



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

# **EMBEDDED SYSTEM DESIGN FOR AUTOMATIC ROOM TEMPERATURE STABILIZATION**

## **GROUP 21**

<b>Rivi Yasha Hafizhan</b>	<b>2306250635</b>
<b>Grace Yunike Margaretha Sitorus</b>	<b>2306267031</b>
<b>Nabiel Harits Utomo</b>	<b>2306267044</b>
<b>Muhammad Arya Wiandra Utomo</b>	<b>2306218295</b>

## **PREFACE**

Segala puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga kelompok kami dapat menyelesaikan proyek akhir mata kuliah Sistem Embedded dengan judul "Embedded System Design for Automatic Room Temperature Stabilization" tepat pada waktunya.

Proyek ini bertujuan untuk merancang dan mengimplementasikan sistem pengatur suhu otomatis berbasis mikrokontroler ATmega328P, yang diberi nama TempShield. Sistem ini dirancang untuk mengaktifkan kipas secara otomatis berdasarkan suhu lingkungan yang terdeteksi oleh sensor. Kegunaannya dapat diaplikasikan pada ruang penyimpanan sensitif, inkubator, maupun ruangan pribadi seperti kamar tidur, di mana kestabilan suhu sangat penting. Melalui proyek ini, kami mempelajari berbagai aspek pemrograman mikrokontroler, integrasi sensor suhu, aktuator, serta pemanfaatan komunikasi I2C untuk tampilan data secara real-time.

Kami, kelompok 21, mengucapkan terima kasih sebesar-besarnya kepada seluruh anggota kelompok atas kerja sama, dedikasi, dan kontribusi selama proses perancangan, pembuatan rangkaian fisik maupun proteus, melakukan simulasi, serta penyusunan laporan ini. Ucapan terima kasih juga kami sampaikan kepada asisten laboratorium yang telah membimbing kami selama proses pengerjaan berlangsung.

Kami menyadari bahwa proyek ini masih memiliki kekurangan dan ruang untuk perbaikan. Oleh karena itu, kami sangat terbuka terhadap kritik dan saran yang membangun dari para pembaca, agar karya ini dapat menjadi lebih baik dan bermanfaat di masa yang akan datang.

Depok, 18 Mei 2025

Group 21

## **TABLE OF CONTENTS**

### **CHAPTER 1: INTRODUCTION**

- 1.1 Problem Statement
- 1.2 Proposed Solution
- 1.3 Acceptance Criteria
- 1.4 Roles and Responsibilities

### **CHAPTER 2: IMPLEMENTATION**

- 2.1 Hardware Design and Schematic
- 2.2 Software Development
- 2.3 Hardware and Software Integration

### **CHAPTER 3: TESTING AND EVALUATIO**

- 3.1 Testing
- 3.2 Result
- 3.3 Evaluation

### **CHAPTER 4: CONCLUSION**

### **REFERENCES**

### **APPENDICES**

- Appendix A: Project Schematic
- Appendix B: Documentation

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 PROBLEM STATEMENT**

Kestabilan suhu di ruang tertutup menjadi faktor krusial dalam berbagai konteks, terutama ketika berhubungan dengan objek atau makhluk hidup yang sensitif terhadap perubahan suhu. Dua contoh nyata yang menyoroti pentingnya pengendalian suhu adalah inkubator bayi dan ruang penyimpanan alat elektronik. Pada kedua kasus tersebut, fluktuasi suhu sekecil apapun dapat berakibat fatal, baik terhadap kesehatan maupun performa alat.

Dalam konteks medis, suhu yang tidak stabil pada inkubator dapat mengganggu termoregulasi bayi baru lahir, khususnya bayi prematur. Menurut National Institutes of Health (NIH), bayi yang mengalami suhu tubuh tidak stabil memiliki risiko lebih tinggi mengalami gangguan metabolisme dan komplikasi lainnya. Oleh karena itu, sistem inkubator modern harus mampu memantau dan menjaga suhu dalam rentang yang sangat presisi secara berkelanjutan.

Sementara itu, alat elektronik seperti komputer server, perangkat laboratorium, atau penyimpanan digital sangat rentan terhadap overheating. Peningkatan suhu dapat mempercepat degradasi komponen, memperpendek umur pakai, dan menyebabkan gangguan sistem. Berdasarkan studi IEEE, kenaikan suhu sekitar 10°C saja dapat memangkas masa pakai komponen elektronik hingga 50%.

