

Magnetika

IMPLEMENTASI PENGOLAHAN CITRA MENGGUNAKAN KAMERA UNTUK SISTEM DETEKSI OKUPANSI DI LABORATORIUM OTOMASI INDUSTRI

Fadhil Al Munir¹, Aryuanto Soetedjo², I Komang Somawirata³

Teknik Elektro S1, ITN Malang, Malang Indonesia

¹fadhilalmunir87@gmail.com, ²aryuanto@lecturer.itn.ac.id, ³kmgssomawirata@lecturer.itn.ac.id

Abstrak— Penelitian ini mengembangkan sistem deteksi okupansi ruangan berbasis *computer vision* menggunakan algoritma YOLOv11n pada perangkat edge computing Raspberry Pi 5 di Laboratorium Otomasi Industri. Kamera CCTV dipasang di luar ruangan untuk menjaga privasi pengguna, dan sistem mengintegrasikan algoritma pelacakan SORT serta ByteTrack untuk mendeteksi, melacak, dan menghitung orang yang masuk maupun keluar ruangan. Model dioptimalkan ke format NCNN guna meningkatkan efisiensi pemrosesan data pada perangkat dengan sumber daya terbatas. Hasil pengujian pada tiga skenario video menunjukkan performa deteksi terbaik dicapai oleh kombinasi pelacakan SORT dan format ncnn pada Video 3 dengan F-1 Score sebesar 97,58 persen dan *Precision* 100 persen. Dari sisi pelacakan, kombinasi pelacakan SORT dengan format pt mencapai MOTA tertinggi sebesar 95,27 persen pada Video 3. Untuk akurasi penghitungan okupansi, ByteTrack berhasil mencapai akurasi 100 persen pada Video 1 dan 2. Dari sisi efisiensi komputasi, format NCNN mampu memproses video pada kecepatan 12–14 FPS, jauh lebih unggul dibandingkan format pt yang hanya mencapai 2,5 FPS. Sistem ini terbukti mampu beroperasi secara efisien pada perangkat embedded dengan konsumsi daya rendah tanpa ketergantungan pada server eksternal, mendukung terwujudnya konsep laboratorium cerdas yang adaptif dan berkelanjutan.

Kata Kunci: Deteksi Okupansi, YOLOv11n, Raspberry Pi 5, SORT, ByteTrack

Abstract— This study develops a computer vision-based room occupancy detection system using the YOLOv11n algorithm on a Raspberry Pi 5 edge computing device in the Industrial Automation Laboratory. A CCTV camera was mounted outside the room to preserve user privacy, and the system integrates the SORT and ByteTrack tracking algorithms to detect, track, and count people entering and leaving the room. The model was optimized into the NCNN format to improve data processing efficiency on resource-constrained devices. Testing results on three video scenarios show that the best detection performance was achieved by the combination of SORT tracking and the NCNN format on Video 3, with an F1-Score of 97.58 percent and a precision

of 100 percent. In terms of tracking, the combination of SORT tracking with the PyTorch (pt) format achieved the highest MOTA of 95.27 percent on Video 3. For occupancy counting accuracy, ByteTrack successfully achieved 100 percent accuracy on Videos 1 and 2. In terms of computational efficiency, the NCNN format was able to process video at a speed of 12–14 FPS, far superior to the PyTorch (pt) format, which only reached 2.5 FPS. The system has proven capable of operating efficiently on embedded devices with low power consumption and without dependence on an external server, supporting the realization of an adaptive and sustainable smart laboratory concept.

Keywords: Occupancy Detection, YOLOv11n, Raspberry Pi 5, SORT, ByteTrack

I. PENDAHULUAN

Perkembangan teknologi informasi dan kecerdasan buatan telah mengalami kemajuan pesat, memberikan dampak signifikan terhadap berbagai aspek kehidupan. Salah satu bidang yang mengalami transformasi besar adalah teknologi *computer vision*, yang memungkinkan mesin untuk "melihat" dan memahami informasi visual selayaknya manusia [1]. Kemampuan *computer vision* ini telah dimanfaatkan secara luas dalam berbagai aplikasi, mulai dari sistem keamanan, analisis video, hingga proses otomatisasi industri [2].

Perkembangan teknologi digital tersebut memicu peningkatan aktivitas di berbagai fasilitas publik, termasuk kampus, perkantoran, dan laboratorium. Hal ini menimbulkan kebutuhan mendesak akan pengelolaan ruang yang efisien dan adaptif. Namun, masalah sosial dan operasional yang sering terjadi adalah sulitnya memantau tingkat okupansi ruang secara akurat. Ketidadaan data okupansi menyebabkan kerugian, terutama pemborosan energi, penurunan efisiensi operasional, dan kesulitan dalam pengambilan keputusan manajerial terkait alokasi sumber daya. Dalam konteks akademik seperti Laboratorium Otomasi Industri, masalah ini

mencerminkan kebutuhan akan sistem secara cerdas untuk mencapai konsep smart laboratory yang berkelanjutan.

Sistem deteksi okupansi secara umum dibedakan menjadi dua pendekatan, yaitu pendekatan konvensional dan pendekatan berbasis visual. Sistem konvensional biasanya menggunakan sensor Passive Infrared (PIR), sensor ultrasonik dan sensor CO_2 untuk mendeteksi keberadaan manusia. Namun, metode konvensional ini memiliki keterbatasan dalam mendeteksi jumlah individu secara spesifik, tidak mampu membedakan arah pergerakan serta rentan terhadap gangguan dari lingkungan [3], [4], [5]. Sementara itu, pendekatan berbasis visual (*computer vision*) memungkinkan sistem untuk beradaptasi secara dinamik berdasarkan data visual. Penerapan *computer vision* dapat mendeteksi keberadaan manusia melalui kamera serta mampu memberikan informasi spasial dan temporal secara lebih kaya, sehingga tidak hanya mendeteksi keberadaan, tetapi juga menghitung jumlah orang dalam ruangan [6], [7], [8].

Penggunaan *computer vision* juga memiliki masalahnya tersendiri karena objek dalam hal ini adalah manusia akan mampu diamati secara visual oleh kamera sehingga dapat mengganggu privasi seseorang. Oleh karena itu, letak kamera yang dipasang diluar ruangan diharapkan mampu menghilangkan pelanggaran privasi individu karena tidak mendeteksi kegiatan didalam ruangan. Pada [9] telah berhasil merancang sistem untuk deteksi objek dan penghitung okupansi ruangan dengan letak kamera berada pada luar ruangan. Hasil penelitian tersebut menunjukkan hasil deteksi dengan akurasi tertinggi sebesar 82.76%. Namun, sistem akan mengalami penurunan akurasi signifikan pada kondisi ramai atau ketika objek mengalami tumpang tindih. Hasil dari penelitian tersebut hanya mendeteksi orang keluar masuk ruangan dengan hanya gerakan maju mundur dan tidak mendeteksi pergerakan objek secara lebih luas. Penelitian tersebut juga masih menggunakan YOLOv3 dan DeepSORT yang dijalankan pada server komputasi sehingga memiliki ketergantungan pada perangkat pemrosesan berkapasitas tinggi. Pendekatan tersebut belum sesuai diterapkan pada sistem *edge computing* yang membutuhkan efisiensi sumber daya dan konsumsi daya yang rendah.

Untuk mengatasi permasalahan tersebut, dibutuhkan sistem cerdas yang mampu melakukan deteksi dan penghitungan objek manusia secara otomatis dengan akurasi tinggi dan robust meskipun kamera deteksi berada pada luar ruangan. Teknologi deep learning, khususnya dalam bidang deteksi objek (*object detection*), telah menunjukkan performa yang sangat baik dalam mengenali dan melacak objek dalam citra digital maupun video [10]. Algoritma deteksi objek diantaranya seperti YOLO (You Only Look Once), Faster R-CNN, dan Single Shot Detector (SSD).

Salah satu pendekatan berbasis visual yang paling populer adalah algoritma YOLO (You Only Look Once). Algoritma ini merupakan bagian dari metode *one-stage object detection* yang mengintegrasikan proses klasifikasi dan deteksi objek dalam satu jaringan tunggal. Berbeda dengan metode dua tahap seperti Faster R-CNN, YOLO melakukan

prediksi posisi dan kelas objek secara langsung dari citra input, sehingga lebih cepat dan efisien. YOLO unggul dalam kecepatan inferensi dan menjadi pilihan utama untuk aplikasi *real-time object detection* seperti pemantauan manusia, kendaraan, atau objek lain di lingkungan dinamis [11].

Dalam keluarga YOLOv11, terdapat lima skala Model YOLO dalam berbagai ukuran, seperti yolo11n.pt (nano), yolo11s.pt (small), yolo11m.pt (medium), yolo11l.pt (large), dan yolo11x.pt (extra-large). Pemilihan model tergantung pada kebutuhan kecepatan dan akurasi. Varian YOLOv11n (Nano) dipilih dikarenakan model ini adalah model yang paling ringan dan tercepat untuk diaplikasikan dengan perangkat sumber daya komputasi yang rendah. Meskipun akurasi lebih rendah dibanding varian lain, YOLOv11n mampu mencapai lebih dari 60 FPS bahkan pada perangkat terbatas dengan konsumsi daya minimal, menjadikannya ideal untuk sistem deteksi okupansi yang membutuhkan respons cepat dan efisiensi tinggi.

Raspberry Pi 5 merupakan *single-board computer* generasi terbaru yang memiliki spesifikasi lebih canggih dibandingkan pendahulunya, didukung dengan prosesor ARM Cortex-A76 quad-core 64-bit dan memori RAM hingga 8GB, sehingga mampu menjalankan algoritma *deep learning* dengan performa yang memadai. Pada [12] menunjukkan bahwa YOLOv11 berhasil diimplementasikan pada perangkat embedded seperti Raspberry Pi 5 untuk aplikasi deteksi manusia.

