

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



SECURITY SYSTEM SENSOR

GROUP 6

MUHAMMAD DAFFA RIZKYANDRI

2206829194

RAJA YONANDRO RUSLITO

2206059553

RYAN SAFA TJENDANA

2206826835

YASMIN DEVINA SINURAYA

2206817244

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 praktikum Sistem Siber-Fisik, yang berjudul "Security System Sensor", tepat pada waktunya. Laporan ini disusun untuk memberikan pemahaman kepada pembaca tentang bagaimana carakkerja 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 asisten laboratorium pembimbing, Kak Alif Masum, serta 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 "Security System Sensor" 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, 26 Mei 2024

Group 6

DAFTAR ISI

BAB 1	1
PENDAHULUAN	1
1.1. Perumusan Masalah	1
1.2. Proposal Solusi.....	1
1.3. Penggunaan Modul	2
1.4. Pembagian Tugas	2
BAB 2	2
IMPLEMENTASI.....	2
2.1. Skema dan Desain Perangkat Keras	2
2.2. Pengembangan Perangkat Lunak.....	4
2.3. Penghubungan Perangkat Keras dan Perangkat Lunak	8
BAB 3	10
PERCOBAAN DAN ANALISA	10
3.1. Pengujian.....	10
3.2. Hasil	12
3.3. Analisa Kesalahan.....	13
BAB 4	14
KESIMPULAN.....	14
REFERENSI	15
LAMPIRAN.....	16

BAB 1

PENDAHULUAN

1.1. Perumusan Masalah

Keamanan rumah dan properti merupakan aspek penting dalam kehidupan sehari-hari. Banyak orang mencari cara efektif untuk melindungi rumah mereka dari pencurian atau intrusi, dan salah satu solusi yang banyak digunakan adalah sistem keamanan berbasis sensor yang dapat mendeteksi pergerakan mencurigakan dan memberikan peringatan dini kepada pemilik rumah. Keamanan rumah sering kali terganggu oleh pencurian dan intrusi yang tidak terdeteksi serta keterbatasan sistem keamanan konvensional yang seringkali mahal dan rumit untuk dipasang serta dioperasikan. Berdasarkan latar belakang dan identifikasi masalah tersebut, permasalahan yang akan dipecahkan dalam program Arduino "Security System Sensor" adalah bagaimana merancang dan mengimplementasikan sistem keamanan yang sederhana dan efektif menggunakan sensor PIR untuk mendeteksi pergerakan, bagaimana memprogram Arduino untuk memproses sinyal dari sensor PIR dan mengaktifkan buzzer sebagai alarm ketika mendeteksi pergerakan mencurigakan, dan bagaimana memastikan bahwa sistem ini dapat berfungsi dengan baik dalam kondisi lingkungan yang berbeda serta meminimalkan kesalahan deteksi (false alarm).

Tujuan dari program Arduino "Security System Sensor" ini adalah mengembangkan sistem keamanan rumah yang menggunakan sensor PIR untuk mendeteksi pergerakan, meningkatkan keamanan dengan memberikan peringatan dini melalui bunyi buzzer ketika mendeteksi pergerakan yang mencurigakan, dan menyediakan solusi keamanan yang mudah diimplementasikan dan terjangkau bagi pemilik rumah.

1.2. Proposal Solusi

Implementasi dari program ini diharapkan dapat memberikan beberapa manfaat, antara lain meningkatkan rasa aman dan nyaman bagi pemilik rumah, menyediakan sistem keamanan yang efisien dan terjangkau, serta memudahkan pemantauan dan deteksi dini terhadap potensi ancaman keamanan. Proyek ini akan mencakup pemilihan dan konfigurasi hardware seperti Arduino, sensor PIR, dan buzzer, pengembangan kode program untuk Arduino yang mampu mengintegrasikan sensor PIR dan buzzer, serta pengujian dan evaluasi sistem dalam berbagai kondisi untuk memastikan keandalannya. Dengan demikian, proposal solusi ini diharapkan dapat memberikan gambaran yang jelas mengenai tujuan dan langkah-langkah yang akan diambil dalam pengembangan sistem keamanan berbasis sensor PIR menggunakan Arduino.

