



TUGAS AKHIR (EE43430)

**RANCANG BANGUN SISTEM PEMANTAUAN GAS BERBAHAYA
BERBASIS ESP32 DENGAN PENGGUNAAN WSN PADA
*POWERPACK***

Mochamad Zaka Ananta Agva Firdaus
NRP. 0422040041

Dosen Pembimbing

1. Alief Nur Aisyi Maulidhia, S.T., M.T.
2. Adianto, S.T., M.T.

PROGRAM STUDI D4 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR (EE43430)

**RANCANG BANGUN SISTEM PEMANTAUAN GAS BERBAHAYA
BERBASIS ESP32 DENGAN PENGGUNAAN WSN PADA
POWERPACK**

Mochamad Zaka Ananta Agva Firdaus
NRP. 0422040041

Dosen Pembimbing

1. Alief Nur Aisyi Maulidhia, S.T., M.T.
2. Adianto, S.T., M.T.

**PROGRAM STUDI D4 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

RANCANG BANGUN SISTEM PEMANTAUAN GAS BERBAHAYA BERBASIS ESP32 DENGAN PENGGUNAAN WSN PADA *POWERPACK*

Disusun Oleh:

Mochamad Zaka Ananta Agva Firdaus
0422040041

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi D4 Teknik Kelistrikan Kapal
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim Penguji Tugas Akhir Tanggal Ujian : 30 Juli 2026
Periode Wisuda : Oktober 2026

Menyetujui,

Dosen Penguji

1. Isa Rachman, S.T., M.T.

NIDN

(0016088003)

Tanda Tangan

(.....)

2. Alief Nur Aisyi Maulidhia, S.T., M.T.

(8060775676230173)

(.....)

3. Anggarjuna Puncak Pujiputra, S.T., M.T.

(6535772673130252)

(.....)

4. Aslikha Amalia, S.ST., M.T.

(0546770671230402)

(.....)

Dosen Pembimbing

1. Alief Nur Aisyi Maulidhia, S.T., M.T.

NIDN

(8060775676230173)

Tanda Tangan

(.....)

2. Adianto, S.T., M.T.

(0002077704)

(.....)

Menyetujui,
Ketua Jurusan



Isa Rachman, S.T., M.T.
NIP. 198008162008121001

Mengetahui,
Koordinator Program Studi

Dr. Yuning Widiarti, S.T., M.T.
NIP. 198005162006042001

(Halaman ini sengaja dikosongkan)

 PPNS PROSES PENYERAPAN NUSANTARA	PERNYATAAN BEBAS PLAGIAT	No. : F.WD I. 021 Date : 3 Nopember 2015 Rev. : 01 Page : 1 dari 1
--	-------------------------------------	--

Yang bertandatangan dibawah ini :

Nama : Mochamad Zaka Ananta Agva Firdaus

NRP. 0422040041

Jurusan/Prodi : Teknik Kelistrikan Kapal

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

Rancang Bangun Sistem Pemantauan Gas Berbahaya Berbasis ESP32 Dengan Penggunaan WSN Pada Powerpack

Adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain.

Apabila dikemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut, maka saya bersedia menerima **sanksi** sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 23 Juli 2026

Yang membuat pernyataan,



(Moch. Zaka Ananta A. F)
NRP. 0422040041

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah senantiasa melimpahkan rahmat, karunia, kekuatan, serta petunjuk-Nya kepada penulis. Penulis juga tidak lupa mengucapkan shalawat serta salam yang senantiasa tercurahkan kepada Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Sistem Pemantauan Gas Berbahaya Berbasis ESP32 Dengan Penggunaan WSN Pada *Powerpack*” dengan baik dan lancar sebagai salah satu syarat kelulusan dalam menyelesaikan program studi Diploma 4 Teknik Kelistrikan kapal di Politeknik Perkapalan Negeri Surabaya.

Penulis menyadari bahwa selama proses penyusunan tugas akhir, penulis mendapatkan banyak bantuan dari banyak pihak yang senantiasa membimbing penulis baik secara moral, pengetahuan, dan materi. Penulis ingin mengucapkan rasa syukur dan terimakasih sebesar-besarnya kepada:

1. Kedua orang tua, Papa dan Mama beserta keluarga besar penulis yang senantiasa selalu memberikan kasih sayang, do'a, dukungan moral dan materi yang sangat berharga bagi penulis.
2. Bapak Rachmad Tri Soelistijono, S.T., M.T., selaku Direktur Politeknik Perkapalan Negeri Surabaya.
3. Bapak Isa Rachman, S.T., M.T., selaku Ketua Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya.
4. Ibu Dr. Yuning Widiarti, S.T., M.T., selaku Ketua Prodi Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya.
5. Ibu Alief Nur Aisyi Maulidhia, S.T., M.T., selaku Dosen Pembimbing 1 yang telah membimbing dan mengarahkan penulis dalam menyusun Tugas Akhir. Terimakasih banyak atas waktu, kesabaran, dan ilmu yang telah diamalkan kepada penulis.
6. Bapak Adianto, S.T., M.T., selaku Dosen Pembimbing 2. Terimakasih atas saran dan masukan yang diberikan selama membimbing penulis. Terimakasih atas waktu, kesabaran, dan juga ilmu yang diberikan kepada penulis.

7. Bapak Ibu Dosen Pengajar Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya yang dengan ikhlas membekali ilmu dan pengalaman yang sangat bermanfaat kepada penulis selama masa perkuliahan.
8. Seluruh Pihak dari PT Meratus Wahana Karya yang telah memberikan kesempatan dan fasilitas kepada penulis untuk melaksanakan pengambilan data tugas akhir, khususnya atas ketersediaan wadah serta izin penggunaan unit powerpack sebagai objek penelitian selama proses pengerjaan tugas akhir ini berlangsung. Dukungan dan kepercayaan yang diberikan oleh pihak perusahaan sangat berarti bagi kelancaran dan keberhasilan penelitian ini.
9. Terima kasih yang tulus penulis sampaikan kepada Shafa Salisa Nabila yang selalu hadir dengan sabar dan setia menemani setiap proses panjang pengerjaan tugas akhir ini, memberikan semangat di saat penulis merasa lelah, serta selalu percaya bahwa penulis mampu menyelesaikannya. Kehadiranmu bukan sekadar menemani, tetapi menjadi bagian dari alasan mengapa tugas akhir ini akhirnya dapat terselesaikan.
10. Seluruh rekan Setrum Pride yang telah menjadi teman setia penulis dari awal hingga akhir masa perkuliahan, selalu hadir berbagi suka maupun duka dalam setiap perjalanan yang dilalui bersama.
11. Sahabat-sahabat penulis sejak SMA yang meskipun kini menempuh jalan masing-masing, namun tidak pernah berhenti memberikan dukungan dan semangat kepada penulis hingga tugas akhir ini dapat terselesaikan.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun untuk perbaikan ke depan. Semoga Tugas Akhir ini dapat memberikan manfaat serta menjadi referensi bagi penelitian selanjutnya. Penulis juga memohon maaf apabila terdapat kesalahan dalam penyusunan karya ini.

Surabaya, 30 Agustus 2026
Penulis

Mochamad Zaka Ananta Agva Firdaus
NRP. 0422040041

Rancang Bangun Sistem Pemantauan Gas Berbahaya Berbasis ESP32 Dengan Penggunaan WSN Pada Powerpack

Mochamad Zaka Ananta Agva Firdaus

ABSTRAK

Salah satu masalah serius yang dihadapi adalah kurangnya penerapan Kesehatan dan Keselamatan Kerja (K3) di ruang *powerpack* kapal, di mana kesehatan awak kapal sangat rentan terhadap paparan gas berbahaya seperti Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂). Kedua gas ini tidak berwarna, tidak berbau, dan beracun, sehingga sulit dideteksi tanpa alat khusus. Penelitian ini bertujuan merancang sistem pemantau gas CO dan NO₂ di ruang *powerpack* kapal menggunakan ESP32, dengan sensor MQ-135, MQ-7, dan SHT20 untuk suhu. Sistem menerapkan protokol MQTT untuk pemantauan *real-time*, serta *Wireless Sensor Network* (WSN) yang menghubungkan antar node sensor melalui protokol ESP-NOW. Data sensor diolah menggunakan *rule-based algorithm* untuk mengklasifikasikan status kondisi udara (*Normal*, *Warning*, *Danger*) serta mengendalikan kecepatan *exhaust fan* secara otomatis berdasarkan konsentrasi gas terdeteksi. Hasil pengujian menunjukkan *rule-based algorithm* memiliki akurasi perhitungan tinggi, dengan error hanya 0,025% berdasarkan perbandingan MATLAB, Arduino IDE, dan perhitungan manual. Pengujian pada kondisi gas aktual di panel Master dan tiga panel Slave menunjukkan sistem mampu mengklasifikasikan status kondisi udara secara konsisten sesuai NAB Permenaker RI Nomor 5 Tahun 2018, dengan tingkat kesesuaian klasifikasi terhadap rancangan sistem mencapai 100%. Sistem ini layak diterapkan sebagai solusi peringatan dini demi keselamatan kerja di ruang *powerpack* kapal.

Kata kunci: ESP32, WSN, gas CO, gas NO₂, *rule-based algorithm*, *powerpack* kapal, MQTT

(Halaman ini sengaja dikosongkan)

Design of a Hazardous Gas Monitoring System Based on ESP32 Using WSN on Powerpack

Mochamad Zaka Ananta Agva Firdaus

ABSTRACT

One of the serious issues in ship powerpack rooms is the lack of Occupational Health and Safety (K3) implementation, leaving crew vulnerable to hazardous gases like Carbon Monoxide (CO) and Nitrogen Dioxide (NO₂), which are colorless, odorless, toxic, and difficult to detect without specialized equipment. This research designs a CO and NO₂ monitoring system for ship powerpack rooms using an ESP32 microcontroller, MQ-135 and MQ-7 gas sensors, and an SHT20 temperature sensor. The system applies MQTT for real-time monitoring and a Wireless Sensor Network (WSN) connecting sensor nodes via ESP-NOW. Data is processed using a rule-based algorithm to classify air conditions (Normal, Warning, Danger) and automatically control exhaust fan speed based on detected gas concentration. Test results show the algorithm achieves high accuracy, with only 0.025% error compared to MATLAB, Arduino IDE, and manual calculations. Testing under actual gas conditions on the Master panel and three Slave panels shows the system consistently classifies air conditions per the Threshold Limit Value (TLV) set by Indonesian Minister of Manpower Regulation No. 5 of 2018, achieving 100% classification consistency with the design. This system is feasible as an early-warning solution for occupational safety in ship powerpack rooms.

Keyword: *ESP32, WSN, CO gas, NO₂ gas, rule-based algorithm, ship powerpack, MQTT*

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
PERNYATAAN BEBAS PLAGIAT	v
KATA PENGANTAR.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
DAFTAR ISI.....	xiii
DAFTAR TABEL	xvii
DAFTAR GAMBAR.....	xix
DAFTAR NOTASI.....	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Penelitian	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Kajian Penelitian Terdahulu	5
2.2 Dasar Teori	6
2.2.1 <i>Powerpack</i>	6
2.2.2 Sistem Pemantauan Gas Berbahaya	8
2.2.3 <i>Wireless Sensor Network (WSN)</i>	9
2.2.4 Protokol MQTT (<i>Message Queuing Telemetry Transport</i>)	10
2.3 Kajian Pustaka Objek Perangkat Keras & Lunak.....	12

2.3.1	ESP32 Mikrokontroler	12
2.3.2	Sensor Gas MQ-135	13
2.3.3	Sensor MQ-7	15
2.3.4	Sensor SHT20	17
2.3.5	<i>Exhaust fan</i> Maspion CEF-25 10 inch	18
2.3.6	LCD 20x4.....	20
2.3.7	<i>Buzzer</i> SFM-27	21
2.3.8	Modul LSA-002 AC Dimmer	23
2.3.9	Raspberry pi 3	24
2.3.10	Battery Lithium 18650 3.7V	26
2.3.11	Power Supply 12V 10A	27
2.3.12	Modul MP1584 <i>Buck Converter</i>	30
2.3.13	Modul LM2596 Step down.....	31
2.3.14	Modul GSM SIM800L.....	33
2.3.14	Software Arduino IDE	34
2.3.15	Phyton	35
2.3.16	Software Visual Studio Code (VS Code)	37
BAB 3 METODE PENELITIAN		39
3.1	Konsep Penelitian	39
3.2	Tahapan Penelitian.....	39
3.2.1	Identifikasi Masalah.....	40
3.2.2	Diagram Blok Sistem.....	40
3.2.3	Flowchart Sistem	41
3.2.4	Perancangan Hardware	44
3.2.5	Perancangan <i>Software</i>	47
3.2.6	Sinkronisasi Hardware dan <i>Software</i>	47

3.2.7 Pengujian Parsial	47
3.2.8 Denah Pengujian.....	48
3.2.9 Pengambilan Data dan Analisis Data	48
3.2.10 Penyusunan Buku Tugas Akhir	48
3.3 Rancangan Perangkat Keras/Lunak.....	49
3.3.1 Analisis Kebutuhan Sistem	49
3.3.2 Perhitungan Kebutuhan Sistem	50
3.3.3 Wiring Diagram Sistem.....	54
3.3.4 Wiring diagram Supply Power ESP32	55
3.3.5 Wiring diagram sensor MQ-135.....	55
3.3.6 Wiring diagram sensor MQ-7.....	56
3.3.7 Wiring diagram Sensor SHT20	57
3.3.8 Wiring diagram LCD 20x4.....	57
3.3.9 Wiring diagram <i>buzzer</i>	58
3.3.10 Wiring diagram Modul Dimmer LSA-002.....	59
3.3.11 Wiring diagram <i>Exhaust Fan</i>	59
3.3.12 Rancangan Desain PCB.....	60
3.3.13 Rancangan Desain Panel	61
3.3.14 Rancangan Software.....	61
3.3.15 Rancangan Rule Base Algoritm	62
BAB 4 HASIL DAN PEMBAHASAN	65
4.1 Implementasi Mekanik	65
4.1.1 Desain dan Realisasi <i>Box Panel</i>	65
4.2 Pengujian Parsial	66
4.2.1 Pengujian Parsial PCB.....	67
4.2.2 Pengujian Parsial Sensor MQ-7	68

4.2.3 Pengujian Parsial Sensor MQ-135	72
4.2.4 Pengujian Parsial Sensor SHT20	76
4.2.5 Pengujian Parsial LCD 20x4	79
4.2.6 Pengujian Parsial <i>Buzzer</i>	80
4.2.7 Pengujian Komunikasi antar ESP32	81
4.2.8 Pengujian <i>Web Interface</i>	82
4.3 Pengujian Sistem menggunakan Rule Base Algoritm	84
4.3.1 Pengujian <i>Rule-Base Algorithm</i> menggunakan toolbox matlab	84
4.3.2 Pengujian <i>Rule-Base Algorithm</i> menggunakan Arduino IDE.....	85
4.3.3 Pengujian <i>Rule-Base Algorithm</i> menggunakan perhitungan manual.....	88
4.3.4 Pengujian Sistem terhadap Kondisi Aktual	90
4.3.4.1 Pengujian Panel Master	90
4.3.4.2 Pengujian Panel Slave 1	92
4.3.4.3 Pengujian Panel Slave 2	94
4.3.4.4 Pengujian Panel Slave 3	96
BAB 5 KESIMPULAN DAN SARAN	99
5.1 Kesimpulan	99
5.2 Saran	100
DAFTAR PUSTAKA	101
LAMPIRAN	103

DAFTAR TABEL

Tabel 2.1 Spesifikasi Generator dan Mesin Diesel	8
Tabel 2.2 Nilai Ambang Batas Konsentrasi Gas.....	8
Tabel 2.3 Hasil Pengujian Gas	9
Tabel 2.4 Spesifikasi Mikrokontroler ESP32.....	13
Tabel 2.5 Spesifikasi Sensor MQ-135	15
Tabel 2.6 Spesifikasi Sensor MQ-7	17
Tabel 2.7 Spesifikasi Sensor SHT20.....	18
Tabel 2.8 Spesifikasi Exhaust fan	20
Tabel 2.9 Spesifikasi LCD 20x4	21
Tabel 2.10 Spesifikasi Buzzer.....	23
Tabel 2.11 Spesifikasi Modul LSA-002 AC Dimmer.....	24
Tabel 2.12 Spesifikasi Raspberry pi 3.....	25
Tabel 2.13 Spesifikasi Battery Lithium 18650	27
Tabel 2.14 Spesifikasi Power Supply 12V 10A.....	30
Tabel 2.15 Spesifikasi Modul MP1584 Buck Converter	31
Tabel 2.16 Spesifikasi Modul LM2596 Step down	32
Tabel 2.17 Spesifikasi Modul GSM SIM800L	34
Tabel 2.18 Spesifikasi Software Arduino IDE.....	35
Tabel 3.1 Kebutuhan Sistem	49
Tabel 4.1 Pengujian Sensor MQ-7 nomor 1	69
Tabel 4.2 Pengujian Sensor MQ-7 nomor 2	70
Tabel 4.3 Pengujian Sensor MQ-7 nomor 3	70
Tabel 4.4 Pengujian Sensor MQ-7 nomor 4	71
Tabel 4.5 Pengujian Sensor MQ-135 nomor 1	73
Tabel 4.6 Pengujian Sensor MQ-135 nomor 2	74
Tabel 4.7 Pengujian Sensor MQ-135 nomor 3	75
Tabel 4.8 Pengujian Sensor MQ-135 nomor 4	75
Tabel 4.9 Pengujian Parsial sensor SHT20 nomor 1	77
Tabel 4.10 Pengujian Parsial sensor SHT20 nomor 2.....	77
Tabel 4.11 Pengujian Parsial sensor SHT20 nomor 3.....	78
Tabel 4.12 Pengujian Parsial sensor SHT20 nomor 4.....	78

Tabel 4.13 Pengujian Parsial Buzzer	80
Tabel 4.14 Pengujian dengan Arduino IDE.....	86
Tabel 4.15 Perbandingan pengujian menggunakan toolBox matlab dengan Arduino IDE	87
Tabel 4.16 Evaluasi Rule Base Perhitungan Manual Data 1	88
Tabel 4.17 Evaluasi Rule Base Perhitungan Manual Data 9	88
Tabel 4.18 Hasil Inferensi Rule-Base Algoritm dari ketiga metode percobaan	89
Tabel 4.19 Data Suhu, Gas CO dan NO ₂ Powerpack Panel Master	91
Tabel 4.20 Data Suhu, Gas CO dan NO ₂ Powerpack Panel Slave 1	93
Tabel 4.21 Data Suhu, Gas CO dan NO ₂ Powerpack Panel Slave 2	95
Tabel 4.22 Data Suhu, Gas CO dan NO ₂ Powerpack Panel Slave 3	97

DAFTAR GAMBAR

Gambar 2.1 <i>Powerpack</i>	7
Gambar 2.2 sistem Wireless Sensor Network.....	10
Gambar 2.3 Protokol MQTT.....	11
Gambar 2.4 ESP32 Mikrokontroler	12
Gambar 2.5 Sensor Gas MQ-135	15
Gambar 2.6 Sensor Gas MQ-7	16
Gambar 2.7 Sensor Gas SHT20	18
Gambar 2.8 <i>Exhaust fan</i> Maspion CEF-25 10 inch	19
Gambar 2.9 LCD 20x4.....	21
Gambar 2.10 <i>Buzzer</i> SFM-27.....	22
Gambar 2.11 Modul LSA-002 AC Dimmer	24
Gambar 2.12 Raspberry pi 3	25
Gambar 2.13 Battery Lithium 18650	27
Gambar 2.14 Power Supply 12V 10A.....	29
Gambar 2.15 Modul MP1584 <i>Buck Converter</i>	31
Gambar 2.16 Modul LM2596 Step down	32
Gambar 2.17 Modul GSM SIM800L	33
Gambar 2.18 Software Arduino IDE	35
Gambar 2.19 Bahasa Python	36
Gambar 2.20 Software VS Code.....	37
Gambar 3.1 Tahapan Penelitian.....	39
Gambar 3.2 Diagram Blok Sistem	41
Gambar 3.3 Flowchart Node Sensor	42
Gambar 3.4 Flowchart Node sink	43
Gambar 3.5 Perancangan Hardware.....	44
Gambar 3.6 Wiring Diagram Sistem.....	54
Gambar 3.7 Wiring Supply Power ESP32	55
Gambar 3.8 Wiring diagram Sensor MQ-135.....	56
Gambar 3.9 Wiring diagram Sensor MQ-7	56
Gambar 3.10 Wiring diagram Sensor SHT20	57
Gambar 3.11 Wiring Diagram LCD	58

Gambar 3.12 Wiring Diagram <i>Buzzer</i>	58
Gambar 3.13 Wiring Diagram Modul Dimmer LSA-002	59
Gambar 3.14 Wiring Diagram <i>Exhaust Fan</i>	60
Gambar 3.15 Design PCB	60
Gambar 3.16 Desain Panel Node Sensor	61
Gambar 3.17 Rancangan Software	61
Gambar 4.1 Realisasi <i>Box</i> Panel Master.....	65
Gambar 4.2 Realisasi Box Panel Slave	66
Gambar 4.3 Pengujian Parsial Input PCB	67
Gambar 4.4 Pengujian Parsial Output Buck Converter	67
Gambar 4.5 Pengujian Parsial Sensor MQ-7	68
Gambar 4.6 Pengujian Parsial Sensor MQ-135	72
Gambar 4.7 Pengujian Parsial SHT20	76
Gambar 4.8 Pengujian Parsial LCD 20x4	79
Gambar 4.9 Pengujian Parsial Buzzer	80
Gambar 4.10 Pengujian Komunikasi antar ESP32	81
Gambar 4.11 Dashboard Web Monitoring	82
Gambar 4.12 Tampilan Detail Data Web Monitoring	83
Gambar 4.13 Data Logger Web Monitoring	83
Gambar 4.14 Pengujian dengan Toolbox Matlab	84
Gambar 4.15 Pengujian Rule Base Algoritma menggunakan Arduino IDE	85
Gambar 4.16 Grafik Fluktuasi Suhu, Gas CO dan NO ₂ Panel Master dalam 1 jam pengujian	92
Gambar 4.17 Grafik Fluktuasi Suhu, Gas CO dan NO ₂ Panel Slave 1 dalam 1 jam pengujian	94
Gambar 4.18 Grafik Fluktuasi Suhu, Gas CO dan NO ₂ Panel Slave 2 dalam 1 jam pengujian	96
Gambar 4.19 Grafik Fluktuasi Suhu, Gas CO dan NO ₂ Panel Slave 3 dalam 1 jam pengujian	98

DAFTAR NOTASI

V	= Tegangan
I	= Arus
Watt	= Daya
Rs	= Nilai resistansi sensor MQ-135
Ro	= Tahanan sensor pada udara bersih
RL	= Nilai Tahanan (10K)
VRL	= Nilai Tahanan pada RL
PPM	= Parts Per Millions
P	= daya yang dibutuhkan (Watt)
Q	= debit udara (m^3/s)
ΔP	= tekanan statis (static pressure/Pascal, Pa)
η	= efisiensi mekanis kipas (dalam desimal, $0 < \eta \leq 1$)
ACH	= Air Circulation Per Hour

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Powerpack atau yang dikenal sebagai generator bantu memiliki peran yang sangat vital bagi kelancaran operasional kapal. Di dalam ruang kontainer *powerpack*, gas berbahaya hasil pembakaran mesin diesel seperti Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂) berpotensi terakumulasi dalam kadar yang membahayakan. Gas-gas tersebut bersifat tidak berbau dan tidak terlihat, sehingga sangat sulit dideteksi tanpa adanya sistem pemantauan khusus, meskipun dampaknya terhadap kesehatan awak kapal sangat serius. Peraturan Menteri Ketenagakerjaan Republik Indonesia Nomor 5 Tahun 2018 yang berlandaskan Undang-Undang Nomor 1 Tahun 1970 tentang Keselamatan Kerja telah mengatur ketentuan keselamatan dan kesehatan kerja yang mencakup berbagai aktivitas pekerjaan, termasuk penggunaan bahan kimia dan pekerjaan di lingkungan tertutup. Peraturan tersebut menetapkan bahwa batas paparan gas Karbon Monoksida adalah 25 PPM dan Nitrogen Dioksida adalah 5 PPM selama 8 jam kerja per hari, serta mewajibkan penggunaan Alat Pelindung Diri (APD) apabila kadar gas melampaui ambang batas yang telah ditentukan (Kementerian Ketenagakerjaan Republik Indonesia, 2018). Berdasarkan kondisi tersebut, diperlukan sebuah sistem pemantauan gas berbahaya secara *real-time* yang mampu mendeteksi dan memberikan peringatan dini terhadap akumulasi gas CO dan NO₂ di dalam ruang kontainer *powerpack*, guna melindungi keselamatan dan kesehatan awak kapal secara berkelanjutan.

Suhu di ruang *powerpack* yang tidak terkontrol juga dapat mempengaruhi kinerja mesin dan perangkat lainnya. Peningkatan suhu yang tinggi dapat menyebabkan kerusakan pada sistem kelistrikan dan mempercepat keausan perangkat sehingga diperlukan Sistem pemantauan suhu yang baik. Untuk mengurangi resiko Oleh karena itu, pengawasan suhu menjadi hal yang penting untuk menjaga kinerja sistem dan kenyamanan awak kapal.

Tujuan dari penelitian ini adalah untuk merancang sistem pemantauan berbasis ESP32 yang terhubung dengan Wireless Sensor Network (WSN) untuk

mendeteksi gas CO, NO₂, dan suhu ruang secara *real-time*. Sensor yang digunakan adalah MQ-135 untuk mendeteksi NO₂, MQ-7 untuk CO, dan SHT20 untuk *monitoring* suhu di sekitar ruang *powerpack*. Sistem akan menghubungkan sensor dan mengirimkan data melalui protokol MQTT. Protokol ini memungkinkan komunikasi data nirkabel yang efektif antara sensor dan pusat pemantauan.

Sistem ini diharapkan dapat meningkatkan keselamatan awak kapal dengan memberikan peringatan dini jika terjadi peningkatan kadar gas berbahaya atau suhu yang tidak terkendali. Penelitian ini merupakan bagian dari kemajuan dalam teknologi pemantauan berbasis *Internet of Things (IoT)* yang bertujuan untuk meningkatkan efisiensi operasional dan keselamatan kapal, dan menemukan bahwa sistem ini dapat mempertahankan kinerja *powerpack* dengan mengurangi risiko kerusakan perangkat yang disebabkan oleh paparan gas berbahaya dan suhu ekstrim.

1.2 Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah penelitian ini adalah:

1. Bagaimana cara menghubungkan sensor MQ-135, MQ-7, dan SHT20 ke mikrokontroler ESP32 melalui jaringan sensor nirkabel (WSN) pada generator kapal atau *powerpack* untuk memantau suhu, CO, dan NO₂?
2. Bagaimana sebenarnya protokol MQTT bekerja dalam memproses dan mengirimkan data antar sensor secara *real-time*?
3. Bagaimana sistem ini *memonitoring* suhu di ruang *powerpack* dan memberi peringatan awal serta mengatur sirkulasi udara untuk menghindari paparan gas CO dan NO₂ bagi awak kapal?

1.3 Batasan Penelitian

Agar penelitian lebih terarah, batasan penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya membahas sistem pemantauan gas CO, NO₂, dan suhu pada *powerpack* atau generator bantu kapal menggunakan sensor MQ-135, MQ-7, dan SHT20.
2. Protokol MQTT mengirimkan data dari Wireless Sensor Network (WSN) menuju ke mikrokontroler ESP32.

3. Sensor suhu SHT20 hanya berfungsi untuk memantau suhu di sekitar ruang *powerpack* atau generator kapal, tanpa berperan dalam pengaturan otomatis sistem.
4. Fokus utama penelitian ini adalah pada ruang *powerpack*, yang memiliki ventilasi terbatas dan lebih rentan terhadap paparan gas berbahaya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mendesain dan membuat sistem yang dapat memantau gas Karbon Monoksida (CO), Nitrogen Dioksida (NO₂), dan suhu yang berbasis ESP32 dan terintegrasi dengan jaringan sensor nirkabel (WSN) pada *powerpack* atau generator bantu kapal.
2. Menerapkan protokol MQTT untuk komunikasi data antar titik sensor dan pengolahan data secara *real-time*.
3. Menguji efektivitas sistem dalam memberikan peringatan dini serta mengendalikan kualitas udara di lingkungan kapal.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian, diharapkan penulis pada Tugas Akhir ini dapat memberikan manfaat sebagai berikut:

1. Penelitian ini bertujuan memberi wawasan tentang desain sistem berbasis *Internet of Things* (IoT) dan MQTT. Penulis berharap memahami penerapan teknologi nirkabel dan sensor dalam sistem pemantauan, khususnya di lingkungan maritim. Penulis ingin melihat bagaimana perangkat keras dan perangkat lunak saling terhubung dan bekerja bersama, serta tantangan teknis yang muncul ketika keduanya berkolaborasi dalam sistem.
2. Studi ini dapat menawarkan metode alternatif untuk meningkatkan keselamatan awak kapal dengan memantau kondisi gas berbahaya seperti CO dan NO₂. Sistem ini juga diharapkan dapat memberikan peringatan dini kepada awak kapal untuk mencegah potensi bahaya yang disebabkan oleh gas berbahaya, sehingga meningkatkan keselamatan dan kenyamanan dalam bekerja.
3. Hasil penelitian ini dapat digunakan sebagai referensi untuk penelitian lebih lanjut tentang pemantauan kualitas udara di lingkungan maritim,

khususnya dengan menerapkan teknologi *Internet of Things* (IoT) dan sistem sensor berbasis nirkabel serta bisa digunakan untuk mengembangkan sistem pemantauan gas berbahaya

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Pada sub-bab ini menjelaskan tentang beberapa penelitian terdahulu yang relevan dengan sistem pemantauan gas berbahaya berbasis ESP32 dan protokol MQTT telah dilakukan untuk memantau kualitas udara, yang sangat relevan dengan penelitian ini.

(Darmawan, 2025a) dalam penelitiannya yang berjudul “Rancang Bangun Alat Monitor Kualitas Udara Berbasis Web pada Kamar Mesin Kapal” menjelaskan mengenai pentingnya memonitor kualitas udara di ruang mesin kapal yang berfungsi sebagai pusat penggerak utama kapal. Studi ini berkonsentrasi pada pembuatan perangkat yang dapat mendeteksi gas berbahaya seperti Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂). Alat ini dibuat dengan sensor MQ-135 dan MQ-7 yang terhubung ke mikrokontroler ESP32, yang memungkinkan pengawasan dan penampilan data melalui antarmuka web. Pengujian sistem dilakukan di ruang mesin kapal dengan hasil pengukuran gas yang menunjukkan konsentrasi gas yang sesuai dengan kebutuhan *monitoring* lingkungan kapal. Penulis tidak membahas metode secara terperinci dalam penelitian ini, tetapi lebih menekankan pada pentingnya alat yang dapat mengawasi kualitas udara secara *real-time* dalam lingkungan ruang mesin kapal untuk meningkatkan keselamatan awak kapal dan mengurangi risiko keracunan gas.

Dengan menambahkan protokol MQTT untuk mengolah data dari sensor, pengembangan sistem ini dapat ditingkatkan. Misalnya, penelitian yang dilakukan oleh Prasetyo dkk. (2020), "Rancang Bangun Kontrol dan *Monitoring* Sensor Node WSN Menggunakan Protokol Message Queue Telemetry Transport," berfokus pada pembuatan alat yang dapat mendeteksi suhu secara *real-time*. Alat ini dibuat dengan menggunakan sensor DHT-11, yang kemudian disambungkan ke mikrokontroler ESP32. Mikrokontroler bertanggung jawab untuk mengelola dan menampilkan data melalui antarmuka web, dan sistem MQTT digunakan untuk berkomunikasi antara perangkat. Bagian ini memberi gambaran yang cukup jelas tentang bagaimana protokol MQTT bekerja secara efektif dalam memantau sekaligus mengendalikan

data secara *real-time*. Dari pengujian tersebut, sensor gas mencatat nilai konsentrasi yang stabil dan relevan, sehingga data yang diperoleh benar benar mencerminkan kondisi aktual di lapangan. Hasil ini menunjukkan bahwa sistem tidak hanya berjalan secara teoritis, tetapi juga mampu memenuhi kebutuhan pemantauan lingkungan secara nyata, terutama untuk mendukung aspek keselamatan dan keandalan operasional. Penulis tidak membahas metode secara terperinci dalam penelitian ini, tetapi lebih menekankan pada pentingnya alat yang dapat mengawasi kualitas udara secara *real-time* dalam lingkungan.

Penggunaan Wireless Sensor Network (WSN) dan protokol MQTT sangat membantu dalam pengembangan sistem pemantauan gas beracun, pengiriman data menggunakan protokol MQTT sementara untuk memaksimalkan diperlukan beberapa titik pemantauan sehingga tidak berpaku disatu titik saja, WSN berperan untuk memfasilitasi komunikasi antara sensor yang dipasang di lokasi yang berbeda di dalam ruang *powerpack*. Sebuah penelitian yang berjudul “Sistem *Monitoring* Karbon Monoksida dengan Metode *Wireless Sensor Network* (WSN) Berbasis *Internet of Things* (IoT)” (Nursuwars, 2023) menunjukkan bahwa pengiriman data dengan sistem ini sangat efisien dan cepat, dengan waktu pengiriman data sekitar 10.1 detik pada skema pengujian pertama dan 5.31 detik pada skema kedua. Penelitian memberikan gambaran yang jelas tentang penerapan IoT dan WSN dalam pemantauan gas berbahaya yang menggunakan mikrokontroler ESP32, MQTT, dan sensor untuk memantau gas CO. Selain itu, penelitian ini juga menunjukkan bagaimana sistem ini dapat memberikan solusi *real-time* untuk memantau dan mengendalikan kondisi lingkungan yang berisiko.

2.2 Dasar Teori

Dalam menyelesaikan Tugas Akhir ini, peneliti menggunakan teori sebagai berikut:

2.2.1 *Powerpack*

powerpack sangat penting Untuk memastikan kelancaran operasional kapal, terutama dalam hal menyediakan energi yang dibutuhkan untuk menggerakkan kapal dan mendukung sistem lainnya yang penting. Meskipun ruang *powerpack* biasanya dirancang dengan kondisi terkontrol, sebenarnya mereka sering terpapar gas berbahaya seperti Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂), yang

jelas merugikan kesehatan awak kapal. *Powerpack* dihimpun dalam satu kontainer berukuran 20 feet dengan dimensi (PxLxT) 6 meter x 2 meter x 2,5 meter. Dalam sebuah penelitian yang dilakukan, dikatakan bahwa pemantauan kualitas udara secara *real-time* sangat efektif dalam mendeteksi gas berbahaya, seperti karbon monoksida dan nitrogen dioksida, terutama di ruang tertutup (Abdilah & Putri, 2024)

Selain itu, suhu dalam ruang *powerpack* yang bisa meningkat sangat tajam, juga berdampak pada kinerja sistem dan membahayakan keselamatan awak kapal. Oleh karena itu, pemantauan gas berbahaya dan suhu secara terus-menerus menjadi hal yang wajib dilakukan demi menjaga keselamatan dan efisiensi operasional kapal. Dengan adanya teknologi *Internet of Things* (IoT) dan protokol MQTT, sistem pemantauan gas berbahaya dan suhu dapat dikendalikan lebih efisien dan akurat, sehingga memungkinkan adanya peringatan dini dan pengendalian otomatis jika kondisi berbahaya terdeteksi. ESP32 sebagai mikrokontroler untuk menghubungkan sensor ke sistem berbasis IoT menjadi sangat relevan dalam implementasi pemantauan di ruang *powerpack* kapal.



