



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

**IC CHECKER**

**GROUP 15**

<b>EDGRANT HENDERSON SURYAJAYA</b>	<b>2206025016</b>
<b>FAIRUZ MUHAMMAD</b>	<b>2206814324</b>
<b>DIMAS DANDOSSI WICAKSONO</b>	<b>2206059780</b>
<b>FADLIHAJJAN CAREL AGFATA</b>	<b>2206826822</b>

## PREFACE

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena rahmat dan karunia-Nya, kami berhasil menyelesaikan proyek akhir untuk mata kuliah Sistem Siber Fisik. Adapun proyek kami ini yang berjudul “IC Checker”.

Pada kesempatan ini, kami ingin menyampaikan terima kasih yang sebesar-besarnya kepada dosen mata kuliah Sistem Siber Fisik, Bapak Fransiskus Astha Ekadiyanto, atas ilmu dan bimbingan yang telah beliau berikan, sehingga kami dapat menyelesaikan proyek ini dengan baik. Kami juga berterima kasih kepada seluruh asisten laboratorium Digital Laboratory, yang telah memberikan dukungan dan bantuan selama proses pengerjaan proyek ini. Secara khusus, kami ingin mengucapkan terima kasih kepada asisten pendamping Juan Jonathan atas bimbingan dan kontribusinya yang berharga.

Kami menyadari bahwa proyek ini masih jauh dari sempurna, karena kami masih dalam tahap pembelajaran. Oleh karena itu, kami sangat terbuka untuk menerima kritik dan saran yang konstruktif dari para pembaca. Kami berharap, proyek ini dapat bermanfaat bagi banyak orang dan menjadi kontribusi yang berarti bagi mata kuliah Sistem Siber Fisik.

Depok, Mei 28, 2024

Group 15

## 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**

Integrated Circuit (IC) adalah komponen penting dalam berbagai perangkat elektronik. Memeriksa kondisi dan fungsi setiap pin IC secara manual dapat menjadi tugas yang rumit dan memakan waktu, terutama bagi teknisi dan insinyur yang bekerja dengan banyak jenis IC. Tanpa alat yang tepat, proses pengecekan ini bisa rentan terhadap kesalahan dan tidak efisien. Dalam industri elektronik, kebutuhan akan alat yang dapat memeriksa IC secara otomatis sangatlah penting untuk memastikan keandalan dan kinerja perangkat yang menggunakan IC tersebut.

### **1.2 PROPOSED SOLUTION**

Untuk mengatasi masalah ini, kami merancang sebuah alat bernama “IC Checker” yang dibuat menggunakan bahasa AVR assembly. Alat ini memungkinkan pengguna untuk memilih jenis IC yang akan diperiksa melalui 2 buah tombol. Setelah jenis IC dipilih, alat ini akan secara otomatis memeriksa semua pin IC dan menampilkan hasilnya pada LCD I2C. Setiap pin IC akan diperiksa dan jika semua pin berfungsi dengan baik dan tidak ada yang rusak, alat ini akan menampilkan jenis IC tersebut pada LCD I2C. Solusi ini tidak hanya meningkatkan efisiensi dan akurasi pengecekan IC, tetapi juga memudahkan teknisi dan insinyur dalam melakukan diagnosis dan perbaikan perangkat elektronik yang menggunakan berbagai jenis IC.

### **1.3 ACCEPTANCE CRITERIA**

Berikut ini adalah parameter kesuksesan dari proyek IC Checker yang kami kerjakan:

1. Alat dapat memilih jenis IC melalui tombol
2. Alat dapat memeriksa setiap pin dari IC yang dipilih secara otomatis
3. Alat dapat menampilkan jenis IC yang diperiksa pada LCD
4. Sistem harus mudah digunakan dan memberikan hasil yang akurat

## 1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Merancang dan menyusun rangkaian, program.	Merancang <i>hardware</i> dan <i>software</i> sesuai dengan ide yang telah diajukan. Setelah itu menjelaskannya dalam laporan pada bagian <i>software</i> .	Edgrant Henderson Suryajaya Fairuz Muhammad
Menyusun rangkaian pada Proteus	Mengaplikasikan rangkaian pada Proteus.	Fairuz Muhammad
Menyusun laporan, dan Readme.md	Menyusun sebagian besar laporan selain implementation, dan menyusun Readme.md	Edgrant Henderson Suryajaya Fairuz Muhammad Dimas Dandossi Wicaksono Fadlihajjan Carel Agfata
Menyusun PPT	Menyusun PPT yang akan digunakan pada presentasi proyek.	Edgrant Henderson Suryajaya Fairuz Muhammad

