



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

**CAR PARKING SENSOR SYSTEM**

**GROUP 23**

<b>Zikri Zulfa Azhim</b>	<b>2206028390</b>
<b>Faruq Sami Ramadhan</b>	<b>2206026675</b>
<b>Muhammad Abrisam Cahyo Juhartono</b>	<b>2206026050</b>
<b>Muhammad Rifki Pratama</b>	<b>2206828903</b>
<b>Nakita Rahma Dinanti</b>	<b>2206059401</b>

## PREFACE

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga kami dapat menyelesaikan proyek akhir yang berjudul "Car Parking Sensor System" ini dengan baik. Laporan ini disusun untuk memenuhi salah satu syarat dalam pemenuhan persyaratan tugas mata kuliah Praktikum Sistem Siber Fisik pada program studi Teknik Komputer, Fakultas Teknik Universitas Indonesia.

Dalam era modern ini, kebutuhan akan teknologi yang dapat mempermudah aktivitas manusia semakin meningkat. Salah satu teknologi yang sangat berguna adalah sistem sensor parkir. Proyek akhir ini bertujuan untuk merancang dan mengimplementasikan sistem sensor parkir yang dapat membantu pengemudi dalam memarkir kendaraan dengan lebih aman dan efisien.

Sistem sensor parkir yang kami rancang terdiri dari dua unit Arduino. Arduino *master* akan dihubungkan dengan sensor ultrasonik HC-SR04 yang berfungsi untuk mengukur jarak antara kendaraan dengan objek di belakangnya. Arduino *slave* akan dihubungkan dengan dua *buzzer* dan dua LED yang berfungsi sebagai indikator audio dan visual bagi pengemudi. Proses pengukuran jarak oleh sensor dibagi menjadi dua kategori: jika jarak yang terdeteksi masuk ke kategori pertama, satu *buzzer* akan aktif dan dua LED akan menyala, jika masuk ke kategori kedua, dua *buzzer* akan aktif dan satu LED akan menyala.

Kami menyadari bahwa laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat kami harapkan. Semoga laporan ini dapat memberikan manfaat dan kontribusi positif dalam pengembangan teknologi sensor parkir di masa mendatang.

Depok, May 28, 2024

Group 23

## 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.....	6
<b>CHAPTER 2.....</b>	<b>7</b>
<b>IMPLEMENTATION.....</b>	<b>7</b>
2.1 HARDWARE DESIGN AND SCHEMATIC.....	7
2.2 SOFTWARE DEVELOPMENT.....	8
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	8
<b>CHAPTER 3.....</b>	<b>10</b>
<b>TESTING AND EVALUATION.....</b>	<b>10</b>
3.1 TESTING.....	10
3.2 RESULT.....	10
3.3 EVALUATION.....	11
<b>CHAPTER 4.....</b>	<b>12</b>
<b>CONCLUSION.....</b>	<b>12</b>

## CHAPTER 1

### INTRODUCTION

#### 1.1 PROBLEM STATEMENT

Di era modern ini, mobilitas dan transportasi menjadi bagian integral dari kehidupan manusia. Seiring dengan berjalananya waktu, jumlah pengguna transportasi pribadi, terutama mobil, mengalami peningkatan. Karena itu, kebutuhan lahan parkir yang memadai juga mendesak. Tantangan yang dihadapi oleh pengemudi mobil terutama dalam situasi tempat parkir yang sempit dan padat adalah memarkir kendaraan dengan aman tanpa merusak kendaraan sendiri atau kendaraan lain.

Salah satu solusi yang bisa dilakukan untuk menangani masalah ini adalah menggunakan sensor parkir. Sensor parkir merupakan perangkat yang dapat membantu pengemudi mengetahui jarak antara kendaraannya dengan kendaraan atau objek di belakangnya, sehingga dapat meminimalisir risiko menabrak kendaraan atau objek saat parkir. Dengan adanya sensor yang bisa memberi tahu situasi di belakang mobil dapat membantu pengemudi mengetahui jarak sehingga bisa memarkirkan mobil dengan baik dan aman. Maka dari itu, kami tertarik untuk membuat perangkat sensor parkir mobil yang dapat mengeluarkan indikator berupa suara untuk memberi tahu jarak kepada pengemudi.

