



**EMBEDDED SYSTEM FINAL PROJECT REPORT  
DEPARTMENT OF ELECTRICAL ENGINEERING  
UNIVERSITAS INDONESIA**

**NOIR (Noise and Air Quality Monitoring System)**

**GROUP 20**

<b>Raka Arrayan Muttaqien</b>	<b>2306161800</b>
<b>Fathan Yazid Satriani</b>	<b>2306250560</b>
<b>Musyaffa Iman Supriadi</b>	<b>2306208464</b>
<b>Daffa Hardhan</b>	<b>2306161763</b>



## PREFACE

Dengan meningkatnya kesadaran akan pentingnya kualitas lingkungan dalam mendukung proses belajar yang efektif, kami mengembangkan sebuah sistem bernama NOIR (Noise and Air Quality Monitoring System). Sistem ini dirancang untuk memantau tingkat kebisingan dan kualitas udara secara real-time di lingkungan belajar seperti ruang kelas dan perpustakaan. Melalui kombinasi sensor suara dan gas, serta integrasi sistem komunikasi antar mikrokontroler menggunakan protokol SPI, NOIR mampu mendeteksi kondisi lingkungan yang tidak ideal dan memberikan peringatan secara visual maupun audio.

Sistem ini tidak hanya bertujuan untuk menciptakan suasana belajar yang kondusif, tetapi juga untuk menumbuhkan kesadaran akan pentingnya menjaga kualitas lingkungan sekitar. Motivasi kami dalam mengembangkan NOIR berasal dari keprihatinan terhadap gangguan belajar yang disebabkan oleh kebisingan maupun udara yang tidak sehat.

Dengan sistem ini, kami berharap dapat memberikan kontribusi nyata dalam meningkatkan kualitas pendidikan melalui teknologi pemantauan lingkungan. Kami mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan selama pengembangan proyek ini. Proyek NOIR tidak hanya memperluas pemahaman kami dalam bidang elektronika dan pemrograman sistem embedded, tetapi juga memperkuat semangat kami untuk menciptakan solusi inovatif yang berdampak positif bagi masyarakat.

Depok, May 17, 2025

Group 20



## CHAPTER 1

### INTRODUCTION

#### **1.1 PROBLEM STATEMENT**

Lingkungan belajar seperti ruang kelas dan perpustakaan sangat bergantung pada suasana yang tenang dan kualitas udara yang baik untuk menunjang konsentrasi dan efektivitas kegiatan belajar. Namun, dalam praktiknya, kondisi tersebut sering kali terganggu oleh tingkat kebisingan yang tinggi atau kualitas udara yang menurun akibat ventilasi yang buruk atau polusi udara dalam ruangan. Ketidakmampuan mendeteksi dan merespons kondisi lingkungan yang tidak ideal secara real-time dapat menyebabkan penurunan kenyamanan belajar, meningkatnya stres pada siswa, hingga berkurangnya produktivitas.

Selain itu, sebagian besar ruang belajar belum dilengkapi dengan sistem otomatis yang dapat memberikan peringatan saat kondisi lingkungan memburuk. Tidak adanya umpan balik langsung kepada pengguna membuat masalah lingkungan ini tidak segera ditangani. Penggunaan sistem monitoring konvensional juga sering memerlukan biaya mahal dan instalasi kompleks, sehingga tidak sesuai untuk diterapkan di berbagai institusi pendidikan dengan keterbatasan sumber daya. Oleh karena itu, dibutuhkan sebuah solusi teknologi yang hemat biaya, mudah diimplementasikan, serta mampu memantau kondisi lingkungan secara terus menerus dan memberikan peringatan secara real-time untuk menjaga kualitas lingkungan belajar tetap optimal.

#### **1.2 PROPOSED SOLUTION**

Untuk mengatasi masalah tersebut, kami mengembangkan sistem bernama NOIR (Noise and Air Quality Monitoring System). NOIR adalah perangkat yang dirancang untuk memantau tingkat kebisingan dan kualitas udara di lingkungan belajar seperti ruang kelas dan perpustakaan. Sistem ini akan membaca data dari sensor suara dan sensor gas secara berkala.

Sistem NOIR terdiri dari dua mikrokontroler yang berkomunikasi menggunakan protokol SPI satu berfungsi sebagai Slave yang bertugas mengumpulkan data sensor, dan satu lagi sebagai Master yang memproses data dan mengendalikan sistem peringatan. Selain itu, informasi kondisi lingkungan juga ditampilkan melalui Serial Monitor untuk memudahkan proses pemantauan. Dengan adanya sistem ini, pengguna dapat segera menyadari kondisi



lingkungan yang tidak kondusif dan mengambil tindakan yang diperlukan untuk menjaga kenyamanan belajar.

