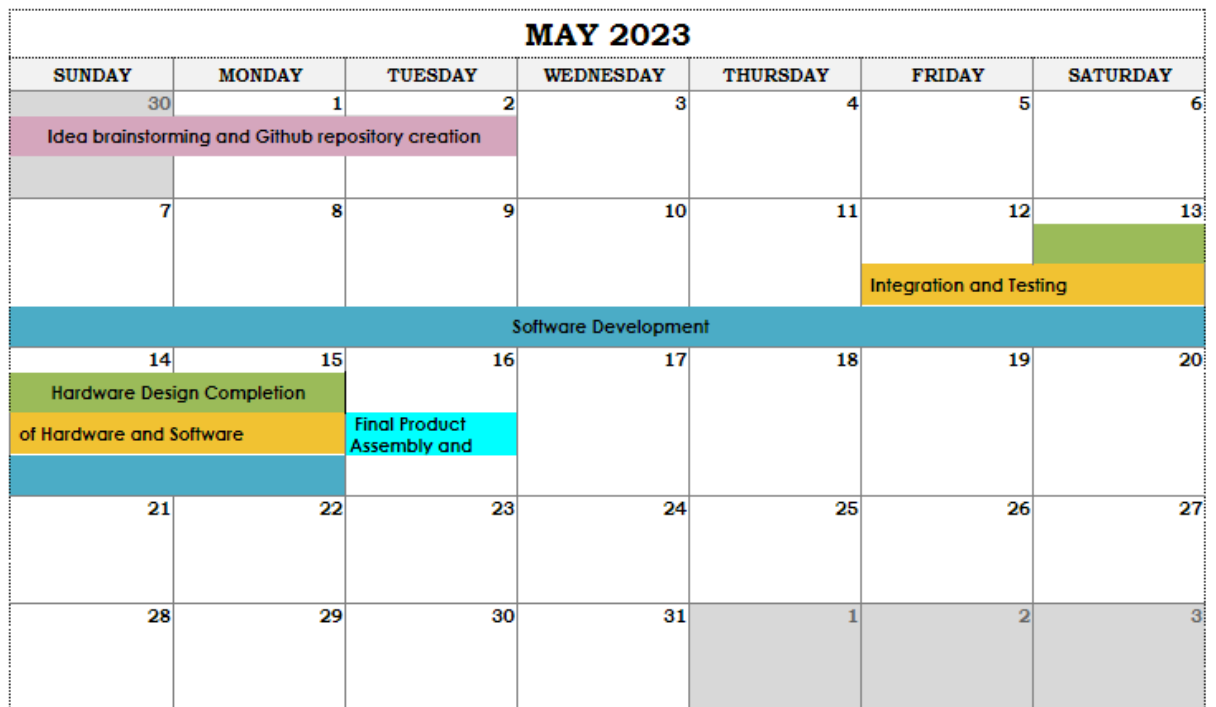


Figure 1. Gantt chart

Insert Gantt Chart here. The Gantt Chart should consist of date interval for:

- Hardware Design completion: 13 Mei - 15 Mei 2023.
- Software Development: 7 Mei - 15 Mei 2023.
- Integration and Testing of Hardware and Software: 12 Mei - 15 Mei 2023.
- Final Product Assembly and Testing: 16 Mei 2023.

## CHAPTER 2

### IMPLEMENTATION

#### a. 2.1 HARDWARE DESIGN AND SCHEMATIC

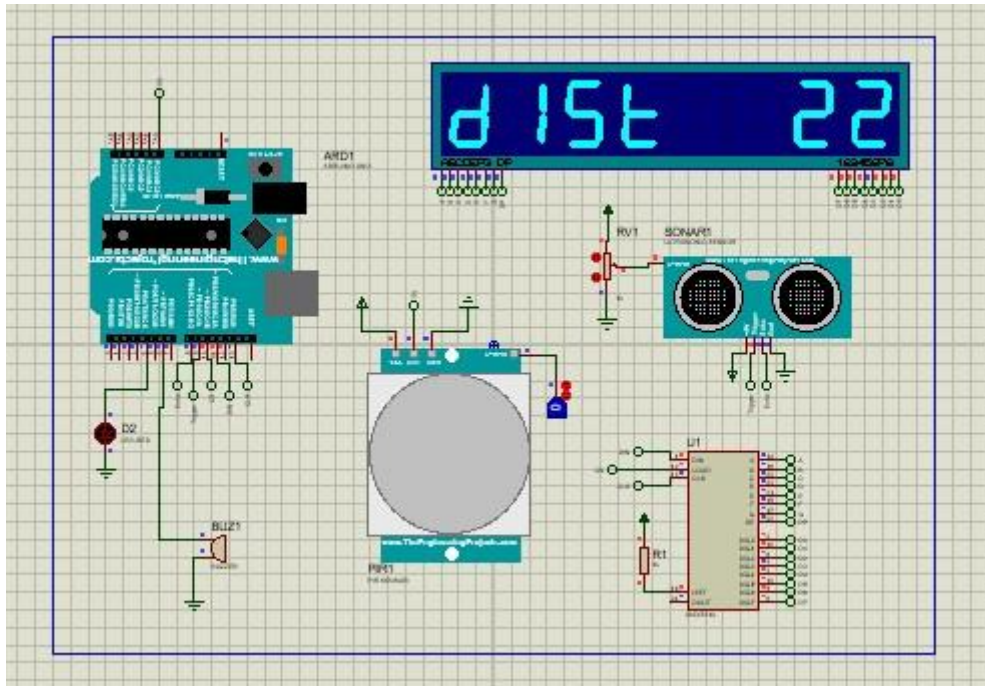


Figure 2. Schema

Pembuatan alat dilakukan dengan membuat skema alat atau prototype alat terlebih dahulu pada software Proteus. Skema dibuat untuk memudahkan kelompok dalam melakukan perangkaian alat sebelum proses perangkaian pada komponen asli sehingga meminimalisir kerusakan komponen asli akibat kesalahan rangkaian atau rangkaian yang belum terintegrasi dengan baik.

Komponen yang dibutuhkan dalam mendesain dan merangkai skema alat serta fungsinya dalam rangkaian antara lain adalah :

1. Satu buah Arduino Uno ATMEGA328p

Mikrokontroler berfungsi sebagai otak dari sebuah sistem. Mikrokontroler ini bertanggung jawab untuk mengontrol dan mengkoordinasikan berbagai komponen sistem keamanan. Mikrokontroler ini menjalankan program yang



membaca input dari sensor, memproses data, dan mengontrol output yang sesuai.

## 2. Satu buah Sensor PIR

Modul sensor PIR digunakan untuk mendeteksi gerakan di sekitarnya. Sensor ini bekerja dengan merasakan perubahan radiasi inframerah yang dipancarkan oleh makhluk hidup dan objek. Ketika gerakan terdeteksi, sensor PIR mengeluarkan sinyal yang digunakan oleh mikrokontroler untuk memicu buzzer untuk mengeluarkan suara dan menyalakan LED.

