



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

AUTOMATED JAIL DOOR CONTROLLER

KELOMPOK A10

RIZKI AWANTA JORDHIE	2106655034
MUHAMMAD RIZKY UTOMO	2106731320
FARRAS RAFI' PERMANA	2106700990
SYAUQI AULIYA MUHAMMAD	2106707201

PREFACE

Kami persembahkan laporan proyek ini yang berjudul *Automated Jail Door Control* yang dikembangkan sebagai proyek dari Praktikum Sistem Siber Fisik. Tujuan proyek kami adalah merancang dan menerapkan sistem pengendalian otomatis berbasis Arduino untuk pintu penjara, dengan penekanan pada keamanan, efisiensi, serta keselamatan baik narapidana dan staf, dengan mengimplementasikan berbagai komponen listrik yang telah kami pelajari selama semester ini.

Pendekatan kami melibatkan riset yang mendalam mengenai sistem pengendalian pintu penjara agar dapat dengan cermat merencanakan dan mendesain segala fungsionalitas dan mekanisme pada proyek kami. Dengan menggunakan Arduino sebagai basis, kami merancang proyek kami dengan mengintegrasikan sensor, motor, button, dan modul komunikasi serial. Melalui pengujian yang ketat, kami memastikan keberhasilan dan efektivitas dari solusi yang kami rancang, dan laporan proyek ini pun berfungsi sebagai dokumentasi singkat tentang perkembangan proyek kami, tantangan yang dihadapi, pembagian tugas dan tanggung jawab, serta solusi akhir yang ditemukan.

Kami ingin mengucapkan rasa terima kasih kepada seluruh tim asisten laboratorium digital dan teman-teman kami atas bimbingan dan dukungan yang mereka berikan. Proyek ini merupakan hasil kerja kolektif kami, dan komitmen kami dalam mengembangkan sistem pengendalian pintu penjara yang aman dan efisien.

Depok, 15 Mei 2023

Kelompok 10

TABLE OF CONTENTS

BAB 1.....	4
PENDAHULUAN.....	4
1.1 LATAR BELAKANG.....	4
1.2 SOLUSI.....	5
1.3 KRITERIA KEBERHASILAN.....	5
1.4 PERAN DAN TANGGUNG JAWAB.....	5
1.5 LINIMASA DAN TARGET.....	5
BAB 2.....	7
IMPLEMENTASI.....	7
2.1 DESAIN HARDWARE.....	7
2.2 SOFTWARE.....	7
2.3 INTEGRASI HARDWARE DAN SOFTWARE.....	8
BAB 3.....	9
EVALUASI DAN TESTING.....	9
3.1 TESTING.....	9
3.2 HASIL.....	9
3.3 EVALUASI.....	10
BAB 4.....	11
KESIMPULAN.....	11

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Kami menemukan bahwa sistem pengendalian tradisional pintu penjara seringkali kurang dalam hal otomatisasi, keamanan, dan efisiensi yang krusial untuk menciptakan lingkungan yang ideal baik bagi para narapidana dan staf. Pengoperasian manual dengan tenaga manusia di penjara rentan terhadap kesalahan dan memerlukan lebih banyak usaha dan waktu daripada yang sebenarnya diperlukan. Menyadari hal tersebut, dengan kemajuan teknologi di masa sekarang, kami memutuskan untuk mendedikasikan proyek kami untuk masalah ini dengan mengimplementasikan desain Arduino, tombol, LCD, komunikasi serial, motor dan lain sebagainya untuk memaksimalkan keamanan dan efisiensi sambil menampilkan informasi status pintu secara real-time.

Dengan mengintegrasikan otomatisasi berbasis Arduino, kami bermaksud untuk mengatasi tantangan yang muncul mekanisme operasional tradisional dalam fasilitas penjara. Tantangan tersebut mulai dari menghilangkan *delay* dan ketidakefisienan yang dapat mempengaruhi keseluruhan alur kerja di penjara, meningkatkan pemantauan dan keamanan dengan menampilkan status pintu sel secara *real-time*, serta menjaga kesehatan dan keselamatan narapidana dan staf.

Melalui proyek ini, kami ingin menunjukkan bahwa sistem pengendalian pintu penjara saat ini sudah usang dan memiliki banyak keterbatasan. Dengan mengembangkan solusi ini, kami bertujuan untuk meningkatkan produktivitas, keamanan, dan efisiensi operasional di penjara.

1.2 SOLUSI

Sebagai solusi dari masalah yang sudah disebutkan di atas, kami mengimplementasikan Arduino untuk mengotomasi fungsionalitas dan mekanisme-mekanisme pada suatu penjara. Kami mengintegrasikan sebuah *button* atau tombol yang terhubung dengan satu sel penjara spesifik untuk mengatur status buka atau tutupnya. Terdapat juga sebuah *button* yang terhubung kepada semua sel sekaligus untuk melakukan aksi kolektif terhadap seluruh tahanan. Hal tersebut diharapkan untuk memberi

kemudahan kepada operator untuk mengatur pintu sel, dan dapat diperbesar skalanya untuk perkembangan lebih lanjut seperti pendekripsi jika terjadi abnormalitas.

Untuk menampilkan *real-time door status update*, kami menggunakan LED (*Light Emitting Diode*) sebagai indikator visual, dimana masing-masing LED akan merepresentasikan sebuah pintu sel.. Jika pintu sel terbuka maka LED akan menyala dan berlaku sebaliknya. Dengan itu diharapkan dapat meningkatkan monitoring dan kesadaran bagi para staf dan petugas keamanan.

Kemudian, kami juga melibatkan penggunaan motor untuk mensimulasikan pengaturan buka-tutup pintu otomatis pada masing-masing sel, sehingga tidak diperlukan adanya pengaturan manual lagi. Pintu-pintu tersebut pun terhubung kepada sensor-sensor LDR (*Light Dependant Resistor*) dan DHT11 yang berfungsi untuk memindai suhu dan kelembaban udara. Kami mendefinisikan siang dan malam berdasarkan cahaya yang terdeteksi LDR, sehingga semua pintu akan otomatis terbuka ketika siang dan tertutup ketika malam. Sedangkan, untuk mengantisipasi adanya bencana kebakaran, ketika sensor DHT11 mendeteksi suhu yang tinggi, semua pintu akan otomatis terbuka untuk menyelamatkan para tahanan. Implementasi semua fungsionalitas pada *Vault-Tec Jail Security Door System* diharapkan dapat meningkatkan protokol keamanan, alur kerja, efisiensi, serta keamanan baik bagi para petugas keamanan maupun tahanan.