Gambar 2.1 *Powerpack*
Sumber: Dokumentasi Pribadi

Tabel 2.1 Spesifikasi Generator dan Mesin Diesel

Parameter	Generator	Mesin Diesel
Merek / Tipe	STAMFORD N200G4	DOOSAN Diesel 222 L
Tipe	Alternator AC Sinkron	4-Langkah, Turbocharged
Daya Nominal	200 kVA	222 HP
Daya Aktif (PF 0.8)	160 kW	±165 kW
Daya Standby	220 kVA	—
Frekuensi	50 Hz / 60 Hz	Menyesuaikan (1500 / 1800 RPM)
Kecepatan Nominal	1500 RPM (50 Hz)	1500 RPM
Jumlah Fasa	3 Fasa	6 Silinder Segaris
Sistem Eksitasi	Brushless	Water Cooled
Regulasi Tegangan	±1%	Governor Mechanical

2.2.2 Sistem Pemantauan Gas Berbahaya

Sistem pemantauan gas berbahaya di ruang *powerpack* kapal bertujuan untuk mendeteksi konsentrasi gas berbahaya, Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂), serta memantau suhu di ruang sekitar secara *real-time*. Sistem ini memanfaatkan sensor MQ-135 dan MQ-7 untuk mendeteksi gas-gas tersebut. Data yang terkumpul dari sensor akan diproses oleh ESP32 dan diteruskan ke sistem pemantauan menggunakan protokol MQTT. Dalam studi yang dilakukan, mereka menjelaskan bahwa penggunaan sensor MQ-135 dan MQ-7 sangat efektif dalam mendeteksi gas berbahaya seperti CO dan NO₂ dalam sistem *monitoring* berbasis IoT untuk menjaga kualitas udara yang aman (Utami dkk., 2025). Menurut ISO 8861 yang mengatur tentang “*Engine room ventilation diesel ships*” *Air Circulation per Hour (ACH)* memiliki nilai batas 15 untuk gas beracun akibat gas pembakaran mesin diesel.

Nilai ambang batas (NAB) yang ditetapkan oleh Peraturan Menteri Ketenagakerjaan Tahun 2018 Nomor 5. Konsentrasi CO dibatasi hingga 25 PPM untuk 8 jam kerja per hari, sedangkan NO₂ memiliki batas yang lebih ketat karena tingkat toksisitasnya yang lebih tinggi, berikut klasifikasinya:

Tabel 2.2 Nilai Ambang Batas Konsentrasi Gas

NO	Gas	Klasifikasi	Nilai Ambang Batas Konsentrasi Gas (PPM)
1	Karbon Monoksida (CO)	Normal	0 – 10
	Karbon Monoksida (CO)	Warning	>10 – 25
	Karbon Monoksida (CO)	Danger	>25
2	Nitrogen Dioksida (NO ₂)	Normal	0 – 1
	Nitrogen Dioksida (NO ₂)	Warning	>1 – 5
	Nitrogen Dioksida (NO ₂)	Danger	>5

Penelitian yang dilakukan oleh (Darmawan, 2025) sebelumnya mengatakan bahwa hasil pengujian gas Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂) menunjukkan 8 skenario dengan variasi konsentrasi gas berbahaya berdasarkan Tabel 2.1 NAB diatas, ditunjukkan dengan tabel 2.2 dibawah ini:

Tabel 2.3 Hasil Pengujian Gas

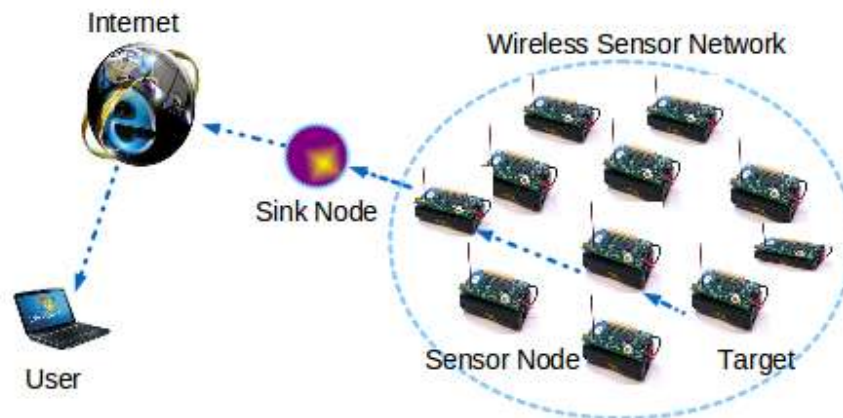
No	CO (PPM)	NO2 (PPM)	Keterangan
1	0.78	0.2	Normal semua
2	8.80	1.76	Buruk semua
3	9.83	2.45	CO buruk, NO2 buruk
4	2.09	3.54	CO bersih, NO2 kotor
5	1.10	9.83	CO buruk, NO2 kotor
6	1.0	2.55	CO buruk, NO2 kotor
7	7.16	17.24	CO buruk, NO2 kotor
8	8.45	8.90	CO buruk, NO2 kotor

2.2.3 Wireless Sensor Network (WSN)

Wireless Sensor Network (WSN) merupakan komponen utama dalam sistem pemantauan gas berbahaya yang dirancang pada penelitian ini, berfungsi menghubungkan sensor CO (Carbon Monoxide) dan NO₂ (Nitrogen Dioksida) yang ditempatkan pada 4 titik strategis di ruang *powerpack* kapal. Penempatan multi-titik ini dipilih bukan tanpa alasan, melainkan didasarkan pada pertimbangan bahwa ruang *powerpack* memiliki dimensi yang cukup luas dengan sumber potensi kebocoran gas yang tersebar, seperti sistem pembakaran mesin utama, generator, dan komponen pendukung lainnya, sehingga satu titik sensor saja tidak akan mampu merepresentasikan kondisi udara di seluruh area secara akurat.

WSN dalam sistem ini terdiri dari beberapa sensor node yang saling terhubung secara nirkabel, di mana masing-masing node bertanggung jawab mendeteksi parameter lingkungan tertentu, meliputi konsentrasi gas berbahaya (CO dan NO₂), suhu, serta kelembaban ruangan. Penambahan parameter suhu dan kelembaban bukan sekadar pelengkap, melainkan memiliki fungsi teknis penting, yaitu untuk mengoreksi dan mendukung akurasi pembacaan sensor gas, mengingat karakteristik sensor semikonduktor seperti MQ series diketahui memiliki

sensitivitas yang dipengaruhi oleh kondisi suhu dan kelembaban lingkungan sekitarnya.



Gambar 2.2 sistem Wireless Sensor Network

Sumber: <https://microcontrollerslab.com/wireless-sensor-networks-wsn-applications/>

Sebagai mikrokontroler utama, ESP32 dipilih karena memiliki modul konektivitas nirkabel terintegrasi (Wi-Fi dan Bluetooth) tanpa memerlukan modul tambahan, sehingga lebih efisien dari sisi biaya dan desain sistem, sekaligus tetap mampu menangani proses akuisisi data dari beberapa sensor node sekaligus dalam topologi WSN. Dalam arsitektur sistem ini, ESP32 berperan sebagai koordinator yang mengumpulkan data dari setiap sensor node yang tersebar di 4 titik tersebut, melakukan pra-pemrosesan data bila diperlukan, kemudian meneruskannya ke sistem pemantauan pusat untuk ditampilkan kepada pengguna.

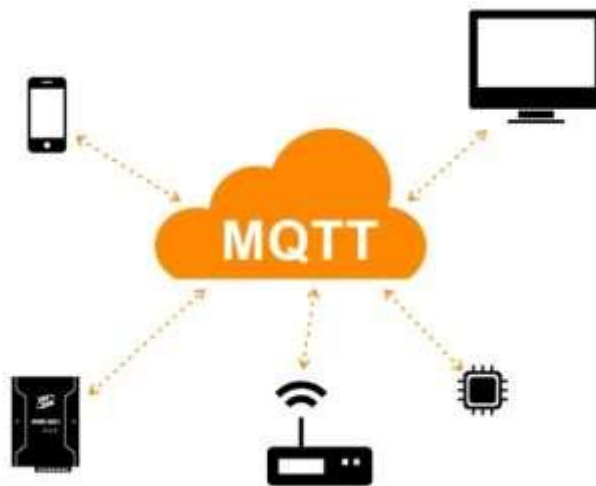
Sarmin dkk., (2025) dalam penelitiannya mengatakan IoT dan WSN merupakan teknologi modern yang bisa digunakan untuk *monitoring* kualitas udara secara *real time*. Sistem ini membantu memantau kualitas udara di ruang *powerpack* kapal secara *real-time* dan memberikan respons cepat apabila ada perubahan signifikan pada kadar gas atau suhu, sehingga meningkatkan keselamatan dan efisiensi operasional kapal secara keseluruhan.

2.2.4 Protokol MQTT (*Message Queuing Telemetry Transport*)

Pada sistem yang dirancang, protokol MQTT diterapkan dengan skema publish/subscribe untuk menghubungkan tiga jenis sensor, yaitu MQ-135, MQ-7, dan SHT20, dengan ESP32 yang berperan sebagai pengendali sekaligus pengolah data. Sensor MQ-135 berfungsi mendeteksi kualitas udara secara umum termasuk

kandungan gas NO₂, sensor MQ-7 secara spesifik mendeteksi konsentrasi gas CO, sedangkan SHT20 digunakan untuk mengukur suhu dan kelembaban udara di ruang *powerpack*. Ketiga sensor tersebut bertindak sebagai *publisher* yang mengirimkan data hasil pembacaannya secara berkala ke ESP32, yang kemudian meneruskan data tersebut ke broker MQTT untuk didistribusikan ke *subscriber* pada sisi *monitoring*.

Pemilihan protokol MQTT pada sistem *monitoring* kualitas udara berbasis Internet of Things (IoT) ini didasarkan pada karakteristiknya yang ringan (*lightweight*) dan hemat bandwidth, sehingga lebih sesuai diterapkan pada lingkungan ruang *powerpack* kapal yang umumnya memiliki keterbatasan infrastruktur jaringan, dibandingkan protokol komunikasi konvensional seperti HTTP yang membutuhkan overhead data lebih besar pada setiap proses request-response. Melalui mekanisme publish/subscribe ini, data sensor dapat dikirim ke server cloud dan ditampilkan secara *real-time* (Surbakti dkk., 2024), sebuah karakteristik yang sejalan dengan kebutuhan sistem pemantauan gas berbahaya pada penelitian ini, di mana keterlambatan informasi sekecil apa pun berpotensi berakibat fatal terhadap keselamatan awak kapal.



Gambar 2.3 Protokol MQTT

Sumber: <https://IoTstudio.labs.telkomuniversity.ac.id/berkenalan-dengan-MQTT/>

Lebih lanjut, protokol ini memungkinkan sistem memberikan peringatan dini secara otomatis apabila kadar gas berbahaya, seperti CO atau NO₂, melebihi ambang batas aman yang telah ditentukan. Ketika nilai pembacaan sensor

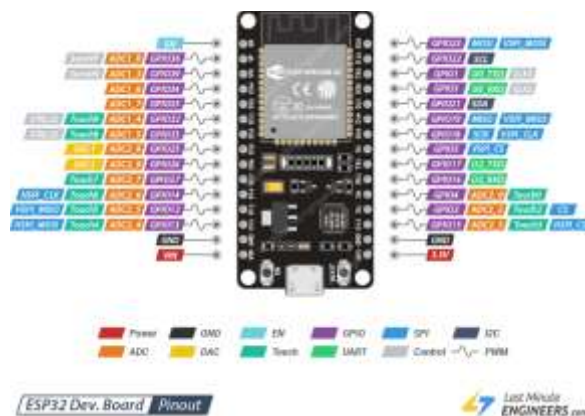
melampaui ambang batas tersebut, ESP32 akan mempublikasikan status peringatan melalui topik MQTT khusus, yang kemudian diteruskan ke antarmuka *monitoring* dalam bentuk notifikasi visual (indikator pada *Dashboard*) maupun *audible alarm*. Mekanisme ini memberikan jeda waktu yang krusial bagi awak kapal untuk mengambil tindakan mitigasi yang diperlukan, seperti melakukan evakuasi dari ruang *powerpack*, mengaktifkan sistem ventilasi tambahan, atau menghentikan sementara operasional mesin hingga kondisi udara kembali pada level aman.

2.3 Kajian Pustaka Objek Perangkat Keras & Lunak

Dalam menyelesaikan Tugas Akhir ini, peneliti menggunakan Hardware dan Software sebagai berikut:

2.3.1 ESP32 Mikrokontroler

ESP32 digunakan untuk mengumpulkan data dari beberapa sensor yang terhubung untuk mendeteksi gas karbon monoksida (CO) dan Nitrogen Dioksida (NO₂), dalam sistem pemantauan berbasis *Internet of Things* (IoT). Dikarenakan kemampuan untuk konektivitas antar nirkabelnya yang terintegrasi, seperti Wi-Fi dan Bluetooth, ESP32 menjadi pilihan populer di antara berbagai jenis mikrokontroler untuk pengembangan sistem IoT (Sri Surani dkk., 2025)



Gambar 2.4 ESP32 Mikrokontroler

Sumber: <https://lastminuteengineers.com/esp32-pinout-reference/>

ESP32 mendukung komunikasi nirkabel berbasis WSN (*Wireless Sensor Network*), yang menghubungkan beberapa titik pemantauan sensor dalam ruang *powerpack* kapal. Protokol MQTT pada sistem ini memungkinkan pengiriman data secara *real-time* yang efisien, meminimalkan latensi, dan memberikan respons cepat terhadap perubahan konsentrasi gas atau suhu yang terdeteksi. Sensor suhu

SHT20 memungkinkan pemantauan kondisi suhu yang dapat mempengaruhi kinerja mesin dan kenyamanan kerja awak kapal. Selain itu, ESP32 memiliki kinerja yang lebih baik, kapasitas memori yang lebih memadai, dan *support* berbagai jenis sensor dibandingkan pendahulunya ESP8266 (Ikhsan Afif Asrory dkk., 2025), menjadikannya pilihan ideal untuk sistem pemantauan yang responsif di ruang terbatas seperti ruang *powerpack* kapal.

Mikrokontroler ESP32 juga memiliki banyak pin input/output (I/O) yang memudahkan penghubungannya dengan perangkat eksternal, seperti LCD yang dapat menampilkan informasi tentang kualitas udara dan suhu. Mikrokontroler ESP32 memiliki *buzzer* untuk memberi tahu awak kapal jika kadar gas berbahaya atau suhu melebihi ambang batas yang telah ditentukan. ESP32 juga mengontrol *Exhaust Fan* yang digunakan untuk menetralkan udara yang tercemar dan secara otomatis meningkatkan sistem ventilasi. Sistem pemantauan berbasis ESP32 ini tidak hanya efektif dalam mengumpulkan dan mengirimkan data gas berbahaya, tetapi juga dapat memantau suhu secara langsung dan melakukan tindakan pengendalian otomatis untuk menjaga kondisi ruang *powerpack* kapal tetap aman dan nyaman.

Tabel 2.4 Spesifikasi Mikrokontroler ESP32

Parameter	Nilai / Keterangan
Prosesor	Dual-core Xtensa LX6 32-bit, hingga 240 MHz
Memori SRAM	520 KB
WiFi/Bluetooth	802.11 b/g/n, BLE 4.2
Tegangan Operasi	3.3V, input 5V via USB
GPIO Total	48 pin (34 usable pada DevKit)

2.3.2 Sensor Gas MQ-135

Sensor MQ-135 mampu mendeteksi berbagai jenis gas berbahaya, seperti nitrogen dioksida (NO₂), amonia (NH₃), alkohol, dan senyawa gas lainnya. Sensor ini umumnya digunakan pada aplikasi pengukuran kualitas udara, seperti pemantauan udara dalam ruangan, pengendalian sistem ventilasi, serta sebagai pendukung awal pada peralatan deteksi kebakaran (Muttaqin dkk., 2024). Prinsip kerja sensor ini didasarkan pada perubahan resistansi elemen sensitif berbahan SnO₂ (Tin Dioxide) ketika bersentuhan dengan gas target; semakin tinggi konsentrasi gas yang terdeteksi, semakin besar pula perubahan nilai resistansi yang dihasilkan oleh lapisan sensitif tersebut.

Data output dari sensor MQ-135 berupa perubahan nilai resistansi tersebut selanjutnya dibaca dan diproses oleh mikrokontroler ESP32 untuk mengendalikan sistem berdasarkan kadar gas yang terdeteksi. Namun, nilai resistansi mentah yang dihasilkan sensor tidak dapat langsung diinterpretasikan sebagai satuan konsentrasi gas yang baku, sehingga diperlukan proses kalibrasi agar pembacaan sensor dapat dikonversi ke dalam nilai ppm (*parts per million*). Nilai ppm ini diperoleh melalui perhitungan rasio R_s/R_o , di mana R_s merupakan nilai resistansi sensor pada kondisi udara yang sedang diukur (mengandung gas target), sedangkan R_o adalah nilai resistansi sensor pada kondisi udara bersih sebagai acuan kalibrasi awal (Saputra, 2023). Persamaan untuk memperoleh nilai R_s dirumuskan sebagai berikut:

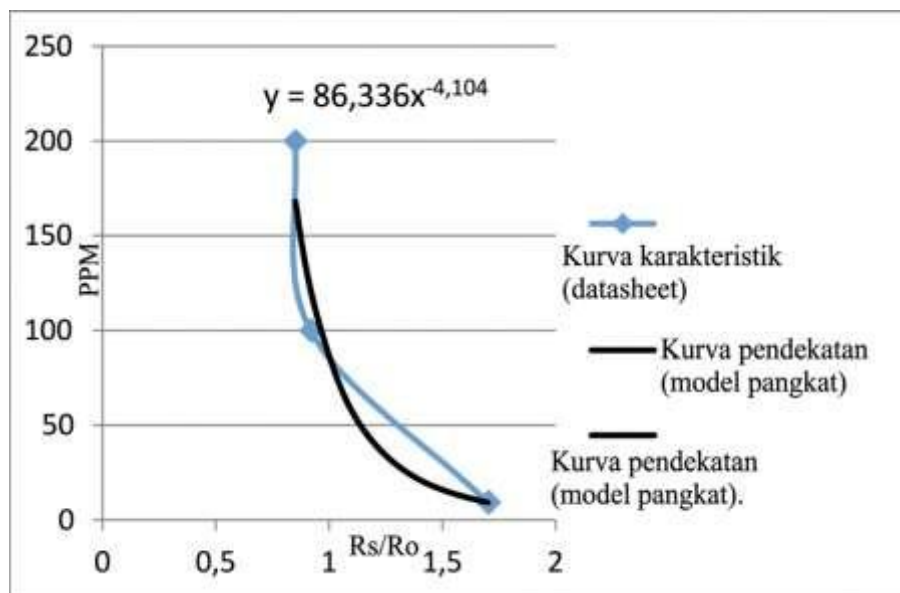
$$R_s = \frac{V_C}{V_{RL} - 1} \times R_L \quad (2.1)$$

Dimana,

V_C = TeganganInput

R_L = Nilai Tahanan (10K)

V_{RL} = Nilai Tahanan pada R_L



Gambar 2.5 Kurva kalibrasi Sensor MQ-135 (Ilham Ainul Bashir, 2024)

Pada saat udara bersih, sensor MQ-135 memiliki rasio sebesar (6000Ω) maka,

$$R_o = R_s/3,6 \quad (2.2)$$

Setelah nilai R_s dan R_o diperoleh, hubungan antara rasio sensor dan konsentrasi gas dalam satuan ppm perlu ditentukan. Pada tahap ini, persamaan logaritma digunakan agar nilai rasio tersebut dapat dikonversi menjadi besaran ppm, sebagaimana ditunjukkan pada persamaan berikut.

$$\text{Log}(y) = m * \log(x) + b \quad (2.3)$$

Dimana,

y = ratio (R_s/R_o)

x = ppm

m = kemiringan garis pada grafik

b = titik persimpangan



Gambar 2.6 Sensor Gas MQ-135
Sumber: <https://sl1nk.com/38jydx6>

Tabel 2.5 Spesifikasi Sensor MQ-135

Parameter	Nilai / Keterangan
Target Gas	NH ₃ , NO _x , alkohol, benzene, smoke, CO ₂ , sulfida
Range Deteksi	10-1000 ppm (NH ₃ /benzene), 10-300 ppm (alkohol)
Tegangan Operasi	VCC: 5V ±0.1V (AC/DC), VH: 5V ±0.1V
Konsumsi Daya	Heater: ≤950 mW, RH: 30-35 Ω (room temp.)
Output	Analog (0-5V), Digital (TTL HIGH/LOW)
Sensitivitas	$R_s(\text{air})/R_s(100\text{ppm NH}_3) \geq 5$, slope ≤0.65
Preheat Time	>24-48 jam
Suhu Operasi	-10°C ~ +50°C, RH 65%±5%
Dimensi	35x22x23 mm (module), pin 6

2.3.3 Sensor MQ-7

Sensor MQ-7 digunakan untuk mendeteksi kadar gas karbon monoksida (CO) di ruang *powerpack* kapal, mengingat CO merupakan gas yang tidak berwarna dan tidak berbau, sehingga keberadaannya sulit dideteksi secara langsung oleh awak kapal namun memiliki tingkat bahaya yang tinggi terhadap keselamatan jiwa

apabila terhirup dalam konsentrasi berlebih. Sama halnya dengan MQ-135, prinsip kerja sensor ini didasarkan pada perubahan resistansi elemen sensitif berbahan SnO_2 ketika bereaksi dengan gas CO di udara sekitar. Namun demikian, sensor MQ-7 memiliki karakteristik khusus yang membedakannya dari sensor MQ series lainnya, yaitu penerapan siklus pemanasan ganda (*heating cycle*) pada elemen pemanasnya, yang terdiri atas fase tegangan tinggi (*high voltage*) sebesar 5V selama kurang lebih 60 detik untuk membakar residu gas pada permukaan sensor, dan fase tegangan rendah (*low voltage*) sebesar 1,4V selama kurang lebih 90 detik untuk mengambil data pembacaan sensitivitas terhadap gas CO secara optimal.

ESP32 membaca nilai resistansi sensor tersebut melalui pin analog dan mengonversinya menjadi data digital melalui proses ADC (*Analog to Digital Converter*) yang tersedia secara internal pada mikrokontroler. Sebagaimana pada sensor MQ-135, nilai keluaran sensor MQ-7 juga memerlukan proses kalibrasi menggunakan rasio R_s/R_o agar dapat dikonversi ke dalam satuan ppm yang merepresentasikan konsentrasi gas CO secara akurat. Data hasil pembacaan tersebut selanjutnya dikirimkan oleh ESP32 ke sistem pemantauan berbasis MQTT untuk diteruskan ke *dashboard monitoring* secara *real-time*.

Melalui integrasi ini, data kadar gas CO yang terkumpul dapat dipantau secara berkelanjutan, sekaligus memungkinkan sistem memberikan peringatan dini secara otomatis apabila kadar gas CO melampaui batas aman yang telah ditentukan. Dengan demikian, sistem ini diharapkan dapat memberikan waktu respons yang memadai bagi awak kapal untuk melakukan tindakan mitigasi yang diperlukan, sehingga risiko keracunan gas CO di ruang *powerpack* kapal dapat diminimalkan secara signifikan.



Gambar 2.7 Sensor Gas MQ-7
Sumber: <https://11nq.com/ym2nxh0>

Tabel 2.6 Spesifikasi Sensor MQ-7

Parameter	Nilai / Keterangan
Target Gas	Carbon Monoxide (CO)
Range Deteksi	20-2000 ppm CO
Tegangan Operasi	Vc: 5V $\pm$ 0.1V (AC/DC), VH High: 5V $\pm$ 0.1V, VH Low: 1.4V $\pm$ 0.1V
Konsumsi Daya	PH: $\leq$ 350-900 mW
Output	Analog (voltage proportional), Digital (threshold)
Sensitivitas	$R_s(\text{air})/R_s(150\text{ppm CO}) \geq 5$, slope ≤ 0.6
Preheat Time	≥ 48 jam
Suhu Operasi	-20°C ~ 50°C, RH <95%
Resistor Heater	RH: 29-33 Ω $\pm$ 3-5% (room temp.)
Resistor Load	RL: Adjustable (tipikal 10 k Ω)

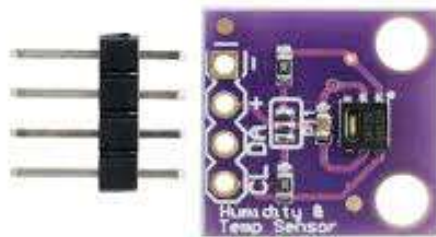
2.3.4 Sensor SHT20

Penggunaan sensor SHT20 pada penelitian ini ditujukan untuk mengukur parameter suhu dan kelembaban udara di sekitar ruang *powerpack* kapal, sebagai data pendukung yang melengkapi pembacaan sensor gas CO dan NO₂. Berbeda dengan sensor MQ-135 dan MQ-7 yang bekerja berdasarkan prinsip perubahan resistansi elemen semikonduktor, SHT20 merupakan sensor digital yang bekerja berdasarkan elemen kapasitif (*capacitive sensing element*) untuk mendeteksi perubahan kelembaban relatif, serta elemen sensor *bandgap* untuk mendeteksi perubahan suhu lingkungan. Sensor ini berkomunikasi dengan mikrokontroler ESP32 melalui protokol I2C (*Inter-Integrated Circuit*), sehingga data suhu dan kelembaban yang terukur dapat langsung diperoleh dalam bentuk data digital tanpa memerlukan proses konversi ADC sebagaimana pada sensor gas MQ series.

Pemilihan SHT20 pada penelitian ini didasarkan pada beberapa pertimbangan teknis dibandingkan sensor suhu-kelembaban sejenis, seperti DHT22 atau DHT11. SHT20 memiliki tingkat akurasi dan resolusi pembacaan yang lebih tinggi, konsumsi daya yang lebih rendah, serta waktu respons yang lebih cepat, sehingga lebih sesuai digunakan pada sistem pemantauan kontinu seperti pada penelitian ini, di mana kestabilan pembacaan jangka panjang menjadi faktor penting mengingat sensor akan beroperasi secara terus-menerus di lingkungan ruang *powerpack* kapal yang cenderung memiliki fluktuasi suhu akibat aktivitas mesin. Hal ini didukung oleh penelitian yang menunjukkan hasil kalibrasi sensor SHT20 dengan tingkat error rata-rata sebesar 0,7% untuk parameter suhu dan 1,74% untuk parameter kelembaban, yang setara dengan akurasi masing-masing sebesar 99,30% dan 98,26% (Luh dkk., 2021). Tingkat akurasi tersebut

menunjukkan bahwa sensor SHT20 layak digunakan sebagai sensor pendukung yang andal dalam sistem pemantauan gas berbahaya pada penelitian ini, khususnya sebagai data koreksi terhadap pembacaan sensor gas yang sensitif terhadap perubahan kondisi lingkungan sekitar.

Data suhu dan kelembaban yang diperoleh dari sensor SHT20 selanjutnya dikirimkan ke mikrokontroler ESP32 untuk diproses bersama data dari sensor gas lainnya. Data tersebut kemudian ditampilkan secara *real-time* pada layar LCD yang berfungsi sebagai antarmuka pemantauan lokal di ruang *powerpack*, sehingga memungkinkan awak kapal yang berada di lokasi untuk melakukan pengecekan kondisi udara secara langsung tanpa bergantung pada koneksi jaringan. Selain itu, data yang sama juga dikirimkan melalui protokol MQTT ke sistem *monitoring* pusat untuk mendukung pemantauan jarak jauh secara *real-time*. Dengan adanya dua jalur tampilan ini, sistem tetap dapat memberikan informasi kondisi lingkungan di ruang *powerpack* kapal secara andal, baik dalam kondisi jaringan internet tersedia maupun dalam situasi darurat ketika konektivitas jaringan mengalami gangguan.



Gambar 2.8 Sensor Gas SHT20

Sumber: <https://www.ebay.com/itm/395561140471>

Tabel 2.7 Spesifikasi Sensor SHT20

Parameter	Nilai / Keterangan
Tipe	Sensor suhu & kelembapan digital
Vcc	2,1 – 3,6 V DC
Rentang suhu	-40 hingga +125 °C
Akurasi suhu	±0,3 °C (kisaran kerja utama)
Antarmuka	I ² C digital

2.3.5 Exhaust fan Maspion CEF-25 10 inch

Exhaust fan berfungsi untuk mengalirkan udara dari dalam ruangan yang tercemar gas berbahaya dan mengeluarkannya ke luar ruangan atau ke sistem

ventilasi, sehingga konsentrasi gas berbahaya di dalam ruang *powerpack* dapat berkurang secara bertahap seiring dengan masuknya udara segar dari luar. Komponen ini memegang peranan penting dalam sistem, karena berfungsi sebagai aktuator utama yang secara langsung menindaklanjuti hasil pembacaan sensor gas, memastikan sirkulasi udara di dalam ruangan tetap sehat, serta mengurangi risiko akumulasi gas berbahaya seperti karbon monoksida (CO) dan nitrogen dioksida (NO₂) hingga mencapai konsentrasi yang membahayakan keselamatan awak kapal.

Pengendalian *exhaust fan* pada sistem ini dilakukan secara otomatis melalui program yang telah ditanamkan pada mikrokontroler ESP32, dengan memanfaatkan modul dimmer AC LSA-002 sebagai antarmuka saklar (*switching interface*) antara ESP32 dan catu daya *exhaust fan*. Data hasil pembacaan sensor gas MQ-7 dan MQ-135 secara terus-menerus dipantau oleh ESP32 dan dibandingkan dengan Nilai Ambang Batas (NAB) yang telah ditentukan sebelumnya. Ketika kadar gas CO atau NO₂ yang terdeteksi melebihi ambang batas tersebut, ESP32 akan mengirimkan sinyal untuk mengaktifkan modul LSA-002 pada kondisi *full on*, sehingga *exhaust fan* menyala pada kondisi penuh untuk mempercepat proses ventilasi dan membersihkan udara di dalam ruang *powerpack*. Sebaliknya, apabila konsentrasi gas telah kembali berada di bawah ambang batas aman, ESP32 akan mengirimkan sinyal untuk menonaktifkan modul LSA-002 ke kondisi *full off*, sehingga *exhaust fan* berhenti beroperasi dan konsumsi energi sistem dapat dihemat selama kondisi udara di ruangan berada pada level yang aman.



Gambar 2.9 Exhaust fan Maspion CEF-25 10 inch
Sumber: <https://11nq.com/5e2tgew>

Tabel 2.8 Spesifikasi Exhaust fan

Parameter	Nilai / Keterangan
Jenis	<i>Exhaust fan</i> dinding / dapur 10 inch AC
Tegangan kerja	220–240 V AC, 50/60 Hz
Daya input listrik	30 W
Debit udara	500–700 m ³ /jam
Airflow ekuivalen CFM	Sekitar 410–576 CFM
Kecepatan putar	±2800 rpm

2.3.6 LCD 20x4

Liquid Crystal Display (LCD) 20x4 merupakan salah satu jenis perangkat keluaran (*output device*) yang umum digunakan dalam sistem embedded untuk menampilkan informasi secara visual kepada pengguna. Nama "20x4" merujuk pada kapasitas tampilan LCD tersebut, yaitu mampu menampilkan karakter sebanyak 20 kolom dan 4 baris secara bersamaan, sehingga total kapasitas tampilan mencapai 80 karakter dalam satu waktu. Dibandingkan dengan LCD berukuran lebih kecil seperti 16x2, LCD 20x4 memiliki kapasitas tampilan yang lebih luas, sehingga lebih sesuai digunakan pada sistem yang perlu menampilkan beberapa parameter data sekaligus secara bersamaan tanpa harus bergantian (*scrolling*).

Secara umum, LCD 20x4 menggunakan chip pengendali (*driver IC*) HD44780 atau yang kompatibel dengannya, yang berfungsi mengatur proses penampilan karakter pada layar berdasarkan data digital yang diterima dari mikrokontroler. Untuk mempermudah proses pengkabelan (*wiring*) dan mengurangi jumlah pin yang digunakan pada mikrokontroler, LCD 20x4 umumnya dilengkapi dengan modul tambahan berupa I2C (*Inter-Integrated Circuit converter*), yang memungkinkan komunikasi antara LCD dan mikrokontroler hanya menggunakan dua jalur data, yaitu SDA (*Serial Data*) dan SCL (*Serial Clock*), dibandingkan dengan metode koneksi paralel konvensional yang membutuhkan lebih banyak jalur pin.

Dalam konteks sistem pemantauan berbasis mikrokontroler, LCD berperan penting sebagai antarmuka tampilan lokal (*local display interface*) yang memungkinkan pengguna memperoleh informasi secara langsung tanpa harus mengakses sistem pemantauan jarak jauh, seperti *Dashboard* berbasis web atau aplikasi *mobile*. Informasi yang umum ditampilkan pada LCD dalam sistem pemantauan gas berbahaya meliputi nilai konsentrasi gas hasil pembacaan sensor

dalam satuan ppm (*parts per million*), serta status kondisi lingkungan secara kualitatif, seperti indikasi "aman" atau "berbahaya", yang biasanya ditentukan berdasarkan perbandingan nilai pembacaan sensor terhadap Nilai Ambang Batas (NAB) yang telah ditetapkan. Dengan adanya tampilan informasi secara langsung tersebut, pengguna atau petugas yang berada di lokasi dapat memperoleh gambaran kondisi lingkungan secara cepat, sehingga dapat meningkatkan kesiapsiagaan dan respons terhadap potensi bahaya yang mungkin timbul di sekitar area pemantauan.



Gambar 2.10 LCD 20x4

Sumber: <https://shorturl.at/iT0tv>

Tabel 2.9 Spesifikasi LCD 20x4

Parameter	Nilai / Keterangan
Tipe	LCD karakter 20 kolom × 4 baris
Tegangan kerja	5 V DC
Kontroler	HD44780 atau kompatibel
Jumlah karakter	20 Karakter x 4 baris
Ukuran modul	± 98 × 60 mm
Antarmuka	Paralel (4-bit / 8-bit), sering via I2C module
Backlight	LED (hijau)

2.3.7 Buzzer SFM-27

Buzzer merupakan komponen elektronik yang berfungsi sebagai alat peringatan (*alarm*) dengan mengubah energi listrik menjadi gelombang bunyi, sehingga dapat memberikan sinyal audio kepada pengguna ketika suatu kondisi tertentu terpenuhi. Berdasarkan prinsip kerjanya, *buzzer* secara umum dapat dibedakan menjadi dua jenis, yaitu *buzzer* aktif dan *buzzer* pasif. *Buzzer* aktif memiliki rangkaian osilator internal yang sudah terintegrasi di dalam komponennya, sehingga dapat menghasilkan bunyi hanya dengan diberikan sumber tegangan DC secara langsung, tanpa memerlukan sinyal frekuensi tambahan dari

mikrokontroler. Hal ini berbeda dengan *buzzer* pasif, yang memerlukan sinyal frekuensi (*square wave*) dari mikrokontroler melalui pin PWM agar dapat menghasilkan bunyi, sehingga frekuensi dan nada suara pada *buzzer* pasif dapat diatur secara bervariasi, sementara pada *buzzer* aktif nada yang dihasilkan cenderung tetap (*fixed tone*).

Pada penelitian ini, digunakan *buzzer* aktif sebagai komponen keluaran (*output device*) untuk memberi tahu pengguna apabila kadar gas berbahaya telah melampaui ambang batas yang telah ditentukan. Pemilihan *buzzer* aktif didasarkan pada kesederhanaan rangkaian pengendaliannya, di mana mikrokontroler cukup memberikan sinyal logika HIGH atau LOW pada pin digital untuk mengaktifkan atau menonaktifkan *buzzer*, tanpa memerlukan konfigurasi sinyal PWM tambahan sebagaimana pada *buzzer* pasif.

Dalam sistem pemantauan gas ini, *buzzer* akan berbunyi secara otomatis ketika kadar gas berbahaya, seperti karbon monoksida (CO) atau nitrogen dioksida (NO₂), terdeteksi mencapai tingkat yang membahayakan berdasarkan Nilai Ambang Batas (NAB) yang telah ditetapkan, sehingga memberikan sinyal peringatan kepada pekerja atau awak kapal untuk segera mengambil tindakan pencegahan atau mitigasi yang diperlukan. Penggunaan *buzzer* sebagai indikator peringatan audio ini penting untuk memastikan bahwa pekerja atau pengguna lain dapat memperoleh informasi yang tepat mengenai kondisi lingkungan di sekitar mereka secara cepat, terutama pada situasi darurat di mana perhatian visual terhadap layar tampilan (LCD atau *Dashboard*) mungkin tidak selalu memungkinkan, sehingga sinyal audio dari *buzzer* dapat menjadi mekanisme peringatan tambahan yang lebih responsif.



