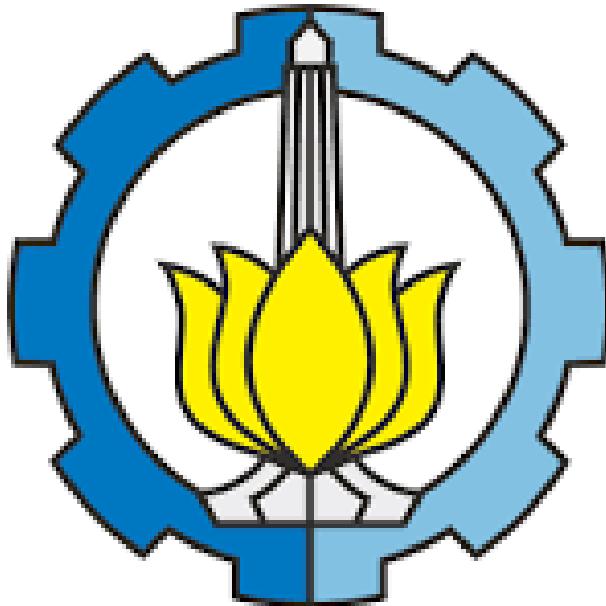


LAPORAN PROJECT-BASED LEARNING

PENGEMBANGAN APLIKASI GUI UNTUK VISUALISASI DATA ELECTRONIC NOSE DENGAN BACKEND RUST, FRONTEND QT PYTHON, DAN INTEGRASI EDGE IMPULSE

Dosen: Ahmad Radhy S.Si, M.Si



Group 4:

Aldi Rochmad Romdoni 2042241050

Adrianta Fajar Surapramana 2042241052

Hanif Raditya Pratama 2042241065

DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2025

Daftar Isi

1 Latar Belakang	2
2 Metodologi	3
2.1 Tujuan Penelitian/Proyek	3
2.1.1 Tujuan Umum	3
2.1.2 Tujuan Khusus	3
2.2 Objek Penelitian dan Karakteristik Sampel	3
2.2.1 Sampel Daun yang Diuji	3
2.2.2 Jumlah Sampel dan Pengujian	4
3 Perancangan Sistem	5
3.1 Akuisisi Hardware	5
3.2 Backend Rust	5
3.3 Frontend Python dengan PyQt6	6
3.4 Visualisasi Data	7
4 Diagram Alur Sistem	8
5 Implementasi	9
5.1 Spesifikasi Sistem	9
5.2 Implementasi Perangkat Keras	9
5.3 Implementasi Protokol Pengujian	11
6 Hasil Pengujian	12
7 Visualisasi Grafik	13
7.1 Percobaan 1 (1 Pompa Purging)	13
7.2 Percobaan 2 (2 Pompa Purging)	15
8 Desain 3D	17
9 Analisis Data	19
10 Kesimpulan	21

1 Latar Belakang

Dalam bidang instrumentasi modern, pemrosesan sinyal dari sensor gas, yang dikenal sebagai electronic nose (e-nose), memiliki peran krusial dalam mendekripsi dan mengklasifikasikan aroma serta komposisi gas dari berbagai sampel uji. Teknologi ini menemukan aplikasi luas dalam bidang industri farmasi, keamanan pangan, monitoring lingkungan, dan diagnosis medis. Untuk memahami integrasi mendalam antara sistem akuisisi data, pemrosesan sinyal, dan visualisasi hasil secara real-time, proyek ini mengembangkan sebuah aplikasi desktop interaktif yang mampu menerima dan memproses data dari multiple sensor gas yang terhubung dengan Arduino Uno R4 WiFi.

Aplikasi ini dirancang dengan arsitektur modern yang memisahkan tanggung jawab antara lapisan backend dan frontend untuk memastikan skalabilitas, maintainability, dan performa optimal. Stack teknologi yang digunakan mencakup:

Rust sebagai backend untuk pemrosesan sinyal, manajemen data sensor, dan logika komputasi yang demand tinggi terhadap performa

Qt Python (PySide6/PyQt6) sebagai framework GUI untuk menghadirkan antarmuka pengguna yang responsif dan intuitif

Edge Impulse sebagai platform untuk pengembangan dan deployment model machine learning yang melakukan analisis dan klasifikasi data sensor secara real-time

Kombinasi teknologi ini memungkinkan pengembangan sistem yang tidak hanya efisien secara komputasional, tetapi juga user-friendly dan mampu diintegrasikan dengan ekosistem IoT modern.

2 Metodologi

2.1 Tujuan Penelitian/Proyek

2.1.1 Tujuan Umum

Mengembangkan sistem electronic nose (eNose) berbasis Arduino Uno R4 WiFi dan aplikasi desktop Rust-PyQt6 yang mampu mendeteksi, membedakan, dan mengklasifikasikan aroma dari empat jenis daun (kemangi, kari, jeruk, dan pandan) menggunakan model machine learning dari Edge Impulse.

2.1.2 Tujuan Khusus

Tujuan khusus dari penelitian ini adalah:

1. Melakukan akuisisi data real-time dari multiple sensor gas terhadap 4 jenis sampel daun dengan 4 kali pengujian per sampel (total 16 pengujian)
2. Mengidentifikasi karakteristik sinyal sensor tiap jenis daun
3. Melatih model machine learning berbasis Edge Impulse untuk klasifikasi jenis daun
4. Membangun aplikasi desktop yang dapat mengidentifikasi jenis daun secara real-time
5. Menganalisis perbedaan pola aroma antar sampel daun

2.2 Objek Penelitian dan Karakteristik Sampel

2.2.1 Sampel Daun yang Diuji

Penelitian ini menggunakan empat jenis daun dengan karakteristik yang berbeda, seperti ditunjukkan pada Tabel 1.