Berdasarkan uraian di atas, dapat disimpulkan bahwa teknologi dengan dikombinasikan oleh Raspberry Pi 5 memiliki potensi besar untuk digunakan dalam sistem deteksi okupansi laboratorium otomatis. Dengan kemampuan deteksi, efisiensi tinggi, serta fleksibilitas penerapan pada perangkat tertanam, algoritma ini dapat menjadi dasar dalam pengembangan sistem yang mampu mendeteksi dan menghitung jumlah orang masuk dan keluar ruangan secara akurat dan mendukung realisasi konsep laboratorium pintar yang efisien, adaptif, dan berbasis teknologi cerdas.

II. KAJIAN PUSTAKA

A. Penelitian Terdahulu

Pada penelitian [3] mengembangkan sistem manajemen energi laboratorium berbasis okupansi dengan mengombinasikan sensor PIR, CO_2 , dan ultrasonik. Sistem ini hanya mendeteksi keberadaan secara biner tanpa informasi jumlah individu, di mana sensor PIR rentan mengalami kesalahan ketika orang terlalu lama diam, sensor CO_2 memiliki respon waktu yang lambat, dan sensor ultrasonik sulit membedakan beberapa objek sekaligus. Sementara itu, pada [9] mengembangkan sistem *people counting* menggunakan kombinasi YOLOv3 dan Deep SORT pada kamera yang terletak diluar ruangan. Sistem tersebut telah mencapai akurasi 82,76% pada kondisi lengang, namun menurun saat kepadatan tinggi dan objek tumpang tindih. Namun, penelitian untuk menghitung orang pada ritel ini masih menggunakan YOLOv3 yang dijalankan pada perangkat berspesifikasi tinggi.

Sebagai dasar validasi secara teknis, dalam [12] membuktikan kelayakan implementasi YOLOv11 pada Raspberry Pi 5 (8 GB RAM) untuk sistem deteksi tenggelam secara real-time pada perangkat tertanam. Karena *people counting* pada penelitian penulis relatif lebih sederhana tanpa membutuhkan analisis perilaku dan postur tubuh yang kompleks, sistem yang diusulkan berpotensi mencapai kecepatan pemrosesan yang lebih tinggi. Di sisi lain, pada [13] menggunakan kombinasi YOLOv5 dan Faster R-CNN pada transportasi publik yang menghasilkan kecepatan hingga 75 FPS dengan presisi 86%. Namun, sistem tersebut memiliki keterbatasan pada deteksi objek kecil dan memerlukan komputasi berulang karena penerapan dua model secara bersamaan. Selain itu sistem perhitungan penumpang pada transportasi publik menghadapi variasi pencahayaan yang bervariasi dan gerakan kendaraan yang cepat yang dapat menurunkan tingkat akurasi saat melakukan pendeteksian..

Untuk metode deteksi lainnya, dalam [14] menggunakan klasifikasi Haar Cascade untuk deteksi wajah dengan akurasi hingga 86%, namun terbatas pada variasi sudut wajah, intensitas pencahayaan, jarak deteksi maksimal 5 meter, serta memerlukan beban komputasi seleksi frame. Pada [15] menggabungkan metode HOG dengan Kalman Filter yang andal dalam kondisi sederhana, namun akurasinya menurun drastis pada keadaan yang ramai atau saat mengalami oklusi ekstrem serta membutuhkan banyak konfigurasi jika dihadapkan dengan keadaan yang berbeda. Pada [16] menerapkan MobileNet-SSD *two-stage detection* yang bekerja lebih lambat dibandingkan YOLOv11n dengan pendekatan *single-stage detection*, serta berpotensi menimbulkan isu privasi akibat penggunaan deteksi wajah. Dalam [17] mengimplementasikan SSD MobileNet v1 berbasis server dengan pemrosesan secara *offline*, yang belum sepenuhnya optimal dalam mendeteksi objek kecil dan tumpang tindih. Terakhir, pada [18] mengembangkan sistem berbasis SSD dengan metode centroid tracking yang menunjukkan performa cukup baik pada lingkungan tertutup. Meskipun efektif, penelitian tersebut belum sepenuhnya akan berhasil mengatasi permasalahan jika ditempatkan pada lingkungan publik dengan peletakan kamera diluar ruangan yang mempunyai gangguan pencahayaan yang bervariasi.

B. Raspberry Pi 5

Raspberry Pi 5 adalah single-board computer performa tinggi yang berfungsi sebagai unit *edge computing* mandiri untuk menangani beban kerja komputasi berat secara lokal. Ditenagai oleh prosesor Quad-core Arm Cortex-A76 2.4 GHz dan RAM hingga 8 GB, Raspberry Pi 5 mampu menjalankan algoritma-algoritma kompleks seperti computer vision. Raspberry Pi 5 memiliki kemampuan yang sangat cepat dan memiliki penghubung data yang jauh lebih lancar. Oleh karena itu, perangkat ini bisa mengolah banyak data rumit sendirian tanpa harus bergantung pada perangkat berspesifikasi tinggi.

C. Ultralytics YOLOv11n (You Only Look Once versi 11 Nano)

YOLOv11n (Nano) merupakan algoritma inti deteksi pada sistem deteksi okupansi, yang bertugas mengidentifikasi objek manusia dalam setiap frame video. Algoritma YOLO menggunakan pendekatan *single-stage detection*, yang berarti proses klasifikasi dan regresi bounding box dilakukan dalam satu tahap komputasi. YOLOv11n, yang memiliki parameter relatif sedikit dibanding versi YOLO11 lain (sekitar 2.6 juta parameter), dirancang sebagai model ringan namun tetap mampu memberikan performa deteksi yang baik. Dalam sistem ini, YOLO memproses setiap frame video dan menghasilkan koordinat *bounding box* untuk setiap objek yang terdeteksi sebagai manusia. Informasi tersebut kemudian digunakan sebagai dasar bagi modul pelacakan objek untuk menentukan pergerakan setiap individu..

D. Normalized Convolutional Neural Network (NCNN)

NCNN adalah sebuah kerangka kerja saraf tiruan berperforma tinggi yang diprakarsai oleh Tencent. Kerangka kerja ini dirancang secara spesifik untuk diimplementasikan pada platform mobile dan *embedded system*, seperti Smartphone, Raspberry Pi, hingga berbagai perangkat IoT. Keunggulan utama NCNN terletak pada kemampuannya beroperasi secara lintas platform, mencakup ekosistem Linux, Android, iOS, hingga macOS. Melalui desain yang ringan dan efisien, NCNN memungkinkan model deep learning yang kompleks untuk dieksekusi pada perangkat dengan keterbatasan sumber daya komputasi (CPU dan RAM) tanpa memerlukan ketergantungan eksternal yang rumit.

E. SORT Tracker

Algoritma SORT (Simple Online and Realtime Tracking) adalah algoritma yang berfungsi untuk melacak dan mempertahankan identitas setiap orang yang terdeteksi ketika bergerak dari satu frame ke frame berikutnya. SORT memberikan kemampuan bagi sistem untuk “mengingat” bahwa objek tertentu yang muncul di frame berikutnya adalah orang yang sama seperti sebelumnya. Dengan memanfaatkan Kalman Filter, SORT memprediksi lokasi objek pada frame selanjutnya berdasarkan pergerakannya, sementara algoritma Hungarian digunakan untuk mencocokkan prediksi tersebut dengan *bounding box* deteksi terbaru. Sistem ini memberikan ID unik pada setiap objek yang terdeteksi. Hasil resmi dari penelitian ini dapat dilihat pada link github berikut: <https://github.com/abewley/sort>.

F. ByteTrack Tracker

ByteTrack adalah algoritma Multi-Object Tracking (MOT) yang merubah pengelompokan data melalui strategi asosiasi dua tahap bernama BYTE. Berbeda dengan pelacak konvensional (SORT) yang membuang deteksi dengan skor kepercayaan rendah, ByteTrack tetap memproses deteksi tersebut untuk mengembalikan objek saat mengalami oklusi atau kekaburan. Dengan memanfaatkan Kalman Filter untuk prediksi posisi dan IoU untuk pencocokan, algoritma ini

mampu menyambungkan lintasan yang terputus tanpa memerlukan fitur Re-ID yang berat. Ultralytics juga telah mengintegrasikan ByteTrack ke dalam kode Pustaka bawaan, sehingga saat ingin menjalankan perintah pelacakan di YOLO11, ByteTrack langsung dapat dipanggil tanpa perlu menginstal library tambahan. Hasil resmi dari penelitian mengenai ByteTrack ini dapat dilihat pada link GitHub berikut: <https://github.com/ifzhang/ByteTrack>.

G. OpenCV

OpenCV (Open Source Computer Vision Library) adalah pustaka perangkat lunak open source yang dirancang untuk pengolahan gambar dan video. Pustaka ini menyediakan fungsi untuk deteksi wajah, segmentasi gambar, pendeteksian objek atau pelacakan objek, dan lain-lain. OpenCV mendukung banyak Bahasa pemrograman termasuk Python, C++, dan Java, sehingga fleksibel untuk berbagai kebutuhan pengembangan..

H. Confusion Matrix

Confusion matrix adalah sebuah tabel evaluasi yang digunakan untuk mengukur performa model klasifikasi dengan membandingkan hasil prediksi model terhadap data aktual. Ada empat istilah yang merupakan hasil proses klasifikasi pada tabel *confusion matrix* yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negatif (FN).

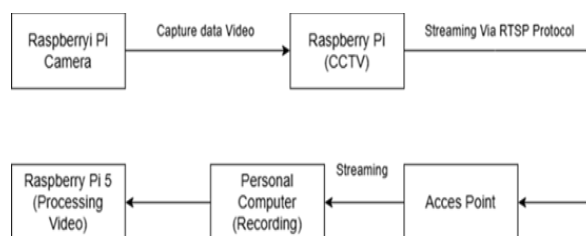
III. METODE PENELITIAN

A. Rancangan Sistem

Pada bagian ini menjelaskan tentang gambaran keseluruhan sistem, alur kerja sistem serta blok diagram secara keseluruhan pada penelitian ini.