## 3. Satu buah HC-SR04/Sensor Jarak

Sensor HC-SR04 adalah sensor pengukuran jarak ultrasonik. Sensor ini memancarkan gelombang suara ultrasonik dan mengukur waktu yang diperlukan gelombang suara untuk memantul kembali setelah mengenai suatu objek. Dalam sistem keamanan ruangan, sensor HC-SR04 digunakan untuk mendeteksi kedekatan atau mengukur jarak, sehingga memungkinkan fitur keamanan tambahan atau pemantauan zona.

## 4. Satu buah LED

LED (Light Emitting Diode) adalah perangkat output visual yang dapat memancarkan cahaya ketika arus listrik melewatinya. LED dapat digunakan untuk mengindikasikan status sistem atau memberikan umpan balik visual, seperti menampilkan keadaan bahaya.

## 5. Satu buah Active Buzzer

Buzzer adalah perangkat output yang digunakan untuk menghasilkan peringatan atau alarm yang dapat didengar. Perangkat ini dapat digunakan untuk menghasilkan sinyal suara untuk menunjukkan adanya penyusup atau untuk membunyikan alarm jika terjadi pelanggaran keamanan.

## 6. Breadboard

Breadboard adalah platform yang digunakan untuk membuat prototype sirkuit elektronik tanpa menyolder. Hal ini memungkinkan koneksi yang mudah dan eksperimen dengan komponen yang berbeda dengan menyediakan kisi-kisi soket yang saling berhubungan. Breadboard memungkinkan perakitan dan modifikasi sirkuit dengan cepat untuk tujuan pengujian dan pengembangan.

#### 7. Jumper

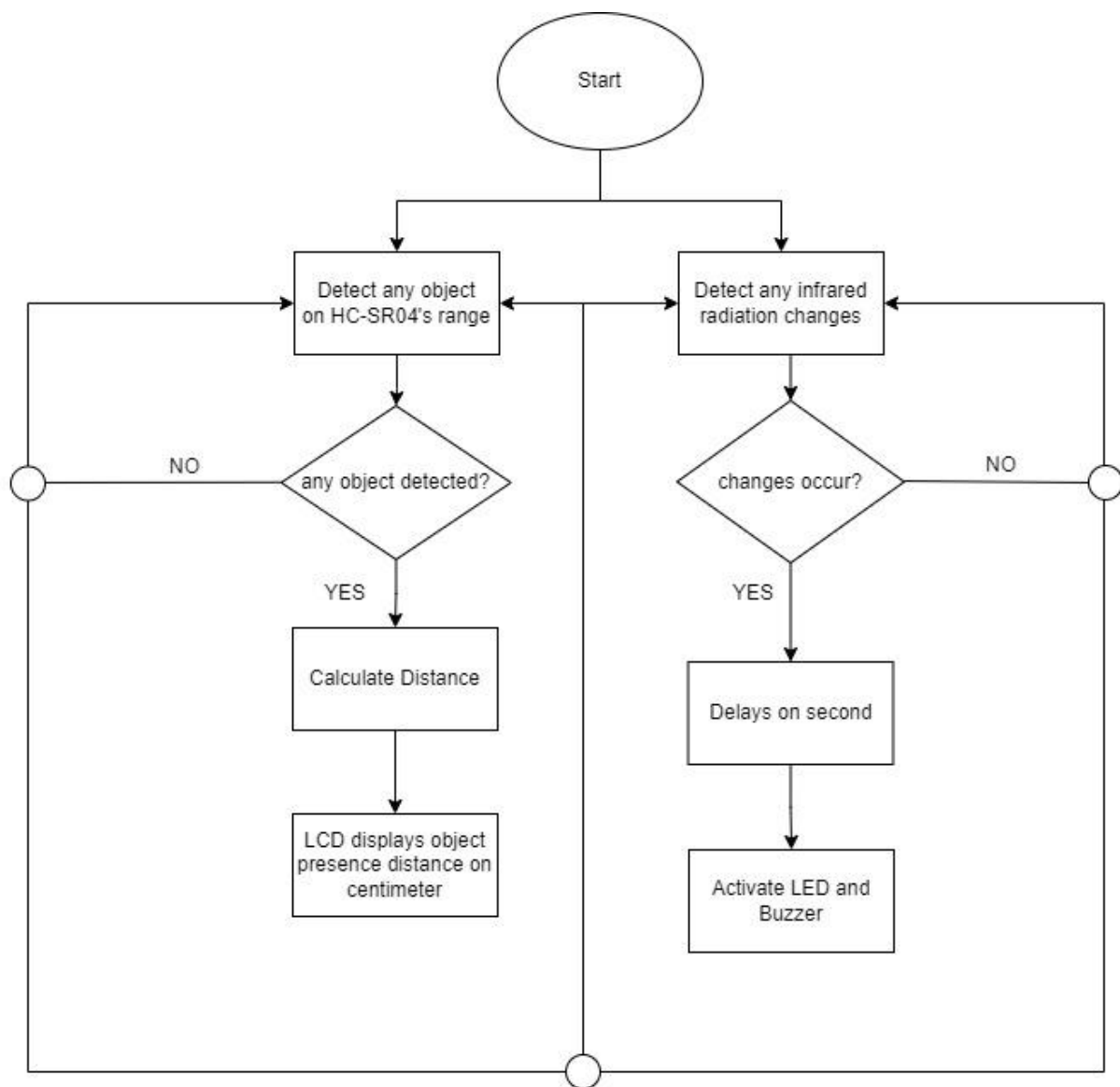
Kabel jumper digunakan untuk membuat sambungan listrik antara berbagai komponen pada breadboard atau antara breadboard dan mikrokontroler. Kabel ini menyediakan cara yang fleksibel dan nyaman untuk menghubungkan berbagai titik di sirkuit.

#### 8. Catu Daya

Catu daya berfungsi untuk menyediakan energi listrik yang diperlukan untuk mengoperasikan sistem keamanan ruangan. Catu daya dapat berupa baterai maupun AC/DC Adaptor. Tugasnya adalah memastikan bahwa semua komponen menerima tegangan dan arus yang sesuai untuk pengoperasiannya.

## **b. 2.2 SOFTWARE DEVELOPMENT**

Untuk mempermudah penyusunan cara kerja alat, diperlukan kerangka kerja yang dibuat di website draw.io sebagai berikut :



*Figure 3. Flowchart*

Pengujian dilakukan dengan perakitan skema pada komponen asli sesuai dengan desain yang tertera di **Figure 2** dan dijalankan dengan kode berikut :

```

#define __SFR_OFFSET 0x00

#include "avr/io.h"

.global main

main :

    RCALL SPI_MAX7219_init
    RCALL MAX7219_disp_text
    RCALL all_sensor

all_sensor:

    ldi R24, 0xFF
    OUT DDRD, R24        ;set PORTD (PORTD as o/p - led_buzzer)
    CBI DDRC, 0          ;clear PC0 (pin A0 as i/p - PIR Sensor)
    SBI DDRB, 1          ;pin PB1 as o/p (Trigger)
    CBI DDRB, 0          ;pin PB0 as i/p (Echo)

loop:

    sbic PINC, 0         ; skip next instruction if PC0 is low
    rjmp led_buzzer_on   ; jump to led_buzzer_on if PC0 is high
    ldi R24, 0b00000000  ; turn off led_buzzer if PC0 is low
    out PORTD, R24

    RCALL agn

agn:

    SBI PORTB, 1
    RCALL delay_timer0
    CBI PORTB, 1         ;send 10us high pulse to sensor
    ;-----
    RCALL echo_PW        ;compute Echo pulse width count
    ;-----
    RCALL byte2decimal    ;covert & display on MAX7219