### 1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows :

1. Sistem dapat membaca data kebisingan menggunakan sensor suara KY-037 secara berkala melalui Arduino Slave.
2. Sistem dapat membaca data kualitas udara menggunakan sensor gas MQ-135 atau MQ-2 secara berkala melalui Arduino Slave.
3. Komunikasi antar Arduino Slave dan Master menggunakan protokol SP harus berjalan stabil dan responsif.
4. Arduino Master dapat mengevaluasi kondisi lingkungan dan menyalakan LED indikator yang sesuai:
  - LED Hijau untuk kondisi normal
  - LED Merah untuk kebisingan tinggi
  - LED Kuning untuk kualitas udara rendah
5. Sistem memberikan peringatan suara menggunakan buzzer ketika kondisi kebisingan atau udara tidak ideal.
6. Pengguna dapat mematikan buzzer secara manual dengan menekan push button.
7. Data kebisingan dan kualitas udara ditampilkan ke Serial Monitor melalui Arduino Master.

### 1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Role 1	Laporan, Rangkaian fisik, PPT, membantu membuat kode	Raka Arrayan Muttaqien
Role 2	Membantu membuat kode (mostly master) dan rangkaian proteus	Fathan Yazid Satriani
Role 3	Membuat kode dan rangkaian fisik	Musyaffa Iman Supriadi
Role 4	Rangkaian fisik dan membantu debug	Daffa Hardhan



## 1.5 TIMELINE AND MILESTONES

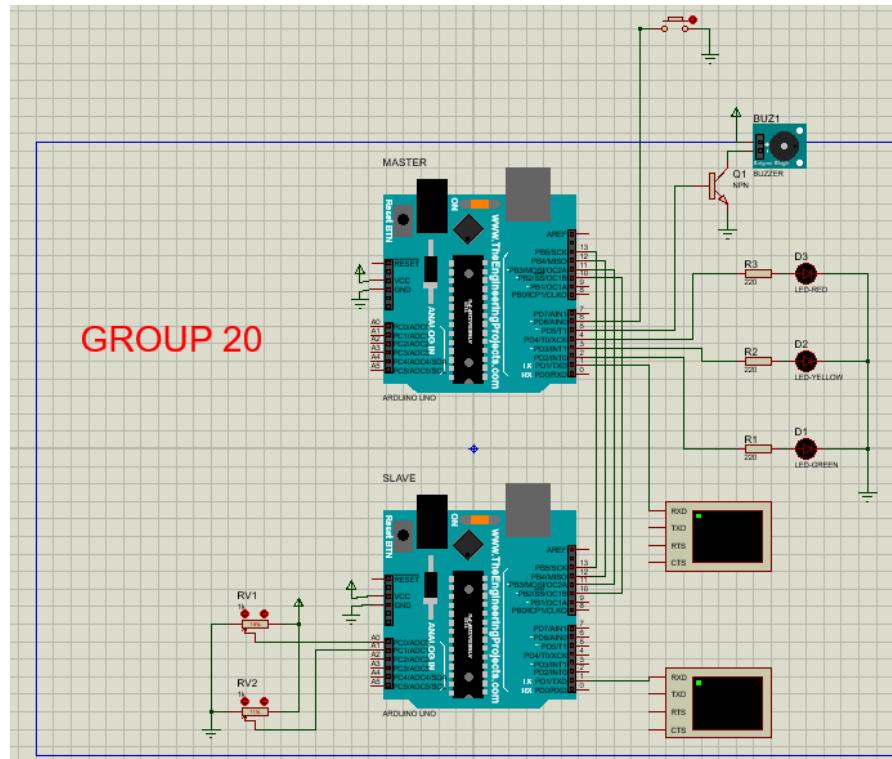
Kegiatan	Waktu Pelaksanaan
Penentuan Ide	5 - 7 Mei 2025
Kode	9- 18 Mei 2025
Merangkai proteus	9 - 18 Mei 2025
Merangkai rangkaian asli	16 Mei 2025
Final Product	18 Mei 2025



## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC



Hardware design pada proyek ini menggunakan bantuan Proteus, di mana schematic dari rangkaian dapat dilihat pada gambar di atas. Proyek ini mengusung sistem pemantauan kebisingan dan kualitas udara secara real-time di ruang belajar, dengan memanfaatkan dua buah Arduino (master dan slave) yang saling berkomunikasi melalui protokol SPI

Proyek ini menggunakan berbagai komponen utama, yaitu:

- Sensor suara KY-037 untuk mendeteksi tingkat kebisingan di ruangan
  - Sensor gas MQ-135/MQ-2 untuk memantau kualitas udara
  - Tiga buah LED indikator (merah, kuning, dan hijau) sebagai peringatan visual
  - Buzzer sebagai peringatan
  - Push button sebagai input user untuk menonaktifkan buzzer
  - serta komponen pendukung seperti resistor dan kabel jumper.
- Desain rangkaian menggunakan konfigurasi sebagai berikut
- LED merah (pin D2) digunakan sebagai indikator kebisingan berlebih,



- LED kuning (pin D3) sebagai indikator kualitas udara yang buruk,
- LED hijau (pin D4) menandakan kondisi lingkungan normal,
- Buzzer terhubung ke pin D5 sebagai peringatan saat terjadi pelanggaran ambang batas
- Push button pada pin D6 digunakan untuk mematikan suara buzzer secara manual.
- MOSI, MISO, dan SCK masing-masing berada pada pin 11, 12, dan 13 untuk komunikasi SPI antara master dan slave, dengan pin 10 sebagai Slave Select (SS).

Slave Arduino berfungsi untuk membaca data sensor secara berkala melalui pin analog A0 (untuk KY-037) dan A1 (untuk MQ-135), lalu mengirimkan data tersebut ke Master, Master kemudian memproses data dan mengendalikan LED serta buzzer berdasarkan nilai ambang batas yang telah ditentukan.

Modul ini juga menampilkan data sensor melalui Serial Monitor untuk memudahkan debugging dan kalibrasi. Terdapat fitur delay dan timer interrupt untuk memastikan pembacaan dan pengiriman data berlangsung stabil dan akurat.

Desain dan implementasi dari NOIR (Noise and Air Quality Monitoring System) bertujuan untuk meningkatkan kenyamanan dan keamanan lingkungan belajar. Dengan sistem peringatan visual dan auditori yang terintegrasi, pengguna dapat dengan cepat merespons kondisi ruangan yang tidak ideal, menjaga fokus dan produktivitas dalam kegiatan belajar.

Tetapi untuk sementara waktu pada rangkaian proteus kita menggunakan potentiometer terlebih dahulu hal ini bertujuan agar nilai tegangan input ke pin ADC dapat diatur secara manual, sehingga kita dapat menguji respon sistem terhadap berbagai kondisi input sensor.

## 2.2 SOFTWARE DEVELOPMENT

Kode assembly ini ditulis dalam bentuk .S yang digunakan untuk mengimplementasikan NOIR - Noise and Air Quality Monitoring System yang menggunakan sensor suara KY-037 untuk mendeteksi tingkat kebisingan di ruangan dan sensor gas MQ-135/MQ-2 untuk memantau kualitas udara.

Terdapat dua bagian kode yaitu master dan juga slave, berikut penjelasan singkat pada masing-masing kode. Berikut penjelasannya pada NOIR master.S:



## 1. Penentuan Pin

Pada bagian ini, program mendefinisikan beberapa konstanta (nilai tetap) untuk menyebut pin-pin yang digunakan

- LED\_SOUND = 0 → LED untuk suara dipasang di pin digital 0.
- LED\_GAS = 1 → LED untuk gas dipasang di pin digital 1.
- BUZZER = 2 → Buzzer dipasang di pin digital 2.
- BUTTON = 3 → Tombol (push button) dipasang di pin digital 3.
- SS\_PIN = 10 → Slave Select (pin SPI) dipasang di pin digital 10.

## 2. Deklarasi Label String dan Konstanta Memori

Bagian ini menyimpan teks statis (seperti "Sound: ", "Gas: ") di memori program menggunakan .section .progmem.

Dalam bagian ini, string yang ditampilkan ke pengguna (misalnya melalui UART/serial) dideklarasikan dan disimpan di flash memory untuk efisiensi. Contohnya, str\_sound dan str\_gas adalah label yang digunakan saat menampilkan hasil pembacaan sensor, dan string seperti str\_warning

## 3. Variabel Status Sistem (State Variables)

Variabel seperti button\_state, buzzer\_state, dan warnings\_active digunakan untuk menyimpan status sistem secara dinamis selama program berjalan.

Variabel-variabel ini disimpan di section .bss dan dialokasikan menggunakan .space. Mereka menyimpan data yang dapat berubah, seperti apakah tombol sedang ditekan (button\_state), apakah buzzer sedang aktif (buzzer\_state), atau apakah sedang ada peringatan (warnings\_active).