Sayangnya, sistem pengendalian suhu konvensional yang digunakan di banyak tempat seringkali masih bersifat manual atau non-adaptif, tidak responsif terhadap kondisi lingkungan aktual secara real-time, serta kurang efisien secara energi. Hal ini menciptakan kebutuhan akan solusi pengatur suhu yang otomatis, adaptif, dan terjangkau yang dapat memberikan respons langsung terhadap perubahan suhu serta dapat digunakan dalam berbagai skenario lingkungan mikro.

## **1.2 PROPOSED SOLUTION**

Sebagai respon terhadap kebutuhan pengendalian suhu yang adaptif dan efisien pada lingkungan sensitif seperti inkubator bayi dan penyimpanan alat elektronik, kami mengembangkan TempShield, sebuah sistem embedded berbasis mikrokontroler ATmega328P. Sistem ini memanfaatkan sensor DHT11 untuk memantau suhu lingkungan secara real-time dan secara otomatis mengatur kecepatan kipas pendingin berdasarkan tiga mode operasional: low, medium, dan high speed. Pendekatan ini memungkinkan sistem memberikan pendinginan yang proporsional sesuai kondisi aktual, sehingga mencegah overheating sekaligus menjaga efisiensi energi.

Sebagai bentuk umpan balik visual kepada pengguna, sistem ini dilengkapi dengan LCD 16x2 berbasis I2C yang menampilkan nilai suhu ruangan dan status kecepatan kipas secara real-time. Dengan logika yang sederhana namun responsif, TempShield menawarkan solusi pengaturan suhu otomatis yang fleksibel, hemat energi, dan mudah diimplementasikan. Sistem ini dirancang agar dapat diterapkan dalam berbagai skenario, dari kebutuhan medis skala kecil hingga ruang penyimpanan perangkat elektronik, tanpa bergantung pada mekanisme pemanas atau sistem pendingin konvensional yang lebih kompleks dan mahal.

## **1.3 ACCEPTANCE CRITERIA**

Tujuan dari proyek ini adalah sebagai berikut:

1. Sistem mampu membaca suhu lingkungan secara real-time menggunakan sensor DHT11.
2. Sistem dapat menampilkan nilai suhu yang terbaca pada LCD 16x2 (I2C) secara akurat dan berkelanjutan.
3. Sistem dapat mengatur kecepatan kipas secara otomatis berdasarkan kategori suhu yang telah ditentukan: kecepatan rendah (low speed) untuk suhu  $\leq 24^{\circ}\text{C}$ , kecepatan sedang (medium speed) untuk suhu  $25 - 29^{\circ}\text{C}$ , dan kecepatan tinggi (high speed) untuk suhu  $\geq 30^{\circ}\text{C}$ .
4. Sistem menampilkan informasi mode kipas aktif saat ini pada LCD secara bersamaan dengan nilai suhu.
5. Sistem bekerja secara stabil dan responsif terhadap perubahan suhu lingkungan.

## 1.4 ROLES AND RESPONSIBILITIES

Berikut merupakan pembagian peran dan tanggung jawab dalam kelompok selama proses perancangan dan pembuatan proyek TempShield:

<b>Roles</b>	<b>Responsibilities</b>	<b>Person</b>
Ketua	<ul style="list-style-type: none"><li>● Membahas dan merancang proyek</li><li>● Membuat program assembly</li><li>● Membuat README</li><li>● Membuat rangkaian proteus</li></ul>	Rivi Yasha Hafizhan
Anggota	<ul style="list-style-type: none"><li>● Membahas dan merancang proyek</li><li>● Membuat rangkaian fisik</li><li>● Membuat program assembly</li><li>● Membuat rangkaian proteus</li></ul>	Nabiel Harits Utomo
Anggota	<ul style="list-style-type: none"><li>● Membahas dan merancang proyek</li><li>● Membuat laporan proyek</li><li>● Membuat Flowchart</li><li>● Membuat Powerpoint</li></ul>	Grace Yunike Margaretha Sitorus
Anggota	<ul style="list-style-type: none"><li>● Membahas dan merancang proyek</li><li>● Membuat rangkaian proteus</li><li>● Membuat rangkaian fisik</li><li>● Membuat Powerpoint</li></ul>	Muhammad Arya Wiandra Utomo

Table 1. Roles and Responsibilities

## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC

Hardware yang digunakan dalam sistem TempShield terdiri dari sebuah mikrokontroler ATmega328P pada board Arduino Uno, yang terintegrasi dengan sensor suhu DHT11, modul motor driver L298N, dan kipas DC sebagai aktuator utama. Sistem ini dirancang untuk mendeteksi suhu ruangan secara otomatis dan mengatur kecepatan kipas dalam tiga mode berbeda berdasarkan hasil pembacaan suhu dari sensor. Output sistem berupa data suhu, kelembapan, dan mode kecepatan kipas ditampilkan melalui serial monitor menggunakan komunikasi UART.