```

```

;-----
RCALL delay_ms
RJMP loop

echo_PW:
;-----
LDI R20, 0b00000000
STS TCCR1A, R20 ;Timer 1 normal mode
LDI R20, 0b11000101 ;set for rising edge detection &
STS TCCR1B, R20 ;prescaler=1024, noise cancellation ON
;-----
11: IN R21, TIFR1
SBRS R21, ICF1
RJMP 11 ;loop until rising edge is detected
;-----
LDS R16, ICR1L ;store count value at rising edge
;-----
OUT TIFR1, R21 ;clear flag for falling edge detection
LDI R20, 0b10000101
STS TCCR1B, R20 ;set for falling edge detection
;-----
12: IN R21, TIFR1
SBRS R21, ICF1
RJMP 12 ;loop until falling edge is detected
;-----
LDS R28, ICR1L ;store count value at falling edge
;-----
SUB R28, R16 ;count diff R22 = R22 - R16
OUT TIFR1, R21 ;clear flag for next sensor reading
RET

led_buzzer_on:

```

```

    ldi R24, 0b01010000    ; turn on led_buzzer if PC0 is high
    out PORTD, R24

delay_loop:
    LDI    R21, 255
a: LDI    R22, 255
b: LDI    R23, 50
c: DEC    R23

    BRNE   c

    DEC    R22

    BRNE   b

    DEC    R21

    BRNE   a

    ldi R16, 0b00000000    ; turn off led_buzzer after delay
    out PORTD, R16

    rjmp  loop

;=====

SPI_MAX7219_init:

    .equ   SCK, 5
    .equ   MOSI, 3
    .equ   SS, 2

;-----

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

;-----

    LDI    R17, (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, 0b00110000 ;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, 0x3D      ;data = d
RCALL send_bytes      ;send command & data to MAX7219
;-----

LDI    R17, 0x07      ;select digit 6
LDI    R18, 0x00      ;data = space
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, 0x0D        ;data = c
    RCALL  send_bytes      ;send command & data to MAX7219
    ;-----

    LDI    R17, 0x02        ;select digit 1
    LDI    R18, 0x15        ;data = n
    RCALL  send_bytes      ;send command & data to MAX7219
    ;-----

    LDI    R17, 0x01        ;select digit 0
    LDI    R18, 0x0F        ;data = t
    RCALL  send_bytes      ;send command & data to MAX7219
    ;-----

    RET

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

112:    IN     R19, SPSR
    SBRS   R19, SPIF        ;wait for byte transmission
    RJMP   112              ;to complete
    ;-----

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

113:    IN     R19, SPSR
    SBRS   R19, SPIF        ;wait for byte transmission
    RJMP   113              ;to complete
    ;-----

    SBI    PORTB, SS        ;disable slave device MAX7219
    RET

;=====
byte2decimal:

```



```

;-----
    CLR    R26                ;set counter1, initial value 0
    CLR    R27                ;set counter2, initial value 0
    ;-----
170:  CPI    R28, 100          ;compare R28 with 100
Ret:  BRMI   R28               ;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
      LDI    R17, 0x06         ;select digit 5 for MSD
      RCALL  send_bytes        ;send command & data to MAX7219
    ;-----
      MOV    R18, R28
      LDI    R17, 0x05         ;select digit 4 for LSD
      RCALL  send_bytes        ;send command & data to MAX7219
    ;-----
      RET
;=====
;delay subroutines
;=====
delay_timer0:                ;10 usec delay via Timer 0
;-----
      CLR    R20
      OUT    TCNT0, R20        ;initialize timer0 with count=0

```

```

LDI    R20, 20

OUT    OCR0A, R20      ;OCR0 = 20

LDI    R20, 0b00001010

OUT    TCCR0B, R20     ;timer0: CTC mode, prescaler 8

;-----

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

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

RJMP   10              ;else, loop back & check OCF0 flag

;-----

CLR    R20

OUT    TCCR0B, R20     ;stop timer0

;-----

LDI    R20, (1<<OCF0A)

OUT    TIFR0, R20      ;clear OCF0 flag

RET

;=====

delay_ms:

;-----

LDI    R21, 255

16: LDI    R22, 255

17: LDI    R23, 50

18: DEC    R23

BRNE   18

DEC    R22

BRNE   17

DEC    R21

BRNE   16

RET

```

Program ini ditulis dalam bahasa assembly untuk menghubungkan sensor PIR, sensor jarak HC-SR04 dan LED seven-segment MAX7219 dalam mikrokontroler ATmega328P. Program ini menggunakan protokol komunikasi serial synchronous yaitu SPI (Serial

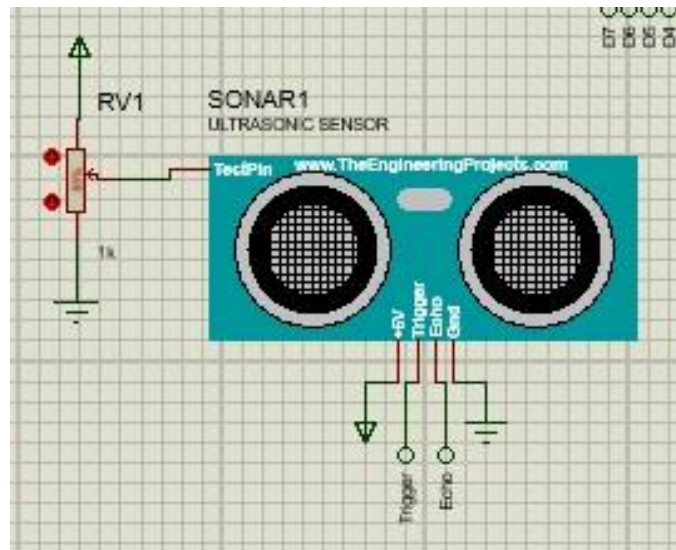
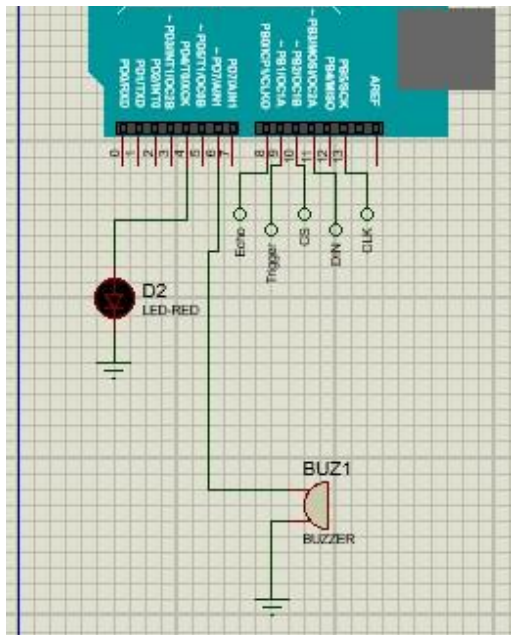
Peripheral Interface) untuk berkomunikasi dengan MAX7219 dan menghitung jarak yang diukur sensor HC-SR04 dengan mengukur lebar pulsa dari sinyal echo. Program dimulai dengan mengatur port input atau output. Port D dikonfigurasi sebagai output untuk LED merah dan buzzer yang terhubung, dan pin A0 (PC0) dikonfigurasi sebagai input untuk button. Pin PB0 dan PB1 dikonfigurasi sebagai pin input dan output untuk sensor HC-SR04.