## 4. Integrasi Fungsi Arduino

Fungsi setup hanya dipanggil sekali saat awal menyalakan Arduino dan akan memanggil main\_init untuk melakukan konfigurasi awal sistem. Sementara itu, loop adalah bagian yang dipanggil berulang-ulang dan memanggil main\_loop untuk menjalankan logika

## 5. Fungsi main\_init untuk Inisialisasi Sistem

Bagian ini menginisialisasi komunikasi UART dan SPI, menetapkan arah pin input/output, menyalakan pull-up resistor, dan mengatur nilai awal variabel.



Di dalam main\_init, sistem mempersiapkan komunikasi serial UART pada baudrate 9600 untuk menampilkan informasi sensor ke komputer. Lalu, SPI dikonfigurasi sebagai master dengan pin-pin output yang sesuai. Semua LED dan buzzer diatur ke kondisi awal (mati), dan tombol disiapkan sebagai input dengan pull-up internal.

#### 6. Tes Awal dan Delay Sinkronisasi

Setelah konfigurasi awal, buzzer dinyalakan sebentar, dan master memberi jeda waktu 50ms agar slave siap menerima komunikasi SPI agar komunikasi master-slave dapat berjalan lancar. Master tidak langsung membaca data dari slave, melainkan memberi delay menggunakan RCALL delay\_ms untuk memastikan slave sudah selesai inisialisasinya

#### 7. Fungsi main\_loop

Di bagian main\_loop, master melakukan pekerjaan utamanya yaitu berkomunikasi secara rutin dengan slave melalui SPI untuk mengambil data sensor suara dan gas. Berdasarkan nilai tersebut, master menentukan apakah perlu menyalakan LED peringatan dan buzzer. Jika tombol ditekan, buzzer bisa dimatikan sementara. Logika ini terus berulang

### Penjelasan pada NOIR\_Slave.S

#### 1. Penentuan Pin

Pada bagian awal, program tidak secara eksplisit mendefinisikan nama-nama pin seperti pada master, tetapi melalui konfigurasi register DDRB dan DDRD, pin SPI seperti MISO, MOSI, dan SCK diatur. Pin MISO (PB4) diset sebagai output karena data dikirim dari slave ke master, sementara MOSI dan SCK diset sebagai input.

#### 2. Deklarasi String dan Memori Program

Tidak ada deklarasi string seperti di master karena slave tidak menampilkan teks ke pengguna. Fokus utamanya adalah membaca data sensor dan mengirimnya ke master melalui SPI.

#### 3. Variabel dan Konstanta di Memori Data

Program menggunakan beberapa variabel untuk menyimpan hasil pembacaan sensor seperti sound\_value dan gas\_value. Variabel ini disimpan dalam section .bss dengan



menggunakan direktif .def dan .byte, artinya mereka akan berubah selama program berjalan dan digunakan untuk menyimpan data sementara dari sensor.

#### 4. Inisialisasi UART

UART diatur untuk keperluan debugging (menampilkan nilai sensor lewat serial ke komputer). Komunikasi UART diatur dengan baudrate 9600. Register seperti UBRR0, UCSR0B, dan UCSR0C diatur agar bisa mengirim dan menerima data dalam format 8-bit.

#### 5. Inisialisasi SPI sebagai Slave

Program mengaktifkan SPI dalam mode slave. Pin MOSI dan SCK disiapkan sebagai input, sedangkan MISO sebagai output. SPI diaktifkan dengan mengatur bit SPE pada register SPCR. Slave akan menerima data dari master dan menyiapkan data balasan

#### 6. Inisialisasi ADC (Analog to Digital Converter)

ADC digunakan untuk membaca data dari dua sensor yaitu sensor suara dan sensor gas. ADC diatur dengan referensi AVcc (5V) dan hasil konversi dibaca 8-bit (disimpan di ADCH). Prescaler diatur agar kecepatan ADC tepat (125 kHz), dan saluran ADC dipilih sesuai dengan sensor yang akan dibaca (misalnya ADC0 dan ADC1).

#### 7. Fungsi read\_adc

Fungsi read\_adc dipakai untuk membaca data dari sensor analog. Program memilih saluran ADC sesuai input (sensor suara atau gas), lalu memulai konversi dan menunggu sampai selesai. Hasilnya disimpan di register untuk dikirim ke master.

#### 8. Fungsi Setup (setup)

Fungsi setup memanggil fungsi-fungsi inisialisasi seperti uart\_init, spi\_slave\_init, dan adc\_init. Fungsinya adalah untuk menyiapkan semua sistem agar siap digunakan, termasuk komunikasi dan pembacaan sensor.