1.3 KRITERIA KEBERHASILAN

Kriteria keberhasilan dari proyek ini ditentukan oleh faktor berikut:

1. Button terhubung dan dapat membuka dan menutup pintu yang sesuai
2. LED menyala sesuai dengan pintu yang sedang terbuka, sedangkan mati untuk pintu yang tertutup
3. Emergency button terhubung ke semua pintu dan LED sekaligus
4. LCD menampilkan suhu saat ini berdasarkan bacaan dari sensor DHT11 menggunakan I2C
5. LDR mengatur buka-tutupnya semua pintu berdasarkan input cahaya
6. DHT11 dapat membuka semua pintu ketika terdeteksi suhu yang tinggi

1.4 PERAN DAN TANGGUNG JAWAB

Peran dan tanggung jawab masing-masing anggota kelompok terbagi menjadi berikut:

Roles	Responsibilities	Person
Brainstorming and Design	Memikirkan ide desain, membuat prototype, dan meng-spesifikasi komponen yang dibutuhkan pada proyek	Jordhie, Syauqi, Rizki, Farras
Kode and Software	Menulis kode	Jordhie, Farras, Rizki
Proteus	Mengimplementasikan desain pada software Proteus	Jordhie, Farras
Laporan dan Dokumentasi	Membuat laporan yang komprehensif dan mengumpulkan dokumentasi	Syauqi
Testing	Memastikan bahwa semua fungsionalitas bekerja semestinya	Jordhie, Farras
Final Assembly	Merakit rangkaian fisik akhir	Jordhie, Farras, Rizki, Syauqi

Tabel 1. Peran dan Tanggung Jawab

1.5 LINIMASA DAN TARGET

TASK	Mei															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Hardware Design Brainstorm	■	■	■	■	■	■	■	■	■	■	■	■	■			
Software Development										■	■	■	■	■		
Integration and Testing												■	■	■	■	■
Final Product Assembly													■	■	■	■

Tabel 2. Linimasa dan Target

BAB 2

IMPLEMENTASI

2.1 DESAIN HARDWARE

Dalam membuat sistem otomatisasi pintu penjara, dibutuhkan banyak komponen hardware yang perlu diselaraskan. Pada Arduino, terhubung sebuah LDR (*Light Dependant Resistor*) yang berfungsi untuk mengukur kadar cahaya yang ada, dan sebuah sensor DHT11 yang mengukur suhu dan kelembaban udara sekitar. Terdapat juga sebuah LCD yang akan menampilkan hasil bacaan dari pengukuran LDR dan DHT11 dalam format “TEMP: X/nLIGHT: Y” menggunakan protokol komunikasi serial I2C. Terakhir, untuk mengatur buka-tutupnya pintu penjara, digunakan tombol untuk masing-masing sel, dimana terdapat *emergency button* yang terhubung kepada semua sel sekaligus.

Selanjutnya, data yang terbaca kemudian akan diproses, kemudian terhubung ke beberapa motor yang akan merepresentasikan engsel pada pintu masing-masing sel, dan juga terhubung beberapa LED yang masing-masing merepresentasikan status buka-tutupnya setiap sel. Ketika pintu terbuka, maka LED yang bersangkutan akan menyala, dan berlaku sebaliknya.

2.2 SOFTWARE

Untuk menjalankan proyek ini, kami menggunakan bahasa assembly yang terbagi menjadi beberapa bagian utama:

a. Master

Dalam bagian ini, didefinisikan subroutine *main* yang berisikan RCALL initialization untuk semua subroutine penting lainnya yang akan digunakan dalam keseluruhan proyek, seperti untuk serial, I2C, LCD, delay, dan lain-lain. Semua komponen akan diinisialisasi terlebih dahulu, kemudian mengeset LCD untuk menampilkan bacaan suhu, memindahkan kursor pada LCD, kemudian menampilkan bacaan cahaya dari *photoresistor*. Terdapat juga subroutine *again* yang akan loop terus menerus dan mengupdate pembacaan sensor DHT11 dan ADC (*Photoresistor*) untuk setiap iterasinya.

b. Read Photoresistor Data

Menginisialisasi ADC dengan meng-set DDR port yang sesuai, yaitu dengan menerima *raw data* dari photoresistor pada Pin 0. Data akan disimpan dalam register. Terakhir, akan memanggil subroutine LCD_disp_light untuk kemudian mendisplay hasil bacaan ADC pada LCD.

c. Determining Photoresistor Intensity

Data yang didapatkan dari *photoresistor* kemudian diproses dalam beberapa subroutine. Pada determine_if_light, nilai pada register yang mengandung bacaan LDR akan dibandingkan dengan 0x10 yang dijadikan standar terang. Jika lebih besar atau sama dengan nilai tersebut, maka akan dikategorikan sebagai terang dan lompat ke subroutine open_door_unlocked, dan kembali lagi ke master routine.

