

Pemantauan Keamanan *Real-Time* pada *Base Transceiver Station* (BTS) Menggunakan YOLOv8 dan Integrasi Telegram: Optimasi Model dan Evaluasi Performa

Muhammad Sobirin^{1,*}, Panji Wijonarko²

¹Program Studi Teknik Elektro, Universitas 17 Agustus 1945 Jakarta, Indonesia

²Program Studi Informatika, Universitas 17 Agustus 1945 Jakarta, Indonesia

¹muhammad.sobirin@uta45jakarta.ac.id; ²panji.wijonarko@uta45jakarta.ac.id;

*penulis korespondensi

INFO ARTIKEL

Sejarah Artikel

Diterima: 19 November 2025

Direvisi: 15 Desember 2025

Diterbitkan: 31 Desember 2025

Kata Kunci

Deteksi Orang
Integrasi Telegram
Keamanan BTS
Sistem Pengawasan
YOLOv8

ABSTRAK

Infrastruktur *Base Transceiver Station* (BTS) memiliki peran vital dalam telekomunikasi namun menghadapi risiko keamanan fisik yang tinggi, seperti pencurian dan vandalisme, yang dapat mengganggu ketersediaan jaringan. Metode pemantauan manual yang ada saat ini dinilai tidak efisien karena membutuhkan sumber daya intensif dan memiliki latensi respon yang tinggi terhadap pelanggaran keamanan. Penelitian ini mengusulkan optimalisasi sistem pengawasan otomatis *real-time* menggunakan algoritma *Deep Learning* YOLOv8 yang terintegrasi dengan notifikasi IoT berbasis Telegram. Empat varian arsitektur model (YOLOv8n, YOLOv8n-p2, YOLOv8n-p6, dan YOLOv8s) dievaluasi secara komparatif menggunakan metrik *mean Average Precision* (mAP), *F1-score*, dan kecepatan inferensi (*Frames Per Second*/FPS). Evaluasi dilakukan menggunakan *Human Dataset* yang terdiri dari 17.300 citra dengan pelatihan pada platform *Google Colab* dan pengujian pada perangkat *edge* NVIDIA Jetson Nano. Hasil eksperimen menunjukkan adanya *trade-off* signifikan antara akurasi dan kecepatan; YOLOv8s mencapai akurasi tertinggi dengan mAP@0.5 sebesar 61,4%, namun dengan kecepatan inferensi rendah (9,67 FPS). Sebaliknya, YOLOv8n menawarkan keseimbangan optimal dengan mAP@0.5 sebesar 59,3% dan kecepatan tertinggi mencapai 22,02 FPS. Sementara itu, varian modifikasi YOLOv8n-p2 (14,84 FPS) dan YOLOv8n-p6 (21,18 FPS) menunjukkan kemampuan kompetitif dalam menangani variasi skala objek namun tidak melampaui efisiensi YOLOv8n. Secara praktis, penelitian ini merekomendasikan implementasi YOLOv8n pada perangkat *edge* berdaya rendah karena kemampuannya memproses video secara *real-time* dan mengirimkan peringatan dini via Telegram secara instan, sehingga secara signifikan meningkatkan responsivitas sistem keamanan BTS.

PENDAHULUAN

Base Transceiver Station (BTS) merupakan komponen fundamental dalam infrastruktur telekomunikasi modern yang berfungsi sebagai penghubung utama antara perangkat pengguna dan jaringan inti operator. Gangguan terhadap keamanan fisik BTS, seperti pencurian komponen, vandalisme, atau akses tidak sah, dapat menyebabkan terhentinya layanan komunikasi (*downtime*) dan menimbulkan kerugian masif. Urgensi keamanan fisik ini didukung oleh data empiris yang signifikan. Laporan *International Telecommunication Union* (ITU) [1] mencatat bahwa sekitar 15% gangguan jaringan di negara berkembang berakar pada masalah keamanan fisik infrastruktur. Senada dengan hal tersebut, kajian *European Union Agency for Network and Information Security* (ENISA) menunjukkan

bahwa insiden ketidaktersediaan layanan telekomunikasi publik dapat memicu dampak domino sosial-ekonomi yang luas [2]. Di konteks nasional, laporan tahunan operator telekomunikasi di Indonesia mencatat lebih dari 200 kasus pencurian dan vandalisme BTS per tahun, yang secara langsung berdampak pada kerugian operasional dan reputasi penyedia layanan. Fakta-fakta ini menegaskan kebutuhan mendesak akan sistem keamanan cerdas yang proaktif.