1.3. Penggunaan Modul

- 1) Modul 2 : Introduction to AVR Assembly
- 2) Modul 4 : Serial Port
- 3) Modul 5 : Arithmetic
- 4) Modul 6 : Timer
- 5) Modul 8 : SPI & I2C
- 6) Modul 9 : Sensor Interfacing

1.4. Pembagian Tugas

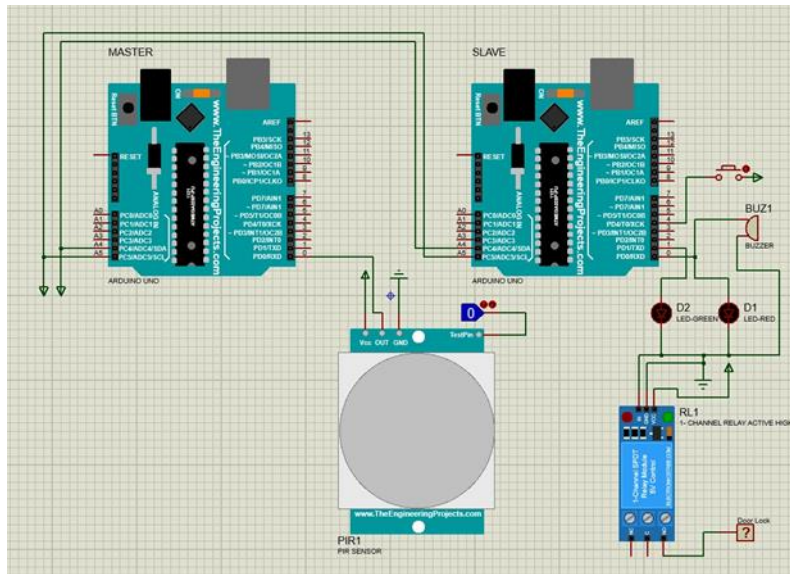
1. Ryan Safa Tjendana :
 - Pembuatan Proteus
 - Pembuatan Kode
 - Pembuatan Laporan & Flowhart
2. Muhammad Daffa Rizkyandri:
 - Pembuatan Kode
 - Pembuatan Proteus
 - Pembuatan Flowchart
3. Yasmin Devina Sinuraya:

- Pembuatan Rangkaian
 - Pembuatan Laporan
4. Raja Yonandro Ruslito
- Pembuatan Laporan
 - Pembuatan Rangkaian

BAB 2

IMPLEMENTASI

2.1. Skema dan Desain Perangkat Keras



Gambar 1. Rangkaian Proteus

Pembuatan alat dimulai dengan merancang skema atau prototipe menggunakan software Proteus. Skema ini dibuat untuk membantu kelompok dalam merakit alat secara virtual sebelum merakit komponen asli. Tujuannya adalah untuk mengurangi risiko kerusakan pada komponen asli yang dapat terjadi akibat kesalahan rangkaian atau integrasi yang tidak sempurna.

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

- 1) Dua buah Arduino UNO
 - Arduino UNO (Master)

Berfungsi sebagai pengendali utama yang memproses sinyal dari sensor PIR. Ketika sensor PIR mendeteksi pergerakan, Arduino Master akan mengirimkan sinyal ke Arduino Slave.

- Arduino UNO (Slave)

Berfungsi untuk menerima sinyal dari Arduino Master dan mengendalikan perangkat output seperti LED dan buzzer. Arduino Slave mengaktifkan LED merah dan buzzer ketika pergerakan terdeteksi dan mengaktifkan LED hijau ketika tidak ada pergerakan.

2) Satu buah Sensor PIR

Sensor PIR (Passive Infrared Sensor) digunakan untuk mendeteksi pergerakan di sekitarnya. Sensor ini mengirimkan sinyal ke Arduino Master ketika mendeteksi adanya pergerakan. Sensor PIR biasanya memiliki tiga pin: Vcc (power supply), GND (ground), dan OUT (output signal).

3) Satu buah Relay

Relay digunakan sebagai saklar elektronik yang diaktifkan oleh sinyal dari Arduino Slave. Relay ini mengendalikan perangkat eksternal yang memerlukan arus lebih besar daripada yang bisa disediakan langsung oleh pin Arduino. Dalam konteks ini, relay dapat digunakan untuk mengendalikan sistem keamanan yang lebih besar atau perangkat lainnya yang terhubung.

4) Satu buah Buzzer

Buzzer digunakan sebagai alarm untuk memberikan peringatan suara ketika sensor PIR mendeteksi pergerakan. Buzzer diaktifkan oleh sinyal dari Arduino Slave setelah menerima sinyal dari Arduino Master.

5) Satu buah LED Hijau

LED hijau digunakan sebagai indikator bahwa tidak ada pergerakan yang terdeteksi oleh sensor PIR. LED ini diaktifkan oleh Arduino Slave sebagai bagian dari umpan balik visual untuk menunjukkan kondisi aman.