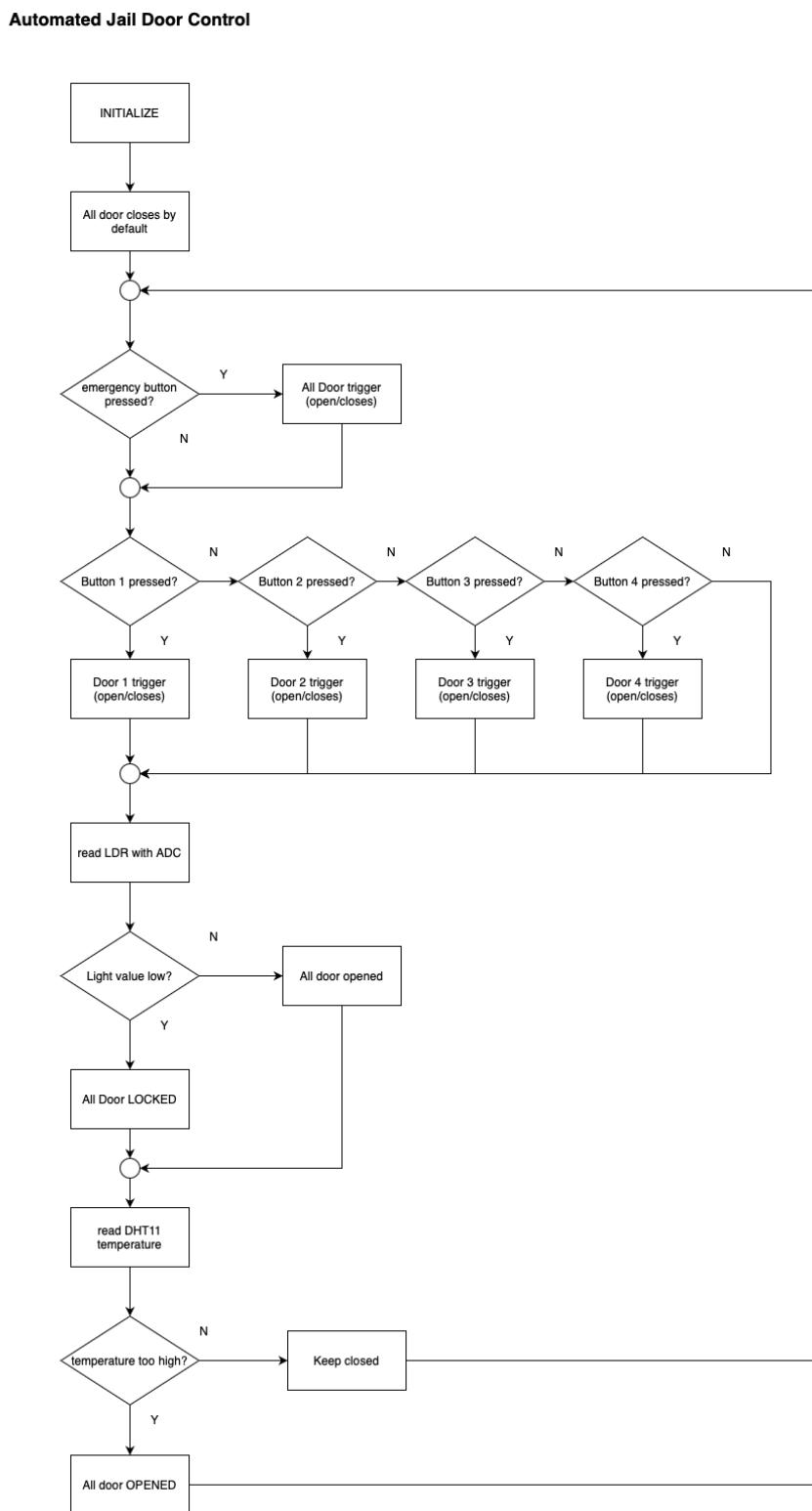
d. Read DHT11 Data

Membaca nilai pada sensor DHT11 melalui input port PB1, dimana pertama akan mengirimkan *start signal* untuk memulai pembacaan, kemudian menunggu respon dari DHT11 dan memulai pembacaan dengan membuat recall function DHT11_reading. Data kemudian akan dibandingkan dalam subroutine fire_alert untuk menentukan apakah suhu yang terbaca sudah terlalu tinggi. Jika iya, maka akan lompat ke subroutine door_open dan kembali ke main subroutine.

e. LCD

Menampilkan hasil bacaan *photoresistor* dan DHT11 ditampilkan pada resistor, dimulai dengan menampilkan starting message berupa “START”, dilanjutkan dengan menampilkan message melalui subroutine LCD_temp_message berupa “TEMP:” dan subroutine LCD_light_message berupa “LIGHT:”.

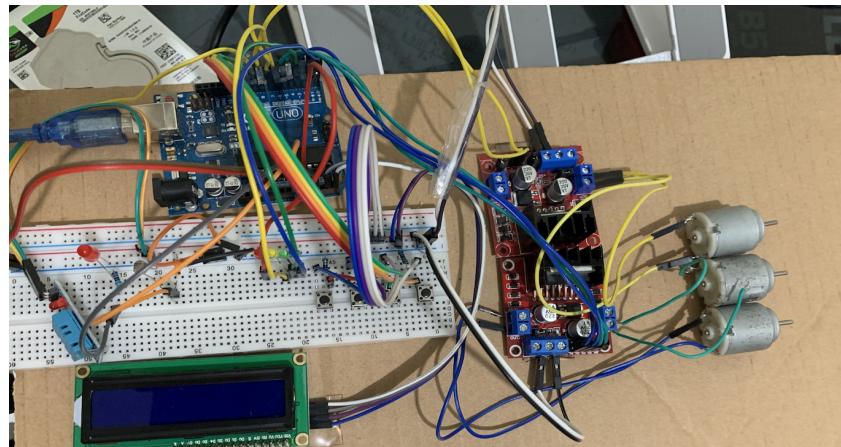
Berikut merupakan flowchart utama dari proyek Arduino kami:



Gbr 1. Flowchart

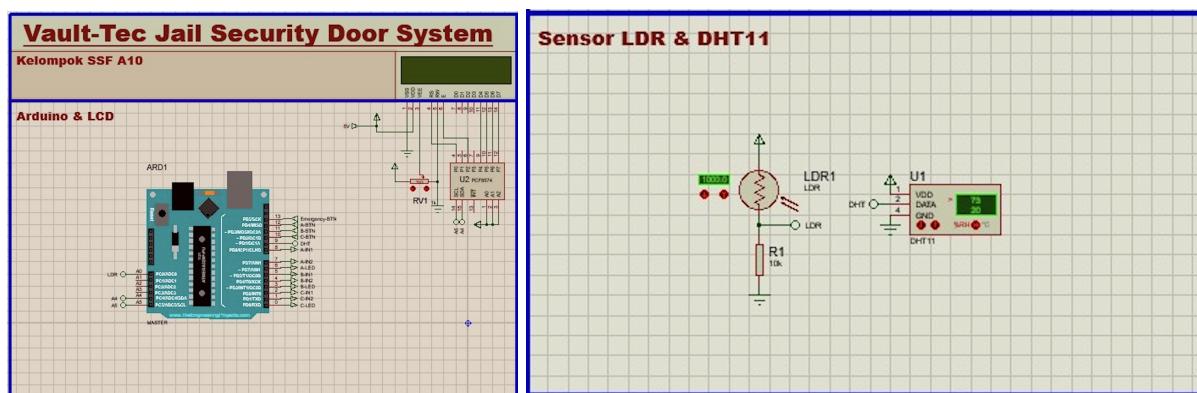
2.3 INTEGRASI HARDWARE DAN SOFTWARE

Spesifikasi hardware dan kode yang sudah dibuat kemudian digabung dalam satu proyek Arduino menggunakan Arduino IDE untuk mengupload programnya. Rangkaian final terlihat seperti pada gambar berikut:



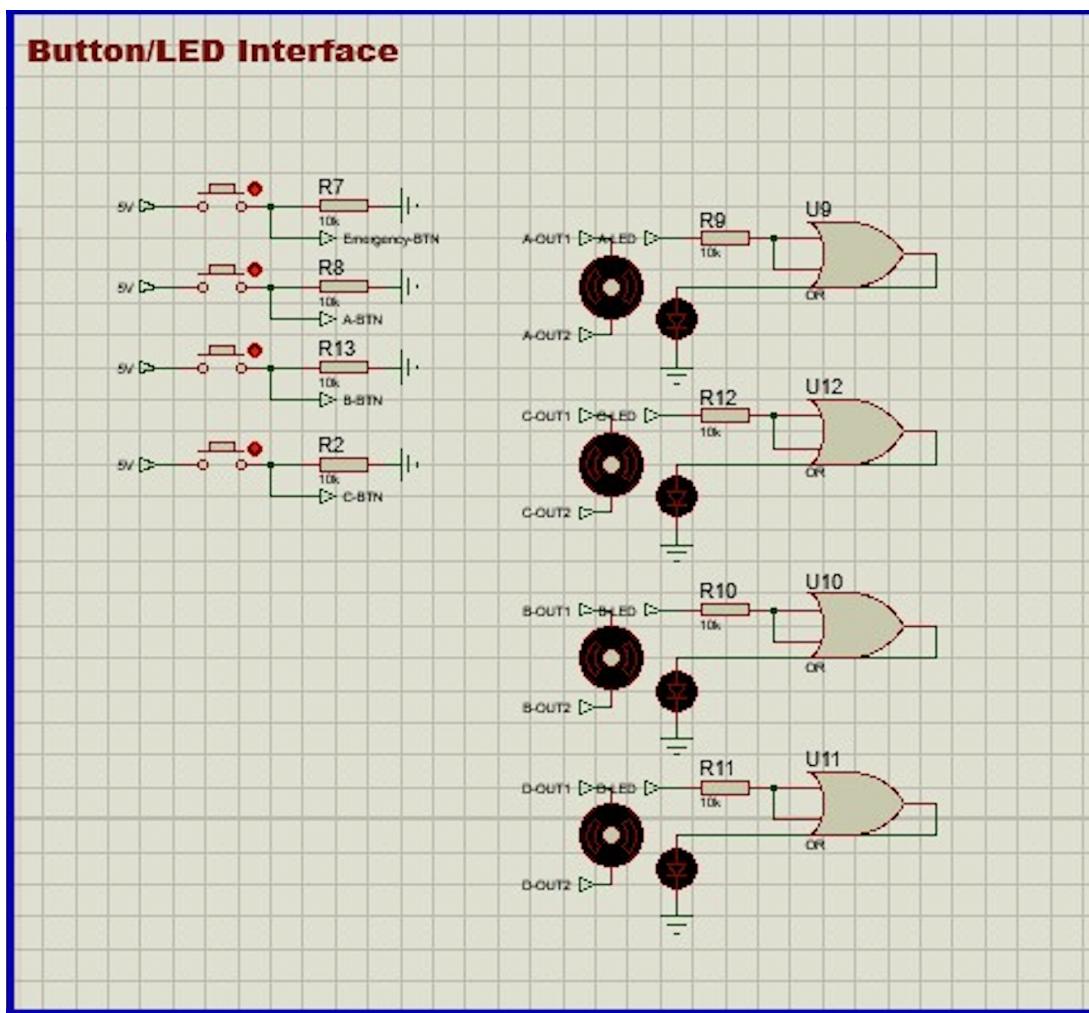
Gbr 2. Rangkaian Final

Dapat terlihat pada breadboard, terdapat komponen-komponen listrik berupa button, LED yang merepresentasikan status buka-tutupnya pintu sel. Terhubung juga *photoresistor* dan juga DHT11. LCD terhubung langsung kepada Arduino yang akan menampilkan hasil bacaan dari *photoresistor* dan DHT11. Arduino juga terhubung kepada dua buah driver yang akan mengoperasikan tiga buah motor yang masing-masing merepresentasikan engsel dari satu pintu sel. Kami juga mensimulasikan proyek pada software Proteus untuk meminimalisir terjadinya error diakibatkan oleh kecacatan pada hardware yang digunakan.



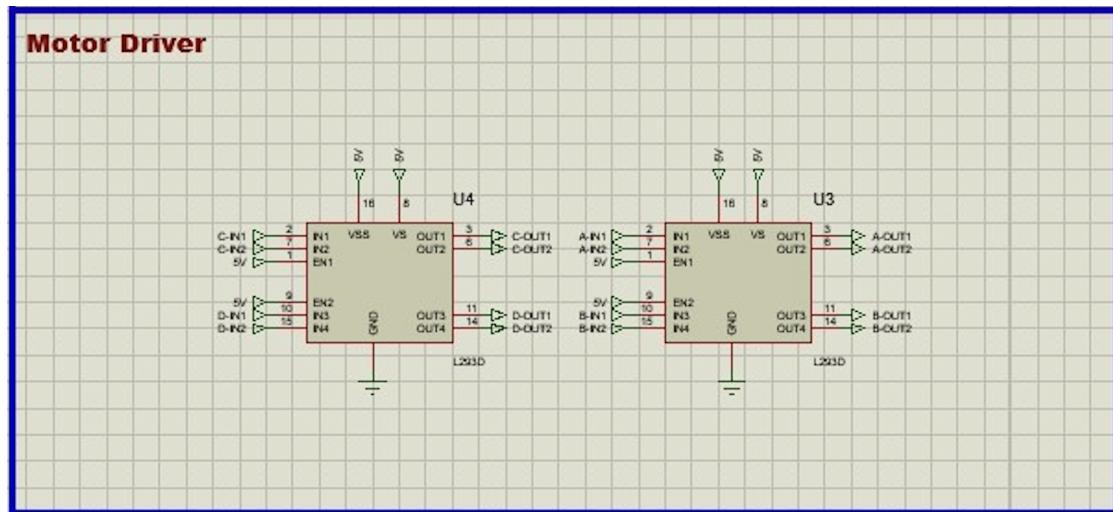
Gbr 3. Master Arduino dan Sensor pada Proteus