Sensor DHT11 terhubung ke pin PD7 pada Arduino Uno dan bertugas mengukur suhu secara digital. Data yang diperoleh dari sensor ini kemudian digunakan oleh mikrokontroler untuk menentukan kecepatan putar kipas. Sistem ini tidak menggunakan LED indikator seperti pada sistem konvensional, namun menggunakan PWM yang dikirimkan dari pin PD6 menuju pin ENA pada modul L298N untuk mengontrol tingkat kecepatan kipas. Nilai duty cycle PWM diatur berdasarkan rentang suhu: suhu  $\leq 24^{\circ}\text{C}$  menghasilkan kecepatan low, suhu  $25 - 29^{\circ}\text{C}$  menghasilkan medium, dan suhu  $\geq 30^{\circ}\text{C}$  menghasilkan kecepatan high. Arah putaran motor ditentukan oleh pin IN1 dan IN2 pada driver motor, sementara suplai daya untuk kipas diberikan oleh sumber tegangan eksternal 12V agar motor dapat bekerja optimal.

Informasi suhu dan mode kipas saat ini dikirimkan secara berkala ke serial monitor menggunakan komunikasi UART melalui pin PD0 (RX) dan PD1 (TX). Sistem ini tidak menggunakan LCD I2C secara fisik, namun telah disimulasikan secara penuh menggunakan software Proteus untuk memverifikasi keakuratan pembacaan suhu, kestabilan sinyal PWM, serta pengiriman data serial secara real-time. Dengan perancangan hardware yang sederhana namun fungsional, sistem TempShield mampu merespons perubahan suhu dengan cepat, efisien, dan stabil dalam menjaga kestabilan suhu lingkungan.

## 2.2 SOFTWARE DEVELOPMENT

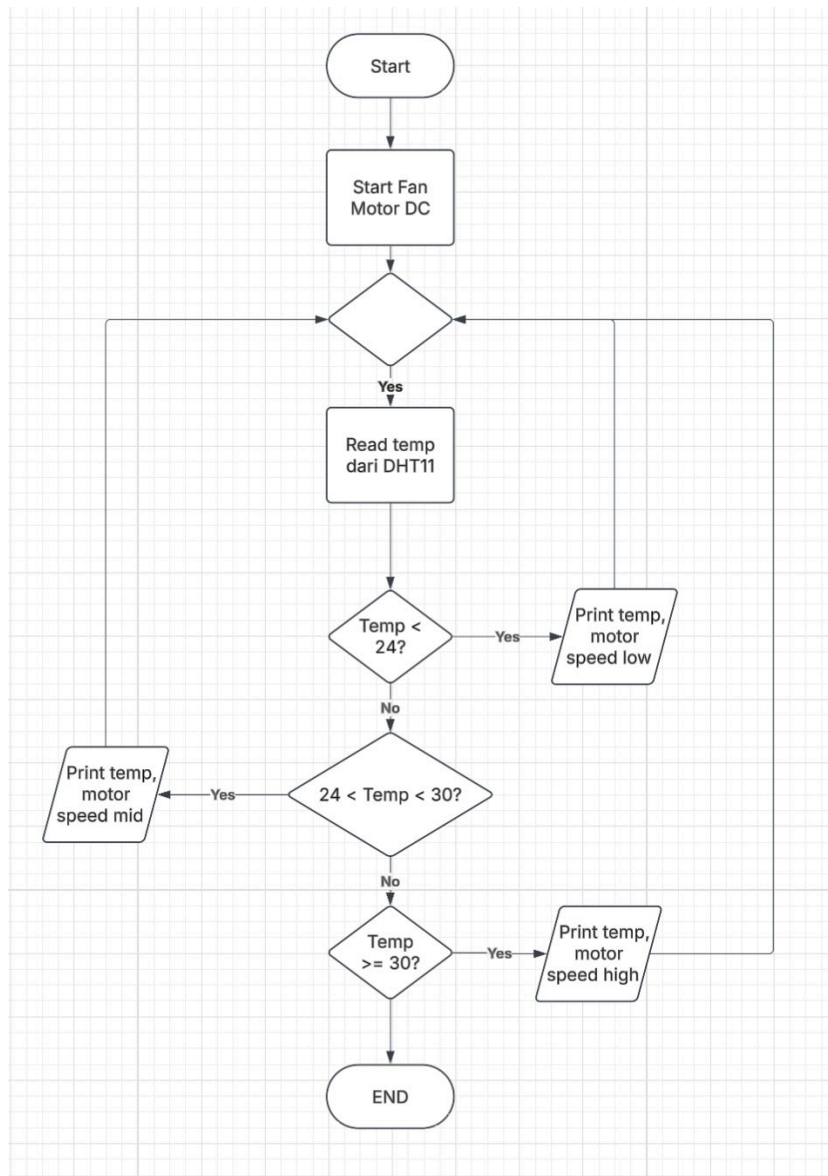
Software pada sistem TempShield dikembangkan menggunakan bahasa Assembly AVR pada platform Arduino IDE, dan ditujukan untuk mikrokontroler ATmega328P. Program ini dibuat untuk membaca data dari sensor suhu DHT11, memproses data tersebut, mengatur kecepatan kipas berdasarkan rentang suhu yang telah ditentukan, serta menampilkan data suhu, kelembapan, dan status kipas secara real-time melalui serial monitor menggunakan komunikasi UART.