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES



## **CHAPTER 2**

### **IMPLEMENTATION**

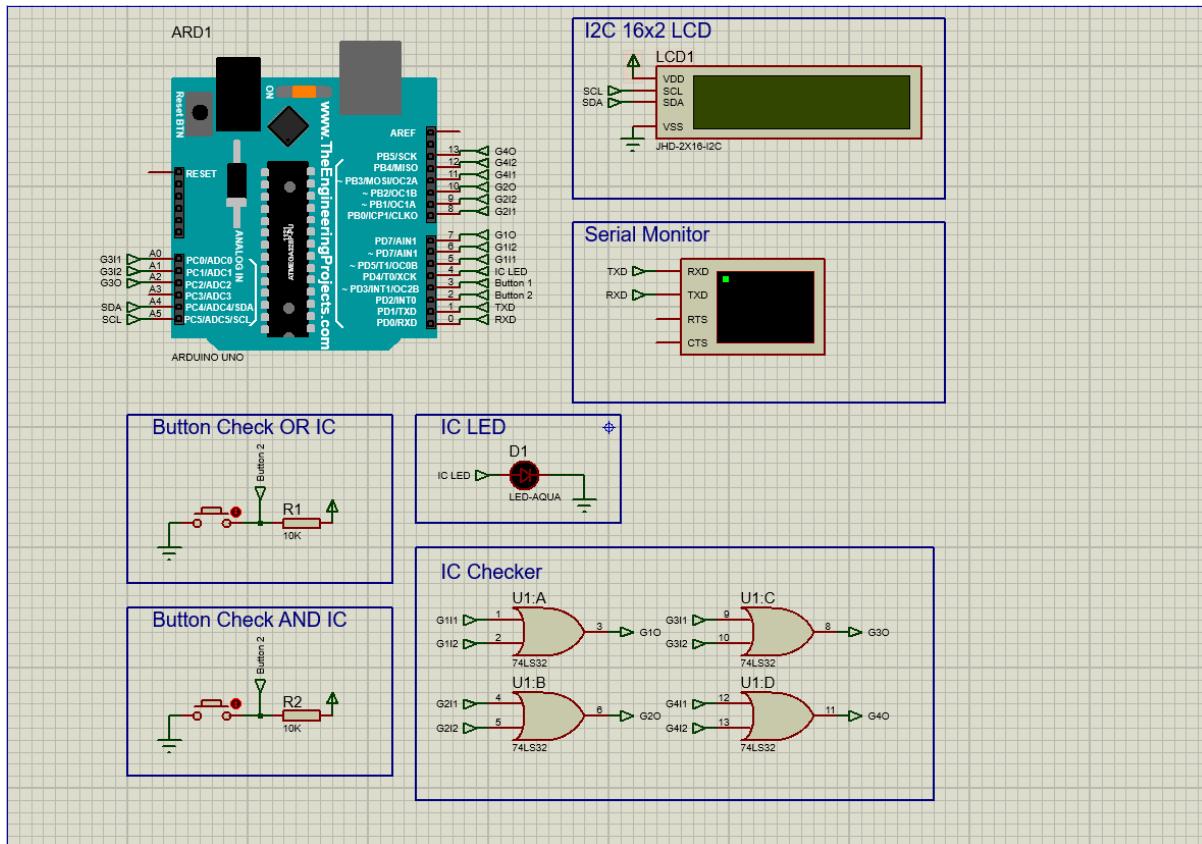
#### **2.1 HARDWARE DESIGN AND SCHEMATIC**

Komponen yang dibutuhkan:

- Arduino Uno - 1 buah
- Breadboard - 1 buah
- LED - 1 buah
- LCD 16x2 with I2C
- Kabel jumper male to male
- Kabel jumper male to female
- Baterai 9V - 1 buah
- 7805 voltage regulator

Rangkaian fisik yang kami gunakan ini secara garis besar menggunakan breadboard untuk menyambungkan para komponen dan arduino uno sebagai controller logikanya. Hal yang membedakan dari perangkat lainnya adalah kami menggunakan sebuah I2C LCD 16x2, dimana kita akan menampilkan IC AND atau OR tergantung dengan IC yang sedang dilakukan pengecekan. Untuk menghubungkan I2C LCD ke Arduino, pin (Serial Data Line) SDA dan SCL (Serial Clock Line) digunakan dikarenakan kedua pin tersebut merupakan jalur komunikasi utama untuk protokol I2C, di mana SDA digunakan untuk mengirim dan menerima data, sementara SCL mengatur timing sinyal untuk sinkronisasi komunikasi data.