Pada loop utama, program mengirimkan high pulse 10us ke sensor HC-SR04 untuk memulai pengukuran jarak. Kemudian menghitung lebar pulsa dari sinyal echo dengan subroutine `echo_PW`, yang menggunakan `Timer1` (normal mode). Program kemudian mengubah lebar pulse terukur menjadi nilai desimal dan menampilkannya pada MAX7219 menggunakan subroutine `MAX7219_disp_text`. Subroutine `MAX7219_disp_text` akan menampilkan string text "d" pada 2 digit pertama. Jika tombol yang terhubung ke pin A0 ditekan, LED dan buzzer yang terhubung ke port D akan menyala sebentar, dan kemudian mati.

Subroutine `SPI_MAX7219_init` mengatur SPI untuk berkomunikasi dengan MAX7219 dengan menyalakan pin MOSI (Master In Slave Out), SCK (clock signal) dan SS sebagai output dengan menuliskan bit 1 ke `DDRB` dan mengaktifkan SPI pada bit pertama (SPE). Subroutine `send_bytes` digunakan untuk mengatur intensitas segment LED dengan nilai 8 dan mengatur scan limit dengan nilai 7 yang artinya menampilkan 8 digit pada MAX7219 dan menyalakan MAX7219.

### **c. 2.3 HARDWARE AND SOFTWARE INTEGRATION**

Sensor Passive Infra Red (PIR) mampu mendeteksi pergerakan pada jarak tertentu melalui perubahan radiasi infra merah di lingkungan sekitar. Sensor PIR memberikan dua keadaan output yaitu logika High 1 sebesar 3,3 V dan logika Low sebesar 0 V. Pada program untuk sensor HC-SR04, pin trigger diatur menjadi high atau sebesar 10 us untuk mengindikasikan bunyi. Pin echo dalam status high jika ultrasonic burst ditransmisikan dalam logika high hingga sensor mencapai echo dalam logika low. Jarak diukur selama pin echo tetap dalam status high.

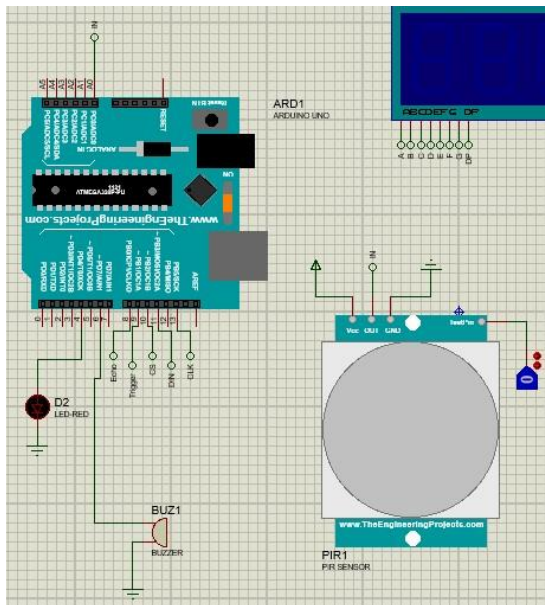


```

agn:

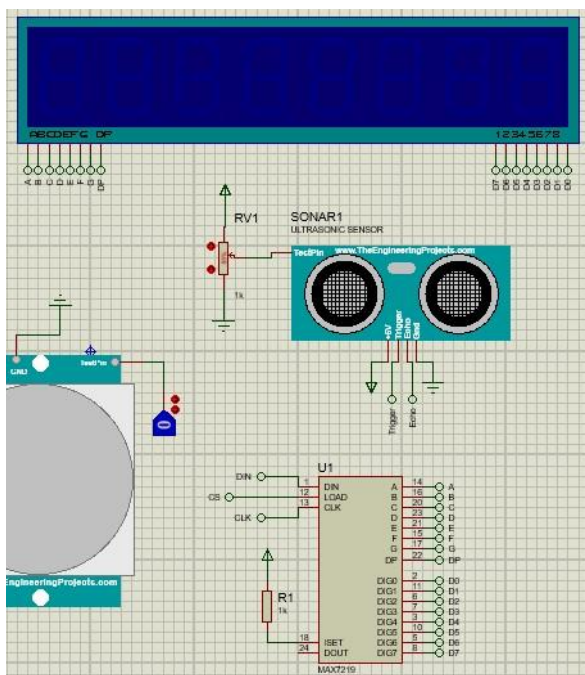
    SBI    PORTB, 1
    RCALL  delay_timer0
    CBI    PORTB, 1      ;send 10us high pulse to sensor
    ;-----
    RCALL  echo_PW       ;compute Echo pulse width count
    ;-----
    RCALL  byte2decimal   ;convert & display on MAX7219
    ;-----
    RCALL  delay_ms
    RJMP   loop
  
```

Slave device pada MAX7219 menggunakan port B sebagai output dari sensor HC-SR04 yaitu pin 8 sebagai echo, pin 9 sebagai trigger yang keduanya terdapat pada sensor HC-SR04, pin 10 sebagai CS, dan pin 13 sebagai CLK yang terdapat di seven-segment MAX7219. Program menggunakan pin MOSI (Master Out Slave In).



```
echo_PW:
;-----
LDI R20, 0b00000000
STS TCCR1A, R20 ;Timer 1 normal mode
LDI R20, 0b11000101 ;set for rising edge detection &
STS TCCR1B, R20 ;prescaler=1024, noise cancellation ON
;-----
11: IN R21, TIFR1
SBRS R21, ICF1
RJMP 11 ;loop until rising edge is detected
;-----
LDS R16, ICR1L ;store count value at rising edge
;-----
OUT TIFR1, R21 ;clear flag for falling edge detection
LDI R20, 0b10000101
STS TCCR1B, R20 ;set for falling edge detection
;-----
12: IN R21, TIFR1
SBRS R21, ICF1
RJMP 12 ;loop until falling edge is detected
;-----
LDS R28, ICR1L ;store count value at falling edge
;-----
SUB R28, R16 ;count diff R22 = R22 - R16
OUT TIFR1, R21 ;clear flag for next sensor reading
RET
```

Output buzzer di pin 7 dan LED merah di pin 4 yang akan menyala jika PC0 yaitu output pada sensor PIR dalam keadaan high.



```
SPI_MAX7219_init:
;-----
.equ SCK, 5
.equ MOSI, 3
.equ SS, 2
;-----
LDI R17, (1<<MOSI)|(1<<SCK)|(1<<SS)
OUT DDRB, R17 ;set MOSI, SCK, SS as o/p
;-----
LDI R17, (1<<SPE)|(1<<MSTR)|(1<<SPR0)
OUT SPCR, R17 ;enable SPI as master, fscck=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, 0b00110000 ;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
```

