



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



JUDUL :

WEB-BASED ENVIRONMENTAL MONITORING SYSTEM

PT KAMIKO CIPTA SOLUSI

TUGAS AKHIR

Arya Ramadan

2203321089

**POLITEKNIK
NEGERI
JAKARTA**

PROGRAM STUDI ELEKTRONIKA INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2025



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



SUB JUDUL :

**RANCANG BANGUN SISTEM PEOPLE COUNTING DENGAN
INTEGRASI SENSOR SUHU BME280 DAN MONITORING
SUB-DISTRIBUTION PANEL**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Diploma Tiga**

**POLITEKNIK
NEGERI
JAKARTA**

**Arya Ramadan
2203321089**

PROGRAM STUDI ELEKTRONIKA INDUSTRI

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI JAKARTA

2025



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Arya Ramadan

NIM : 2203321089

Tanda Tangan : 

Tanggal : 24 Juni 2025



**POLITEKNIK
NEGERI
JAKARTA**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN TUGAS AKHIR

Tugas Akhir diajukan oleh :
Nama : Arya Ramadan
NIM : 2203321089
Program Studi : Elektronika Industri
Judul Tugas Akhir : Rancang Bangun Sistem People Counting
dengan Integrasi Sensor Suhu BME280, dan
Monitoring Sub-Distribution Panel

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 24 Juni 2025
dan dinyatakan LULUS

Pembimbing I : Hariyanto, S.Pd., M.T.,
(NIP. 199101282020121008)

(Tanda Tangan)

**POLITEKNIK
NEGERI
JAKARTA**
Depok, 3 Juli 2025

Disahkan Oleh

Ketua Jurusan Teknik Elektro



Dr. Murni Dwiyaniti, S.T., M.T.
NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa atas berkat rahmat-Nya Tugas Akhir ini dapat diselesaikan. Laporan ini disusun sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga Politeknik. Penulis menyadari bahwa penyelesaian Tugas Akhir ini tidak lepas dari dukungan dan bimbingan berbagai pihak selama masa perkuliahan hingga penyusunan tugas akhir. Untuk itu, penulis mengucapkan terima kasih kepada:

1. Dr. Murie Dwiyanti, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Jakarta;
2. Bapak Hariyanto, S.Pd., M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan Tugas Akhir ini;
3. Pak Azizi Akbar Maqfiroh, selaku Pembimbing perusahaan dan pihak PT. Kamiko Cipta Solusi yang telah banyak membantu dalam menyelesaikan Tugas Akhir;
4. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
5. Alverta Orlandia Prijono sebagai partner yang telah membantu penulis dalam menyusun dan menyelesaikan Tugas Akhir;
6. Elsy Fitri, atas semangat dan bantuan yang selalu membangkitkan penulis;

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 24 Juni 2025

Arya Ramadan

NIM. 2203321089



Rancang Bangun Sistem *People Counting* Dengan Integrasi Sensor Suhu BME280, dan Monitoring *Sub-Distribution Panel*

Abstrak

PT Kamiko Cipta Solusi menghadapi tantangan dalam meningkatkan kenyamanan dan efisiensi operasional gedung. Untuk mengatasi masalah tersebut, penelitian ini merancang sebuah sistem pemantauan lingkungan komprehensif berbasis web. Arsitektur sistem mengintegrasikan tiga fungsi utama yang terpusat pada sebuah Mini PC. Pertama, penghitungan orang dan deteksi okupansi ruangan dikembangkan menggunakan model deep learning YOLOv8. Kedua, pemantauan suhu serta kelembaban secara real-time menggunakan sensor BME280 yang terhubung ke ESP32 untuk sekaligus mengotomatisasi kontrol pendingin ruangan (AC). Ketiga, pemantauan panel distribusi listrik (SDP) diimplementasikan melalui gateway Modbus. Seluruh data diproses menggunakan Node-RED, disimpan dalam basis data MySQL, dan divisualisasikan pada dasbor web interaktif. Hasil pengujian menunjukkan kinerja sistem yang solid: fungsi penghitungan orang mencapai akurasi 90%, kontrol AC otomatis memiliki keandalan 96,7%, dan sensor lingkungan terbukti sangat akurat. Meskipun demikian, akurasi deteksi okupansi (70%) dan waktu respons gateway Modbus (rata-rata 2,77 detik) diidentifikasi sebagai area yang dapat dioptimalkan lebih lanjut. Kesimpulannya, sistem yang dikembangkan ini berhasil menjadi solusi fungsional dan terintegrasi untuk mewujudkan manajemen gedung yang lebih cerdas, efisien, dan responsif.

Kata kunci: BME280; Dasbor Web; Deep Learning; ESP32; Manajemen Gedung Cerdas; Modbus; Node-RED; People Counting; Sistem Pemantauan; YOLOv8

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Design and Construction of People Counting System with BME280 Temperature Sensor Integration, and Sub-Distribution Panel Monitoring

Abstract

PT Kamiko Cipta Solusi faces challenges in enhancing building comfort and operational efficiency. To address these issues, this research designs a comprehensive web-based environmental monitoring system. The system architecture integrates three main functions centralized on a Mini PC. First, a people counting and room occupancy detection system was developed using the YOLOv8 deep learning model. Second, real-time temperature and humidity monitoring is performed using a BME280 sensor connected to an ESP32, which also automates air conditioning (AC) control. Third, electrical sub-distribution panel (SDP) monitoring is implemented via a Modbus gateway. All data is processed using Node-RED, stored in a MySQL database, and visualized on an interactive web dashboard. Test results demonstrate the system's solid performance: the people counting function achieved 90% accuracy, the automated AC control showed 96.7% reliability, and the environmental sensors proved to be highly accurate. However, the occupancy detection accuracy (70%) and the Modbus gateway's average response time (2.77 seconds) were identified as areas for further optimization. In conclusion, the developed system provides a functional and integrated solution for achieving smarter, more efficient, and responsive building management.

Keywords: *BME280; Deep Learning; ESP32; Modbus; Monitoring System; Node-RED; People Counting; Smart Building Management; Web Dashboard; YOLOv8*



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	iii
LEMBAR PENGESAHAN TUGAS AKHIR.....	iv
KATA PENGANTAR.....	v
<i>Abstrak</i>	vi
<i>Abstract</i>	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Luaran	3
BAB II TINJAUAN PUSTAKA	4
2.1 Konsep dasar People Counting dan Okupansi	4
2.1.1 Definisi People Counting	4
2.1.2 Penerapan People Counting dalam Sistem Okupansi Ruangan	5
2.2 YOLOv8.....	5
2.3 Modbus.....	6
2.3.1 Tipe Modbus	7
2.4 Message Queue Telemetry Transport (MQTT).....	8
2.5 Node-RED.....	9
2.6 MySQL.....	10
2.7 Mini Personal Computer (PC).....	11
2.8 Closed Circuit Television (CCTV).....	12
2.9 ESP32.....	13
2.10 BME280	14
2.11 Sensor Pemancar IR	15
2.12 Sensor Penerima IR.....	16

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2.13 Modul UART TTL to RS485	18
BAB III PERANCANGAN DAN REALISASI	19
3.1 Perancangan	19
3.1.1 Deskripsi Alat.....	19
3.1.2 Cara Kerja Alat.....	20
3.1.3 Spesifikasi Alat	23
3.1.4 Diagram Blok.....	29
3.1.5 Penempatan Alat.....	31
3.1.6 <i>Flowchart</i>	32
3.2 Realisasi Alat.....	46
3.2.1 Realisasi Sistem Pusat (Integrasi Data, People Counting dan Okupansi)	46
3.2.2 Realisasi Alat Monitoring Suhu, Kelembaban dan Pengaturan Air Conditioner.....	48
3.2.3 Realisasi Alat Monitoring Sub-Ditribution Panel	49
BAB IV PEMBAHASAN.....	51
4.1 Pengujian I: Akurasi Deteksi <i>People Counting</i>	51
4.1.1 Deskripsi Pengujian	51
4.1.2 Prosedur Pengujian	51
4.1.3 Data Hasil Pengujian.....	52
4.1.4 Analisa Data	55
4.2 Pengujian II: Akurasi Okupansi	56
4.2.1 Deskripsi Pengujian	56
4.2.2 Prosedur Pengujian	57
4.2.3 Data Hasil Pengujian.....	57
4.2.4 Analisa Data	58
4.3 Pengujian III: Akurasi Sensor BME280.....	59
4.3.1 Deskripsi Pengujian	59
4.3.2 Prosedur Pengujian	59
4.3.3 Data Hasil Pengujian.....	60
4.3.4 Analisa Data	62
4.4 Pengujian IV: Responsivitas Sistem Kontrol AC.....	63
4.4.1 Deskripsi Pengujian	63
4.4.2 Prosedur Pengujian	63