Program diawali dengan mengatur stack pointer dan mengonfigurasi Timer0 dalam mode Fast PWM, yang berfungsi menghasilkan sinyal PWM untuk mengatur kecepatan kipas melalui modul motor driver L298N yang terhubung ke pin PD6. Setelah itu, bagian program `dht11_read` dijalankan untuk membaca data dari sensor DHT11 yang terhubung ke pin PD7. Nilai suhu kemudian disimpan dalam variabel `dht_temp` dan dibandingkan dengan dua batas suhu, yaitu 25°C dan 30°C. Berdasarkan hasil perbandingan tersebut, program akan memilih salah satu dari tiga bagian pengatur kipas: `set_slow`, `set_medium`, atau `set_fast`. Masing-masing bagian ini akan mengatur nilai PWM melalui register OCR0A, yang mengatur kecepatan kipas: rendah, sedang, atau tinggi.

Selain mengatur kecepatan kipas, program juga menampilkan nilai suhu, kelembapan, dan status kecepatan kipas melalui serial monitor. Tampilan ini dikendalikan oleh bagian program `send_temp_message` untuk menampilkan suhu dan kelembapan dalam bentuk angka, dan bagian `send_slow_message`, `send_medium_message`, atau `send_fast_message` untuk menampilkan mode kipas yang sedang aktif. Semua proses ini berjalan secara berulang dalam satu loop utama yang terus memperbarui data dan menyesuaikan kondisi kipas terhadap suhu lingkungan yang berubah-ubah.



Secara keseluruhan flowchart dari program akan terlihat seperti berikut:



## 2.3 HARDWARE AND SOFTWARE INTEGRATION

Untuk mengintegrasikan hardware dan software, kami memprogram mikrokontroler ATmega328P menggunakan bahasa Assembly. Sebagai langkah awal, kami menyusun rangkaian menggunakan software Proteus untuk mendapatkan gambaran awal mengenai skema dan cara kerja sistem sebelum diimplementasikan secara fisik pada breadboard. Komponen utama yang digunakan dalam sistem ini antara lain: sensor DHT11, modul motor driver L298N, kipas DC, serta komunikasi UART untuk menampilkan data melalui serial monitor.

Setelah skema selesai, kami memprogram mikrokontroler melalui Arduino IDE dengan kode Assembly yang telah ditulis. Program dirancang untuk membaca suhu dari sensor DHT11, lalu menentukan kecepatan kipas berdasarkan suhu tersebut. Jika suhu yang

terbaca berada di bawah atau sama dengan 24°C, maka kipas akan berputar dengan kecepatan rendah. Ketika suhu berada antara 25°C hingga 29°C, kipas akan berputar dengan kecepatan sedang. Sedangkan jika suhu mencapai 30°C atau lebih, kipas akan berputar dengan kecepatan tinggi. Kecepatan ini dikendalikan oleh sinyal PWM yang dikirim dari pin PD6 ke modul L298N, dan informasi suhu serta mode kecepatan kipas ditampilkan secara berkala melalui serial monitor.

Setelah kode berhasil dijalankan dalam simulasi Proteus, rangkaian kemudian diuji secara langsung pada perangkat fisik untuk memastikan bahwa sistem bekerja sesuai dengan logika yang telah dirancang. Pengujian dilakukan untuk memastikan bahwa data suhu yang dibaca sensor benar, sinyal PWM yang dihasilkan sesuai dengan kondisi suhu, dan kipas dapat berputar dalam tiga tingkat kecepatan yang berbeda. Hasil pengujian menunjukkan bahwa sistem mampu merespons perubahan suhu dengan baik dan menampilkan informasi secara real-time. Dengan integrasi yang selaras antara hardware dan software, sistem TempShield dapat berfungsi secara otomatis dan stabil sesuai tujuan awal perancangan.