Saat ini, pengawasan BTS umumnya masih bergantung pada metode tradisional seperti patroli fisik berkala atau pemantauan CCTV konvensional (pasif). Metode ini memiliki kelemahan mendasar: intensif sumber daya manusia, sulit menjamin pengawasan 24 jam yang efektif, dan memiliki latensi respons yang tinggi karena mengandalkan pengamatan operator manusia. Oleh karena itu, diperlukan transformasi menuju pengawasan otomatis berbasis *Artificial Intelligence* (AI). Dalam penelitian ini, fokus utama dibatasi secara spesifik pada deteksi manusia (kelas “*person*”), mengingat ancaman fisik mayoritas berasal dari intrusi manusia, bukan objek lain.

Perkembangan algoritma *Deep Learning*, khususnya seri *You Only Look Once* (YOLO), menawarkan solusi menjanjikan untuk deteksi objek *real-time*. YOLOv8, sebagai iterasi terkini, menghadirkan arsitektur canggih yang mendukung berbagai tugas visi komputer [3], [4]. Meskipun demikian, literatur menunjukkan bahwa deteksi objek kecil (seperti manusia yang terlihat dari jarak jauh pada kamera sudut lebar) masih menjadi tantangan teknis. Beberapa studi terdahulu berupaya mengatasi hal ini melalui modifikasi arsitektur, seperti penambahan *detection head* pada resolusi tinggi (P2) atau optimasi *Feature Pyramid Network* (FPN) untuk meningkatkan sensitivitas terhadap objek kecil [5], [6]. Selain itu, tren riset keamanan fasilitas kritical kini mengarah pada integrasi *Internet of Things* (IoT) untuk notifikasi instan. Penggunaan *platform* seperti Telegram sebagai antarmuka telemetri terbukti efektif dalam mempercepat respons keamanan pada infrastruktur vital [7], [8].

Berdasarkan latar belakang tersebut, penelitian ini merumuskan beberapa masalah utama sebagai berikut:

1. Bagaimana kinerja komparatif varian model YOLOv8 (YOLOv8n, YOLOv8s, dan varian modifikasi) dalam mendeteksi objek kelas “*person*” untuk simulasi keamanan BTS?
2. Sejauh mana modifikasi arsitektur melalui penambahan *detection head* P2 (untuk objek kecil) dan P6 (untuk objek besar) berpengaruh terhadap akurasi deteksi multi-skala?
3. Bagaimana efektivitas dan efisiensi sistem saat diimplementasikan pada perangkat *edge computing* terbatas untuk menghasilkan notifikasi *real-time* berbasis IoT?

Sejalan dengan rumusan masalah di atas, tujuan penelitian ini adalah:

1. Mengevaluasi performa deteksi manusia menggunakan model YOLOv8n dan YOLOv8s dalam konteks pengawasan keamanan.
2. Menganalisis dampak penambahan *detection head* P2 dan P6 terhadap peningkatan akurasi deteksi objek kecil.
3. Mengukur kinerja sistem pada perangkat *edge* (Jetson Nano) ditinjau dari aspek latensi, *throughput* (FPS), dan kecepatan pengiriman notifikasi via Telegram.

Penelitian ini memiliki batasan masalah sebagai berikut: (1) Deteksi dibatasi pada satu kelas objek (*single-class detection*), yaitu “*person*”; (2) Pengujian menggunakan dataset publik (*Human Dataset*) sebagai representasi lingkungan visual BTS; (3) Sistem tidak mencakup pengenalan wajah (*face recognition*) atau identitas individu; dan (4) Tidak membahas deteksi ancaman lingkungan lain seperti kebakaran atau banjir.

Penelitian ini diharapkan memberikan kontribusi signifikan dalam dua aspek. Secara teoretis, penelitian ini berkontribusi pada literatur optimasi arsitektur YOLOv8 untuk deteksi

objek kecil pada perangkat dengan daya komputasi rendah. Secara praktis, hasil penelitian menawarkan solusi keamanan infrastruktur telekomunikasi yang hemat biaya, otomatis, dan responsif guna meminimalkan risiko vandalisme pada BTS.