Button/LED Interface



Gbr 4. Interface Button, LED, dan Motor pada Proteus

Motor Driver



Gbr 5. Driver Motor pada Proteus

Kode Assembly yang diupload pada Arduino adalah sebagai berikut:

```
;-----
; Automated Jail Door Control (Master)
;-----

#define __SFR_OFFSET 0x00
#include "avr/io.h"

.global main

main:
    RCALL init_IO
    RCALL init_ADC
    RCALL I2C_init
    RCALL LCD_init
    RCALL delay_sec
    RCALL LCD_start_message
    RCALL delay_sec

    LDI R28, 0x00 ; mode = 0 for command
    LDI R27, 0x02 ; return home
    RCALL LCD_sendChar
    RCALL delay_ms
    RCALL LCD_temp_message

    LDI R28, 0x00 ; mode = 0 for command
    LDI R27, 0xC0 ; move cursor to line 2
    RCALL LCD_sendChar
    RCALL delay_ms
    RCALL LCD_light_message

again:
    LDI R30, 0x00 ; clear fire_alarm flag
    LDI R31, 0x00 ; clear night flag
    RCALL delay_sec
    RCALL Read_DHT11 ; subroutines for read DHT11
    RCALL read_ADC ; subroutines for reading photoresistor
    SBRC R31, 7
    RJMP again
    RCALL read_button ; subroutines for button input & it's output
    RJMP again ; loop back

;=====
```

```
;           I/O Subroutines
;=====

init_IO:
LDI R21, 0xFF    ; output for door LED & motor pin D
OUT DDRD, R21

LDI R21, 0x01    ; input for button & output for LED pin B
OUT DDRB, R21

LDI R21, 0x00
OUT PORTD, R21  ; default condition (close all door)
RET

read_button:
SBIC PINB, 5
RCALL emergency_trigger_close
SBIC PINB, 4
RCALL door_A_trigger
SBIC PINB, 3
RCALL door_B_trigger
SBIC PINB, 2
RCALL door_C_trigger
RET

emergency_trigger_open:
IN R12, PORTD
SBRS R12, 6
SBI PORTD, 7
SBRS R12, 3
SBI PORTD, 4
SBRS R12, 0
SBI PORTD, 1
RCALL delay_sec
LDI R31, 0xFF
OUT PORTD, R31
RCALL motor_stop
RET

emergency_trigger_close:
IN R12, PORTD
SBRC R12, 6
SBI PORTB, 0
SBRC R12, 3
SBI PORTD, 5
SBRC R12, 0
```

```
SBI PORTD, 2
RCALL delay_sec
CBI PORTB, 0
LDI R30, 0x00
OUT PORTD, R30
RET

door_A_trigger:
IN R12, PORTD
RCALL motor_A_trigger
RCALL delay_sec
RCALL motor_stop
LDI R31, 0b01000000
EOR R12, R31
OUT PORTD, R12
RET

door_B_trigger:
IN R12, PORTD
RCALL motor_B_trigger
LDI R31, 0b00001000
EOR R12, R31
OUT PORTD, R12
RET

door_C_trigger:
IN R12, PORTD
RCALL motor_C_trigger
LDI R31, 0b00000001
EOR R12, R31
OUT PORTD, R12
RET

motor_A_trigger:
SBRS R12, 6
RCALL motor_A_open
SBRC R12, 6
RCALL motor_A_close
RET

motor_A_close:
SBI PORTB, 0
RET

motor_A_open:
SBI PORTD, 7
```

```
RET

motor_B_trigger:
    SBRS R12, 3
    RCALL motor_B_open
    SBRC R12, 3
    RCALL motor_B_close
    RET

motor_B_close:
    SBI PORTD, 5
    RCALL delay_sec
    CBI PORTD, 5
    RET

motor_B_open:
    SBI PORTD, 4
    RCALL delay_sec
    CBI PORTD, 4
    RET

motor_C_trigger:
    SBRS R12, 0
    RCALL motor_C_open
    SBRC R12, 0
    RCALL motor_C_close
    RET

motor_C_close:
    SBI PORTD, 2
    RCALL delay_sec
    CBI PORTD, 2
    RET

motor_C_open:
    SBI PORTD, 1
    RCALL delay_sec
    CBI PORTD, 1
    RET

motor_stop:
    CBI PORTB, 0
    CBI PORTD, 1
    CBI PORTD, 2
    CBI PORTD, 4
    CBI PORTD, 5
    CBI PORTD, 7
```

```

RET

;=====
;           Read Photoresistor Data
; Photoresistor Pin -> Analog Pin 0 (PC0)
;=====

init_ADC:
    SBI DDRC, 0 ;set pin PC0 as i/p for ADC0
    LDI R20, 0xC0 ;internal 2.56V, right-justified data, ADC0
    STS ADMUX, R20
    LDI R20, 0x87 ;enable ADC, ADC prescaler CLK/128
    STS ADCSRA, R20
    RET

read_ADC:
    LDI    R20, 0xC7      ;set ADSC in ADCSRA to start conversion
    STS    ADCSRA, R20
    ;-----

wait_ADC:
    LDS    R21, ADCSRA   ;check ADIF flag in ADCSRA
    SBRS  R21, 4         ;skip jump when conversion is done (flag set)
    RJMP  wait_ADC      ;loop until ADIF flag is set
    ;-----
    LDI    R17, 0xD7      ;set ADIF flag again
    STS    ADCSRA, R17    ;so that controller clears ADIF
    ;-----
    LDS    R15, ADCL      ;get low-byte result from ADCL
    LDS    R13, ADCH      ;get high-byte result from ADCH

    MOV    R12, R15        ; move ADCL result to unused register
    RCALL LCD_disp_light ; display to LCD

    SBRC R31, 1
    RJMP again
    RCALL determine_if_light
    ;-----
    RET

;=====
;   For Determining the Photoresistor Intensity
;   Uses REGISTER to store result

```

```

;=====

determine_if_light:
    MOV R31, R15
    CPI R31, 0x10
    BRSH open_door_unlocked

open_door_locked:
    RCALL emergency_trigger_close
    LDI R31, 0xFF
    RET
open_door_unlocked:
    RET

;=====
;           Read DHT11 Data
; DHT11 Pin -> Digital Pin 9 (PB1)
;=====

Read_DHT11:
;-----Send Signal Start-----
    SBI DDRB, 1 ;pin PB1 as o/p
    CBI PORTB, 1 ;first, send low pulse
    RCALL delay_20ms ;for 20ms
    SBI PORTB, 1 ;then send high pulse
    CBI DDRB, 1 ;pin PB1 as i/p

;-----Wait for respond-----
w1: SBIC PINB, 1
    RJMP w1 ;wait for DHT11 low pulse
w2: SBIS PINB, 1
    RJMP w2 ;wait for DHT11 high pulse
w3: SBIC PINB, 1
    RJMP w3 ;wait for DHT11 low pulse
;-----Read DHT11-----
    RCALL DHT11_reading ;read integer humidity (1st byte of 40-bit data)
    RCALL DHT11_reading ;read decimal humidity (2nd byte of 40-bit data)
    RCALL DHT11_reading ;read integer temp (3rd byte of 40-bit data)
    MOV R29, R18      ; for LCD data register
    RCALL DHT11_reading ;read decimal temp (4th byte of 40-bit data)
    RCALL LCD_disp_temp
    RCALL fire_alert
    RET

```

```

fire_alert:
    CPI R29, 0x28
    BRSW fire_trigger_open
    RET

fire_trigger_open:
    LDI R30, 0xFF
    RCALL emergency_trigger_open
    RET

;=====
;   For Reading the DHT11
;=====

DHT11_reading:
    LDI R17, 8 ;set counter for receiving 8 bits
    CLR R18 ;clear data register
    ;-----
w4:
    SBIS PINB, 1
    RJMP w4 ;detect data bit (high pulse)
    RCALL delay_timer0 ;wait 50us & then check bit value
    ;-----
    SBIS PINB, 1 ;if received bit=1, skip next inst
    RJMP skp ;else, received bit=0, jump to skp
    SEC ;set carry flag (C=1)
    ROL R18 ;shift in 1 into LSB data register
    RJMP w5 ;jump & wait for low pulse
skp:
    LSL R18 ;shift in 0 into LSB data register
    ;-----
w5:
    SBIC PINB, 1
    RJMP w5 ;wait for DHT11 low pulse
    ;-----
    DEC R17 ;decrement counter
    BRNE w4 ;go back & detect next bit
    RET ;return to calling subroutine

;=====
;   LCD Subroutines
;=====

LCD_start_message:
    LDI R28, 1 ; mode = 1 for data