## CHAPTER 3

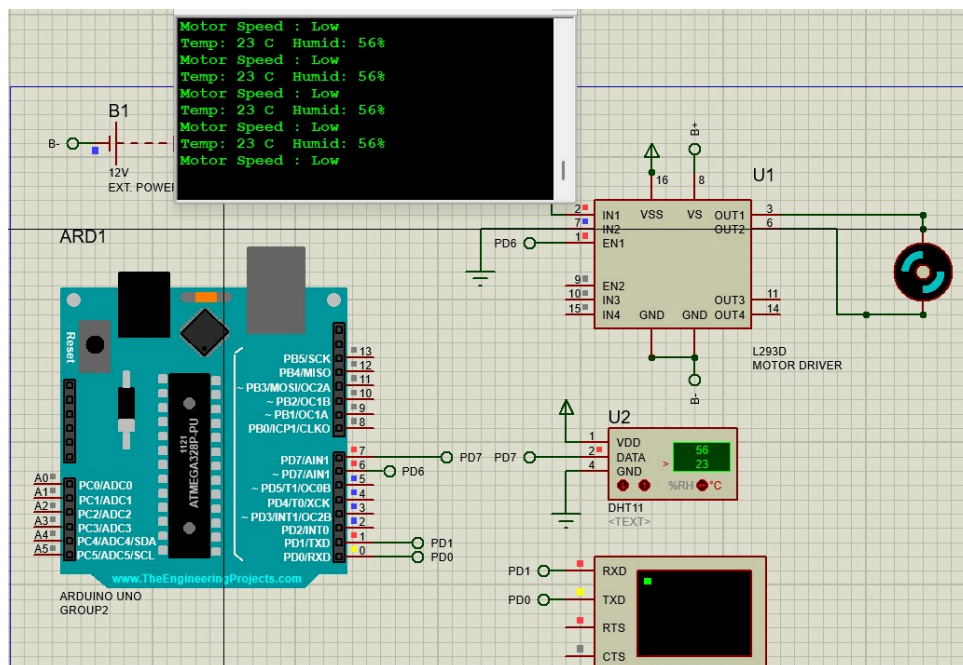
### TESTING AND EVALUATION

#### 3.1 TESTING

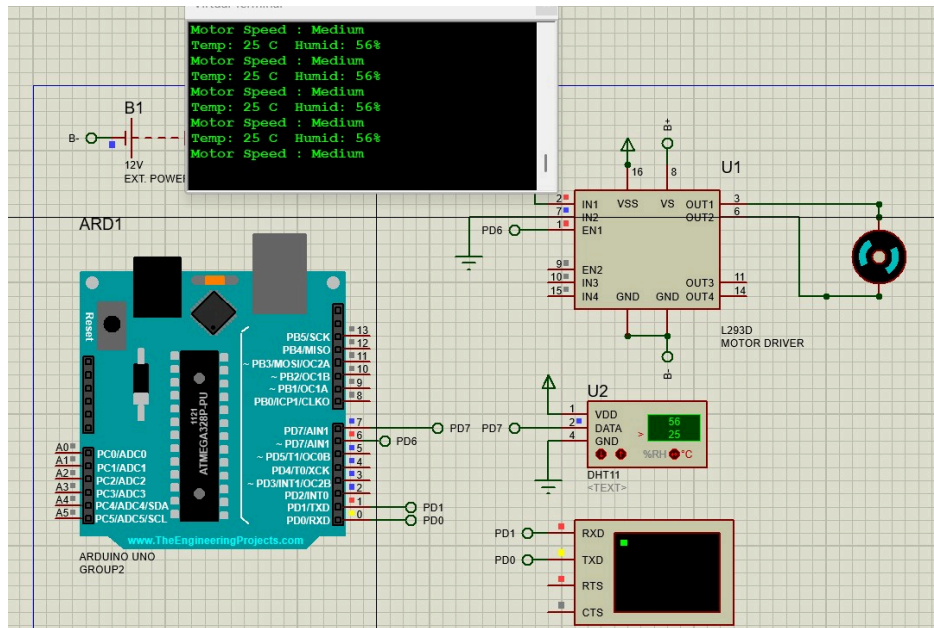
Pengujian dilakukan untuk memastikan bahwa sistem bekerja dengan baik sesuai dengan kriteria penerimaan yang telah ditentukan sebelumnya. Setiap komponen diuji secara individual sebelum dilakukan pengujian integrasi secara menyeluruh. Uji coba dilakukan dengan memberikan input suhu pada berbagai kondisi berbeda untuk memastikan respons sistem terhadap perubahan suhu dan kestabilan kinerja secara keseluruhan.