METODE

Penelitian Terkait

Meningkatnya kebutuhan akan langkah-langkah keamanan yang kuat di BTS berasal dari peran penting fasilitas ini dalam menjaga kelancaran komunikasi nasional. Pengawasan manual tradisional pada lokasi BTS memerlukan banyak tenaga, waktu, dan rentan terhadap celah keamanan. Penerapan sistem pengawasan *real-time* berbasis algoritma deteksi objek modern menawarkan solusi menjanjikan untuk meningkatkan keamanan fisik dan memberikan peringatan dini terhadap potensi ancaman. Penelitian ini berfokus pada optimalisasi deteksi manusia dalam konteks keamanan BTS dengan memanfaatkan kemampuan YOLOv8 yang dimodifikasi dan integrasi *Internet of Things* (IoT) melalui Telegram untuk notifikasi *real-time*.

Algoritma deteksi objek berbasis *deep learning* telah mengalami perkembangan pesat, terutama melalui seri YOLO yang dikenal karena efisiensi dan kemampuannya dalam mendeteksi objek secara *real-time*. Sejak YOLOv1 hingga YOLOv8, setiap iterasi membawa peningkatan signifikan dari segi arsitektur, kecepatan inferensi, dan akurasi deteksi. YOLOv8 menghadirkan kemajuan melalui dukungan terpadu untuk *object detection*, *instance segmentation*, dan *pose estimation*. Versi ini menyediakan varian seperti YOLOv8n (*nano*) dan YOLOv8s (*small*) yang dirancang untuk perangkat dengan keterbatasan daya komputasi [9]. Model YOLOv8n, misalnya, memiliki ukuran ringan dan waktu inferensi cepat sehingga cocok untuk penerapan pada perangkat *edge*, sedangkan YOLOv8s menawarkan keseimbangan yang baik antara akurasi dan kecepatan [3], [4].

Meskipun demikian, deteksi objek kecil seperti manusia dari jarak jauh atau dalam kondisi visual yang kompleks masih menjadi tantangan dalam visi komputer. Penelitian sebelumnya menunjukkan bahwa peningkatan deteksi target kecil dapat dicapai melalui optimasi *feature extraction* dan *feature fusion* menggunakan FPN [10]. Beberapa studi lanjutan telah berhasil menambahkan *detection head* pada resolusi tinggi (P2) untuk memperbaiki deteksi objek kecil secara signifikan, serta memperluas arsitektur dengan penambahan *detection head* P6 guna memperkuat deteksi objek besar dalam citra beresolusi tinggi [11]. Pendekatan modifikasi arsitektur ini sering diiringi dengan strategi fusi multi-skala dan mekanisme atensi untuk meningkatkan sensitivitas model terhadap objek di berbagai ukuran [12].

Di sisi lain, integrasi sistem deteksi cerdas dengan teknologi IoT menjadi bagian krusial dalam mendukung keamanan fasilitas kritis. Sistem berbasis IoT memungkinkan pemantauan jarak jauh, pengumpulan data *real-time*, dan pengiriman peringatan otomatis [13]. Platform pesan instan seperti Telegram telah terbukti efektif digunakan sebagai antarmuka telemetri pada berbagai sistem deteksi otomatis seperti pemantauan kesehatan dan deteksi kebakaran karena latensi yang rendah, API yang terbuka, dan kemudahan integrasi dengan mikrokontroler atau mini-PC [14], [15]. Integrasi ini memastikan bahwa notifikasi langsung dikirim ke perangkat seluler operator segera setelah aktivitas mencurigakan terdeteksi di sekitar BTS.

Tren terbaru dalam penelitian pengawasan cerdas juga menyoroti peran penting *edge AI*, yaitu implementasi model AI langsung pada perangkat di lapangan tanpa ketergantungan penuh pada *cloud server*. Pendekatan ini mampu mengurangi latensi jaringan, menghemat *bandwidth*, dan meningkatkan keandalan sistem keamanan *real-time*, terutama di lokasi terpencil dengan konektivitas terbatas. Penelitian terkini menunjukkan bahwa kombinasi