Rancangan perangkat ini adalah dengan mengimplementasikan sistem sensor parkir berbasis Arduino. Perangkat ini akan menggunakan dua buah unit Arduino yaitu satu sebagai Arduino *master* yang terhubung dengan sensor ultrasonik HC-SR04 untuk mengukur jarak, dan satu sebagai Arduino *slave* yang terhubung dengan dua *buzzer* dan dua LED sebagai indikator audio dan visual bagi pengemudi. Sistem ini dirancang untuk memberikan peringatan bertahap berdasarkan jarak yang terdeteksi oleh sensor, sehingga pengemudi dapat mengambil tindakan yang sesuai untuk menghindari tabrakan.

#### 1.2 PROPOSED SOLUTION

Proyek ini menggunakan sensor jarak HC-SR04 untuk mendeteksi adanya kendaraan atau objek di belakang mobil dimana sensor ultrasonik HC-SR04 akan mendeteksi objek tersebut dan memancarkan gelombang ultrasonik yang akan memantul dari objek yang

terdeteksi sehingga dapat mengukur jarak kedekatan dari objek terhadap mobil yang sedang mundur. Sensor yang terhubung dengan Arduino *master* akan mendeteksi objek dengan dua kategori. Arduino master akan mengirimkan sinyal ke Arduino *slave* untuk mengaktifkan *buzzer* dan akan menyalakan satu *buzzer* dan dua LED ketika ada di kategori pertama atau jarak jauh. Jika sudah masuk ke kategori kedua atau jarak dekat, dua *buzzer* akan menyala dan satu LED akan mati. Dengan menggunakan “*Car Parking Sensor System*”, akan memudahkan pengemudi dalam memarkirkan mobilnya.

### 1.3 ACCEPTANCE CRITERIA

Tujuan dari proyek ini adalah:

1. Keamanan: memberikan peringatan dini kepada pengemudi untuk mencegah tabrakan saat parkir.
2. Efisiensi: membantu pengemudi memarkirkan kendaraan dengan lebih aman dan akurat.
3. Kenyamanan: mengurangi kerepotan yang biasanya terjadi saat memarkirkan mobil di area yang sempit dan padat.

### 1.4 ROLES AND RESPONSIBILITIES

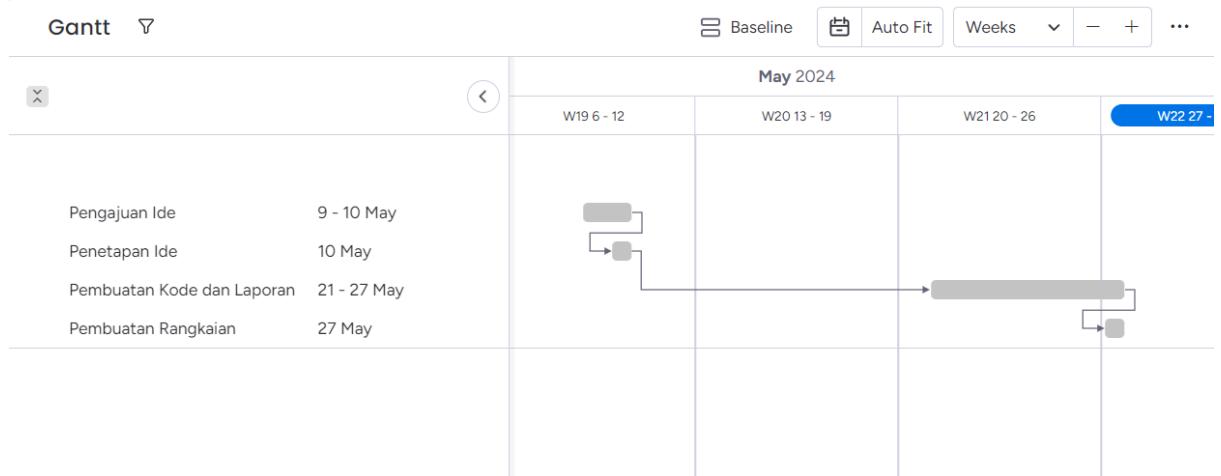
Berikut adalah peran dan tanggung jawab yang diberikan untuk tiap anggota kelompok:

Roles	Responsibilities	Person
Role 1	Membuat program dan membuat rangkaian asli dan proteus	Zikri Zulfa Azhim
Role 2	Membuat PPT, membantu membuat rangkaian asli	Faruq Sami Ramadhan
Role 3	Membuat PPT, membantu membuat rangkaian asli	Muhammad Abrisam Cahyo Juhartono
Role 4	Membuat laporan, membantu membuat rangkaian asli	Muhammad Rifki Pratama
Role 5	Membuat laporan, membantu membuat rangkaian asli	Nakita Rahma Dinanti

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

Gantt Chart untuk timeline pelaksanaan proyek akhir



## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC

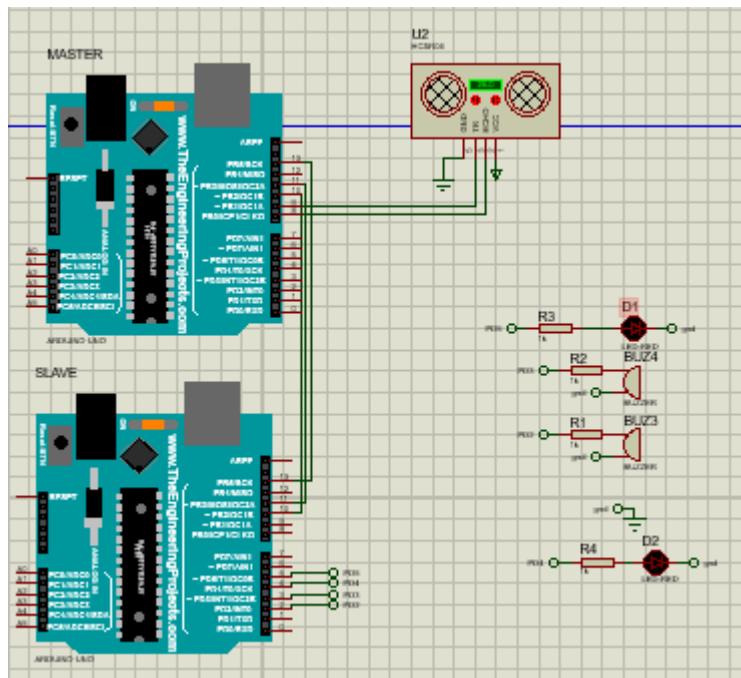
Sistem "Sensor Parkir Otomatis" ini tentunya memerlukan beberapa komponen untuk mencapai tujuan yang ingin dicapai. Dalam sistem ini, digunakan dua Arduino, yaitu Arduino *master* dan Arduino *slave*. Arduino master akan dihubungkan dengan sensor jarak HC-SR04, sedangkan Arduino *slave* akan dihubungkan dengan dua buah *buzzer*.

Sensor jarak HC-SR04 digunakan untuk menentukan jarak antara mobil dan objek yang ada di depannya. Sensor ini akan mengukur jarak dan mengkategorikannya ke dalam dua kategori. Kategori pertama adalah ketika jarak objek relatif jauh, dan kategori kedua ketika jarak objek sangat dekat. Berdasarkan pengukuran jarak tersebut, sistem akan mengaktifkan *buzzer* sesuai dengan kategori yang terdeteksi. Jika jarak objek masuk ke kategori pertama, maka terdapat satu *buzzer* yang akan dihidupkan dan Jika jarak objek masuk ke kategori kedua maka dua *buzzer* akan dihidupkan. Peralatan dan komponen yang digunakan dalam pembuatan proyek ini adalah:

- Arduino Uno dengan Microcontroller ATMega328P
- Sensor jarak HC-SR04
- Breadboard
- *Buzzer*
- Kabel Jumper
- Resistor

Untuk mengatur dan mengendalikan sistem ini, digunakan *microcontroller* pada kedua Arduino. Arduino *master* yang terhubung dengan sensor HC-SR04 akan mengirimkan data jarak ke Arduino *slave*. Arduino *slave* kemudian akan mengolah data tersebut dan mengaktifkan *buzzer* sesuai dengan kategori jarak yang telah ditentukan. Sistem ini dirancang agar pengemudi dapat mengetahui seberapa dekat mobil dengan objek di depannya melalui bunyi *buzzer*. Bunyi *buzzer* yang semakin banyak menunjukkan bahwa objek semakin dekat dengan mobil. Selain itu, sistem ini juga dilengkapi dengan tombol untuk menghidupkan atau mematikan sistem.

Dengan menggunakan dua Arduino, sistem ini memungkinkan pengukuran jarak yang lebih akurat dan responsif. Penggunaan dua *buzzer* dan dua LED memberikan *feedback* yang jelas kepada pengguna tentang jarak objek, sehingga meningkatkan keamanan dan kenyamanan dalam melakukan parkir.