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

4.4.3 Data Hasil Pengujian.....	64
4.4.4 Analisa Data	65
4.5 Pengujian V: Kinerja <i>Gateway Modbus</i>	66
4.5.1 Dekripsi Pengujian	66
4.5.2 Prosedur Pengujian	66
4.5.3 Data Hasil Pengujian.....	67
4.5.4 Analisa Data	68
BAB V PENUTUP.....	69
5.1 Kesimpulan	69
5.2 Saran.....	70
DAFTAR PUSTAKA	lxxi
LAMPIRAN.....	lxxv
Lampiran 1 Daftar Riwayat Hidup Penulis.....	lxxv
Lampiran 2 Dokumentasi	lxxvi
Lampiran 3 Source Code.....	lxxviii

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 1. 1 Spesifikasi CCTV	12
Tabel 1. 2 Spesifikasi ESP32	14
Tabel 1. 3 Spesifikasi BME280.....	15
Tabel 1. 4 Spesifikasi TSAL6400	16
Tabel 1. 5 Spesifikasi TSOP1838.....	17
Tabel 3. 1 Spesifikasi Software Node-Red	26
Tabel 3. 2 Spesifikasi Hardware.....	27
Tabel 4. 1 Hasil Pengujian Akurasi People Counting	52
Tabel 4. 2 Hasil Pengujian Akurasi Okupansi.....	57
Tabel 4. 3 Hasil Pengujian Akurasi Sensor BME280	61
Tabel 4. 4 Hasil Pengujian Akurasi Kontrol AC	64
Tabel 4. 5 Hasil Pengujian Respon Modbus Gateway	67

**POLITEKNIK
NEGERI
JAKARTA**



DAFTAR GAMBAR

Gambar 2. 1 YOLOv8.....	5
Gambar 2. 2 Modbus.....	6
Gambar 2. 3 MQTT.....	8
Gambar 2. 4 Node-RED.....	9
Gambar 2. 5 MySQL.....	10
Gambar 2. 6 Mini PC	11
Gambar 2. 7 CCTV Hikvision	12
Gambar 2. 8 ESP32.....	13
Gambar 2. 9 BME280	14
Gambar 2. 10 TSAL6400.....	15
Gambar 2. 11 TSOP1838	16
Gambar 2. 12 Modul UART TTL ke RS485	18
Gambar 3. 1 Rangkaian Sistem People Counting dan Okupansi	23
Gambar 3. 2 Rangkaian Sistem Monitoring Suhu dan Kelembaban	24
Gambar 3. 3 Rangkaian Sistem Monitoring SDP	25
Gambar 3. 4 Blok Diagram	29
Gambar 3. 5 Flowchart Utama.....	32
Gambar 3. 6 Flowchart People Counting dan Okupansi 1	35
Gambar 3. 7 Flowchart People Counting dan Okupansi 2.....	36
Gambar 3. 8 Flowchart People Counting dan Okupansi 3.....	37
Gambar 3. 9 Flowchart Monitoring Suhu dan Kelembaban 1	39
Gambar 3. 10 Flowchart Monitoring Suhu dan Kelembaban 2	40
Gambar 3. 11 Flowchart Monitoring Suhu dan Kelembaban 3	41
Gambar 3. 12 Flowchart Monitoring Suhu dan Kelembaban 4	41
Gambar 3. 13 Flowchart Monitoring SDP 1	44
Gambar 3. 14 Flowchart Monitoring SDP 2	45
Gambar 3. 15 Penempatan CCTV People Counting.....	47
Gambar 3. 16 Penempatan CCTV Okupansi	47
Gambar 3. 17 Flow Node-RED.....	47
Gambar 3. 18 Board Alat Monitoring Suhu, Kelembaban dan Pengaturan AC....	48
Gambar 3. 19 Halaman Konfigurasi Alat Monitoring Suhu dan Kelembaban	48

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 3. 20 Halaman Konfigurasi Alat Monitoring Suhu dan Kelembaban 2 ..	49
Gambar 3. 21 Board Alat Monitoring SDP	49
Gambar 3. 22 Halaman Konfigurasi Alat Monitoring SDP	50
Gambar 3. 23 Halaman Konfigurasi Alat Monitoring SDP 2	50
Gambar 4. 1 Pengujian People Counting	52
Gambar 4. 2 Pengujian Okupansi.....	57
Gambar 4. 3 Grafik Pengujian Suhu Sensor BME280.....	60
Gambar 4. 4 Grafik Pengujian Kelembaban Sensor BME280.....	60
Gambar 4. 5 Pengujian Pengaturan AC	64
Gambar 4. 6 Grafik Pengujian Respon Modbus Gateway	67





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup Penulis.....	lxxv
Lampiran 2 Dokumentasi.....	lxxvi
Lampiran 3 Source Code.....	lxxviii



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I PENDAHULUAN

1.1 Latar Belakang

Pada era digital saat ini, teknologi informasi dan komunikasi telah berkembang pesat dan memberi dampak besar terhadap berbagai sektor. Berbagai lokasi publik seperti pusat perbelanjaan, rumah sakit, perpustakaan, dan sejenisnya sering dikunjungi oleh masyarakat umum. Semakin tinggi jumlah pengunjung yang datang dapat menyebabkan kepadatan yang melebihi kapasitas ruangan atau area yang tersedia. Hal ini berpotensi menimbulkan masalah baru karena ketidakseimbangan antara jumlah pengunjung dengan fasilitas yang ada, sehingga mengganggu kenyamanan pengguna lain di lokasi tersebut (Fahmawaty, 2020). Lebih jauh, kepadatan yang tidak terkendali dapat memberi dampak negatif pada masyarakat dan lingkungan, seperti penurunan produktivitas dan peningkatan polusi. Bagi pengelola, pemantauan kondisi lingkungan serta jumlah pengunjung sangat penting untuk menjaga kenyamanan pelanggan, efisiensi operasional, dan pengambilan keputusan yang lebih tepat (Putra et al., 2025).

Salah satu aspek kunci yang memberikan informasi penting adalah data jumlah pengunjung (*people counting*) serta kondisi lingkungan seperti suhu, kelembaban, dan kualitas udara. Untuk mengatasi tantangan ini, diperlukan sistem pemantauan terintegrasi yang mampu menyajikan data konsumsi energi dan Kualitas Lingkungan Dalam Ruangan (*Indoor Environmental Quality* - IEQ) secara komprehensif untuk manajemen gedung yang efektif (Gbouna & Lee, 2022).

Untuk mendukung pemantauan ini, sistem ini menggunakan YOLOv8 (*You Only Look Once version 8*). Model *anchor-free* dengan *decoupled head* memungkinkan YOLOv8 menjalankan *objectness*, klasifikasi, dan regresi secara terpisah (Shyaa & Hashim, 2024). Berfokus pada simpul



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

khusus meningkatkan akurasi model (Shyaa & Hashim, 2024). Sebuah model *deep learning* berbasis *object detection* yang mampu mendeteksi dan menghitung pengunjung secara akurat melalui analisis video *real-time* dari kamera CCTV. Selain itu, sistem ini dilengkapi sensor suhu dan kelembaban yang dipasang di berbagai area ruangan untuk memantau kondisi lingkungan secara *real-time*. Untuk memastikan cakupan yang komprehensif, sistem ini juga terintegrasi dengan pemantauan Sub-Distribution Panel (SDP) yang bertugas mengawasi status jaringan distribusi listrik.

PT Kamiko Cipta Solusi, sebagai perusahaan yang bergerak dalam bidang pengembangan solusi teknologi berbasis web, menyadari kebutuhan ini dalam dunia usaha. Pengelola yang tidak memiliki sistem pemantauan yang efisien sering kali mengalami kesulitan dalam mengelola operasional, termasuk pengaturan kapasitas, pengawasan kenyamanan pelanggan, serta kontrol terhadap faktor-faktor yang dapat mempengaruhi suasana dan kualitas layanan. Oleh karena itu, dibutuhkan sebuah sistem yang dapat memonitor jumlah pengunjung serta kondisi lingkungan secara *real-time* untuk meningkatkan kualitas pelayanan dan pengelolaan sumber daya.

1.2 Perumusan Masalah

Berdasarkan latar belakang, dapat diperoleh perumusan masalah sebagai berikut.