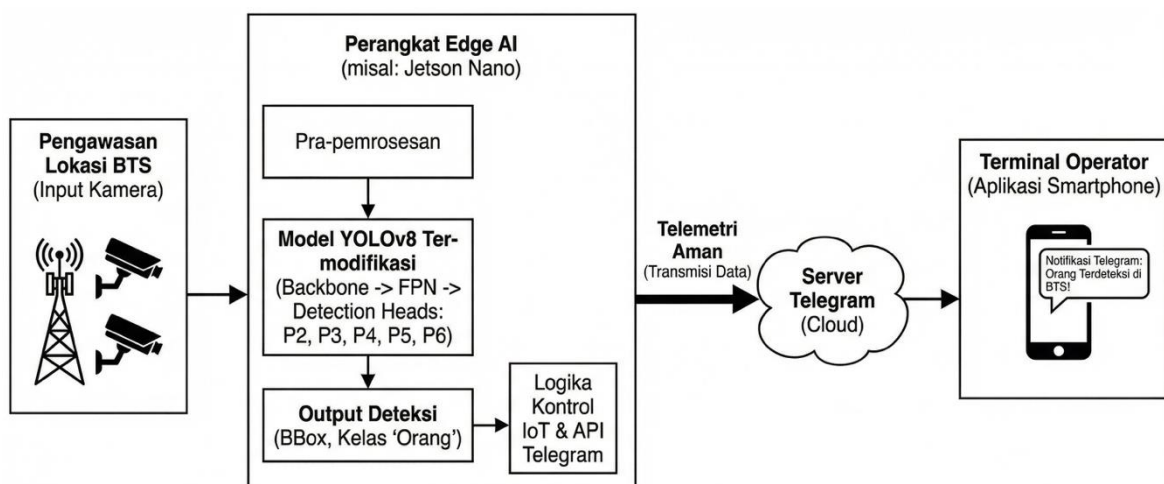
model YOLO ringan dengan perangkat *edge computing* seperti NVIDIA Jetson Nano atau Raspberry Pi dapat mencapai waktu respons di bawah 300 ms dengan konsumsi daya yang rendah [16], [17], [18], [19]. Dengan demikian, integrasi antara model YOLOv8 yang dioptimalkan dan perangkat *edge AI* menjadi solusi ideal untuk karakteristik lokasi BTS.

Secara keseluruhan, berbagai penelitian telah berfokus pada modifikasi arsitektur YOLO untuk meningkatkan performa deteksi dalam berbagai konteks, termasuk lalu lintas, industri, dan medis [20], [21], [22], [23]. Namun, sebagian besar studi tersebut belum secara spesifik menyoroti aspek keamanan fisik pada infrastruktur telekomunikasi. Padahal, konteks BTS memiliki tantangan unik seperti kondisi lingkungan luar ruang, keterbatasan catu daya, dan kebutuhan sistem *alerting* yang ringan namun sangat responsif. Hal ini menunjukkan masih adanya ruang eksplorasi dalam mengadaptasi dan memodifikasi YOLOv8 untuk mendeteksi ancaman manusia di lingkungan BTS secara efisien.

Berdasarkan tinjauan literatur tersebut, teridentifikasi beberapa kesenjangan penelitian (*research gap*) utama:

1. Kajian tentang deteksi manusia yang dikhususkan untuk tantangan lingkungan dan keamanan BTS masih terbatas dibandingkan aplikasi umum lainnya [11], [24].
2. Analisis komparatif mengenai pengaruh modifikasi spesifik penambahan *detection head* (P2 untuk objek kecil dan P6 untuk objek besar) pada YOLOv8 dalam konteks area BTS belum dilakukan secara menyeluruh [10].
3. Integrasi menyeluruh antara sistem deteksi YOLO yang dimodifikasi di perangkat *edge* dengan mekanisme telemetri notifikasi *real-time* berbasis Telegram belum banyak dibahas secara terpadu dalam literatur [13], [14], [15].
4. Pendalaman terkait bagaimana sistem keamanan fisik dapat dioptimalkan melalui kolaborasi antara *edge AI*, algoritma deteksi ringan, dan sistem peringatan dini masih menjadi area penelitian potensial.

Penelitian ini menghadirkan kebaruan dengan mengatasi kesenjangan tersebut melalui tiga aspek utama. Pertama, fokus pada pengoptimalan deteksi kelas "*person*" secara spesifik untuk konteks keamanan BTS, yang jarang dieksplorasi dalam penelitian terdahulu. Kedua, modifikasi arsitektur YOLOv8 dengan penambahan *detection head* P2 dan P6 dikaji secara komparatif untuk meningkatkan deteksi multi-skala pada lingkungan nyata BTS. Ketiga, integrasi sistem deteksi YOLOv8 dengan mekanisme notifikasi *real-time* berbasis IoT pada perangkat *edge* dirancang untuk menciptakan sistem keamanan yang tangguh, efisien, dan siap diterapkan di lapangan.