Gambar 2.11 *Buzzer* SFM-27
Sumber: <https://11nk.dev/itgm0dy>

Tabel 2.10 Spesifikasi *Buzzer*

Parameter	Nilai / Keterangan
Jenis	<i>Buzzer</i> aktif (built-in oscillator)
Tegangan kerja	3V - 24V DC
Arus konsumsi	< 30 mA
Frekuensi suara	3000 ± 500 Hz
Cara kendali	On/Off dengan tegangan DC
Fungsi	Indikator bunyi / alarm

2.3.8 Modul LSA-002 AC Dimmer

Modul LSA-002 AC Dimmer merupakan modul elektronik yang umum digunakan sebagai pengatur tegangan AC pada berbagai sistem otomasi, khususnya untuk mengendalikan beban berupa motor AC atau lampu. Modul ini bekerja berdasarkan prinsip *phase control* menggunakan komponen TRIAC (*Triode for Alternating Current*), yaitu dengan memotong sebagian gelombang sinusoidal tegangan AC pada setiap siklusnya melalui pengaturan sudut penyalan (*firing angle*) yang disinkronkan terhadap titik potong nol (*zero-crossing*) gelombang tegangan. Dengan mengatur sinyal kendali yang diberikan pada terminal input modul, tegangan efektif (RMS) yang diteruskan ke beban dapat diatur, sehingga secara teoritis modul ini mampu mengendalikan kecepatan putaran motor AC secara bertingkat. Modul ini umum digunakan pada aplikasi otomasi karena kemampuannya menangani beban induktif seperti motor AC secara lebih andal dan minim risiko percikan bunga api (*arcing*) dibandingkan penggunaan relai mekanik konvensional.

Pada penelitian ini, Modul LSA-002 AC Dimmer dimanfaatkan sebagai antarmuka saklar elektronik (*switching interface*) untuk mengendalikan *exhaust fan* berdasarkan data yang dibaca oleh sensor MQ-135 dan MQ-7 dalam mendeteksi konsentrasi gas karbon monoksida (CO) serta nitrogen dioksida (NO₂). Meskipun modul ini pada dasarnya mampu mengatur kecepatan motor secara bertingkat, pada implementasi penelitian ini modul dioperasikan secara penuh (*full on*) atau dimatikan sepenuhnya (*full off*), sehingga berfungsi sebagai saklar on/off berbasis TRIAC. Ketika konsentrasi gas yang terpantau melebihi Nilai Ambang Batas (NAB) yang telah ditetapkan, ESP32 mengirimkan sinyal kendali untuk mengaktifkan modul LSA-002 pada kondisi *full on*, sehingga *exhaust fan* menyala dan berputar pada kecepatan penuh untuk mempercepat sirkulasi udara. Sebaliknya, apabila konsentrasi gas telah kembali berada di bawah ambang batas aman, ESP32

akan menonaktifkan modul tersebut ke kondisi *full off*, sehingga *exhaust fan* berhenti beroperasi. Mekanisme ini membantu memperlancar sirkulasi udara di ruang *powerpack*, yang pada akhirnya berperan dalam menurunkan kadar gas berbahaya di lingkungan sekitar secara otomatis dan responsif.



Gambar 2.12 Modul LSA-002 AC Dimmer

Sumber: <https://www.arduinoindonesia.id/2022/10/driver-motor-LSA-002.html>

Tabel 2.11 Spesifikasi Modul LSA-002 AC Dimmer

Parameter	Nilai / Keterangan
Jenis modul	Modul dimmer AC 1 channel berbasis triac + ZCD
Tegangan AC beban	110–240 V AC, 50/60 Hz
Tegangan kontrol (VCC)	5 V DC (kompatibel dengan logika 3,3 V–5 V, mis. ESP32)
Metode kontrol	Sinyal PWM / phase control dari mikrokontroler
Arus beban maks.	2 A (normal), beban hingga sekitar 400 W pada 220 V
Daya beban maks.	±400 W (disarankan tidak melebihi untuk keandalan)
Tegangan keluaran	0 V sampai mendekati tegangan input (tergantung duty)

2.3.9 Raspberry pi 3

Raspberry Pi 3 merupakan komputer papan tunggal (*single board computer*, SBC) yang memiliki kemampuan pemrosesan data jauh lebih tinggi dibandingkan mikrokontroler konvensional seperti ESP32, sehingga sering diposisikan sebagai perangkat pengolah data tingkat lanjut dalam suatu sistem. Raspberry Pi 3 Model B dibangun menggunakan System on Chip (SoC) Broadcom BCM2837, yang terdiri atas prosesor *quad-core* ARM Cortex-A53 64-bit dengan kecepatan clock sebesar 1,2 GHz, dipadukan dengan memori RAM sebesar 1 GB tipe LPDDR2. Selain itu, perangkat ini juga dilengkapi dengan berbagai antarmuka konektivitas yang cukup lengkap untuk ukurannya yang kompak, di antaranya port HDMI untuk keluaran tampilan visual, empat port USB 2.0, port LAN (Ethernet) untuk koneksi

jaringan kabel, modul Wi-Fi dan Bluetooth terintegrasi untuk konektivitas nirkabel, serta pin GPIO (*General Purpose Input/Output*) yang dapat dimanfaatkan untuk berinteraksi dengan perangkat elektronik eksternal.

Dalam konteks Wireless Sensor Network (WSN) pada sistem pemantauan gas berbahaya, Raspberry Pi 3 diposisikan sebagai *node server* sekaligus *data logger*, yaitu perangkat yang berperan mengumpulkan, mengolah, dan menyimpan data dari seluruh sensor node yang tersebar di berbagai titik pemantauan. Sebagai *node server*, Raspberry Pi 3 bertugas menerima data yang dikirimkan oleh ESP32 melalui protokol MQTT, kemudian melakukan pengolahan lanjutan terhadap data tersebut, seperti agregasi data dari beberapa titik sensor, perhitungan rata-rata konsentrasi gas, maupun visualisasi data dalam bentuk grafik sederhana. Sebagai *data logger*, Raspberry Pi 3 bertanggung jawab menyimpan data hasil pemantauan secara berkala ke dalam basis data lokal maupun media penyimpanan eksternal, sehingga riwayat data konsentrasi gas CO dan NO₂ pada ruang *powerpack* dapat ditelusuri kembali untuk keperluan analisis maupun pelaporan di kemudian hari. Kemampuan pemrosesan yang lebih tinggi serta ketersediaan konektivitas jaringan kabel dan nirkabel pada satu perangkat menjadikan Raspberry Pi 3 pilihan yang efisien sebagai simpul pengolah data tingkat atas dalam arsitektur WSN, tanpa memerlukan penambahan modul eksternal yang berlebihan.



Gambar 2.13 Raspberry pi 3
Sumber: <https://l1nq.com/elnayuj>

Tabel 2.12 Spesifikasi Raspberry pi 3

Parameter	Nilai / Keterangan
SoC	Broadcom BCM2837, quad-core ARM Cortex-A53 1,2 GHz 64-bit
Memori RAM	1 GB LPDDR2 SDRAM
Penyimpanan	Slot microSD untuk sistem operasi dan penyimpanan data
Konektivitas	Ethernet 10/100 Mbps, Wi-Fi 2,4 GHz 802.11b/g/n, Bluetooth 4.1 (BLE)
Antarmuka	1× HDMI penuh, 4× USB 2.0, 40-pin GPIO, CSI camera, DSI display
Tegangan suplai	5 V DC melalui micro USB, arus disarankan sekitar 2,5 A

2.3.10 Battery Lithium 18650 3.7V

Baterai 18650 yang digunakan pada penelitian ini disusun dalam konfigurasi 2S2P (*2 series, 2 parallel*), yaitu menggunakan total 4 sel lithium-ion tipe 18650, di mana 2 sel disusun secara seri untuk mendapatkan tegangan yang lebih tinggi, dan disusun secara paralel sebanyak 2 *string* untuk meningkatkan kapasitas arus (Ah) yang dapat disuplai oleh baterai. Setiap sel 18650 memiliki tegangan nominal sekitar 3,7V, sehingga ketika 2 sel disusun secara seri (2S), diperoleh tegangan nominal paket baterai sebesar 7,4V, dengan tegangan penuh (*fully charged*) mencapai sekitar 8,4V ($2 \times 4,2\text{V}$ per sel). Sementara itu, konfigurasi paralel (2P) berfungsi menggandakan kapasitas total baterai tanpa mengubah nilai tegangan keluaran, sehingga daya tahan (*runtime*) baterai dalam menyuplai beban menjadi lebih lama dibandingkan konfigurasi tanpa paralel.

Paket baterai dengan konfigurasi 2S2P ini digunakan secara khusus untuk menyuplai daya pada panel *slave*, yaitu unit sensor node yang bersifat portabel dan terpisah dari panel utama (*master*). Tegangan nominal 7,4V yang dihasilkan dari konfigurasi 2S selanjutnya diturunkan menggunakan modul *step-down* MP1584 ke level tegangan yang aman dan sesuai untuk mencatu daya mikrokontroler ESP32 pada panel slave tersebut.

Modul BMS 2S digunakan sebagai papan proteksi dan penyeimbang (*balancer*) antar sel baterai dalam konfigurasi 2S2P ini. BMS berfungsi memantau tegangan pada setiap sel baterai secara individual, serta memutus hubungan beban maupun sumber pengisian daya (*charger*) apabila terjadi kondisi *overcharge* (tegangan per sel melebihi sekitar 4,2V) maupun *overdischarge* (tegangan per sel turun di bawah batas aman, sekitar 2,5–3,0V), sehingga umur pakai baterai dapat lebih panjang dan risiko kerusakan atau kebakaran akibat pengisian maupun pemakaian berlebih dapat diminimalkan. Selain itu, modul BMS 2S ini juga menyediakan perlindungan terhadap kondisi arus lebih (*overcurrent*) dan hubungan singkat (*short-circuit*), serta pada beberapa tipe dilengkapi dengan fungsi *cell balancing*, yang berfungsi menyeimbangkan tegangan antar sel selama proses

pengisian daya berlangsung, sehingga tidak ada sel yang mengalami pengisian berlebih maupun tertinggal dibandingkan sel lainnya.

Dengan kombinasi 4 sel 18650 dalam konfigurasi 2S2P dan modul BMS 2S, paket baterai yang dihasilkan menjadi lebih aman, ringkas, dan praktis untuk diterapkan sebagai sumber daya portabel pada panel slave berbasis ESP32 di ruang *powerpack* kapal, tanpa memerlukan sumber listrik AC langsung pada titik pemasangan sensor node yang tersebar di berbagai lokasi. Sifat portabel ini menjadi pertimbangan penting mengingat panel slave ditempatkan pada beberapa titik yang tersebar di dalam ruang *powerpack* sesuai dengan rancangan arsitektur WSN yang telah dibahas sebelumnya, sehingga penarikan kabel daya AC ke setiap titik sensor node akan menambah kompleksitas instalasi serta berpotensi mengurangi fleksibilitas penempatan sensor di masa mendatang.



Gambar 2.14 Battery Lithium 18650
Sumber: <https://11nq.com/tt71b1e>

Tabel 2.13 Spesifikasi Battery Lithium 18650

Parameter	Nilai / Keterangan
Jenis Sel	Lithium-ion 18650
Konfigurasi	2S2P (2 Seri, 2 Paralel)
Jumlah Sel	4 sel
Tegangan Nominal per Sel	3,7 V
Tegangan Nominal Paket Baterai	7,4 V
Tegangan Penuh Paket Baterai	8,4 V
Modul Proteksi	BMS 2S
Batas Overcharge	$\pm 4,2$ V per sel
Batas Overdischarge	$\pm 2,5-3,0$ V per sel

2.3.11 Power Supply 12V 10A

Power Supply Unit (PSU) merupakan perangkat elektronik yang berfungsi mengonversi sumber tegangan listrik arus bolak-balik (AC) dari jaringan listrik

utama menjadi tegangan arus searah (DC) yang stabil, sesuai dengan kebutuhan catu daya perangkat elektronik yang disuplai. Secara umum, PSU yang digunakan pada aplikasi elektronik modern merupakan jenis *switching power supply* (SMPS), yang bekerja dengan prinsip menyalakan dan mematikan (*switching*) tegangan input pada frekuensi tinggi menggunakan komponen semikonduktor seperti MOSFET, kemudian disearahkan dan difilter untuk menghasilkan tegangan DC yang stabil pada keluarannya. Dibandingkan dengan jenis PSU linier konvensional, SMPS memiliki keunggulan berupa efisiensi konversi daya yang lebih tinggi, ukuran dan bobot yang lebih ringkas, serta menghasilkan panas berlebih yang lebih sedikit, karena tidak bergantung pada trafo besar berfrekuensi rendah sebagaimana pada PSU linier.

Kapasitas daya sebesar 120 Watt ini menjadikan PSU jenis ini umum digunakan pada aplikasi yang membutuhkan suplai daya cukup besar namun tetap dalam kategori tegangan rendah dan aman (*low voltage DC*), seperti pencatuan daya untuk perangkat CCTV, lampu LED strip, motor DC, maupun perangkat komputer papan tunggal (*single board computer*) seperti Raspberry Pi yang memerlukan tegangan turunan lebih lanjut melalui modul *buck converter*.

Pada bagian internalnya, PSU umumnya dilengkapi dengan berbagai komponen proteksi, seperti pembatas arus lebih (*over current protection*), proteksi terhadap tegangan lebih (*over voltage protection*), serta proteksi hubung singkat (*short circuit protection*), yang berfungsi untuk melindungi baik perangkat PSU itu sendiri maupun beban yang terhubung dari kerusakan akibat gangguan kelistrikan. Ketiga jenis proteksi ini bekerja secara otomatis tanpa memerlukan intervensi manual; misalnya, apabila arus yang ditarik oleh beban melebihi batas maksimum 10A, PSU akan secara otomatis membatasi atau memutus keluaran arus untuk mencegah kerusakan pada komponen internal maupun perangkat yang disuplai. Selain itu, PSU jenis ini juga umumnya dilengkapi trimpot (potensiometer kecil) yang memungkinkan penyesuaian nilai tegangan output dalam rentang tertentu di sekitar nilai nominalnya, guna mengakomodasi kebutuhan kalibrasi tegangan pada instalasi tertentu, misalnya untuk mengompensasi rugi tegangan (*voltage drop*) akibat panjangnya kabel distribusi daya di dalam panel.

Dalam konteks penelitian ini, PSU 12V/10A digunakan sebagai sumber daya utama pada panel master, yang menyuplai tegangan tetap sebesar 12V DC ke seluruh rangkaian yang berada pada panel tersebut. Tegangan 12V ini selanjutnya didistribusikan ke dua modul *step-down* yang berbeda, yaitu LM2596 dan MP1584, yang masing-masing bertugas mencatu daya perangkat dengan kebutuhan tegangan kerja yang berbeda. Modul LM2596 digunakan untuk menurunkan tegangan 12V menjadi 5V guna mencatu daya Raspberry Pi 3 yang berfungsi sebagai *node server* pada arsitektur WSN, sedangkan modul MP1584 digunakan untuk menurunkan tegangan 12V menjadi level tegangan yang sesuai bagi rangkaian ESP32 dan komponen pendukung lainnya pada PCB Master. Pembagian ini dilakukan agar masing-masing modul *step-down* dapat bekerja secara independen sesuai dengan kebutuhan arus dan tegangan spesifik dari beban yang disuplai, sehingga tidak terjadi pembebanan berlebih (*overload*) pada satu modul akibat menanggung dua jenis beban dengan karakteristik yang berbeda sekaligus.

Penggunaan PSU sebagai sumber daya tetap (bukan baterai) pada panel master ini dipilih karena panel master bersifat statis dan tidak berpindah tempat, sehingga ketersediaan sumber listrik AC tetap dari kapal dapat dimanfaatkan secara langsung tanpa memerlukan sistem baterai portabel sebagaimana pada panel slave. Pendekatan ini memberikan sejumlah keuntungan, di antaranya suplai daya yang lebih stabil dan tidak terbatas oleh kapasitas baterai, tidak memerlukan proses pengisian ulang (*recharging*) secara berkala, serta mampu menanggung beban daya yang lebih besar secara terus-menerus (*continuous duty*), mengingat panel master menjalankan perangkat dengan kebutuhan daya relatif lebih tinggi dibandingkan panel slave, seperti Raspberry Pi 3 yang berperan sebagai pusat pengolahan dan penyimpanan data pada keseluruhan sistem WSN.



Gambar 2.15 Power Supply 12V 10A
Sumber: <https://acesse.one/btb08ta>

Tabel 2.14 Spesifikasi Power Supply 12V 10A

Parameter	Nilai / Keterangan
Jenis	Switching Power Supply (SMPS)
Tegangan Output	12 V DC
Arus Maksimum	10 A
Daya Output	120 Watt
Proteksi	Over Current, Over Voltage, Short Circuit
Pengaturan Tegangan	Trimpot (adjustable)
Jenis	Switching Power Supply (SMPS)
Tegangan Output	12 V DC
Arus Maksimum	10 A

2.3.12 Modul MP1584 *Buck Converter*

MP1584 merupakan modul *DC-DC buck converter* berukuran kecil yang berfungsi menurunkan tegangan input yang lebih tinggi menjadi tegangan output yang lebih rendah dengan tingkat efisiensi konversi yang tinggi, serta mampu menyalurkan arus keluaran hingga 3 A. Sebagai modul *step-down* (bukan *boost*), MP1584 lebih tepat digunakan pada kondisi di mana sumber tegangan yang tersedia berada di atas tegangan kerja beban yang dituju, seperti pada sistem penelitian ini, di mana tegangan sumber dari paket baterai 18650 perlu diturunkan ke level yang sesuai untuk mencatu daya ESP32, sensor, dan rangkaian pendukung lainnya.

Modul ini memiliki rentang tegangan input yang cukup lebar, yaitu sekitar 4,5–28 V, dengan tegangan output yang dapat diatur secara manual pada rentang 0,8–20 V melalui potensiometer yang tersedia pada modul. MP1584 bekerja pada frekuensi *switching* tinggi, yaitu sekitar 1 MHz, yang memungkinkan penggunaan komponen induktor berukuran lebih kecil tanpa mengorbankan efisiensi konversi daya. Efisiensi konversi modul ini dapat mencapai sekitar 92–96%, sehingga rugi daya dan panas yang timbul pada sistem dapat diminimalkan dibandingkan penggunaan regulator linier konvensional.

Selain itu, MP1584 dirancang untuk mendukung aplikasi catu daya yang kompak, seperti pada mikrokontroler, modul IoT, dan beban digital lainnya, dengan tingkat riak tegangan keluaran (*output ripple*) yang relatif kecil, yaitu sekitar 30 mV. Modul ini juga dilengkapi dengan sejumlah fitur proteksi internal, seperti pembatas arus (*current limiting*) dan pemutusan otomatis akibat panas berlebih (*thermal shutdown*), yang berfungsi meningkatkan keandalan dan keamanan sistem catu daya secara keseluruhan, khususnya pada penerapan sistem pemantauan gas berbahaya yang membutuhkan kestabilan suplai daya secara terus-menerus.



Gambar 2.16 Modul MP1584 *Buck Converter*
 Sumber: <https://shorturl.at/JTHRW>

Tabel 2.15 Spesifikasi Modul MP1584 *Buck Converter*

Parameter	Nilai / Keterangan
Jenis konverter	DC-DC step-down (buck) regulator
Tegangan input	4,5 V – 28 V
Tegangan output	0,8 V – 20 V, dapat disetel dengan potensiometer
Arus output maksimum	3 A (puncak), arus kontinu disarankan ~1,5–1,8 A
Efisiensi konversi	hingga sekitar 92–96% bergantung selisih V_{in} – V_{out}
Ripple output	sekitar < 30 mV
Frekuensi switching	sekitar 1 MHz (maks 1,5 MHz)
Suhu kerja	sekitar –20 °C sampai +85 °C (kadang datasheet –40/–45 sampai +85 °C)
Ukuran modul	±22 mm × 17 mm × 4 mm i

2.3.13 Modul LM2596 Step down

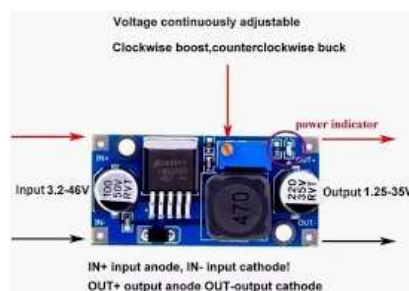
Modul LM2596 adalah modul konverter DC-DC tipe *buck (step-down)* yang berfungsi menurunkan tegangan DC dari sumber yang lebih tinggi ke tegangan yang lebih rendah dan stabil, misalnya 5V untuk catu daya Raspberry Pi 3. Modul ini dibangun berbasis IC LM2596, yaitu integrated circuit monolitik yang menyediakan seluruh fungsi aktif untuk regulator switching step-down, mampu menggerakkan beban hingga 3A dengan regulasi saluran dan beban yang sangat baik. Modul ini cocok digunakan pada sistem catu daya kompak karena memiliki efisiensi konversi tinggi dan ukuran fisik yang kecil, sehingga mudah diintegrasikan ke dalam rangkaian panel.

Secara spesifikasi teknis, modul LM2596 memiliki rentang tegangan input sekitar 3,2–46V DC dan mampu menurunkan tegangan hingga keluaran sebesar 1,25–35V DC dengan arus maksimal 3A, di mana tegangan output dapat diatur secara manual melalui potensiometer yang tersedia pada modul. Modul ini bekerja

pada frekuensi switching sekitar 65 kHz dengan efisiensi konversi mencapai sekitar 92%, serta memiliki riak tegangan keluaran (output ripple) sekitar 30 mV.

Pada penelitian ini, modul LM2596 digunakan secara khusus pada panel master, di mana sumber tegangan yang disuplai berasal langsung dari Power Supply Unit (PSU) dengan spesifikasi 12V/10A, bukan dari baterai. Tegangan 12V dari PSU tersebut diturunkan oleh LM2596 menjadi 5V untuk mencatu daya Raspberry Pi 3 sebagai *node server* pada panel master. Penggunaan sumber PSU tetap (bukan baterai) ini memberikan keuntungan berupa tegangan input yang senantiasa stabil pada 12V, sehingga selisih antara tegangan input dan output (12V ke 5V = 7V) jauh melampaui syarat selisih minimal 1,5V DC yang dipersyaratkan oleh LM2596, dan tidak berisiko mengalami kondisi *dropout* sebagaimana yang berpotensi terjadi bila disuplai dari baterai yang tegangannya menurun seiring pemakaian.

Kapasitas arus PSU sebesar 10A juga jauh melebihi kebutuhan arus Raspberry Pi 3 beserta beban tambahan seperti Wi-Fi aktif dan perangkat USB yang terhubung, bahkan setelah memperhitungkan batas arus keluaran LM2596 sebesar 3A, sehingga suplai daya pada panel master dapat dikatakan memiliki margin keamanan yang sangat memadai untuk mendukung operasional Raspberry Pi 3 secara terus-menerus (*continuous duty*) sebagai pusat pengolahan data pada arsitektur WSN.



Gambar 2.17 Modul LM2596 Step down

Sumber: <https://acesse.one/jg3fp5s>

Tabel 2.16 Spesifikasi Modul LM2596 Step down

Parameter	Nilai / Keterangan
Jenis Modul	DC-DC Buck converter (Step-Down)
IC Utama	LM2596
Tegangan Input	3,2–46 V DC (aktual: 12 V dari PSU)
Tegangan Output	1,25–35 V DC, adjustable (aktual: 5 V)
Arus Maksimum	3 A
Efisiensi	±92%
Frekuensi Switching	±65 kHz
Output Ripple	±30 mV

2.3.14 Modul GSM SIM800L

SIM800L merupakan modul komunikasi seluler (*GSM/GPRS module*) yang memungkinkan suatu perangkat mikrokontroler untuk terhubung dengan jaringan telepon seluler, sehingga dapat melakukan komunikasi data tanpa bergantung pada ketersediaan jaringan Wi-Fi maupun internet satelit. Modul ini bekerja dengan memanfaatkan kartu SIM (*Subscriber Identity Module*) layaknya perangkat telepon seluler pada umumnya, dan berkomunikasi dengan mikrokontroler melalui antarmuka serial UART menggunakan perintah AT (*AT Command*) untuk mengatur berbagai fungsi komunikasinya, termasuk koneksi data GPRS untuk pengiriman data ke server.

Pada sistem penelitian ini, sumber koneksi internet utama pada kapal disediakan melalui jaringan Starlink atau Wi-Fi kapal, yang digunakan sebagai jalur komunikasi utama bagi ESP32 untuk mengirimkan data hasil pembacaan sensor ke server melalui protokol MQTT. Namun, mengingat kondisi operasional kapal yang dapat mengalami gangguan konektivitas, baik akibat gangguan teknis pada perangkat Starlink maupun kondisi cuaca dan geografis yang memengaruhi kestabilan sinyal, sistem ini dilengkapi dengan SIM800L sebagai jalur komunikasi cadangan (*failover*).

Modul SIM800L akan aktif menggantikan fungsi pengiriman data hanya pada kondisi ketika koneksi Wi-Fi kapal maupun Starlink terdeteksi tidak berfungsi, sehingga data pemantauan konsentrasi gas CO dan NO₂ tetap dapat dikirimkan ke server melalui jaringan seluler operator, selama sinyal GSM/GPRS di lokasi kapal masih tersedia. Dengan adanya mekanisme cadangan ini, kontinuitas pengiriman data pemantauan dapat tetap terjaga meskipun jalur komunikasi utama mengalami gangguan, sehingga risiko hilangnya data pemantauan gas berbahaya akibat putusnya koneksi internet utama dapat diminimalkan.



Gambar 2.18 Modul GSM SIM800L
Sumber: <https://shorturl.at/lkm8X>

Tabel 2.17 Spesifikasi Modul GSM SIM800L

Parameter	Nilai / Keterangan
Chip utama	SIM800L
Jenis jaringan	GSM/GPRS 2G
Frekuensi	850 / 900 / 1800 / 1900 MHz
Tegangan kerja	3,4–4,4 V
Antarmuka	UART / TTL
SMS	Ya
Panggilan suara	Ya
Internet	Ya
Arus puncak	Hingga 2 A
Jenis SIM	microSIM

2.3.14 Software Arduino IDE

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak *open-source* yang dirancang untuk mengembangkan dan memprogram berbagai jenis mikrokontroler, seperti ESP8266, Arduino, dan ESP32. Arduino IDE dapat mengontrol dan membangun program pada *development board* (Kadir, 2017 dalam Ramadhani dkk., 2025). Karena antarmukanya yang sederhana serta dukungan ekosistem *library* yang luas, perangkat lunak ini telah menjadi platform utama untuk pemrograman sistem pemantauan berbasis ESP32. Melalui koneksi USB, pengguna dapat menuliskan program dalam bahasa pemrograman C/C++ dan mengunggahnya (*upload*) secara langsung ke perangkat keras mikrokontroler yang digunakan.

Pada penelitian ini, Arduino IDE digunakan untuk menulis program pembacaan data sensor, seperti MQ-135, MQ-7, dan SHT20, melakukan proses pengolahan data menggunakan algoritme berbasis aturan (*rule-based algorithm*), hingga eksekusi perintah keluaran ke perangkat seperti LCD, *buzzer*, dan modul pengendali aktif/nonaktif *exhaust fan* (LSA-002). Algoritme berbasis aturan ini bekerja dengan membandingkan nilai pembacaan sensor terhadap Nilai Ambang Batas (NAB) yang telah ditentukan melalui serangkaian pernyataan kondisional (*if-else*), sehingga sistem dapat mengambil keputusan secara otomatis dan deterministik, misalnya mengaktifkan *buzzer* dan *exhaust fan* ketika kadar gas CO atau NO₂ melampaui ambang batas aman. Proses pemrograman ini turut memanfaatkan sejumlah *library* tambahan, seperti *Wire* untuk komunikasi I2C, *LiquidCrystal* untuk antarmuka tampilan LCD, serta *library* khusus sensor lainnya,

guna mempercepat proses pengembangan sistem tanpa perlu menulis seluruh fungsi komunikasi perangkat keras dari awal.

Selain itu, fitur Serial Monitor yang tersedia pada Arduino IDE turut memberikan manfaat signifikan selama proses pengembangan sistem, khususnya dalam melakukan proses *debugging* serta memantau nilai pembacaan sensor secara *real-time* langsung melalui koneksi USB, sebelum data tersebut dikirimkan ke sistem pemantauan pusat melalui protokol MQTT. Kemampuan ini mempermudah proses identifikasi kesalahan program maupun anomali pembacaan sensor pada tahap pengembangan, sebelum sistem diimplementasikan secara penuh pada ruang *powerpack* kapal.



Gambar 2.19 Software Arduino IDE

Sumber : <https://apps.microsoft.com/detail/9nblggh4rsd8?hl=id-ID&gl=ID>

Tabel 2.18 Spesifikasi Software Arduino IDE

Parameter	Nilai / Keterangan
Jenis	Perangkat lunak open-source (IDE)
Fungsi	Menulis dan mengunggah program ke mikrokontroler
Bahasa Pemrograman	C/C++
Mikrokontroler Didukung	ESP32, ESP8266, Arduino
Koneksi	USB
Library Digunakan	Wire, LiquidCrystal, library sensor
Fitur Debugging	Serial Monitor
Jenis	Perangkat lunak open-source (IDE)
Fungsi	Menulis dan mengunggah program ke mikrokontroler
Bahasa Pemrograman	C/C++

2.3.15 Python

Python merupakan bahasa pemrograman tingkat tinggi (*high-level programming language*) yang bersifat interpretatif, dengan sintaks yang relatif sederhana dan mudah dipahami dibandingkan bahasa pemrograman tingkat rendah. Popularitas Python tidak hanya didukung oleh kemudahan mempelajarinya, tetapi juga oleh ekosistem pustaka (*library*) yang luas serta dukungan komunitas pengembang yang aktif, mulai dari kebutuhan proyek akademik hingga aplikasi

skala industri. Dalam bidang *Internet of Things* (IoT), pengolahan data, dan otomasi, Python menjadi pilihan yang umum digunakan karena ketersediaan pustaka siap pakai, termasuk modul komunikasi jaringan, pengolahan data sensor, serta dukungan terhadap protokol MQTT yang lazim digunakan pada sistem terdistribusi. Kombinasi karakteristik tersebut memungkinkan proses integrasi antara mikrokontroler seperti ESP32 dan komputer papan tunggal seperti Raspberry Pi dapat dilakukan tanpa memerlukan proses pengembangan yang rumit.

Pada penelitian ini, Python dijalankan pada Raspberry Pi 3 untuk menerima data hasil pembacaan sensor secara berkala melalui protokol MQTT, kemudian memproses dan menyimpan data tersebut ke dalam basis data MySQL untuk keperluan *data logging*. Selain itu, Python turut berperan dalam membangun *web interface* menggunakan *framework* Flask, yaitu *framework* web berbasis Python yang bersifat ringan (*lightweight*) dan sesuai untuk aplikasi web berskala kecil hingga menengah pada perangkat dengan sumber daya terbatas seperti Raspberry Pi. Melalui kombinasi Flask dan MySQL tersebut, data konsentrasi gas CO dan NO₂ yang telah diproses dapat ditampilkan secara *real-time* melalui peramban (*browser*) dalam bentuk visual seperti grafik, tabel, maupun indikator status kondisi lingkungan (aman/berbahaya), tanpa perlu mengakses langsung layar LCD pada panel di ruang *powerpack*. Selain menampilkan data, sistem berbasis Flask ini juga memicu pengiriman peringatan kepada pengguna ketika konsentrasi gas berbahaya yang tersimpan dalam basis data MySQL terdeteksi melewati batas aman yang telah ditentukan.



Gambar 2.20 Bahasa Python

Sumber: <https://niltechedu.com/blog/shop/courses/corporate-python/>

2.3.16 Software Visual Studio Code (VS Code)

Visual Studio Code (VS Code) merupakan editor kode sumber (*source code editor*) yang bersifat ringan, fleksibel, dan modern, yang banyak digunakan dalam berbagai jenis pengembangan perangkat lunak, termasuk pengembangan aplikasi berbasis web. VS Code mendukung berbagai bahasa pemrograman melalui sistem ekstensi (*extension*) yang dapat disesuaikan dengan kebutuhan pengembangan, seperti dukungan *syntax highlighting*, *auto-completion*, *linting*, hingga integrasi langsung dengan terminal dan sistem kontrol versi seperti Git, sehingga memudahkan proses pengembangan program secara efisien dalam satu antarmuka yang terintegrasi.

Pada penelitian ini, VS Code digunakan khusus untuk pengembangan web interface berbasis Python dan *framework* Flask, yang berjalan pada Raspberry Pi 3 sebagai antarmuka pemantauan pusat sistem. Melalui VS Code, proses penulisan kode backend (Flask), pengelolaan struktur basis data MySQL, serta penyusunan tampilan antarmuka pengguna (*front-end*) seperti HTML, CSS, dan JavaScript untuk *dashboard* pemantauan dapat dilakukan secara terintegrasi dalam satu lingkungan pengembangan. Dukungan ekstensi tambahan seperti Python *extension*, Prettier, maupun *Live Server* turut mempermudah proses pengujian tampilan web interface secara langsung selama tahap pengembangan, sebelum diterapkan secara penuh sebagai sistem pemantauan gas berbahaya pada ruang *powerpack* kapal. Dengan demikian, penggunaan VS Code dalam penelitian ini berperan sebagai lingkungan pengembangan (*development environment*) khusus untuk sisi perangkat lunak *server* dan antarmuka pengguna berbasis web, yang terpisah dari Arduino IDE yang digunakan untuk pemrograman firmware pada mikrokontroler ESP32.



Gambar 2.21 Software VS Code

Sumber : <https://apps.microsoft.com/detail/xp9kxm4bk9fz7q?hl=en-US&gl=US>

(Halaman ini sengaja dikosongkan)

BAB 3

METODE PENELITIAN

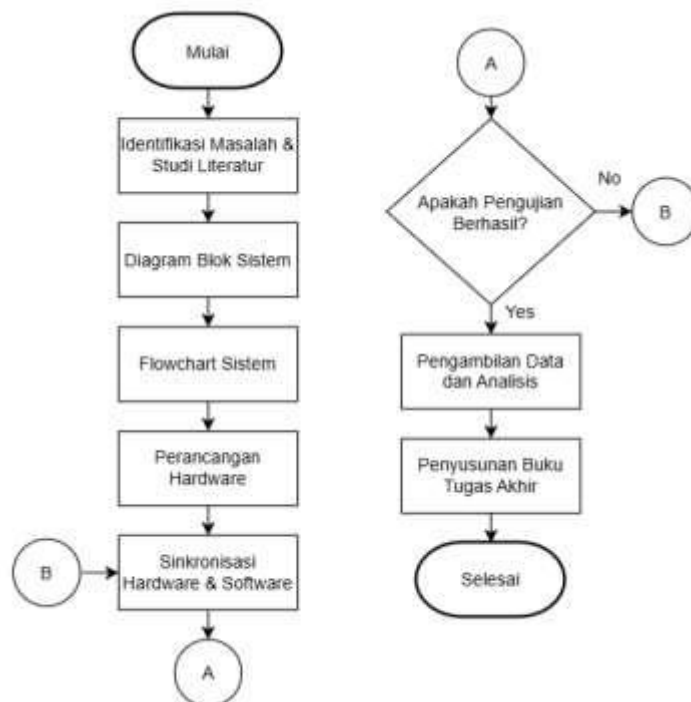
Pada Bab ini berisi alur penelitian yang akan dilakukan. Alur Penelitian ini sendiri berisi tentang konsep penelitian, tahapan penelitian, perencanaan, desain hardware dan *software*, pengimplementasian Wireless Sensor Network (WSN), *timeline* penelitian tugas akhir, serta rancangan anggaran pada penelitian.

3.1 Konsep Penelitian

Penelitian ini bertujuan untuk merancang serta mengembangkan sebuah sistem pemantauan gas berbahaya, khususnya pada gas berbahaya gas Karbon Monoksida (CO) dan Nitrogen Dioksida (NO₂), di ruang *powerpack* kapal untuk meningkatkan keselamatan kerja awak kapal. Proses Dimulai dengan penyusunan diagram blok sistem kemudian dilanjutkan dengan pembuatan Flowchart alur kerja sistem yang menjelaskan setiap langkah yang akan dilakukan dalam proses operasional sistem.

3.2 Tahapan Penelitian

Diagram alir pada Gambar 3.1 merupakan tahapan penelitian untuk menyelesaikan Tugas Akhir ini.