1. Gambaran Keseluruhan Sistem

Bagian ini menjelaskan bagaimana gambaran sistem yang ada pada penelitian ini. Berikut ada gambaran keseluruhan sistem.



Gambar 1. Diagram Blok Keseluruhan Sistem

Berdasarkan Gambar 1 cara kerja sistem ini, Perangkat kamera dipasang dengan Raspberry Pi yang berfungsi sebagai CCTV diletakkan didepan laboratorium otomasi dengan tinggi 2,5 meter, sudut 30° menghadap bawah dan berjarak sekitar 4 meter dari objek pintu yang dideteksi, merekam video dengan resolusi 1280x720 30fps. Hasil pemantauan ini disiarkan secara streaming menggunakan protokol RTSP (

Real Time Streaming Protocol) dengan menggunakan IP address sebagai berikut: `rtsp://10.10.12.101:8554/cam`. Video tersebut diteruskan melalui Access Point (WIFI), yang menjadi penghubung antara Raspberry Pi dan Personal Computer (PC). PC bertindak sebagai *client* yang mengakses URL RTSP tersebut untuk merekam video secara penuh menggunakan aplikasi RTSP cam atau VLC media player sebagai data utama dalam penelitian. Pada tahap ini Raspberry pi 5 akan mendeteksi objek menggunakan YOLOv11n, melakukan *tracking* menggunakan algoritma pelacakan SORT (Simple Online and Real-time Tracking) dan ByteTrack dari frame ke frame berikutnya untuk memastikan objek masuk atau keluar ruangan. Selanjutnya, sistem yang dijalankan Raspberry Pi akan menghitung objek masuk dan keluar ruangan sesuai dengan ambang batas garis yang telah ditentukan.

2. Alur Kerja sistem



Gambar 2. Diagram Blok Computer Vision

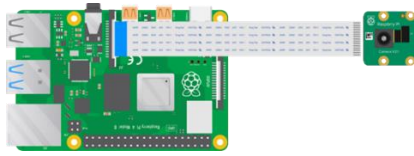
Berikut adalah penjelasan detail untuk setiap alur kerja sistem yang akan digunakan.

- Pengumpulan Gambar: pada tahap ini, kamera CCTV akan mengakuisisi data berupa video yang kemudian akan dipecah menjadi frame-frame gambar.
- Deteksi objek: Sistem akan memproses gambar yang telah dikumpulkan dan memberi label kepada objek deteksi.
- Pelacakan Objek: Setelah objek terdeteksi, selanjutnya adalah melacak objek yang bergerak masuk atau keluar ruangan
- Perhitungan objek: Pada tahap ini, menghitung objek yang sudah diberi label dari setiap gerakan keluar atau masuk ruangan yang telah melewati batas ambang.
- Pencatatan data: Pada tahap ini, jumlah orang yang keluar atau masuk ruangan akan disimpan untuk mengetahui jumlah orang didalam ruangan.

B. Sistem Perangkat Keras

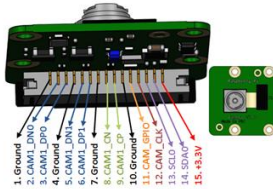
1. Konfigurasi pengambilan video data

Perangkat yang digunakan dalam proses pengambilan data adalah menggunakan Raspberry Pi 4 dengan Raspberry Pi modul camera 2. Agar Raspberry Pi dapat mengakses modul kamera, module kamera dihubungkan dengan Raspberry Pi seperti pada gambar berikut.



Gambar 3. Konfigurasi Raspberry Pi dengan pi cam

Terdapat 15 pin yang harus dihubungkan dari camera module pada port cam Raspberry Pi, sehingga Raspberry Pi dapat mengakses camera module dengan benar, Urutan pin tertera pada Gambar 4 sebagai berikut.



Gambar 4. Urutan pinout module pi camera

2. Spesifikasi komponen pemrosesan video data
- Raspberry Pi 5

Tabel 1. Spesifikasi Raspberry Pi 5

Aspek	Keterangan
Prosesor	Broadcom BCM2712 Quad-core Arm Cortex-A76 @ 2.4GHz VideoCore VII GPU
Grafis	(Mendukung Vulkan 1.2 & OpenGL ES 3.1).
Memori	LPDDR4X-4267 8GB.
Output	Dual 4K@60fps HDMI.
I/O	Slot PCIe 2.0
Kamera	2x Port MIPI 4-lane
Koneksi	Gigabit Ethernet, Wi-Fi Dual-band, dan Bluetooth 5.0.

Tabel 1 adalah spesifikasi perangkat Raspberry Pi 5 yang akan digunakan dalam pemrosesan video data, dengan menggunakan seri tertinggi dan didukung dengan memori sebesar 8 GB, sehingga diharapkan sistem akan berjalan dengan optimal walaupun bekerja pada perangkat embedded.

- Sandisk Ultra 16 GB Micro SD

Tabel 2. Spesifikasi Sandisk Ultra 16 GB MicroSD

Aspek	Keterangan
Kapasitas	16 GB.
Tipe	MicroSDHC (UHS-I)
Kecepatan Transfer	Up to 100 MB/s
Kelas Kecepatan	Class 10, UHS-I (U1).
Peringkat Kinerja	A1

Tabel 2 adalah spesifikasi perangkat MicroSD sandisk, Perangkat ini berfungsi sebagai media penyimpanan sistem operasi (OS) dan penyimpanan data pada device Raspberry Pi 5 yang digunakan dalam pemrosesan data video.

C. Perancangan Perangkat Lunak

1. Pendeteksian objek

Pendeteksian objek merupakan suatu proses untuk mengenali objek dimana pada penelitian ini menggunakan objek dengan label “person” sebagai data utama yang akan diambil. Berikut adalah alur perancangan pendeteksian bentuk

- Pengumpulan dataset
Pada tahap ini, dataset dikumpulkan sebagai bahan utama untuk melatih model. Dataset ini berupa gambar-gambar yang berisi objek yang akan dideteksi oleh sistem. Gambar dalam dataset diperoleh dari kamera CCTV yang dipantau di depan Laboratorium Otomasi Industri Dataset mencakup berbagai kondisi sudut pandang.
- Labeling
Gambar yang terkumpul, selanjutnya diberi label. Untuk pemberian label menggunakan Roboflow. Roboflow adalah platform yang dirancang untuk mempermudah proses pengelolaan data set, labelling, dan pelatihan pada proyek berbasis computer vision seperti deteksi objek, klasifikasi gambar, dan segmentasi gambar. Pada tahap ini, setiap objek dalam gambar diberi *bounding box* dan label yaitu “Person”.
- Training Data
Dataset yang sudah dilabeli digunakan untuk melatih model YOLOV11n. Proses pelatihan data dilakukan di Google Colab. Google Colab adalah platform yang disediakan oleh google untuk menjalankan kode python langsung di peramban tanpa memerlukan instalasi perangkat lunak tambahan. Model dilatih dengan 50 epoch dengan total dataset berkisar 1250 gambar untuk mencapai akurasi yang optimal.
- Implementasi YOLOV11n
Model yang sudah dilatih tadi diintegrasikan ke dalam aplikasi. Implementasi dilakukan menggunakan bahasa pemrograman python dan untuk aplikasi yang digunakan adalah Thonny IDE
- Pengujian
Pada tahap ini, dilakukan pengujian dengan berbagai kondisi untuk memastikan model dapat mendeteksi, melacak dan menghitung objek dengan baik

2. Pelacakan Objek

Proses pendeteksian objek difokuskan pada pengenalan manusia sebagai data utama dengan menerapkan metode pelacakan objek untuk mempertahankan identitas dan lokasi setiap orang dalam urutan frame video. Penelitian ini mengintegrasikan algoritma SORT (Simple Online and Realtime Tracking) dan ByteTrack untuk mengelola identitas objek dari satu frame ke frame berikutnya guna memberikan kemampuan 'mengingat' objek setelah dikenali oleh detektor YOLO. Pada algoritma SORT, identitas dipertahankan melalui kombinasi Kalman Filter untuk memprediksi lintasan dan Hungarian Algoritma untuk mencocokkan prediksi

dengan deteksi terbaru. Sementara itu, ByteTrack memperkuat sistem dengan memproses deteksi berskor kepercayaan rendah akibat objek yang terhalang (oklusi) sehingga meminimalisir terputusnya identitas. Dengan pemberian ID unik yang stabil dari kedua tracker ini, sistem dapat secara akurat mencatat dan memvalidasi setiap individu yang melintasi garis batas masuk maupun keluar.

3. Counting Objek dan Data Logging

Counting Object adalah metode yang digunakan dalam menghitung jumlah orang “keluar” dan “Masuk” dalam penelitian ini. Metode counter ini adalah hasil akhir dari proses pendeteksian dan pelacakan yang telah dilakukan oleh komputer dengan menggunakan computer vision. Pada tahap akhir, jumlah orang “keluar” dan “masuk” akan dihitung jumlahnya untuk mendeteksi jumlah orang yang berada didalam ruangan.