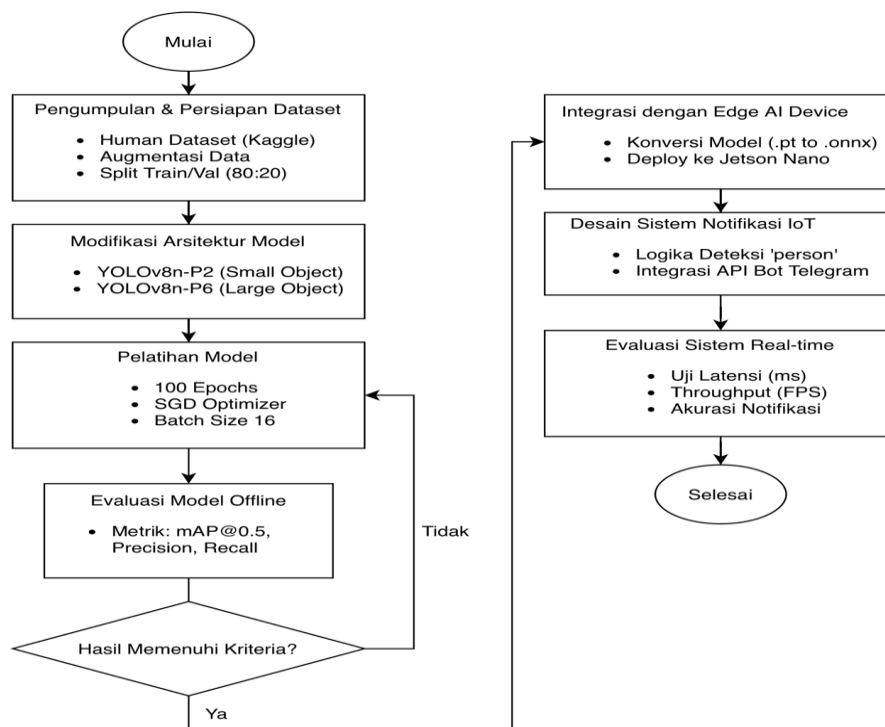
Gambar 1. Kerangka Konseptual Sistem Keamanan BTS Berbasis *Edge AI* dan IoT.

Sinergi antara konsep-konsep utama dalam penelitian ini (mulai dari akuisisi visual, pemrosesan AI dengan modifikasi arsitektur, hingga telemetri notifikasi) digambarkan dalam kerangka konseptual pada Gambar 1. YOLOv8 berfungsi sebagai inti deteksi, yang dimodifikasi melalui *multi-scale feature fusion* (FPN) dengan penambahan *detection head* P2 dan P6 untuk memastikan akurasi deteksi manusia pada berbagai skala di area BTS. Model ini dijalankan pada perangkat *edge AI* untuk menjamin respons *real-time* di lapangan. Selanjutnya, hasil deteksi positif memicu mekanisme IoT untuk mengirimkan peringatan dini kepada operator melalui Telegram, menciptakan siklus deteksi-notifikasi yang efisien. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi sistem deteksi manusia berbasis Edge AI dan YOLOv8 yang dioptimalkan untuk keamanan BTS dengan notifikasi *real-time* via Telegram. Metodologi penelitian mencakup persiapan dataset, modifikasi arsitektur YOLOv8n, desain eksperimen, dan evaluasi performa sistem.

Persiapan Dataset

Penelitian ini menggunakan *Human Dataset* yang diunduh dari *platform Kaggle* sebagai sumber data utama. Dataset ini dirancang secara spesifik untuk mendeteksi keberadaan manusia (*person*) dalam gambar dengan tingkat presisi anotasi yang tinggi. Dataset mencakup total 17.300 gambar dengan dua jenis format anotasi: *bounding box* dalam koordinat piksel pada resolusi asli gambar dan *bounding box* dalam format YOLO yang telah dinormalisasi. Sumber data ini diperoleh melalui *toolkit* *OIDv4*, yang memanfaatkan *Google Open Images Dataset* untuk menyediakan gambar yang relevan dalam deteksi manusia. Resolusi gambar yang tinggi serta anotasi *bounding box* yang akurat memberikan dukungan optimal bagi model *deep learning* dalam melakukan proses deteksi objek. Dataset kemudian dibagi menjadi dua subset: subset pelatihan yang berisi 13.800 gambar (sekitar 80%) untuk proses pembelajaran model, dan subset validasi sebanyak 3.500 gambar (sekitar 20%) untuk mengukur performa model selama pelatihan dan *fine-tuning*.