```

```

LDI R27, 'S'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'T'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'A'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'R'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'T'
RCALL LCD_sendChar
RCALL delay_sec
LDI R28, 0x00 ; mode = 0 for command
LDI R27, 0x01 ; clear LCD
RCALL LCD_sendChar
RET
;=====
LCD_temp_message:
LDI R28, 1 ; mode = 1 for data
LDI R27, 'T'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'E'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'M'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, 'P'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, ':'
RCALL LCD_sendChar
LDI R28, 1 ; mode = 1 for data
LDI R27, ' '
RCALL LCD_sendChar
RET

LCD_light_message:
LDI R28, 1 ; mode = 1 for data
LDI R27, 'L'

```

```

RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, 'I'
RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, 'G'
RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, 'H'
RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, 'T'
RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, ':'
RCALL LCD_sendChar
LDI R28, 1    ; mode = 1 for data
LDI R27, ' '
RCALL LCD_sendChar
RET

;=====
LCD_init:
;initialize LCD for 4-bit data
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x33
RCALL LCD_sendChar
RCALL delay_ms
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x32
RCALL LCD_sendChar
RCALL delay_ms
;-----
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x28    ; LCD 2 lines, 5x7 matrix
RCALL LCD_sendChar
RCALL delay_ms
;-----
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x0C    ; display ON, cursor OFF
RCALL LCD_sendChar
;-----
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x01    ; clear LCD

```

```
RCALL LCD_sendChar
RCALL delay_ms
;-----
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x06    ; shift cursor right
RCALL LCD_sendChar
;-----
RET

;=====
LCD_disp_temp:
LDI R28, 0x00    ; mode = 0 for command
LDI R27, 0x87    ; move cursor to line 1 pos
```

BAB 3

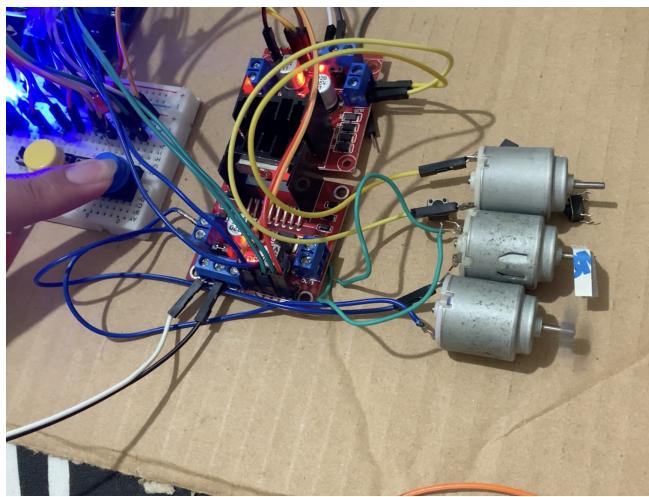
EVALUASI DAN TESTING

3.1 TESTING

Kami membagi fase testing menjadi beberapa kasus yang didasari oleh kriteria keberhasilan yang sudah ditentukan pada bab pertama. Pertama kami akan memastikan bahwa semua button terhubung dengan benar dan dapat mengatur motor yang sesuai. Kedua kami memastikan bahwa LED menyala sesuai dengan pintu yang terbuka, dan mati untuk yang tertutup. Ketiga kami memastikan bahwa emergency button terhubung dengan benar kepada semua pintu sel. Keempat kami memastikan bahwa LCD menampilkan suhu dan intensitas cahaya sesuai dengan yang terbaca dari *photoresistor* dan DHT11. Kelima kami memastikan bahwa buka-tutup pintu mengikuti pergantian siang dan malam sesuai dengan bacaan dari *photoresistor*. Terakhir, kami pun memastikan bahwa semua pintu akan terbuka ketika terdeteksi suhu yang tinggi dari DHT11 sebagai aksi preventif bencana kebakaran.

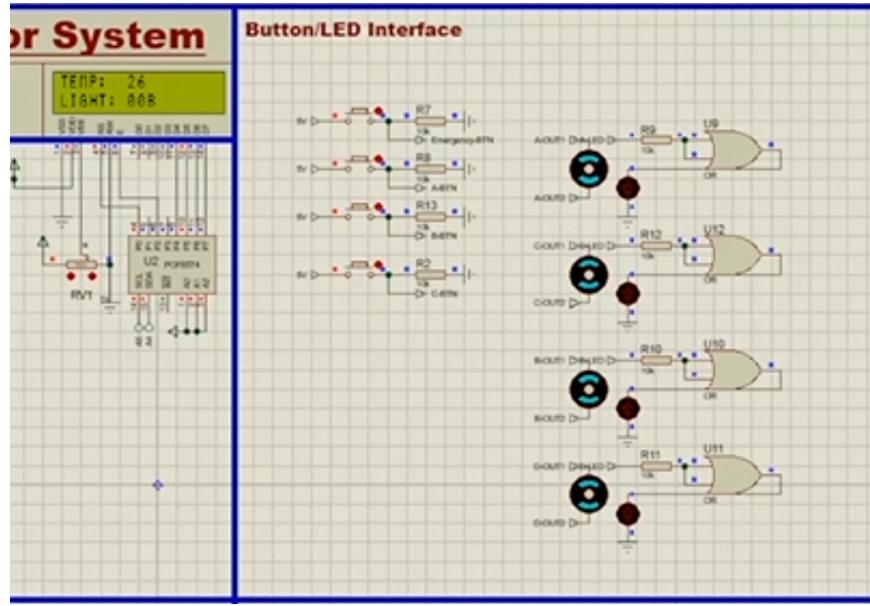
3.2 HASIL

Berdasarkan testing yang sudah ditentukan, kami mendapatkan hasil sebagai berikut sesuai dengan masing-masing kriteria keberhasilan:



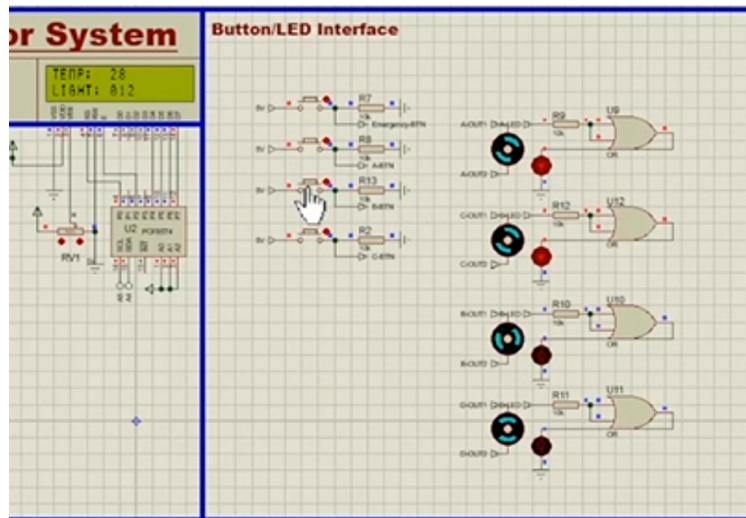
Gbr 6. Testing Kasus Pertama

Terlihat bahwa tombol sedang ditekan dan terhubung langsung kepada pintu sel pertama. Pergerakan pintu sel tersebut direpresentasikan oleh motor. Terlihat pergerakan tersebut sebagai blur berwarna abu-abu.