Untuk schematicnya berikut adalah bentuk rangkaian dalam simulasi Proteus,



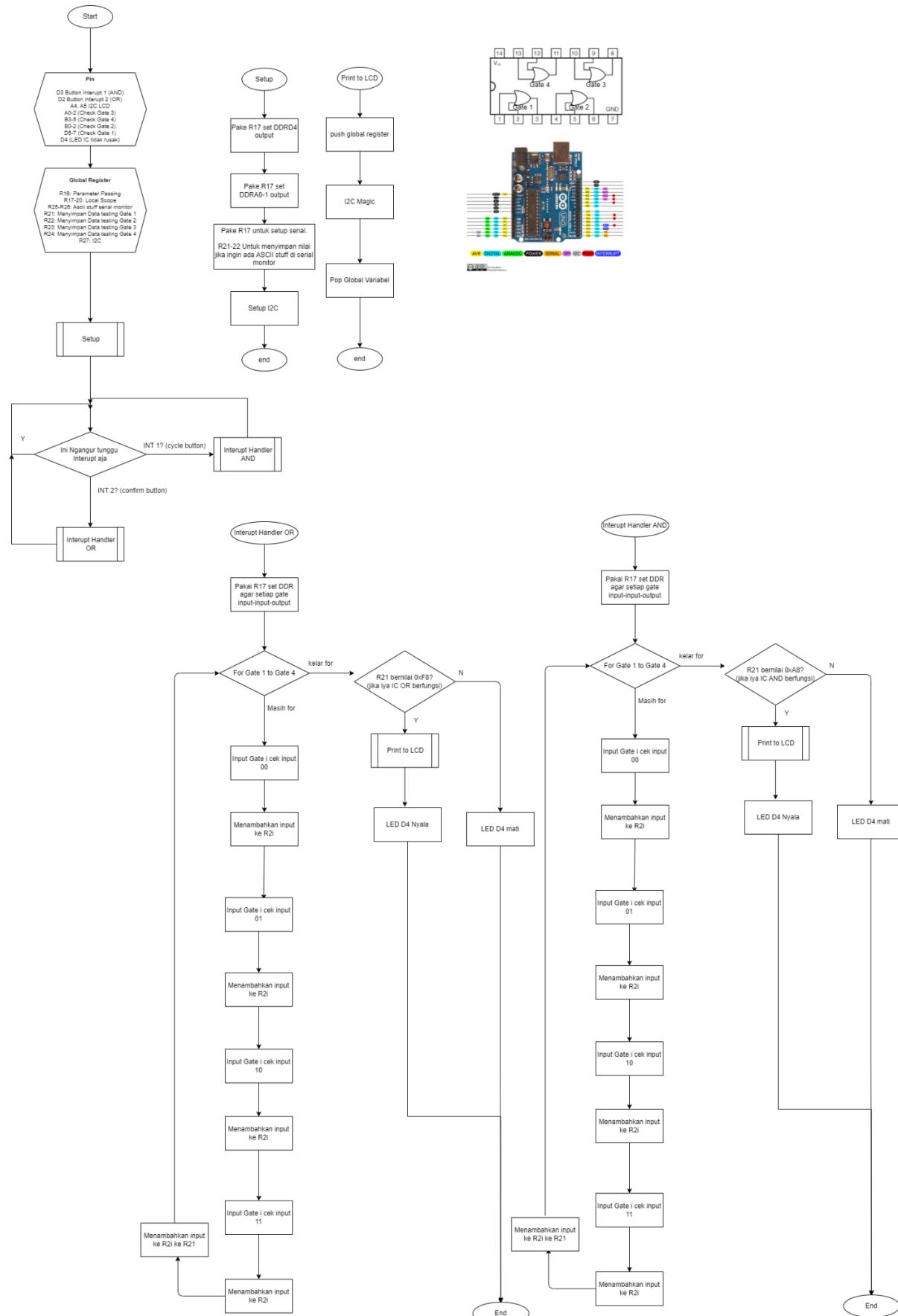
## 2.2 SOFTWARE DEVELOPMENT

Perancangan dan pengembangan software pada proyek berbasis pada dokumentasi flowchart yang dibuat pertama. Flowchart ini berfungsi sebagai panduan dalam menentukan alur program yang akan dibuat, khususnya dalam penentuan pin dan register yang akan digunakan.

Software dikembangkan secara modular dengan melakukan abstraksi sebanyak mungkin agar memudahkan dalam pengembangan dan pemeliharaan kode. Software yang dibuat terdiri dari beberapa modul, yaitu:

- Deklarasi variabel: modul ini berisi deklarasi variabel pin input dan output
- Setup: modul ini berisi inisialisasi pin input dan output, serta inisialisasi serial communication, I2C, dan Interrupt
- Delay: modul ini berisi fungsi delay yang digunakan dalam program
- Serial: modul ini berisi fungsi-fungsi yang digunakan dalam komunikasi serial
- LCD: modul ini berisi fungsi-fungsi yang digunakan menampilkan data pada LCD
- Checker: modul ini berisi fungsi-fungsi yang digunakan dalam pengecekan IC