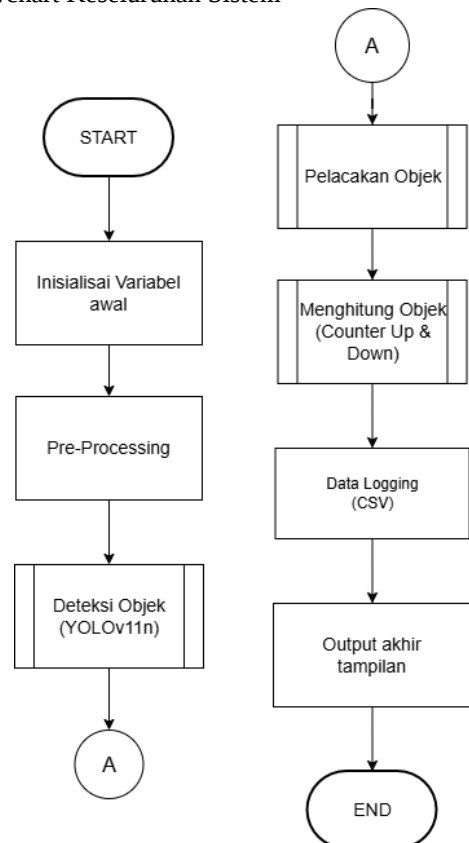
Gambar 5. Contoh Hasil Implementasi

Gambar 5 merupakan implementasi visual dari sistem counter object yang dirancang untuk memantau okupansi ruangan secara otomatis. Dalam sistem ini, logika penghitungan didasarkan pada interaksi antara objek yang terlacak dengan dua garis virtual, yaitu garis hijau dan garis merah, yang berfungsi sebagai pemicu perhitungan okupansi. Secara teknis, mekanisme penghitungan bekerja sebagai berikut:

- **Logika Masuk:** Objek divalidasi sebagai "Masuk" jika urutan pergerakannya melewati garis merah terlebih dahulu, kemudian dilanjutkan dengan melewati garis hijau.
- **Logika Keluar:** Sebaliknya, objek divalidasi sebagai "Keluar" jika urutan pergerakannya dimulai dari melewati garis hijau baru kemudian melewati garis merah.
- **Buffer zone :** Jika objek hanya menyentuh atau melewati salah satu garis saja maka akan memasuki area *buffer zone*, dan secara otomatis sistem akan mengabaikan perhitungan dari data tersebut karena tidak memenuhi kriteria sekuensial yang telah ditetapkan.

D. Flowchart

1. Flowchart Keseluruhan Sistem



Gambar 6. Flowchart Keseluruhan Sistem

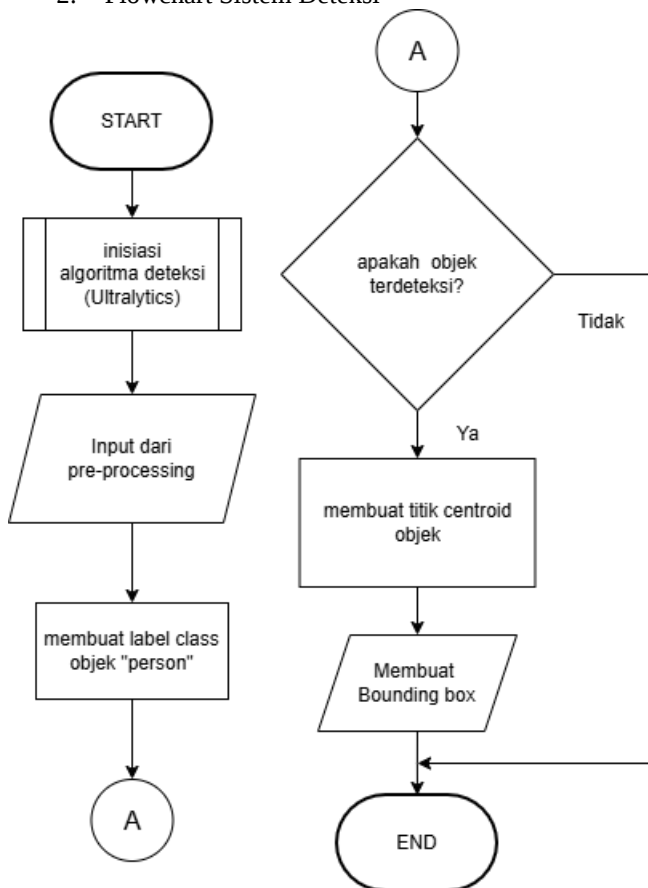
Berdasarkan Gambar 6, berikut adalah penjelasan flowchart:

Alur kerja dimulai dengan inisialisasi variabel awal dan tahap *pre-processing* citra, yang kemudian dilanjutkan ke fase Deteksi Objek menggunakan model YOLOv11n untuk melokalisasi keberadaan manusia. Tahap krusial berikutnya adalah Pelacakan Objek, di mana sistem mengombinasikan algoritma SORT dan ByteTrack untuk menjaga konsistensi identitas (ID) objek antar frame, memastikan setiap individu terlacak secara akurat meskipun terjadi oklusi atau penurunan skor deteksi.

Setelah pelacakan stabil, sistem masuk ke tahap Menghitung Objek dengan logika *Counter Up & Down*, yang berfungsi mencatat jumlah orang yang masuk atau keluar berdasarkan pergerakan ID melewati garis virtual tertentu. Data hasil perhitungan tersebut kemudian diproses melalui *Data Logging* untuk disimpan secara permanen dalam format berkas CSV sebagai basis data laporan. Seluruh rangkaian proses ini diakhiri dengan penyajian Output akhir tampilan pada

layar yang memvisualisasikan hasil deteksi, pelacakan, dan penghitungan serta okupansi secara komprehensif.

2. Flowchart Sistem Deteksi

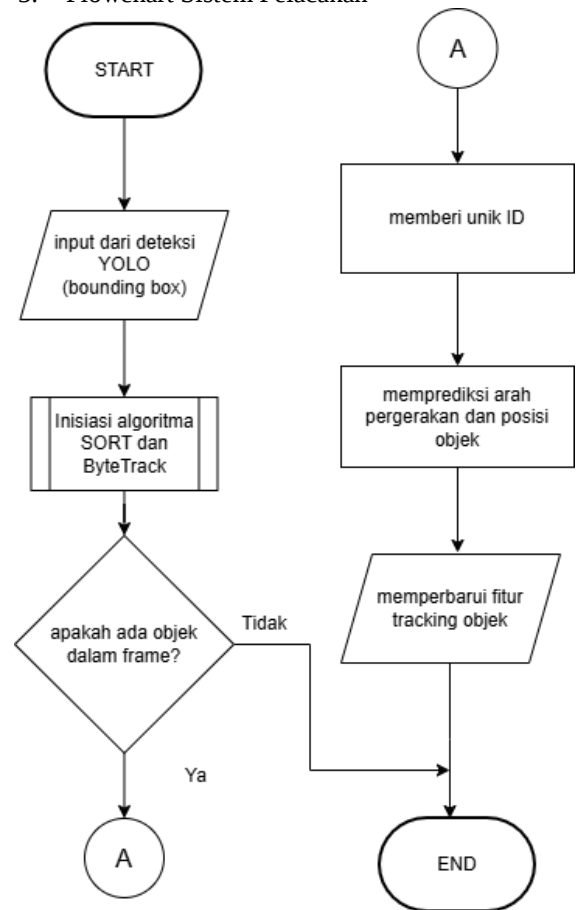


Gambar 7. Flowchart Algoritma Deteksi

Berdasarkan Gambar 7, berikut adalah penjelasan flowchart:

Alur kerja sistem deteksi objek manusia dimulai dengan inisiasi algoritma melalui library Ultralytics, lalu menerima input data berupa citra atau frame video yang telah melewati tahap *pre-processing*. Selanjutnya, sistem menetapkan label kelas spesifik "person" untuk memfokuskan pemindaian hanya pada objek manusia. Jika objek tidak terdeteksi dalam frame, program langsung menuju tahap akhir bersiap memproses data berikutnya. Namun, jika objek berhasil ditemukan, sistem akan menghitung koordinat titik tengah atau titik pusat objek sebagai referensi posisi serta membuat kotak pembatas (*bounding box*) sebagai penanda visual. Seluruh rangkaian proses ditutup pada titik End, yang berulang secara terus-menerus untuk setiap frame yang masuk guna menghasilkan pemantauan yang konsisten.

3. Flowchart Sistem Pelacakan



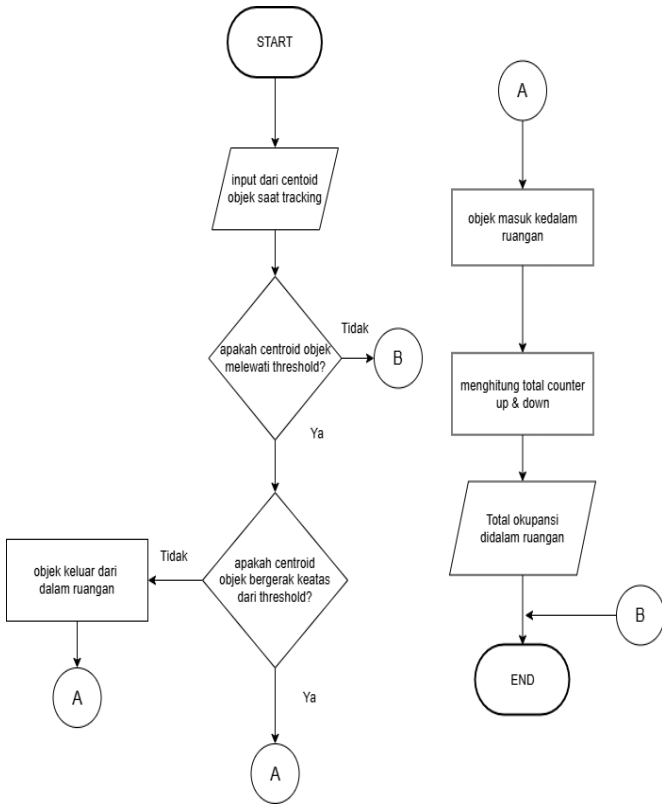
Gambar 8. Flowchart Algoritma Pelacakan

Berdasarkan Gambar 8, berikut adalah penjelasan flowchart:

Alur kerja dimulai dengan menerima input berupa *bounding box* dari detektor, yang kemudian diolah melalui inisialisasi algoritma pelacakan SORT dan ByteTrack. Jika objek terdeteksi dalam frame, sistem akan memberikan ID unik yang konsisten untuk setiap individu agar tidak tertukar.