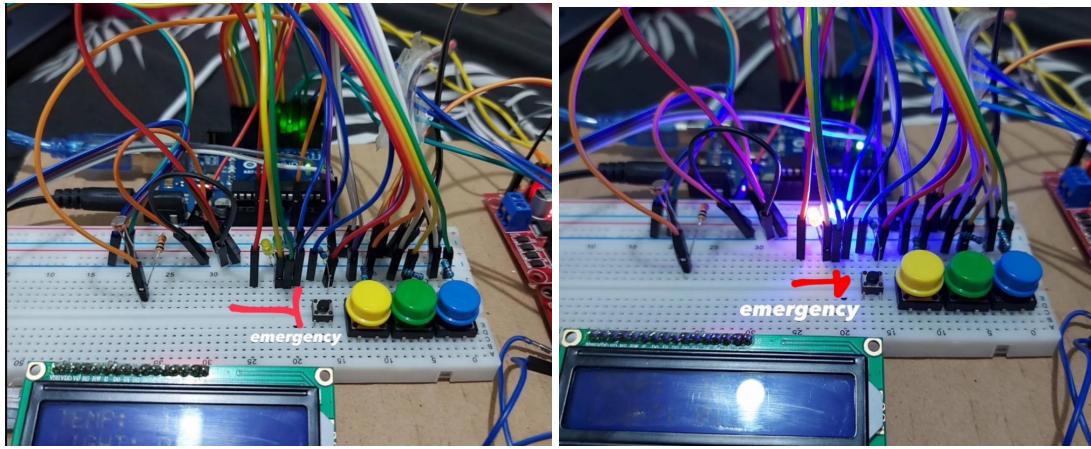
Gbr 7. Testing Kasus Kedua

Terlihat bahwa pada intensitas cahaya yang sangat rendah mengakibatkan semua pintu untuk tertutup. LED yang bersangkutan pun akan mati dan tidak akan terbuka walaupun tombol ditekan.



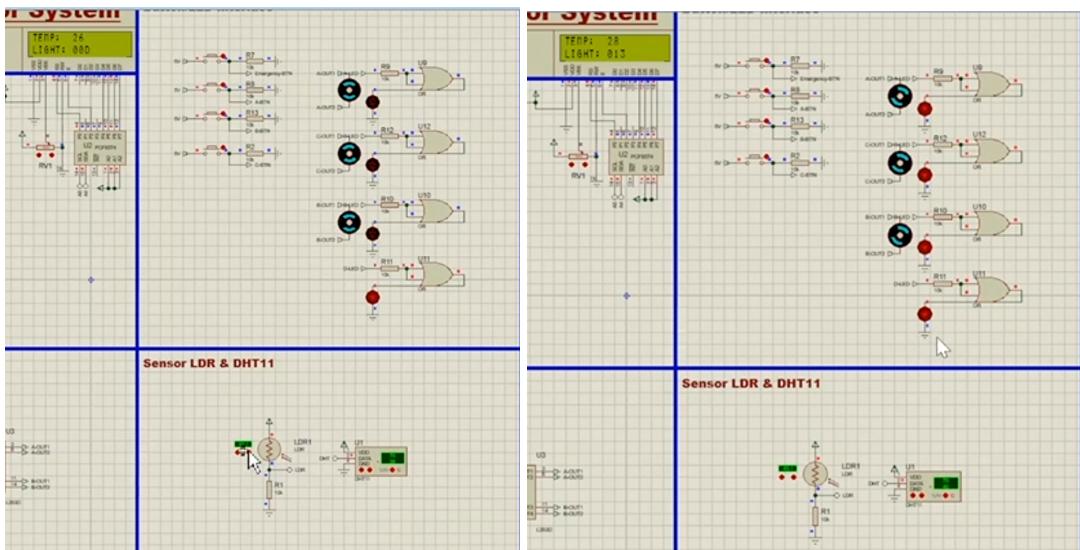
Gbr 8. Testing Kasus Kedua

Selanjutnya, terlihat bahwa pada intensitas cahaya yang cukup tinggi, pintu akan terbuka dan tertutup sesuai dengan tombol yang ditekan. Terlihat bahwa tombol ketiga menggerakkan motor pada pintu ketiga dari atas yang juga mematikan LED yang bersangkutan.



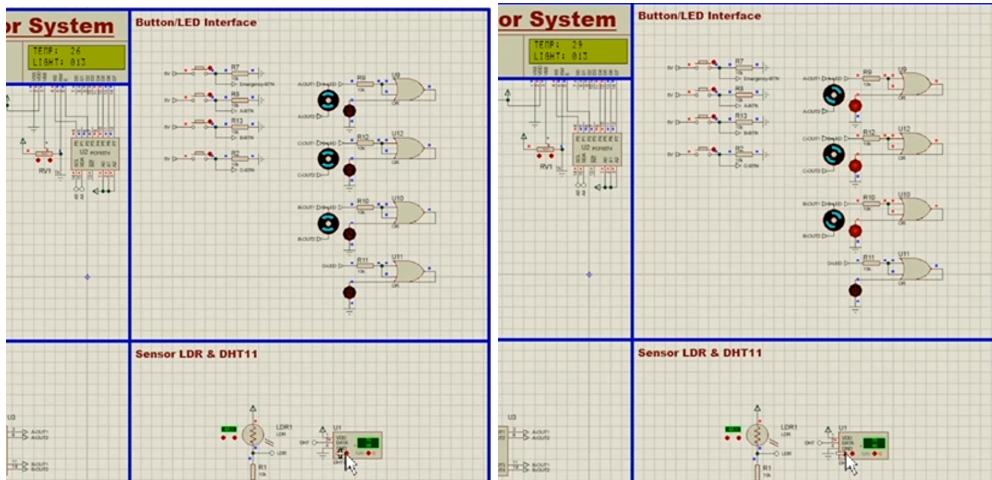
Gbr 9. Testing Kasus Ketiga

Terlihat bahwa *emergency button* berwarna hitam akan terhubung langsung kepada semua pintu sel yang direpresentasikan oleh LED. Ketika tombol ditekan, maka semua pintu akan terbuka atau tertutup bersamaan.



Gbr 10. Testing Kasus Kelima

Terlihat bahwa pada saat semua pintu tertutup, *photoresistor* memberikan bacaan 00D yang masih tergolong malam. Ketika cahaya yang terdeteksi dinaikkan menjadi 013, program membacanya sebagai siang dan semua pintu pun terbuka.



Gbr 11. Testing Kasus Keenam

Terlihat bahwa pada saat semua pintu tertutup, DHT11 memberikan bacaan 26 dalam hexadecimal yang masih tergolong normal. Ketika suhu yang terdeteksi dinaikkan menjadi 29, program membacanya sebagai terjadi kebakaran dan semua pintu pun terbuka.