## Flowchart:



## **2.3 HARDWARE AND SOFTWARE INTEGRATION**

Cara kerja I2C pada kode ini dengan mengatur PC4 (SDA) dan PC5 (SCL) sebagai input komunikasi. Setelah itu, kode menginisialisasi modul I2C dengan mengatur prescaler dan frekuensi SCL agar sesuai dengan kecepatan komunikasi yang diinginkan, serta mengaktifkan modul TWI (Two Wire Interface). Selanjutnya, kode akan mengirimkan karakter yang akan di print ke LCD dengan mengirimkan kondisi START, diikuti dengan alamat slave LCD dan bit penulisan (SLA+W). Alamat slave LCD yang digunakan adalah 0x27 untuk rangkaian asli, dan 0x3E untuk rangkaian pada proteus. Kemudian, kode mengirim byte kontrol untuk memilih register instruksi atau data, dan mengatur alamat DDRAM ke posisi pertama pada baris pertama LCD. Proses ini diulangi dengan mengirim kondisi START lagi, alamat slave, dan byte kontrol untuk menulis data. Setelah itu, kode mengirimkan karakter yang akan di print lagi dan mengakhiri transmisi dengan kondisi STOP. Kode ini terus mengulangi proses tersebut untuk mengirimkan karakter secara berulang-ulang ke LCD hingga selesai.

I2C LCD perlu menggunakan I2C karena protokol I2C memungkinkan komunikasi dengan hanya 2 pin, mengurangi jumlah koneksi yang diperlukan dan mempermudah pengaturan perangkat dengan banyak slave.

Koneksi pin I2C LCD ke Arduino Uno menggunakan pin SDA(A4) dan SCL(A5). Koneksi pin lain untuk proses pengecekan IC dan menyalaikan LED menggunakan pin digital. Setup pin untuk IC dan LED dilakukan dengan menentukan setiap bit pada Data Direction Register (DDR) sebagai input atau output. Pin Input yang diatur dalam keadaan floating karena IC selalu akan memberi output berupa 0 atau 1.

## CHAPTER 3

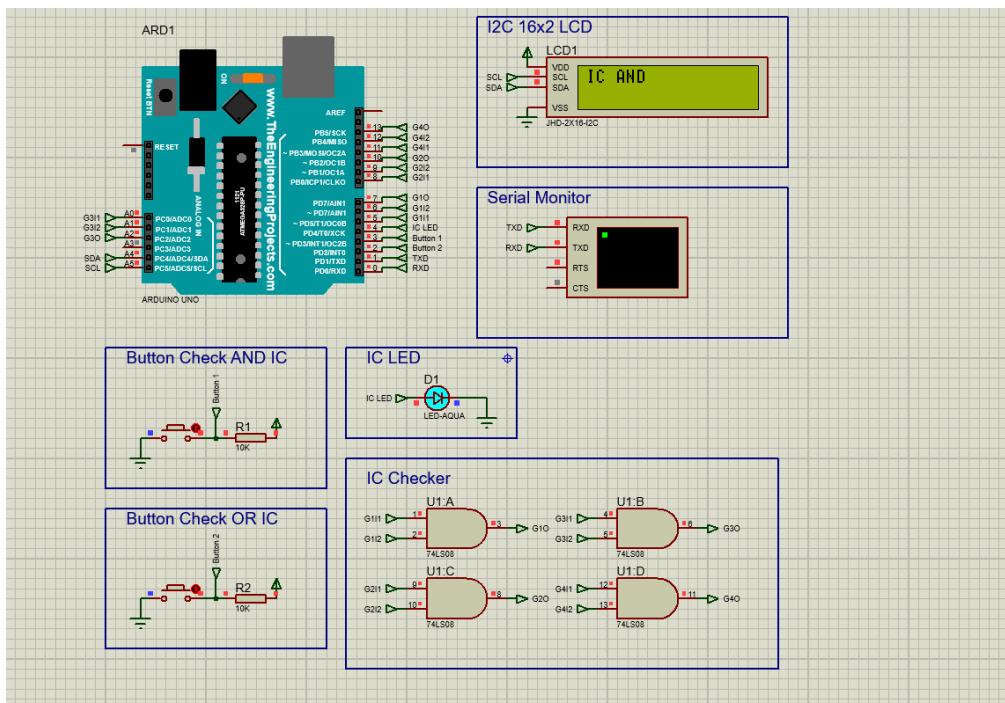
### TESTING AND EVALUATION

#### 3.1 TESTING

Proses pengujian sistem dilakukan dalam dua lingkungan yang berbeda: simulasi menggunakan Proteus dan implementasi pada rangkaian fisik. Pengujian ini bertujuan untuk memverifikasi fungsionalitas dan keandalan dari setiap komponen serta integrasi keseluruhan sistem.