Proses dilanjutkan dengan memprediksi arah pergerakan serta posisi masa depan objek menggunakan estimasi matematis seperti Kalman Filter. Seluruh informasi trajektori ini kemudian diperbarui ke dalam fitur pelacakan sistem sebelum siklus berakhir, memastikan bahwa setiap pergerakan objek dalam rekaman video dapat dipantau secara akurat dan berkesinambungan.

4. Flowchart Perhitungan



Gambar 9. Flowchart Perhitungan Objek

Berdasarkan Gambar 9, berikut adalah penjelasan flowchart:

Sistem menerima input titik pusat objek dari proses pelacakan. Titik pusat pada *bounding box* objek yang melewati dua garis ambang batas akan dianalisis arah pergerakannya: jika bergerak ke atas ambang batas dengan melewati garis merah kemudian melewati garis hijau, maka perhitungan bertambah (+1) dan objek akan dihitung sebagai "masuk", jika objek bergerak ke bawah ambang batas dengan melewati garis hijau kemudian melewati garis merah maka perhitungan akan berkurang (-1) dan objek akan dihitung sebagai "keluar". Total okupansi didalam ruangan dihitung dari selisih total objek masuk dan keluar.

E. Skema Pengujian

Skema Pengujian yang akan dilakukan adalah dengan menggunakan tiga video dimana setiap video memiliki karakteristik masing masing. Video 1 dengan pencahayaan terang, video 2 dengan perbedaan sudut pandang kamera dan video 3 dengan pencahayaan gelap dan terang. Setiap videonya juga dilakukan pengujian menggunakan berbagai kombinasi format model dan algoritma pelacakan yang berbeda-beda. Hasil dari percobaan dilakukan dengan perhitungan *Confusion Matrix* pada evaluasi sistem deteksi, perhitungan MOTA (Multi Object Tracking Accuracy) untuk evaluasi sis-

tem pelacakan dan melakukan perhitungan MAE (Mean Average Error), MAPE (Mean Average Percentage Error) serta *Counting Accuracy* untuk evaluasi sistem okupansi.

Perhitungan dari evaluasi sistem deteksi untuk mencari *Precision*, *Recall*, dan *F1* dengan rumus *Confusion Matrix* dijelaskan sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3)$$

Dengan keterangan sebagai berikut :

- True Positive (TP) : Model berhasil mendeteksi orang dan memang benar terdapat orang pada gambar/frame
- False Positive (FP) : Model mendeteksi orang padahal sebenarnya tidak ada orang
- False Negative (FN) : Ada orang pada gambar/frame tetapi model gagal mendeteksinya
- True Negative (TN) : Tidak ada orang dan model juga tidak mendeteksi apa pun

Perhitungan dari evaluasi sistem pelacakan untuk mencari MOTA dijelaskan dalam rumus sebagai berikut:

$$MOTA = 1 - \frac{FN_t + FP_t + IDSW_t}{GT_t} \quad (4)$$

Dengan keterangan sebagai berikut:

- False Positive (FP_t) : Jumlah objek yang gagal dideteksi oleh sistem pada frame t
- False Negative (FN_t) : Jumlah objek yang salah dideteksi (deteksi palsu) pada frame t
- $IDSW_t$: Jumlah kesalahan di mana sistem pelacak salah mengubah identitas/ID suatu objek pada frame t
- GT_t : Jumlah total objek sebenarnya yang ada pada frame t

Perhitungan dari evaluasi sistem okupansi untuk mencari nilai MAE, MAPE, dan *Accuracy Rate* dijelaskan dengan rumus sebagai berikut:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

$$MAPE = \frac{100\%}{1} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (6)$$

$$Accuracy = \left(1 - \frac{|y - \hat{y}|}{y} \right) \times 100\% \quad (7)$$

Dengan keterangan sebagai berikut:

- n = jumlah data pengujian
 y = nilai aktual
 $\hat{y}$ = nilai prediksi
 y_i = nilai aktual pada data ke- i
 $\hat{y}_i$ = nilai prediksi pada data ke- i

IV. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Dalam penelitian kali ini, pengumpulan dataset gambar hanya dilakukan dari dataset pribadi. Dataset pribadi ini merupakan dataset yang diambil dengan cara merekam menggunakan kamera CCTV yang telah dipasang didepan laboratorium Otomasi. Dataset pribadi ini hanya satu-satunya sumber dataset yang digunakan dalam penelitian kali ini karena lokasi pemantauan yang sangat spesifik yaitu berada didepan pintu laboratorium otomasi industri dan robotika. Penggunaan person dataset yang dikoleksi oleh webcam publik tidak secara representatif dengan penelitian yang dilakukan saat ini.

B. Pengolahan Dataset

Tahap selanjutnya setelah mengumpulkan dataset adalah pengolahan dataset. Pada tahap ini akan dilakukan menjadi 2 tahap yang terdiri dari preprocessing dataset dan augmentasi dataset. Berikut merupakan penjelasan dari masing-masing proses dari pengolahan dataset.

1. Preprocessing Dataset

Preprocessing dataset merupakan suatu tahap awal yang dilakukan untuk menyiapkan dataset agar dapat digunakan secara layak ketika berada pada tahap training model. *Preprocessing* dalam penelitian ini hanya meliputi proses *resizing*.

Resizing merupakan suatu proses untuk mengubah ukuran panjang gambar dan lebar gambar. Secara umum dalam *resizing* gambar untuk sistem deteksi adalah 640 x 640 px. Oleh karena itu, penelitian kali ini melakukan perubahan (*Resize*) gambar-gambar pada dataset yang digunakan menjadi 640 x 640 px.

2. Augmentasi Dataset

Augmentasi data dilakukan untuk memodifikasi dataset guna meningkatkan keragaman dan membantu model menggeneralisasi gambar agar lebih adaptif, tidak mudah mengalami *overfitting*, serta mampu mempertahankan akurasi tinggi pada kualitas gambar buruk di lapangan.

Tabel 3. Jumlah Total Augmentasi Dataset

Augmentasi	Jumlah
Brightness +20%	208
Brightness -20%	208
Exposure +10%	206
Exposure -10%	206
Blur	209
Noise	209
Total	1246

Berdasarkan Tabel 3, didapatkan jumlah total augmentasi dataset sebanyak 1246 gambar yang terdiri dari berbagai Teknik augmentasi. Dari berbagai parameter augmentasi yang digunakan memiliki jumlah yang hampir setara untuk menjaga keseimbangan dataset agar model tidak mengalami bias.

C. Model

1. Pengujian dengan Pre-Trained YOLO11n

Terdapat 2 kondisi yang akan diuji dengan *pre-trained* YOLO11n dalam diantaranya adalah kondisi dimana diuji menggunakan dataset gambar lapangan tanpa augmentasi dan dataset gambar lapangan yang telah teraugmentasi. Berikut adalah hasil uji dari *pre-trained* YOLO11n dengan berbagai kondisi tersebut. Pengujian dengan *pre-trained* YOLO11n dipaparkan dalam tabel berikut.

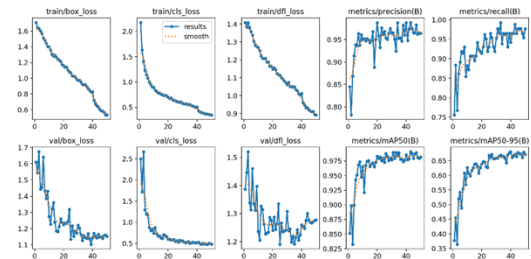
Tabel 4. Pengujian menggunakan pre-trained YOLO11n

Jenis	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)
Dataset non-augmentasi	0.936	0.873	0.931	0.394
Dataset Augmentasi	0.898	0.767	0.872	0.388

Berdasarkan Tabel 4, hasil pengujian menunjukkan bahwa model *pre-trained* YOLO11n memiliki nilai mAP50-95 yang masih kurang akurat dalam menentukan posisi dan ukuran *bouding box* dibandingkan anotasi sebenarnya. Meskipun kemampuan deteksi objek sudah baik, kualitas lokalisasi objek masih perlu ditingkatkan agar hasil deteksi menjadi lebih presisi. Peningkatan nilai mAP50-95 dapat dilakukan melalui penerapan *transfer learning* menggunakan dataset yang dikumpulkan langsung dari lapangan.

2. Transfer Learning Model

Proses pelatihan model dilakukan melalui platform Google Colaboratory menggunakan runtime Python 3 dengan akselerator GPU Tesla T4 (VRAM 16 GB, RAM 12GB – 15GB). Arsitektur YOLOv11 dimuat menggunakan bobot *pre-trained* yolo11n.pt dengan parameter ukuran gambar 640 piksel, ukuran batch 16, serta dilatih sebanyak 50 epoch untuk meminimalkan tingkat kesalahan secara bertahap.



Gambar 10. Hasil Training Dataset YOLO11n

Berdasarkan Gambar 10 visualisasi hasil training, grafik box loss, cls_loss, dan dfl_loss menunjukkan pola penurunan yang konsisten dan stabil pada data *training* maupun *validation*, mengindikasikan model telah mampu belajar secara optimal tanpa gejala *overfitting* ataupun *underfitting* yang signifikan. Di sisi lain, grafik metrik *precision*, *recall*, *mAP50*, dan *mAP50-95* mengalami peningkatan performa cukup tajam di awal dan terus meningkat bertahap hingga mencapai titik stabil di akhir masa pelatihan, dengan total durasi komputasi menyelesaikan 50 epoch selama 27 menit 50 detik.