#### 9. Fungsi Loop (loop)

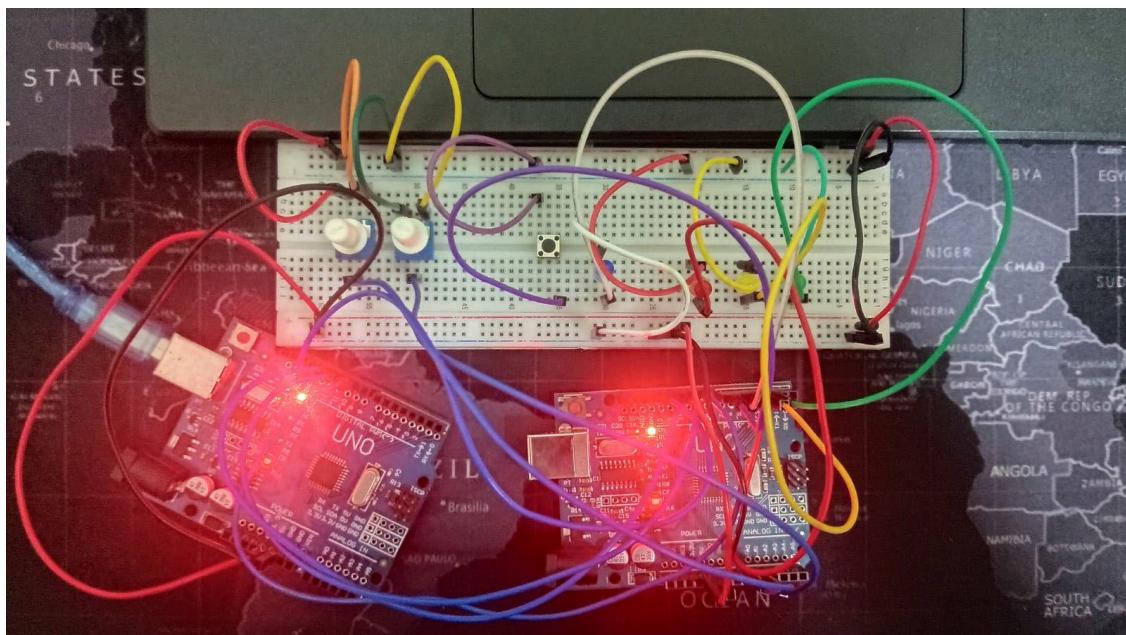
Fungsi loop dijalankan terus-menerus. Di dalamnya, slave menerima perintah dari master via SPI, memeriksa data perintah tersebut, dan memberikan respon berupa nilai sensor suara atau gas sesuai permintaan. Hasil pembacaan sensor juga bisa dikirim lewat UART untuk ditampilkan di serial monitor.

#### 10. Fungsi SPI Transfer (spi\_transfer)



Fungsi ini digunakan untuk mengirim data ke master dan juga menerima data dari master. Karena dalam SPI slave tidak bisa memulai komunikasi, maka slave hanya menunggu sampai ada data masuk, lalu membalas dengan nilai sensor.

### 2.3 HARDWARE AND SOFTWARE INTEGRATION



NOIR (Noise and Air Quality Monitoring System) mengintegrasikan berbagai komponen hardware seperti dua board Arduino (sebagai Master dan Slave), sensor kebisingan KY-037, sensor gas MQ-135, tiga buah LED indikator (merah, kuning, hijau), buzzer, serta sebuah push button. Penggunaan software dalam proyek ini mencakup konfigurasi pin I/O, inisialisasi komunikasi SPI antara Arduino Master dan Slave, pembacaan data sensor secara berkala, serta evaluasi kondisi lingkungan berdasarkan ambang batas tertentu. Komunikasi data antar Arduino dilakukan melalui protokol SPI dengan memanfaatkan pin MOSI (11), MISO (12), SCK (13), dan SS (10). Master secara rutin meminta data dari Slave dan kemudian memproses data tersebut untuk menentukan kondisi lingkungan.

Sistem akan memberikan peringatan secara visual melalui LED dan auditori melalui buzzer. Selain itu, pengguna juga dapat menghentikan peringatan dengan menekan tombol yang terhubung ke Master. Dengan demikian, integrasi antara hardware dan software pada