6) Satu buah LED Merah

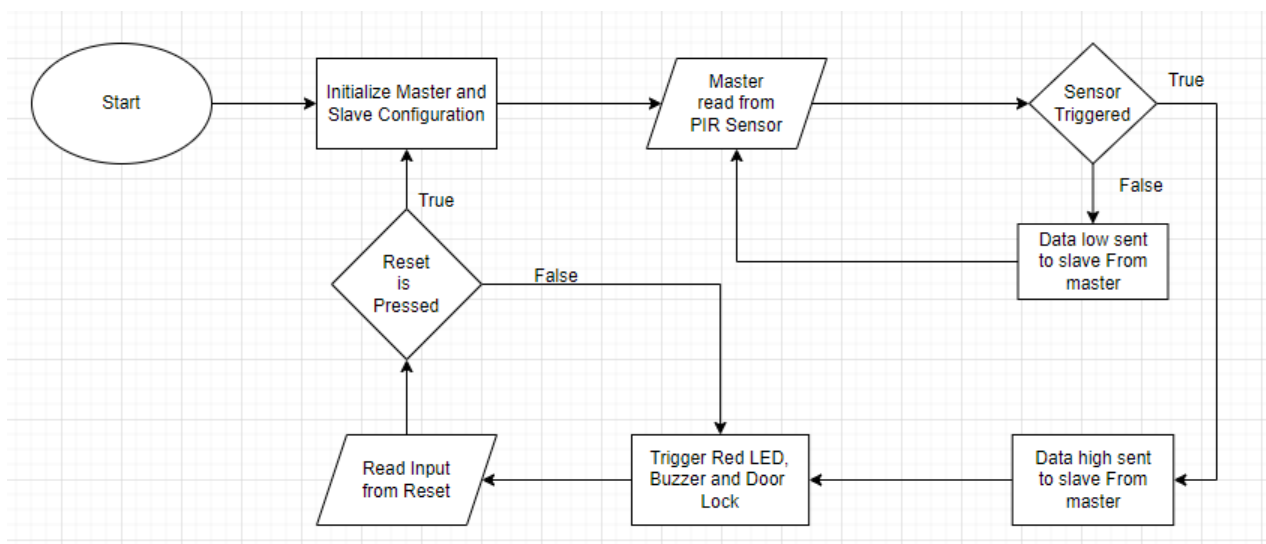
LED merah digunakan sebagai indikator bahwa pergerakan telah terdeteksi oleh sensor PIR. LED ini diaktifkan oleh Arduino Slave ketika menerima sinyal dari Arduino Master setelah sensor PIR mendeteksi pergerakan, memberikan umpan balik visual bahwa ada pergerakan yang terdeteksi.

7) Satu buah Button

Button digunakan sebagai indikator apabila situasi sudah aman dan bertujuan untuk mematikan Buzzer yang sebelumnya dijadikan sebagai alarm dan melakukan reset agar rangkaian dapat melakukan tugasnya Kembali.

2.2. Pengembangan Perangkat Lunak

Untuk mempermudah penyusunan cara kerja alat, perlu dibuat kerangka kerja menggunakan website draw.io sebagai berikut:



Gambar 2. Flowchart Program

Dengan demikian terciptalah dua buah kode *assembly* untuk masing- masing Arduino, yaitu Arduino Master dan Arduino Slave. Arduino Master berguna untuk mengirimkan sinyal dari pembacaan pergerakan pada Sensor PIR ke Arduino Slave. Berikut adalah kode untuk

Arduino Master:

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
.global uart_init
.global uart_transmit
.global RESET

; Initialize UART
uart_init:
    ldi r16, hi8(103)
    STS UBRR0H, r16
    ldi r16, lo8(103)
    STS UBRR0L, r16
    ldi r16, 1<<RXEN0 | 1<<TXEN0
    STS UCSR0B, r16
    ldi r16, 0x86
    STS UCSR0C, r16
    ret

; Send data over UART
uart_transmit:
    LDS r18, UCSR0A
    sbrc r18, UDRE0
    rjmp uart_transmit
    STS UDR0, r16
    ret

; Main program
    rjmp RESET

RESET:
    ; Initialize UART
    rcall uart_init

    ; Set PIR sensor as input (A0), Button as input (A1)
    ldi r16, 0x00
    out PORTD, r16

    ; Main loop
main_loop:
    ; Read Button (A1)
    sbic PORTC, 1
    rjmp main_loop
```