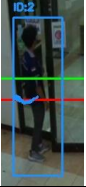
3. Optimalisasi Format Model

Proses optimalisasi model dilakukan dengan proses pengonversian dari format default ultralytics yaitu PyTorch (.pt) ke format Normalized Convolutional Neural Network (.ncnn) proses ini dilakukan untuk mengoptimalkan performa pemrosesan pada perangkat dengan sumber daya terbatas, seperti Raspberry Pi 5. NCNN merupakan kerangka kerja komputasi berperforma tinggi yang dirancang khusus untuk penggunaan pada CPU perangkat seluler atau sistem tertanam.

D. Evaluasi Kinerja Sistem Deteksi

Evaluasi kinerja sistem deteksi ini bertujuan untuk memastikan bahwa algoritma deteksi (YOLO11n) yang digunakan dapat mendeteksi keberadaan objek manusia dengan tepat. Terdapat tiga pengujian dengan 3 karakteristik video data yang digunakan dalam penelitian ini. Video data 1 memiliki karakteristik pencahayaan yang normal, video data 2 memiliki perbedaan sudut pandang kamera yang berbeda, dan video data 3 memiliki karakteristik pencahayaan gelap-terang. Skema pengujian untuk mengevaluasi kinerja sistem deteksi dilakukan dengan menghitung *precision*, *recall* dan *F-1 Score*, Berikut adalah hasil dari pengujian dalam sistem deteksi.

Tabel 5. Contoh Hasil Implementasi Sistem Deteksi

No. Event	Sample Gambar Event	Total Frame	Klasifikasi confusion Matrix (Frame)			
			TP	FP	FN	TN
1		271	253	0	18	0
2		316	301	0	15	0

Tabel 5 menunjukkan contoh hasil implementasi sistem deteksi yang dilakukan pada setiap kejadian yang berada pada video data. Setiap kejadian akan dihitung setiap frame ketika objek tersebut terdeteksi oleh kamera. Total dari frame pada setiap kejadian tersebut kemudian langsung di

klasifikasikan dalam confusion matrix, sebagai rincian mendetail dari kondisi-kondisi yang ada pada setiap kejadian.

1. Footage Video 1

Tabel 6. Hasil Evaluasi Kinerja Sistem Deteksi Video 1

Tracking/Format	Precision (%)	Recall (%)	F-1 (%)
ByteTrack/.ncnn	100	88.37	93.87
ByteTrack/.Pt	100	87.33	93.24
SORT/.ncnn	100	88.76	94.04
SORT/.Pt	100	88.88	94.11

Berdasarkan Tabel 6 Hasil Evaluasi Kinerja Sistem Deteksi 1, hasil seluruh kombinasi algoritma pelacakan dan format model menunjukkan performa yang sangat optimal dan seimbang. Semua pengujian berhasil mencapai nilai *Precision* sempurna (100%), yang berarti tidak ada kesalahan deteksi. Nilai *Recall* berkisar antara 87,33% hingga 88,88%, menghasilkan *F-1 Score* yang stabil dan tinggi di kisaran 93% – 94%. Kombinasi SORT/.Pt mencatatkan hasil tertinggi pada pengujian ini dengan F-1 Score sebesar 94,11%.

2. Footage Video 2

Tabel 6. Hasil Evaluasi Kinerja Sistem Deteksi Video 2

Tracking/Format	Precision (%)	Recall (%)	F-1 (%)
ByteTrack/.ncnn	100	74,17	85,17
ByteTrack/.Pt	100	75,92	86,31
SORT/.ncnn	43,28	93,35	59,14
SORT/.Pt	42,25	93,22	58,15

Berdasarkan Tabel 7 Hasil Evaluasi Kinerja Sistem Deteksi, video 2 menjadi pembuktian yang menunjukkan perbedaan kualitas yang kontras antara algoritma pelacakan yang digunakan dalam pengujian. ByteTrack terbukti sangat andal dengan mempertahankan *Precision* mutlak (100%) dan *F1-Score* yang kuat di kisaran 85% – 86%. Sebaliknya, SORT mengalami penurunan performa yang drastis. Nilai *Precision* merosot tajam hingga ke angka 42% – 43% hal tersebut diakibatkan karena kesalahan deteksi (*false positive*) yang muncul hampir disetiap frame video, sehingga *F-1 Score* jatuh ke angka 58% – 59%, meskipun nilai *Recall*-nya mendominasi di angka 93%.

3. Footage Video 3

Tabel 7. Hasil Evaluasi Kinerja Sistem Deteksi Video 3

Tracking/Format	Precision (%)	Recall (%)	F-1 (%)
ByteTrack/.ncnn	100	94,54	97,19
ByteTrack/.Pt	100	94,0	96,91
SORT/.ncnn	100	95,29	97,58
SORT/.Pt	100	95,15	97,51

Berdasarkan Tabel 8 Hasil Evaluasi Kinerja Sistem Deteksi 3, hasil seluruh algoritma deteksi kembali menunjukkan performa tertinggi dan mendapat hasil paling maksimal di antara ketiga video uji. Nilai *Precision* mencapai 100% di

semua kombinasi. Tingkat keberhasilan penemuan objek (*Recall*) juga sangat tinggi, menyentuh angka 94% hingga 95,29%. Nilai dari precision dan recall yang optimal menghasilkan nilai *F-1 Score* di kisaran 96% – 97%, dengan kombinasi SORT/.ncnn sebagai yang tertinggi secara mutlak sebesar 97,58%.

E. Evaluasi Kinerja Sistem Pelacakan

Evaluasi kinerja sistem pelacakan ini bertujuan untuk memastikan bahwa algoritma pelacakan (SORT tracking dan ByteTrack tracking) yang digunakan dapat melacak dan memberikan ID unik kepada setiap objek manusia yang berada dalam frame dengan tepat dan akurat. Terdapat tiga pengujian dengan 3 karakteristik video data yang digunakan dalam penelitian ini. Video data 1 memiliki karakteristik pencahayaan yang normal, video data 2 memiliki perbedaan sudut pandang kamera yang berbeda, dan video data 3 memiliki karakteristik pencahayaan gelap-terang. Skema pengujian untuk mengevaluasi kinerja sistem pelacakan dilakukan dengan menghitung MOTA, Berikut adalah hasil dari masing-masing pengujian video data.

1. Footage Video 1

Tabel 8. Evaluasi Kinerja Sistem Pelacakan Video 1

Tracking / Format	Total Orang Real	Objek Terdeteksi	Total ID Terdeteksi	ID Switch	Jumlah frame mendeteksi Objek	MOTA
ByteTrack/.ncnn	10	10	10	0	2986	88,37%
ByteTrack/.Pt	10	10	10	0	2983	87,32%
SORT/.ncnn	10	10	12	2	2990	88,76%
SORT/.Pt	10	10	11	1	2940	88,87%

Berdasarkan Tabel 9 Evaluasi Kinerja Sistem Pelacakan Video 1, performa pelacakan untuk total 10 orang berjalan sangat baik pada kedua algoritma. ByteTrack unggul mutlak dalam menjaga konsistensi identitas objek, terbukti dengan nilai *ID Switch* sebesar 0 dan jumlah Total ID Terdeteksi yang sempurna (sesuai dengan jumlah objek asli yaitu 10). Di sisi lain, SORT mengalami kendala dalam pemberian ID unik sehingga mengalami 1 hingga 2 kali *ID Switch*, yang menyebabkan lonjakan Total ID Terdeteksi menjadi 11 dan 12. Meski demikian, dari segi akurasi pelacakan keseluruhan, SORT mencatatkan nilai MOTA sedikit lebih tinggi (~88%) dibanding ByteTrack (~87%) karena unggul tipis dalam jumlah frame yang berhasil mendeteksi objek.

2. Footage Video 2

Tabel 9. Evaluasi Kinerja Sistem Pelacakan Video 2

Tracking/Forma t	Total Oran g Real	Objek Terdeteks i	Total ID Terdeteks i	ID Switc h	Jumlah frame mendeteks i Objek	MOTA
ByteTrack/.ncnn	7	7	7	0	1541	74,17 %
ByteTrack/.Pt	7	7	7	0	1540	75,90 %
SORT/.ncnn	7	8	13	5	3444	41,84 %
SORT/.Pt	7	8	11	3	3522	41,41 %

Berdasarkan Tabel 10 Evaluasi Kinerja Sistem Pelacakan Video 2, terdapat perbedaan kualitas antar algoritma pelacakan yang sangat kontras dalam melacak 7 orang/objek. ByteTrack kembali menunjukkan konsistensi sempurna dengan nilai *ID Switch* tetap 0 dan Total ID Terdeteksi yang akurat (7 objek), menghasilkan nilai MOTA yang solid di kisaran 74% – 75%. Sebaliknya, performa SORT merosot di video ini, terjadi kelebihan pendeteksian karena *false positive* yang mengakibatkan Objek Terdeteksi menjadi 8 dan *ID Switch* yang sangat tinggi (3 hingga 5 kali), menyebabkan Total ID Terdeteksi membengkak hingga 13 ID untuk kombinasi SORT/.ncnn dan 11 ID untuk kombinasi SORT/.pt. Hal ini mengakibatkan jatuhnya nilai akurasi MOTA SORT yang mendapat hasil sebesar (~41%).

3. Footage Video 3

Tabel 10. Evaluasi Kinerja Sistem Pelacakan Video 3

Tracking / Format	Total Orang Real	Objek Terdeteksi	Total ID Terdeteksi	ID Switch	Jumlah frame mendeteksi Objek	MOTA
ByteTrack/.ncnn	18	18	22	4	4655	94,45
ByteTrack/.Pt	18	18	23	5	4474	93,89
SORT/.ncnn	18	18	23	4	4694	95,20
SORT/.Pt	18	18	21	3	4791	95,27

Berdasarkan Tabel 11 Evaluasi Kinerja Sistem Pelacakan Video 3 memiliki jumlah objek paling tinggi sebanyak 18 orang, seluruh sistem menunjukkan performa pelacakan yang sangat baik namun mengalami gangguan pada algoritma tracking dalam jumlah yang hampir mirip. Semua kombinasi berhasil mendeteksi jumlah total objek asli secara tepat (18 objek). Namun, pada video ini, baik ByteTrack maupun SORT sama-sama mengalami kendala *ID Switch* yang cukup banyak (berkisar antara 3 hingga 5 kali), sehingga Total ID Terdeteksi melonjak di kisaran 21 hingga 23 ID. Kendati demikian, nilai frame deteksi yang sangat tinggi membuat nilai MOTA di video ini menjadi yang tertinggi secara keseluruhan, di mana SORT memimpin tipis dengan nilai mencapai 95,2%.