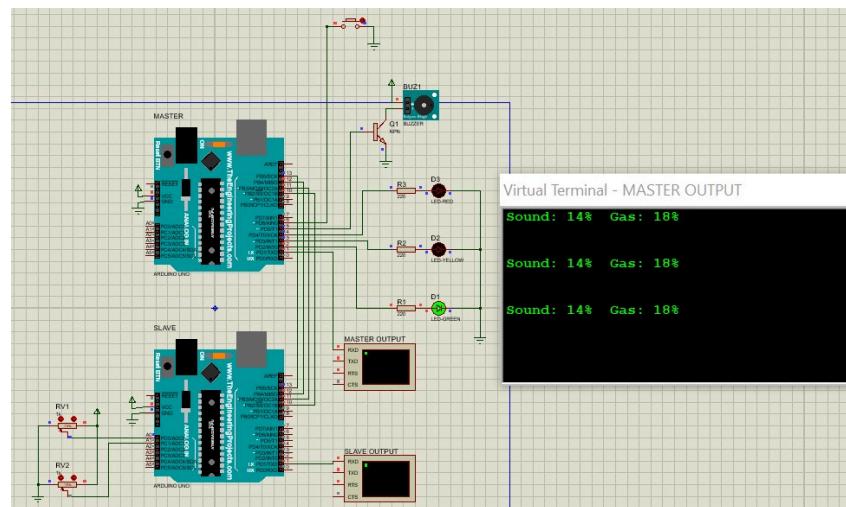
sistem NOIR mampu menjalankan fungsinya secara efektif dalam memantau dan menjaga kenyamanan lingkungan belajar melalui pemantauan kebisingan dan kualitas udara secara real-time.