## 2.2 SOFTWARE DEVELOPMENT

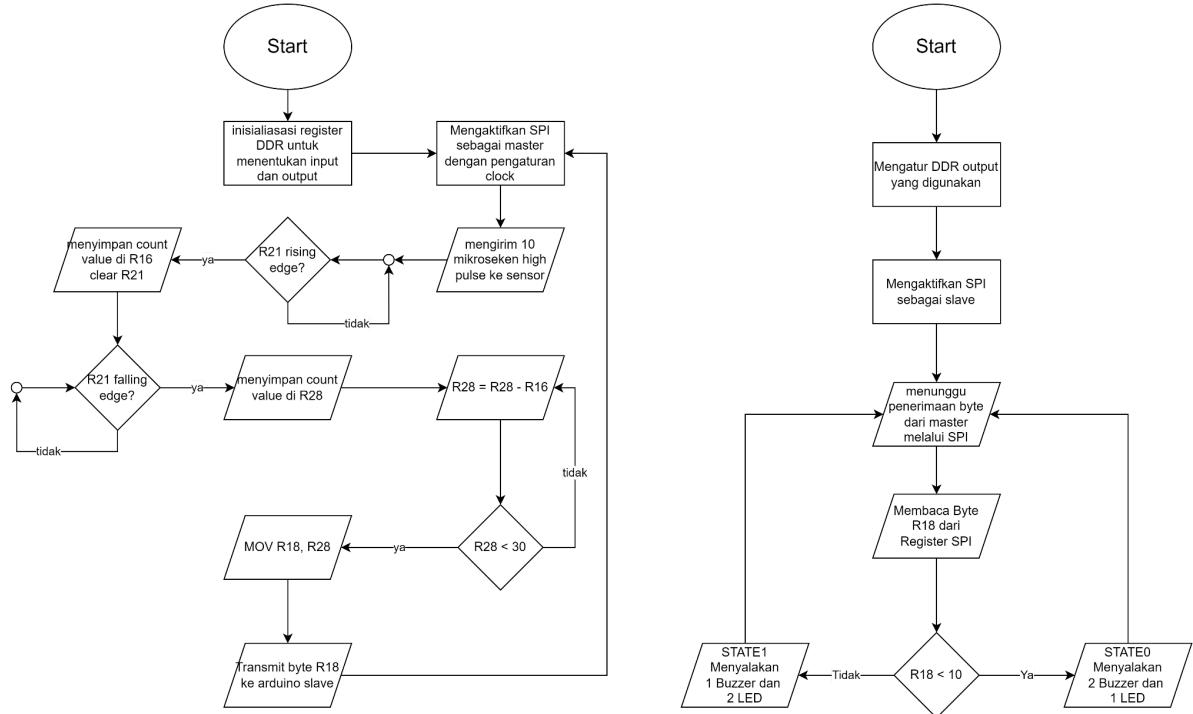
Pada pengembangan software ini, kami membuat kode yang sesuai dengan materi yang telah kami pelajari. Terdapat kode untuk sensor HC-SR04 dalam membaca jarak, serta kode untuk mengaktifkan *buzzer* berdasarkan kategori jarak yang terdeteksi. Terdapat juga kode untuk menghubungkan dua Arduino, di mana Arduino *master* mengirim data ke Arduino *slave*.

Arduino master akan menerima input dari sensor HC-SR04, mengukur jarak, dan mengkategorikannya ke dalam dua tingkat. Arduino slave akan mengaktifkan buzzer sesuai dengan kategori jarak yang diterima:

- Satu *buzzer* dan dua LED untuk kategori jarak pertama ( jauh).
  - Dua *buzzer* dan satu LED untuk kategori jarak kedua ( dekat).

Sebelum membuat kode program kami, pertama-tama kami membuat *flowchart* untuk menjadi gambaran dari jalannya program kami. *Flowchart* ini membantu dalam merancang

alur logika dan memastikan bahwa semua kondisi dan respon sistem telah dipertimbangkan dengan baik.