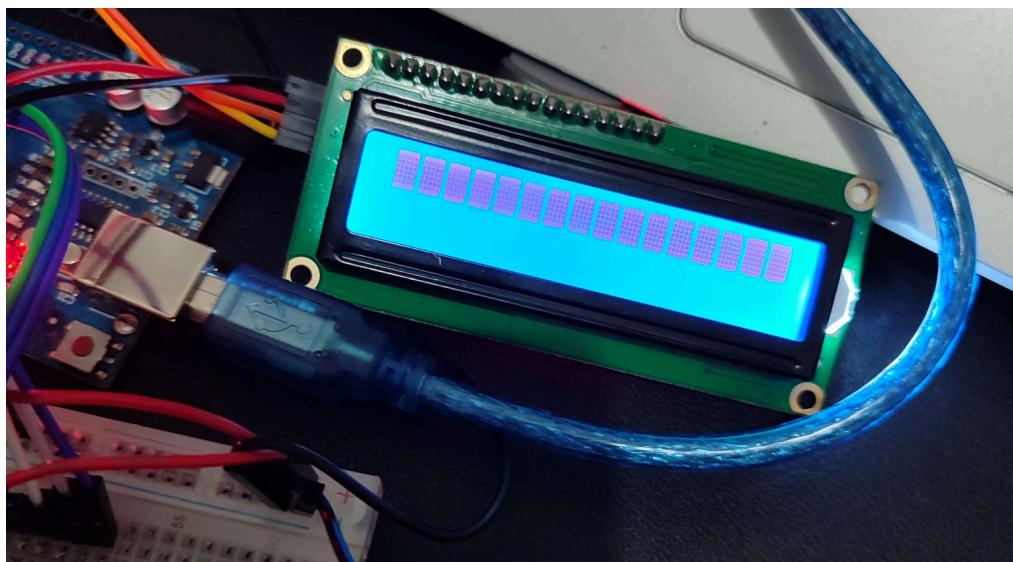
##### 3.1.1 Pengujian Menggunakan Proteus

Pengujian awal dilaksanakan melalui simulasi dengan menggunakan Proteus, dimana semua komponen sistem termasuk IC, mikrokontroler, dan LCD diuji. Dalam simulasi, semua fungsi berjalan dengan lancar tanpa masalah. Program Arduino berhasil mengintegrasikan input dari sensor dan mengirimkan output yang tepat ke LCD dan perangkat output lainnya. LCD mampu menampilkan teks dengan jelas, dan tidak ada masalah dengan penerimaan atau pengolahan data oleh mikrokontroler.



### **3.1.2 Pengujian Rangkaian Fisik**

Setelah berhasil dalam simulasi Proteus, sistem kemudian diimplementasikan pada rangkaian fisik. Walaupun komponen seperti pengecekan IC dan mikrokontroler bekerja sesuai harapan, kami menghadapi masalah dengan modul LCD. Dalam implementasi fisik, LCD hanya menampilkan kotak-kotak, dan tidak berhasil menampilkan huruf atau teks yang diharapkan. Selain itu setiap pemanggilan fungsi untuk menampilkan teks pada LCD, program akan mengalami hang dan tidak berjalan dengan baik. Kami mencoba untuk mengganti modul LCD dengan yang baru, namun masalah yang sama masih terjadi. Kami menduga bahwa masalah ini disebabkan oleh kesalahan dalam pengaturan konfigurasi LCD, namun kami belum berhasil menemukan solusi yang tepat.



## **3.2 RESULT**

Hasil pengujian mengindikasikan bahwa terdapat ketidaksesuaian antara simulasi dan implementasi nyata, khususnya dalam kinerja LCD. Kegagalan LCD untuk menampilkan teks secara tepat di rangkaian fisik menunjukkan kemungkinan masalah pada konfigurasi pin atau masalah pada kode yang tidak terdeteksi selama simulasi Proteus. Namun, secara keseluruhan, logika pengecekan IC berhasil dan dapat dilihat melalui LED yang menyala sesuai dengan kondisi IC yang diuji dan output yang ditampilkan pada serial monitor.