Gambar 3.1 Tahapan Penelitian

3.2.1 Identifikasi Masalah

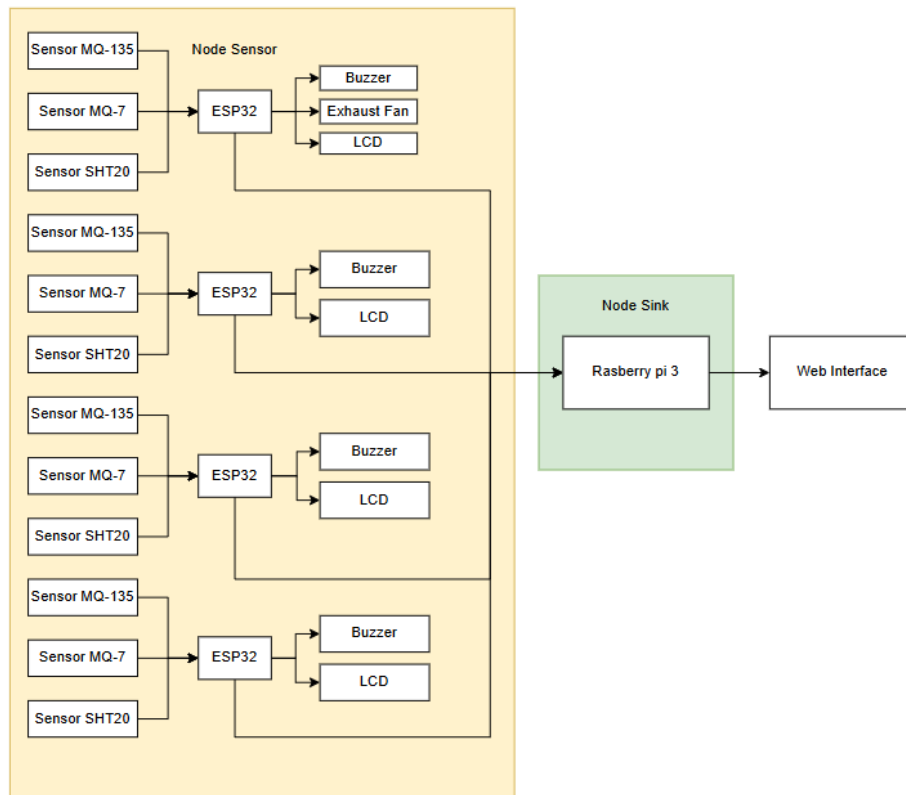
Langkah pertama dalam penelitian ini, memahami dengan lebih mendalam permasalahan yang ingin diteliti, dengan mengamati langsung kondisi lapangan yang menjadi fokus utama. Proses ini penting untuk mengetahui, seberapa besar perbedaan antara keadaan yang ada saat ini dengan tujuan atau kondisi ideal yang ingin dicapai. Barulah kita beralih ke kajian pustaka, yang tujuannya menggali lebih banyak solusi yang mungkin sudah diterapkan dalam penelitian-penelitian sebelumnya. Dalam sub bab 2.1, yang menjelaskan berbagai referensi terkait. Kajian pustaka ini bukan hanya mencari informasi dari jurnal ilmiah, tetapi juga dari buku-buku atau publikasi yang terbit dalam kurun waktu lima tahun terakhir, ini penting agar sumber yang digunakan masih relevan dan up to date.

3.2.2 Diagram Blok Sistem

langkah selanjutnya adalah membuat diagram blok untuk sistem kerja alat yang akan dirancang. Diagram Blok Sistem menggambarkan dengan jelas bagaimana masing-masing komponen utama dalam sistem saling terhubung dan berinteraksi. Diagram ini bukan hanya sekadar gambar, melainkan memberi gambaran yang lebih mendalam tentang aliran informasi dan bagaimana proses berjalan dalam sistem secara keseluruhan. Kita bisa mendapatkan gambaran yang lebih jelas tentang fungsi setiap komponen, serta bagaimana semuanya bekerja bersama-sama untuk mencapai tujuan yang diinginkan. Diagram ini sangat membantu untuk memetakan proses, karena memberikan visualisasi yang memudahkan pemahaman.

Diagram blok ini menggambarkan bagaimana sistem bekerja secara keseluruhan. Bagian Input, yang terdiri dari tiga sensor utama: Sensor MQ-135 mendeteksi gas Nitrogen Dioksida, Sensor MQ-7 berfungsi mendeteksi gas karbon monoksida, Sensor SHT20 mengukur suhu dan kelembapan. Ketiga sensor ini memberikan data yang akan diproses oleh ESP32, yang berada di tengah diagram. ESP32 bertugas sebagai otak sistem yang akan menerima data dari sensor, menganalisisnya, dan mengambil keputusan untuk mengendalikan perangkat output. Bagian Output, yang mencakup tiga komponen utama: *Exhaust Fan* untuk mengatur ventilasi, *Buzzer* untuk memberikan peringatan, dan LCD untuk menampilkan informasi terkait kondisi ruang *powerpack*. Semua komponen ini

bekerja bersama untuk menciptakan sistem yang dapat memantau dan merespons perubahan lingkungan ruang *powerpack* secara otomatis.



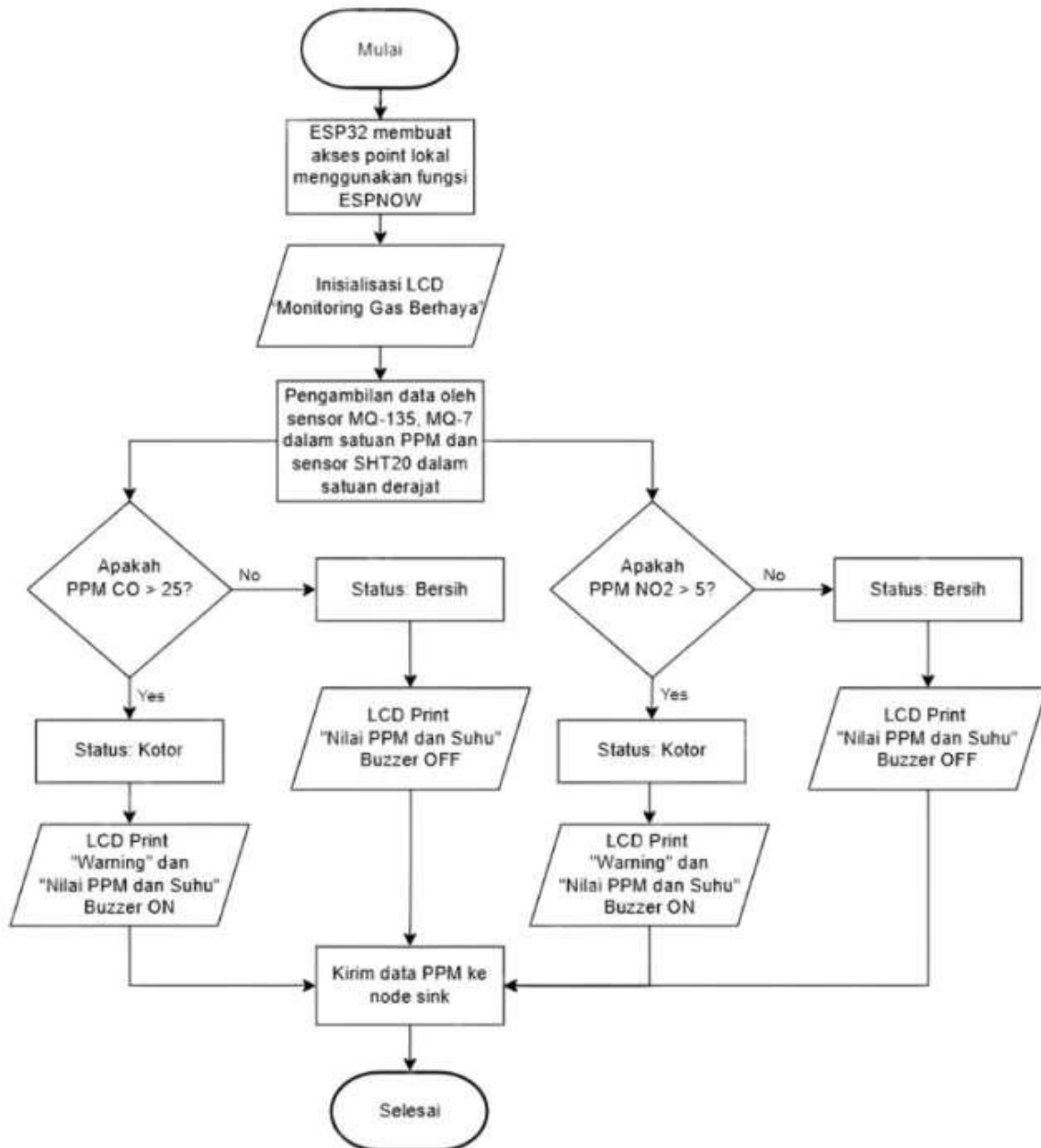
Gambar 3.2 Diagram Blok Sistem

3.2.3 Flowchart Sistem

Setelah diagram blok sistem, dilanjutkan dengan pembuatan Flowchart sistem dengan tujuan untuk mempermudah pemahaman tentang urutan operasi kerja sistem dengan menekankan setiap keputusan serta tindakan pada setiap tahap yang dilakukan sehingga juga dapat membantu dalam mendefinisikan logika operasional sistem secara menyeluruh.

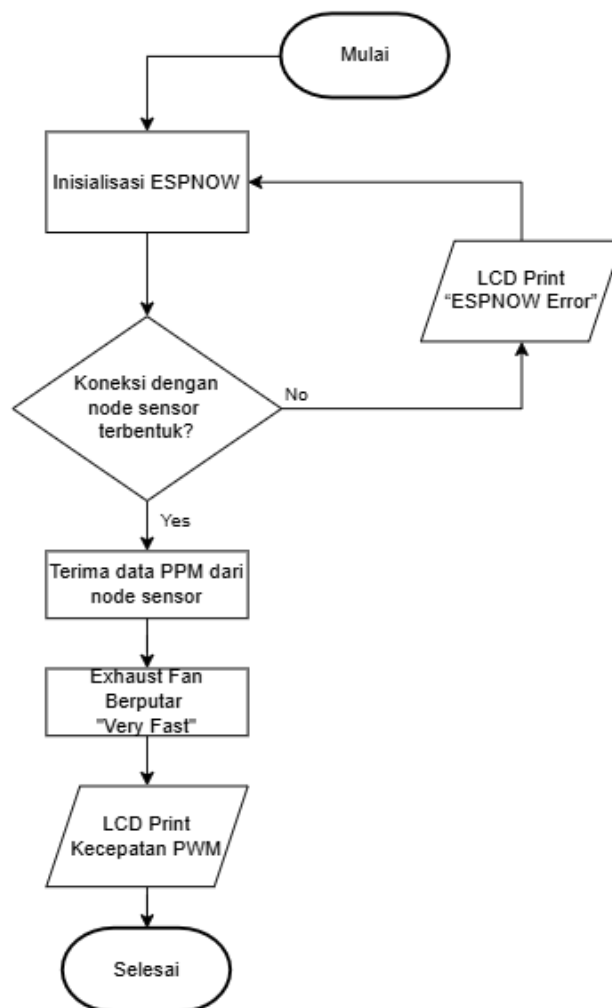
Flowchart *node sensor* menjelaskan, pertama ESP32 membuat akses point lokal menggunakan fungsi ESPNOW untuk memulai komunikasi. setelah akses terhubung, layar LCD menampilkan informasi seperti "*Monitoring Gas Berbahaya*" sebagai indikasi bahwa sistem sedang aktif. Ketiga sensor yaitu, sensor MQ-135, MQ-7 mendeteksi gas berbahaya dalam satuan PPM dan sensor SHT20 mendeteksi suhu ruangan *powerpack* dalam satuan derajat yang akan diolah menggunakan ESP32. Data PPM yang terkumpul akan dikirim ke *Node Sink* untuk diproses lebih lanjut lalu sistem memeriksa apakah nilai PPM CO lebih besar dari 25 dan nilai

PPM NO₂ lebih besar dari 5, sementara data suhu hanya akan ditampilkan di LCD. Jika Nilai PPM angkanya melebihi nilai ambang batas maka akan menyebabkan status sistem berubah menjadi "Kotor", dan nilai PPM akan ditampilkan pada LCD sementara *buzzer* aktif (peringatan). Sebaliknya, Nilai PPM di bawah ambang batas akan menyebabkan sistem menampilkan status "Bersih" pada layar dan *buzzer* dimatikan, yang berarti proses selesai.



Gambar 3.3 Flowchart Node Sensor

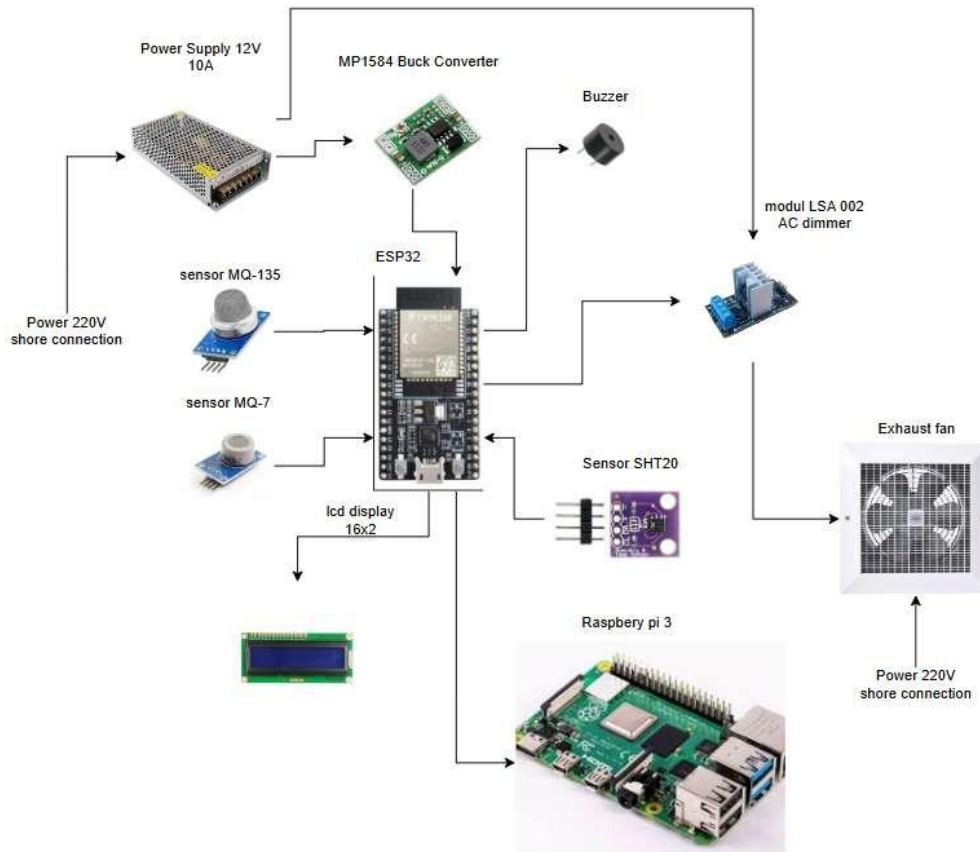
Flowchart node server menjelaskan sistem. Ini dimulai dengan inisialisasi ESPNOW, yang merupakan langkah pertama dalam membangun koneksi antara perangkat. Sistem akan memastikan terbentuknya koneksi dengan node sensor. Node server akan menerima data PPM, dan *exhaust fan* akan berputar dengan kecepatan tinggi (*very fast*) dengan nilai PWM sebesar 204 (2400 rpm) sesuai perintah node sensor setelah koneksi berhasil. Kemudian, kecepatan motor ini, yang dikendalikan oleh PWM dan ditampilkan pada LCD yang menunjukkan kecepatan motor secara *real-time*. Jika tidak ada koneksi, LCD akan menampilkan pesan "ESPNOW ERROR" sebagai indikasi bahwa sistem gagal terhubung. Alur sistem ditutup setelah semua proses selesai. Flowchart ini menunjukkan langkah-langkah operasional guna memastikan bahwa *node sink* menerima data dan bertindak sesuai dengan perintah node sink.



Gambar 3.4 Flowchart Node sink

3.2.4 Perancangan Hardware

Langkah selanjutnya yaitu perancangan hardware, Proses ini mencakup pemilihan dan pembuatan bagian fisik sistem seperti sensor, modul komunikasi, dan perangkat keras lainnya.



Gambar 3.5 Perancangan Hardware Panel Master

Perancangan hardware pada panel Master merupakan bagian inti dari sistem pemantauan gas berbahaya berbasis ESP32 dengan penggunaan WSN pada *powerpack*, yang berfungsi sebagai pusat pemrosesan data sekaligus titik kendali utama dalam jaringan sensor nirkabel. Diagram blok perancangan hardware panel Master ditunjukkan pada Gambar 3.5.

Sumber tegangan utama panel Master berasal dari listrik 220V shore connection, yang kemudian dikonversi menggunakan Power Supply 12V/10A dan diturunkan lebih lanjut oleh modul MP1584 *Buck converter* agar sesuai dengan kebutuhan tegangan kerja ESP32 dan komponen elektronik lainnya. ESP32 berperan sebagai mikrokontroler utama yang menerima data masukan dari sensor MQ-135 untuk mendeteksi konsentrasi gas NO₂, sensor MQ-7 untuk mendeteksi

konsentrasi gas CO, serta sensor SHT20 untuk memantau suhu di sekitar ruang *powerpack*. Data yang diterima kemudian diproses menggunakan *rule-based algorithm* untuk menentukan status kondisi udara, yang hasilnya ditampilkan secara langsung melalui LCD display 16x2 sebagai indikator visual di lokasi alat.

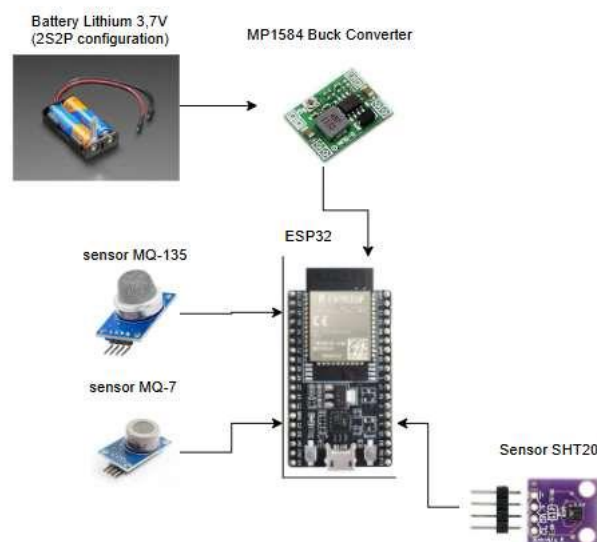
Sebagai bentuk peringatan dini, ESP32 mengaktifkan *buzzer* apabila status kondisi udara berada pada level Warning maupun Danger. Selain itu, ESP32 juga mengendalikan modul LSA 002 AC Dimmer yang berfungsi mengatur kecepatan putaran *exhaust fan* berdasarkan hasil klasifikasi status gas CO dan NO₂ yang terdeteksi, sehingga sirkulasi udara di ruang *powerpack* dapat ditingkatkan secara otomatis ketika konsentrasi gas berbahaya melebihi ambang batas yang ditentukan. *Exhaust fan* sendiri dicatu secara terpisah dari sumber listrik 220V shore connection, mengingat kebutuhan tegangan operasionalnya yang berbeda dari komponen elektronik lain pada sistem ini. Perlu ditegaskan bahwa pengendalian *exhaust fan* pada sistem ini murni berdasarkan hasil pembacaan konsentrasi gas, bukan berdasarkan suhu, sehingga sensor SHT20 tetap berfungsi sebatas pemantauan kondisi lingkungan sesuai dengan batasan penelitian yang telah ditetapkan pada sub-bab 1.3.

Selain komponen-komponen tersebut, panel Master juga dilengkapi dengan Raspberry Pi 3 yang berperan sebagai node server sekaligus data logger dalam jaringan WSN, sebagaimana telah dijelaskan pada BAB 2. Raspberry Pi 3 bertugas menerima, menyimpan, dan meneruskan data dari seluruh node (Master dan Slave) yang terhubung dalam jaringan, sehingga data hasil pemantauan gas dan suhu dari setiap titik dapat terpusat dan diakses melalui sistem pemantauan menggunakan protokol MQTT.

Selain komponen-komponen tersebut, panel Master juga dilengkapi dengan Raspberry Pi 3 yang berperan sebagai node server sekaligus data logger dalam jaringan WSN, sebagaimana telah dijelaskan pada BAB 2. Raspberry Pi 3 bertugas menerima, menyimpan, dan meneruskan data dari seluruh node (Master dan Slave) yang terhubung dalam jaringan, sehingga data hasil pemantauan gas dan suhu dari setiap titik dapat terpusat dan diakses melalui sistem pemantauan menggunakan protokol MQTT. Penempatan Raspberry Pi 3 pada panel Master ini didasari oleh pertimbangan bahwa panel Master merupakan titik pusat dari keseluruhan

arsitektur WSN, sehingga lebih efisien untuk menempatkan unit pengolahan data terpusat pada titik yang sama, dibandingkan menempatkannya secara terpisah pada masing-masing panel Slave.

Dengan adanya Raspberry Pi 3 sebagai node server, sistem tidak hanya mampu menampilkan status kondisi udara secara lokal melalui LCD display pada tiap panel, tetapi juga mampu mendokumentasikan riwayat data pembacaan sensor secara berkelanjutan dalam bentuk data logging. Riwayat data ini menjadi dasar bagi analisis dan evaluasi kinerja sistem sebagaimana telah dibahas pada BAB 4, sekaligus dapat dimanfaatkan sebagai bahan pertimbangan bagi operator maupun awak kapal dalam melakukan tindakan pencegahan lebih lanjut apabila ditemukan pola kenaikan konsentrasi gas yang berulang pada titik-titik tertentu di ruang *powerpack*.



Gambar 3.6 Perancangan Hardware Panel Slave

Berbeda dengan panel Master yang terhubung langsung dengan sumber listrik 220V shore connection, panel Slave dirancang menggunakan sumber tegangan mandiri berupa Baterai Lithium 3,7V dengan konfigurasi 2S2P (dua sel disusun seri dan dua susunan seri tersebut disusun paralel), sehingga panel Slave dapat beroperasi secara portabel tanpa bergantung pada jaringan listrik kapal secara langsung. Hal ini menjadikan panel Slave lebih fleksibel untuk ditempatkan pada berbagai titik di ruang *powerpack* sesuai kebutuhan cakupan area pemantauan dalam jaringan WSN. Tegangan keluaran dari baterai kemudian diturunkan

menggunakan modul MP1584 *Buck converter* agar sesuai dengan kebutuhan tegangan kerja ESP32 dan sensor-sensor yang terhubung.

ESP32 pada panel Slave berfungsi sebagai unit pemroses data lokal, yang menerima input dari sensor MQ-135 untuk mendeteksi konsentrasi gas NO₂, sensor MQ-7 untuk mendeteksi konsentrasi gas CO, serta sensor SHT20 untuk memantau suhu di sekitar titik pemasangan panel. Ketiga sensor ini bekerja dengan prinsip yang sama seperti pada panel Master, yaitu mengirimkan data hasil pembacaan langsung ke ESP32 untuk diproses menggunakan rule-based algorithm.

Berbeda dengan panel Master, panel Slave dirancang lebih sederhana tanpa dilengkapi LCD display, *buzzer*, modul AC dimmer, exhaust fan, maupun Raspberry Pi 3. Hasil pemrosesan data pada panel Slave selanjutnya dikirimkan menuju panel Master melalui jaringan WSN, sehingga seluruh keputusan notifikasi (*buzzer*), pengendalian exhaust fan, maupun penyimpanan data (data logging melalui Raspberry Pi 3) tetap terpusat pada panel Master. Rancangan ini bertujuan untuk menjaga efisiensi daya pada panel Slave yang menggunakan sumber baterai, mengingat komponen aktuator dan unit pengolahan data tambahan memerlukan konsumsi daya yang lebih besar dibandingkan unit sensor dan mikrokontroler saja.

3.2.5 Perancangan Software

Setelah mengetahui komponen apa saja yang dibutuhkan, kemudian dilakukan perancangan *software*. Fokus perancangan *software* adalah untuk membuat program, algoritma, dan kode untuk mengendalikan dan mengolah data dari perangkat keras.

3.2.6 Sinkronisasi Hardware dan Software

Pada tahap ini, memastikan bahwa perangkat keras (Hardware) dan perangkat lunak (*Software*) yang telah dirancang dapat saling berinteraksi dengan optimal. Proses sinkronisasi ini sangat penting agar semua komponen dalam sistem dapat berkomunikasi dengan lancar dan bekerja bersama secara harmonis untuk mencapai hasil penelitian yang diinginkan.

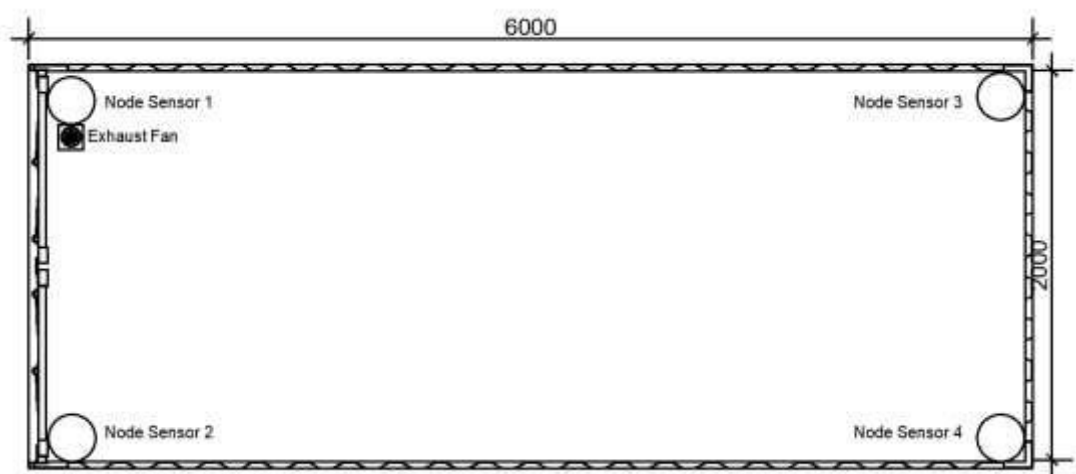
3.2.7 Pengujian Parsial

Selanjutnya dilakukan pengujian parsial guna memvalidasi sensor-sensor yang digunakan seperti sensor gas MQ-135 dan MQ-7 serta sensor suhu SHT20. Digunakan gas *analyzer* pada dua sensor gas tersebut sedangkan untuk sensor suhu

menggunakan termogun, dengan tujuan membuktikan bahwa data yang diambil oleh ketiga sensor tersebut mampu dinyatakan valid dan akurat.

3.2.8 Denah Pengujian

Tahap ini adalah bagian paling krusial dalam penelitian ini. Semua sistem yang sudah dirancang akan diuji secara komprehensif guna memastikan bahwa semua fungsi berjalan sesuai yang diinginkan. Tahap ini meliputi, uji fungsional, uji kinerja, serta deteksi masalah yang mungkin timbul untuk kemudian diperbaiki. Denah Pengujian dapat menunjukkan bagaimana sistem akan bekerja sehingga diharapkan dapat menyelesaikan masalah yang ada.



Gambar 3.7 Denah Pengujian

3.2.9 Pengambilan Data dan Analisis Data

Setelah pengujian, sistem berpotensi mengalami kegagalan maka akan dilakukan analisa untuk mengetahui penyebab terjadinya kegagalan tersebut dan melakukan perbaikan serta sinkronisasi alat hingga kriteria sistem telah tercapai. Tetapi ketika pengujian berhasil maka dilanjutkan pengambilan data serta analisis dari hasil data yang telah didapat.

3.2.10 Penyusunan Buku Tugas Akhir

Penyusunan buku tugas akhir adalah tahap paling akhir dari penelitian ini yang mencakup penyusunan laporan penelitian yang merangkum seluruh rangkaian proses yang telah dilalui dari awal hingga akhir. Laporan tersebut mencakup penjelasan rinci mengenai metodologi yang diterapkan, hasil dari pengujian yang dilakukan, analisis data, kesimpulan yang diperoleh, serta saran atau rekomendasi untuk pengembangan di masa depan.

3.3 Rancangan Perangkat Keras/Lunak

Analisis kebutuhan, desain hardware, dan pengembangan software yang saling mendukung adalah fokus utama pada tahap rancangan ini. Untuk memastikan bahwa sistem beroperasi sesuai dengan harapan, proses analisis kebutuhan dilakukan. Pemilihan komponen fisik seperti sensor dan mikrokontroler yang akan digunakan adalah bagian dari desain hardware. Fokusnya pada pengembangan algoritma, pengkodean, dan desain antarmuka pengguna yang ramah pengguna. Untuk membangun sistem yang responsif dan efektif, kedua komponen ini saling berhubungan dan bekerja sama. Kesuksesan sistem akan lebih terjamin dengan perencanaan yang tepat.

3.3.1 Analisis Kebutuhan Sistem

Pada tahap ini, analisis kebutuhan sistem dilakukan untuk menentukan spesifikasi perangkat keras dan perangkat lunak yang dibutuhkan dalam prototipe. Tujuan dari analisis ini adalah untuk mengidentifikasi semua kebutuhan teknis dan fungsional yang mendukung pencapaian tujuan penelitian. Proses ini melibatkan penentuan persyaratan yang harus dipenuhi, baik untuk perangkat keras maupun perangkat lunak, agar sistem berfungsi sesuai dengan yang diharapkan. Pada tahap ini, kebutuhan dipetakan sesuai dengan tujuan penelitian, termasuk pemilihan alat dan komponen yang tepat. Beberapa hal yang dianalisis adalah:

Tabel 3.1 Kebutuhan Sistem

No	Alat yang dibutuhkan	Jumlah
1	Sensor MQ-135	4 Unit
2	Sensor MQ-7	4 Unit
3	Sensor SHT20	4 Unit
4	Mikrokontroler ESP32	4 Unit
5	Raspberry pi 3	1 Unit
6	Battery Lithium 18650	4 Unit
7	Battery Lithium 18650 3 cell	4 Unit
8	Battery holder 1 cell	4 Unit
9	MP1584 Boost Converter	1 Unit
10	Modul Driver LSA-002	1 Unit
11	<i>Exhaust Fan</i>	1 Unit
12	LCD 20x4	1 Unit
13	<i>Buzzer</i>	1 Unit
14	<i>Box Panel 30x25x7</i>	1 Unit
16	<i>Box Panel 25x20x7</i>	3 Unit
15	Kabel konektor JST XH 20cm	12 Unit

3.3.2 Perhitungan Kebutuhan Sistem

Ruang *powerpack* pada penelitian ini berbentuk kontainer berukuran 20 feet yang digunakan sebagai tempat generator bantu (*auxiliary generator*) kapal. Selama proses operasionalnya, generator tersebut menghasilkan emisi gas sisa pembakaran mesin diesel, di antaranya karbon monoksida (CO) dan nitrogen dioksida (NO₂), yang apabila terakumulasi dalam ruang tertutup dapat mencapai konsentrasi yang membahayakan keselamatan awak kapal. Karakteristik ruang *powerpack* yang berbentuk kontainer tertutup dengan sirkulasi udara alami yang terbatas menjadikan potensi akumulasi gas ini perlu ditangani melalui sistem ventilasi mekanis, agar gas berbahaya yang terdeteksi oleh sistem sensor (MQ-7 dan MQ-135) dapat segera diencerkan dan dikeluarkan dari ruangan ketika kadarnya melebihi Nilai Ambang Batas (NAB) yang telah ditentukan. Untuk merancang sistem ventilasi tersebut, diperlukan perhitungan kebutuhan sistem secara bertahap, meliputi perhitungan dimensi ruang, kebutuhan debit udara, kebutuhan daya kipas, hingga pemilihan spesifikasi *exhaust fan* yang sesuai.

A. Perhitungan Dimensi dan Volume Ruang *Powerpack*

Perhitungan volume ruang *powerpack* diperlukan sebagai dasar dalam menentukan kebutuhan debit udara ventilasi pada sub-bab berikutnya, mengingat semakin besar volume ruangan, semakin besar pula kebutuhan aliran udara untuk mencapai tingkat pertukaran udara yang sama. Berdasarkan dimensi ruang *powerpack* yang telah disebutkan pada Bab 2 mengenai dasar teori *powerpack*, diperoleh data dimensi sebagai berikut:

Panjang (P) = 6 meter

Lebar (L) = 2 meter

Tinggi (T) = 2,5 meter

Sehingga volume ruang *powerpack* didapatkan sebagai berikut:

$$V = P \times L \times T$$

$$V = 6 \times 2 \times 2,5 = 30 \text{ m}^3$$

Volume ruang *powerpack* adalah 30 m³

B. Perhitungan Kebutuhan Debit Udara (Airflow)

Setelah volume ruangan diketahui, langkah selanjutnya adalah menghitung kebutuhan debit udara (*airflow rate*) yang harus dipenuhi oleh sistem ventilasi agar udara di dalam ruang *powerpack* dapat mengalami pertukaran secara memadai.

Perhitungan ini menggunakan pendekatan *Air Changes per Hour* (ACH), yaitu jumlah pergantian udara penuh yang terjadi dalam satu ruangan selama satu jam. Semakin tinggi nilai ACH yang digunakan, semakin sering udara di dalam ruangan mengalami pertukaran, sehingga semakin cepat pula konsentrasi gas berbahaya dapat diencerkan.

Pada penelitian ini, digunakan nilai ACH sebesar 15 kali/jam, yang merupakan nilai umum yang direkomendasikan untuk ruang mesin atau ruang generator dengan tingkat risiko akumulasi gas berbahaya, mengingat ruangan jenis ini memerlukan tingkat pertukaran udara yang lebih tinggi dibandingkan ruangan biasa (yang umumnya berkisar 4–6 ACH untuk ruang kerja standar). Kebutuhan debit udara dihitung menggunakan persamaan berikut:

$$Q = \text{ACH} \times V$$

$$Q = 15 \times 30 = 450 \text{ m}^3/\text{jam}$$

Untuk keperluan analisis dan konversi satuan lebih lanjut, nilai debit udara tersebut juga dinyatakan dalam satuan meter kubik per detik (m^3/s), dengan konversi sebagai berikut:

$$Q = \frac{450}{3600} = 0,125 \text{ m}^3/\text{detik}$$

Dengan demikian, sistem ventilasi ruang *powerpack* membutuhkan kapasitas aliran udara minimum sebesar $450 \text{ m}^3/\text{jam}$ (setara $0,125 \text{ m}^3/\text{detik}$ atau $7,5 \text{ m}^3/\text{menit}$) agar tingkat pertukaran udara sebesar 15 ACH dapat tercapai, sehingga konsentrasi gas CO dan NO_2 yang terakumulasi di dalam ruangan dapat diencerkan secara efektif.

C. Perhitungan Kebutuhan Daya Exhaust Fan

Setelah kebutuhan debit udara diketahui, selanjutnya dilakukan estimasi kebutuhan daya minimum (*power requirement*) yang harus dimiliki oleh *exhaust fan* agar mampu menghasilkan debit udara tersebut. Kebutuhan daya kipas secara teoritis dihitung menggunakan persamaan daya aerodinamis fan berikut:

$$P = \frac{Q \times \Delta P}{\eta}$$

Dimana:

- P = daya yang dibutuhkan (Watt)
- Q = debit udara (m^3/s)
- ΔP = tekanan statis (static pressure) yang harus diatasi kipas akibat hambatan saluran/ducting (Pascal, Pa)

- η = efisiensi mekanis kipas (dalam desimal, $0 < \eta \leq 1$)

1) Penentuan Nilai Tekanan Statis (ΔP)

Nilai tekanan statis pada penelitian ini mengacu pada standar (ANSI/ASHRAE Standard 62.2-2019 Addendum h, 2019), Section 7.1 untuk memenuhi tekanan statis minimum sebesar 0,25 inci air atau setara 62,5 Pa. Ketentuan ini sesuai dengan kondisi instalasi pada penelitian ini, di mana *exhaust fan* dipasang langsung pada panel master di ruang *powerpack*. Oleh karena itu, penelitian ini menetapkan nilai $\Delta P = 62,5$ Pa sebagai acuan perhitungan kebutuhan daya exhaust fan.

2) Penentuan Nilai Efisiensi Mekanis Kipas (η)

Exhaust fan yang digunakan pada penelitian ini merupakan jenis kipas aksial (*axial/propeller fan*) berukuran kecil-menengah dengan diameter blade 10 inch. Berdasarkan literatur teknik kipas industri, kipas jenis aksial pada umumnya memiliki efisiensi mekanis pada kisaran 40%–65%, bergantung pada desain sudut, kecepatan putar, dan kondisi operasionalnya. Mengacu pada rentang tersebut, penelitian ini menetapkan asumsi efisiensi sebesar $\eta = 0,5$ (50%), yang merepresentasikan nilai tengah (*mid-range*) yang wajar untuk kipas aksial berdaya kecil sekelas *exhaust fan* rumah tangga/komersial.