Tahap-Tahap Penelitian



Gambar 2. Bagan Tahap-Tahap Penelitian.

Tahapan penelitian ini disusun dalam kerangka kerja sistematis yang diilustrasikan pada Gambar 2, mencakup langkah-langkah berikut:

1. Pengumpulan dan Persiapan Dataset: Mengunduh dan memverifikasi Human Dataset dari Kaggle, memastikan kualitas anotasi dan kelengkapan data. Pembagian dataset menjadi data latih dan validasi dilakukan dengan rasio 80:20.
2. Modifikasi Arsitektur Model: Implementasi dua varian model YOLOv8n.
 - a. YOLOv8n-P2: Modifikasi *detection head* untuk menyertakan skala P2, P3, P4, dan P5 guna meningkatkan sensitivitas terhadap objek kecil hingga menengah.
 - b. YOLOv8n-P6: Modifikasi *detection head* untuk menyertakan skala P3, P4, P5, dan P6 guna meningkatkan kemampuan deteksi objek besar hingga sangat besar.
3. Pelatihan Model (*Training*): Tahap ini melibatkan pelatihan kedua varian model (YOLOv8n-P2 dan YOLOv8n-P6) menggunakan subset pelatihan. Proses pelatihan dilaksanakan dengan konfigurasi *hyperparameter* eksperimen yang telah ditetapkan, yaitu 100 *epochs*, penggunaan *optimizer* SGD, dan *batch size* sebesar 16.
4. Evaluasi Model *Offline*: Mengevaluasi performa deteksi kedua model menggunakan subset validasi berdasarkan metrik *precision*, *recall*, dan *mean Average Precision* (mAP).
5. Integrasi dengan *Edge AI Device*: Menerapkan model terbaik yang telah dilatih dan dioptimalkan ke perangkat *Edge AI* (Jetson Nano).
6. Desain Sistem Notifikasi IoT: Mengembangkan modul IoT *control logic* dan Telegram API untuk menerima output deteksi dari model pada perangkat *edge* dan mengirimkan notifikasi *real-time* ke terminal operator.
7. Evaluasi Sistem *Real-time*: Melakukan pengujian sistem secara end-to-end di lingkungan simulasi BTS untuk mengukur performa *real-time*, termasuk latensi deteksi dan throughput notifikasi.

Arsitektur Model YOLOv8n

Pada penelitian ini, dua varian YOLOv8n yang dimodifikasi digunakan untuk eksplorasi deteksi objek multi-skala: YOLOv8n-P2 dan YOLOv8n-P6. Pemilihan varian "n" (nano) didasarkan pada kebutuhan akan model yang ringan dan efisien, sangat sesuai untuk penerapan pada perangkat *edge* dengan sumber daya komputasi terbatas, yang merupakan karakteristik umum lokasi BTS.

Arsitektur YOLOv8n-P2

YOLOv8n-P2 merupakan varian dari model YOLOv8 yang dirancang untuk deteksi objek dengan ukuran kecil hingga menengah. Arsitektur ini mengoptimalkan performa dengan menggunakan beberapa skala keluaran pada P2 hingga P5, memberikan fleksibilitas dalam mendeteksi objek dari berbagai ukuran. Bagian *backbone* dari YOLOv8n-P2 dirancang untuk mengekstraksi fitur penting dari citra input dengan struktur yang efisien dan berlapis. Komponen *backbone* diawali dengan lapisan konvolusi (Conv) yang bertugas untuk mengekstraksi fitur dasar. Proses dimulai dengan konvolusi menggunakan *kernel* 3×3 dan *stride* 2 untuk menurunkan resolusi citra input, diikuti oleh lapisan-lapisan konvolusi berikutnya yang berfungsi memperluas saluran serta menangkap fitur yang lebih mendalam. Setelah lapisan konvolusi, *backbone* dilengkapi dengan C2f (*Cross-Stage Partial Bottleneck*), yang dirancang untuk meningkatkan efisiensi pengolahan fitur tanpa meningkatkan jumlah parameter secara signifikan. Elemen ini berperan penting dalam menjaga keseimbangan antara efisiensi komputasi dan akurasi deteksi. Selanjutnya, jumlah saluran meningkat secara bertahap dari 64 hingga 1024. Pada tahap akhir *backbone*, digunakan SPPF (*Spatial Pyramid Pooling Fast*), yang memungkinkan pengumpulan

informasi spasial dari berbagai tingkat resolusi. Komponen ini memberikan keunggulan dalam mendeteksi objek pada berbagai ukuran, baik objek kecil maupun besar.