## 2.3 HARDWARE AND SOFTWARE INTEGRATION

Proses integrasi melibatkan penghubungan antara Arduino sebagai kontrol kendali dengan perangkat keras seperti sensor, arduino, dan *buzzer* sebagai input dan output. Kode yang dibuat diunggah ke Arduino sehingga *microcontroller* ATMega 328 dapat memproses input yang diberikan dan mengubahnya menjadi output yang sesuai dengan input tersebut.

Dengan melakukan integrasi, terciptalah sistem yang dapat berjalan untuk mengaktifkan satu atau dua *buzzer* dan LED berdasarkan hasil pembacaan dari sensor jarak HC-SR04. Hal ini dapat mempermudah pengemudi mobil dalam mengetahui seberapa dekat mobil dengan objek di sekitarnya melalui bunyi *buzzer* dan LED yang menyala, sehingga meningkatkan keamanan dan kenyamanan dalam parkir.

## CHAPTER 3

### TESTING AND EVALUATION

#### 3.1 TESTING

Pada proyek ini akan dilakukan pengujian, pengujian sendiri dilakukan untuk memastikan sistem yang sudah dibuat bekerja dengan semestinya dan mengecek mengecek output dari sistem berdasarkan input yang berbeda-beda.

- Menyalakan sistem: : Menyalakan sistem bertujuan untuk menguji apakah sistem dapat dinyalakan saat sistem dalam keadaan mati. Ketika sistem dinyalakan, maka sensor akan masuk kedalam mode reading untuk mengecek apakah ada objek atau tidak didepannya
- Pengujian sensor jarak: Ketika sistem sudah dinyalakan, sensor jarak HC-SR04 akan mulai membaca dan memberikan input kepada microcontroller. Pengukuran jarak dibagi ke dalam dua kategori yaitu jarak kategori jauh dan dekat, jika jauh maka dua led akan menyala dan satu buzzer akan dihidupkan, sementara jika dekat maka satu led akan menyala dan dua buzzer akan dihidupkan

Berikut adalah kode rangkaian yang dibuat pada Arduino IDE

- Master Arduino Code

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;-----
.global main
;=====
main:
.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<<SPI2X) | (1<<SPR0)
    OUT SPCR, R17 ;enable SPI as master, fsck=fosc/8, SPI mode 0
;-----
```

```

        SBI    DDRB, 1           ;pin PB1 as o/p (Trigger)
        CBI    DDRB, 0           ;pin PB0 as i/p (Echo)
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          ;call delay
        RJMP  agn
;
;-----
send: CBI  PORTB, SS ;enable slave device
        OUT  SPDR, R18 ;transmit byte to slave
;
;-----
loop:
        IN   R19, SPSR
        SBRS R19, SPIF ;wait for byte transmission
        RJMP loop ;to complete
;
;-----
        SBI  PORTB, SS ;disable slave device
        RET
;=====
=
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
;
;-----
l1: IN   R21, TIFR1
        SBRS R21, ICF1
        RJMP l1                 ;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
;
;-----
l2: 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 R28 = R28 - R16
OUT   TIFR1, R21      ;clear flag for next sensor reading
RET
;=====

byte2decimal:
;-----
CLR   R26          ;set counter1, initial value 0
CLR   R27          ;set counter2, initial value 0
CLR   R18
;-----
170: CPI   R28, 100      ;compare R28 with 100
Ret: BRMI 180          ;jump when R28 < 100
INC   R26          ;increment counter1 by 1
SUBI  R28, 100      ;R28 = R28 - 100
RJMP  170
;-----
180: CPI   R28, 30       ;compare R28 with 30
BRMI  dsp           ;jump when R28 < 30
INC   R27          ;increment counter2 by 1
SUBI  R28, 30       ;R28 = R28 - 30
RJMP  180
;-----
dsp: MOV   R18, R28      ;send R28 to R18
RCALL send
RET
;=====

;delay_subroutine
;=====

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
SBRS  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
;=====
=====

delay_2s:          ;delay 2s
    LDI R21, 255 ;LDI 255 into R21
116: LDI R22, 255 ;LDI 255 into R22
117: LDI R23, 164 ;LDI 164 into R23
118: DEC R23      ;Decrement R23
    BRNE 18      ;Branch if not equal to zero to label 18
    DEC R22      ;Decrement R22
    BRNE 17      ;Branch if not equal to zero to label 17
    DEC R21      ;Decrement R21
    BRNE 16      ;Branch if not equal to zero to label 16
    RET          ;Return from subroutine

```

## - Slave Arduino Code