Tabel 1: Karakteristik Aroma Sampel Daun yang Diuji

No.	Nama Sampel	Karakteristik Aroma
1	Daun Kemangi	Aroma segar, pedas, sedikit manis
2	Daun Kari	Aroma kuat, spicy, wangi rempah
3	Daun Jeruk	Aroma citrus, segar, tajam
4	Daun Pandan	Aroma manis, vanilla-like, khas

2.2.2 Jumlah Sampel dan Pengujian

Desain eksperimental dalam penelitian ini dirancang dengan mempertimbangkan replikasi dan validasi data. Parameter pengujian adalah sebagai berikut:

- **Total sampel daun:** 4 jenis
- **Pengujian per sampel:** 4 kali (untuk replikasi dan validasi)
- **Total pengujian:** 16 pengujian
- **Distribusi data:**
 - Training set: 12 pengujian (75%)
 - Test set: 4 pengujian (25%)

Pembagian data ini memastikan bahwa model machine learning dilatih dengan data yang cukup (12 sampel) dan divalidasi dengan data yang terpisah (4 sampel) untuk menghindari overfitting dan memastikan generalisasi yang baik.

3 Perancangan Sistem

Sistem e-nose yang dirancang terdiri dari tiga lapisan utama, yaitu lapisan akuisisi, lapisan pemrosesan dan komunikasi, serta lapisan presentasi. Arsitektur berlapis ini memungkinkan modularitas tinggi dan memfasilitasi pengembangan independen pada setiap komponen tanpa mengganggu keseluruhan sistem.

3.1 Akuisisi Hardware

Pada lapisan akuisisi, sistem dirancang menggunakan modul sensor gas yang membentuk sebuah array untuk menangkap pola aroma dari sampel daun yang ditempatkan di ruang sampel tertutup. Sensor-sensor ini terhubung ke mikrokontroler Arduino Uno R4 WiFi yang bertugas melakukan pembacaan nilai tegangan atau nilai digital yang mewakili respon tiap sensor terhadap sampel. Mikrokontroler kemudian mengolah hasil pembacaan tersebut menjadi data terformat dan mengirimkannya secara berkala ke backend melalui koneksi jaringan nirkabel.

Format data dirancang agar sederhana dan konsisten, misalnya dalam bentuk deret nilai sensor yang disertai informasi waktu pengambilan atau identitas sampel jika diperlukan. Desain ini memungkinkan backend dan GUI untuk menginterpretasikan data dengan mudah, serta mendukung perluasan jumlah sensor atau perubahan konfigurasi tanpa perlu mengubah konsep dasar alur data.

Spesifikasi Sensor:

- Modul sensor gas SEN0440 DFROBOT, Grove Multichannel Gas Sensor
- Arduino Uno R4 WiFi sebagai mikrokontroler utama
- Koneksi WiFi untuk transmisi data ke backend
- Format output: JSON dengan timestamp dan nilai sensor

3.2 Backend Rust

Pada sisi backend, berkas `main.rs` merancang dua mode operasi utama, yaitu mode dummy dan mode normal (WiFi Arduino), yang dipilih melalui argumen saat program dijalankan. Pada mode dummy, backend menjalankan `TcpServer` pada alamat `127.0.0.1:8080` dan menghasilkan data terstruktur menggunakan fungsi `generate_structured_dummy_data` dengan variasi tipe sampel (misalnya “Daun Pandan”, “Daun Kari”, “Daun Kemangi”) dan kombinasi kecepatan motor M1 dan M2 yang disimulasikan; data kemudian disiarkan ke klien GUI melalui mekanisme broadcast.

Pada mode normal, backend mengaktifkan server GUI di port 8080 dan sekaligus membuka listener WiFi Arduino di port 8081 melalui fungsi `start_arduino_receiver`, sehingga data nyata dari perangkat e-nose dapat diteruskan ke GUI tanpa mengubah sisi frontend.

Komponen `TcpServer` di `server.rs` dirancang sebagai server TCP asinkron yang mengelola koneksi banyak klien dengan kanal broadcast `tokio::sync::broadcast`. Server memisahkan penanganan pesan menjadi tiga jenis, yaitu `SensorData`, `Status`, dan `Command`, dan untuk tiap klien menjalankan dua tugas paralel: satu tugas membaca perintah (`command`) dari klien dan meneruskannya sebagai `Message::Command`, dan satu tugas lain mengirimkan paket data bertanda awalan “DATA:” atau “STATUS:” dalam format JSON sesuai yang diharapkan oleh klien Python. Desain ini memungkinkan GUI dan modul lain menerima data sensor dan status sistem secara real-time, sekaligus tetap membuka kemungkinan pengiriman perintah balik dari GUI ke backend.

Fitur Backend:

- Server TCP asinkron dengan support multiple klien
- Dual mode: dummy (simulasi) dan normal (hardware real)
- Broadcast channel untuk distribusi data real-time
- Parsing dan validasi data sensor otomatis
- Command handling untuk kontrol dua arah (GUI ↔ Backend)

3.3 Frontend Python dengan PyQt6

Lapisan komunikasi pada sisi frontend direalisasikan oleh kelas `TcpClient` di `tcp.py` yang diturunkan dari `QThread` dan bertugas menjaga koneksi TCP dengan backend Rust. Kelas ini mengelola soket secara manual, membaca data byte demi byte, menyusunnya dalam buffer baris, kemudian memisahkan paket berdasarkan karakter newline; setiap baris yang diawali “DATA:” diparse menjadi dictionary JSON dan diteruskan melalui sinyal `data_received`, sedangkan baris “STATUS:” diparse dan dikirim sebagai `status_received`. Selain menerima data, `TcpClient` juga menyediakan fungsi `send_command` untuk mengirim perintah string ke backend, yang kelak dapat dimanfaatkan untuk kontrol lebih lanjut terhadap perangkat atau konfigurasi sistem.