Sebelum merakit rangkaian secara fisik, pengujian awal dilakukan melalui simulasi Proteus. Dalam simulasi ini, sistem diuji dalam tiga kondisi suhu yang telah diprogram sebagai kategori utama, yaitu suhu rendah, sedang, dan tinggi. Pengujian ini bertujuan untuk mengevaluasi apakah sistem mampu mengatur kecepatan kipas secara otomatis sesuai logika program yang telah dirancang. Berikut adalah hasil dari pengujian simulasi:

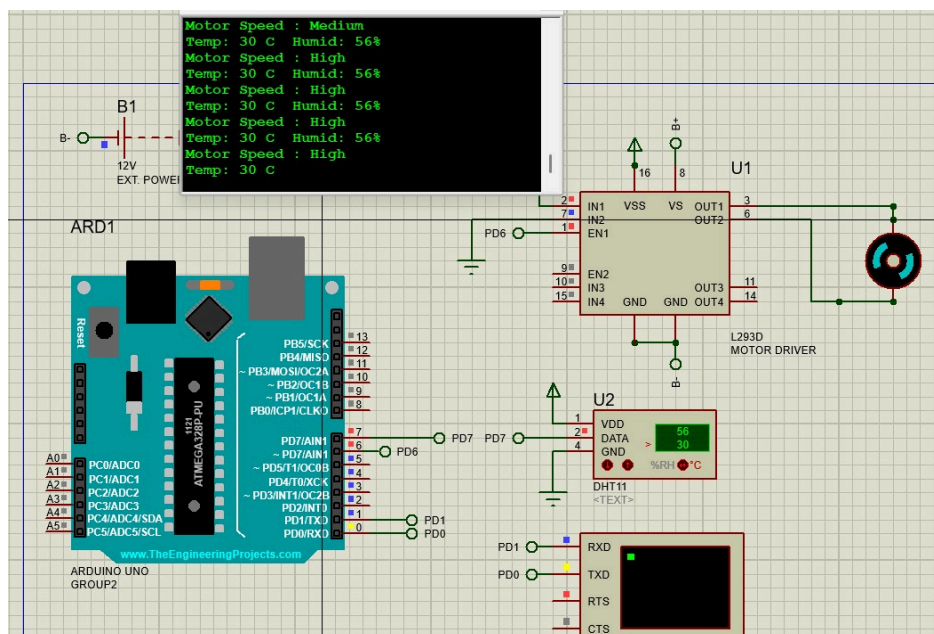
**Low Condition:** input suhu dari DHT11 berada dalam rentang  $\leq 24^{\circ}\text{C}$ , sistem mengaktifkan kipas dengan kecepatan rendah (Low Speed)



**Medium Condition:** suhu berada dalam rentang  $25^{\circ}\text{C} - 29^{\circ}\text{C}$ , sistem mengatur kipas ke kecepatan sedang (Medium Speed)



**High Condition:** suhu mencapai  $\geq 30^{\circ}\text{C}$ , sistem mengaktifkan kecepatan tinggi (High Speed) pada kipas.