Bagian *head* dari YOLOv8n-P2 dirancang untuk meningkatkan resolusi fitur dan mendukung deteksi objek dengan berbagai ukuran secara efisien. Proses dimulai dengan lapisan *upsampling* yang bertujuan meningkatkan resolusi fitur, sehingga memungkinkan model untuk mendeteksi objek-objek kecil dengan lebih detail. Selanjutnya, fitur dari berbagai skala digabungkan melalui lapisan *concatenate* (P2, P3, P4, P5). Penambahan *detection head* P2 ini krusial untuk meningkatkan kemampuan model dalam mendeteksi objek kecil yang mungkin muncul dari jarak jauh atau di lingkungan *wide-angle* seperti area BTS.

Arsitektur YOLOv8n-P6

YOLOv8n-P6 merupakan varian dari YOLOv8 yang dirancang secara khusus untuk mendeteksi objek dengan ukuran besar hingga sangat besar. Arsitektur ini menggunakan keluaran pada skala P3 hingga P6. Bagian *backbone* YOLOv8n-P6 juga menggunakan struktur serupa dengan *kernel* 3×3 dan *stride* 2, namun dioptimalkan untuk dataset yang lebih besar. Bagian *head* YOLOv8n-P6 menggabungkan fitur melalui lapisan *concat* dari beberapa resolusi (P3, P4, P5, P6), dengan fokus utama pada skala P6. Penambahan *detection head* P6 ini penting untuk memastikan model dapat mendeteksi manusia yang mungkin berada sangat dekat dengan kamera atau memenuhi sebagian besar *frame*.

Perbandingan YOLOv8n-p2 dan YOLOv8n-p6

Kedua varian dimodifikasi dari arsitektur dasar YOLOv8n namun memiliki perbedaan mendasar pada skala deteksi output. YOLOv8n-P2 menggunakan output pada skala P2-P5 (fokus objek kecil/menengah), sedangkan YOLOv8n-P6 menggunakan skala P3-P6 (fokus objek besar). Secara kompleksitas, YOLOv8n-P6 cenderung memiliki jumlah parameter dan komputasi yang sedikit lebih besar dibandingkan YOLOv8n-P2. Meskipun demikian, keduanya tetap mempertahankan karakteristik "nano" yang ringan, sehingga cocok untuk implementasi pada perangkat edge AI.

Metrik Evaluasi

Evaluasi kinerja sistem dilakukan melalui dua pendekatan utama, yaitu evaluasi kualitas deteksi objek dan evaluasi efisiensi komputasi pada perangkat *edge*.

Kualitas Deteksi Objek

Untuk mengukur keandalan model dalam mengenali objek manusia, penelitian ini menggunakan metrik standar visi komputer. Akurasi prediksi positif diukur menggunakan *Precision* (P), sedangkan kemampuan model untuk mengidentifikasi seluruh objek relevan diukur menggunakan *Recall* (R). Kedua metrik ini didefinisikan dalam Persamaan (1) dan (2):

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

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

Dimana TP (*True Positive*) merepresentasikan jumlah deteksi manusia yang benar, FP (*False Positive*) adalah deteksi salah, dan FN (*False Negative*) adalah objek manusia yang gagal terdeteksi.

Validasi kinerja model secara menyeluruh dilakukan menggunakan *mean Average Precision* (mAP). Nilai *Average Precision* (AP) dihitung berdasarkan integrasi *Precision* P sebagai fungsi dari *Recall* R, sebagaimana ditunjukkan pada Persamaan (3):

$$AP = \int_0^1 p(r)dr \quad (3)$$

Penelitian ini melaporkan mAP pada dua ambang batas IoU: mAP@0.5 dan mAP@0.5:0.95.

Efisiensi Komputasi *Real-Time*

Mengingat implementasi dilakukan pada perangkat *edge* dengan sumber daya terbatas, efisiensi operasional menjadi parameter kritis:

- a. Latensi (*Latency*): Waktu inferensi rata-rata per *frame* (ms).
- b. *Throughput*: Jumlah *frame* per detik (FPS). Hubungan antara latensi rata-rata (L_{avg}) dalam detik dan FPS dirumuskan dalam Persamaan (4):

$$FPS = \frac{1}{L_{avg}} \quad (4)$$

Tingkat FPS yang tinggi dengan latensi yang rendah mengindikasikan kelayakan sistem untuk diterapkan dalam skenario pengawasan keamanan *real-time*.