Antarmuka grafis utama dirancang di `window.py` melalui kelas `MainWindow`, yang mengatur layout jendela, komponen kontrol, area plot, panel log, serta panel opsional untuk integrasi Edge Impulse. Pada saat inisialisasi, `MainWindow` membuat

objek TcpClient dan DataManager, menyiapkan timer dengan interval 100 ms untuk memperbarui plot, dan mengatur tampilan dengan judul “Electronic Nose Project SPS”, control panel untuk koneksi ke backend di 127.0.0.1:8080, tombol simpan CSV/JSON, dan informasi status koneksi.

Jika TensorFlow tersedia, jendela ini juga menampilkan panel tambahan untuk memuat model .tflite dan melakukan prediksi berdasarkan data sensor terkini, dengan label kelas yang telah ditentukan sebagai ["Jeruk", "Kari", "Kemangi", "Pandan"] .

Fitur Frontend:

- Koneksi TCP multi-thread dengan backend
- Real-time parsing dan buffering data
- Integrasi sinyal/slot PyQt6 untuk event handling
- Support perintah dua arah (send_command)
- Integration dengan TensorFlow untuk inference model

3.4 Visualisasi Data

Area visualisasi real-time dibangun menggunakan pyqtgraph.PlotWidget dengan tujuh kurva yang masing-masing merepresentasikan kanal sensor co_m, eth_m, voc_m, no2, eth_gm, voc_gm, dan co_gm, lengkap dengan legenda, grid, dan skala waktu relatif dalam detik.

Kelas DataManager di data_manager.py mengelola seluruh data sesi, mulai dari penentuan waktu awal, penambahan data baru dengan perhitungan relative_time, penyediaan data ter-downsample untuk plotting, hingga penyimpanan dalam format CSV maupun JSON lengkap dengan metadata nama sampel dan timestamp. Selain itu, DataManager menyediakan fungsi untuk ekspor ke format Edge Impulse (struktur protected, payload, sensors, values), membangun array fitur rata-rata untuk input model, dan menghitung statistik seperti mean, standar deviasi, serta nilai minimum dan maksimum tiap kanal, yang menjadi dasar untuk analisis pada bab hasil pengujian dan analisis data.

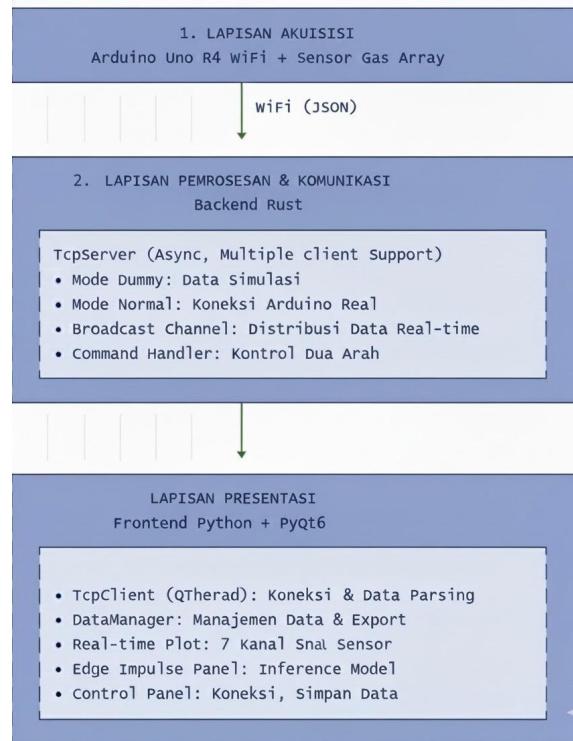
Komponen Visualisasi:

- Real-time plot dengan 7 kanal sensor menggunakan pyqtgraph
- Downsampling otomatis untuk performa optimal
- Legenda, grid, dan label waktu interaktif

- Export ke CSV dan JSON dengan metadata
- Statistik otomatis (mean, std dev, min, max)
- Format export kompatibel dengan Edge Impulse

4 Diagram Alur Sistem

Arsitektur sistem dapat diilustrasikan dalam diagram berikut:



Gambar 1: Diagram Alur Sistem

5 Implementasi

5.1 Spesifikasi Sistem

Implementasi sistem e-nose menggunakan kombinasi perangkat keras dan perangkat lunak yang terintegrasi untuk melakukan akuisisi data gas sensor secara real-time. Spesifikasi perangkat keras yang digunakan meliputi power supply Mean Well LRS-150-12 sebagai sumber daya utama dengan step-down DC-DC converter LM2596 dengan LED indicator untuk mengatur tegangan ke komponen yang memerlukan voltase berbeda. Mikrokontroler Arduino Uno R4 WiFi dipilih sebagai unit pemrosesan utama karena memiliki konektivitas WiFi built-in yang memungkinkan komunikasi dengan backend server. Grove Base Shield digunakan untuk mempermudah koneksi sensor I2C, sedangkan Monster Motor Shield berfungsi mengendalikan kipas DC dan pompa udara dengan kontrol PWM.

Di dalam chamber akrilik, sistem dilengkapi dengan kipas DC berukuran 60x60x25mm untuk mengambil udara (aroma) dari sampel, micro air pump untuk mengeluarkan udara dari dalam chamber, sensor SEN0440 DFRobot (MiCS-5524) untuk deteksi CO, ethanol, dan VOC, serta Grove Multichannel Gas Sensor (GM-XXX) untuk mengukur NO₂, ethanol, VOC, dan CO dengan teknologi electrochemical. Kombinasi dua jenis sensor ini memberikan redundansi dan validasi silang untuk meningkatkan akurasi deteksi.