F. Evaluasi Kinerja Sistem Okupansi

Evaluasi kinerja sistem okupansi ini bertujuan untuk memastikan bahwa algoritma perhitungan (SORT tracking dan ByteTrack tracking) yang digunakan dapat menghitung setiap objek manusia saat masuk maupun keluar ruangan dengan tepat dan akurat. Terdapat tiga pengujian dengan 3 karakteristik video data yang digunakan dalam penelitian ini. Video data 1 memiliki karakteristik pencahayaan yang normal, video data 2 memiliki perbedaan sudut pandang kamera yang berbeda, dan video data 3 memiliki karakteristik pencahayaan gelap-terang. Skema pengujian untuk mengevaluasi kinerja sistem pelacakan dilakukan dengan menghitung MAE (Mean Average Error), MAPE (Mean Average Percentage Error), dan *Counting accuracy*. Berikut adalah hasil dari masing-masing pengujian video data.

Tabel 11. Contoh Hasil Implementasi Sistem Okupansi

No	Sample Gambar	Deteksi (✓ / ✕)	Pelacakan (✓ / ✕)	counter		okupansi
				In	Out	
1.		✓	✓	1	0	1
2.		✓	✓	1	0	2

Tabel 12 menunjukkan contoh hasil implementasi sistem okupansi yang dilakukan pada setiap kejadian yang berada pada video data. Setiap kejadian akan dipantau apakah fungsi deteksi dan pelacakan sudah bekerja dengan baik saat objek melintasi garis virtual. Sistem akan mengukur orang yang masuk atau keluar yang akan membuat nilai okupansi akan +1 saat ada objek masuk dan akan -1 saat ada objek keluar.

1. Footage Video 1

Tabel 12. Hasil Evaluasi Kinerja Sistem Okupansi Video 1

Tracker+ Format	Event IN	Event OUT	MAE IN	MAE OUT	MAPE IN (%)	MAPE OUT (%)	ACC IN (%)	ACC OUT (%)
ByteTrack + .ncnn	6	4	0	0	0	0	100	100
ByteTrack + .Pt	6	4	0	0	0	0	100	100
SORT + .ncnn	5	4	1	0	16.66	0	83.33	100
SORT+ .Pt	5	4	1	0	16.66	0	83.33	100

Berdasarkan Tabel 13 Hasil Evaluasi Kinerja Sistem Okupansi Video 1, ByteTrack menunjukkan performa yang sempurna pada seluruh indikator pengukuran. Baik menggunakan format .ncnn maupun .pt, ByteTrack berhasil mendeteksi 6 Event IN dan 4 Event OUT dengan tingkat kesalahan nol (MAE = 0, MAPE = 0%), menghasilkan Akurasi mutlak 100%. Di sisi lain, SORT mengalami sedikit galat pada penghitungan objek masuk (IN) dengan nilai MAE

sebesar 1 dan MAPE 16,66%, yang menurunkan akurasi ACC IN menjadi 83,33%, walaupun akurasi OUT-nya tetap bertahan di 100%.

2. Footage Video 2

Tabel 13. Hasil Evaluasi Kinerja Sistem Okupansi Video 2

Tracker+Format	Event IN	Event OUT	MAE IN	MAPE IN (%)	ACC (%)
ByteTrack + .ncnn	7	0	0	0	100
ByteTrack + .Pt	7	0	0	0	100
SORT + .ncnn	7	0	0	0	100
SORT+ .Pt	7	0	0	0	100

Berdasarkan Tabel 14, Hasil Evaluasi Kinerja Sistem Okupansi Video 2, seluruh kombinasi algoritma pelacakan dan format model menunjukkan performa yang sama dan optimal. Baik ByteTrack maupun SORT (pada format .ncnn dan .Pt) berhasil mencatatkan 7 Event IN dan 0 Event OUT secara tepat. Tidak ditemukannya kesalahan membuat seluruh sistem pada pengujian video kedua mencapai tingkat akurasi sempurna yaitu 100%.

3. Footage Video 3

Tabel 14. Hasil Evaluasi Kinerja Sistem Okupansi Video 3

Tracker+ Format	Event IN	Event OUT	MAE IN	MAE OUT	MAPE IN (%)	MAPE OUT (%)	ACC IN (%)	ACC OUT (%)
ByteTrack + .ncnn	10	8	0	2	0	25	100	75
ByteTrack + .Pt	10	8	0	2	0	25	100	75
SORT + .ncnn	10	8	0	2	0	25	100	75
SORT+ .Pt	10	8	0	2	0	25	100	75

Berdasarkan Tabel 15, Hasil Evaluasi Kinerja Sistem Okupansi Video 3, seluruh kombinasi algoritma menunjukkan performa yang identik. Pada deteksi objek masuk, semua kombinasi berhasil mencapai akurasi sebesar 100% dari total 10 Event IN. Namun, seluruh sistem mengalami kendala yang sama pada deteksi objek keluar, di mana tercatat nilai *error* MAE OUT sebesar 2 dan MAPE OUT mencapai 25%, sehingga akurasi keluar secara kompak menurun di angka 75%.

G. Evaluasi Kinerja Sistem Pada Raspberry Pi 5

Evaluasi kinerja sistem pada Raspberry Pi 5 ini bertujuan untuk memastikan bahwa Raspberry Pi dapat mengomputasi data video yang digunakan dengan optimal dan efisien. Terdapat tiga pengujian dengan 3 karakteristik video data yang digunakan dalam penelitian ini. Video data 1 memiliki karakteristik pencahayaan yang normal, video data 2 memiliki perbedaan sudut pandang kamera yang berbeda, dan video data 3 memiliki karakteristik pencahayaan gelap-terang. Skema pengujian untuk mengevaluasi kinerja sistem pelacakan dilakukan dengan menghitung lama waktu yang diperlukan Raspberry Pi 5 dalam mengolah suatu video data dan memperkirakan nilai FPS (Frame Per Second) yang diperoleh selama pengujian. Berikut adalah hasil dari masing-masing pengujian video data.

1. Footage Video 1

Tabel 15. Hasil Evaluasi Kinerja Sistem Pada Raspberry Pi 5 Video 1

Tracking/format	FPS	Waktu Inferensi
ByteTrack/.ncnn	~12-14	4 menit 12 detik
ByteTrack/.Pt	~2,5	21 menit 44 detik
SORT/.ncnn	~12-14	4 menit 8 detik
SORT/.Pt	~2,5	21 menit 44 detik

Berdasarkan Tabel 16 menampilkan hasil fps dan waktu inferensi yang dibutuhkan Raspberry Pi 5 untuk mengolah video 1 yang berdurasi 110 detik, Raspberry Pi 5 harus memproses total sekitar 3.300 frame video dengan 10 kejadian orang keluar atau masuk ruangan. Format .ncnn berhasil memproses video secara real-time atau bahkan lebih cepat, karena dengan kecepatan ~12-14 FPS, proses pengolahan data selesai hanya dalam waktu ~4 menit. Sebaliknya, format .Pt mengalami beban komputasi yang sangat berat, kecepatan yang rendah di ~2,5 FPS memaksa sistem membutuhkan waktu hingga 21 menit 44 detik untuk menyelesaikan video yang durasinya bahkan tidak sampai 2 menit.

2. Footage Video 2

Tabel 16. Hasil Evaluasi Kinerja Sistem Pada Raspberry Pi 5 Video 2

Tracking/format	FPS	Waktu Inferensi
ByteTrack/.ncnn	~12-14	2 menit 34 detik
ByteTrack/.Pt	~2,5	13 menit 21 detik
SORT/.ncnn	~12-14	2 menit 34 detik
SORT/.Pt	~2,5	13 menit 19 detik

Berdasarkan Tabel 17 menampilkan hasil fps dan waktu inferensi yang dibutuhkan Raspberry Pi 5 untuk mengolah video 2 yang berdurasi 70 detik, Raspberry Pi 5 harus memproses total sekitar 2.024 frame video dengan 7 kejadian orang masuk kedalam ruangan. Format .ncnn menunjukkan efisiensi tinggi dengan menyelesaikan seluruh inferensi hanya dalam waktu 2 menit 34 detik. Di sisi lain, format .Pt tetap menunjukkan performa yang tidak efisien karena membutuhkan waktu ~13 menit untuk memproses video singkat ini akibat keterbatasan kecepatannya yang stagnan di ~2,5 FPS.

3. Footage Video 3

Tabel 17. Hasil Evaluasi Kinerja Sistem Pada Raspberry Pi 5 Video 3

Tracking/format	FPS	Waktu Inferensi
ByteTrack/.ncnn	~12-14	6 menit 6 detik
ByteTrack/.Pt	~2,5	31 menit 33 detik
SORT/.ncnn	~12-14	6 menit 0 detik
SORT/.Pt	~2,5	31 menit 36 detik

Berdasarkan Tabel 18 menampilkan hasil fps dan waktu inferensi yang dibutuhkan Raspberry Pi 5 untuk mengolah video 3 yang berdurasi 160 detik, Raspberry Pi 5 harus memproses total sekitar 4.875 frame video dengan 18 ke-

jadi orang keluar atau masuk ruangan. Dengan durasi terpanjang Footage Video 3 menjadi ujian komputasi terbesar bagi kedua format. Format .ncnn membuktikan skalabilitasnya yang stabil karena tetap mampu mempertahankan kecepatan ~12-14 FPS dan menyelesaikan pengolahan video dalam waktu 6 menit. Berbanding terbalik, akibat durasi video yang panjang dan kecepatan yang sangat lambat di ~2,5 FPS, format .pt membutuhkan waktu pengolahan yang sangat lama, yaitu selama 31 menit 33-36 detik.