Mikrokontroler sebagai master device mengirimkan data ke MAX7219 melalui interface serial SCK, CS, dan MOSI. Master memulai transfer data dengan CS low, dan mengirimkan data melalui MOSI dan SCK untuk menyinkronkan transfer data. Untuk menampilkan angka pada segmen tertentu, master mengirimkan sepasang byte, misalnya memilih segment ke-8

atau digit 7 untuk menampilkan “d”, maka master mengirimkan byte 0x08 diikuti dengan byte 0x3D untuk menampilkan “d”.

```
MAX7219_disp_text:
;-----
    LDI    R17, 0x08      ;select digit 7
    LDI    R18, 0x3D      ;data = d
    RCALL  send_bytes     ;send command & data to MAX7219
;-----
```

## CHAPTER 3

### TESTING AND EVALUATION

#### d. 3.1 TESTING

Pengujian rangkaian dilakukan di software Proteus terlebih dahulu sebelum untuk memastikan penyusunan software atau program dari alat sudah berfungsi dengan baik. Skema yang disusun pada akhirnya menjadi sebagai berikut :

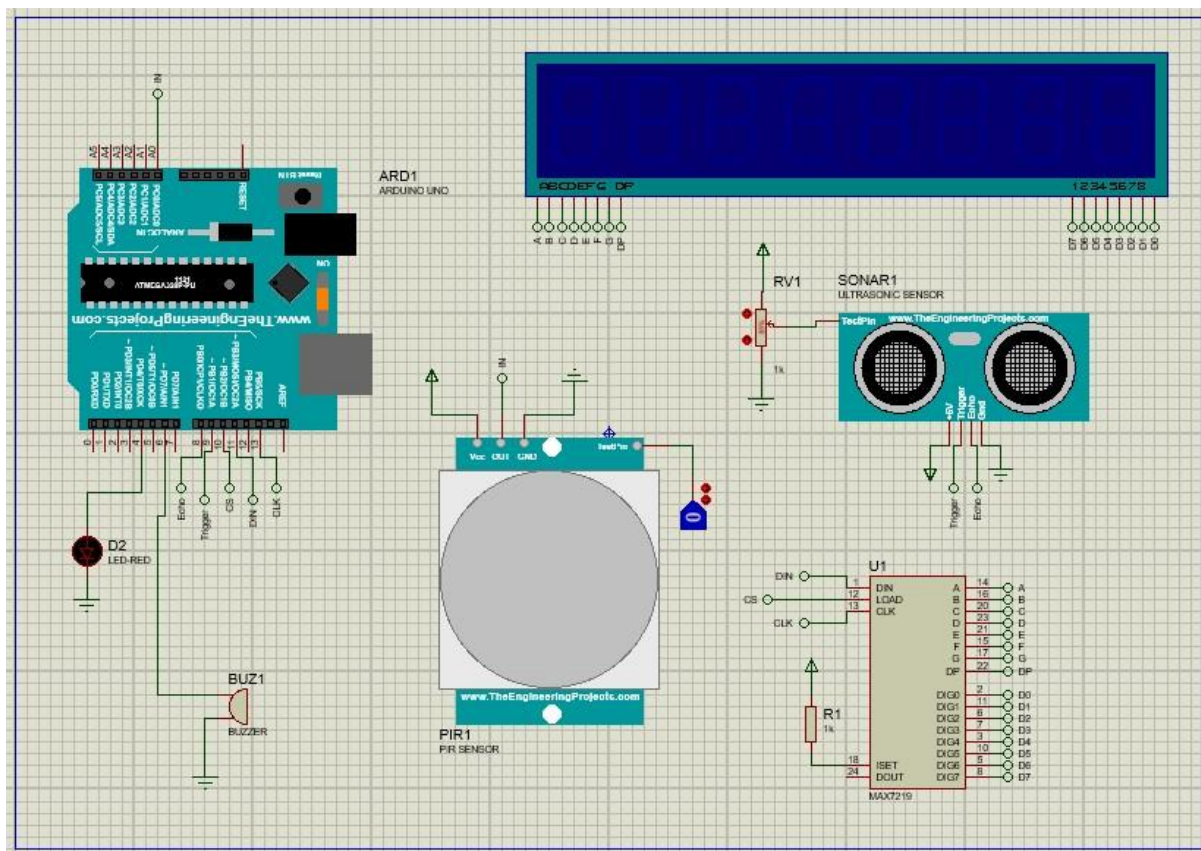
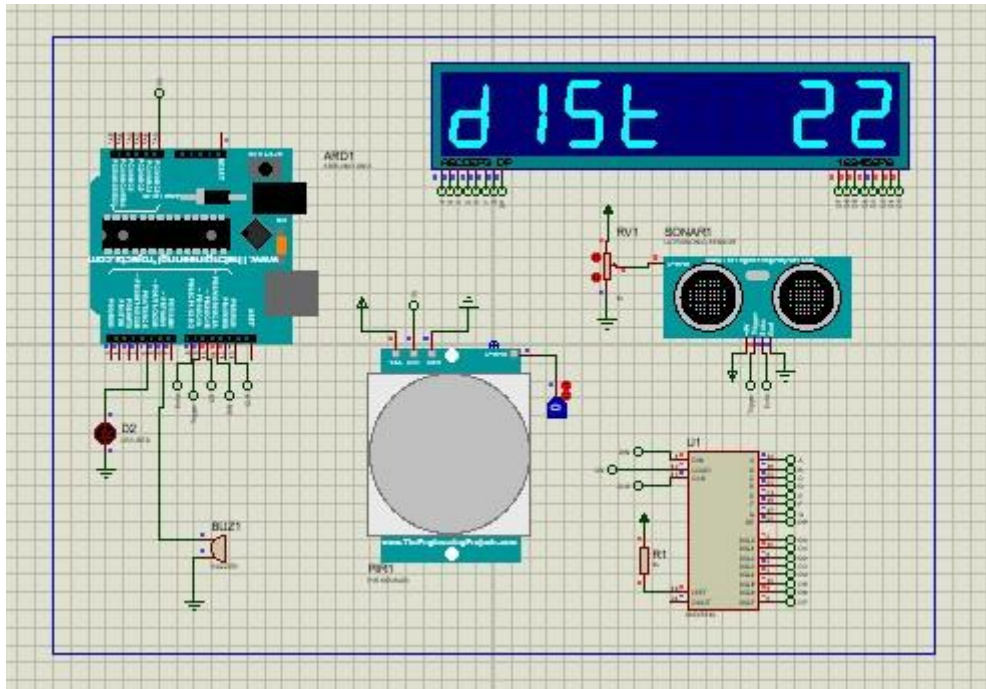


Figure 4. Testing in Proteus

Dalam pengujian yang dilakukan, kami memanfaatkan pin TestPin pada kedua sensor yang mana pin tersebut dihubungkan dengan logic state. Sehingga pada kondisi nilai logic state adalah 1 artinya high, maka sensor jarak akan mengirimkan data pada MAX7219 berupa jarak objek yang terdeteksi oleh sensor. Sedangkan sensor PIR akan memicu buzzer dan led untuk memberikan output berupa suara dan pemberitahuan visual. Output dari skema di proteus menjadi sebagai berikut :