5.2 Implementasi Perangkat Keras

Perakitan sistem dimulai dengan instalasi komponen di dalam chamber akrilik yang berfungsi sebagai ruang pengukuran tertutup. Kipas DC 60x60x25mm dipasang pada inlet chamber dengan fungsi menarik udara yang mengandung aroma dari sampel daun ke dalam chamber menuju area sensor, sementara micro air pump dipasang pada outlet chamber untuk mengeluarkan udara dari dalam chamber sehingga menciptakan sistem aliran udara yang kontinu. Kedua sensor gas (SEN0440 dan Grove Multichannel) ditempatkan di posisi strategis di antara kipas inlet dan pompa outlet agar mendapat paparan gas yang optimal sebelum udara dibuang keluar.

Koneksi elektronik menggunakan Arduino Uno R4 WiFi sebagai pusat kontrol dengan Grove Base Shield yang terpasang di atasnya untuk mempermudah koneksi I2C ke Grove Multichannel Gas Sensor pada alamat 0x08. Monster Motor Shield dipasang di atas Grove Base Shield untuk mengendalikan kipas dan pompa, dengan konfigurasi pin kipas menggunakan PWM pin 6 dan direction pins 9-10, sedangkan pompa menggunakan PWM pin 5 dan direction pins 7-8. Sensor MiCS-5524 dihu-

bungkan ke pin analog A1 dengan load resistor 820 untuk pembacaan nilai resistansi sensor.

Power management menggunakan PSU Mean Well LRS-150-12 yang menyuplai 12V ke Monster Motor Shield untuk motor driver, kemudian step-down LM2596 menurunkan tegangan menjadi 5V untuk Arduino dan sensor. Indikator LED pada modul LM2596 digunakan untuk monitoring status power supply secara visual.

Implementasi firmware Arduino dirancang menggunakan arsitektur Finite State Machine (FSM) dengan tujuh state utama: IDLE, PRE_COND, RAMP_UP, HOLD, PURGE, RECOVERY, dan DONE. Setiap state memiliki durasi dan kontrol motor yang spesifik untuk mengoptimalkan proses sampling dan pembersihan chamber. State IDLE menunggu perintah START_SAMPLING dari backend atau serial monitor, kemudian transisi ke PRE_COND untuk precondition chamber selama 5 detik dengan kipas berkecepatan sedang (PWM 120) mengambil udara dari sampel.

State RAMP_UP melakukan percepatan kipas secara bertahap menuju kecepatan target level (51-255) dalam 3 detik untuk menghindari turbulensi mendadak yang dapat mengganggu stabilitas sensor. State HOLD adalah fase pengambilan data utama selama 20 detik dimana kipas beroperasi pada kecepatan konstan sesuai level (1-5) untuk menarik aroma masuk ke chamber dan data sensor dikirim ke backend setiap 250ms. State PURGE berdurasi 40 detik dengan mekanisme sequential activation dimana kipas menyala maksimal (PWM 255) dengan arah reverse (membalik fungsi menjadi exhaust untuk mengeluarkan udara) terlebih dahulu selama 5 detik (fan lead time), kemudian pompa menyusul dengan kecepatan maksimal untuk mengeluarkan sisa udara dari chamber secara efektif dan membersihkan chamber dari kontaminan sebelum pengukuran level berikutnya.

Komunikasi WiFi menggunakan library WiFiS3 untuk koneksi ke access point dan socket TCP untuk komunikasi dengan Rust backend pada port 8081. Sistem mengirimkan data dalam format JSON yang berisi timestamp, state name, level, dan nilai dari ketujuh sensor. Fungsi ensureConnected() melakukan auto-reconnect setiap 5 detik jika koneksi terputus untuk menjaga kontinuitas data streaming.

Kalibrasi sensor MiCS-5524 dilakukan sekali saat startup dengan mengukur nilai R0 (resistansi baseline) di udara bersih menggunakan fungsi calculateRs() yang menghitung resistansi sensor berdasarkan tegangan output dengan rumus voltage divider. Konversi resistansi ke ppm menggunakan kurva karakteristik logaritmik yang spesifik untuk setiap gas (CO, ethanol, VOC) dengan persamaan power law yang diturunkan dari datasheet sensor.

5.3 Implementasi Protokol Pengujian

Konfigurasi pengujian dirancang untuk membandingkan efektivitas sistem dengan satu pompa versus dua pompa dalam mempercepat waktu sampling. Percobaan pertama menggunakan satu pompa untuk exhaust dengan durasi total sekitar 30 menit per sampel, sementara percobaan kedua menggunakan dua pompa (kipas sebagai intake dan pompa sebagai exhaust) dengan modifikasi timing untuk mencapai durasi sekitar 6 menit per sampel melalui peningkatan laju aliran udara yang lebih tinggi. Setiap sampel daun (Jeruk, Kari, Kemangi, Pandan) diuji melalui lima level kecepatan kipas yang berbeda untuk mengeksplorasi variasi respons sensor terhadap laju aliran udara masuk.

Data sensor disimpan dalam format CSV dengan kolom timestamp (milliseconds), relative_time (seconds), sample identifier, dan nilai tujuh sensor untuk memudahkan analisis post-processing dan training model machine learning. Serial monitor menampilkan informasi real-time tentang state transition, timing, dan sensor readings setiap 2 detik pada fase HOLD untuk monitoring proses sampling.