- a) Bagaimana merancang dan mengimplementasikan sistem penghitungan orang (*people counting*) dan deteksi okupansi menggunakan model YOLOv8 untuk menghasilkan data yang akurat secara *real-time*?
- b) Bagaimana mengintegrasikan sensor suhu dan kelembaban BME280 dengan mikrokontroler ESP32 untuk memonitor kondisi ruangan serta mengotomatisasi kontrol perangkat *Air Conditioner* (AC)?
- c) Bagaimana mengimplementasikan fungsionalitas *gateway* Modbus pada mikrokontroler untuk memonitor status *Sub-*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Distribution Panel (SDP) dan mengirimkan datanya ke sistem pusat?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah, ruang lingkup masalah dibatasi pada aspek-aspek berikut:

- a) Sistem kontrol *Air Conditioner* (AC) hanya sebatas mengirimkan perintah *infrared* (IR) berdasarkan variabel suhu yang telah ditentukan (*setpoint*).
- b) Deteksi dan penghitungan orang memanfaatkan model *pre-trained* YOLOv8 tanpa melakukan pelatihan ulang (*retraining*) atau *fine-tuning*.
- c) Pemantauan *Sub-Distribution Panel* (SDP) terbatas pada pembacaan status ON/OFF dari *Miniature Circuit Breaker* (MCB) pada panel sub-distribusi.

1.4 Tujuan

Adapun tujuan yang ingin dicapai melalui penelitian ini adalah:

- a) Merancang dan membangun sistem penghitungan orang (*people counting*) yang akurat berbasis YOLOv8 dan kamera CCTV.
- b) Membangun sistem pemantauan suhu dan kelembaban menggunakan sensor BME280 dan ESP32 yang mampu mengontrol AC secara otomatis.
- c) Mengimplementasikan mikrokontroler sebagai *gateway* Modbus untuk memonitor status panel listrik.

1.5 Luaran

- a) *Source Code*
- b) Laporan Tugas Akhir
- c) Jurnal
- d) Hak Cipta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem pemantauan lingkungan berbasis web, dapat ditarik kesimpulan yang menjawab rumusan masalah penelitian sebagai berikut:

- a) Sistem penghitungan orang (*people counting*) dan deteksi okupansi berbasis YOLOv8 telah berhasil dirancang dan diimplementasikan. Sistem ini mampu mendeteksi pergerakan orang (masuk/keluar) dengan akurasi solid sebesar 90%. Namun, untuk deteksi jumlah orang di dalam ruangan (okupansi), akurasi yang dicapai adalah 70%. Perbedaan performa ini utamanya disebabkan oleh tantangan seperti pergerakan objek yang terlalu cepat dan oklusi (objek terhalang) pada kondisi padat, yang menjadi area untuk optimasi di masa depan.
- b) Integrasi sensor BME280 dengan mikrokontroler ESP32 untuk memonitor lingkungan dan mengontrol AC secara otomatis telah berhasil dilakukan dengan tingkat keandalan yang sangat tinggi. Sensor BME280 terbukti sangat presisi dengan deviasi rata-rata yang sangat rendah ($\pm 0,103$ °C untuk suhu dan $\pm 0,557$ %RH) sehingga datanya valid untuk dijadikan acuan. Sistem kontrol AC berbasis sinyal inframerah juga menunjukkan reliabilitas sebesar 96,7%, membuktikan efektivitas otomasi berdasarkan data sensor.
- c) Implementasi mikrokontroler ESP32 sebagai *gateway* Modbus untuk memonitor status panel listrik (SDP) telah berhasil dilakukan. *Gateway* mampu menjembatani komunikasi antara perangkat di lapangan dengan klien Modbus TCP secara fungsional. Dari hasil pengujian, diperoleh karakteristik kinerja berupa waktu respons rata-rata 2,77 detik, yang menjadi informasi krusial untuk mengonfigurasi interval permintaan data yang efisien dari sisi klien.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Secara keseluruhan, penelitian ini telah berhasil mewujudkan sebuah prototipe sistem pemantauan lingkungan berbasis web yang fungsional dan terintegrasi. Meskipun terdapat ruang untuk optimasi, sistem ini telah membuktikan kelayakannya sebagai solusi untuk mendukung manajemen gedung yang lebih cerdas dan responsif.

5.2 Saran

Untuk pengembangan sistem di masa mendatang, disarankan beberapa tindakan berikut:

- a) **Tingkatkan Akurasi Deteksi Visual:** Performa *people counting* dan okupansi dapat ditingkatkan dengan mengadopsi algoritma pelacakan yang lebih canggih, menempatkan kamera pada posisi *top-down* untuk mengurangi penumpukan, serta melatih ulang model deteksi (YOLO) menggunakan dataset gambar dari lokasi spesifik.
- b) **Pengembangan Fleksibilitas Kontrol AC:** Untuk meningkatkan fleksibilitas dan kompatibilitas sistem kontrol AC, pengembangan selanjutnya disarankan untuk beralih dari metode pengiriman ulang data mentah (*raw data*) ke sistem yang mampu menganalisis dan membangun kembali sinyal inframerah berdasarkan protokol yang digunakan.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Bochkovski, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <http://arxiv.org/abs/2004.10934>
- Chen, Y. S., Cheng, K. H., Hsu, C. S., & Zhang, H. L. (2022). MiniDeep: A Standalone AI-Edge Platform with a Deep Learning-Based MINI-PC and AI-QSR System. *Sensors*, 22(16), 1–35. <https://doi.org/10.3390/s22165975>
- Ery Hartati. (2022). SISTEM INFORMASI TRANSAKSI GUDANG BERBASIS WEBSITE PADA CV. ASYURA. *Klik - Jurnal Ilmu Komputer*, 3(1), 12–18. <https://doi.org/10.56869/klik.v3i1.323>
- Fahmawaty. (2020). Perancangan Alat Penghitung Jumlah Pengunjung Di Perpustakaan Unis Tangerang Menggunakan Sensor Pir Berbasis IoT. *JIMTEK : Jurnal Ilmiah Fakultas Teknik*, 1(3), 253–261. <http://ejournal.unis.ac.id/index.php/jimtek/article/download/1124/Mega/>
- Fauzan, M., Hanafi, U. B., & Irfan, T. (2022). Implementasi TLS Sebagai Metode Keamanan Protokol Jaringan Pada MQTT Berbasis Raspberry PI. *The 13th Industrial Research Workshop and National Seminar*, 13–14.
- Gbouna, Z. V., & Lee, M. (2022). An integrated design of energy and indoor environmental quality monitoring system for effective building performance management. *The Future of Construction in the Context of Digital Transformation and Decarbonization - Proceedings of the 22nd International Conference on Construction Applications of Virtual Reality*, 455–466.
- Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). *YOLOX: Exceeding YOLO Series in 2021*. 1–7. <http://arxiv.org/abs/2107.08430>
- Gunoto, P., Rahmadi, A., & Susanti, E. (2022). Perancangan Alat Sistem Monitoring Daya Panel Surya Berbasis Internet of Things. *Sigma Teknika*, 5(2), 285–294. <https://doi.org/10.33373/sigmateknika.v5i2.4555>



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Harjanto, I. (2020). IoT Gateway Menggunakan Protokol MQTT pada Perangkat Kendali Berbasis Modbus-RTU. *Jurnal Ilmiah Teknosains*, 6(1), 12–19.

Hermiati, R., Asnawati, A., & Kanedi, I. (2021). Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman Php Dan Database Mysql. *Jurnal Media Infotama*, 17(1), 54–66.
<https://doi.org/10.37676/jmi.v17i1.1317>

Hong, C. J., & Mazlan, M. H. (2023). Development of Automated People Counting System using Object Detection and Tracking. *International Journal of Online and Biomedical Engineering*, 19(6), 18–30.
<https://doi.org/10.3991/ijoe.v19i06.38515>

Kartika, K., Harahap, H. F., Asran, A., Jannah, M., Misriana, M., & Suryati, S. (2024). ENERGY AUTOMATION SUPPORTS ENERGY CONSUMPTION INTENSITY IN CLASSROOMS. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3S1). <https://doi.org/10.23960/jitet.v12i3S1.5006>

Nizam, M. N., Haris Yuana, & Zunita Wulansari. (2022). Mikrokontroler Esp 32 Sebagai Alat Monitoring Pintu Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 767–772. <https://doi.org/10.36040/jati.v6i2.5713>

Oberai, M., & Jindal, G. (2021). Intensity Control of LED using IR Remote. ... on *Advances in Science, Technology and ...*.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3888303

Ortiz-Garcés, I., Andrade, R. O., Sanchez-Viteri, S., & Villegas-Ch, W. (2023). Prototype of an Emergency Response System Using IoT in a Fog Computing Environment. *Computers*, 12(4). <https://doi.org/10.3390/computers12040081>

Putra, M. A., Harjoko, A., & Wahyono. (2025). A Systematic Review on Vision-Based Traffic Density Estimation for Intelligent Transportation Systems. *IET Intelligent Transport Systems*, 19(1). <https://doi.org/10.1049/itr2.70038>

Ramadah, F., Wibawa, P. D., & Rizal, A. (2022). Fire Detection System Using Image Processing on Webcam with Yolov3 Method. *E-Proceeding of*



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Engineering, 9(2), 226–231.

Reddy, K. V., Salman, M., & Reddy, A. P. (n.d.). *Raspberry PI Mini PC for Education And Productivity*.