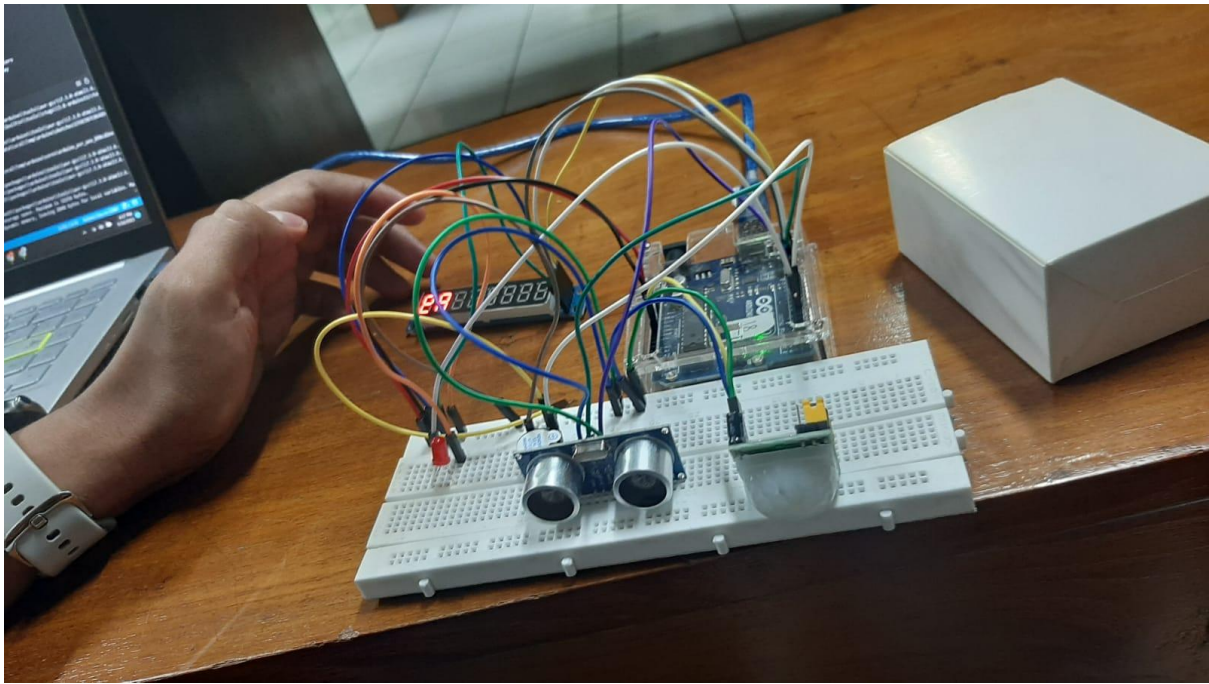


*Figure 5. Testing in Proteus*

Dari display yang tertera di MAX7219, 'dist' merupakan distance, 52 merupakan jarak terukur objek yang terdeteksi sensor, dimana satuan untuk jarak tersebut yaitu centimeter. Kemudian dapat terlihat bahwa LED berhasil menyala, dan pada saat yang bersamaan buzzer juga mengeluarkan suara.

Setelah melakukan simulasi di Proteus, maka program dijalankan pada Arduino IDE dengan menyambungkannya ke rangkaian asli.





*Figure 6. Testing with real components*

Dari hasil pengujian yang dilakukan, terdapat beberapa hal yang menjadi catatan penting. Diantaranya :

- a. Sensor jarak berhasil mendeteksi keberadaan objek pada jarak maksimal 7 meter.
- b. Sensor PIR berhasil mendeteksi perubahan radiasi inframerah dari makhluk hidup dengan sensitivitas serta delay time yang dapat diatur secara langsung dan dinamis.
- c. LED dan Buzzer menyala secara bersamaan pada saat sebuah objek bergerak di lingkungan kedua sensor.

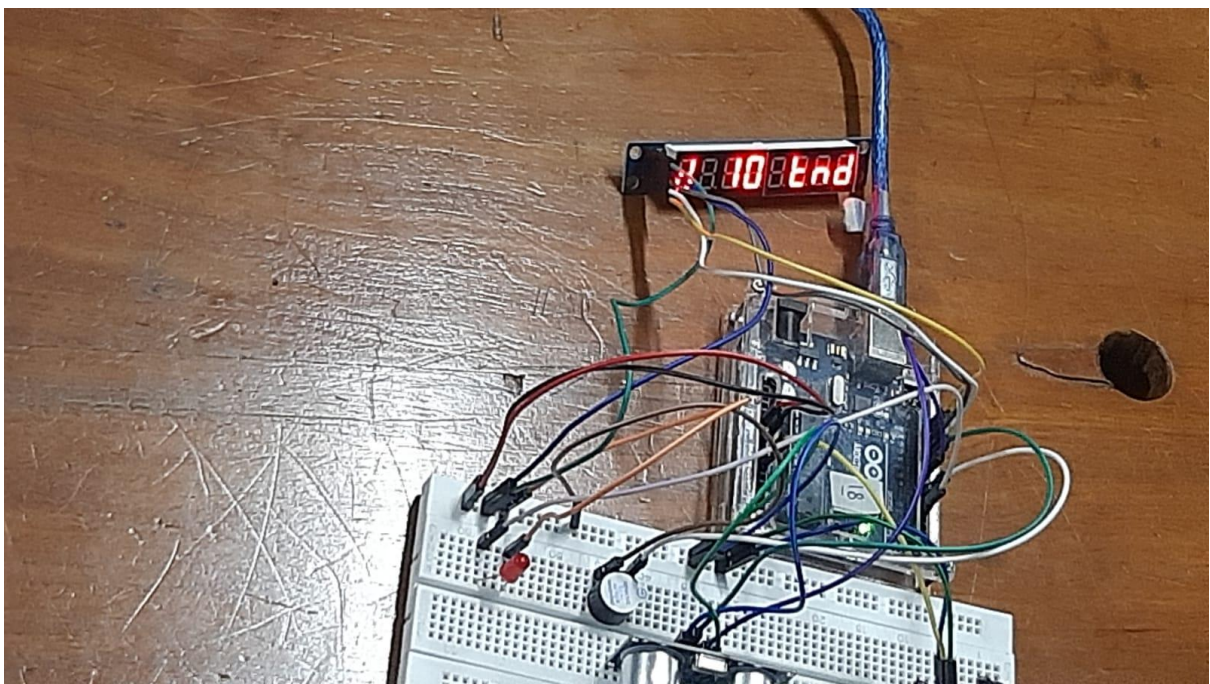
Meski begitu, selama masa pengujian, terdapat beberapa kendala yang dialami oleh Kelompok A1 diantaranya seperti program hanya dapat berfungsi sebagaimana mestinya pada proteus dan mengalami error ketika dijalankan di rangkaian asli. Hal ini dikarenakan beberapa komponen yang digunakan mengalami kerusakan atau kecacatan seperti kerusakan jumper, kerusakan MAX7219, kualitas sensor PIR yang sudah tidak baik, penempatan transistor pada buzzer yang mengakibatkan buzzer mengeluarkan suara yang lebih kecil daripada yang seharusnya bahkan terkadang tidak menyala sama sekali sehingga menghambat program berfungsi dengan baik.