```

; Button pressed, read PIR sensor (A0)
sbic PORTC, 0
rjmp pir_triggered

; PIR sensor not triggered
ldi r16, '0'
rcall uart_transmit
RCALL delay_sec
rjmp main_loop

pir_triggered:
; PIR sensor triggered
ldi r16, '1'
rcall uart_transmit
rjmp main_loop

delay_sec: ;1s delay
LDI R20, 255
17:
LDI R21, 255
18:
LDI R22, 80
19:
DEC R22
BRNE 19
DEC R21
BRNE 18
DEC R20
BRNE 17
RET

```

Sementara itu, untuk menerima sinyal yang telah dikirimkan oleh Arduino Master, Arduino Slave akan ter-*trigger* untuk berjalan. Berikut adalah kode untuk Arduino Slave:

```

#define __SFR_OFFSET 0x00
#include "avr/io.h"
.global uart_receive
.global uart_init
.global RESET

; Initialize UART
uart_init:

```

```

    ldi r16, hi8(103)
    STS UBRR0H, r16
    ldi r16, lo8(103)
    STS UBRR0L, r16
    ldi r16, 1<<RXEN0 | 1<<TXEN0
    STS UCSR0B, r16
    ldi r16, 0x86
    STS UCSR0C, r16
    ret

; Receive data over UART
uart_receive:
    LDS r18, UCSR0A
    sbrs r18, RXC0
    rjmp uart_receive
    LDS r16, UDR0
    ret

; Main program
    rjmp RESET

RESET:
    ; Initialize UART
    rcall uart_init

    ; Set LEDs and Buzzer as output (Green D2 -> A0, Red D1 -> A1, Buzzer -> A2)
    ldi r16, 0x07
    out PORTD, r16

    ; Main loop
main_loop:
    rcall uart_receive
    cpi r16, '0'
    breq disable_alert

    ; Enable alert (PIR sensor triggered)
    ldi r16, 0x06
    out PORTD, r16
    rjmp main_loop

disable_alert:
    ; Disable alert
    ldi r16, 0x01
    out PORTD, r16
    rjmp main_loop