Robi, J. R. (2023). Optimalisasi Fungsi Power Meter Sebagai Sistem Monitoring Dan Early Warning System - Peringatan Dini Pada Titik Kerawanan Di Sistem Kelistrikan Untuk Mencegah Gangguan Dan Meningkatkan Awareness Operator Sistem. *Jurnal Energi Dan Ketenagalistrikan*, 1(1), 47–63. <https://doi.org/10.33322/juke.v1i1.15>

Saptadi, A. H., & Kiswanto, A. (2020). Rancang Bangun Web Server Penampil Data Cuaca Berbasis Arduino Menggunakan Sensor BME280 dan BH1750FVI dengan Tiga Mode Tampilan Data. *Jurnal Teknik Elektro Dan Komputasi (ELKOM)*, 2(2), 112–121. <https://doi.org/10.32528/elkom.v2i2.3516>

Sari, U. L. (2021). Sistem Pendukung Keputusan Penentuan Lokasi Pemasangan CCTV dengan Metode MOORA (Studi Kasus : Dinas Perhubungan Kota Binjai). *Seminar Nasional Informatika (SENATIKA)*, 123–133. <https://ejournal.pelitaindonesia.ac.id/ojs32/index.php/SENATIKA/article/view/1146>

Shyaa, T. A. R., & Hashim, A. A. (2024). Enhancing real human detection and people counting using YOLOv8. *BIO Web of Conferences*, 97, 1–8. <https://doi.org/10.1051/bioconf/20249700061>

Suryana Taryana. (2022). Membangun Stasiun Cuaca dengan BME 280 Untuk Monitoring Suhu, Kelembaban, Tekanan Udara dan Ketinggian. *Komputa*, 1–21. <https://github.com/nodemcu/nodemcu-devkit>

Ullah, R., Hayat, H., Siddiqui, A. A., Siddiqui, U. A., Khan, J., Ullah, F., Hassan, S., Hasan, L., Albattah, W., Islam, M., & Karami, G. M. (2022). A Real-Time Framework for Human Face Detection and Recognition in CCTV Images. *Mathematical Problems in Engineering*, 2022.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

<https://doi.org/10.1155/2022/3276704>

Usha Rani, J., & Raviraj, P. (2023). Real-Time Human Detection for Intelligent Video Surveillance: An Empirical Research and In-depth Review of its Applications. *SN Computer Science*, 4(3), 1–11.

<https://doi.org/10.1007/s42979-022-01654-4>

Utomo, I. T., Firdaus, M. N., Adinandra, S., & Murnani, S. (2024). An Intelligent System for Light and Air Conditioner Control Using YOLOv8. *Journal of Innovation Information Technology and Application (JINITA)*, 6(2), 141–152. <https://doi.org/10.35970/jinita.v6i2.2446>

Wang, C., Zhang, Y., Zhou, Y., Sun, S., Zhang, H., & Wang, Y. (2023). Automatic detection of indoor occupancy based on improved YOLOv5 model. *Neural Computing and Applications*, 35(3), 2575–2599. <https://doi.org/10.1007/s00521-022-07730-3>

**POLITEKNIK
NEGERI
JAKARTA**



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup Penulis



ARYA RAMADAN

Anak pertama dari empat bersaudara, lahir di Jakarta, 01 Desember 2002. Lulus dari SD Muhammadiyah 35 Sear 2015, SMP Negeri 1 Cisoka 2018, SMK Muhammadiyah 1 Jakarta 2021. Gelar diploma tiga (D3) diperoleh tahun 2025 dari Jurusan Teknik Elektro, Program Studi Elektronika Industri, Politeknik Negeri Jakarta.

**POLITEKNIK
NEGERI
JAKARTA**

Lampiran 2 Dokumentasi

Hak Cipta :

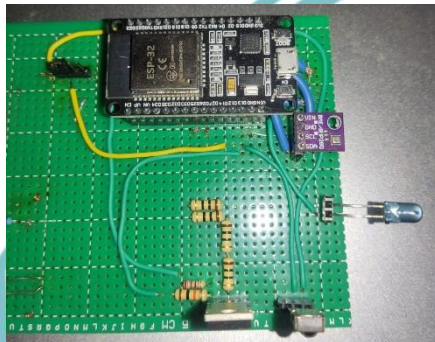
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



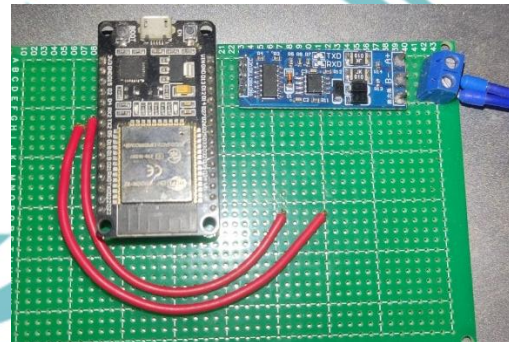
Pengujian Sistem People Counting



Pengujian Sistem Okupansi



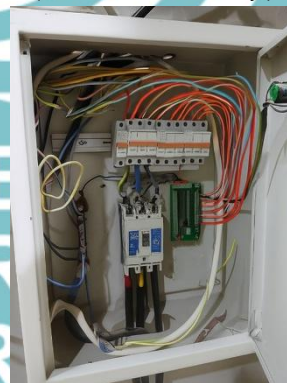
Alat Monitoring Suhu, Kelembaban
dan Pengaturan AC



Alat Monitoring SDP
(Modbus Gateway)



Pengujian Sensor BME280



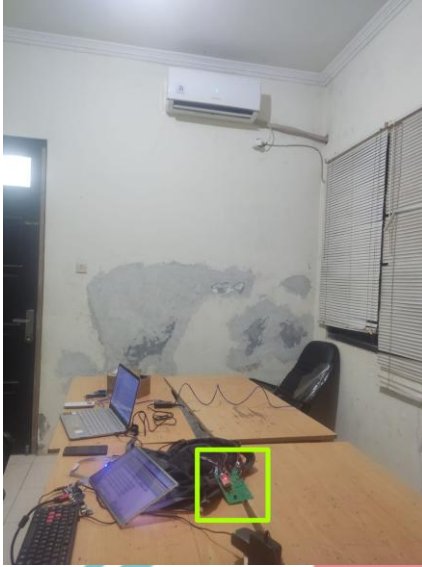
Pengujian Alat Monitoring SDP



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Pengujian Pengaturan Suhu AC



Pengujian Respon Alat Monitoring
SDP

POLITEKNIK
NEGERI
JAKARTA



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Lampiran 3 Source Code

Config.yaml

sources:

```
counting: "rtsp://admin:@192.168.1.14:554/Streaming/Channels/101?tcp"
occupancy: "rtsp://admin:@192.168.1.10:554/Streaming/Channels/101?tcp"
```

mysql:

```
host: "localhost"
user: "root"
password: ""
database: "tugas_akhir_db"
connect_timeout: 5 # Optional: connection timeout in seconds
```

video_processing:

```
buffer_size: 4 # OpenCV VideoCapture buffer size
frame_skip_rate: 1 # 0 = process all frames, 1 = process 1 skip 1, etc.
```

detection:

```
yolo_model_path: 'yolov8s.pt' # Path to your YOLO model file (e.g., yolov8s.pt, yolov8m.pt)
coco_file_path: 'coco.txt' # Path to COCO class names file
confidence_threshold: 0.15 # Minimum confidence for object detection
```

flask_server:

```
host: '0.0.0.0'
port: 8000
debug: False # Set to False for production
use_reloader: False # Set to False when debug is False, especially with threads
```

logging:

```
occupancy_log_interval_seconds: 60 # Atur interval logging data okupansi (dalam detik)
```

line_crossing:

```
line1:
  y: 350
  x1: 10
  x2: 1000
line2:
  y: 450
  x1: 10
  x2: 1200
offset: 50
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

App.py

```
import yaml
from flask import Flask, Response, render_template, request, jsonify, current_app
import cv2
from ultralytics import YOLO
from tracker import Tracker
import cvzone
import datetime
import pymysql
import logging
import time
import threading
import queue
import atexit
import numpy as np

# --- Basic Logging Configuration ---
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(threadName)s - %(name)s - %(message)s'
)
logger = logging.getLogger(__name__)

# --- Load Configuration ---
try:
    with open("config.yaml", 'r') as f:
        cfg = yaml.safe_load(f)
        logger.info("Configuration file 'config.yaml' loaded successfully.")
except FileNotFoundError:
    logger.error("FATAL: config.yaml not found! Please create the configuration file. Exiting.")
    cfg = {}
    exit()
except yaml.YAMLError as e:
    logger.error(f"FATAL: Error parsing config.yaml: {e}. Exiting.")
    cfg = {}
    exit()
except Exception as e:
    logger.error(f"FATAL: An unexpected error occurred while loading config.yaml: {e}. Exiting.")
    cfg = {}
    exit()