3.3 EVALUASI

Dapat dilihat bahwa proyek berhasil untuk memberikan *output* sesuai dengan kriteria keberhasilan. Pengecualian untuk poin keempat karena sudah terwakilkan oleh testing kasus-kasus lainnya. Terdapat beberapa kendala pada rangkaian fisik seperti cacat pada hardware yang menyebabkan mayoritas testing dilakukan pada rangkaian Proteus.

BAB 4

KESIMPULAN

Untuk menyimpulkan, proyek untuk mata kuliah Sistem Siber Fisik berjudul *Automated Jail Door Control* terlaksana dengan cukup sukses dan menghasilkan sistem yang fungsional dan efisien untuk mengotomatisasi operasional pintu penjara. Rangkaian mengimplementasi berbagai macam komponen seperti *photoresistor*, sensor DHT11, button, LCD, LED, motor, serta komunikasi I2C untuk mendisplay data pada LCD.

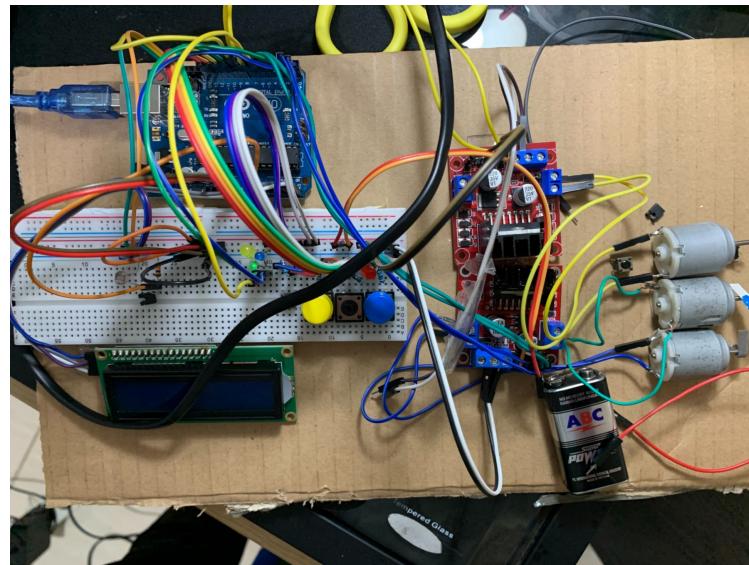
Kami juga mendefinisikan berbagai kriteria yang dijadikan patokan untuk keberhasilan proyek yang masing-masing dibuktikan saat masa testing. Dapat disimpulkan bahwa keenam kriteria keberhasilan telah tercapai, baik melalui medium rangkaian fisik dan juga rangkaian virtual menggunakan software Proteus. Kami harap dengan adanya proyek ini, operasional pada penjara dapat dibuat lebih efisien dan aman baik bagi para petugas maupun tahanan

REFERENSI

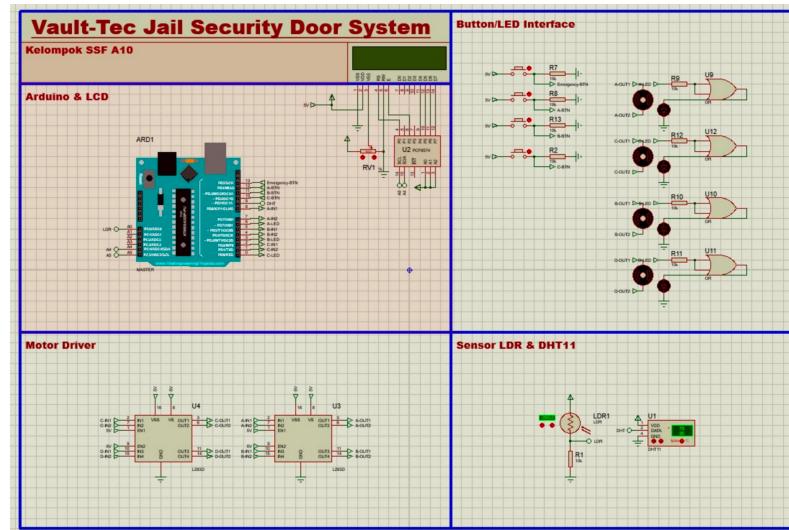
- [1] Harditya, M., & Faza, M. N. (n.d.). *Modul 2 SSF: Introduction to Assembly_I_O Programming.* [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw> [Accessed: May. 6, 2023].
- [2] Harditya, M., & Faza, M. N. (n.d.) *Modul 3 SSF: Analog to Digital Converter.* [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw/a/NTk1MTc3NTE0MzI5/details> [Accessed: May. 6, 2023].
- [3] Harditya, M., & Faza, M. N. (n.d.) *Modul 4 SSF: Serial Port.* [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw/m/NTk2NzYzMjQ2MTUx/details> [Accessed: May. 6, 2023].
- [4] Harditya, M., & Faza, M. N. (n.d.) *Modul 5 SSF: Aritmatika.* [Online] Available: https://emas2.ui.ac.id/pluginfile.php/3744299/mod_resource/content/3/Modul%205%20SSF%20Aritmatika.pdf [Accessed: May. 6, 2023].
- [5] Harditya, M., & Faza, M. N. (n.d.) *Modul 6 SSF: Timer.* [Online] Available: https://emas2.ui.ac.id/pluginfile.php/3754317/mod_resource/content/1/Modul%206%20SSF%20Timer.pdf [Accessed: May. 6, 2023]
- [6] Harditya, M., & Faza, M. N. (n.d.) *Modul 7 SSF: Interrupt.* [Online] Available: https://emas2.ui.ac.id/pluginfile.php/3767711/mod_resource/content/3/Modul%207%20SSF%20Interrupt.pdf [Accessed: May. 6, 2023]
- [7] Harditya, M., & Faza, M. N. (n.d.) *Modul 8 SSF: I2C & SPI.* [Online] Available: https://emas2.ui.ac.id/pluginfile.php/3797142/mod_resource/content/1/Modul%208%20SSF%20I2C%20%20SPI.pdf [Accessed: May. 6, 2023]
- [8] Harditya, M., & Faza, M. N. (n.d.) *Modul 9 SSF: Sensor Interfacing.* [Online] Available: https://emas2.ui.ac.id/pluginfile.php/3797143/mod_resource/content/1/Modul%209%20SSF-%20Sensor%20Interfacing.pdf [Accessed: May. 6, 2023].
- [9] K. Anas. (2023). “Creating Arduino Library for I2C LCD”. *AnasKuzechi*. <https://akuzechie.blogspot.com/2022/06/creating-arduino-library-for-i2c-lcd.html>. [Accessed May 12—16 2023]

APPENDICES

Appendix A: Project Schematic



Gbr 12. Rangkaian Fisik Akhir



Gbr 13. Rangkaian Proteus Akhir

Appendix B: Documentation



Gbr 14, 15, & 16. Dokumentasi