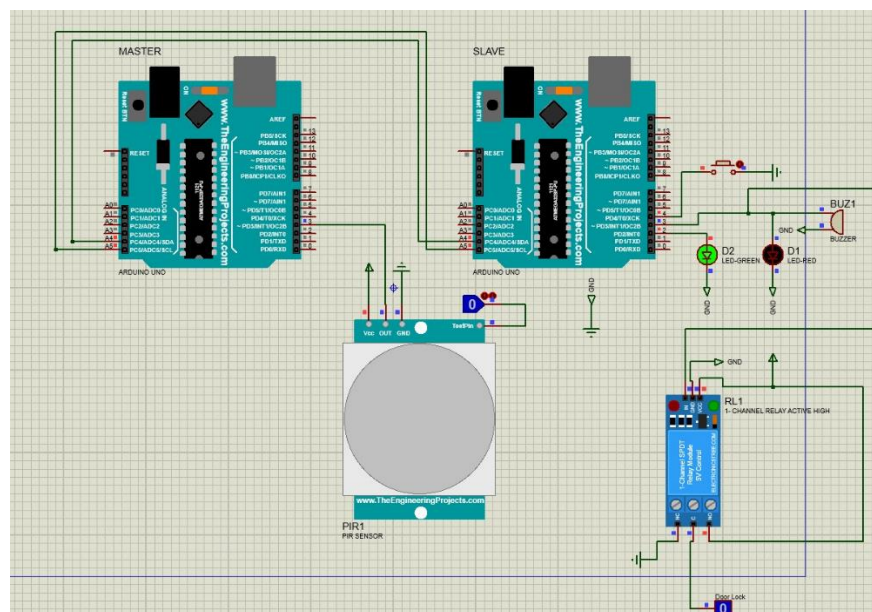
```

2.3. Penghubungan Perangkat Keras dan Perangkat Lunak

Penghubungan perangkat keras dengan perangkat lunak dilakukan dengan arduino, dimana arduino akan berperan sebagai perantara antar semua perangkat keras seperti sensor, buzzer, dan LED untuk saling berkomunikasi menggunakan perangkat lunak pada arduino yang telah diprogram agar sesuai dengan keinginan. Perangkat lunak tersebut diprogram bagaimana suatu sistem dari beberapa perangkat keras akan berperan menjadi door security system yang dimana akan menggunakan PIR sensor untuk menangkap gerakan yang ada, jika PIR tertrigger maka buzzer dan led akan menyala, dan door lock pun menjadi aktif selama tombol reset tidak dipencet, jika dipencet maka program akan memulai untuk membaca PIR sensor kembali.

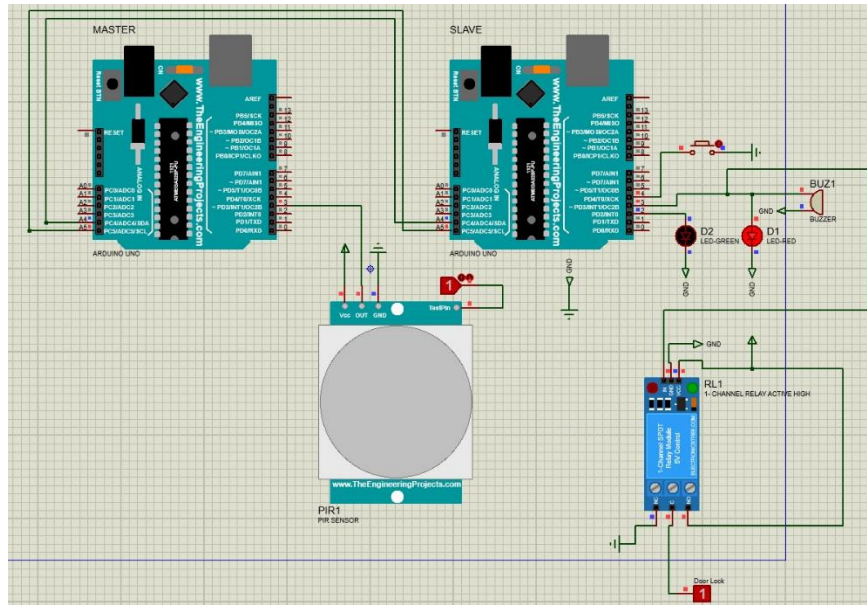
Hasil dalam simulasi Proteus

Kondisi Led pada saat pertama dinyalakan:



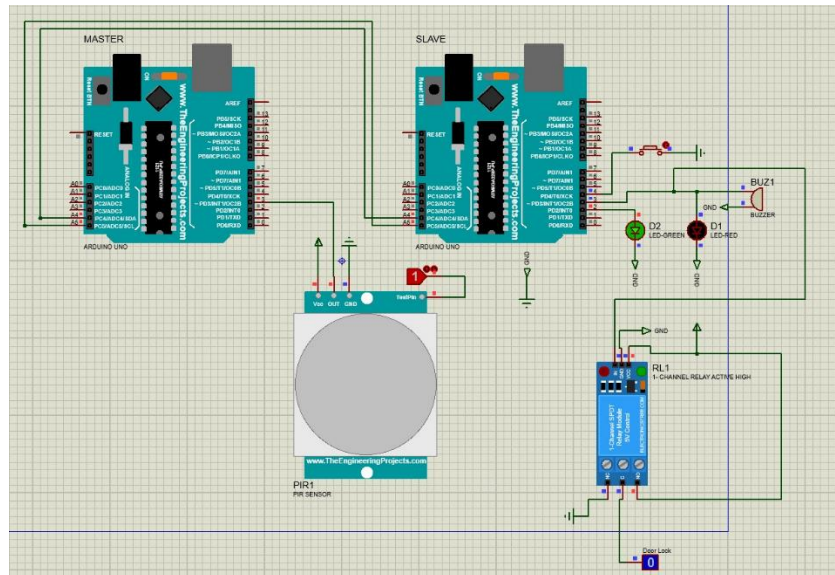
Gambar 3. Kondisi pada kondisi default

Kondisi ketika sensor mendeteksi Gerakan:



Gambar 4. Kondisi apabila trigger

Kondisi Setelah Tombol Reset ditekan



Gambar 5. Kondisi apabila reset

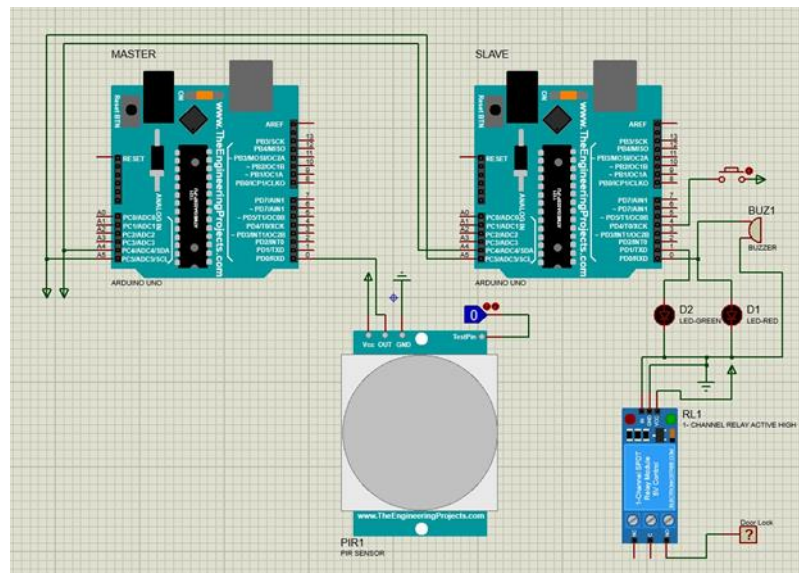
BAB 3

PERCOBAAN DAN ANALISA

3.1. Pengujian

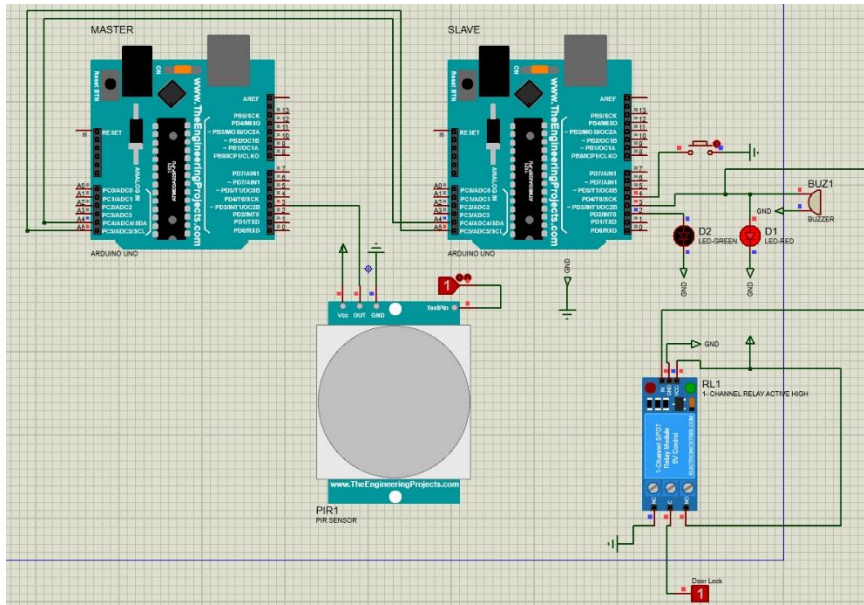
3.1.1 Pengujian Rangkaian Pada Proteus

Pengujian rangkaian pertama dilakukan menggunakan software Proteus terlebih dahulu untuk memastikan hubungan dari hardware dan software alat sudah dapat bekerja dengan baik. Berikut adalah skema rangkaian yang telah disusun pada Proteus:



Gambar 6. Rangkaian pada Proteus

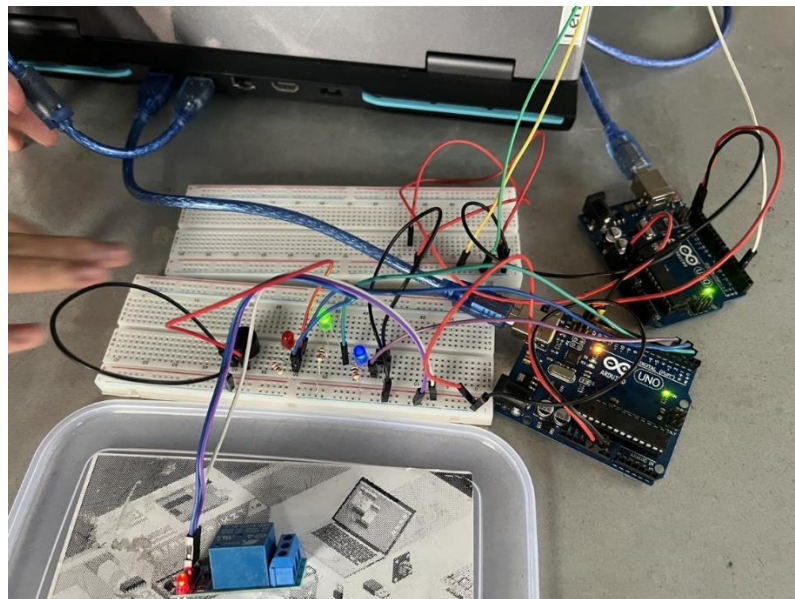