# Extract configuration values with defaults
DEFAULT_COUNT_SRC = cfg.get('sources', {}).get('counting', 0)
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
DEFAULT_OCCUPANCY_SRC = cfg.get('sources', {}).get('occupancy', 0)
DB_CFG = cfg.get('mysql', {})
VIDEO_BUFFER_SIZE = cfg.get('video_processing', {}).get('buffer_size', 4)
FRAME_SKIP_RATE = cfg.get('video_processing', {}).get('frame_skip_rate', 0)
YOLO_MODEL_PATH = cfg.get('detection', {}).get('yolo_model_path',
' yolov8s.pt')
COCO_FILE_PATH = cfg.get('detection', {}).get('coco_file_path', 'coco.txt')
CONF_THRESHOLD = cfg.get('detection', {}).get('confidence_threshold', 0.5)

# Configure Line Crossing
frame_width_default_for_lines = cfg.get('video_processing',
{}).get('frame_width', 1020)
default_line_crossing_cfg = {
    'line1': {'y': 194, 'x1': 0, 'x2': frame_width_default_for_lines},
    'line2': {'y': 220, 'x1': 0, 'x2': frame_width_default_for_lines},
    'offset': 6
}
LINE_CROSSING_CFG_RAW = cfg.get('line_crossing',
default_line_crossing_cfg)
GLOBAL_LINE1_CFG = LINE_CROSSING_CFG_RAW.get('line1',
default_line_crossing_cfg['line1'])
GLOBAL_LINE2_CFG = LINE_CROSSING_CFG_RAW.get('line2',
default_line_crossing_cfg['line2'])
GLOBAL_OFFSET = LINE_CROSSING_CFG_RAW.get('offset',
default_line_crossing_cfg['offset'])

logger.info(f'Line Crossing Configuration: L1(Y={GLOBAL_LINE1_CFG['y']},
X1={GLOBAL_LINE1_CFG['x1']}, X2={GLOBAL_LINE1_CFG['x2']}), "
f'L2(Y={GLOBAL_LINE2_CFG['y']},
X1={GLOBAL_LINE2_CFG['x1']}, X2={GLOBAL_LINE2_CFG['x2']}),
Offset={GLOBAL_OFFSET}')

# --- Flask App, YOLO Model, and Global Threading Variables ---
app = Flask(__name__)
model = None

video_sources_cache = {}
video_cache_lock = threading.Lock()

db_log_queue = queue.Queue(maxsize=1000)
stop_db_logger_event = threading.Event()
db_logger_thread_instance = None

try:
    model = YOLO(YOLO_MODEL_PATH)
    logger.info(f'YOLO model ('{YOLO_MODEL_PATH}') loaded successfully.")
except Exception as e:
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
logger.error(f'Failed to load YOLO model from '{YOLO_MODEL_PATH}':  
{e}'. CV functionality will be affected.")
```

```
# --- Database Connection Function ---
```

```
def get_db_connection():
```

```
    if not DB_CFG or not DB_CFG.get('host'):
```

```
        logger_instance = current_app.logger if current_app else logger
```

```
        logger_instance.error("[DB] Database configuration is missing or  
incomplete.")
```

```
        return None
```

```
    try:
```

```
        connection = pymysql.connect(
```

```
            host=DB_CFG.get('host'),
```

```
            user=DB_CFG.get('user'),
```

```
            password=DB_CFG.get('password'),
```

```
            db=DB_CFG.get('database'),
```

```
            cursorclass=pymysql.cursors.DictCursor,
```

```
            connect_timeout=DB_CFG.get('connect_timeout', 5)
```

```
        )
```

```
        return connection
```

```
    except pymysql.Error as e:
```

```
        logger_instance = current_app.logger if current_app else logger
```

```
        logger_instance.error(f'[DB] Failed to connect to the database: {e}')  
        return None
```

```
    except Exception as e:
```

```
        logger_instance = current_app.logger if current_app else logger
```

```
        logger_instance.error(f'[DB] An unexpected error occurred during database  
connection: {e}')  
        return None
```

```
# --- Asynchronous Database Logging ---
```

```
def database_logger_worker():
```

```
    logger.info("Database logger worker thread started.")
```

```
    while not stop_db_logger_event.is_set() or not db_log_queue.empty():
```

```
        try:
```

```
            log_item = db_log_queue.get(timeout=1)
```

```
            log_type = log_item.get('type')
```

```
            data = log_item.get('data')
```

```
        with app.app_context():
```

```
            db = get_db_connection()
```

```
            if not db:
```

```
                logger.error(f'[DB_WORKER] Failed to get DB connection for log  
type: {log_type}.')
```

```
                db_log_queue.task_done()
```

```
                continue
```

```
            try:
```

```
                with db.cursor() as cur:
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        if log_type == 'counting':
            sql = "INSERT INTO people_counting (person_id, direction,
            `timestamp`) VALUES (%s, %s, %s)"
            cur.execute(sql, (data['person_id'], data['direction'],
            data['timestamp']))
        elif log_type == 'occupancy':
            sql = "INSERT INTO occupancy (`count`, `timestamp`) VALUES
            (%s, %s)"
            cur.execute(sql, (data['count'], data['timestamp']))
        else:
            logger.warning(f"[DB_WORKER] Unknown log type:
            {log_type}")
            db_log_queue.task_done()
            continue
        db.commit()
    except pymysql.Error as e:
        logger.error(f"[DB_WORKER] SQL execution failed for {log_type}:
        {e}")
        if db.open: db.rollback()
    except Exception as e:
        logger.error(f"[DB_WORKER] An unexpected error occurred during
        DB operation for {log_type}: {e}")
        if db.open: db.rollback()
    finally:
        if db and db.open: db.close()
        db_log_queue.task_done()
    except queue.Empty:
        continue
    except Exception as e:
        logger.error(f"[DB_WORKER] A critical error occurred in the
        database_logger_worker loop: {e}", exc_info=True)
        time.sleep(1)
    logger.info("Database logger worker thread stopped.")

def log_count_async(person_id, direction):
    if db_log_queue.full():
        logger.warning("DB log queue is full. Counting data may be lost for ID: %s",
        person_id)
        return
    db_log_queue.put({
        'type': 'counting',
        'data': {'person_id': person_id, 'direction': direction, 'timestamp':
        datetime.datetime.now()}
    })

def log_occupancy_async(count):
    if db_log_queue.full():
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
logger.warning("DB log queue is full. Occupancy data may be lost for count:
%s", count)
return
db_log_queue.put({
    'type': 'occupancy',
    'data': {'count': count, 'timestamp': datetime.datetime.now()}
})

# --- Threaded Frame Reader for Video Sources ---
def threaded_frame_reader(source_identifier, video_url_or_index):
    global video_sources_cache
    source_logger = logging.getLogger(f"FrameReader-{source_identifier}")
    source_logger.info(f"Thread started for source: {video_url_or_index}")

    cap = cv2.VideoCapture(video_url_or_index)
    if not cap.isOpened():
        source_logger.error(f"Failed to open video source: {video_url_or_index}")
        with video_cache_lock:
            if source_identifier in video_sources_cache:
                video_sources_cache[source_identifier]['stop_event'].set()
        return

    with video_cache_lock:
        if source_identifier in video_sources_cache:
            video_sources_cache[source_identifier]['capture_object'] = cap

    cap.set(cv2.CAP_PROP_FRAME_WIDTH,          cfg.get('video_processing',
    {})).get('frame_width', 1020))
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT,         cfg.get('video_processing',
    {})).get('frame_height', 500))
    cap.set(cv2.CAP_PROP_BUFFERSIZE, VIDEO_BUFFER_SIZE)

    stop_event = video_sources_cache[source_identifier]['stop_event']
    frame_counter = 0
    consecutive_read_failures = 0

    while not stop_event.is_set():
        if video_sources_cache[source_identifier]['active_clients'] == 0:
            source_logger.info("No active clients. Stopping thread.")
            break

        ret, frame = cap.read()
        if ret:
            consecutive_read_failures = 0
            frame_counter += 1
            if FRAME_SKIP_RATE > 0 and frame_counter % (FRAME_SKIP_RATE
+ 1) != 0:
                time.sleep(0.005)
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
        continue

    with video_sources_cache[source_identifier]['lock']:
        video_sources_cache[source_identifier]['frame'] = frame
    else:
        consecutive_read_failures += 1
        source_logger.warning(f'Failed to read frame (attempt {consecutive_read_failures}).')
        if consecutive_read_failures > 20:
            source_logger.error("Too many consecutive read failures. Attempting to reopen the stream.")
            cap.release()
            cap.open(video_url_or_index)
            if not cap.isOpened():
                source_logger.error("Failed to reopen the stream. Stopping thread.")
                break
            else:
                source_logger.info("Stream successfully reopened.")
                consecutive_read_failures = 0
            time.sleep(0.1)