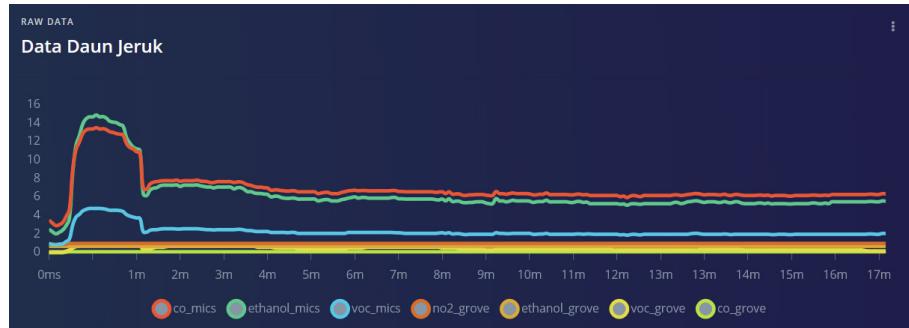
6 Hasil Pengujian

Pengujian pertama menggunakan 1 pompa dengan durasi sekitar 30 menit menunjukkan bahwa setiap jenis daun (Jeruk, Kari, Kemangi, dan Pandan) menghasilkan pola respons yang khas pada ketujuh sensor gas, dengan fase transien di awal pengukuran lalu diikuti kondisi yang lebih stabil. Pada percobaan ini, daun Jeruk memberikan respons tertinggi dan paling jelas pada sensor co_mics dan ethanol_mics dengan puncak besar di awal sebelum turun ke nilai steady-state, sedangkan daun Kari menampilkan pola dinamis dengan beberapa puncak berkala, dan daun Kemangi serta Pandan memberikan respons yang lebih rendah namun relatif stabil. Pola ini menegaskan bahwa kombinasi sensor mampu menangkap perbedaan profil volatil masing-masing daun sehingga dapat digunakan sebagai “fingerprint” aroma di sistem e-nose.

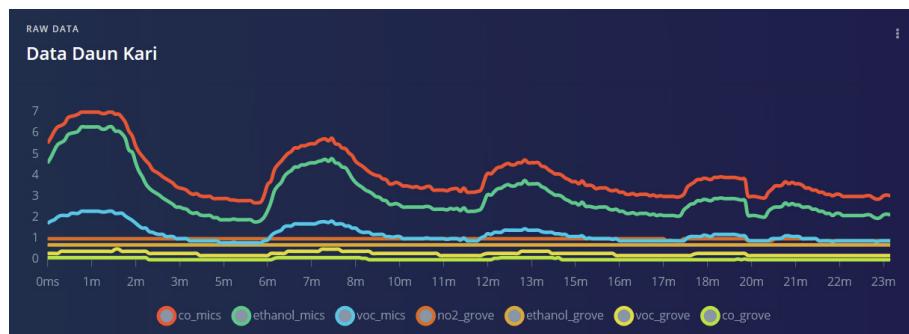
Pada pengujian kedua dengan 2 pompa dan durasi yang dipercepat menjadi sekitar 6 menit, sistem tetap menunjukkan pola respons yang konsisten untuk tiap jenis daun, tetapi waktu total pengukuran berkurang drastis hingga sekitar 80 persen dibandingkan pengujian pertama. Nilai sensor tipe MICS (co_m, eth_m, voc_m) terlihat sangat stabil dengan deviasi standar sangat kecil, menandakan kestabilan dan repeatabilitas yang baik meskipun durasi dipersingkat. Tetapi Sensor tipe Grove (NO₂, Ethanol, VOC, dan CO Grove) kurang memberikan variasi yang cukup besar antar sampel, sehingga klasifikasi kurang terlihat dalam waktu pengukuran yang lebih singkat. Secara keseluruhan, hasil pengujian pertama membuktikan kemampuan sistem dalam membedakan aroma.

7 Visualisasi Grafik

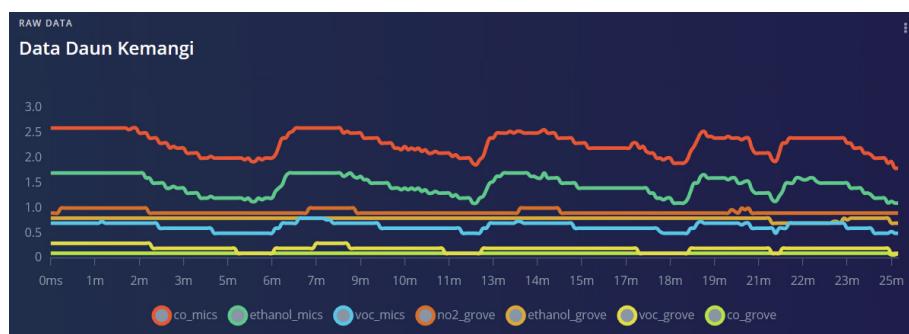
7.1 Percobaan 1 (1 Pompa Purgung)



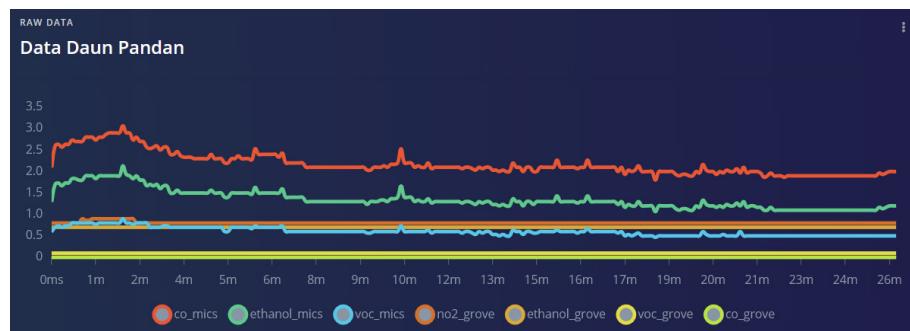
Gambar 2: Visualisasi Grafik Edge Impulse Data Daun Jeruk