Selain itu, penggunaan AC/DC Adaptor sebagai catu daya dari alat ini membuat alat berjalan tidak sebagaimana mestinya. Hal ini terjadi ketika arduino mendapatkan sumber daya dari Laptop dan berjalan dengan baik, namun ketika diganti menggunakan catu daya AC/DC Adaptor, display pada MAX7219 mengalami error atau kerusakan. Bahkan sensor terkadang salah mendeteksi.

Solusi yang kami lakukan adalah dengan mengganti komponen atau meminjam komponen dari kelompok lain.

### e. 3.2 RESULT

Berikut merupakan hasil dari simulasi yang dijalankan langsung pada komponen rangkaian asli :



Berdasarkan cara kerja dan *flow* kode, kondisi di atas adalah kondisi dimana program baru saja *dirun* sehingga Arduino menyala, namun LED seven-segment masih belum dapat menampilkan hasil output yang sesuai, Sedangkan untuk buzzer dan LED merah menyala ketika sensor PIR dan sensor HC-SR04 mendeteksi pergerakan di sekitar.

### f. 3.3 EVALUATION

Program alat ini disusun menggunakan bahasa assembly secara keseluruhannya, *running* dari program itu sendiri dilakukan pada Arduino IDE yang terpasang pada komputer pengujian. Sedangkan untuk mendesain dan menguji coba rangkaian prototipe, digunakan software bernama Proteus dan website pengujian, [wokwi.com](http://wokwi.com). Setelah program dijalankan, maka Arduino IDE akan mengkompilasi program untuk mengecek apakah terdapat kesalahan logika *syntax error*. Setelah berhasil dikompilasi, program di *upload* ke mikrokontroler ATmega328P pada Arduino menggunakan koneksi USB. Setelah dinyalakan, mikrokontroler akan melakukan inisialisasi. Mikrokontroler mengkonfigurasi register arah data untuk menentukan status input/output dari pin yang diperlukan.

Rangkaian akan melakukan pendeteksian pergerakan di lingkungan sekitar sensor, dalam hal ini, jika sensor diletakkan pada pintu, maka PIR Sensor atau sensor gerak akan mendeteksi pergerakan objek, di mana pun dalam jarak pandangnya, dengan mengukur perubahan cahaya inframerah yang dipancarkan atau dipantulkan oleh objek tersebut. Mikrokontroler membaca status sensor PIR yang terhubung ke pin input yang ditentukan. Program ini memeriksa apakah pin sensor PIR high atau low menggunakan operasi AND bitwise. Jika gerakan terdeteksi, pin sensor akan menjadi high, sinyal dari sensor akan dikirimkan ke LED dan Buzzer untuk pengaktifan kedua aktuator tersebut. Buzzer akan menerima getaran listrik untuk kemudian menghasilkan getaran suara. Begitu juga dengan LED disini yang akan menerima arus listrik dari sumber tegangan atau power supply berupa baterai yaitu powerbank. Hal yang sama juga terjadi pada sensor jarak yang akan mendeteksi objek pada jarak pandangnya, kemudian mengkalkulasikan jarak tersebut dan menampilkannya pada MAX7219.

Jika tidak ada gerakan yang terdeteksi oleh sensor PIR, program akan berlanjut ke bagian dimana kode yang dieksekusi akan mematikan Buzzer dan LED. Program ini terus mengulang langkah-langkah di atas, berulang kali memeriksa gerakan dan memperbarui status buzzer dan LED berdasarkan input sensor. Hal ini memungkinkan sistem untuk terus memantau ruangan untuk setiap perubahan gerakan. Sebagaimana rangkaian ini telah dijelaskan, rangkaian mampu memberikan output sesuai dengan tujuan dan kriteria keberhasilan alat sehingga dalam hal ini, alat dinyatakan berfungsi sebagaimana mestinya.

## CHAPTER 4

### CONCLUSION

Kesimpulannya, Room Security System yang menggunakan mikrokontroler ATmega328P, sensor PIR, HC-SR04, buzzer, dan LED memberikan solusi yang sederhana namun efektif untuk memonitor dan mendeteksi gerakan di dalam ruangan. Sistem ini beroperasi dengan terus menerus memeriksa jarak objek, dan status sensor PIR untuk mendeteksi gerakan apa pun. Ketika gerakan terdeteksi, maka akan memicu buzzer dan LED untuk membunyikan alarm dan memberikan indikasi visual. Sebaliknya, ketika tidak ada gerakan yang terdeteksi, buzzer dan LED tetap dinonaktifkan.

Integrasi sensor PIR memberikan deteksi gerakan yang *reliable*, membuat sistem ini sangat responsif terhadap gerakan apa pun di dalam area yang dipantau. Buzzer dan LED berfungsi sebagai mekanisme peringatan yang efektif, dengan cepat memberitahu pengguna tentang potensi pelanggaran keamanan atau perubahan lingkungan ruangan. Fleksibilitas komponen perangkat keras, seperti breadboard, kabel jumper, dan power connector, memfasilitasi perakitan, modifikasi, dan pengujian sistem keamanan. Hal ini membuatnya cocok untuk pembuatan prototipe dan implementasi dalam skenario asli. Secara keseluruhan, Room Security System menawarkan solusi yang hemat biaya dan mudah diakses untuk meningkatkan keamanan ruangan.