```

#define __SFR_OFFSET 0x00
#include "avr/io.h"
;-----
.global main
;=====

```

```

main:
;-----
    LDI    R17, 0b00111100
    OUT    DDRD, R17          ;set PD2 for o/p
;-----
    LDI    R17, (1<<SPE)
    OUT    SPCR, R17          ;enable SPI as slave
;-----
agn:   IN     R18, SPSR
       SBRS  R18, SPIF        ;wait for byte reception
       RJMP  agn             ;to complete
;-----
       IN     R18, SPDR        ;i/p byte from data register
;-----
       CPI   R18, 10
       BRCS STATE0
       RJMP  STATE1
;=====

STATE0:
    SBI  PORTD, 5 ; menyalakan led pertama
    CBI  PORTD, 4 ; mematikan led kedua
    SBI  PORTD, 3 ; menyalakan buzzer pertama
    SBI  PORTD, 2 ; menyalakan buzzer kedua
    RJMP agn
;-----
-- 
STATE1:
    SBI  PORTD, 5 ; menyalakan led pertama
    SBI  PORTD, 4 ; menyalakan led kedua
    SBI  PORTD, 3 ; menyalakan buzzer pertama
    CBI  PORTD, 2 ; mematikan buzzer kedua
    RJMP agn
;=====

```

### 3.2 RESULT

Hasil dari pengujian membuktikan bahwa sistem dapat bekerja sesuai fungsinya pada kondisi-kondisi tertentu. Sistem dapat menghasilkan output yang sesuai ketika jarak objek

dekat atau jauh dari sensor. LED dan *buzzer* yang berfungsi sebagai indikator dapat mati ataupun menyala secara otomatis berdasarkan input dari sensor jarak HC-SR04.

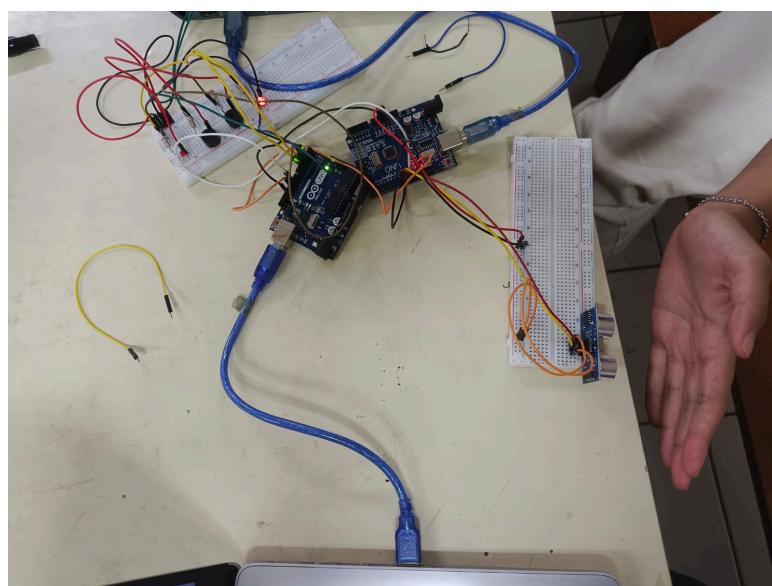
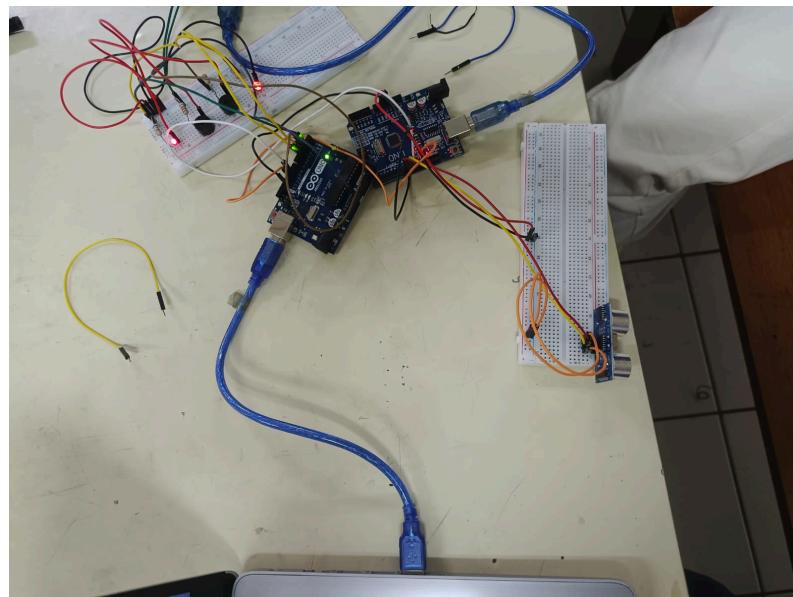


Fig 2. Testing Result

Ketika objek terdeteksi berada dalam jarak kurang dari 10 cm, satu LED akan menyala dan dua buzzer akan aktif, dan ketika objek terdeteksi berada dalam jarak lebih dari 10 cm, dua LED akan menyala dan satu buzzer akan aktif. Berdasarkan pengujian yang dilakukan, sistem *Car Parking Sensor System* memenuhi kriteria yang harus dipenuhi. Karena memenuhi kriteria, maka sistem *Car Parking Sensor System* dapat membantu proses