V.KESIMPULAN

Sistem deteksi okupansi ruangan berbasis *computer vision* berhasil dirancang dan diimplementasikan menggunakan Raspberry Pi 5 dengan memanfaatkan model deteksi objek YOLO11n yang dikombinasikan dengan dua metode pelacakan objek, yaitu SORT Tracking dan ByteTrack

Kombinasi YOLOv11n dengan algoritma pelacakan SORT dan ByteTrack berhasil melakukan deteksi, pelacakan, serta perhitungan objek manusia dengan tepat dan akurat dengan bantuan dua garis virtual sebagai sistem yang bertugas untuk menghitung orang yang akan keluar atau masuk dalam ruangan. Berdasarkan pengujian *Confusion Matrix* pada tiga skenario video, sistem memperoleh nilai rata-rata *precision* sebesar 90,46%, *recall* sebesar 89,08%, dan *F1-Score* sebesar 92,67%.

Sistem mampu melakukan proses deteksi objek orang masuk dan keluar ruangan dengan menghasilkan nilai rata-rata dari *precision* sebesar 90.46%, nilai *recall* sebesar 89,08% serta *F1-Score* sebesar 92.67%.

Hasil pengujian kinerja komputasi pada Raspberry Pi 5 menunjukkan bahwa penggunaan format model yang telah dioptimalisasi (.ncnn) memberikan performa yang lebih optimal dengan kecepatan pemrosesan mencapai sekitar 12–14 FPS dibanding format bawaan ultralytics (.pt) yang hanya mencapai nilai sekitar 2,5 FPS.

Implementasi *computer vision* menggunakan Raspberry Pi 5 dinilai masih memiliki keterbatasan apabila diterapkan untuk kebutuhan pemrosesan *real-time* dengan beban komputasi yang tinggi.

Sistem perhitungan okupansi bekerja lebih baik pada saat memproses orang masuk ruangan dibanding saat keluar dari ruangan, dengan jumlah kegagalan sistem dalam menghitung objek masuk sejumlah 2 kali kejadian dan orang keluar sejumlah 8 kali kejadian.

Saran

Pengembangan penelitian selanjutnya disarankan untuk menerapkan metode *motion detection* sebagai mekanisme pemicu pada sistem agar proses deteksi hanya dilakukan ketika terdapat pergerakan pada area pengamatan. Pendekatan ini diharapkan mampu mengurangi beban komputasi pada Raspberry Pi 5 sehingga performa sistem, khususnya nilai Frame Per Second (FPS), dapat meningkat dan proses pemantauan dapat berjalan lebih optimal secara *real-time*.

Selain itu, penelitian berikutnya disarankan untuk melakukan pengujian menggunakan berbagai varian model

YOLO11n maupun perangkat komputasi yang berbeda guna memperoleh perbandingan performa yang lebih komprehensif. Dengan adanya pengujian yang lebih luas, penelitian akan menghasilkan data evaluasi yang lebih komprehensif sehingga dapat digunakan sebagai acuan dalam menentukan kombinasi model deteksi dan perangkat keras yang paling optimal untuk implementasi sistem deteksi okupansi berbasis *computer vision*.

PERNYATAAN

Pernyataan Penggunaan Kecerdasan Artifisial

Dalam proses penyusunan artikel ini, penulis menggunakan Claude AI dilakukan secara terbatas, terutama untuk membantu meringkas informasi atau pembahasan yang panjang menjadi poin-poin penting, memperjelas susunan kalimat, memeriksa tata bahasa, serta membantu melakukan penyuntingan awal agar tulisan menjadi lebih efektif, sistematis, dan mudah dipahami. Informasi dan hasil yang diberikan oleh Claude AI tidak digunakan secara langsung tanpa melalui proses pemeriksaan dan penyesuaian oleh penulis.

Kontribusi Penulis

Setiap penulis memberikan kontribusi sesuai dengan perannya dalam penelitian, meliputi perumusan dan perancangan penelitian, pelaksanaan penelitian, pengumpulan data, pengolahan dan analisis hasil penelitian, serta penyusunan dan penyuntingan artikel. Seluruh penulis turut berkontribusi dalam meninjau dan menyetujui hasil akhir artikel.

Pendanaan

Penelitian ini tidak memperoleh pendanaan dari pihak eksternal.

Konflik Kepentingan

Penulis menyatakan tidak terdapat konflik kepentingan yang berkaitan dengan penelitian dan publikasi artikel ini.

VI. DAFTAR PUSTAKA

- [1] R. Szeliski, "Computer Vision: Algorithms and Applications 2nd Edition," 2021.
- [2] A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, "Object Detection Algorithms for Video Surveillance Applications," in *Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing, ICCSP 2018*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 563–568. doi: 10.1109/IC-CSP.2018.8524461.
- [3] M. Sofyan, A. Soetedjo, T. S. Elektro, and I. Malang, "Sistem Manajemen Energi untuk Lampu Penerangan dan Beban Peralatan Listrik Berbasis Okupansi di Laboratorium EBT Teknik Elektro ITN Malang," 2023.

- [4] A. Amsar, K. Khairuman, and M. Marlina, "PERANCANGAN ALAT PENDETEKSI CO2 MENGGUNAKAN SENSOR MQ-2 BERBASIS INTERNET OF THING," *METHOMIKA Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 4, no. 1, pp. 73–79, Apr. 2020, doi: 10.46880/jmika.Vol4No1.pp73-79.
- [5] I. M. Aryanto, S. Mansur, and A. Sintianingrum, "Sistem Penghitung Jumlah Pengunjung Ruangan Secara Otomatis," *INSOLOGI: Jurnal Sains dan Teknologi*, vol. 2, no. 1, pp. 157–164, Feb. 2023, doi: 10.55123/insologi.v2i1.1571.
- [6] D. Sayoga, P. D. Kusuma, and F. C. Hasibuan, "Pengembangan Sistem Deteksi Occupancy Menggunakan Computer Vision untuk Smart Building dan Automation," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 4751–4758, Aug. 2020.
- [7] K. Daniel, R. Muhammad, and B. Fajar, "Identifikasi Jumlah dan Tingkat Aktivitas Orang Berbasis Pengolahan Citra Menggunakan Raspberry Pi," *JURNAL TEKNIK ITS Vol. 6, No. 1, (2017) ISSN: 2337-3539*, 2017.
- [8] Hardisal, R. A. Candra, Ihsan, D. N. Ilham, and E. Sipahutar, "Perancangan Smart Security Camera dengan Model Image Processing Menggunakan Raspberry Pi," *METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 3, no. 2, pp. 105–111, Oct. 2019.
- [9] M. Cruz, J. J. Keh, R. Deticio, C. V. Tan, J. A. Jose, and E. Dadios, "A People Counting System for Use in CCTV Cameras in Retail," in *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020. doi: 10.1109/HNICEM51456.2020.9400048.
- [10] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Proc. European Conf. Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21–37.
- [11] A. A. Murat and M. S. Kiran, "A Comprehensive Review on YOLO Versions for Object Detection," *Engineering Science and Technology, an International Journal*, vol. 68, Art. no. 102161, Oct. 2025, doi: 10.1016/j.jestch.2025.102161.
- [12] D. A. Amer, N. Y. Ibrahim, I. K. Ibrahim, A. M. Mohamed, and S. A. Soliman, "Intelligent eyes on water: YOLOv11-based real-time drowning detection system," *Journal of Supercomputing*, vol. 81, no. 12, 2025, doi: 10.1007/s11227-025-07732-7.
- [13] T. A. A. H. Kusuma, K. Usman, and S. Saidah, "PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD," *Jurnal Teknik Informatika (Jutif)*, vol. 2, no. 1, pp. 57–66, Feb. 2021, doi: 10.20884/1.jutif.2021.2.2.77.

- [14] S. S. dan C. S. M. R. A. Pangestu, "Real Time CCTV Deteksi Wajah dengan Haar Cascade Classifiers OpenCV," *e-Proceeding of Engineering*, vol. 6, pp. 3037–3044, Sep. 2019.
- [15] F. D. Adhinata, M. Ikhsan, and W. Wahyono, "People counter on CCTV video using histogram of oriented gradient and Kalman filter methods," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 222–227, Jul. 2020, doi: 10.14710/jtsis-kom.2020.13660.
- [16] M. H. Saputra, D. Erwanto, and R. F. Rizal, "Penghitungan Jumlah Pengunjung Objek Wisata Dengan Metode Deep Learning MobileNet-SSD," *Techné: Jurnal Ilmiah Elektroteknika*, vol. 21, no. 2, pp. 145–154, Oct. 2022, doi: 10.31358/techne.v21i2.313.
- [17] K. Prasetya, A. Aribowo, A. Satyaputra, and J. O. Tjahyadi, "Implementation of Tensorflow in the CCTV-Based People Counter Application at PT Matahari Department Store, Tbk," *IAIC International Conference Series*, vol. 3, no. 1, pp. 38–44, Nov. 2020, doi: 10.34306/conferenceseries.v3i1.380.
- [18] W. I. Kusumawati, H. Pratikno, and Y. P. Admaja, "Sistem Penghitung Jumlah Pengunjung Restoran Menggunakan Kamera Berbasis Single Shot Detector (SSD)," *Journal of Technology and Informatics (JoTI)*, vol. 3, no. 1, pp. 19–26, Oct. 2021, doi: 10.37802/joti.v3i1.197.