Pada pengujian ini kami menggunakan TestPin untuk sensor PIR yang nantinya output tersebut akan dibawa ke arduino Slave untuk memicu aktifnya Buzzer. Output dari skema di Proteus menjadi sebagai berikut:



Gambar 7. Pengujian pada Proteus

3.1.2 Pengujian Pada Rangkaian Fisik

Setelah melakukan uji coba di Proteus, program tersebut kemudian dijalankan pada Arduino menggunakan Arduino IDE dengan menghubungkannya ke rangkaian asli.



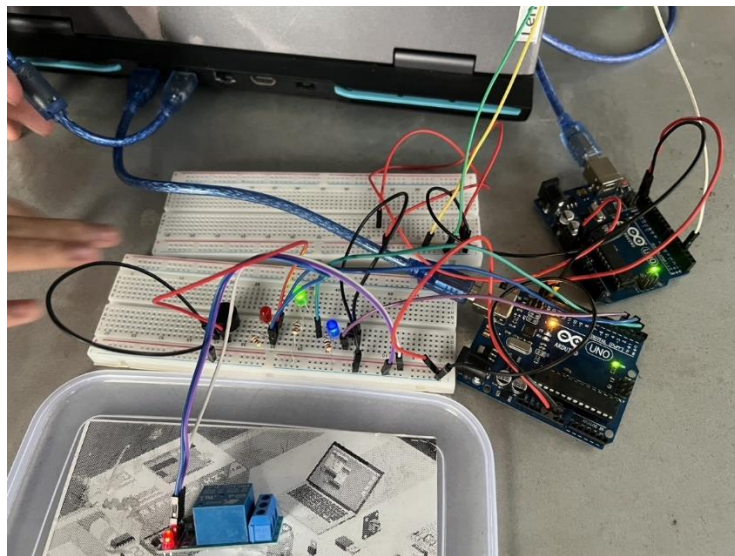
Gambar 8. Pengujian pada rangkaian asli

Berdasarkan hasil pengujian tersebut, terdapat beberapa hal yang patut dijadikan sebuah catatan penting, yaitu:

1. Sensor PIR tidak berhasil mendeteksi perubahan radiasi inframerah dengan sensitivitas dan delay time yang dapat diatur secara langsung dan dinamis.
2. Arduino UNO yang berperan sebagai Arduino Master pada proyek ini mengalami kerusakan kecil yang mengakibatkan kode dan output sinyal yang seharusnya dikirimkan ke Arduino Slave tidak bekerja.
3. LED dan Buzzer dapat menyala tetapi keduanya harus berada dalam kondisi terisolasi dari Arduino Master.

3.2. Hasil

Berikut adalah hasil dari pengujian yang dilakukan pada rangkaian Arduino Master-Slave asli:



Gambar 9. Hasil pada rangkaian asli

Dikarenakan terdapat beberapa komponen yang memiliki kondisi kurang baik, hasil pada rangkaian asli yang kami dapat tidak sesuai dengan hasil yang didapatkan pada rangkaian Proteus. LED dan Buzzer dapat menyala dengan baik pada awalnya, namun seiring berjalannya percobaan

kemungkinan terjadi korslet pada Buzzer yang mengakibatkan suara alarm pada Buzzer menjadi kurang baik.

3.3. Analisa Kesalahan