3) Perhitungan Kebutuhan Daya

Dengan nilai debit udara $Q = 0,125 \text{ m}^3/\text{s}$ yang telah diperoleh pada sub-bab B, serta nilai $\Delta P = 100 \text{ Pa}$ dan $\eta = 0,5$ yang telah dijustifikasi di atas, maka perhitungan kebutuhan daya adalah sebagai berikut:

$$P = \frac{Q \times \Delta P}{\eta} = \frac{0,125 \text{ m}^3/\text{s} \times 62,5 \text{ Pa}}{0,5}$$

$$P = \frac{7,8125}{0,5} = 15,625 \text{ Watt} = 0,0156 \text{ kW}$$

Berdasarkan hasil perhitungan tersebut, diperoleh kebutuhan daya minimum teoritis *exhaust fan* sebesar 15,6 Watt (0,0156 kW) agar mampu menghasilkan debit udara sebesar 450 m^3/jam pada kondisi hambatan saluran yang diasumsikan. Nilai daya minimum ini merupakan batas bawah teoritis yang dihitung berdasarkan asumsi kondisi ideal, di mana faktor-faktor seperti gesekan pada saluran udara, losses pada belokan pipa, serta efisiensi mekanis motor belum diperhitungkan secara penuh. Oleh karena itu, pada implementasi sistem nyata,

daya *exhaust fan* yang digunakan umumnya lebih besar dari nilai teoritis tersebut guna memberikan margin keamanan yang cukup agar sistem ventilasi tetap bekerja optimal meskipun terdapat hambatan tambahan yang tidak terduga di lapangan. Nilai ini selanjutnya digunakan sebagai acuan pembandingan terhadap spesifikasi daya *exhaust fan* aktual yang digunakan pada implementasi sistem, sebagaimana akan dibahas pada sub-bab berikutnya.

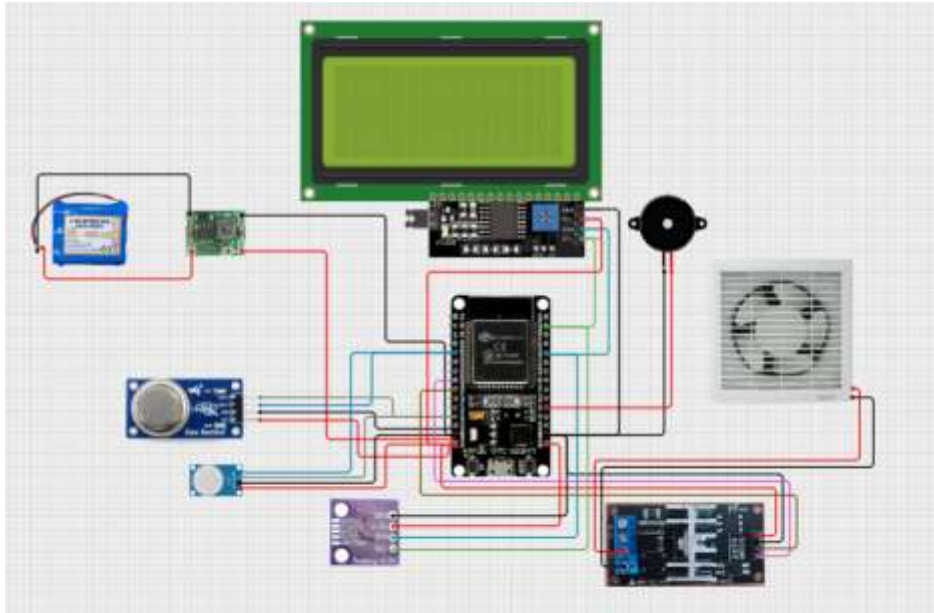
D. Analisis Kebutuhan dan Pemilihan Exhaust Fan

Nilai kebutuhan debit udara minimum sebesar 450 m³/jam dan kebutuhan daya minimum sebesar 15,6 Watt yang telah diperoleh pada sub-bab sebelumnya selanjutnya digunakan sebagai acuan dalam menentukan kesesuaian spesifikasi *exhaust fan* yang diimplementasikan pada sistem penelitian ini. Kedua nilai tersebut merepresentasikan batas minimum yang harus dipenuhi oleh perangkat ventilasi agar mampu menjaga sirkulasi udara ruang *powerpack* tetap sesuai dengan standar keselamatan yang telah ditetapkan, mengingat keterlambatan atau ketidakcukupan kapasitas ventilasi dapat berakibat pada terhambatnya proses pengenceran gas berbahaya hasil pembakaran mesin diesel generator bantu kapal.

Exhaust fan yang digunakan pada implementasi sistem penelitian ini adalah Maspion CEF-25, yaitu jenis ceiling *exhaust fan* dengan diameter blade 10 inch (250 mm). Pemilihan *exhaust fan* ini didasarkan pada beberapa pertimbangan utama, yaitu ketersediaan produk di pasaran yang memudahkan proses pengadaan dan penggantian komponen apabila diperlukan di kemudian hari, kemudahan instalasi pada struktur atap kontainer *powerpack* yang telah dirancang dengan bukaan sesuai ukuran standar ceiling *exhaust fan*, serta kesesuaian spesifikasi teknis perangkat terhadap kebutuhan sistem yang telah dihitung sebelumnya. Perlu dicatat bahwa daya aktual Maspion CEF-25 yang tercantum pada spesifikasi teknis berada di atas nilai kebutuhan daya minimum teoritis sebesar 15,6 Watt yang telah dihitung sebelumnya, sehingga *exhaust fan* ini memiliki margin daya yang cukup untuk tetap bekerja optimal meskipun terdapat hambatan udara tambahan di dalam ruang kontainer *powerpack*. Selain itu, *exhaust fan* jenis ini juga umum digunakan pada aplikasi ventilasi ruang tertutup skala kecil-menengah, sehingga keandalannya dalam mendukung sirkulasi udara pada ruang dengan volume serupa ruang *powerpack* kapal telah teruji secara luas di berbagai penerapan. Adapun spesifikasi

teknis lengkap *exhaust fan* tersebut telah disebutkan secara rinci pada Bab 2 sebagai bagian dari kajian komponen perangkat keras sistem.

3.3.3 Wiring Diagram Sistem



Gambar 3.8 Wiring Diagram Sistem

Perancangan wiring diagram pada sistem ini menghubungkan seluruh komponen perangkat keras ke mikrokontroler ESP32 dalam satu sistem monitoring dan kendali yang terintegrasi. Sistem ini menggunakan tiga jenis antarmuka, yaitu protokol I2C untuk komunikasi data digital, pin ADC untuk pembacaan sinyal analog sensor gas, dan pin GPIO digital untuk pengendalian aktuator.

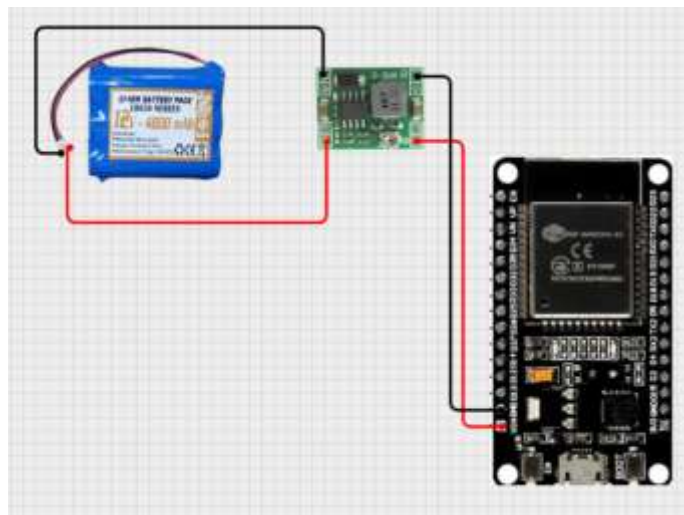
Sensor suhu dan kelembapan SHT20 mendapat suplai tegangan 3,3 V dari ESP32 dan berkomunikasi melalui protokol I2C menggunakan jalur SDA dan SCL. Sensor gas MQ-7 dan MQ-135 mendapat suplai tegangan 5 V dan menghasilkan keluaran sinyal analog yang terhubung ke pin ADC ESP32, di mana nilainya dikonversi menjadi data digital untuk selanjutnya diolah menjadi nilai konsentrasi gas dalam satuan PPM.

LCD 20×4 yang dilengkapi modul adapter I2C dihubungkan ke ESP32 melalui jalur SDA dan SCL yang sama dengan sensor SHT20, dan berfungsi menampilkan informasi suhu, kadar gas, serta status kondisi udara secara real-time. Buzzer dihubungkan ke pin GPIO digital ESP32 dan akan aktif secara otomatis sebagai alarm audio ketika kadar gas melampaui ambang batas yang telah ditetapkan. Modul driver LSA-002 dikendalikan melalui pin GPIO ESP32 dan

digunakan untuk menggerakkan Exhaust Fan, karena arus yang dibutuhkan kipas melebihi kapasitas arus keluaran pin GPIO ESP32 secara langsung. Distribusi daya sistem terbagi menjadi dua jalur, yaitu tegangan 3,3 V untuk sensor SHT20 dan komponen logika, serta tegangan 5 V dari sumber eksternal untuk sensor gas, LCD, buzzer, dan modul driver LSA-002.

3.3.4 Wiring diagram Supply Power ESP32

Wiring ini menghubungkan supply power untuk ESP32 yaitu battery lithium 18650 dengan kapasitas 12V. Namun ESP32 memerlukan daya sebesar 5V, maka digunakan modul MP1584 *buck converter* untuk *step-down* daya dari 12V ke 5V sehingga ESP32 dapat digunakan. Input battery dihubungkan pada pin VIN+ MP1584 lalu pin VOUT+ dihubungkan ke pin VIN 5V pada ESP32, pin output battery dihubungkan ke pin VIN- boost converter lalu untuk pin VOUT- dihubungkan ke pin GND ESP32.

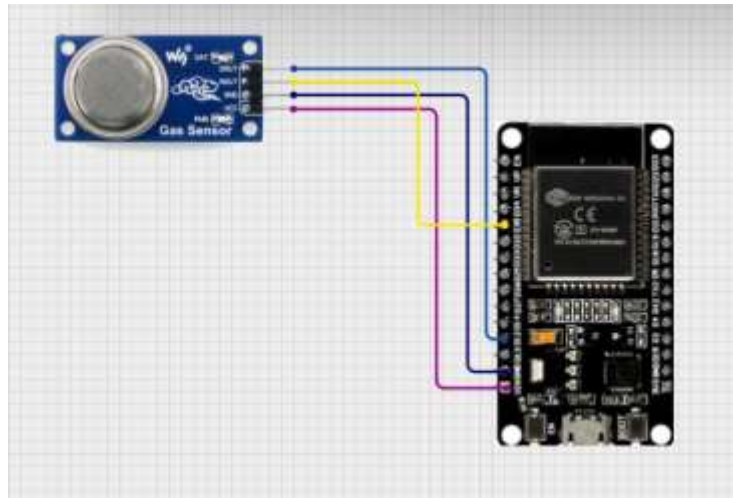


Gambar 3.9 Wiring Supply Power ESP32

3.3.5 Wiring diagram sensor MQ-135

Pada tugas akhir ini, sensor MQ-135 digunakan untuk membaca kadar gas Karbon Monoksida (CO) di udara, dengan Pin A sensor dihubungkan ke Pin 36 (ADC1) pada mikrokontroler ESP32 untuk membaca sinyal analog yang dihasilkan oleh sensor. Sensor ini membutuhkan sumber daya 5V, yang dapat disuplai melalui pin Vin pada ESP32 atau sumber daya eksternal. Setelah sensor mendeteksi kadar CO, data yang terbaca diproses oleh ESP32 dan dikirimkan ke receiver untuk dimonitor, baik melalui LCD atau sistem berbasis web, untuk pemantauan secara *real-time*. Wiring diagram dari sensor MQ-135 menunjukkan koneksi antara

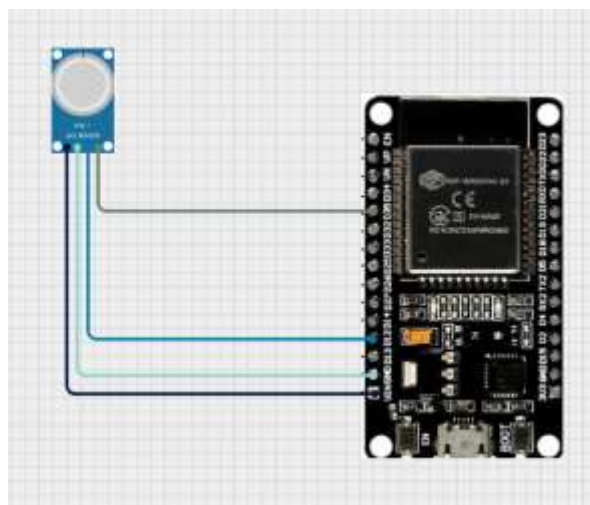
sensor, ESP32, serta sumber daya yang diperlukan untuk operasional sensor dan pengolahan data.



Gambar 3.10 Wiring diagram Sensor MQ-135

3.3.6 Wiring diagram sensor MQ-7

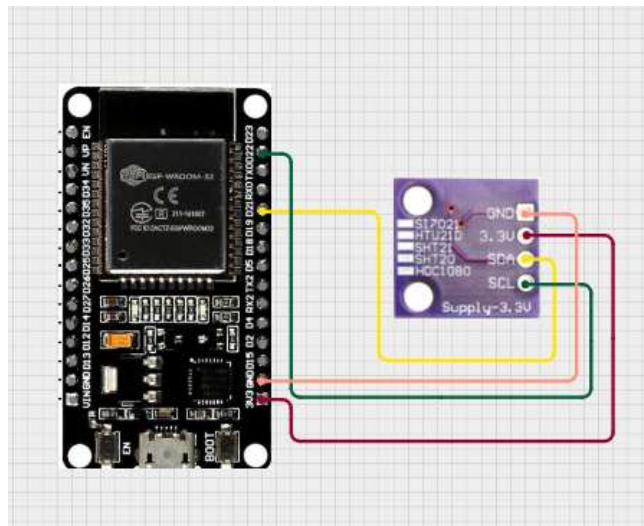
Sensor ini dihubungkan ke Pin A mikrokontroler ESP32, tepatnya ke Pin 36 (ADC1), untuk membaca sinyal analog yang dihasilkan oleh sensor saat terdeteksi gas CO. Sensor MQ-7 membutuhkan sumber daya 5V yang dapat dihubungkan ke pin Vin pada ESP32 atau *supply* daya eksternal. Setelah sensor mendeteksi kadar gas CO, nilai yang terbaca diproses oleh ESP32 dan dikirimkan ke receiver untuk ditampilkan pada layar LCD atau sistem berbasis web untuk pemantauan secara *real-time*. Wiring diagram dari sensor MQ-7 menunjukkan koneksi antara sensor, ESP32, dan sumber daya yang diperlukan untuk operasional sensor serta pengolahan dan transmisi data.



Gambar 3.11 Wiring diagram Sensor MQ-7

3.3.7 Wiring diagram Sensor SHT20

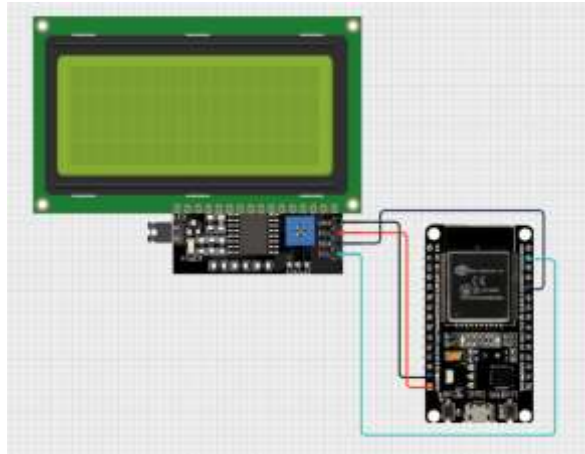
Sensor ini dihubungkan ke Pin GPIO pada mikrokontroler ESP32, misalnya Pin 4, untuk membaca sinyal digital yang dihasilkan oleh sensor. Sensor SHT20 membutuhkan sumber daya 3V, yang dapat disuplai melalui pin Vin pada ESP32 atau sumber daya eksternal. Setelah sensor mengukur suhu dan kelembapan, data yang terbaca akan diproses oleh ESP32 dan dikirimkan ke receiver untuk ditampilkan pada layar LCD atau sistem berbasis web untuk pemantauan secara *real-time*. Wiring diagram dari sensor SHT20 menunjukkan koneksi antara sensor, ESP32, serta sumber daya yang diperlukan untuk operasional sensor dan pengolahan data.



Gambar 3.12 Wiring diagram Sensor SHT20

3.3.8 Wiring diagram LCD 20x4

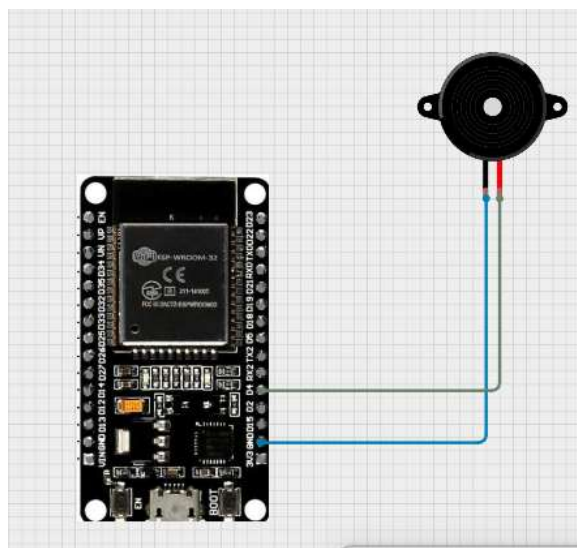
LCD 20x4 diperuntukan sebagai media tampilan guna menampilkan hasil pembacaan kadar gas CO dari sistem. LCD dihubungkan ke ESP32 melalui komunikasi I2C, sehingga hanya membutuhkan dua pin utama, yaitu SDA yang terhubung ke GPIO 21 dan SCL yang terhubung ke GPIO 22 pada ESP32. LCD mendapatkan suplai daya dari pin 5V dan GND ESP32 agar modul I2C dapat bekerja dengan stabil. Setelah ESP32 memproses data dari sensor gas, nilai kadar CO akan dikirimkan ke LCD untuk ditampilkan secara *real-time* sehingga pengguna dapat memantau kondisi udara dengan mudah. Wiring diagram LCD 20x4 menunjukkan hubungan antara modul I2C, jalur komunikasi SDA–SCL, serta koneksi daya ke ESP32 untuk memastikan sistem tampilan bekerja dengan baik.



Gambar 3.13 Wiring Diagram LCD

3.3.9 Wiring diagram *buzzer*

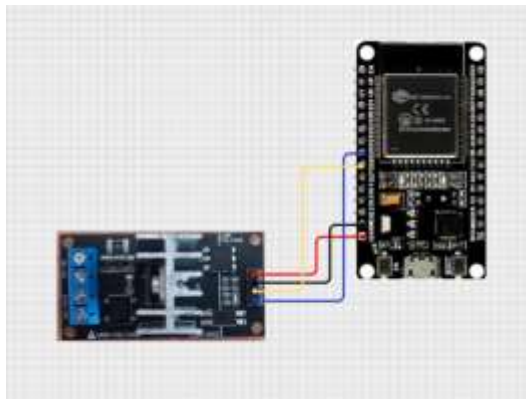
Pada tugas akhir ini, *buzzer* digunakan untuk memberikan peringatan dini jika kadar gas Karbon Monoksida (CO) yang terdeteksi oleh sensor melampaui ambang batas yang sudah ditentukan. *Buzzer* ini dihubungkan ke salah satu Pin GPIO pada mikrokontroler ESP32, misalnya Pin GPIO4, untuk menghasilkan suara alarm saat kadar CO berbahaya terdeteksi. *Buzzer* membutuhkan sumber daya 5V, yang dapat dihubungkan ke pin Vin pada ESP32 atau *supply* daya eksternal. Ketika sensor MQ-135 atau MQ-7 mendeteksi kadar CO yang tinggi, mikrokontroler akan memproses data tersebut dan mengaktifkan *buzzer*, memberi sinyal peringatan kepada pengguna. Wiring diagram *buzzer* menunjukkan hubungan antara *buzzer*, mikrokontroler ESP32, serta sumber daya yang diperlukan untuk pengoperasian *buzzer* dan pengiriman sinyal peringatan.



Gambar 3.14 Wiring Diagram *Buzzer*

3.3.10 Wiring diagram Modul Dimmer LSA-002

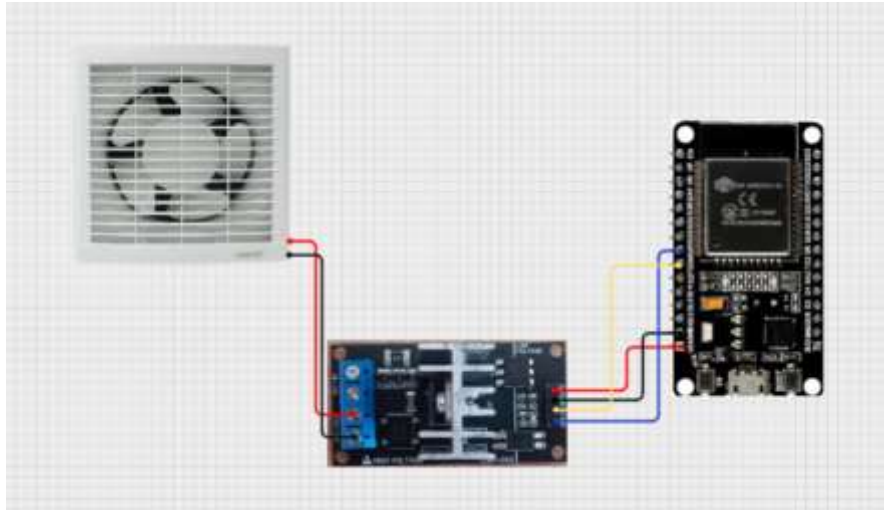
Modul LSA-002 digunakan untuk mengendalikan motor *Exhaust Fan* yang berfungsi mengatur sirkulasi udara di kamar mesin. Modul LSA-002 dihubungkan ke Pin GPIO pada mikrokontroler ESP32, seperti Pin GPIO25 dan Pin GPIO26, untuk mengatur arah dan kecepatan motor kipas menggunakan sinyal PWM (Pulse Width Modulation). Modul ini juga memerlukan sumber daya eksternal untuk motor kipas, yang biasanya disuplai melalui pin 5V atau 12V tergantung pada kebutuhan motor. Dengan mengontrol modul LSA-002, kecepatan *exhaust fan* dapat diatur dengan otomatis berdasar pada data yang telah diterima dari sensor gas CO dan NO2. Wiring diagram dari modul LSA-002 menunjukkan koneksi antara modul, ESP32, motor kipas, dan sumber daya yang dibutuhkan untuk operasional motor.



Gambar 3.15 Wiring Diagram Modul Dimmer LSA-002

3.3.11 Wiring diagram *Exhaust Fan*

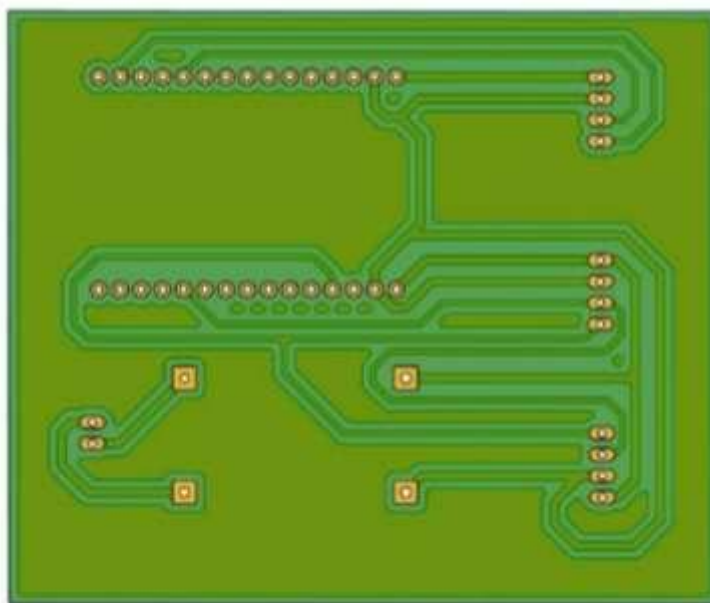
Pada tugas akhir ini, *Exhaust Fan* digunakan untuk mengatur sirkulasi udara di kamar mesin kapal, yang dikendalikan oleh mikrokontroler ESP32 melalui modul LSA-002. *Exhaust Fan* ini dihubungkan ke pin GPIO pada ESP32, misalnya Pin GPIO12 dan Pin GPIO14, untuk mengatur kecepatan dan arah putaran kipas menggunakan sinyal PWM (Pulse Width Modulation). Sumber daya untuk kipas ini biasanya disuplai melalui pin Vin pada ESP32 atau sumber daya eksternal, tergantung pada tegangan yang dibutuhkan oleh kipas. Kecepatan kipas ini dikendalikan berdasarkan pembacaan dari sensor gas CO dan NO2, dan akan diatur secara otomatis agar sesuai dengan kadar CO dan NO2 yang terdeteksi. Wiring diagram *Exhaust Fan* menunjukkan hubungan antara ESP32, modul LSA-002, motor kipas, dan sumber daya yang diperlukan untuk pengoperasian kipas.



Gambar 3.16 Wiring Diagram *Exhaust Fan*

3.3.12 Rancangan Desain PCB

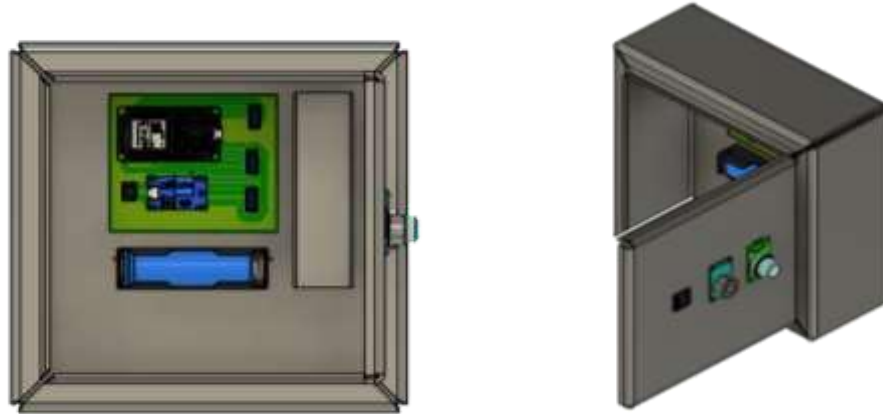
Bedasarkan rancangan wiring hardware yang telah dibuat, guna mengintegrasikan mikrokontroler ESP32 sebagai pusat kendali sistem dengan modul MP1584 *buck converter* dan terminal input untuk menghubungkan battery lithium 18650 3s sebagai *supply power*. Pada *design PCB* tersebut, juga terdapat 3 pin untuk menghubungkan *I2c* ketiga sensor seperti 2 pin untuk sensor gas MQ-135 dan MQ-7 serta 1 pin untuk sensor suhu SHT20, sehingga memudahkan integrasi dari ESP32 dengan ketiga sensor tersebut.



Gambar 3.17 *Design PCB*

3.3.13 Rancangan Desain Panel

Setelah merancang desain pcb, selanjutnya merancang desain untuk panel guna mengetahui tata letak (*layout*) dengan efisien pada *Box* panel.



Gambar 3.18 Desain Panel Node Sensor

3.3.14 Rancangan Software

Sub-bab ini menjelaskan tentang perencanaan software untuk menjalankan sistem dengan mengakuisisi data dari awal hingga dapat menampilkan pada LCD untuk kebutuhan *monitoring* pada ruang *powerpack*.



Gambar 3.19 Rancangan Software

Pada gambar 3.17 diatas menjelaskan rancangan software sistem. ESP32 menjalankan program utama yang dikembangkan menggunakan Arduino IDE, dengan Python membantu proses pengolahan data lanjutan. Data dari sensor MQ-7, MQ-135, dan SHT20 dibaca lalu diubah ke satuan ppm dan derajat suhu agar mudah dipantau. Berdasarkan nilai tersebut, ESP32 mengendalikan *Exhaust Fan* melalui modul driver LSA-002 serta mengaktifkan *buzzer* sebagai tanda peringatan

ketika kadar gas berbahaya melewati batas aman. Seluruh proses pemrograman dan pengelolaan proyek dilakukan menggunakan Visual Studio Code dengan ekstensi pendukung mikrokontroler, sementara protokol MQTT digunakan untuk mengirim data gas dan suhu ke komputer atau smartphone sehingga kondisi ruang *powerpack* dapat dipantau dari jarak jauh.

3.3.15 Rancangan Rule Base Algoritm

Pada penelitian ini digunakan rule base algoritm untuk pengambilan keputusan otomatis, yang diimplementasikan oleh ESP32 dengan mengkonversi nilai analog ke PPM berdasarkan datasheet sensor. Rule base terdiri dari 9 aturan *IF-THEN* yaitu:

1. R1 (Normal semua): IF (CO is Normal) AND (NO2 is Normal) THEN Exhaust_Fan = OFF (0); Aksi = LCD ("Aman"); *Buzzer* = 0.
2. R2 (Warning semua): IF (CO is Warning) AND (NO2 is Warning) THEN Exhaust_Fan = ON (1); Aksi = LCD ("Warning Semua"); *Buzzer* = 1; Kipas = ON.
3. R3 (Danger semua): IF (CO is Danger) AND (NO2 is Danger) THEN Exhaust_Fan = ON (1); Aksi = LCD ("Bahaya!"); *Buzzer* = 1; Kipas = ON.
4. R4 (CO Warning, NO2 Normal): IF (CO is Warning) AND (NO2 is Normal) THEN Exhaust_Fan = ON (1); Aksi = LCD ("CO Warning"); *Buzzer* = 1; Kipas = ON.
5. R5 (CO Normal, NO2 Warning): IF (CO is Normal) AND (NO2 is Warning) THEN Exhaust_Fan = ON (1); Aksi = LCD ("NO2 Warning"); *Buzzer* = 1; Kipas = ON.
6. R6 (CO Normal, NO2 Danger): IF (CO is Normal) AND (NO2 is Danger) THEN Exhaust_Fan = ON (1); Aksi = LCD ("NO2 Bahaya!"); *Buzzer* = 1; Kipas = ON.
7. R7 (CO Danger, NO2 Normal): IF (CO is Danger) AND (NO2 is Normal) THEN Exhaust_Fan = ON (1); Aksi = LCD ("CO Bahaya!"); *Buzzer* = 1; Kipas = ON.
8. R8 (CO Warning, NO2 Danger): IF (CO is Warning) AND (NO2 is Danger) THEN Exhaust_Fan = ON (1); Aksi = LCD ("CO Warning, NO2 Bahaya!"); *Buzzer* = 1; Kipas = ON.

Rule base juga dijelaskan pada tabel dibawah ini:

No	Kondisi CO (MQ-7)	Kondisi NO2 (MQ-135)	Output Status	Output Aksi
1	Normal (0–5 PPM)	Normal (0–0.5 PPM)	CO dan NO2 Normal	LCD: "Aman"; <i>Buzzer</i> OFF; Kipas OFF
2	Warning (10–25 PPM)	Warning (1–5 PPM)	CO dan NO2 Warning	LCD: "Warning Semua"; <i>Buzzer</i> ON; Kipas ON
3	Danger (25–35 PPM)	Danger (4.5–6 PPM)	CO dan NO2 Danger	LCD: "Bahaya!"; <i>Buzzer</i> ON; Kipas ON
4	Warning (10–25 PPM)	Normal (0–0.5 PPM)	CO Warning, NO2 Normal	LCD: "CO Warning"; <i>Buzzer</i> ON; Kipas ON
5	Normal (0–5 PPM)	Warning (1–5 PPM)	NO2 Warning, CO Normal	LCD: "NO2 Warning"; <i>Buzzer</i> ON; Kipas ON
6	Normal (0–5 PPM)	Danger (4.5–6 PPM)	NO2 Danger, CO Normal	LCD: "NO2 Bahaya!"; <i>Buzzer</i> ON; Kipas ON
7	Danger (25–35 PPM)	Normal (0–0.5 PPM)	CO Danger, NO2 Normal	LCD: "CO Bahaya!"; <i>Buzzer</i> ON; Kipas ON
8	Warning (10–25 PPM)	Danger (4.5–6 PPM)	CO Warning, NO2 Danger	LCD: "CO Warning, NO2 Bahaya!"; <i>Buzzer</i> ON; Kipas ON
9	Danger (25–35 PPM)	Warning (1–5 PPM)	CO Danger, NO2 Warning	LCD: "CO Bahaya!, NO2 Warning"; <i>Buzzer</i> ON; Kipas ON

(Halaman ini sengaja dikosongkan)

BAB 4

HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan hasil pengukuran dan pengujian parameter, serta menguraikan pengujian setiap komponen yang digunakan dalam Tugas Akhir ini. Pengujian meliputi hasil pembacaan sensor dan komponen lainnya, dengan tujuan memastikan bahwa setiap komponen berfungsi sesuai dengan rencana yang telah dibuat.

4.1 Implementasi Mekanik

Pengujian mekanik pada sistem pemantauan gas berbahaya ini bertujuan memastikan seluruh komponen terpasang rapi, kokoh, dan aman di dalam panel serta ruang *powerpack*. Kegiatan ini meliputi pemasangan ESP32, sensor MQ-7, MQ-135, sensor SHT20, LCD 20x4, *buzzer*, Raspberry pi 3, modul *driver* LSA-002, *exhaust fan*, rangkaian catu daya berbasis baterai 18650 dan modul step down, serta penataan kabel pada *box* panel sesuai rancangan desain panel dan wiring diagram.

4.1.1 Desain dan Realisasi *Box* Panel

Tahap ini berfokus pada realisasi pemasangan seluruh komponen sistem pemantauan gas berbahaya ke dalam *Box* panel. Tujuan utama realisasi *box* panel adalah menyusun node sensor dan node server beserta rangkaian pendukungnya agar lebih rapi, aman, dan mudah diakses saat proses pengujian maupun pengoperasian. Komponen yang dipasang pada panel master meliputi PCB, sensor MQ-135, sensor MQ-7, sensor SHT20, LCD 20x4, *buzzer*, modul *driver* LSA-002, *exhaust fan*, rangkaian catu daya berbasis baterai 18650, modul konverter tegangan serta raspberry pi 3.



Gambar 4.1 Realisasi *Box* Panel Master

Penempatan komponen di dalam *Box* panel mengikuti rancangan desain panel dan wiring diagram yang telah dibuat sebelumnya. Bagian sensor dan rangkaian logika disusun terpisah dengan jalur catu daya dan driver beban untuk mengurangi gangguan dan mempermudah penelusuran saat perawatan. Node sensor ditempatkan pada posisi yang memudahkan koneksi ke sensor di ruang *powerpack*, sedangkan node server ditempatkan pada area yang mudah dijangkau untuk *monitoring* dan konfigurasi. Jalur kabel diatur dengan pengikat dan penandaan yang jelas untuk mencegah kabel longgar dan meminimalkan risiko korsleting. Sedangkan pada panel slave, komponen yang dipasang hanya PCB, serta 3 sensor meliputi sensor MQ-135, sensor MQ-7 dan sensor SHT20.



Gambar 4.2 Realisasi *Box* Panel Slave

Berdasarkan realisasi yang dilakukan, seluruh komponen dapat terpasang dan terhubung sesuai rancangan. Tata letak di dalam *box* panel dibuat sesederhana mungkin agar memudahkan identifikasi komponen, proses *troubleshooting*, serta pengembangan sistem pada tahap berikutnya. Penggunaan *box* panel juga membantu melindungi rangkaian dari gangguan mekanik dan lingkungan, sehingga mendukung keandalan sistem pemantauan gas berbasis ESP32 yang dirancang.