memarkirkan mobil karena bisa membantu pengemudi memperkirakan jarak parkir agar tidak terlalu jauh dan terlalu dekat.

### **3.3 EVALUATION**

Berdasarkan hasil, kami akan menjelaskan analisis mengapa task yang dilakukan oleh prototipe buatan kami berhasil. Overview dari program kami adalah, begitu sistem diaktifkan, subrutin sistem\_nyala melakukan branching dengan command SBIC terhadap bit pada PIND7, yang merupakan input dari sensor jarak HC-SR04, yang menentukan apakah kondisi jauh atau dekat yang akan dijalankan. Sejauh percobaan kami, semua komponen berfungsi dengan baik sehingga bisa mentranslasi perintah dari kode yang ada dengan cukup baik.

Ketika objek berada dalam jarak jauh, satu buzzer akan menyala. Jika objek berada dalam jarak dekat, dua buzzer akan menyala. Percobaan juga dilakukan dengan kondisi mendekatkan objek ke receiver dari sensor ultrasonik HC-SR04 sebagai representasi keadaan mendekat. Hasilnya, buzzer yang terhubung ke PIND2 akan mati ketika jarak jauh terdeteksi.

Namun, prototipe yang kami buat masih kurang reliabel dari segi konsistensi. Saat percobaan, buzzer tidak menyala secara konsisten, selain itu juga terdapat delay dalam pembacaan sensor dan penyalaan led, terkadang perlu beberapa saat untuk menunggu led menyala. Biasanya karena kabel yang kurang stabil terpasang, dan sensor yang sering mengalami masalah threshold saat pembacaan. Oleh karena itu, prototipe kami butuh disempurnakan sampai bisa mentoleransi input dan menghindari error seperti yang sudah disebutkan, sebelum akhirnya bisa diimplementasikan pada kasus-kasus yang ada di dunia nyata.

Proyek ini tidak menggunakan push button, sehingga sistem diatur untuk aktif secara otomatis saat dinyalakan. Hal ini menghilangkan kebutuhan intervensi manual dari pengguna untuk menyalakan atau mematikan sistem, memastikan bahwa sistem selalu siap untuk mendeteksi jarak dan memberikan umpan balik yang sesuai melalui buzzer.