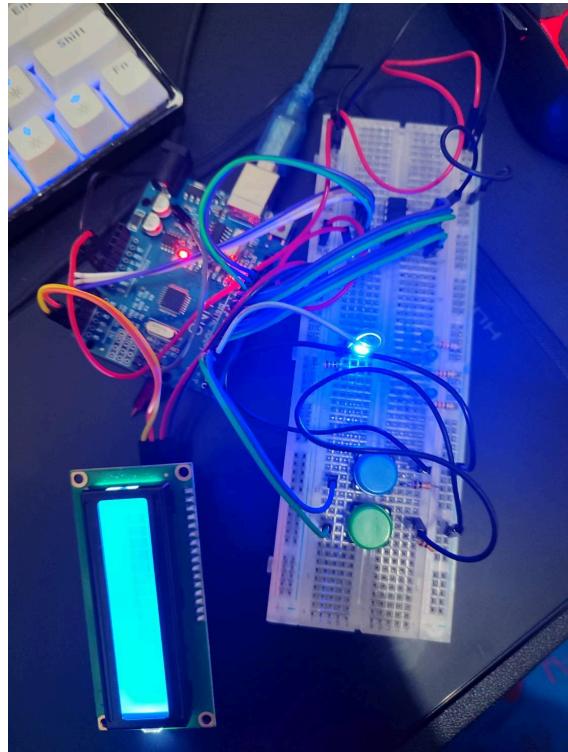


Fig 1. Kondisi Pengecekan IC Sesuai

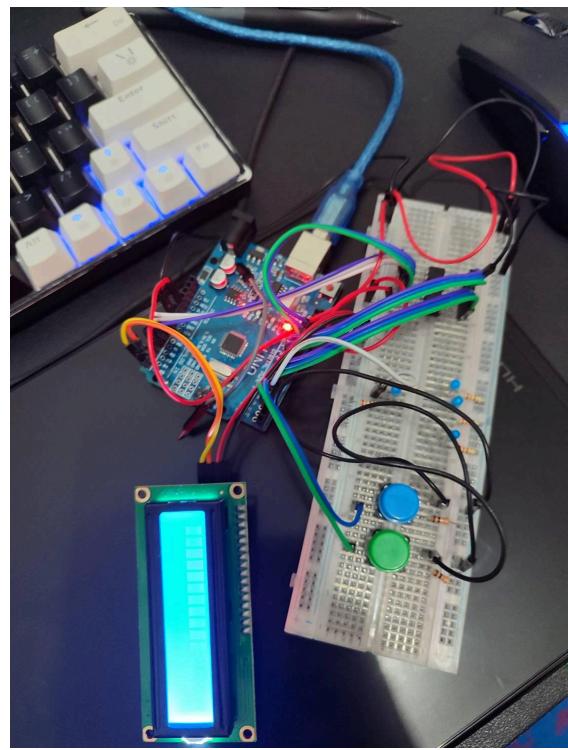


Fig 2. Kondisi Pengecekan IC Tidak Sesuai

### **3.3 EVALUATION**

Evaluasi lebih lanjut diperlukan untuk mengidentifikasi penyebab spesifik dari masalah tampilan LCD. Langkah-langkah yang akan diambil mencakup:

1. Pemeriksaan Hardware: Memeriksa koneksi fisik pada LCD dan memastikan bahwa semua pin terhubung dengan benar.
2. Revisi Kode: Memeriksa kembali kode program Arduino, khususnya bagian konfigurasi dan inisialisasi LCD.
3. Kalibrasi Ulang: Melakukan kalibrasi ulang pada LCD dengan mengacu pada spesifikasi teknis dari produsen.

Pengujian dan evaluasi lebih lanjut ini diharapkan akan mengatasi masalah yang ada dan meningkatkan keandalan sistem saat dioperasikan dalam kondisi nyata.

## **CHAPTER 4**

### **CONCLUSION**

Pada akhir proyek ini, dapat disimpulkan bahwa alat "IC Checker" berhasil dikembangkan sebagai sebuah sistem yang efektif untuk melakukan pengecekan Integrated Circuit (IC). Alat ini dirancang dengan kemampuan untuk memilih jenis IC melalui tombol (button) dan secara otomatis melakukan pengecekan pada setiap pin IC yang dipilih. Hasil pengecekan ini kemudian ditampilkan pada LCD I2C, memberikan informasi secara rinci tentang kondisi tiap pin.

1. Fungsi Utama: Alat ini berhasil menjalankan fungsi utamanya untuk memilih jenis IC dan memeriksa semua pin secara otomatis. Proses ini memastikan bahwa pengguna dapat dengan mudah dan cepat mengecek kondisi IC tanpa perlu melakukan
2. Output yang Jelas dan Informatif: LCD I2C digunakan untuk menampilkan hasil pengecekan tiap pin IC, memberikan informasi yang jelas dan mudah dibaca oleh pengguna. Hal ini penting untuk memastikan bahwa hasil diagnosa dapat dipahami dengan cepat dan akurat.
3. Otomatisasi dan Efisiensi: Dengan kemampuan otomatisasi dalam pengecekan pin IC, alat ini meningkatkan efisiensi proses pengecekan dibandingkan metode manual.