Gambar 3: Visualisasi Grafik Edge Impulse Data Daun Kari

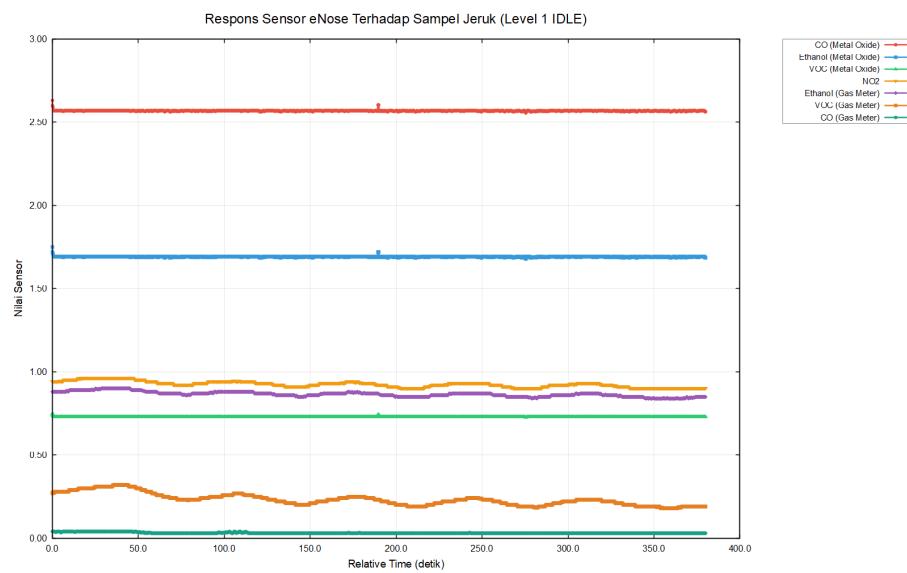


Gambar 4: Visualisasi Grafik Edge Impulse Data Daun Kemangi

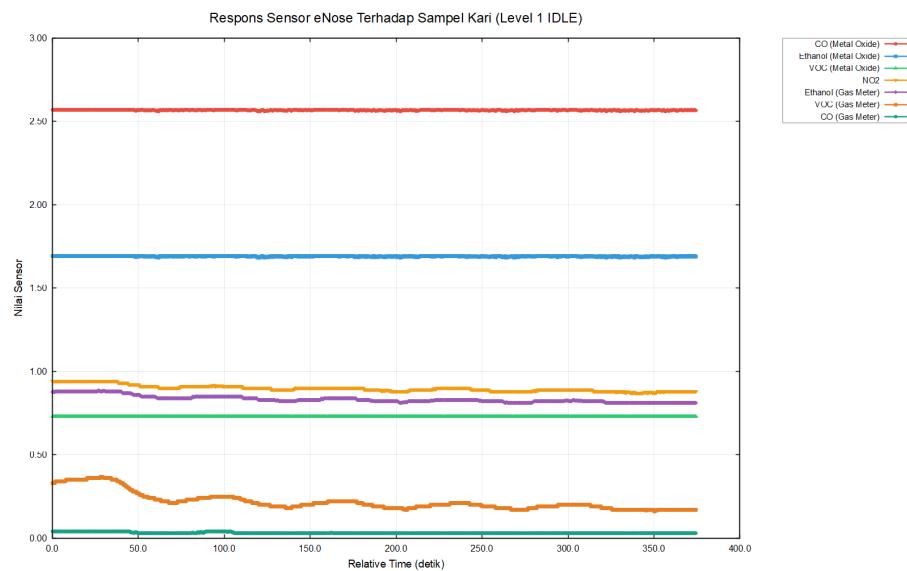


Gambar 5: Visualisasi Grafik Edge Impulse Data Daun Pandan

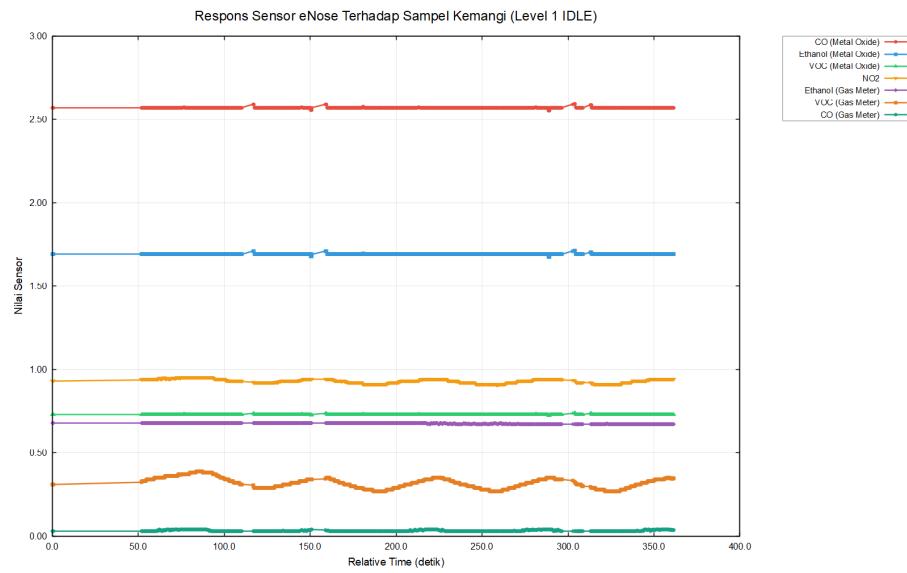
7.2 Percobaan 2 (2 Pompa Purgging)