## **CHAPTER 4**

### **CONCLUSION**

Perangkat “Sensor Parkir Otomatis” cocok untuk digunakan untuk pengemudi agar bisa memudahkan dalam melakukan proses parkir terutama di area parkir yang sempit dan penuh. Rangkaian ini akan menyalakan dua LED ketika objek berada pada jarak diatas 10 cm dan membunyikan satu *buzzer*. Jika objek berada pada jarak kurang dari 10 cm, satu LED akan menyala dan membunyikan dua buzzer.

Sensor HC-SR04 digunakan untuk mendeteksi adanya objek di belakang mobil dengan jarak tertentu. Terdapat 2 buah kondisi dimana apakah jarak mobil dan sensor lebih dari 10 cm atau kurang dari 10 cm, bila jarak lebih dari 10 cm maka device akan menyalakan 2 buah led dan 1 buah buzzer untuk menandakan bahwa jarak mobil masih jauh dari sensor, sementara jika jarak mobil dan sensor kurang dari 10 cm, maka device akan menyalakan 2 buah buzzer dan 1 led untuk menandakan bahwa mobil sudah terlalu dekat dengan sensor.

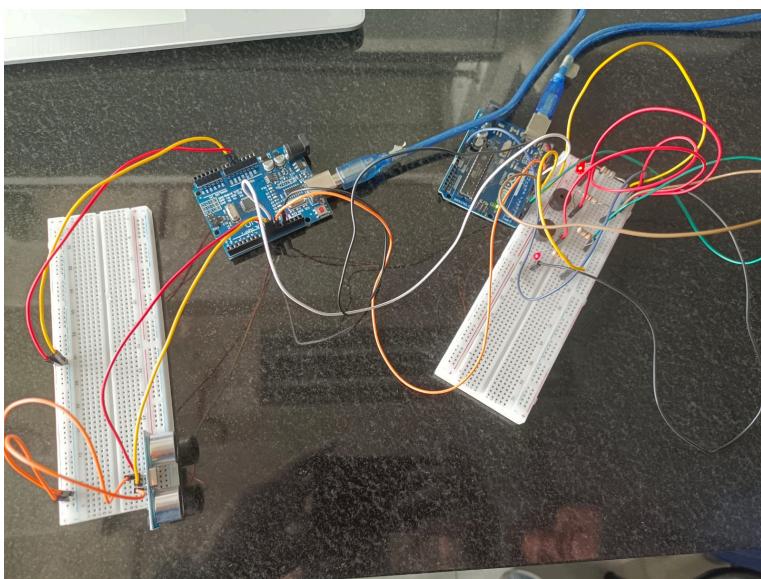
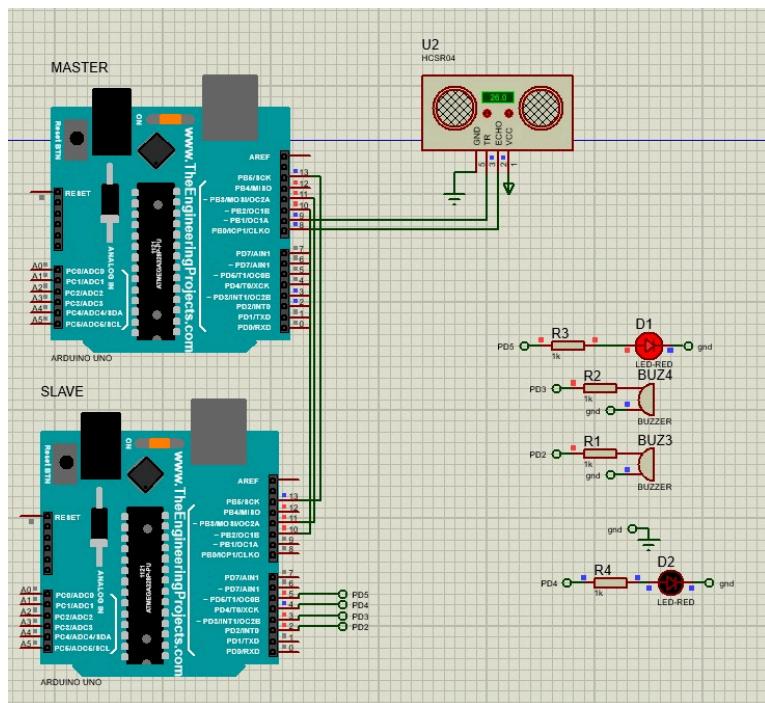
Link	Repo	Github	:
<a href="https://github.com/AbrisamYuhartono/Car-Parking-Sensor_SSF_Kelompok-23">https://github.com/AbrisamYuhartono/Car-Parking-Sensor_SSF_Kelompok-23</a>			

## **REFERENCES**

- [1] Digilab, In MODUL 1-9, Lab Module. Universitas Indonesia, 2023.
- [2] Last Minute Engineers, “How HC-SR04 ultrasonic sensor works & how to interface it with Arduino,” Last Minute Engineers, <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/> (accessed May 16, 2023)
- [3] Admin, “Ultrasonic Module HC-SR04 interfacing with AVR ATmega16/ATmega32 ..,” [www.electronicwings.com.](https://www.electronicwings.com/avr-atmega/ultrasonic-module-hc-sr04-interfacing-with-atmega1632)  
<https://www.electronicwings.com/avr-atmega/ultrasonic-module-hc-sr04-interfacing-with-atmega1632>

## APPENDICES

## **Appendix A: Project Schematic**



## Appendix B: Documentation