    cap.release()
    source_logger.info("Thread stopped and video capture released.")
    with video_cache_lock:
        if source_identifier in video_sources_cache:
            video_sources_cache[source_identifier]['capture_object'] = None

def get_or_start_reader_thread(source_identifier, video_url_or_index):
    with video_cache_lock:
        if source_identifier not in video_sources_cache or \
            not video_sources_cache[source_identifier]['thread'] or \
            not video_sources_cache[source_identifier]['thread'].is_alive():

            logger.info(f'No active reader thread for '{source_identifier}'. Starting a new one.')
            source_data = {
                'frame': None,
                'lock': threading.Lock(),
                'active_clients': 0,
                'stop_event': threading.Event(),
                'thread': None,
                'capture_object': None
            }
            video_sources_cache[source_identifier] = source_data

            thread = threading.Thread(target=threaded_frame_reader,
                                     args=(source_identifier, video_url_or_index), daemon=True)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
thread.name = f"FrameReader-{source_identifier}"
video_sources_cache[source_identifier]['thread'] = thread
thread.start()

video_sources_cache[source_identifier]['active_clients'] += 1
video_sources_cache[source_identifier]['stop_event'].clear()
logger.info(f"Client connected. Active clients for '{source_identifier}':
{video_sources_cache[source_identifier]['active_clients']}")
return video_sources_cache[source_identifier]

def release_client_for_source(source_identifier):
    with video_cache_lock:
        if source_identifier in video_sources_cache:
            video_sources_cache[source_identifier]['active_clients'] -= 1
            logger.info(f"Client disconnected. Active clients for '{source_identifier}':
{video_sources_cache[source_identifier]['active_clients']}")
            if video_sources_cache[source_identifier]['active_clients'] <= 0:
                video_sources_cache[source_identifier]['active_clients'] = 0
                logger.info(f"No active clients for '{source_identifier}'. Signaling stop to
the reader thread.")
                video_sources_cache[source_identifier]['stop_event'].set()
            else:
                logger.warning(f"Attempted to release a client for an unknown
source_identifier: {source_identifier}")

# --- Frame Generation Function for Flask Streaming ---
def generate_counting_frames_threaded(source_identifier, video_url_or_index):
    stream_logger = logging.getLogger(f"CountingStream-{source_identifier}")
    stream_logger.info("Stream started.")

    source_data = get_or_start_reader_thread(source_identifier,
video_url_or_index)

    line1_cfg = GLOBAL_LINE1_CFG
    line2_cfg = GLOBAL_LINE2_CFG
    offset_val = GLOBAL_OFFSET
    CONF_THRESHOLD = 0.5

    line1_y, line1_x1_cfg, line1_x2_cfg = line1_cfg['y'], line1_cfg['x1'],
line1_cfg['x2']
    line2_y, line2_x1_cfg, line2_x2_cfg = line2_cfg['y'], line2_cfg['x1'],
line2_cfg['x2']
    stream_logger.info(f"Using Line Configuration:
L1(Y={line1_y},X1={line1_x1_cfg},X2={line1_x2_cfg}),
L2(Y={line2_y},X1={line2_x1_cfg},X2={line2_x2_cfg}), Offset={offset_val}")

    try:
        stream_tracker = Tracker()
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
stream_logger.info("Per-stream tracker initialized.")
except Exception as e:
    stream_logger.error(f'Failed to initialize Tracker for the stream: {e}. Counting
will be affected.")
    stream_tracker = None

if not model or not stream_tracker:
    stream_logger.error("YOLO model or Tracker is not available. Cannot start
the counting stream.")
    error_img_h = cfg.get('video_processing', {}).get('frame_height', 500)
    error_img_w = cfg.get('video_processing', {}).get('frame_width', 1020)
    error_frame_bytes = cv2.imencode('.jpg', cv2.zeros((error_img_h,
error_img_w, 3), dtype='uint8'))[1].tobytes()
    while True:
        yield (b'--frame\r\nContent-Type: image/jpeg\r\n\r\n' + error_frame_bytes +
b'\r\n')
        time.sleep(1)
        if source_data['stop_event'].is_set(): break
    release_client_for_source(source_identifier)
    return

person_down_stream = {}
person_up_stream = {}

in_count_val = 0
out_count_val = 0
frame_process_counter = 0

def draw_offset_area(frame, y_line, x1, x2, offset, color):
    overlay = frame.copy()
    pt1 = (x1, y_line - offset)
    pt2 = (x2, y_line + offset)
    cv2.rectangle(overlay, pt1, pt2, color, -1)
    alpha = 0.2
    return cv2.addWeighted(overlay, alpha, frame, 1 - alpha, 0)

try:
    while not source_data['stop_event'].is_set():
        current_frame_from_cache = None
        with source_data['lock']:
            if source_data['frame'] is not None:
                current_frame_from_cache = source_data['frame'].copy()

        if current_frame_from_cache is None:
            if source_data['thread'] and not source_data['thread'].is_alive():
                stream_logger.info("Reader thread has stopped or no frames available.
Ending stream.")
                break
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
time.sleep(0.01)
continue

frame = current_frame_from_cache
frame_height, frame_width = frame.shape[:2]
frame_process_counter += 1

actual_line1_x1 = min(max(0, line1_x1_cfg), max(0, line1_x2_cfg))
actual_line1_x2 = max(min(frame_width - 1, line1_x1_cfg),
min(frame_width - 1, line1_x2_cfg))
if actual_line1_x1 > actual_line1_x2: actual_line1_x1, actual_line1_x2 =
actual_line1_x2, actual_line1_x1

actual_line2_x1 = min(max(0, line2_x1_cfg), max(0, line2_x2_cfg))
actual_line2_x2 = max(min(frame_width - 1, line2_x1_cfg),
min(frame_width - 1, line2_x2_cfg))
if actual_line2_x1 > actual_line2_x2: actual_line2_x1, actual_line2_x2 =
actual_line2_x2, actual_line2_x1

results = model.predict(frame, verbose=False, classes=[0],
conf=CONF_THRESHOLD)

detections_for_tracker = []
if results and results[0].boxes and hasattr(results[0].boxes, 'data'):
    for box_data in results[0].boxes.data.cpu().numpy():
        x1_b, y1_b, x2_b, y2_b, conf_b, _ = box_data
        detections_for_tracker.append([int(x1_b), int(y1_b), int(x2_b-x1_b),
int(y2_b-y1_b)])

tracked_objects = stream_tracker.update(detections_for_tracker)

for x_obj, y_obj, w_obj, h_obj, obj_id in tracked_objects:
    cx, cy = x_obj + w_obj // 2, y_obj + h_obj // 2

    is_in_line1_zone = (line1_y - offset_val) < cy < (line1_y + offset_val)
    and (actual_line1_x1 <= cx <= actual_line1_x2)
    is_in_line2_zone = (line2_y - offset_val) < cy < (line2_y + offset_val)
    and (actual_line2_x1 <= cx <= actual_line2_x2)
    is_in_counting_zone = is_in_line1_zone or is_in_line2_zone

    if is_in_line1_zone and obj_id not in person_down_stream and obj_id not
    in person_up_stream:
        person_down_stream[obj_id] = cy

    if obj_id in person_down_stream:
        if cy > line2_y and is_in_line2_zone:
            if cy > person_down_stream.get(obj_id, float('-inf')):
                in_count_val += 1
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
log_count_async(obj_id, 'in')
del person_down_stream[obj_id]
if obj_id in person_up_stream: del person_up_stream[obj_id]
elif cy < line1_y - offset_val:
    del person_down_stream[obj_id]

if is_in_line2_zone and obj_id not in person_up_stream and obj_id not in
person_down_stream:
    person_up_stream[obj_id] = cy

if obj_id in person_up_stream:
    if cy < line1_y and is_in_line1_zone:
        if cy < person_up_stream.get(obj_id, float('inf')):
            out_count_val += 1
            log_count_async(obj_id, 'out')
            del person_up_stream[obj_id]
            if obj_id in person_down_stream: del
person_down_stream[obj_id]
        elif cy > line2_y + offset_val:
            del person_up_stream[obj_id]

if is_in_counting_zone:
    cv2.rectangle(frame, (x_obj, y_obj), (x_obj + w_obj, y_obj + h_obj),
(0, 0, 255), 2)
    cvzone.putTextRect(frame, f'ID:{obj_id}', (x_obj, y_obj - 10),
scale=1, thickness=1, offset=3)
    else:
        cv2.circle(frame, (cx, cy), 4, (255, 0, 255), -1)
        cvzone.putTextRect(frame, f'ID:{obj_id}', (cx + 10, cy - 10),
scale=0.8, thickness=1, offset=3, colorR=(255,0,255))