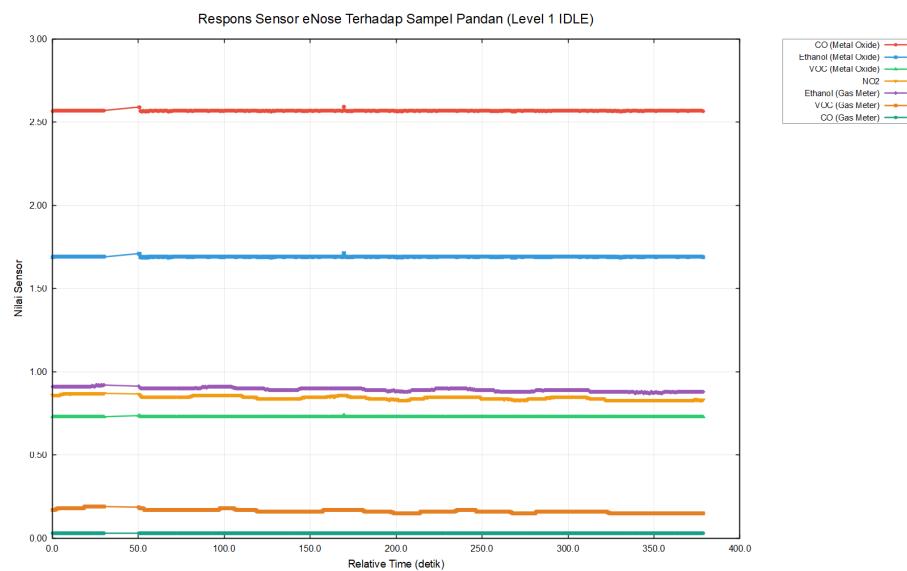
Gambar 6: Visualisasi Grafik GNUPLOT Data Daun Jeruk



Gambar 7: Visualisasi Grafik GNUPLOT Data Daun Kari



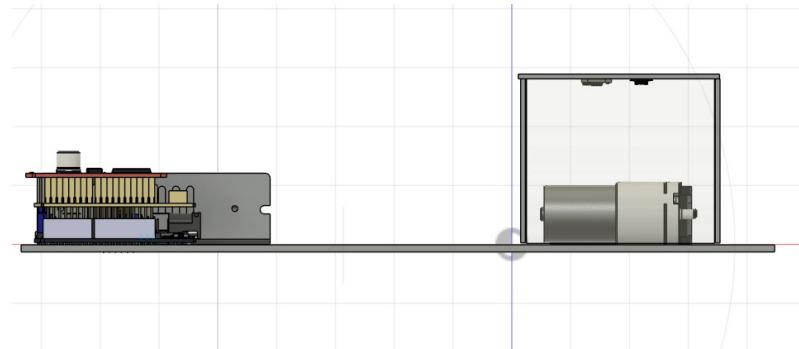
Gambar 8: Visualisasi Grafik GNUPLOT Data Daun Kemangi



Gambar 9: Visualisasi Grafik GNUPLOT Data Daun Pandan

8 Desain 3D

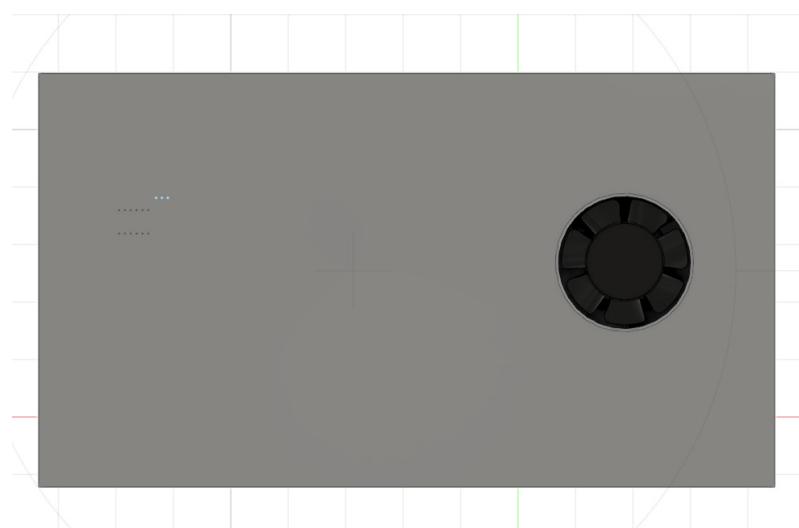
Desain 3D dibuat menggunakan software Autodesk Fusion 360 dengan beberapa model komponen dapat diperoleh di GRABCAD untuk mendapatkan preset model 3D untuk beberapa komponen hardware.



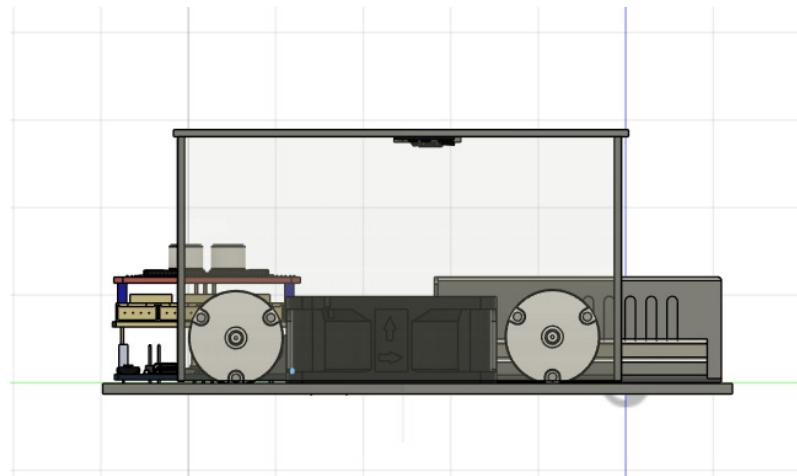
Gambar 10: Desain 3D sistem eNose sisi depan



