

PROJECT-BASED LEARNING (PBL)

Mata Kuliah: **Sistem Pengolahan Sinyal**

Semester: Ganjil 2025/2026

Dosen Pengampu: **Ahmad Radhy, S.Si., M.Si.**

Judul Proyek

Pengembangan Aplikasi GUI untuk Visualisasi Data *Electronic Nose* dengan Backend Rust, Frontend Qt Python, dan Integrasi Edge Impulse

Latar Belakang

Dalam bidang instrumentasi modern, pengolahan sinyal dari sensor gas (dikenal sebagai *electronic nose*) memiliki peran penting untuk mendeteksi dan mengklasifikasikan aroma atau komposisi gas dari berbagai sampel uji. Untuk memahami integrasi antara sistem akuisisi data, pemrosesan sinyal, dan visualisasi hasil, mahasiswa akan mengembangkan aplikasi desktop interaktif yang menerima data secara real-time dari **Arduino Uno R4 WiFi** yang terhubung dengan beberapa sensor gas.

Aplikasi ini akan dibangun menggunakan:

- **Rust** sebagai *backend* untuk pemrosesan dan manajemen data sensor.
- **Qt Python (PySide6/PyQt6)** sebagai *frontend GUI*.
- Integrasi dengan **Edge Impulse** untuk analisis berbasis pembelajaran mesin.

Tujuan Pembelajaran

Setelah menyelesaikan proyek ini, mahasiswa diharapkan mampu:

1. Memahami proses akuisisi dan pengolahan sinyal dari sensor gas.
2. Mengimplementasikan komunikasi serial antara Arduino dan aplikasi komputer.
3. Mengembangkan antarmuka GUI berbasis Qt Python yang interaktif dan responsif.
4. Mengintegrasikan Rust sebagai backend pemrosesan data secara efisien.

5. Melakukan klasifikasi sampel uji berbasis data *electronic nose*.
6. Menghubungkan sistem dengan platform Edge Impulse untuk pengenalan pola aroma.

Deskripsi Tugas

Setiap kelompok mahasiswa akan:

- a. Merancang sistem akuisisi data menggunakan **Arduino Uno R4 WiFi** dengan beberapa sensor gas (misal: MEMS gas sensors, atau sejenisnya).
- b. Mengimplementasikan backend dalam **Rust** yang berfungsi untuk:
 - Membaca data serial dari Arduino.
 - Melakukan *filtering* dasar (misalnya: moving average).
 - Menyediakan data terstruktur (misal: melalui TCP socket atau JSON file) yang akan dibaca oleh GUI.
- c. Membuat antarmuka GUI dengan **Qt Python** yang memiliki fitur:
 - Input **nama dan jenis sampel uji** (contoh: “Kopi Arabika”, “Teh Hijau”, “Tembakau Kering”).
 - Tombol kontrol: **Start Sampling**, **Stop**, **Save Data**, dan **Edge Impulse**.
 - Visualisasi data sensor secara real-time dalam bentuk grafik waktu (*time series*) dan indikator status koneksi.
 - Penyimpanan data hasil sampling dalam format CSV atau JSON.
- d. Melakukan proses **sampling data** untuk beberapa sampel berbeda.
- e. Mengintegrasikan dataset hasil sampling dengan **Edge Impulse** untuk analisis lebih lanjut.
- f. Membuat dokumentasi proyek dalam bentuk laporan dan video demonstrasi aplikasi.

Output yang Diharapkan

- Aplikasi GUI fungsional dengan integrasi Rust–Python dan komunikasi serial aktif.
- File dataset hasil pengambilan sampel dari berbagai jenis uji.
- Video demonstrasi penggunaan aplikasi (durasi maksimal 5 menit).

- Laporan akhir dalam format PDF, berisi:
 1. Pendahuluan dan dasar teori singkat.
 2. Desain sistem dan diagram arsitektur perangkat lunak.
 3. Penjelasan komunikasi data serial dan mekanisme backend–frontend.
 4. Hasil uji coba dan tampilan GUI.
 5. Analisis hasil dan pembahasan.
 6. Kesimpulan dan saran pengembangan.

Kriteria Penilaian

Aspek Penilaian	Bobot
Fungsionalitas sistem (komunikasi, visualisasi, real-time)	25%
Desain GUI (estetika, responsivitas, interaksi pengguna)	20%
Implementasi backend Rust (struktur, efisiensi, modularitas)	20%
Kualitas data sampling dan integrasi Edge Impulse	15%
Laporan dan dokumentasi	10%
Presentasi dan video demo	10%

Ketentuan Umum

- Mahasiswa bekerja dalam kelompok (maksimal 3 orang).
- Waktu pengerjaan: 1 minggu.
- Batas pengumpulan: **[isikan tanggal sesuai kalender akademik]**.
- File yang dikumpulkan dalam format ZIP (kode sumber, laporan PDF, video demo).
- Pengumpulan dilakukan melalui Google Drive atau email dosen pengampu.

Petunjuk Teknis

- Gunakan **PySide6** atau **PyQt6** untuk pengembangan GUI.
- Gunakan crate **serialport** pada Rust untuk komunikasi serial.

- Untuk visualisasi real-time, gunakan **PyQtGraph**.
- Komunikasi antara Rust dan Python dapat dilakukan melalui:
 - localhost TCP/UDP socket, atau
 - File JSON yang di-update secara periodik.
- Gunakan Edge Impulse untuk mengunggah dataset hasil pengukuran dan melakukan klasifikasi sederhana.

“Learning by Doing: membangun sistem nyata untuk memahami konsep pengolahan sinyal secara mendalam.”

Selamat berkarya dan bereksperimen!