cv2.line(frame, (actual_line1_x1, line1_y), (actual_line1_x2, line1_y), (0,
255, 0), 2)
cv2.line(frame, (actual_line2_x1, line2_y), (actual_line2_x2, line2_y), (0,
255, 255), 2)
frame = draw_offset_area(frame, line1_y, actual_line1_x1,
actual_line1_x2, offset_val, (0, 255, 0))
frame = draw_offset_area(frame, line2_y, actual_line2_x1,
actual_line2_x2, offset_val, (0, 255, 255))
cvzone.putTextRect(frame, f'Keluar: {out_count_val}', (50, 60), scale=2.5,
thickness=3, offset=15, colorR=(0,0,255))
cvzone.putTextRect(frame, f'Masuk: {in_count_val}', (frame.shape[1] -
350, 60), scale=2.5, thickness=3, offset=15, colorR=(0,255,0))

ret_encode, buffer = cv2.imencode('.jpg', frame)
if not ret_encode:
    stream_logger.warning("Failed to encode frame to JPEG.")
    continue
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
frame_bytes = buffer.tobytes()

yield (b'--frame\r\n'
      b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n')

except GeneratorExit:
    stream_logger.info("Client disconnected from the counting stream.")
except Exception as e:
    stream_logger.error(f'An error occurred in the counting stream generator:
{e}', exc_info=True)
finally:
    stream_logger.info(f'Counting stream ended. Frames processed:
{frame_process_counter}')
    release_client_for_source(source_identifier)

def generate_occupancy_frames_threaded(source_identifier, video_url_or_index):
    stream_logger = logging.getLogger(f'OccupancyStream-{source_identifier}')
    stream_logger.info("Stream started.")

    source_data = get_or_start_reader_thread(source_identifier,
video_url_or_index)

    if not model:
        stream_logger.error("YOLO model is not available. Cannot start the
occupancy stream.")
        error_img_h = cfg.get('video_processing', {}).get('frame_height', 500)
        error_img_w = cfg.get('video_processing', {}).get('frame_width', 1020)
        error_frame_bytes = cv2.imencode('.jpg', cv2.zeros((error_img_h,
error_img_w, 3), dtype="uint8"))[1].tobytes()
        while True:
            yield (b'--frame\r\nContent-Type: image/jpeg\r\n\r\n' + error_frame_bytes +
b'\r\n')
            time.sleep(1)
            if source_data['stop_event'].is_set(): break
        release_client_for_source(source_identifier)
        return

    last_logged_count = -1
    log_interval_seconds = cfg.get('logging',
{}).get('occupancy_log_interval_seconds', 5)
    last_log_time = datetime.datetime.min
    frame_process_counter = 0

    try:
        while True:
            current_frame_from_cache = None
            with source_data['lock']:
                if source_data['frame'] is not None:
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
current_frame_from_cache = source_data['frame'].copy()

if current_frame_from_cache is None:
    if source_data['stop_event'].is_set() or \
        (source_data['thread'] and not source_data['thread'].is_alive()):
        stream_logger.info("Reader thread has stopped or no frames available.
Ending stream.")
        break
    time.sleep(0.01)
    continue

frame = current_frame_from_cache
frame_process_counter += 1

results = model.predict(frame, verbose=False, classes=[0],
conf=CONF_THRESHOLD)

current_occupancy_count = 0
if results and results[0].boxes and hasattr(results[0].boxes, 'data'):
    for box_data in results[0].boxes.data.cpu().numpy():
        x1_b, y1_b, x2_b, y2_b, _, _ = box_data
        current_occupancy_count += 1
        cv2.rectangle(frame, (int(x1_b), int(y1_b)), (int(x2_b), int(y2_b)),
(255, 0, 0), 2)

cvzone.putTextRect(frame, f'Okupansi: {current_occupancy_count}', (50,
60), scale=2.5, thickness=3, offset=10)

current_time = datetime.datetime.now()
if (current_time - last_log_time) >=
datetime.timedelta(seconds=log_interval_seconds):
    log_occupancy_async(current_occupancy_count)
    last_log_time = current_time
    last_logged_count = current_occupancy_count
    stream_logger.debug(f'Occupancy recorded:
{current_occupancy_count}')

ret_encode, buffer = cv2.imencode('.jpg', frame)
if not ret_encode:
    stream_logger.warning("Failed to encode frame to JPEG.")
    continue
frame_bytes = buffer.tobytes()

yield (b'--frame\r\n'
      b'Content-Type: image/jpeg\r\n\r\n' + frame_bytes + b'\r\n')

except GeneratorExit:
    stream_logger.info("Client disconnected from the occupancy stream.")
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
except Exception as e:
    stream_logger.error(f'An error occurred in the occupancy stream generator:
{e}', exc_info=True)
finally:
    stream_logger.info(f'Occupancy stream ended. Frames processed:
{frame_process_counter}')
    release_client_for_source(source_identifier)

# --- Flask Routes for Video Feeds ---
@app.route('/counting_feed')
def counting_feed_route():
    src_param = request.args.get('src')
    try:
        video_url_or_index = int(src_param) if src_param and src_param.isdigit() else
        (src_param if src_param else DEFAULT_COUNT_SRC)
    except (ValueError, TypeError):
        video_url_or_index = src_param if src_param else DEFAULT_COUNT_SRC
    source_identifier = str(video_url_or_index)

    current_app.logger.info(f'Request for counting_feed from source_id:
'{source_identifier}' (URL/Index: {video_url_or_index})')
    return Response(generate_counting_frames_threaded(source_identifier,
video_url_or_index),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

@app.route('/occupancy_feed')
def occupancy_feed_route():
    src_param = request.args.get('src')
    try:
        video_url_or_index = int(src_param) if src_param and src_param.isdigit() else
        (src_param if src_param else DEFAULT_OCCUPANCY_SRC)
    except (ValueError, TypeError):
        video_url_or_index = src_param if src_param else
        DEFAULT_OCCUPANCY_SRC
    source_identifier = str(video_url_or_index)

    current_app.logger.info(f'Request for occupancy_feed from source_id:
'{source_identifier}' (URL/Index: {video_url_or_index})')
    return Response(generate_occupancy_frames_threaded(source_identifier,
video_url_or_index),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

# --- API Endpoints ---
def get_time_group_format(interval):
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
if interval == 'day': return '%Y-%m-%d', '%Y-%m-%d'
elif interval == 'month': return '%Y-%m-01', '%Y-%m'
else: return '%Y-%m-%d %H:00:00', '%Y-%m-%d %H:%M'
```

```
@app.route('/api/bme_data')
def api_bme_data():
    start_str = request.args.get('start')
    end_str = request.args.get('end')
    interval = request.args.get('interval', 'hour')
    sensor_type = request.args.get('sensor', 'temperature')

    if not start_str or not end_str:
        return jsonify({"error": "Parameters 'start' and 'end' are required"}), 400
    try:
        start_dt = datetime.datetime.fromisoformat(start_str.replace(" ", "T"))
        end_dt = datetime.datetime.fromisoformat(end_str.replace(" ", "T"))
    except ValueError:
        return jsonify({"error": "Invalid date format. Use ISO format YYYY-MM-DDTHH:MM or YYYY-MM-DD HH:MM"}), 400

    if sensor_type not in ['temperature', 'humidity']:
        return jsonify({"error": "Invalid sensor type. Choose 'temperature' or 'humidity'."}), 400

    value_column = 'temperature' if sensor_type == 'temperature' else 'humidity'
    sql_date_format, label_date_format = get_time_group_format(interval)

    db = get_db_connection()
    if not db: return jsonify({"error": "Database connection failed"}), 500

    labels, values = [], []
    try:
        with db.cursor() as cursor:
            query = f"""
            SELECT    DATE_FORMAT(created_at,    %s)    as    time_slot,
            AVG({value_column}) as avg_value
            FROM bme280 WHERE created_at BETWEEN %s AND %s
            GROUP BY time_slot ORDER BY time_slot ASC
            """
            cursor.execute(query, (sql_date_format, start_dt, end_dt))
            results = cursor.fetchall()
            for row in results:
                try:
                    dt_object = datetime.datetime.strptime(str(row['time_slot']),
                    sql_date_format)
                    labels.append(dt_object.strftime(label_date_format))
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
values.append(round(row['avg_value'], 2) if row['avg_value'] is not
None else 0)
except ValueError as ve:
    current_app.logger.warning(f'BME Data: Error parsing time_slot
'{row['time_slot']}' with format '{sql_date_format}': {ve}')
    continue
    return jsonify(labels=labels, values=values)
except pymysql.Error as e:
    current_app.logger.error(f'[DB_API_BME] Database error: {e}')
    return jsonify({"error": f'Database query failed: {e}'}), 500
except Exception as e:
    current_app.logger.error(f'[API_BME] An unexpected error occurred: {e}',
exc_info=True)
    return jsonify({"error": "An unexpected server error occurred"}), 500
finally:
    if db and db.open: db.close()