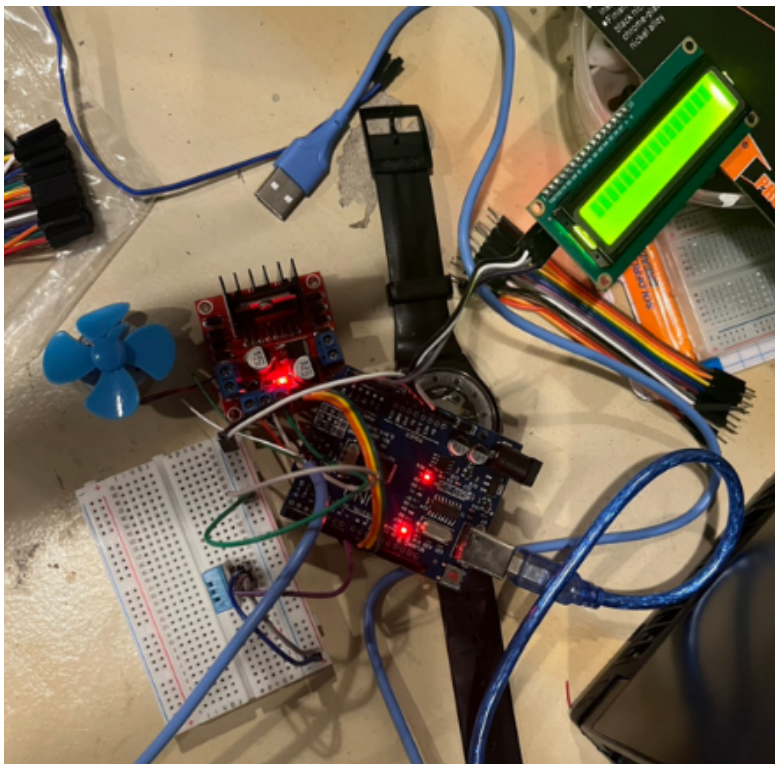
### 3.2 RESULT

Hasil pengujian menunjukkan bahwa sistem TempShield dapat membaca suhu dengan baik menggunakan sensor DHT11, dan menampilkannya secara real-time pada LCD 16x2. Selain nilai suhu, sistem juga menampilkan status mode kipas pada saat itu. Perubahan suhu

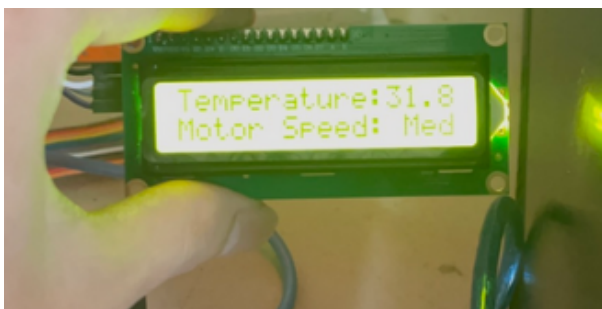
yang terjadi secara langsung dapat memengaruhi kecepatan kipas, sesuai dengan kategori suhu yang telah diprogram dalam sistem.

Motor kipas DC yang dikendalikan melalui modul L298N berfungsi dengan baik. Saat suhu berada dalam rentang rendah ( $\leq 24^{\circ}\text{C}$ ), kipas berputar dengan kecepatan rendah. Saat suhu meningkat ke  $25\text{--}29^{\circ}\text{C}$ , kecepatan kipas bertambah menjadi sedang. Dan ketika suhu mencapai  $30^{\circ}\text{C}$  atau lebih, sistem mengaktifkan kecepatan tinggi secara otomatis. Semua perubahan ini juga tercatat dan ditampilkan melalui serial monitor, menunjukkan sistem bekerja secara sinkron antara logika kontrol dan output aktual.

Tampak atas rangkaian fisik



Hasil pada layar LCD



Dari pengujian tersebut, dapat disimpulkan bahwa program Assembly yang digunakan telah memberikan hasil yang sesuai dengan acceptance criteria. Sistem mampu merespons perubahan suhu dengan cepat, mengatur kecepatan kipas dengan baik, dan menampilkan informasi yang lengkap melalui LCD. Tidak ditemukan kendala berarti dalam proses pengujian, dan semua komponen berfungsi sesuai dengan rancangan. Dengan ini, sistem dinyatakan berhasil dalam menjalankan fungsi utamanya sebagai pengendali suhu otomatis berbasis mikrokontroler.

### **3.3 EVALUATION**

Sistem TempShield telah menunjukkan kinerja yang baik selama pengujian, baik melalui simulasi maupun implementasi langsung. Sensor DHT11 mampu membaca suhu dan secara real-time, dan hasilnya ditampilkan secara akurat melalui LCD dan serial monitor. Program berhasil mengatur kecepatan kipas secara otomatis berdasarkan kategori suhu yang telah ditentukan, yaitu kecepatan rendah untuk suhu  $\leq 24^{\circ}\text{C}$ , kecepatan sedang untuk suhu  $25\text{--}29^{\circ}\text{C}$ , dan kecepatan tinggi untuk suhu  $\geq 30^{\circ}\text{C}$ .

Seluruh proses berjalan secara responsif dan konsisten, mulai dari pembacaan data sensor, pemrosesan logika suhu, hingga pengaturan kecepatan kipas dan tampilan informasi. Dengan menggunakan satu mikrokontroler ATmega328P, sistem mampu menangani semua fungsi dengan baik tanpa gangguan. Hal ini menunjukkan bahwa desain sistem sudah optimal dan dapat digunakan sebagai solusi sederhana dan efisien untuk pengaturan suhu otomatis di berbagai lingkungan aplikasi.