4.2 Pengujian Parsial

Sub bab ini menjelaskan tentang pengujian melalui simulasi dan pengujian parsial untuk menghindari kesalahan sebelum merakit alat yang digunakan dalam Tugas Akhir ini, beberapa simulasi perlu dilakukan terlebih dahulu, di antaranya:

4.2.1 Pengujian Parsial PCB

Uji parsial dilakukan pada rangkaian catu daya PCB untuk memastikan bahwa tegangan masukan dari baterai dan tegangan keluaran *buck converter* sesuai dengan nilai rancangan. Pengujian difokuskan pada jalur catu daya utama yang menyuplai modul ESP32.



Gambar 4.3 Pengujian Parsial Input PCB

Pengujian dilakukan dengan memberikan tegangan masukan dari baterai sebesar 8 V yang dihubungkan ke terminal input rangkaian catu daya pada PCB. Nilai tegangan masukan kemudian diukur menggunakan multimeter digital pada titik uji (test point) input atau langsung pada terminal baterai di PCB. Hasil pengukuran menunjukkan tegangan sebesar 8 V, sehingga dapat disimpulkan bahwa jalur input dari baterai menuju rangkaian *buck converter* berfungsi dengan baik dan tidak terdapat penurunan tegangan yang signifikan.



Gambar 4.4 Pengujian Parsial Output Buck Converter

Selanjutnya dilakukan pengujian pada keluaran *buck converter* yang berfungsi menurunkan tegangan 8 V menjadi 5 V untuk mensuplai modul ESP32. Pengukuran dilakukan dengan multimeter pada titik uji tegangan 5 V atau pada pin suplai 5 V yang terhubung ke ESP32. Hasil pengukuran menunjukkan tegangan keluaran sebesar 5 V. Nilai ini sesuai dengan tegangan yang dirancang pada rangkaian *buck converter* dan berada dalam toleransi yang dapat diterima untuk catu daya ESP32.

4.2.2 Pengujian Parsial Sensor MQ-7

Pengujian sensor MQ-7 bertujuan untuk memastikan bahwa sensor mampu membaca perubahan konsentrasi gas CO dan mengeluarkan nilai tegangan analog yang stabil pada pin keluaran, yang kemudian dikonversi menjadi nilai ADC oleh ESP32. Pengujian dilakukan dengan menghubungkan sensor MQ-7 ke pin ADC ESP32 sesuai wiring diagram, kemudian membaca nilai ADC dan mengamati perubahan pembacaan saat diberikan paparan gas CO dari sumber terkontrol.



Gambar 4.5 Pengujian Parsial Sensor MQ-7

Langkah pengujian yang dilakukan adalah sebagai berikut. Pertama, sensor dinyalakan dan dibiarkan melakukan pemanasan awal hingga kondisi stabil sesuai rekomendasi datasheet. Kedua, dilakukan pembacaan nilai ADC pada kondisi udara ruangan tanpa sumber CO tambahan sebagai referensi awal untuk menentukan nilai R_0 . Ketiga, sensor kemudian diberi paparan gas CO secara bertahap dengan bantuan *gas analyzer* sehingga diperoleh kombinasi data R_s/R_0 dan ppm CO sebagai dasar kalibrasi. Pada setiap tahap konsentrasi CO, dilakukan pencatatan

nilai ADC, perhitungan R_s , rasio R_s/R_0 , serta nilai ppm CO dari *gas analyzer* untuk memastikan hubungan keduanya konsisten. Dilakukan pengambilan data sensor sebanyak 10 kali untuk masing masing sensor MQ-7, dan hasil pengujian menunjukkan bahwa nilai ADC dan rasio R_s/R_0 berubah mengikuti kenaikan ppm CO, sehingga dapat digunakan untuk membentuk persamaan konversi ppm pada bab perhitungan kalibrasi sensor. Berikut adalah tabel pengujian 4 sensor MQ-7 yang dilakukan dengan pengambilan data sebanyak masing masing 10 kali sebagai dasar analisis karakteristik dan penentuan parameter kalibrasi.

Tabel 4.1 Pengujian Sensor MQ-7 nomor 1

No	R_s/R_0	PPM CO (<i>Analyzer</i>)	PPM CO (Sensor)	Error (%)
1	2,649	8	8,07	0,86
2	1,595	20	20,28	1,40
3	0,945	16	15,91	0,53
4	0,945	15	15,06	0,39
5	1,505	33	33,40	1,20
6	1,505	33	33,11	0,33
7	1,486	30	29,54	1,55
8	1,352	27	26,85	0,57
9	1,209	29	29,22	0,76
10	1,343	30	29,72	0,95
Rata-rata	1,453	24,1	24,11	0,85

Tabel pengujian 4.1 diatas menunjukkan bahwa hubungan antara rasio R_s/R_0 dan konsentrasi CO yang diukur oleh *analyzer* sudah tercermin dengan baik pada hasil pembacaan sensor setelah kalibrasi. Nilai R_s/R_0 berada pada rentang sekitar 0,945 hingga 2,649, dengan konsentrasi CO *analyzer* antara 8 ppm sampai 33 ppm, sehingga mencakup kondisi rendah sampai sedang sesuai skenario pengujian. Nilai ppm CO sensor hasil kalibrasi selalu sangat dekat dengan nilai *analyzer*, misalnya pada 8 ppm terbaca 8,07 ppm, pada 20 ppm terbaca 20,28 ppm, dan pada 33 ppm berkisar di sekitar 33 ppm, yang menunjukkan konsistensi respon sensor terhadap variasi kadar CO. Persentase error tiap titik juga relatif kecil, berada di kisaran sekitar 0,33% sampai 1,55%, dengan rata rata error hanya 0,85%, sehingga masih jauh di bawah batas toleransi yang umum digunakan untuk sistem pemantauan kualitas udara. Selain itu, rata rata ppm CO *analyzer* sebesar 24,1 ppm dan rata rata ppm CO sensor sebesar 24,11 ppm praktis identik, yang mengindikasikan bahwa sistem kalibrasi tidak menimbulkan bias yang signifikan terhadap hasil pengukuran.

Tabel 4.2 Pengujian Sensor MQ-7 nomor 2

No	Rs/R0	PPM CO (<i>Analyzer</i>)	PPM CO (Sensor)	Error (%)
1	2,649	8	7,9	1,25
2	2,600	8	8,1	1,25
3	1,315	36	35,6	1,11
4	1,280	36	36,5	1,39
5	1,138	57	56,2	1,40
6	1,100	57	57,8	1,40
7	1,500	30	29,5	1,67
8	1,800	20	19,7	1,50
9	0,900	70	69,0	1,43
10	0,800	80	79,0	1,25
Rata-rata	1,508	40	39,93	1,37

Tabel pengujian 4.2 diatas menunjukkan bahwa sensor sudah memiliki performa kalibrasi yang konsisten pada rentang konsentrasi CO yang cukup lebar, yaitu dari 8 ppm hingga 80 ppm. Nilai Rs/R0 bervariasi antara sekitar 0,800 hingga 2,649, dan perubahan rasio ini diikuti oleh perubahan nilai ppm CO hasil kalibrasi yang sangat dekat dengan nilai acuan *analyzer*, misalnya pada 8 ppm terbaca 7,9–8,1 ppm, pada 36 ppm terbaca 35,6–36,5 ppm, dan pada 80 ppm terbaca 79,0 ppm. Persentase error di setiap titik berada dalam kisaran sempit sekitar 1,11% sampai 1,67%, dengan rata rata error sebesar 1,37%, sehingga masih berada di bawah batas toleransi yang biasanya dipakai pada pengujian kalibrasi sensor gas. Rata rata nilai Rs/R0 sebesar 1,5082 menunjukkan titik kerja sensor pada kondisi uji ini, sementara rata rata ppm CO *analyzer* sebesar 40 ppm dan rata rata ppm CO sensor sebesar 39,93 ppm hampir sama, yang mengindikasikan bahwa kalibrasi tidak menimbulkan bias yang berarti terhadap pengukuran.

Tabel 4.3 Pengujian Sensor MQ-7 nomor 3

No	Rs/R0	PPM CO <i>analyzer</i>	PPM CO (sensor)	Error (%)
1	0,944	15	14,75	1,67
2	0,946	15	15,20	1,33
3	1,590	7	6,88	1,71
4	1,595	7	7,12	1,71
5	1,504	7,75	7,60	1,94
6	1,504	7,75	7,90	1,94
7	1,500	10	9,85	1,50
8	1,500	10	10,15	1,50
9	1,300	12	11,80	1,67
10	1,300	12	12,20	1,67
Rata-rata	1,419	10,65	10,75	1,66

Tabel pengujian 4.3 diatas menunjukkan kinerja sensor pada rentang konsentrasi CO yang relatif rendah, sekitar 7 ppm sampai 15 ppm. Nilai Rs/R0 berada di kisaran 0,9446 sampai 1,5950, dan setiap perubahan rasio ini diikuti oleh perubahan bacaan ppm CO sensor yang sangat dekat dengan nilai acuan *analyzer*. Contohnya, pada konsentrasi 15 ppm, sensor terbaca 14,75 ppm dan 15,20 ppm, sedangkan pada sekitar 7–7,75 ppm sensor terbaca antara 6,88 ppm hingga 7,90 ppm. Persentase error setiap titik berada antara 1,33% sampai 1,94%, dengan rata rata error sebesar 1,66%, sehingga seluruh hasil pengukuran masih memenuhi batas toleransi. Rata rata Rs/R0 sebesar 1,4198 menggambarkan titik kerja sensor pada pengujian ini, sementara rata rata ppm CO *analyzer* sebesar 10,65 ppm dan rata rata ppm CO sensor sebesar 10,75 ppm sangat berdekatan, yang menunjukkan tidak terdapat bias signifikan antara alat pembanding dan sensor setelah kalibrasi.

Tabel 4.4 Pengujian Sensor MQ-7 nomor 4

No	Rs/R0	PPM CO <i>analyzer</i>	PPM CO (sensor)	Error (%)
1	0,596	15	14,70	2,00
2	0,589	15	15,20	1,33
3	0,753	15	14,80	1,33
4	0,532	15	15,10	0,67
5	0,813	12	11,80	1,67
6	0,819	12	12,10	0,83
7	0,700	14	13,75	1,79
8	0,651	13	12,80	1,54
9	0,620	16	15,75	1,56
10	0,600	17	16,70	1,76
Rata-rata	0,6883	14,40	14,27	1,45

Tabel pengujian 4.5 tersebut menggambarkan kinerja sensor pada rentang konsentrasi CO sekitar 12 ppm hingga 17 ppm. Nilai Rs/R0 berada pada kisaran 0,532 sampai 0,819, dan perubahan rasio ini diikuti oleh perubahan pembacaan ppm CO sensor yang mendekati nilai acuan *analyzer*. Sebagai contoh, pada konsentrasi 15 ppm, sensor membaca antara 14,70 ppm hingga 15,20 ppm, sedangkan pada 12 ppm terbaca 11,80 ppm dan 12,10 ppm, dan pada 17 ppm terbaca 16,70 ppm. Persentase error tiap titik berada di rentang sekitar 0,67% hingga 2,00%, dengan rata rata error sebesar 1,45%, sehingga masih memenuhi batas toleransi. Rata rata Rs/R0 sebesar 0,6883 menunjukkan karakteristik titik kerja sensor pada kelompok pengujian ini, sedangkan rata rata ppm CO *analyzer* sebesar 14,40 ppm dan rata

rata ppm CO sensor sebesar 14,27 ppm sangat berdekatan, yang menunjukkan tidak ada bias signifikan antara hasil kalibrasi sensor dan alat pembanding.

4.2.3 Pengujian Parsial Sensor MQ-135

Pengujian sensor MQ-135 dilakukan untuk memverifikasi kemampuan sensor dalam mendeteksi gas NO₂ (dan gas lain yang relevan) di ruang *powerpack*. Sensor dihubungkan ke pin ADC ESP32 sebagaimana ditunjukkan pada wiring diagram dan diberi suplai tegangan sesuai spesifikasi. Pengujian awal dilakukan di udara ruangan untuk memperoleh nilai R_s udara bersih yang akan digunakan sebagai R_0 pada proses kalibrasi.

Setelah itu, sensor diuji dengan beberapa skenario konsentrasi gas menggunakan *gas analyzer* khusus NO₂ untuk mendapatkan karakteristik respon yang lebih rinci. Pada setiap skenario, sumber gas NO₂ diatur pada konsentrasi tertentu kemudian dibiarkan stabil terlebih dahulu agar kondisi di dalam ruang uji seragam. Setelah kondisi stabil tercapai, nilai keluaran sensor diambil secara bertahap dalam bentuk data ADC, lalu dikonversi menjadi tegangan VRL, resistansi R_s , dan rasio R_s/R_0 . Nilai rasio ini kemudian dicatat berdampingan dengan nilai konsentrasi NO₂ (ppm) yang terbaca oleh *analyzer* sebagai acuan referensi.



Gambar 4.6 Pengujian Parsial Sensor MQ-135

Pengambilan data dilakukan pada beberapa titik konsentrasi yang berbeda, mulai dari konsentrasi rendah hingga mendekati batas atas rentang kerja sensor, sehingga kurva hubungan antara R_s/R_0 dan ppm NO₂ dapat terpetakan dengan baik. Untuk setiap titik konsentrasi, pengukuran diulang beberapa kali guna memastikan bahwa nilai yang diperoleh bersifat konsisten dan tidak hanya dipengaruhi oleh noise sesaat. Berikut Tabel 10 kali percobaan untuk kalibrasi sensor MQ-135

Tabel 4.5 Pengujian Sensor MQ-135 nomor 1

No	Rs/R0	ppm NO ₂ analyzer	ppm NO ₂ kalibrasi	Error (%)
1	1,600	3,65	3,61	1,10
2	1,600	3,65	3,69	1,10
3	1,284	2,15	2,13	0,93
4	1,284	2,15	2,17	0,93
5	1,315	4,10	4,05	1,22
6	1,315	4,10	4,15	1,22
7	1,450	3,00	2,97	1,00
8	1,350	2,50	2,48	0,80
9	1,250	3,20	3,16	1,25
10	1,200	2,80	2,77	1,07
Rata-rata	1,3648	3,13	3,12	1,03

Berdasarkan tabel pengujian 4.6, nilai Rs/R0 sensor NO₂ berada pada rentang sekitar 1,200 sampai 1,600 dengan konsentrasi NO₂ analyzer antara 2,15 ppm hingga 4,10 ppm, dan hasil kalibrasi sensor mengikuti nilai acuan tersebut dengan sangat dekat. Selisih antara pembacaan sensor terkalibrasi dan alat analyzer menghasilkan error yang konsisten kecil, yaitu sekitar 0,80% sampai 22% dengan rata-rata sekitar 1,03%, sehingga masih berada di bawah batas toleransi yang ditetapkan. Pola ini menunjukkan bahwa proses kalibrasi berhasil menyesuaikan respon sensor terhadap variasi Rs/R0 sehingga pembacaan ppm NO₂ yang dihasilkan sudah representatif terhadap nilai referensi dan layak digunakan untuk pemantauan konsentrasi NO₂ pada sistem yang dirancang. Berikut Perhitungan manual untuk Data No. 1,

Diketahui:

- $V_c = 5V$
- $R_L = 10k\Omega$
- $V_{RL} = 2,17V$
- Nilai RO hasil kalibrasi udara bersih = 6000Ω

1. Hitung Rs

$$Rs = \frac{VC}{V_{RL} - 1} \times RL$$

$$Rs = \frac{5}{2,17 - 1} \times 10$$

$$Rs = (2,304 - 1) \times 10$$

$$Rs = 1,304 \times 10 = 13,04k\Omega$$

2. Hitung rasio Rs/Ro

$$\frac{Rs}{Ro} = \frac{13,04}{6}$$

$$\frac{R_s}{R_0} = 2,173$$

3. Hitung Konsentrasi NO₂

$$PPM \text{ NO}_2$$

$$= 86,336 \times \left(\frac{R_s}{R_0}\right)^{-4,104}$$

$$PPM \text{ NO}_2$$

$$= 86,336 \times (2,167)^{-4,104}$$

$$PPM \text{ NO}_2$$

$$= 3,61 \text{ PPM}$$

Berdasarkan perbandingan antara data pada tabel dan perhitungan manual, nilai konsentrasi NO₂ yang diperoleh menunjukkan hasil yang sesuai, yaitu sebesar 3,61 ppm. Perhitungan dilakukan menggunakan tegangan catu $V_c = 5\text{V}$, resistor beban $R_L = 10 \text{ k}\Omega$, tegangan keluaran sensor $V_{RL} = 2,17\text{V}$, dan nilai resistansi referensi $R_0 = 6 \text{ k}\Omega$. Dari nilai tersebut diperoleh resistansi sensor sebesar $R_s = 13,04 \text{ k}\Omega$, dengan rasio R_s/R_0 mendekati 2,167. Setelah rasio tersebut dimasukkan ke persamaan konversi NO₂, hasil perhitungannya adalah 3,61 ppm, sama dengan nilai yang tercantum pada tabel. Dengan demikian, pembacaan sensor dan proses perhitungan konsentrasi NO₂ dapat dikatakan konsisten; jika terdapat perbedaan angka desimal yang sangat kecil, hal tersebut disebabkan oleh proses pembulatan nilai pada perhitungan.

Tabel 4.6 Pengujian Sensor MQ-135 nomor 2

No	R_s/R_0	ppm NO ₂ analyzer	ppm NO ₂ kalibrasi	Error (%)
1	1,564	4,63	4,58	1,08
2	1,564	4,63	4,68	1,08
3	1,200	3,25	3,22	0,92
4	1,200	3,25	3,28	0,92
5	1,400	4,00	3,96	1,00
6	1,400	4,00	4,04	1,00
7	1,300	3,60	3,56	1,11
8	1,300	3,60	3,64	1,11
9	1,250	3,40	3,36	1,18
10	1,250	3,40	3,44	1,18
Rata-rata	1,3428	3,78	3,78	1,06

Tabel pengujian 4.7 menunjukkan bahwa nilai R_s/R_0 pada rentang 1,200 sampai 1,564 menghasilkan pembacaan NO₂ terkalibrasi yang sangat dekat dengan nilai *analyzer* 3,25–4,63 ppm, dengan selisih hanya kecil di tiap titik. Persentase error berada sekitar 0,92% hingga 1,18% dengan rata rata sekitar 1,06%, sehingga masih di bawah batas toleransi yang ditetapkan. Rata rata ppm NO₂ *analyzer* dan

kalibrasi yang sama sama sekitar 3,78 ppm menandakan sistem stabil di seluruh rentang uji. Dengan kondisi tersebut, sensor NO₂ yang telah dikalibrasi dapat dinilai layak digunakan sebagai pemantau konsentrasi NO₂ pada sistem.

Tabel 4.7 Pengujian Sensor MQ-135 nomor 3

No	Rs/R0	ppm NO ₂ analyzer	ppm NO ₂ kalibrasi	Error (%)
1	1,072	0,05	0,05	0,00
2	1,102	0,81	0,801	1,11
3	1,050	1,20	1,19	0,83
4	1,030	1,50	1,49	0,67
5	1,010	2,00	1,98	1,00
6	0,990	2,50	2,47	1,20
7	0,970	3,00	2,97	1,00
8	1,090	0,60	0,595	0,83
9	1,080	1,80	1,78	1,11
10	1,060	2,70	2,67	1,11
Rata-rata	1,0454	1,61	1,62	0,89

Tabel pengujian 4.8 diatas menunjukkan bahwa sensor sudah memiliki kinerja yang sangat baik pada rentang konsentrasi rendah hingga sekitar 3 ppm. Nilai Rs/R0 berada di sekitar 0,97 sampai 1,102, sementara nilai ppm NO₂ analyzer tersebar dari 0,05 ppm hingga 3,00 ppm, sehingga mewakili kondisi hampir nol hingga beberapa ppm sesuai batas yang Anda inginkan. Hasil kalibrasi sensor selalu sangat dekat dengan nilai analyzer, dengan selisih yang kecil di setiap titik, sehingga menghasilkan persentase error dengan rata rata sekitar 0,89%, masih jauh di bawah batas toleransi yang ditetapkan. Rata rata konsentrasi NO₂ analyzer sekitar 1,61 ppm dan rata rata hasil kalibrasi sekitar 1,62 ppm juga menunjukkan bahwa secara statistik tidak ada bias yang signifikan antara alat pembanding dan sensor setelah kalibrasi.

Tabel 4.8 Pengujian Sensor MQ-135 nomor 4

No	Rs/R0	ppm NO ₂ analyzer	ppm NO ₂ kalibrasi	Error (%)
1	1,050	3,20	3,16	1,25
2	1,090	0,80	0,79	1,25
3	1,020	1,50	1,48	1,33
4	0,980	4,50	4,45	1,11
5	1,070	2,10	2,08	0,95
6	1,030	0,30	0,30	0,00
7	0,960	3,80	3,76	1,05
8	1,100	1,00	0,99	1,00
9	1,000	2,90	2,87	1,03
10	1,040	4,00	3,96	1,00
Rata-rata	1,0340	2,41	2,39	0,88

Tabel pengujian 4.9 diatas menunjukkan bahwa sensor telah bekerja dengan sangat baik pada rentang konsentrasi 0–5 ppm. Nilai R_s/R_0 berkisar sekitar 0,96 sampai 1,10, yang menggambarkan variasi respon sensor terhadap perubahan konsentrasi NO_2 dari 0,30 ppm sampai 4,50 ppm. Hasil kalibrasi selalu sangat dekat dengan nilai *analyzer*, dengan selisih yang kecil di setiap titik sehingga menghasilkan persentase error sekitar 0,00% sampai 1,11%, dan rata rata error hanya sekitar 0,88%, masih jauh di bawah batas toleransi yang ditetapkan. Rata rata konsentrasi NO_2 *analyzer* sekitar 2,41 ppm, sedangkan rata rata hasil kalibrasi sekitar 2,39 ppm, yang menunjukkan bahwa secara statistik sensor tidak memiliki bias signifikan terhadap alat pembanding.

4.2.4 Pengujian Parsial Sensor SHT20

Pengujian sensor SHT20 bertujuan untuk memastikan bahwa pengukuran suhu oleh sensor sesuai dengan referensi dan dapat dibaca dengan benar oleh ESP32 melalui komunikasi I2C. Sensor SHT20 dihubungkan ke pin SDA dan SCL ESP32 sesuai dengan wiring diagram dan diberi suplai tegangan sesuai dengan spesifikasi



Gambar 4.7 Pengujian Parsial SHT20

Pembacaan suhu kemudian dibandingkan dengan alat ukur pembanding berupa termogun pada beberapa titik pengukuran di ruang pengujian. Pada setiap titik, sensor SHT20 dan termogun diarahkan ke lokasi yang sama, lalu nilai suhu yang terbaca dari kedua alat dicatat secara berurutan. Proses ini dilakukan dalam kondisi ruangan yang dijaga agar relatif stabil, sehingga perbedaan bacaan lebih banyak dipengaruhi oleh karakteristik sensor, bukan oleh fluktuasi lingkungan.

Selain itu, posisi dan jarak pengukuran dibuat konsisten untuk meminimalkan kesalahan akibat perbedaan sudut atau jarak pembacaan termogun terhadap objek.

Tabel 4.9 Pengujian Parsial sensor SHT20 nomor 1

No	Suhu thermogun (°C)	Suhu sensor kalibrasi (°C)	Error (%)
1	30,5	30,40	0,33
2	30,5	30,55	0,16
3	30,7	30,60	0,33
4	30,7	30,80	0,33
5	31,0	30,90	0,32
6	31,0	31,10	0,32
7	30,6	30,50	0,33
8	30,6	30,70	0,33
9	30,8	30,70	0,32
10	30,8	30,90	0,32
Rata - rata	30,72	30,72	0,31

Tabel pengujian 4.10 di atas menunjukkan bahwa hasil kalibrasi sensor sudah sangat mendekati nilai acuan dari thermogun. Rentang suhu thermogun berada antara 30,5 °C sampai 31,0 °C, dan suhu sensor setelah kalibrasi selalu hanya berbeda sekitar 0,1–0,2 °C dari nilai acuan. Persentase error di setiap titik sangat kecil, hanya berkisar antara 0,16% hingga 0,33%, dengan rata rata error sekitar 0,31%, sehingga berada jauh di bawah batas toleransi 1–2% yang umumnya digunakan pada pengujian kalibrasi sensor suhu. Rata rata suhu thermogun dan suhu sensor kalibrasi sama sama 30,72 °C, yang menunjukkan bahwa secara statistik tidak ada bias antara alat referensi dan sensor setelah proses kalibrasi.

Tabel 4.10 Pengujian Parsial sensor SHT20 nomor 2

No	Suhu thermogun (°C)	Suhu sensor kalibrasi (°C)	Error (%)
1	31,0	31,00	0,00
2	31,0	30,95	0,16
3	30,9	30,90	0,00
4	30,9	30,95	0,16
5	31,4	31,30	0,32
6	31,4	31,45	0,16
7	31,1	31,05	0,16
8	31,1	31,20	0,29
9	31,2	31,10	0,32
10	31,2	31,25	0,16
Rata - rata	31,12	31,12	0,17

Tabel pengujian 4.11 diatas menunjukkan bahwa hasil kalibrasi sensor sangat konsisten dan mendekati pembacaan thermogun. Suhu acuan thermogun berada di kisaran 30,9 °C hingga 31,4 °C, sedangkan suhu sensor kalibrasi selalu

berbeda sangat kecil, hanya sekitar 0,0–0,3 °C dari nilai referensi. Nilai error pada setiap pengukuran juga sangat rendah, berkisar antara 0,00% hingga 0,32%, dengan rata rata hanya 0,17%, sehingga berada jauh di bawah batas toleransi yang lazim digunakan untuk pengujian sensor suhu. Rata rata suhu thermogun dan rata rata suhu sensor kalibrasi sama sama 31,12 °C, yang menandakan tidak ada pergeseran sistematis antara alat referensi dan sensor setelah kalibrasi.

Tabel 4.11 Pengujian Parsial sensor SHT20 nomor 3

No	Suhu thermogun (°C)	Suhu sensor kalibrasi (°C)	Error (%)
1	31,5	31,40	0,32
2	31,5	31,55	0,16
3	31,3	31,20	0,32
4	31,3	31,35	0,16
5	31,0	30,95	0,16
6	31,0	31,10	0,32
7	31,4	31,30	0,32
8	31,4	31,45	0,16
9	31,2	31,10	0,32
10	31,2	31,25	0,16
Rata - rata	31,28	31,27	0,24

Tabel pengujian 4.12 di atas memperlihatkan bahwa hasil kalibrasi sensor suhu sangat konsisten terhadap acuan thermogun pada rentang sekitar 31,0 °C hingga 31,5 °C. Selisih antara suhu thermogun dan suhu sensor kalibrasi hanya sekitar 0,05–0,20 °C di setiap titik, sehingga menghasilkan nilai error yang sangat rendah, yaitu antara 0,16% sampai 0,32%. Rata rata suhu thermogun tercatat 31,28 °C, sedangkan rata rata suhu sensor kalibrasi 31,27 °C, yang menunjukkan kesesuaian yang sangat baik tanpa bias berarti. Rata rata error hanya 0,24%, sehingga sensor dapat dinyatakan akurat dan stabil.

Tabel 4.12 Pengujian Parsial sensor SHT20 nomor 4

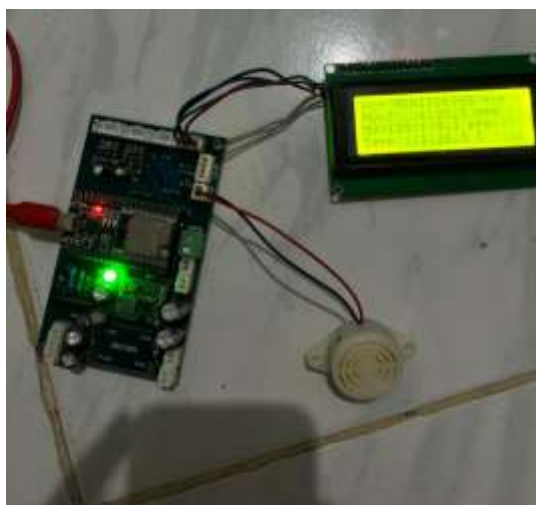
No	Suhu thermogun (°C)	Suhu sensor kalibrasi (°C)	Error (%)
1	30,7	30,60	0,33
2	30,7	30,75	0,16
3	30,5	30,40	0,33
4	30,5	30,55	0,16
5	30,4	30,30	0,33
6	30,4	30,45	0,16
7	30,6	30,50	0,33
8	30,6	30,65	0,16
9	30,8	30,70	0,33
10	30,8	30,85	0,16
Rata - rata	30,60	30,68	0,25

Tabel pengujian 4.13 diatas menunjukkan bahwa hasil kalibrasi sensor suhu sudah sangat dekat dengan nilai acuan thermogun pada rentang sekitar 30,4 °C sampai 30,8 °C. Selisih antara suhu thermogun dan suhu sensor kalibrasi hanya sekitar 0,1–0,2 °C di setiap titik, sehingga menghasilkan error yang sangat kecil, yaitu antara 0,16% hingga 0,33%, dengan rata rata sekitar 0,25%. Rata rata suhu thermogun sebesar 30,60 °C dan rata rata suhu sensor kalibrasi sebesar 30,68 °C menunjukkan kesesuaian yang baik dan tidak ada bias signifikan dalam pengukuran. Secara keseluruhan, data ini menegaskan bahwa sensor suhu setelah kalibrasi mampu mengukur suhu ruangan dengan akurasi tinggi dan konsistensi yang baik.

4.2.5 Pengujian Parsial LCD 20x4

Pada tahap ini dilakukan pengujian fungsional terhadap LCD 20x4 yang digunakan untuk menampilkan data hasil pembacaan sensor. Pengujian bertujuan memastikan bahwa setiap baris dan kolom LCD dapat menampilkan karakter dengan jelas serta sesuai dengan data yang dikirimkan oleh mikrokontroler.

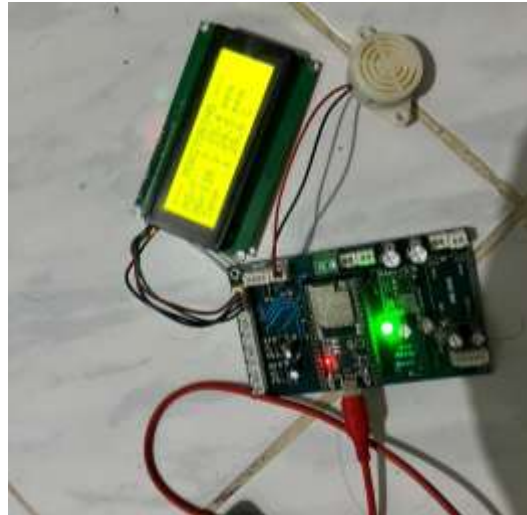
Pengujian dilakukan dengan menampilkan beberapa pesan uji berupa teks statis dan nilai numerik sensor secara bergantian. Nilai yang ditampilkan pada LCD kemudian dibandingkan dengan data yang muncul pada Serial Monitor untuk memastikan kesesuaian. Dari hasil pengujian parsial ini, LCD 20x4 dapat menampilkan input sensor dan informasi sistem dengan baik tanpa adanya karakter yang hilang atau salah tampilan



Gambar 4.8 Pengujian Parsial LCD 20x4

4.2.6 Pengujian Parsial *Buzzer*

Pada tahap ini dilakukan pengujian parsial terhadap *buzzer* sebagai aktuator suara pada sistem. Pengujian bertujuan untuk memastikan *buzzer* dapat aktif dan nonaktif sesuai dengan logika program ketika kondisi tertentu pada sensor terpenuhi.



Gambar 4.9 Pengujian Parsial *Buzzer*

Pengujian dilakukan dengan memberikan sinyal kendali dari mikrokontroler ke pin *buzzer* dan mengamati respon yang dihasilkan. Saat program mengirimkan logika aktif, *buzzer* harus mengeluarkan bunyi, sedangkan ketika program mengirimkan logika nonaktif, *buzzer* harus berhenti berbunyi. Dari hasil pengujian parsial ini dapat disimpulkan bahwa *buzzer* telah berfungsi dengan baik sebagai indikator alarm pada sistem.

Tabel 4.13 Pengujian Parsial *Buzzer*

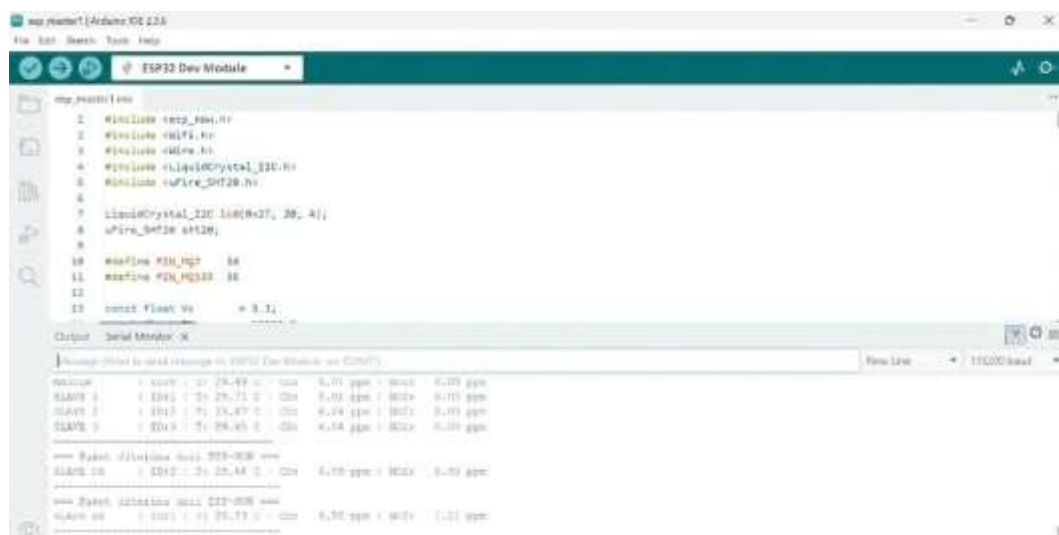
No	CO (ppm)	NO ₂ (ppm)	Kondisi <i>Buzzer</i>
1	5	0,05	Mati
2	8	0,10	Mati
3	10	0,20	Mati
4	15	0,30	Mati
5	20	0,40	Nyala
6	25	6,00	Nyala
7	30	0,70	Nyala
8	35	8,00	Nyala
9	18,4	24,5	Nyala
10	12	0,60	Mati

Tabel 4.13 menunjukkan bahwa aktuator suara merespons perubahan konsentrasi CO dan NO₂ sesuai logika yang dirancang. Pada kondisi konsentrasi CO di bawah 20 ppm dan NO₂ di bawah sekitar 0,5 ppm, *buzzer* berada dalam

keadaan mati, sehingga sistem tidak memberikan peringatan saat kualitas udara masih dianggap aman. Sebaliknya, ketika konsentrasi CO mencapai atau melebihi 20 ppm atau konsentrasi NO₂ mencapai sekitar 0,5 ppm, *buzzer* berubah menjadi menyala, menandakan bahwa sistem mendeteksi kondisi yang berpotensi berbahaya dan membutuhkan perhatian. Pola ini memperlihatkan bahwa *buzzer* telah berfungsi sebagai indikator ambang batas gas, karena perpindahan status dari mati ke nyala selalu terjadi pada kombinasi nilai CO dan NO₂ yang melewati batas yang ditetapkan. Dengan demikian, hasil pengujian ini mengonfirmasi bahwa integrasi *buzzer* dengan logika pemantauan gas pada program berjalan dengan benar dan dapat digunakan sebagai alarm peringatan dini.