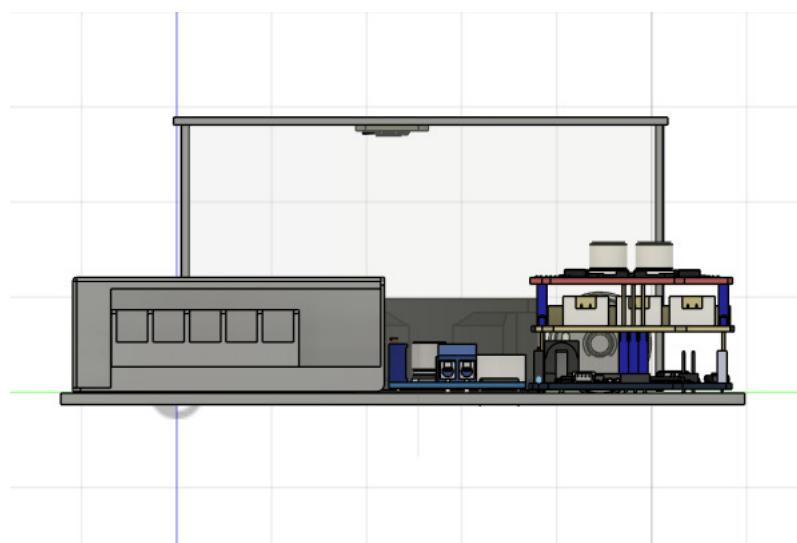
Gambar 11: Desain 3D sistem eNose sisi belakang



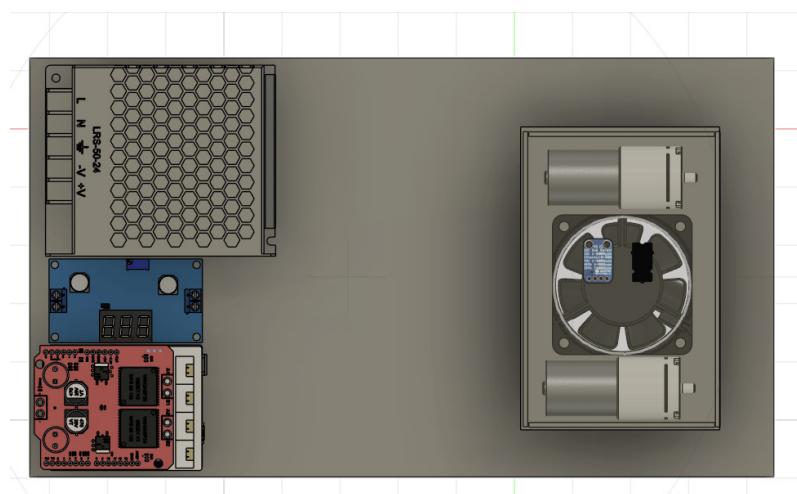
Gambar 15: Desain 3D sistem eNose sisi bawah



Gambar 12: Desain 3D sistem eNose sisi kanan



Gambar 13: Desain 3D sistem eNose sisi kiri



Gambar 14: Desain 3D sistem eNose sisi atas

9 Analisis Data

Pola Temporal Respons Sensor Grafik time-series sensor VOC-GM menunjukkan pola dinamis selama periode pengukuran untuk setiap jenis daun. Pola ini mencerminkan fase-fase pengukuran (hold, purge, recovery) dan memberikan informasi tambahan untuk klasifikasi.

Nama Sampel	Jumlah Data Points	Durasi (detik)	Durasi (menit)	Jumlah Fase
Kemangi	1,296	354.53	5.91	22
Kari	1,360	352.99	5.88	20
Jeruk	1,406	368.52	6.14	22
Pandan	1,429	379.94	6.33	22

Gambar 16: Dataset pengukuran sensor gas terhadap empat jenis daun

Karakteristik temporal:

Daun Jeruk: Menunjukkan respons VOC-GM tertinggi dan paling stabil

Daun Kari: Pola respons tinggi dengan variasi sedang

Daun Kemangi: Respons sedang dengan fluktuasi periodik

Daun Pandan: Respons terendah dan paling stabil sepanjang pengukuran

Fluktuasi dalam respons sensor menunjukkan proses desorpsi dan adsorpsi molekul aroma pada permukaan sensor selama siklus purge-recovery.

Sistem Pengukuran Multi-Fase Setiap pengukuran menggunakan protokol multi-level yang mencakup fase-fase berikut:

PRE-COND: Pra-kondisi sensor

RAMP-UP: Pemanasan sensor

HOLD: Fase pengambilan data stabil

PURGE: Pembersihan sensor

RECOVERY: Fase pemulihan sensor

DONE: Penyelesaian pengukuran

Protokol ini dirancang untuk memastikan respons sensor yang stabil dan dapat direproduksi.

Sensor yang Digunakan Dataset mencakup tujuh saluran sensor gas:

Sensor Metal Oxide (M Series):

co-m: Sensor karbon monoksida

eth-m: Sensor etanol

voc-m: Sensor volatile organic compounds

Sensor Gas Metal Oxide (GM Series):

no2: Sensor nitrogen dioksida

eth-gm: Sensor etanol (tipe GM)

voc-gm: Sensor VOC (tipe GM)

co-gm: Sensor CO (tipe GM)

10 Kesimpulan

Berdasarkan hasil percobaan, sistem eNose mampu mendeteksi gas yang dikeluarkan oleh keempat sampel daun. Namun, grafik keluaran menunjukkan bahwa penggunaan satu pompa dengan durasi pengukuran 30 menit menghasilkan pola sinusoidal yang lebih jelas dibandingkan konfigurasi dua pompa dengan durasi hanya 5 menit. Perbedaan kualitas grafik ini dapat disebabkan oleh beberapa faktor, antara lain perbedaan jumlah pompa, variasi temperatur lingkungan, tingkat kekedapan chamber uji, maupun karakteristik gas dari masing-masing sampel daun.