## **CHAPTER 4**

### **CONCLUSION**

Proyek TempShield berhasil dikembangkan sebagai sistem pengatur suhu otomatis berbasis mikrokontroler ATmega328P yang mampu merespons perubahan suhu lingkungan secara real-time. Sistem ini mengintegrasikan sensor DHT11 sebagai pembaca suhu dan kelembapan, motor driver L298N sebagai pengendali kecepatan kipas, serta LCD 16x2 untuk menampilkan informasi suhu dan status mode kipas. Logika pengendalian suhu yang dirancang berhasil dijalankan sesuai dengan kategori suhu yang telah ditentukan.

Seluruh komponen telah diuji melalui simulasi dan implementasi langsung, dan hasil pengujian menunjukkan bahwa sistem mampu bekerja secara otomatis, responsif, dan akurat. Dengan desain yang efisien serta penggunaan kode Assembly yang ringan, sistem ini menunjukkan potensi untuk diterapkan dalam berbagai skenario seperti ruang penyimpanan alat elektronik, inkubator bayi, dan lingkungan lain yang memerlukan kontrol suhu sederhana namun handal. TempShield telah memenuhi kriteria fungsional utama dan siap dikembangkan lebih lanjut sesuai kebutuhan aplikasi di dunia nyata.

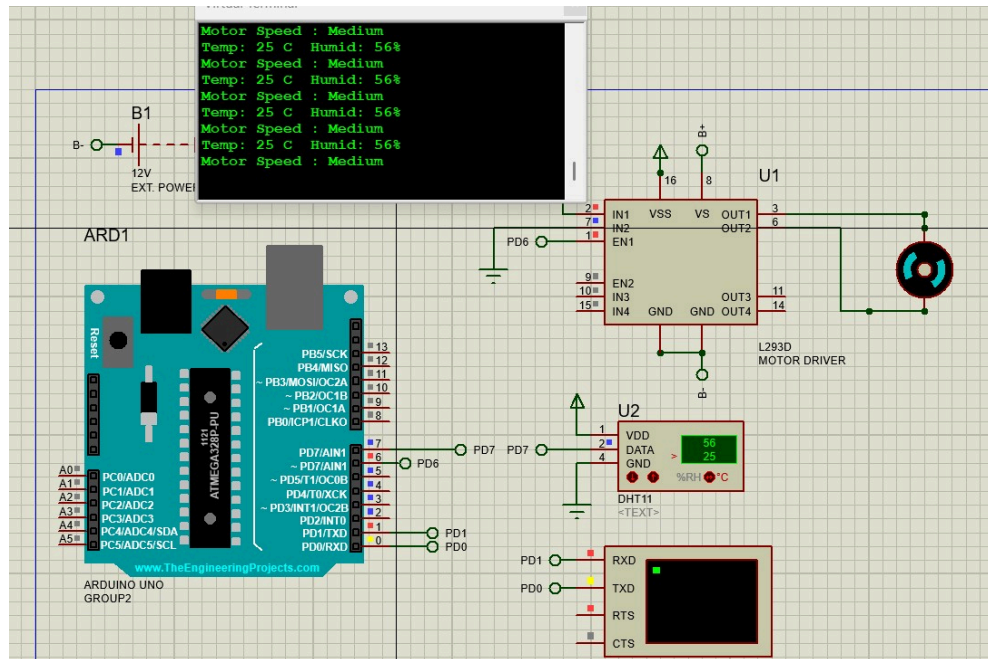
## REFERENCES

- [1] RoboTechZone, “Temperature Based Fan Speed Controller Using Arduino UNO || ARDUINO PROJECTS,” YouTube, <https://youtu.be/Z1vYp8NE9LI?si=QhjHiVHPrOqVRvOA> (accessed May 16, 2025).
- [2] How To Electronics, “Temperature Based Fan Speed Control & Monitoring With Arduino,” YouTube, <https://youtu.be/OCsmMD4wtNQ?si=64THqU05G0kqkt7P> (accessed May 16, 2025).
- [3] SME Dehradun, “DC Motor Speed control through PWM signal using Arduino- Proteus Simulation,” YouTube, [https://youtu.be/Besg5b\\_BfkM?feature=shared](https://youtu.be/Besg5b_BfkM?feature=shared) (accessed May 16, 2025).



## APPENDICES

## Appendix A: Project Schematic



## Appendix B: Documentation