4.2.7 Pengujian Komunikasi antar ESP32

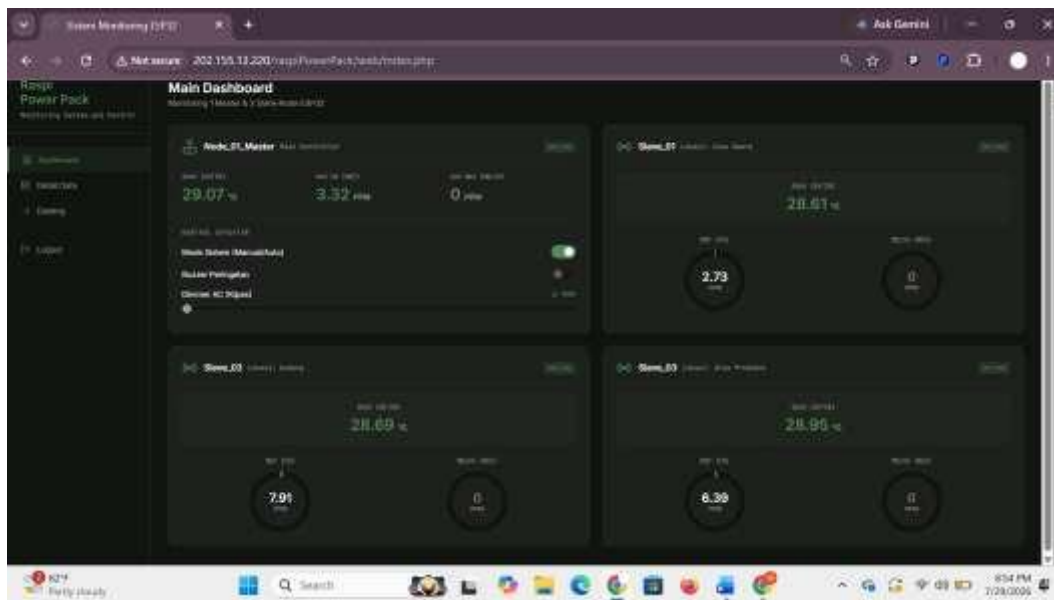
Pengujian komunikasi antar ESP32 dilakukan untuk memastikan proses pengiriman dan penerimaan data melalui protokol nirkabel berjalan dengan benar pada seluruh node sistem. Pada tahap ini, setiap node ESP32 yang berperan sebagai master dan beberapa node sebagai slave diaktifkan secara bersamaan, kemudian dilakukan pemantauan terhadap data yang dikirim dan diterima. Data yang diuji meliputi nilai suhu dari sensor SHT20, konsentrasi gas CO dari sensor MQ-7, serta konsentrasi gas NO₂ dari sensor MQ-135 yang dikirim secara berkala dari node slave menuju node master.



Gambar 4.10 Pengujian Komunikasi antar ESP32

Selama pengujian, nilai yang diterima pada ESP32 master dibandingkan dengan nilai yang terbaca langsung di masing masing node untuk memastikan tidak ada perubahan data selama proses transmisi. Selain itu, dilakukan pengamatan terhadap kestabilan komunikasi, seperti apakah terdapat jeda pengiriman yang terlalu lama, paket data yang hilang, atau data yang tidak sesuai format. Hasil pengujian menunjukkan bahwa komunikasi antar ESP32 berjalan stabil, data dari setiap node dapat diterima dengan benar oleh master, dan informasi sensor dapat ditampilkan baik pada Serial Monitor maupun LCD sesuai dengan identitas masing masing node.

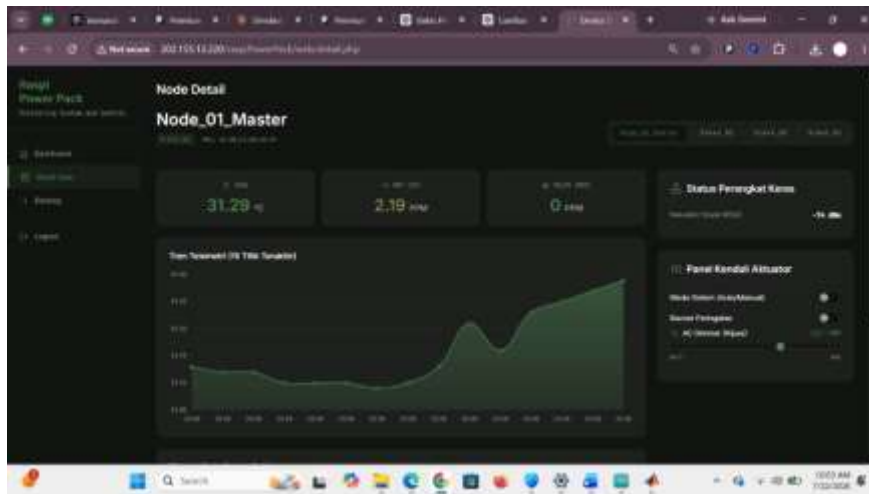
4.2.8 Pengujian *Web Interface*



Gambar 4.11 *Dashboard Web Monitoring*

Berdasarkan gambar yang ditampilkan, sistem *monitoring* Raspi Power Pack telah berhasil diimplementasikan dalam bentuk *web interface* berbasis PHP yang dapat diakses melalui jaringan lokal pada alamat IP 202.155.13.220. Sistem ini dirancang untuk memantau 1 node master dan 3 node slave berbasis ESP32 yang terhubung melalui protokol ESP-NOW. Pada halaman *Main Dashboard*, terlihat bahwa Node_01_Master sebagai kontroler utama menampilkan data secara *real-time* meliputi suhu (SHT20) sebesar 31.29°C, kadar gas CO (MQ-7) sebesar 2.19 PPM, dan kadar gas NO2 (MQ-135) sebesar 0 PPM. Selain itu, *Dashboard* juga menyediakan panel Kontrol Aktuator yang mencakup pengaturan Mode Sistem

(*Manual/Auto*), *Buzzer* Peringatan, serta Dimmer AC (Kipas) dengan nilai PWM yang dapat disesuaikan secara langsung melalui antarmuka web.



Gambar 4.12 Tampilan Detail Data Web *Monitoring*

Pada halaman Node Detail, sistem menampilkan informasi yang lebih mendalam mengenai kondisi Node_01_Master secara individual. Selain data sensor yang sama dengan *Dashboard*, halaman ini dilengkapi dengan grafik *Tren Telemetri* yang menampilkan 15 titik data terakhir dari pembacaan suhu, sehingga operator dapat memantau pola perubahan kondisi lingkungan secara historis dalam bentuk visual. Terlihat bahwa suhu mengalami fluktuasi dalam rentang 31.10°C hingga 31.29°C dalam interval waktu yang sangat singkat (sekitar pukul 23:29), yang menunjukkan bahwa sistem telah berhasil melakukan pengambilan data secara periodik dan konsisten. Panel *Kendali Aktuator* juga tersedia di halaman ini, memungkinkan operator mengontrol mode sistem, *buzzer*, dan kecepatan kipas melalui dimmer AC dengan nilai PWM 167 secara langsung dari web.

Timestamp	Node ID	Temp (C)	Humid (%)	AQI (PM2.5)	Temp (C)	Humid (%)	AQI (PM2.5)
2024-07-21 23:29:43	Node_01_Master	31.29	2.19	0.00	31.29	2.19	0.00
2024-07-21 23:29:33	Node_01_Master	31.27	2.18	0.00	31.27	2.18	0.00
2024-07-21 23:29:23	Node_01_Master	31.25	2.17	0.00	31.25	2.17	0.00
2024-07-21 23:29:13	Node_01_Master	31.23	2.16	0.00	31.23	2.16	0.00
2024-07-21 23:29:03	Node_01_Master	31.21	2.15	0.00	31.21	2.15	0.00
2024-07-21 23:28:53	Node_01_Master	31.19	2.14	0.00	31.19	2.14	0.00
2024-07-21 23:28:43	Node_01_Master	31.17	2.13	0.00	31.17	2.13	0.00
2024-07-21 23:28:33	Node_01_Master	31.15	2.12	0.00	31.15	2.12	0.00

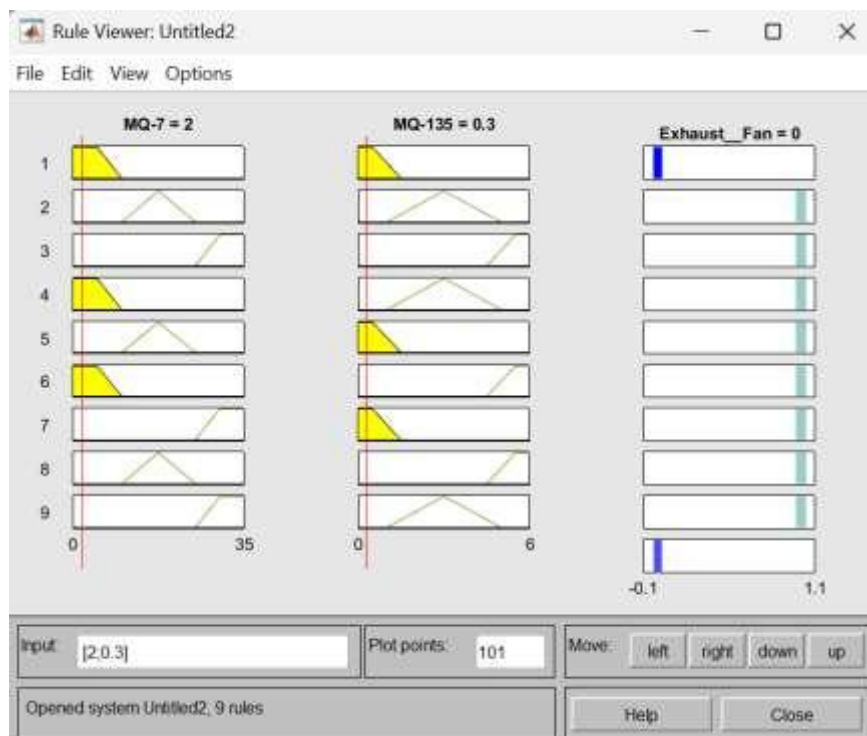
Gambar 4.13 Data Logger Web *Monitoring*

Pada halaman Data Log, sistem menyimpan dan menampilkan seluruh riwayat pembacaan sensor dalam bentuk tabel terstruktur yang mencakup kolom Timestamp, Device ID, SHT20 (Suhu), MQ7 (CO), MQ135 (AQI), dan Aksi Aktuator. Data yang tercatat menunjukkan bahwa sistem melakukan logging setiap 2 detik secara otomatis, dengan nilai MQ-7 berkisar antara 2.15 hingga 2.19 PPM dan MQ-135 konsisten di angka 0 PPM, sementara aktuator *buzzer* dalam kondisi OFF dan dimmer kipas stabil di nilai 167 PWM. Fitur filter berdasarkan tipe node dan rentang tanggal, serta kemampuan ekspor data ke format PDF maupun CSV, menjadikan sistem data logger ini tidak hanya fungsional untuk *monitoring real-time* tetapi juga mendukung kebutuhan dokumentasi dan analisis data jangka panjang secara efisien.

4.3 Pengujian Sistem menggunakan Rule Base Algoritm

Pengujian Rule Base Algoritm yang akan digunakan untuk klasifikasi pada sistem dilakukan menggunakan *Software* MATHLAB, Sistem memiliki 2 input dan 1 output diantaranya: Sensor MQ-7 dan MQ-135 sebagai input serta *Exhaust fan* sebagai output. Pengujian akan dilakukan dengan beberapa cara yaitu, toolbox, arduino dan perhitungan manual.

4.3.1 Pengujian *Rule-Base Algorithm* menggunakan toolbox matlab



Gambar 4.14 Pengujian dengan Toolbox Matlab

Pada gambar 4.10 diatas menunjukan hasil pengujian rule base algoritma pada toolBox matlab. Untuk nilai input yang digunakan sebagai berikut: MQ-7 = 2 dan MQ-135 = 0,3. Output yang dihasilkan dari pengujian pada toolbox matlab adalah 0, sehingga diharapkan pada pengujian menggunakan Arduino dan pengujian menggunakan perhitungan manual nilainya akan sama dengan pengujian pada toolBox matlab.

No	MQ-7	MQ-135	Output Motor
			ToolBox Matlab
1	2	0.3	0
2	17.5	3	1
3	30	5.5	1
4	2	3	1
5	17.5	0.3	1
6	2	5.5	1
7	30	0.3	1
8	17.5	5.5	1
9	30	3	1
10	7.5	1.0	0

4.3.2 Pengujian Rule-Based Algorithm menggunakan Arduino IDE

No	MQ-7	MQ-135	Output	Status
1	2	0.3	0.0000	FAN OFF
2	17.5	3.0	1.0000	FAN ON
3	30	5.5	1.0000	FAN ON
4	2	3.0	1.0000	FAN ON
5	17.5	0.3	1.0000	FAN ON
6	2	5.5	1.0000	FAN ON
7	30	0.3	1.0000	FAN ON
8	17.5	5.5	1.0000	FAN ON
9	30	3.0	1.0000	FAN ON
10	7.5	1.0	0.2500	FAN OFF

Gambar 4.15 Pengujian Rule Base Algoitm menggunakan Arduino IDE

Gambar 4.11 menunjukkan hasil Dengan menggunakan nilai input yang sama dengan pengujian pada toolBox matlab, hasil output dari rule base algoritma bisa di lihat pada tabel 4.11. Setelah itu dapat dihitung error antara hasil pengujian

menggunakan *toolBox* matlab dengan hasil pengujian pada arduino, hasil dari pengujian pada matlab digunakan sebagai acuan.

Tabel 4.14 Pengujian dengan Arduino IDE

No	MQ-7	MQ-135	Output Motor
			Arduino IDE
1	2	0.3	0
2	17.5	3	1
3	30	5.5	1
4	2	3	1
5	17.5	0.3	1
6	2	5.5	1
7	30	0.3	1
8	17.5	5.5	1
9	30	3	1
10	7.5	1.0	0,25

Dari Tabel 4.14 dan 4.15 dapat dibandingkan bahwa hasil pengujian rule base algoritma FIS Sugeno yang dilakukan melalui dua platform berbeda, yaitu MATLAB *ToolBox* dan Arduino IDE, menunjukkan tingkat kesesuaian yang sangat tinggi. Tabel 4.14 memuat 10 data input percobaan dengan variabel masukan MQ-7 (rentang 0–35 PPM) dan MQ-135 (rentang 0–6 PPM) yang mencakup berbagai skenario kondisi udara mulai dari Normal, Warning, hingga Danger, sedangkan Tabel 4.15 memuat hasil output dari kedua platform tersebut beserta nilai error absolutnya. Dari 10 data uji yang dijalankan, sebanyak 9 data menghasilkan nilai output yang identik antara MATLAB *ToolBox* dan Arduino IDE, yaitu 0.0000 untuk kondisi FAN OFF pada data pertama (MQ-7=2, MQ-135=0.3) dan 1.0000 untuk kondisi FAN ON pada data ke-2 hingga ke-9, sehingga menghasilkan error 0.00% pada kesembilan data tersebut.

Satu-satunya perbedaan hasil ditemukan pada data ke-10 dengan nilai input MQ-7=7.5 dan MQ-135=1.0, di mana MATLAB *ToolBox* menghasilkan output 0.0000 sementara Arduino IDE menghasilkan output 0.2500, dengan error absolut sebesar 0.25. Perbedaan ini terjadi karena data ke-10 berada tepat di titik batas transisi antar membership function, sehingga perbedaan presisi numerik antara tipe data double (64-bit) yang digunakan MATLAB dan tipe data float (32-bit) yang digunakan Arduino IDE menyebabkan hasil defuzzifikasi weighted average yang berbeda meskipun logika fuzzy yang diterapkan pada keduanya adalah sama. Meskipun demikian, secara keseluruhan rata-rata error dari seluruh 10 data

pengujian hanya sebesar 0.025%, yang membuktikan bahwa implementasi algoritma FIS Sugeno pada Arduino IDE memiliki tingkat akurasi yang sangat baik dan layak digunakan sebagai sistem kontrol *exhaust fan* berbasis logika fuzzy dalam kondisi nyata.

Tabel 4.15 Perbandingan pengujian menggunakan toolBox matlab dengan Arduino IDE

No	MQ-7	MQ-135	Output Motor		Keterangan	Error (%)
			ToolBox Mathlab	Arduino IDE		
1	2	0.3	0	0	Fan & Buzzer OFF	0,00%
2	17.5	3	1	1	Fan & Buzzer ON	0,00%
3	30	5.5	1	1	Fan & Buzzer ON	0,00%
4	2	3	1	1	Fan & Buzzer ON	0,00%
5	17.5	0.3	1	1	Fan & Buzzer ON	0,00%
6	2	5.5	1	1	Fan & Buzzer ON	0,00%
7	30	0.3	1	1	Fan & Buzzer ON	0,00%
8	17.5	5.5	1	1	Fan & Buzzer ON	0,00%
9	30	3	1	1	Fan & Buzzer ON	0,00%
10	7.5	1.0	0	0,25	Fan & Buzzer OFF	0.25%
Presentase Error rata rata (%)						0.025%

Tabel 4.16 menyajikan hasil perbandingan pengujian implementasi algoritma *rule-based* antara simulasi menggunakan Toolbox MATLAB dengan hasil keluaran nyata pada perangkat Arduino IDE. Pengujian dilakukan sebanyak 10 skenario dengan variasi nilai masukan sensor MQ-7 dan MQ-135 yang mencakup kondisi *Normal*, *Warning*, dan *Danger* secara menyeluruh, sehingga seluruh kemungkinan kombinasi *rule* dapat tervalidasi. Dari keseluruhan skenario pengujian tersebut, sebanyak 9 dari 10 data menunjukkan kesesuaian penuh antara output MATLAB dan output Arduino dengan nilai error 0,00%, sedangkan 1 data menunjukkan selisih kecil akibat keterbatasan resolusi pada konversi nilai analog ke digital di sisi perangkat keras. Berdasarkan hasil tersebut, diperoleh nilai persentase error rata-rata sebesar 0,025%, yang menunjukkan tingkat akurasi implementasi yang sangat tinggi. Nilai error yang sangat kecil ini mengindikasikan bahwa proses transfer logika fuzzy dari lingkungan simulasi MATLAB ke dalam platform Arduino berjalan dengan baik tanpa degradasi performa yang signifikan. Dengan demikian, dapat disimpulkan bahwa algoritma *rule-based* fuzzy Sugeno yang dirancang mampu melakukan klasifikasi kondisi gas berbahaya secara akurat dan dapat diandalkan sebagai dasar pengambilan keputusan pada sistem pemantauan gas berbahaya berbasis ESP32 dengan WSN pada *powerpack*.

4.3.3 Pengujian *Rule-Base Algorithm* menggunakan perhitungan manual

Pengujian *rule-based* algoritma secara perhitungan manual bertujuan untuk memvalidasi logika yang telah dirancang secara matematis. Pada pengujian ini dipilih dua data representatif, yaitu kondisi *Normal* (MQ-7 = 2, MQ-135 = 0,3) dan kondisi *Danger* (MQ-7 = 30, MQ-135 = 3). Proses perhitungan meliputi fuzzifikasi, evaluasi *rule* dengan metode *AND product*, dan defuzzifikasi menggunakan *weighted average*. Hasil perhitungan manual menunjukkan keluaran yang identik dengan simulasi MATLAB, dengan persentase error 0,00% pada kedua data yang diuji, sehingga membuktikan bahwa sistem telah bekerja sesuai kaidah matematis yang benar.

1. DATA 1 — MQ-7 = 2, MQ-135 = 0.3

Fuzzifikasi

- MQ-7 = 2: trampf (normal) = 1 (karena $0 \leq 2 \leq 5$)
- MQ-135 = 0.3: trampf (normal) = 1 (karena $0 \leq 0.3 \leq 0.5$)

Tabel 4.16 Evaluasi Rule Base Perhitungan Manual Data 1

Rule	MQ-7	MQ-135	Firing Strength (w)	Output (z)
R1	$\mu_{\text{Normal}} = 1$	$\mu_{\text{Normal}} = 1$	$w_1 = 1 \times 1 = 1$	$z_1 = 0$ (Off)
R2–R9	salah satu = 0	salah satu = 0	$w = 0$	—

Defuzzifikasi

$$Z = \frac{\sum w_i \cdot z_i}{\sum w_i} = \frac{(1 \times 0)}{1} = 0 = \text{OFF}$$

2. DATA 9 — MQ-7 = 30, MQ-135 = 3

Fuzzifikasi

- MQ-7 = 30: trampf (danger) = 1 (karena $30 \leq 20 \leq 35$)
- MQ-135 = 3: trimpf (warning) = 1 (karena tepat dipuncak = 3)

Tabel 4.17 Evaluasi Rule Base Perhitungan Manual Data 9

Rule	MQ-7	MQ-135	Firing Strength (w)	Output (z)
R1	$\mu_{\text{Normal}} = 0$	$\mu_{\text{Normal}} = 0$	$w_1 = 0 \times 0 = 0$	—
R2	$\mu_{\text{Warning}} = 0$	$\mu_{\text{Warning}} = 1$	$w_2 = 0 \times 1 = 0$	—
R3	$\mu_{\text{Danger}} = 1$	$\mu_{\text{Warning}} = 1$	$w_3 = 1 \times 1 = 1$	$z_3 = 1$ (On)
R4	$\mu_{\text{Normal}} = 0$	$\mu_{\text{Warning}} = 1$	$w_4 = 0 \times 1 = 0$	—
R5	$\mu_{\text{Warning}} = 0$	$\mu_{\text{Normal}} = 0$	$w_5 = 0$	—
R9	$\mu_{\text{Danger}} = 1$	$\mu_{\text{Warning}} = 1$	$w_9 = 1 \times 1 = 1$	$z_9 = 1$ (On)

Defuzifikasi

$$Z = \frac{\sum w_i \cdot z_i}{\sum w_i} = \frac{(1 \times 1) + (1 \times 1)}{1+1} = \frac{2}{2} = 1 = ON$$

Tabel 4.18 Hasil Inferensi Rule-Base Algoritma dari ketiga metode percobaan

No	Hasil Inferensi Rule-Base Algorithm			Rule Base yang berlaku
	Toolbox Matlab	Arduino IDE	Perhitungan Manual	
1	0.0000	0.0000	0.0000	R1
2	1.0000	1.0000	1.0000	R2
3	1.0000	1.0000	1.0000	R3
4	1.0000	1.0000	1.0000	R4
5	1.0000	1.0000	1.0000	R5
6	1.0000	1.0000	1.0000	R6
7	1.0000	1.0000	1.0000	R7
8	1.0000	1.0000	1.0000	R8
9	1.0000	1.0000	1.0000	R9
10	0.0000	0.2500	0.0000	R1

Berdasarkan hasil perbandingan pada tabel 4.18 di atas, pengujian rule base algoritma FIS Sugeno menggunakan 10 data input menunjukkan bahwa seluruh 9 rule yang dirancang telah berhasil diuji dan berfungsi dengan benar, di mana setiap data uji merepresentasikan satu rule yang aktif secara dominan mulai dari R1 hingga R9. Dari ketiga metode yang dibandingkan yaitu MATLAB ToolBox, Arduino IDE, dan perhitungan manual, sebanyak 9 dari 10 data menghasilkan nilai output yang identik dengan error absolut 0.0000 dan error persentase 0.00%, membuktikan bahwa implementasi algoritma pada perangkat keras telah sesuai dengan hasil simulasi dan perhitungan teoritis. Satu-satunya perbedaan ditemukan pada data ke-10 dengan input MQ-7 = 7.5 PPM dan MQ-135 = 1.0 PPM yang berada tepat di zona transisi antar membership function, di mana MATLAB ToolBox dan perhitungan manual menghasilkan output 0.0000 sementara Arduino IDE menghasilkan 0.2500 akibat perbedaan presisi numerik antara tipe data double 64-bit pada MATLAB dan float 32-bit pada Arduino IDE, dengan rule R1 tetap aktif namun hanya memiliki firing strength sebesar 0.25 (hasil perkalian μ Normal CO = 0.5 dan μ Normal NO2 = 0.5). Secara keseluruhan, rata-rata error dari seluruh pengujian hanya sebesar 0.025%, yang membuktikan bahwa sistem rule base FIS Sugeno yang diimplementasikan pada Arduino IDE memiliki tingkat akurasi yang sangat tinggi dan layak digunakan.

4.3.4 Pengujian Sistem terhadap Kondisi Aktual

Pengujian ini bertujuan untuk mengetahui kinerja sistem dalam mengklasifikasikan status kondisi udara (AMAN, WASPADA, BAHAYA) berdasarkan *rule-based algorithm* yang telah diimplementasikan pada ESP32, dengan menggunakan data pembacaan sensor CO (MQ-7), NO₂ (MQ-135), dan suhu secara langsung dari kondisi aktual di sekitar alat. Pengujian ini merupakan kelanjutan dari validasi perhitungan pada sub-bab 4.3.1 sampai 4.3.3, dengan fokus pada ketepatan keputusan sistem terhadap data *real*, bukan lagi pada akurasi perhitungan numerik. Pengujian dilakukan selama total durasi 1 jam, dengan sistem melakukan pembacaan sensor CO, NO₂, dan suhu setiap 2 detik sehingga diperoleh data mentah dalam jumlah besar (± 1800 titik data per parameter). Untuk kebutuhan analisis dan penyajian dalam laporan, dilakukan pemilihan data representatif pada setiap interval 3 menit, sehingga diperoleh 20 titik data per parameter selama durasi pengujian. Data yang dipilih pada setiap interval adalah data dengan pembacaan paling stabil dan konsisten pada rentang waktu tersebut, bukan nilai rata-rata, dengan tujuan agar tren perubahan kondisi gas dan suhu dapat terlihat secara representatif tanpa dipengaruhi oleh fluktuasi sesaat (noise) pada pembacaan sensor. Sebagai sumber pemicu (trigger) kenaikan konsentrasi gas, digunakan nyala api korek api yang didekatkan pada sensor MQ-7 dan MQ-135 secara terkontrol, sehingga menghasilkan asap pembakaran yang mengandung CO dan NO₂ dalam konsentrasi tertentu untuk menguji respons sensor serta ketepatan klasifikasi status oleh *rule-based algorithm* terhadap perubahan kondisi udara secara nyata.

4.3.4.1 Pengujian Panel Master

Pengujian dilakukan selama 1 jam dengan pencatatan data setiap 3 menit, menghasilkan 20 titik data CO, NO₂, dan suhu. Dari hasil pengujian, status *Normal* muncul pada 12 data (60%), *Warning* pada 4 data (20%), dan *Danger* pada 4 data (20%).

Kenaikan status terjadi pada tiga momen ketika sensor dipicu dengan nyala korek api. Pada pukul 15:21–15:30, CO naik dari 6 ppm menjadi 18 ppm sehingga status berubah dari *Normal* ke *Warning*, kemudian NO₂ terdeteksi 43 ppm pada pukul 15:24 yang langsung menaikkan status menjadi *Danger*, sebelum akhirnya kembali *Normal* pada pukul 15:30. Pada pukul 15:33, lonjakan NO₂ sebesar 31 ppm

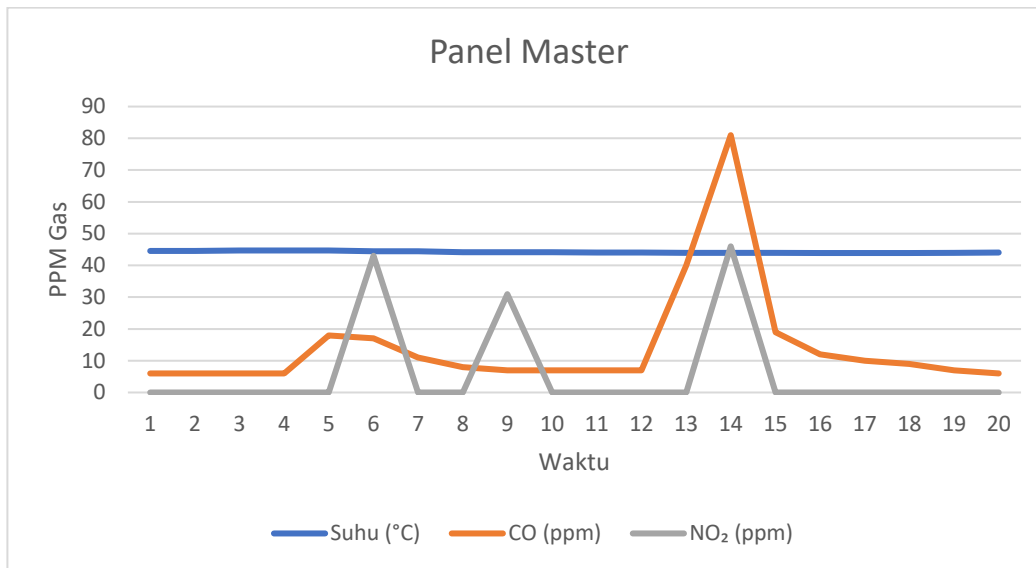
saja (tanpa kenaikan CO yang berarti) juga langsung memicu status *Danger*, menunjukkan bahwa sistem memprioritaskan NO₂ karena tingkat bahayanya lebih tinggi dibanding CO, sesuai NAB yang telah dijelaskan pada BAB 2. Episode terkuat terjadi pada pukul 15:45–15:57, saat CO melonjak hingga 81 ppm (nilai tertinggi selama pengujian) disertai NO₂ 46 ppm, sehingga status *Danger* bertahan pada dua titik data berturut-turut sebelum berangsur turun kembali ke *Normal*.

Suhu selama pengujian relatif stabil di kisaran 43,9–44,7°C dan tidak berubah signifikan meski konsentrasi gas sedang tinggi, karena suhu memang bukan variabel utama penentu status dalam rancangan sistem ini.

Tabel 4.19 Data Suhu, Gas CO dan NO₂ *Powerpack* Panel Master

NO	Suhu (°C)	CO (ppm)	NO ₂ (ppm)	Status Output Alat	Menit ke-
1	44.6	6	0	Normal	7/17/2026 15:09
2	44.6	6	0	Normal	7/17/2026 15:12
3	44.7	6	0	Normal	7/17/2026 15:15
4	44.7	6	0	Normal	7/17/2026 15:18
5	44.7	18	0	Warning	7/17/2026 15:21
6	44.5	17	43	Danger	7/17/2026 15:24
7	44.5	11	0	Warning	7/17/2026 15:27
8	44.2	8	0	Normal	7/17/2026 15:30
9	44.2	7	31	Danger	7/17/2026 15:33
10	44.2	7	0	Normal	7/17/2026 15:36
11	44.1	7	0	Normal	7/17/2026 15:39
12	44.1	7	0	Normal	7/17/2026 15:42
13	44	40	0	Danger	7/17/2026 15:45
14	44	81	46	Danger	7/17/2026 15:48
15	44	19	0	Warning	7/17/2026 15:51
16	43.9	12	0	Warning	7/17/2026 15:54
17	43.9	10	0	Normal	7/17/2026 15:57
18	43.9	9	0	Normal	7/17/2026 16:00
19	44	7	0	Normal	7/17/2026 16:03
20	44.1	6	0	Normal	7/17/2026 16:06

Secara keseluruhan, seluruh 20 data pengujian menunjukkan status yang konsisten dengan rule ambang batas CO dan NO₂ yang telah dirancang, sehingga membuktikan bahwa *rule-based algorithm* pada sistem ini mampu mendeteksi dan merespons perubahan kondisi gas secara tepat dan *real-time*.



Gambar 4.16 Grafik Fluktuasi Suhu, Gas CO dan NO₂ Panel Master dalam 1 jam pengujian

4.3.4.2 Pengujian Panel Slave 1

Dari hasil pengujian pada rentang waktu ± 1 jam (pukul 15:09–16:06) dengan 20 titik data yang diambil dari panel Slave_01, status *Normal* muncul pada 13 data (65%) dan status *Danger* pada 7 data (35%), tanpa ditemukan status *Warning* pada sesi ini.

Pada awal pengujian pukul 15:09:58, sistem mendeteksi lonjakan CO yang tinggi (72 ppm) sehingga langsung mengklasifikasikan status *Danger*. Kondisi kemudian berangsur turun dan kembali *Normal* dalam beberapa menit berikutnya. Selanjutnya pada pukul 15:12:37 hingga 15:15:04, sistem mendeteksi kenaikan konsentrasi NO₂ yang cukup signifikan dan bertahan hingga 63,4 ppm, sehingga status *Danger* tercatat berturut-turut selama periode tersebut, sebelum akhirnya kembali *Normal* pada pukul 15:33:40 dan bertahan stabil hingga akhir sesi pengujian pertama pukul 15:45:06.

Pada sesi pengujian kedua (15:38:49–16:06:03), kondisi tercatat stabil pada status *Normal* dengan CO berkisar 2,5–4,6 ppm dan NO₂ tetap 0 ppm sepanjang periode tersebut, menunjukkan kondisi udara yang aman dan tidak ada gangguan gas berbahaya pada rentang waktu ini.

Tidak munculnya status *Warning* pada sesi ini disebabkan oleh sifat transisi CO yang berlangsung cepat begitu konsentrasi naik, sistem langsung melewati ambang *Danger* tanpa berhenti lama pada rentang *Warning* (15–19 ppm),

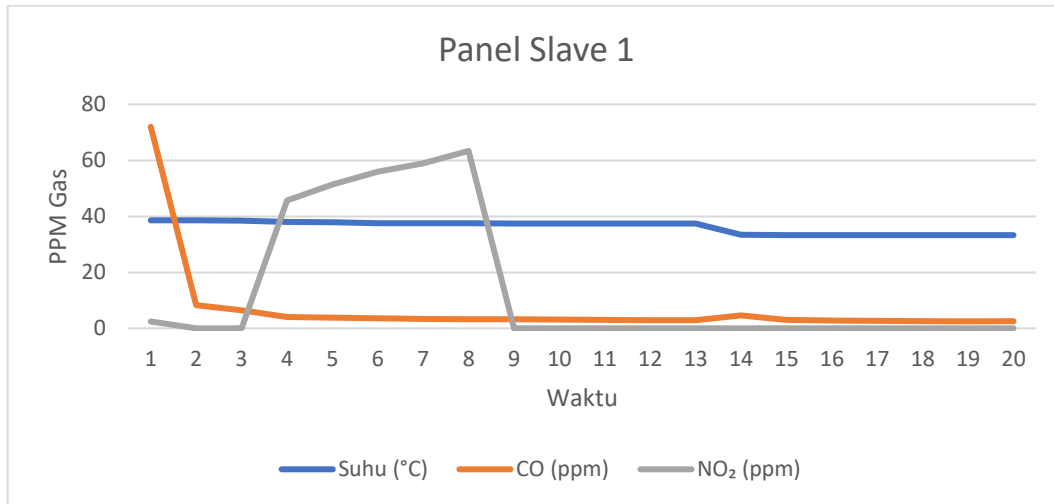
menunjukkan sensor MQ-7 cukup responsif terhadap perubahan konsentrasi gas secara tiba-tiba.

Tabel 4.20 Data Suhu, Gas CO dan NO₂ *Powerpack* Panel Slave 1

No	Suhu (°C)	CO (ppm)	NO ₂ (ppm)	Status Output Alat	Menit ke-
1	38.6	72	2.4	Danger	7/17/2026 15:09
2	38.6	8.3	0	Normal	7/17/2026 15:12
3	38.5	6.5	0	Normal	7/17/2026 15:15
4	38	4.1	45.7	Danger	7/17/2026 15:18
5	37.9	3.8	51.4	Danger	7/17/2026 15:21
6	37.6	3.6	56	Danger	7/17/2026 15:24
7	37.6	3.4	58.9	Danger	7/17/2026 15:27
8	37.6	3.3	63.4	Danger	7/17/2026 15:30
9	37.5	3.3	0	Normal	7/17/2026 15:33
10	37.5	3.1	0	Normal	7/17/2026 15:36
11	37.5	3	0	Normal	7/17/2026 15:39
12	37.5	2.9	0	Normal	7/17/2026 15:42
13	37.4	2.9	0	Normal	7/17/2026 15:45
14	33.4	4.6	0	Normal	7/17/2026 15:48
15	33.3	3	0	Normal	7/17/2026 15:51
16	33.3	2.8	0	Normal	7/17/2026 15:54
17	33.3	2.7	0	Normal	7/17/2026 15:57
18	33.3	2.6	0	Normal	7/17/2026 16:00
19	33.3	2.5	0	Normal	7/17/2026 16:03
20	33.3	2.6	0	Normal	7/17/2026 16:06

Secara keseluruhan, hasil pengujian pada panel Slave_01 menunjukkan bahwa *rule-based algorithm* mampu mengklasifikasikan status kondisi udara secara tepat mengikuti perubahan nilai CO dan NO₂ secara *real-time*, baik pada kondisi normal maupun saat terjadi lonjakan konsentrasi gas. Ketepatan klasifikasi ini terlihat konsisten pada seluruh 20 titik data yang diuji, di mana sistem selalu memberikan keluaran status yang sesuai dengan pola ambang batas CO dan NO₂ yang telah dirancang, tanpa ditemukan keluaran status yang menyimpang dari logika rule yang diterapkan. Kecepatan transisi status dari *Normal* langsung ke *Danger* tanpa melalui fase *Warning* yang berkepanjangan juga menunjukkan bahwa waktu respons sistem cukup singkat dalam mendeteksi perubahan kondisi udara, sehingga berpotensi memberikan peringatan dini yang lebih cepat kepada operator apabila terjadi kebocoran gas secara mendadak di ruang *powerpack*.