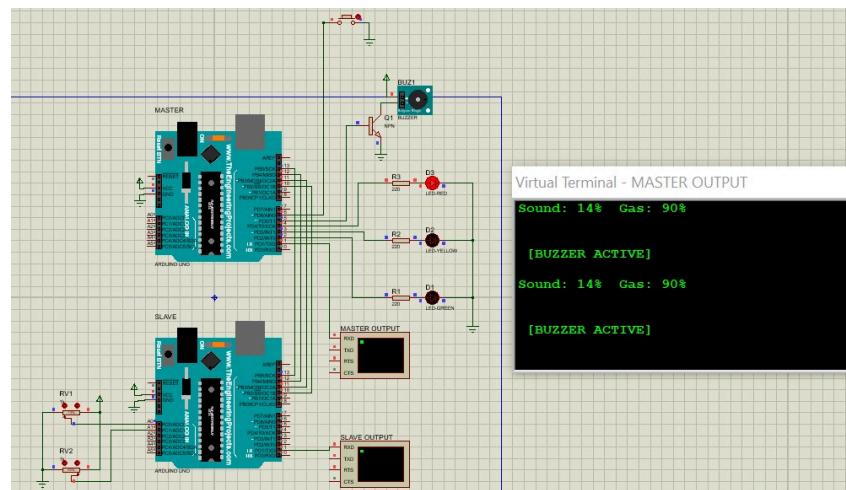
## CHAPTER 3

### TESTING AND EVALUATION

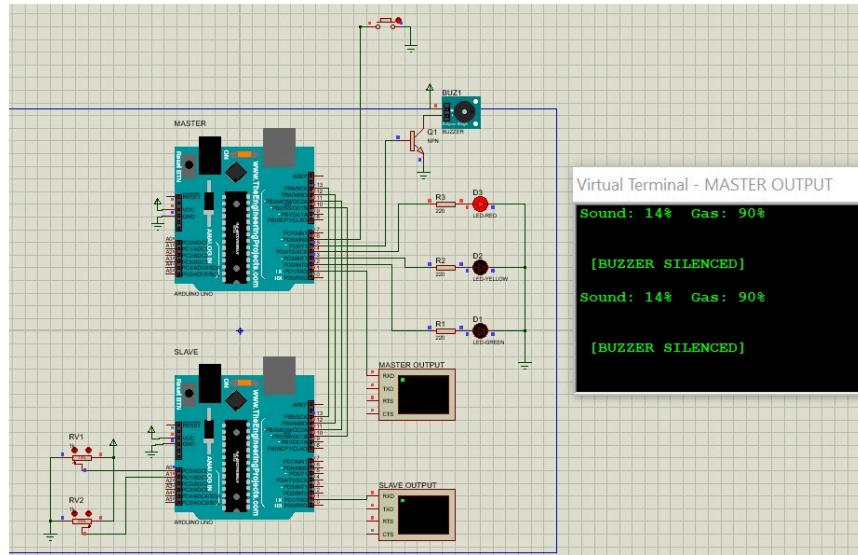
#### 3.1 TESTING



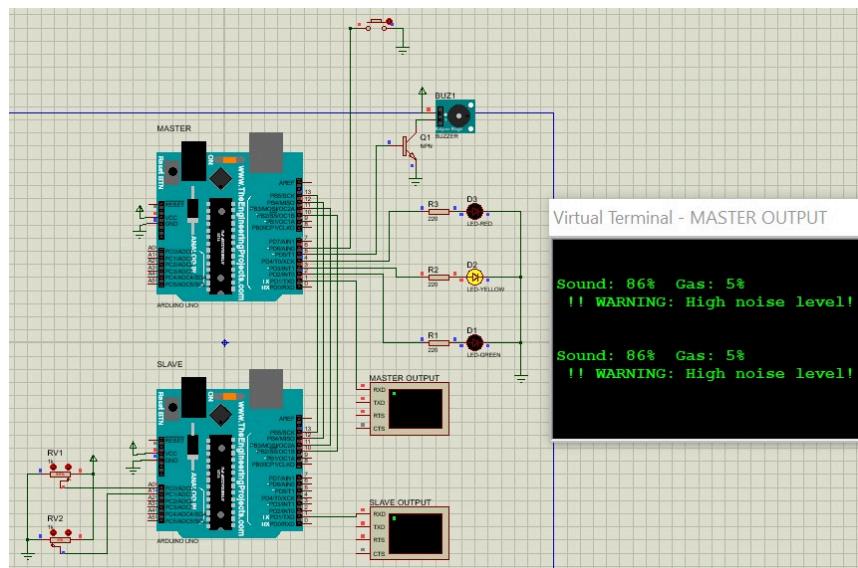
Kondisi Normal



Peringatan Kebisingan (Buzzer Menyalा)



Peringatan Kebisingan (Buzzer Dimatikan)



Peringatan Kualitas Udara

Berdasarkan hasil testing yang telah dilakukan, seluruh percobaan dapat berjalan dengan baik sesuai dengan ekspektasi. Sistem berhasil melakukan komunikasi SPI antara master dan slave, serta mampu membaca data sensor suara dan gas melalui ADC pada slave yang kemudian dikirimkan ke master. Master juga mampu merespons kondisi tersebut dengan menyalakan LED atau buzzer sebagai bentuk peringatan. Untuk keperluan simulasi pada Proteus, sensor suara dan sensor gas yang seharusnya berupa sensor fisik pada rangkaian asli,tetapi dalam tahap ini sementara digantikan menggunakan potensiometer. Hal ini



bertujuan agar nilai tegangan input ke pin ADC dapat diatur secara manual, sehingga kita dapat menguji respon sistem terhadap berbagai kondisi input sensor. Dengan demikian, dapat disimpulkan bahwa sistem mampu bekerja dengan baik dan logika komunikasi master-slave serta proses pengambilan keputusan telah terintegrasi dengan efektif antara perangkat keras dan perangkat lunak.

### 3.2 RESULT

Hasil dari implementasi sistem NOIR (Noise and Air Quality Monitoring System) menunjukkan performa yang cukup baik dalam mendeteksi kebisingan dan kualitas udara di lingkungan belajar. Seluruh fungsi berjalan sesuai dengan rancangan, di mana data dari sensor suara dan sensor gas berhasil dikumpulkan oleh Arduino Slave dan dikirim secara berkala ke Arduino Master menggunakan komunikasi SPI. Komunikasi SPI menunjukkan kestabilan dan akurasi yang tinggi dalam pertukaran data antar perangkat, serta mampu mempertahankan sinkronisasi yang baik antara Master dan Slave.

Fitur monitoring bekerja secara real-time, di mana sistem dapat membedakan antara kondisi normal dan tidak ideal berdasarkan ambang batas (threshold) yang telah dikalibrasi. Sistem juga mampu memberikan peringatan visual (melalui LED Merah, Kuning, dan Hijau) dan peringatan auditori (melalui buzzer) dengan baik. LED Hijau menyala saat lingkungan dalam kondisi normal, LED Merah aktif saat tingkat kebisingan melebihi batas, dan LED Kuning menandakan kualitas udara buruk. Buzzer menyala sebagai bentuk peringatan tambahan dan dapat dimatikan oleh pengguna melalui tombol yang disediakan.

Seluruh peringatan dieksekusi dengan lancar, termasuk pengendalian nyala LED yang disesuaikan dengan kondisi, serta bunyi buzzer yang dapat dinonaktifkan secara manual. Fitur ini memastikan bahwa peringatan tidak hanya otomatis tetapi juga dapat direspon oleh pengguna secara langsung.