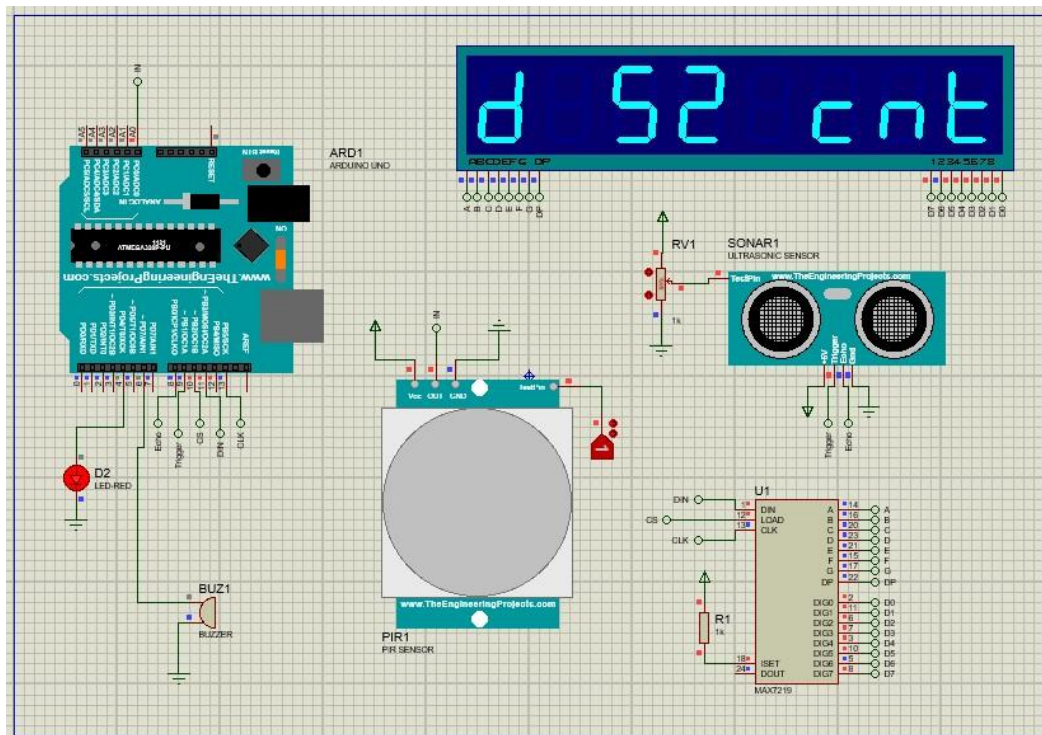
## REFERENCES

- [1] Jeremy S Cook  
Jeremy S. Cook has a BSME from Clemson University, J. S. Cook, and Jeremy S. Cook has a BSME from Clemson University, “Understanding Active & Passive Infrared Sensors (PIR) and their uses,” Arrow.com,
- [2] Jeremy S Cook  
Jeremy S. Cook has a BSME from Clemson University, J. S. Cook, and Jeremy S. Cook has a BSME from Clemson University, “Understanding Active & Passive Infrared Sensors (PIR) and their uses,” Arrow.com,  
[https://www.arrow.com/en/research-and-events/articles/understanding-active-and-passive-infrared-sensors#:~:text=How%20Do%20PIR%20Sensors%20Work,\)%2C%20the%20sensor%20will%20engage.](https://www.arrow.com/en/research-and-events/articles/understanding-active-and-passive-infrared-sensors#:~:text=How%20Do%20PIR%20Sensors%20Work,)%2C%20the%20sensor%20will%20engage.) (accessed May 16, 2023).
- [3] M. Bala, “PIR motion sensor interface with AVR-microcontroller ATMEGA32,” Gadgetronicx, <https://www.gadgetronicx.com/pir-motion-sensor-interface-with-avr/> (accessed May 16, 2023).
- [4] “Assembly via Arduino (part 21) - HC SR04 sensor,” YouTube, <https://youtu.be/dIIBG7Qz-E> (accessed May 16, 2023).

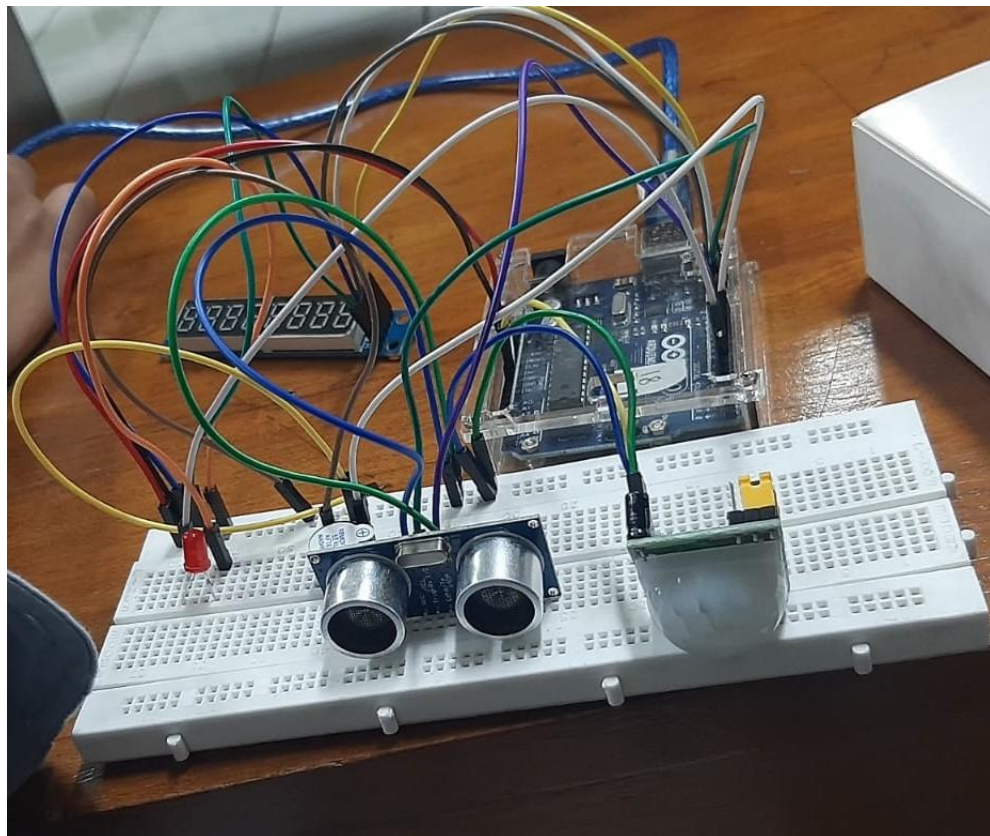
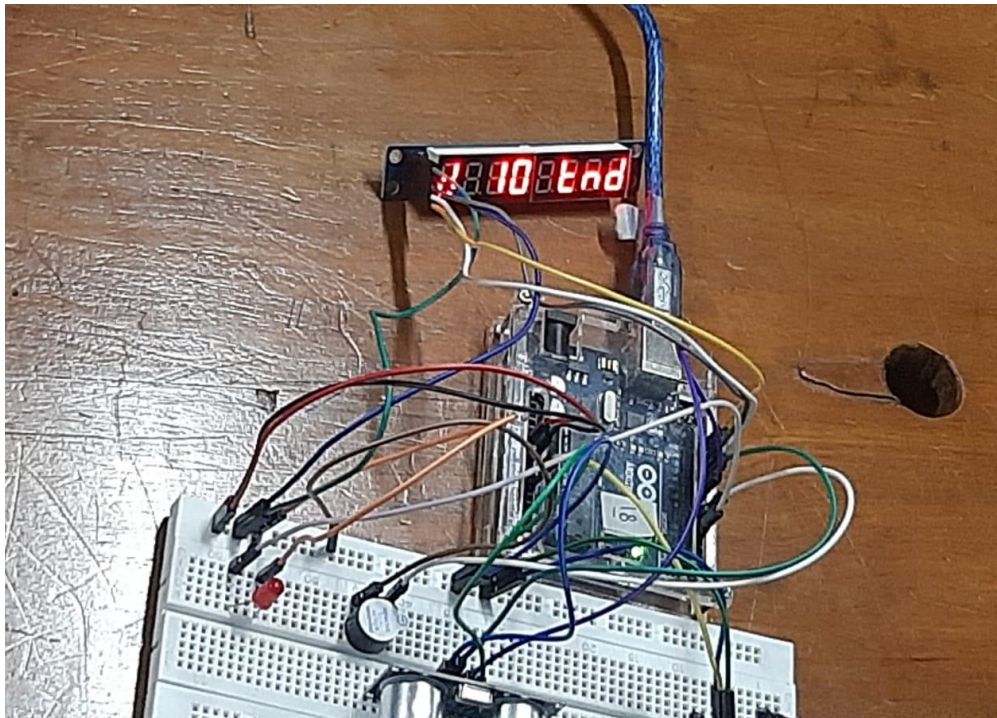


## APPENDICES

### Appendix A: Project Schematic



### Appendix B: Documentation



## Appendix C: Related Links

- [Testing Video and Photos](#)