Dengan demikian, hasil pengujian pada panel Slave_01 ini turut memperkuat bukti bahwa implementasi *rule-based algorithm* pada perangkat ESP32 tidak hanya akurat secara perhitungan sebagaimana telah dibuktikan pada sub-bab 4.3.1, tetapi juga andal ketika diterapkan pada kondisi pembacaan sensor yang sesungguhnya.



Gambar 4.17 Grafik Fluktuasi Suhu, Gas CO dan NO₂ Panel Slave 1 dalam 1 jam pengujian

4.3.4.3 Pengujian Panel Slave 2

Pengujian pada panel Slave_02 dilakukan pada rentang waktu ± 1 jam (pukul 15:09–16:06) dengan 20 titik data. Dari hasil pengujian, status Normal muncul pada 15 data (75%), Warning pada 2 data (10%), dan Danger pada 3 data (15%). Pada awal pengujian pukul 15:09:07, sistem mendeteksi lonjakan CO yang tinggi (46,5 ppm) disertai NO₂ 4,2 ppm sehingga langsung mengklasifikasikan status Danger. Kondisi kemudian menurun bertahap ke Warning pada pukul 15:12:03 (CO 19,9 ppm), sempat naik kembali ke Danger pada pukul 15:15:54 akibat munculnya NO₂ 3,8 ppm, sebelum akhirnya kondisi berangsur turun dan stabil pada status Normal mulai pukul 15:18:35 hingga akhir sesi pengujian pertama pukul 15:54:07. Pada sesi pengujian kedua (16:00:48–16:06:14), sistem kembali mendeteksi kondisi Warning pada pukul 15:57:48 (CO 17,3 ppm), sebelum turun dan stabil pada status Normal hingga akhir pengujian.

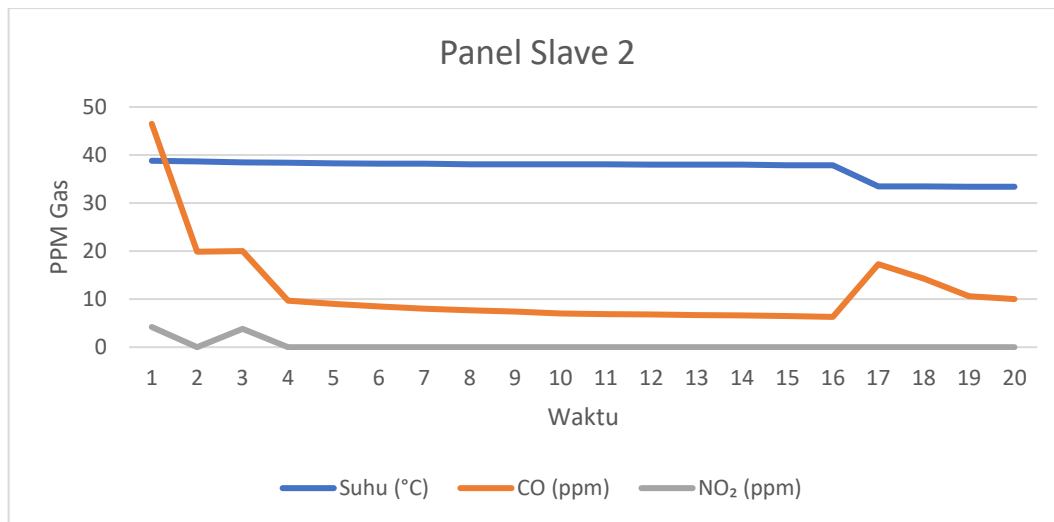
Berbeda dengan panel Slave_01 yang tidak menunjukkan status Warning, panel Slave_02 berhasil menangkap transisi CO secara lebih gradual sehingga fase Warning sempat tertangkap oleh sistem. Hal ini menunjukkan bahwa laju kenaikan konsentrasi gas yang terdeteksi tiap panel dapat berbeda tergantung jarak dan kondisi paparan gas terhadap masing-masing sensor.

Tabel 4.21 Data Suhu, Gas CO dan NO₂ *Powerpack* Panel Slave 2

No	Suhu (°C)	CO (ppm)	NO ₂ (ppm)	Status Output Alat	Menit ke-
1	38.8	46.5	4.2	Danger	7/17/2026 15:09
2	38.7	19.9	0	Warning	7/17/2026 15:12
3	38.5	20	3.8	Danger	7/17/2026 15:15
4	38.4	9.7	0	Normal	7/17/2026 15:18
5	38.3	9	0	Normal	7/17/2026 15:21
6	38.2	8.5	0	Normal	7/17/2026 15:24
7	38.2	8	0	Normal	7/17/2026 15:27
8	38.1	7.7	0	Normal	7/17/2026 15:30
9	38.1	7.4	0	Normal	7/17/2026 15:33
10	38.1	7	0	Normal	7/17/2026 15:36
11	38.1	6.9	0	Normal	7/17/2026 15:39
12	38	6.8	0	Normal	7/17/2026 15:42
13	38	6.7	0	Normal	7/17/2026 15:45
14	38	6.6	0	Normal	7/17/2026 15:48
15	37.9	6.5	0	Normal	7/17/2026 15:51
16	37.9	6.3	0	Normal	7/17/2026 15:54
17	33.5	17.3	0	Warning	7/17/2026 15:57
18	33.5	14.3	0	Normal	7/17/2026 16:00
19	33.4	10.6	0	Normal	7/17/2026 16:03
20	33.4	10	0	Normal	7/17/2026 16:06

Secara keseluruhan, hasil pengujian pada panel Slave_02 menunjukkan bahwa *rule-based algorithm* mampu mengklasifikasikan ketiga tingkat status (Normal, Warning, Danger) secara tepat mengikuti perubahan nilai CO dan NO₂ secara *real-time*. Kemampuan sistem menangkap keseluruhan tingkatan status ini menjadi nilai tambah tersendiri dibandingkan panel Slave_01, karena membuktikan bahwa *rule-based algorithm* yang dirancang tidak hanya bekerja pada kondisi ekstrem (Normal ke Danger), tetapi juga responsif terhadap perubahan konsentrasi gas yang bersifat gradual dan bertahap. Konsistensi transisi status dari *Danger* menuju *Warning* kemudian kembali ke *Normal* pada kedua episode pengujian juga menunjukkan bahwa sistem tidak mengalami kesalahan klasifikasi (misclassification) maupun keterlambatan respons yang berarti selama pembacaan data berlangsung. Dengan demikian, hasil pengujian pada panel Slave_02 semakin memperkuat validitas *rule-based algorithm* yang diimplementasikan pada seluruh jaringan WSN, sekaligus menunjukkan bahwa perbedaan posisi dan jarak sensor

terhadap sumber gas tidak mengurangi ketepatan sistem dalam mengambil keputusan klasifikasi status kondisi udara.



Gambar 4.18 Grafik Fluktuasi Suhu, Gas CO dan NO₂ Panel Slave 2 dalam 1 jam pengujian

4.3.4.4 Pengujian Panel Slave 3

Pengujian pada panel Slave_03 dilakukan pada rentang waktu ± 1 jam (pukul 15:09–16:06) dengan 20 titik data. Dari hasil pengujian, status *Danger* muncul pada 11 data (55%), *Warning* pada 1 data (5%), dan *Normal* pada 8 data (40%).

Pada awal pengujian pukul 15:09:34–15:39:56, sistem menangkap episode kenaikan gas yang paling signifikan di antara seluruh panel yang diuji. Konsentrasi CO melonjak cepat hingga mencapai puncak 55,2 ppm pada pukul 15:15:38, kemudian mulai menurun. Seiring turunnya CO, konsentrasi NO₂ justru meningkat tajam hingga 62,7 ppm pada pukul 15:39:56, menunjukkan perubahan komposisi asap pembakaran korek api seiring waktu paparan. Selama seluruh rentang ini, status sistem tetap tercatat *Danger* karena salah satu dari kedua parameter (CO atau NO₂) berada di atas ambang batas.

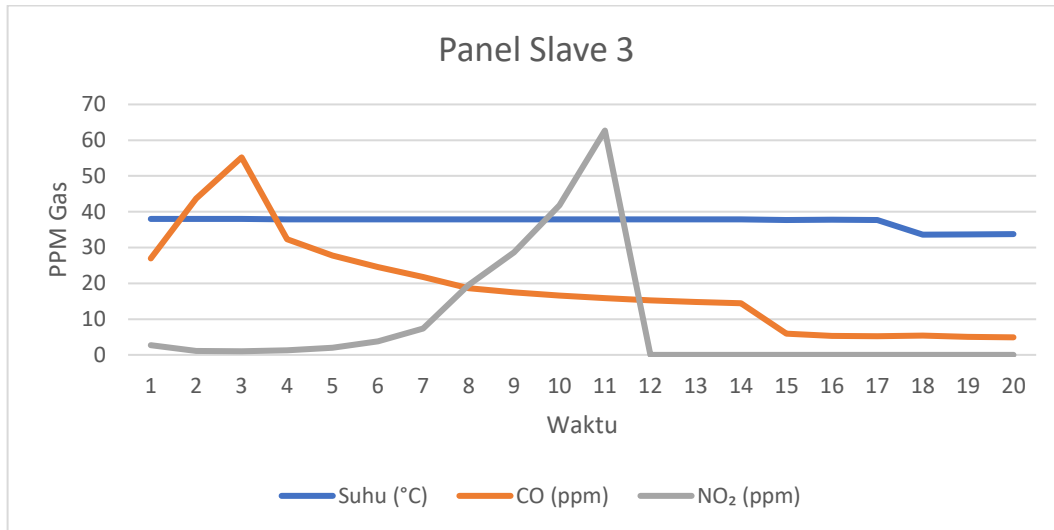
Setelah puncak paparan, kondisi gas berangsur turun secara bertahap. Pada pukul 15:42:59, status berubah menjadi *Warning* saat CO mencapai 15,2 ppm dengan NO₂ sudah kembali ke 0 ppm, sebelum akhirnya stabil pada status *Normal* mulai pukul 15:45:03 dan bertahan hingga akhir sesi pengujian pada pukul 16:06:03.

Tabel 4.22 Data Suhu, Gas CO dan NO₂ Powerpack Panel Slave 3

No	Suhu (°C)	CO (ppm)	NO ₂ (ppm)	Status Output Alat	Menit ke-
1	38	27	2.7	Danger	7/17/2026 15:09
2	38	43.7	1.1	Danger	7/17/2026 15:12
3	38	55.2	1	Danger	7/17/2026 15:15
4	37.9	32.3	1.3	Danger	7/17/2026 15:18
5	37.9	27.8	2	Danger	7/17/2026 15:21
6	37.9	24.5	3.8	Danger	7/17/2026 15:24
7	37.9	21.8	7.4	Danger	7/17/2026 15:27
8	37.9	18.6	19.5	Danger	7/17/2026 15:30
9	37.9	17.5	28.7	Danger	7/17/2026 15:33
10	37.9	16.6	41.8	Danger	7/17/2026 15:36
11	37.9	15.9	62.7	Danger	7/17/2026 15:39
12	37.9	15.2	0	Warning	7/17/2026 15:42
13	37.9	14.8	0	Normal	7/17/2026 15:45
14	37.9	14.4	0	Normal	7/17/2026 15:48
15	37.7	5.9	0	Normal	7/17/2026 15:51
16	37.8	5.3	0	Normal	7/17/2026 15:54
17	37.7	5.2	0	Normal	7/17/2026 15:57
18	33.6	5.4	0	Normal	7/17/2026 16:00
19	33.7	5	0	Normal	7/17/2026 16:03
20	33.8	4.9	0	Normal	7/17/2026 16:06

Pola ini menunjukkan bahwa *rule-based algorithm* mampu menangkap transisi status secara berurutan dan logis, dari *Danger* akibat lonjakan gas, melalui fase *Warning* saat konsentrasi mulai mereda, hingga kembali ke *Normal* saat kondisi udara pulih. Berbeda dengan panel Slave_01 dan Slave_02 yang episodenya lebih singkat atau terputus, panel Slave_03 berhasil merekam keseluruhan siklus transisi tiga status secara utuh dalam satu rangkaian waktu yang berdekatan. Keutuhan siklus ini memberikan gambaran yang paling lengkap mengenai bagaimana sistem merespons sejak awal mula paparan gas hingga kondisi kembali aman, sehingga dapat dijadikan acuan utama dalam menilai kecepatan dan ketepatan waktu respons (*response time*) *rule-based algorithm* secara menyeluruh. Selain itu, munculnya nilai CO tertinggi (55,2 ppm) dan NO₂ tertinggi (62,7 ppm) di antara seluruh panel yang diuji pada panel Slave_03 turut mengindikasikan bahwa titik pemasangan sensor pada panel ini kemungkinan berada paling dekat dengan sumber pemicu gas, sehingga wajar apabila intensitas maupun durasi status

Danger yang tercatat lebih tinggi dibandingkan panel lainnya. Dengan demikian, hasil pengujian pada panel Slave_03 melengkapi bukti bahwa *rule-based algorithm* yang diimplementasikan mampu bekerja secara konsisten dan adaptif terhadap variasi intensitas paparan gas di berbagai titik pemantauan dalam jaringan WSN.



Gambar 4.19 Grafik Fluktuasi Suhu, Gas CO dan NO₂ Panel Slave 3 dalam 1 jam pengujian

Secara keseluruhan, hasil pengujian pada panel Slave_03 memperkuat bukti bahwa *rule-based algorithm* bekerja konsisten dalam mengklasifikasikan tingkat bahaya gas sesuai ambang batas yang dirancang, baik pada kondisi paparan tinggi maupun saat kondisi kembali aman.

Berdasarkan hasil pengujian pada panel Master, Slave_01, Slave_02, dan Slave_03, *rule-based algorithm* yang diimplementasikan pada sistem terbukti mampu mengklasifikasikan status kondisi udara (Normal, Warning, Danger) secara tepat dan konsisten mengikuti perubahan nilai CO dan NO₂ secara *real-time* pada seluruh titik pemantauan dalam jaringan WSN.

BAB 5

KESIMPULAN DAN SARAN

Bab ini merupakan tahapan akhir dari tugas akhir yang berisi kesimpulan dari hasil penelitian yang telah dilakukan serta saran untuk pengembangan penelitian berikutnya berdasarkan topik tugas akhir yang dibuat oleh penulis.

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa :

1. Penelitian ini berhasil merancang dan mengimplementasikan sistem pemantauan gas CO, NO₂, dan suhu berbasis ESP32 yang terintegrasi dengan Wireless Sensor Network (WSN). Sistem ini menghubungkan sensor MQ-135 (NO₂), MQ-7 (CO), dan SHT20 (suhu) pada satu panel Master dan tiga panel Slave yang saling terhubung melalui protokol ESP-NOW, dan terbukti mampu membaca serta memproses data secara terdistribusi dari beberapa titik di ruang *powerpack* sebagaimana ditunjukkan pada hasil pengujian BAB 4.
2. Protokol MQTT berhasil diterapkan sebagai jalur komunikasi antara Raspberry Pi 3 pada panel Master dengan sistem pemantauan pusat, sehingga data hasil pembacaan sensor dari seluruh node WSN dapat diakses secara *real-time*. Selain itu, validasi perhitungan *rule-based algorithm* yang mendasari pengolahan data menghasilkan tingkat error yang sangat kecil, yaitu sebesar 0,025%, berdasarkan perbandingan tiga metode perhitungan (*toolBox* MATLAB, implementasi Arduino IDE, dan perhitungan manual), sehingga membuktikan bahwa proses pengolahan data pada sistem ini memiliki akurasi yang tinggi.
3. Efektivitas sistem dalam memberikan peringatan dini serta mengendalikan kualitas udara di lingkungan kapal turut terbukti melalui hasil pengujian. Peringatan dini ditunjukkan lewat aktivasi *buzzer* ketika status kondisi udara berada pada level Warning maupun Danger, sementara pengendalian kualitas udara terlihat dari pengaturan otomatis kecepatan putaran *exhaust fan* menggunakan modul LSA 002 AC Dimmer, yang merespons sesuai tingkat konsentrasi gas CO dan NO₂ berdasarkan *rule-based algorithm*. Hasil pengujian pada panel Master dan ketiga panel Slave juga menunjukkan bahwa sistem mampu mengklasifikasikan status kondisi udara secara konsisten sesuai Nilai

Ambang Batas (NAB) Permenaker RI Nomor 5 Tahun 2018. Adapun untuk parameter suhu, sensor SHT20 berperan memantau kondisi suhu ruang *powerpack* secara *real-time*, sehingga informasi tersebut turut mendukung operator dalam menilai kondisi lingkungan secara menyeluruh.

5.2 Saran

Berdasarkan hasil penelitian, pembahasan, dan kesimpulan yang telah diperoleh, maka saran yang dapat diberikan untuk penelitian selanjutnya adalah sebagai berikut.

1. Sistem notifikasi pada penelitian ini dapat dikembangkan lebih lanjut dengan mengintegrasikan aplikasi mobile atau SMS gateway, sehingga peringatan dini dapat diterima oleh awak kapal secara lebih cepat meskipun berada di luar jangkauan *monitoring* pusat.
2. Modul SIM800L yang digunakan sebagai jalur komunikasi cadangan mengalami kendala dalam mencari sinyal 2G di dalam ruang *powerpack*. Untuk pengembangan selanjutnya, disarankan menggunakan modul komunikasi dengan dukungan jaringan yang lebih baik seperti modul 4G (contoh: SIM7600), atau menambahkan antena eksternal guna memperkuat penerimaan sinyal.

DAFTAR PUSTAKA

- Abdilah, M. R., & Putri, R. A. (2024). *INFORMASI (Jurnal Informatika dan Sistem Informasi) Volume 16 No.1 / Mei / 2024. 16(1)*, 1–20.
- ANSI/ASHRAE Standard 62.2-2019 Addendum h. (2019). *Ventilation and Acceptable Indoor Air Quality in Residential Buildings*. www.ashrae.org
- Darmawan, M. F. Y. (2025a). Rancang Bangun Alat Monitor Kualitas Udara Berbasis Web Pada Kamar Mesin Kapal Mt. Harsanadi. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2). <https://doi.org/10.23960/jitet.v13i2.6334>
- Darmawan, M. F. Y. (2025b). RANCANG BANGUN ALAT MONITOR KUALITAS UDARA BERBASIS WEB PADA KAMAR MESIN KAPAL MT. HARSANADI. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2). <https://doi.org/10.23960/jitet.v13i2.6334>
- Ikhsan Afif Asrory, Muhammad Shifa, Moch Ali Imron Sya'roni, & Budi Pramono Jati. (2025). Perancangan dan Implementasi Sistem Peringatan Dini Banjir Berbasis IoT dengan ESP32, MQTT, dan Aplikasi Kodular. *Uranus : Jurnal Ilmiah Teknik Elektro, Sains dan Informatika*, 3(3), 54–64. <https://doi.org/10.61132/uranus.v3i3.1056>
- Ilham Ainul Bashir. (2024). *Penerapan Drone dan Internet of Things sebagai Alat Pemantau Pencemaran Udara SO₂ dan NO₂ pada Lingkungan Industri*.
- Kementerian Ketenagakerjaan Republik Indonesia. (2018). *Peraturan Menteri Ketenagakerjaan Nomor 5 Tahun 2018 tentang Keselamatan dan Kesehatan Kerja Lingkungan Kerja*. https://jdihn.go.id/files/218/Permen_5_2018.pdf
- Nursuwars, F. maulana sugiartana. (2023). SISTEM MONITORING KARBON MONOKSIDA DENGAN METODE WIRELESS SENSOR NETWORK (WSN) BERBASIS INTERNET OF THINGS (IoT). *E-JOINT (Electronica and Electrical Journal Of Innovation Technology)*, 4(1), 21–27. <https://doi.org/10.35970/e-joint.v4i1.1904>
- Sarmin, Isnawaty, Aksara, L. B., & Sarita, M. I. (2025). *Sistem Monitoring Lingkungan Berbasis IoT Dan WSN Untuk Mendeteksi Pencemaran Udara*. 3(1), 1–9.

- Sri Surani, B., Dharmawan, A., Asa Bakti, C., & AKI Semarang, U. (2025). Implementasi Mikrokontroler ESP32 Untuk Kontrol Berbasis Pengenalan Suara Dan Cahaya Dalam Sistem Smart Home. *Journal of Information Technology and Computer Science (INTECOMS)*, 8(5), 1375–1387.
- Utami, B. K., Fajri, M. R., & Chaerani, A. D. (2025). Sistem Internet Of Things (IoT) Untuk Pemantauan Kualitas Udara Dalam Ruangan Jurnal Sains dan Teknologi. *Jurnal Sains dan Teknologi*, 02(02), 25–29.

LAMPIRAN

Lampiran 1

Coding

```
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DFRobot_SHT20.h"

// =====
// KONFIGURASI JARINGAN (HOTSPOT HP & VPS AWAN)
// =====

const char *ssid = "NDtechnology 4G";
const char *password = "NDtech1234";

const char *mqtt_vps_server = "202.155.13.220";
const int mqtt_vps_port = 1883;
const char *mqtt_vps_user = "scada_master";
const char *mqtt_vps_pass = "pi123";

WiFiClient espClient;
PubSubClient mqtt(espClient);

// =====
// KONFIGURASI PIN & OBJEK SENSOR
// =====

LiquidCrystal_I2C lcd(0x27, 20, 4);
DFRobot_SHT20 sht20;

#define PIN_MQ7      34
```

```

#define PIN_MQ135    35
#define PIN_DIMMER_PWM 25
#define PIN_DIMMER_ZC 26
#define PIN_BUZZER    27

// --- PARAMETER KALIBRASI SENSOR ---
const float Vc = 3.3;
const float RL = 10000.0;
const float R0_FIXED = 10000.0;
float MQ7_a = 9.7, MQ7_b = -0.55;
float MQ135_a = 1.0, MQ135_b = 1.0;

// =====
// STRUKTUR DATA & VARIABEL KONTROL
// =====
typedef struct struct_message {
    float temp;
    float mq7_val;
    float mq135_val;
} struct_message;

struct_message dataMaster, dataSlave1, dataSlave2, dataSlave3;

int mode_auto = 1;
int buzzer_status = 0;
int dimmer_value = 0;
float fuzzyOutput = 0.0;
String fanStatus = "OFF";

unsigned long lastDisplayTime = 0;
unsigned long lastReadTime = 0;
unsigned long blinkTimer = 0;

```

```

const unsigned long displayInterval = 5000;
const unsigned long readInterval = 2000;

int displayPage = 0;
bool is_warning = false;
bool blinkState = false;
String status_lcd = "Bersih";

// =====
// FUNGSI DASAR MEMBERSHIP (FUZZY)
// =====

float trapmf(float x, float a, float b, float c, float d) {
    if (x < a || x > d) return 0.0;
    if (x >= b && x <= c) return 1.0;
    if (x >= a && x < b) { if (b == a) return 1.0; return (x - a) / (b - a); }
    if (x > c && x <= d) { if (d == c) return 1.0; return (d - x) / (d - c); }
    return 0.0;
}

float trimf(float x, float a, float b, float c) {
    if (x <= a || x >= c) return 0.0;
    if (x == b) return 1.0;
    if (x > a && x < b) return (x - a) / (b - a);
    if (x > b && x < c) return (c - x) / (c - b);
    return 0.0;
}

// =====
// FUNGSI PEMBACAAN & PERHITUNGAN
// =====

float hitungPPM(int pin, float a, float b) {
    long sum = 0;

```

```

for (int i = 0; i < 15; i++) {
    sum += analogRead(pin);
    delay(2);
}
float adc = sum / 15.0;
float vrl = (adc / 4095.0) * Vc;
if (vrl <= 0.05) return 0.0;
float Rs_sensor = ((Vc - vrl) * RL) / vrl;
float ppm = a * pow((Rs_sensor / R0_FIXED), b);
if (!isfinite(ppm) || ppm < 0) ppm = 0.0;
return ppm;
}

// --- LOGIKA 1: RULE BASE (BUZZER & TEKS STATUS) ---
void evaluasiRuleBase(float co, float no2) {
    if (co < 10.0 && no2 < 1.0) {
        status_lcd = "Bersih";
        is_warning = false;
        if (mode_auto == 1) buzzer_status = 0;
    } else if (co >= 10.0 && no2 >= 1.0) {
        status_lcd = "Kotor";
        is_warning = true;
        if (mode_auto == 1) buzzer_status = 1;
    } else if (co >= 10.0 && no2 < 1.0) {
        status_lcd = "CO Kotor";
        is_warning = true;
        if (mode_auto == 1) buzzer_status = 0;
    } else if (co < 10.0 && no2 >= 1.0) {
        status_lcd = "NO2 Kotor";
        is_warning = true;
        if (mode_auto == 1) buzzer_status = 0;
    }
}

```

```

}

// --- LOGIKA 2: FUZZY SUGENO (KECEPATAN DIMMER KIPAS) ---
float evaluateFuzzyExhaustFan(float mq7, float mq135) {
    float mq7Normal = trapmf(mq7, 0, 0, 5, 10);
    float mq7Warning = trimf(mq7, 10, 17.5, 25);
    float mq7Danger = trapmf(mq7, 25, 30, 35, 35);

    float mq135Normal = trapmf(mq135, 0, 0, 0.5, 1.5);
    float mq135Warning = trimf(mq135, 1, 3, 5);
    float mq135Danger = trapmf(mq135, 4.5, 5.5, 6, 6);

    float r1 = mq7Normal * mq135Normal;
    float r2 = mq7Warning * mq135Warning;
    float r3 = mq7Danger * mq135Danger;
    float r4 = mq7Normal * mq135Warning;
    float r5 = mq7Warning * mq135Normal;
    float r6 = mq7Normal * mq135Danger;
    float r7 = mq7Danger * mq135Normal;
    float r8 = mq7Warning * mq135Danger;
    float r9 = mq7Danger * mq135Warning;

    // Karena semua nilai konstanta keluaran (z) adalah 1.0 (kecuali z1 = 0.0)
    // Perhitungannya dapat disederhanakan:
    float numerator = r2 + r3 + r4 + r5 + r6 + r7 + r8 + r9;
    float denominator = r1 + r2 + r3 + r4 + r5 + r6 + r7 + r8 + r9;

    if (denominator == 0) return 0.0;
    return numerator / denominator;
}

// =====

```

```

// CALLBACK KONEKSI & PESAN MQTT DARI VPS
// =====
void callbackMQTT(char *topic, byte *payload, unsigned int length) {
    String incomingMessage = "";
    for (unsigned int i = 0; i < length; i++) incomingMessage += (char)payload[i];
    String t = String(topic);

    StaticJsonDocument<256> doc;
    DeserializationError error = deserializeJson(doc, incomingMessage);
    if (error) return;

    // 1. TANGKAP PERINTAH KONTROL DARI WEB
    if (t == "scada/node/kontrol") {
        if (doc.containsKey("mode_auto")) mode_auto = doc["mode_auto"].as<int>();
        if (doc.containsKey("buzzer")) buzzer_status = doc["buzzer"].as<int>();
        if (doc.containsKey("dimmer")) dimmer_value = doc["dimmer"].as<int>();

        Serial.printf(" ▼ [WEB IN] Eksekusi Perintah -> Mode Auto:%d | Buzzer:%d | Dimmer:%d\n",
            mode_auto, buzzer_status, dimmer_value);
    }

    // 2. TANGKAP/SADAP DATA TELEMETRI DARI PARA SLAVE
    else if (t == "scada/node/telemetri") {
        String id_node = doc["id_node"] | "";

        if (id_node == "Slave_01") {
            dataSlave1.temp = doc["suhu"] | dataSlave1.temp;
            dataSlave1.mq7_val = doc["mq7"] | dataSlave1.mq7_val;
            dataSlave1.mq135_val = doc["mq135"] | dataSlave1.mq135_val;
        } else if (id_node == "Slave_02") {
            dataSlave2.temp = doc["suhu"] | dataSlave2.temp;
        }
    }
}

```

```

    dataSlave2.mq7_val = doc["mq7"] | dataSlave2.mq7_val;
    dataSlave2.mq135_val = doc["mq135"] | dataSlave2.mq135_val;
  } else if (id_node == "Slave_03") {
    dataSlave3.temp = doc["suhu"] | dataSlave3.temp;
    dataSlave3.mq7_val = doc["mq7"] | dataSlave3.mq7_val;
    dataSlave3.mq135_val = doc["mq135"] | dataSlave3.mq135_val;
  }
}
}

// =====
// FUNGSI HUBUNGKAN JARINGAN & MQTT
// =====

void hubungkanJaringan() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.print("Menghubungkan ke Hotspot WiFi...");
    WiFi.begin(ssid, password);
    int attempts = 0;
    while (WiFi.status() != WL_CONNECTED && attempts < 20) {
      delay(500); Serial.print("."); attempts++;
    }
    if (WiFi.status() == WL_CONNECTED) Serial.println(" OK!");
  }

  if (WiFi.status() == WL_CONNECTED && !mqtt.connected()) {
    Serial.print("Koneksi ke Broker VPS Awan...");
    String clientId = "ESP32Master-" + String(random(0xffff), HEX);
    if (mqtt.connect(clientId.c_str(), mqtt_vps_user, mqtt_vps_pass)) {
      Serial.println(" Terhubung!");
      // Berlangganan ke dua topik sekaligus
      mqtt.subscribe("scada/node/kontrol");
      mqtt.subscribe("scada/node/telemetri");
    }
  }
}

```

```

    } else {
        Serial.print(" Gagal, rc="); Serial.print(mqtt.state());
        Serial.println(" Mencoba lagi...");
    }
}
}

// =====
// TAMPILAN ANTARMUKA LCD
// =====

void updateLCD(struct_message data, String label) {
    lcd.clear();
    lcd.setCursor(0, 0); lcd.print(">> " + label + " <<");
    lcd.setCursor(0, 1); lcd.printf("Temp : %.2f C", data.temp);
    lcd.setCursor(0, 2); lcd.printf("CO   : %.2f PPM", data.mq7_val);

    lcd.setCursor(0, 3);
    if (label == "MONITOR MASTER") {
        lcd.printf("NO2: %.1f | Fan: %s", data.mq135_val, fanStatus.c_str());
    } else {
        lcd.printf("NO2: %.1f | %s", data.mq135_val, status_lcd.c_str());
    }
}

// =====
// SETUP & LOOP UTAMA
// =====

void setup() {
    Serial.begin(115200);

    pinMode(PIN_BUZZER, OUTPUT);
    pinMode(PIN_DIMMER_PWM, OUTPUT);

```

```

pinMode(PIN_DIMMER_ZC, INPUT_PULLUP);
digitalWrite(PIN_BUZZER, LOW);
analogWrite(PIN_DIMMER_PWM, 0);

Wire.begin(21, 22);
lcd.init(); lcd.backlight();
lcd.print("Inisialisasi Sistem");
lcd.setCursor(0, 1); lcd.print("1. Sensor & Jaringan...");
sht20.initSHT20();

mqtt.setServer(mqtt_vps_server, mqtt_vps_port);
mqtt.setCallback(callbackMQTT);

lcd.clear();
Serial.println("\n=== SISTEM MASTER (CLOUD DIRECT) SIAP ===");
}

void loop() {
    hubungkanJaringan();
    if (mqtt.connected()) mqtt.loop();

    unsigned long currentMillis = millis();

    // 1. BACA SENSOR MASTER & EVALUASI GABUNGAN (Tiap 2 Detik)
    if (currentMillis - lastReadTime >= readInterval) {
        lastReadTime = currentMillis;

        // Baca sensor
        dataMaster.temp = sht20.readTemperature();
        dataMaster.mq7_val = hitungPPM(PIN_MQ7, MQ7_a, MQ7_b);
        dataMaster.mq135_val = hitungPPM(PIN_MQ135, MQ135_a, MQ135_b);
    }
}

```

```

// Cari nilai gas tertinggi dari gabungan Master & semua Slave
float max_CO = dataMaster.mq7_val;
if (dataSlave1.mq7_val > max_CO) max_CO = dataSlave1.mq7_val;
if (dataSlave2.mq7_val > max_CO) max_CO = dataSlave2.mq7_val;
if (dataSlave3.mq7_val > max_CO) max_CO = dataSlave3.mq7_val;

float max_NO2 = dataMaster.mq135_val;
if (dataSlave1.mq135_val > max_NO2) max_NO2 = dataSlave1.mq135_val;
if (dataSlave2.mq135_val > max_NO2) max_NO2 = dataSlave2.mq135_val;
if (dataSlave3.mq135_val > max_NO2) max_NO2 = dataSlave3.mq135_val;

// --- PERBAIKAN LOGIKA AKTUATOR ---
// 1. Selalu jalankan evaluasi teks LCD & kedip peringatan (Berlaku Auto &
Manual)
evaluasiRuleBase(max_CO, max_NO2);

// 2. Timpa nilai dimmer kipas HANYA jika mode Auto aktif
if (mode_auto == 1) {
    fuzzyOutput = evaluateFuzzyExhaustFan(max_CO, max_NO2);
    dimmer_value = fuzzyOutput * 255;
}

// 3. SELALU perbarui teks Kipas untuk LCD berdasarkan nilai dimmer terkini
(Berlaku Auto & Manual)
if (dimmer_value > 0) {
    fanStatus = "ON-" + String(dimmer_value);
} else {
    fanStatus = "OFF";
}

// Eksekusi aktuator fisik
digitalWrite(PIN_BUZZER, buzzer_status == 1 ? HIGH : LOW);

```

```

analogWrite(PIN_DIMMER_PWM, dimmer_value);
// -----

// Siapkan data JSON Master untuk disetor ke VPS
StaticJsonDocument<256> doc;
doc["id_node"] = "Node_01_Master";
doc["suhu"] = serialized(String(dataMaster.temp, 2));
doc["mq7"] = serialized(String(dataMaster.mq7_val, 2));
doc["mq135"] = serialized(String(dataMaster.mq135_val, 2));
doc["mode_auto"] = mode_auto;
doc["buzzer_status"] = buzzer_status;
doc["dimmer_value"] = dimmer_value;
doc["rssi"] = WiFi.RSSI();

char buffer[256];
serializeJson(doc, buffer);

if (mqtt.connected()) {
    mqtt.publish("scada/node/telemetry", buffer);
    Serial.println("📡 [VPS OUT] Master sukses setor data ke VPS.");
}

// Rangkuman transparan ke Serial Monitor
Serial.println("\n----- DATA SISTEM TERKINI -----");

Serial.printf("🏠 [MASTER ASLI] T: %.1f C | CO: %.2f ppm | NO2: %.2f ppm\n",
    dataMaster.temp, dataMaster.mq7_val, dataMaster.mq135_val);
Serial.printf("📡 [SLAVE 1] T: %.1f C | CO: %.2f ppm | NO2: %.2f ppm\n",
    dataSlave1.temp, dataSlave1.mq7_val, dataSlave1.mq135_val);
Serial.printf("📡 [SLAVE 2] T: %.1f C | CO: %.2f ppm | NO2: %.2f ppm\n",
    dataSlave2.temp, dataSlave2.mq7_val, dataSlave2.mq135_val);

```

```

Serial.printf(" 🚗 [SLAVE 3]   T: %.1f C | CO: %.2f ppm | NO2: %.2f ppm\n",
              dataSlave3.temp, dataSlave3.mq7_val, dataSlave3.mq135_val);

Serial.printf(" ⚙️ [AKTUATOR]   Mode Auto: %d | PWM Dimmer: %d |
Buzzer: %d\n",
              mode_auto, dimmer_value, buzzer_status);

Serial.println("-----");
}

// 2. LOGIKA ROTASI LAYAR LCD (Tiap 5 Detik)
if (currentMillis - lastDisplayTime >= displayInterval) {
    lastDisplayTime = currentMillis;
    switch (displayPage) {
        case 0: updateLCD(dataMaster, "MONITOR MASTER"); displayPage = 1;
break;
        case 1: updateLCD(dataSlave1, "DATA SLAVE 1"); displayPage = 2; break;
        case 2: updateLCD(dataSlave2, "DATA SLAVE 2"); displayPage = 3; break;
        case 3: updateLCD(dataSlave3, "DATA SLAVE 3"); displayPage = 0; break;
    }
}

// 3. EFEK KEDIP BACKLIGHT LCD JIKA STATUS WARNING
if (is_warning) {
    if (currentMillis - blinkTimer >= 500) {
        blinkTimer = currentMillis;
        blinkState = !blinkState;
        if (blinkState) lcd.noBacklight();
        else lcd.backlight();
    }
} else {
    lcd.backlight();
}
}

```