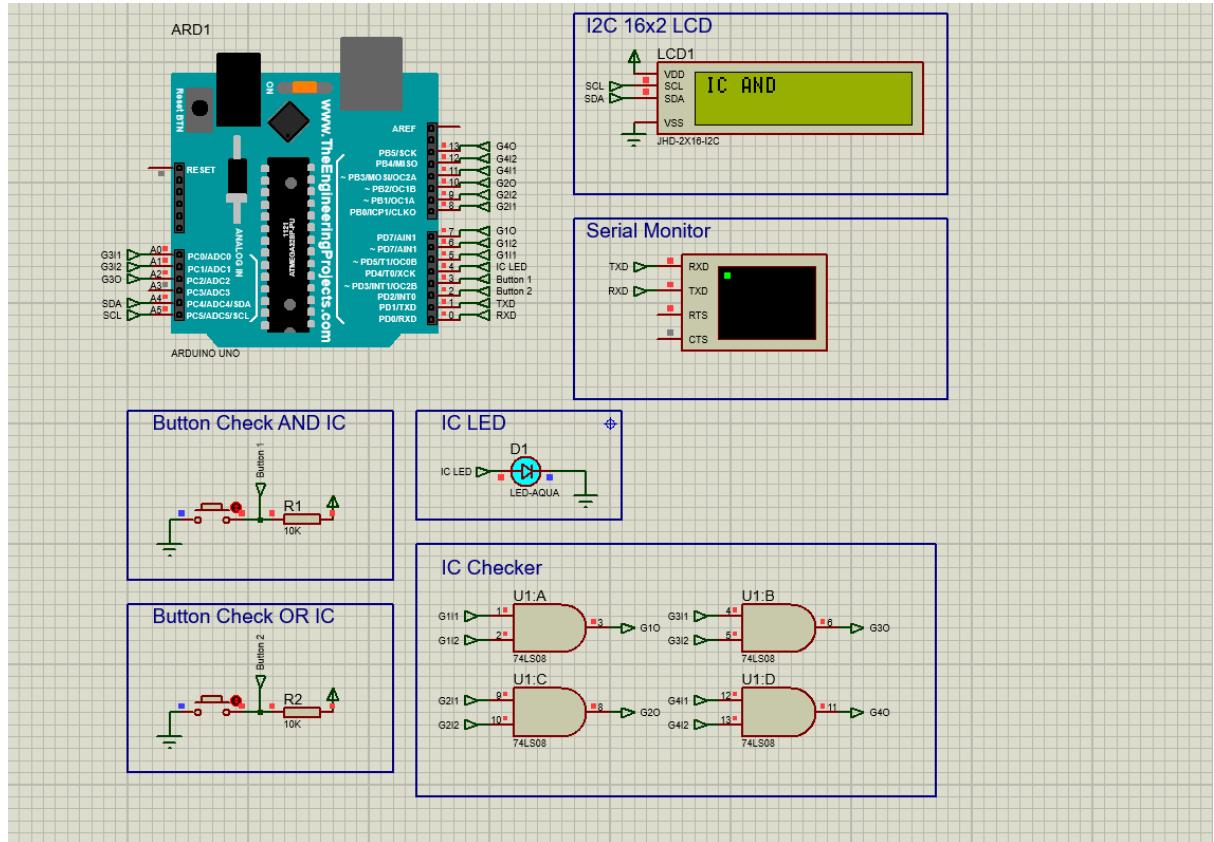
Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena berkat pertolongan-Nya serta kerjasama tim yang baik di kelompok 15, kami berhasil menyelesaikan proyek akhir Sistem Siber Fisik berjudul "IC Checker". Kami juga mengucapkan terima kasih kepada asisten pendamping Juan Jonathan atas bimbingannya selama proyek ini. Alat yang kami rancang mampu memeriksa kondisi dan fungsi setiap pin IC secara otomatis dan menampilkan jenis IC pada LCD I2C jika semua pin berfungsi dengan baik. Alat ini sangat bermanfaat bagi teknisi dan insinyur dalam melakukan diagnosis dan perbaikan perangkat elektronik yang menggunakan berbagai jenis IC. Dengan adanya alat ini, proses pengecekan IC menjadi lebih efisien, akurat, dan mudah digunakan. Kami berharap proyek ini dapat memberikan manfaat yang signifikan dalam industri elektronik dan menjadi kontribusi yang berarti dalam mata kuliah Sistem Siber Fisik.

## REFERENCES

- [1] Microchip Technology Inc., "Atmel 7810 Automotive Microcontrollers ATmega328P Datasheet," [Online]. Available: [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf). [Accessed: 28-May-2024].
- [2] Microchip Technology Inc., "Atmel 0856 AVR Instruction Set Manual," [Online]. Available: <https://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual.pdf>. [Accessed: 28-May-2024].