Hasil pada rangkaian asli tidak sesuai seperti pada rangkaian di Proteus, hal ini dapat terjadi karena beberapa faktor. Salah satu contohnya adalah kondisi alat yang digunakan. Pada rangkaian asli, digunakan 2 buah Arduino yang berperan sebagai Master dan Slave. Tetapi Arduino Master mengalami kerusakan, contoh kerusakan bahwa Arduino Master mengalami kerusakan yaitu terkadang Arduino Master harus dipegang/ditekan agar dapat berfungsi. Selain itu, ketika kami mencoba untuk membaca data yang dikirimkan ke serial monitor, data tidak berhasil dibaca. Hal ini disebabkan karena kualitas dari Arduino Master itu sendiri.

Contoh faktor lainnya yaitu adanya kemungkinan bahwa Buzzer mengalami korslet dan sensor PIR yang tidak berfungsi. Buzzer mengalami korslet dapat disebabkan karena besar arus atau tegangan listrik yang tidak sesuai. Buzzer korslet ini juga menyebabkan LED merah tidak dapat menyala, karena aliran antara LED merah dan buzzer sama, sehingga ketika Buzzer dicabut barulah LED merah dapat menyala. Sensor PIR juga kemungkinan tidak berfungsi, karena data dari sensor tersebut tidak bisa terbaca.

BAB 4

KESIMPULAN

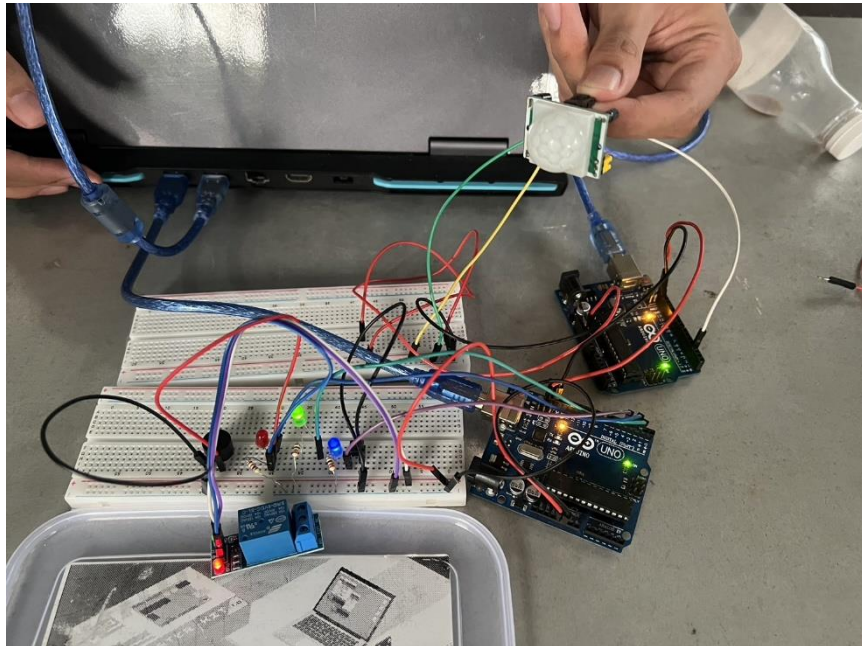
Kesimpulan yang dapat ditarik dari proyek “Security Sensor System” yang dibuat menggunakan mikrokontroler ATmega328P, sensor PIR, Buzzer, Relay, dan LED ini adalah program ini memberikan solusi yang sederhana untuk menjadi sebuah alat untuk mendeteksi adanya kejanggalan yang terdapat pada suatu ruangan. Program ini dapat beroperasi secara kontinu untuk memeriksa status inframerah dari sensor PIR. Ketika terdapat gerakan yang tidak biasa terdeteksi, Buzzer dan LED akan menyala memberikan indikasi audio dan visual. Namun, apabila tidak terdapat gerakan yang terdeteksi, Buzzer dan LED akan tetap berada dalam kondisi non-aktif.