Parameter	Status
LED Gas menyala sesuai nilai sensor	SUCCESS
Buzzer aktif saat nilai sensor melebihi ambang batas	SUCCESS
Tombol (button) dapat menonaktifkan buzzer	SUCCESS



Nilai sensor (potensiometer) terbaca dengan benar oleh slave	SUCCESS
Master menerima data SPI dari slave	SUCCESS
LED dan buzzer kembali mati saat kondisi normal	SUCCESS
Simulasi di Proteus berjalan lancar	SUCCESS

### 3.3 EVALUATION

Berdasarkan hasil dari sistem yang telah diuji, didapatkan beberapa evaluasi penting untuk meningkatkan kualitas dan kinerja proyek ini. Pertama, pada saat simulasi menggunakan Proteus, sensor diganti sementara dengan potensiometer sebagai pengganti nilai input sensor asli, sehingga nilai yang terbaca masih bersifat simulasi dan belum sepenuhnya merepresentasikan kondisi nyata. Selanjutnya, meskipun seluruh komponen seperti LED, buzzer, dan tombol telah berfungsi sesuai harapan, masih diperlukan kalibrasi dan pengujian lebih lanjut untuk memastikan respons sensor yang lebih akurat dan stabil. Selain itu, button yang digunakan untuk menonaktifkan buzzer sudah berjalan dengan baik, namun perlu pengujian dalam kondisi nyata agar respons tombol lebih optimal dan bebas dari gangguan noise.

Pengujian di berbagai kondisi lingkungan juga perlu dilakukan untuk memastikan sistem dapat bekerja dengan baik dalam situasi berbeda. Integrasi antara hardware dan software sudah berjalan lancar, namun pengembangan lebih lanjut bisa dilakukan untuk menambahkan fitur pengaturan sensitivitas sensor atau notifikasi tambahan agar sistem semakin responsif dan dapat diandalkan. Dengan begitu, evaluasi ini menjadi dasar penting untuk penyempurnaan sistem agar dapat bekerja lebih efektif dan efisien dalam penerapan sesungguhnya.



## **CHAPTER 4**

### **CONCLUSION**

Berdasarkan hasil perancangan dan simulasi yang telah dilakukan, sistem monitoring master-slave berbasis komunikasi SPI berhasil dijalankan dengan baik. Komponen-komponen seperti LED, buzzer, dan tombol bekerja sesuai dengan fungsinya. Data dari slave berupa nilai sensor suara dan gas dapat diterima oleh master secara berkala, kemudian digunakan untuk menentukan kondisi sistem, seperti menyalakan LED peringatan atau buzzer. Sistem ini mampu merespons dengan baik terhadap perubahan nilai sensor secara real-time melalui komunikasi SPI.

Dalam simulasi menggunakan Proteus, sensor suara dan gas digantikan sementara oleh potensiometer untuk mensimulasikan perubahan nilai input. Hal ini dilakukan untuk mempermudah proses pengujian logika sistem tanpa perlu menggunakan sensor fisik secara langsung. Meskipun begitu, fungsi logika dan alur komunikasi tetap berjalan sesuai harapan, dan sistem mampu memberikan reaksi yang tepat terhadap nilai input yang diberikan.

Secara keseluruhan, sistem ini menunjukkan kinerja yang baik dalam hal pengambilan data sensor, pengambilan keputusan, dan pemberian peringatan. Sistem juga cukup fleksibel untuk dikembangkan lebih lanjut, misalnya dengan menambahkan sensor-sensor lain, menyesuaikan ambang batas deteksi, atau mengintegrasikannya ke dalam sistem monitoring skala lebih besar. Dengan pengujian lanjutan menggunakan sensor asli dan lingkungan nyata, sistem ini berpotensi menjadi solusi yang berguna untuk monitoring.



## REFERENSI

[1] M. Karim Khondaker, “Arduino And MQ2 Gas Sensor,” projecthub.arduino.cc, Mar. 16, 2016. Available:

[https://projecthub.arduino.cc/m\\_karim02/arduino-and-mq2-gas-sensor-f3ae33](https://projecthub.arduino.cc/m_karim02/arduino-and-mq2-gas-sensor-f3ae33)

[2] Digital Laboratory UI, “Modul 3 SSF : Analog to Digital Converter,” Google Docs. Available:

<https://docs.google.com/document/d/1arLt3fqXRw-WgkbqlP1RwYs-XJy9-QAFFEcM21M3u44/edit?usp=sharing>

[3] Digital Laboratory UI, “Modul 4 SSF : Serial Port,” Google Docs, 2019. Available:

[https://docs.google.com/document/d/1rRWvBgL3Nsb\\_h10131A-1kiGQkrGGNLedYoeVIGR9zg/edit?tab=t.0](https://docs.google.com/document/d/1rRWvBgL3Nsb_h10131A-1kiGQkrGGNLedYoeVIGR9zg/edit?tab=t.0)

[4] Digital Laboratory UI, “Modul 8 SSF : SPI & I2C,” Google Docs, 2020. Available:

<https://docs.google.com/document/d/1CsIbwLVUrsKjZ3YhyF0J-gsGWNU1RCQu7JTYMCx3cFY/edit?tab=t.0>