## APPENDICES

### Appendix A: Project Schematic



## Appendix B: Documentation

AVR Source Code - ARD1

FinproSSF.S

```

----- ; -- Delay Functions --
----- ; Function Delay 20ms
delay_20ms:
0108    LDI    R17, 255
010A    delay_13: LDI    R18, 210
010C    delay_14: LDI    R19, 2
010E    delay_15: DEC    R19
0110    BRNE   delay_15
0112    DEC    R18
0114    BRNE   delay_14
0116    DEC    R18
0118    BRNE   delay_13
011A    RET    ; !!!!!!!RETURN!!!!!!!!!!!!!!
----- ; Function Delay 2s
delay_2s:
011C    LDI    R17, 255
011E    delay_16: LDI    R18, 255
0120    delay_17: LDI    R19, 164
0122    delay_18: DEC    R19
0124    BRNE   delay_18

```

AVR Variables - ARD1

Name	Address	Value
UDR0	008000C6	0x0D
UCSR0A	008000C0	...
UCSR0B	008000C1	0x18
UCSR0C	008000C2	0x06
UBRRO	008000C4	51
TWAMR	008000BD	'0'
TWBR	008000B8	'0'
TWCR	008000BC	'0'
TWSR	008000B9	'0'
TWDR	008000BB	0xFF
TWAR	008000BA	'0'
TIMSK1	0080006F	'0'
TIFR1	00800036	...

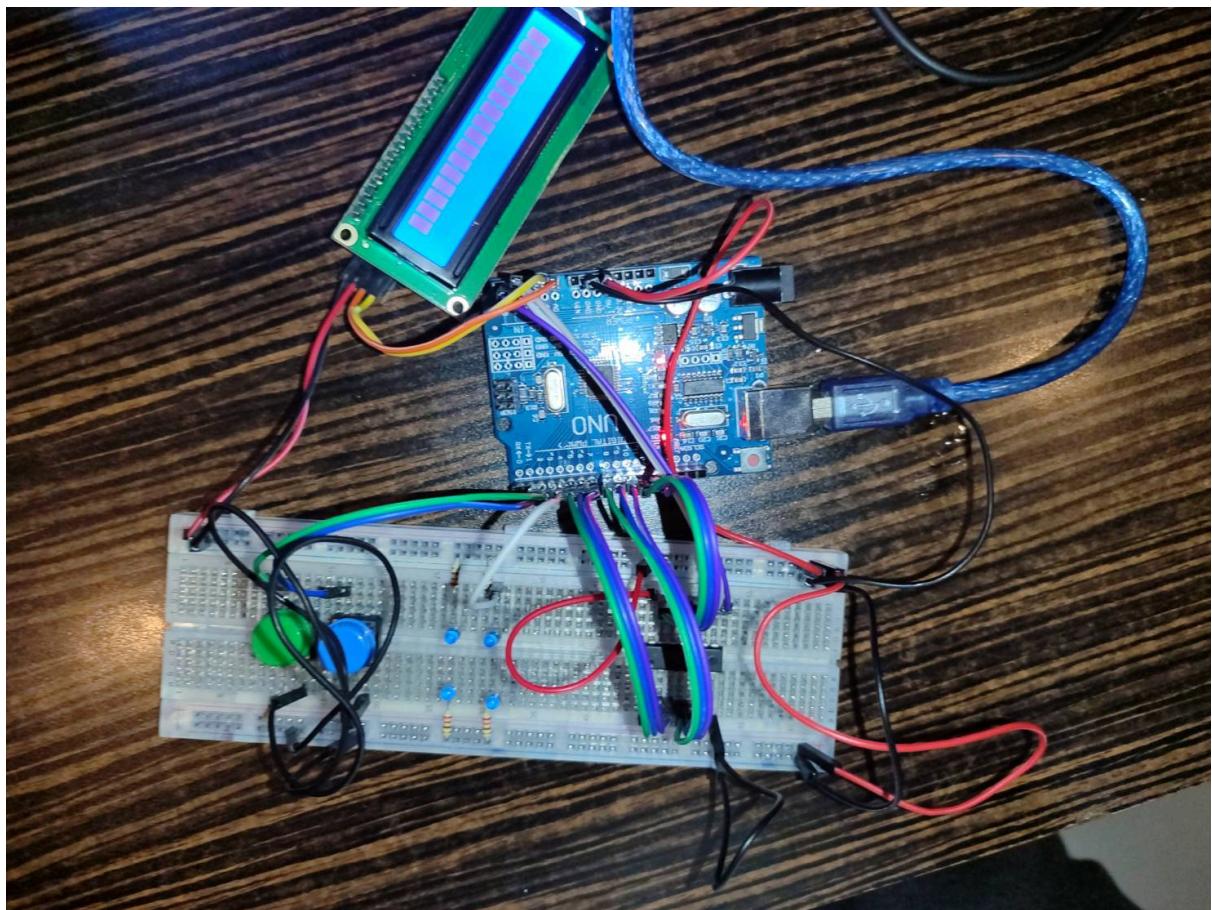
C Checker

```

G111 --- 1 --- U1:A
          |           |
          |           3 --- G3I1

```

Button Interrupt Confirm



## **Appendix C: Github Repository**

<https://github.com/EdgrantHS/Finpro-SSF>