Integrasi sensor PIR memberikan deteksi gerakan yang andal, membuat sistem ini sangat tanggap terhadap setiap gerakan di dalam area yang dipantau. Buzzer dan LED berfungsi sebagai mekanisme peringatan yang efisien, dengan cepat menginformasikan pengguna tentang potensi pelanggaran keamanan atau perubahan kondisi ruangan. Fleksibilitas komponen perangkat keras seperti breadboard, kabel jumper, dan konektor daya memudahkan perakitan, modifikasi, dan pengujian sistem keamanan. Hal ini menjadikannya cocok untuk pembuatan prototipe dan implementasi dalam skenario nyata. Secara keseluruhan, Room Security System menawarkan solusi yang ekonomis dan mudah diakses untuk meningkatkan keamanan ruangan.

REFERENSI

- [1] “Motion detection using HC-SR501 sensor,” YouTube, <https://www.youtube.com/watch?v=wQwHUZmk4Ms> (accessed May 28, 2024).
- [2] “Arduino - LCD I2C: Arduino tutorial,” Arduino Getting Started, <https://arduinogetstarted.com/tutorials/arduino-lcd-i2c> (accessed May 28, 2024).
- [3] K. Magdy, “Arduino I2C Slave [Tutorial & Example Code],” DeepBlue, <https://deepbluembedded.com/arduino-i2c-slave/> (accessed May 28, 2024).
- [4] Lady Ada, “Pir motion sensor,” Adafruit Learning System, <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work> (accessed May 28, 2024).
- [5] T. Agarwal, “Passive Infrared Sensor (PIR) working with applications,” ElProCus, <https://www.elprocus.com/passive-infrared-pir-sensor-with-applications/> (accessed May 28, 2024).

LAMPIRAN



LINK GITHUB PROYEK:

<https://github.com/rvansati/ProvekAkhirSSF-Kelompok6>