@app.route('/api/occupancy_stats')
def api_occupancy_stats():
    start_str = request.args.get('start')
    end_str = request.args.get('end')
    interval = request.args.get('interval', 'hour')

    if not start_str or not end_str: return jsonify({"error": "Parameters 'start' and 'end'
are required"}), 400
    try:
        start_dt = datetime.datetime.fromisoformat(start_str.replace(" ", "T"))
        end_dt = datetime.datetime.fromisoformat(end_str.replace(" ", "T"))
    except ValueError: return jsonify({"error": "Invalid date format."}), 400

    db = get_db_connection()
    if not db: return jsonify({"error": "Database connection failed"}), 500

    labels, values = [], []
    sql_date_format, label_date_format = get_time_group_format(interval)
    try:
        with db.cursor() as cursor:
            query = """
                SELECT DATE_FORMAT(`timestamp`, %s) as time_slot, AVG(`count`)
as avg_occupancy
                FROM occupancy WHERE `timestamp` BETWEEN %s AND %s
                GROUP BY time_slot ORDER BY time_slot ASC
            """
            cursor.execute(query, (sql_date_format, start_dt, end_dt))
            results = cursor.fetchall()
            for row in results:
                try:
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
dt_object = datetime.datetime.strptime(str(row['time_slot']),
sql_date_format)
labels.append(dt_object.strftime(label_date_format))
values.append(round(row['avg_occupancy'], 1) if
row['avg_occupancy'] is not None else 0)
except ValueError as ve:
    current_app.logger.warning(f'Occupancy Data: Error parsing
time_slot '{row['time_slot']}' with format '{sql_date_format}': {ve}')
    continue
return jsonify(labels=labels, values=values)
except pymysql.Error as e:
    current_app.logger.error(f'[DB_API_OCC] Database error: {e}')
    return jsonify({"error": f'Database error: {e}'}, 500)
except Exception as e:
    current_app.logger.error(f'[API_OCC] An unexpected error occurred: {e}',
exc_info=True)
    return jsonify({"error": "An unexpected server error occurred"}), 500
finally:
    if db and db.open: db.close()

@app.route('/api/traffic_data')
def api_traffic_data():
    start_str = request.args.get('start')
    end_str = request.args.get('end')
    interval = request.args.get('interval', 'hour')

    if not start_str or not end_str: return jsonify({"error": "Parameters 'start' and 'end'
are required"}), 400
    try:
        start_dt = datetime.datetime.fromisoformat(start_str.replace(" ", "T"))
        end_dt = datetime.datetime.fromisoformat(end_str.replace(" ", "T"))
    except ValueError: return jsonify({"error": "Invalid date format."}), 400

    db = get_db_connection()
    if not db: return jsonify({"error": "Database connection failed"}), 500

    labels, in_vals, out_vals = [], [], []
    sql_date_format, label_date_format = get_time_group_format(interval)
    try:
        with db.cursor() as cursor:
            query = """
            SELECT DATE_FORMAT(`timestamp`, %s) as time_slot,
            SUM(CASE WHEN direction = 'in' THEN 1 ELSE 0 END) as
in_count,
            SUM(CASE WHEN direction = 'out' THEN 1 ELSE 0 END) as
out_count
            FROM people_counting WHERE `timestamp` BETWEEN %s AND %s
            GROUP BY time_slot ORDER BY time_slot ASC
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
"""
cursor.execute(query, (sql_date_format, start_dt, end_dt))
results = cursor.fetchall()
for row in results:
    try:
        dt_object = datetime.datetime.strptime(str(row['time_slot']),
sql_date_format)
        labels.append(dt_object.strftime(label_date_format))
        in_vals.append(int(row['in_count']) if row['in_count'] is not None else
0)
        out_vals.append(int(row['out_count']) if row['out_count'] is not None
else 0)
    except ValueError as ve:
        current_app.logger.warning(f'Traffic Data: Error parsing time_slot
'{row['time_slot']}' with format '{sql_date_format}': {ve}')
        continue
    return jsonify(labels=labels, in_vals=in_vals, out_vals=out_vals)
except pymysql.Error as e:
    current_app.logger.error(f'[DB_API_TRAFFIC] Database error: {e}')
    return jsonify({"error": f"Database error: {e}"}, 500)
except Exception as e:
    current_app.logger.error(f'[API_TRAFFIC] An unexpected error occurred:
{e}', exc_info=True)
    return jsonify({"error": "An unexpected server error occurred"}), 500
finally:
    if db and db.open: db.close()

@app.route('/api/coil_status')
def api_coil_status():
    db = get_db_connection()
    if not db: return jsonify({"error": "Database connection failed"}), 500

    coils_data = []
    try:
        with db.cursor() as cursor:
            query = """
            SELECT s1.addr_coil, s1.status FROM sdp_coil_status s1
            INNER JOIN (
                SELECT addr_coil, MAX(timestamp) as max_timestamp
                FROM sdp_coil_status WHERE addr_coil BETWEEN 0 AND 7
                GROUP BY addr_coil
            ) s2 ON s1.addr_coil = s2.addr_coil AND s1.timestamp =
s2.max_timestamp
            ORDER BY s1.addr_coil ASC;
            """
            cursor.execute(query)
            results = cursor.fetchall()
            for row in results:
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
coils_data.append({"addr_coil": row['addr_coil'], "status":
bool(row['status'])})
return jsonify(coils=coils_data)
except pymysql.Error as e:
    current_app.logger.error(f"[DB_API_COIL] Database error: {e}")
    return jsonify({"error": f"Database error: {e}"}, 500)
except Exception as e:
    current_app.logger.error(f"[API_COIL] An unexpected error occurred: {e}",
exc_info=True)
    return jsonify({"error": "An unexpected server error occurred"}), 500
finally:
    if db and db.open: db.close()

# --- Main Route for index.html ---
@app.route('/')
def index():
    template_config = {
        'sources': cfg.get('sources', {}),
    }
    if not cfg:
        current_app.logger.warning("Config 'cfg' is empty when rendering
index.html.")
        template_config['sources'] = {'counting': 0, 'occupancy': 0}

    return render_template('index.html', config=template_config)

# --- Graceful Shutdown ---
def shutdown_application():
    logger.info("Initiating application shutdown...")

    logger.info("Signaling stop to all video reader threads...")
    with video_cache_lock:
        for source_id, data in video_sources_cache.items():
            if data.get('thread') and data['thread'].is_alive():
                logger.debug(f"Sending stop signal to the reader for source:
{source_id}")
                data['stop_event'].set()

    time.sleep(1)

    with video_cache_lock:
        active_reader_threads = []
        for source_id, data in list(video_sources_cache.items()):
            if data.get('thread') and data['thread'].is_alive():
                logger.info(f"Waiting for video reader thread '{data['thread'].name}' to
finish...")
                active_reader_threads.append(data['thread'])
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
for thread_to_join in active_reader_threads:
    thread_to_join.join(timeout=5)
    if thread_to_join.is_alive():
        logger.warning(f'Video reader thread '{thread_to_join.name}' did not
stop after the timeout.")
    else:
        logger.info(f'Video reader thread '{thread_to_join.name}' has finished.")

if db_logger_thread_instance and db_logger_thread_instance.is_alive():
    logger.info("Signaling stop to the database logger worker...")
    stop_db_logger_event.set()

logger.info("Waiting for the database log queue to be empty...")
db_log_queue.join()
logger.info("The database log queue is now empty.")

logger.info("Waiting for the database logger worker thread to finish...")
db_logger_thread_instance.join(timeout=10)
if db_logger_thread_instance.is_alive():
    logger.warning("The database logger worker thread did not stop after the
timeout.")
else:
    logger.info("The database logger worker thread has finished.")

logger.info("Application shutdown process complete.")

# --- Main Execution ---
if __name__ == '__main__':
    required_files_check = [COCO_FILE_PATH, YOLO_MODEL_PATH,
'tracker.py']

    db_logger_thread_instance = threading.Thread(target=database_logger_worker,
daemon=True)
    db_logger_thread_instance.name = "DatabaseLoggerWorker"
    db_logger_thread_instance.start()

    atexit.register(shutdown_application)

    flask_host = cfg.get('flask_server', {}).get('host', '0.0.0.0')
    flask_port = cfg.get('flask_server', {}).get('port', 8000)
    flask_debug = cfg.get('flask_server', {}).get('debug', False)
    flask_use_reloader = cfg.get('flask_server', {}).get('use_reloader', False) if
flask_debug else False

    logger.info(f'Starting Flask server at {flask_host}:{flask_port} (Debug:
{flask_debug}, Reloader: {flask_use_reloader})")
    try:
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
app.run(host=flask_host, port=flask_port, debug=flask_debug,  
use_reloader=flask_use_reloader, threaded=True)  
except KeyboardInterrupt:  
    logger.info("Flask server interrupted by user (Ctrl+C).")  
finally:  
    logger.info("The Flask server has stopped.")
```