Desain Eksperimen

Desain eksperimen ini dirancang sebagai studi komparatif terkontrol di mana variasi arsitektur *detection head* (YOLOv8n-P2 vs. YOLOv8n-P6) ditetapkan sebagai variabel independen, sementara seluruh parameter pelatihan lainnya ditetapkan sebagai variabel kontrol. Resolusi citra input distandardisasi pada 640×640 piksel. Konfigurasi hyperparameter pelatihan ditentukan melalui pendekatan empiris untuk menjamin stabilitas dan generalisasi model yang optimal. Proses optimasi memanfaatkan algoritma *Stochastic Gradient Descent* (SGD) dengan momentum 0.9 dan *weight decay* 5×10^{-4} , yang dipilih karena kemampuannya yang terbukti unggul dalam menghasilkan generalisasi model pada arsitektur YOLO serta menghindari *sharp local minima*. Ukuran *batch* ditetapkan sebesar 16, sebuah nilai yang merepresentasikan kompromi optimal antara efisiensi memori GPU (VRAM) dan stabilitas statistik *Batch Normalization*. Selanjutnya, durasi pelatihan dibatasi hingga 100 *epochs* berdasarkan validasi dari eksperimen pendahuluan yang menunjukkan konvergensi kurva *loss* yang stabil pada kisaran *epoch* ke-80 hingga 90. Laju pembelajaran (*learning rate*) awal diinisialisasi pada 0.01 dan dikelola menggunakan skema *cosine annealing scheduler*. Seluruh proses pelatihan dilakukan menggunakan lingkungan *cloud computing Google Colab Pro+* dengan akselerator grafis NVIDIA Tesla T4/A100, sedangkan validasi inferensi real-time dilakukan secara lokal pada perangkat NVIDIA Jetson Nano (CPU Quad-core ARM Cortex-A57, GPU 128-core Maxwell, 4GB RAM).

HASIL DAN PEMBAHASAN

Lingkungan Perangkat Keras dan Lunak

Eksperimen pelatihan model dilakukan menggunakan *platform cloud computing Google Colab Pro+* untuk memanfaatkan akselerasi GPU kinerja tinggi. Spesifikasi lingkungan pelatihan mencakup GPU NVIDIA A100-SXM4 dengan VRAM 40 GB, didukung oleh CPU Intel Xeon @ 2.20GHz, serta lingkungan perangkat lunak berbasis CUDA 12.2 dan driver

GPU versi 535.104.05. Penggunaan infrastruktur ini memastikan efisiensi komputasi yang tinggi, terutama dalam menangani *batch size* yang besar dan arsitektur model yang kompleks.

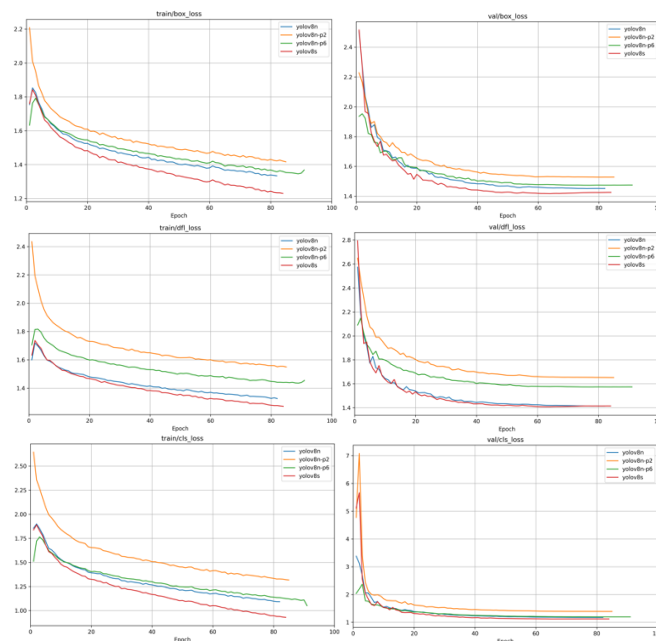
Sebaliknya, pengujian inferensi dilakukan pada perangkat *edge* dengan sumber daya terbatas untuk mensimulasikan kondisi lapangan yang sebenarnya. Perangkat uji utama adalah NVIDIA Jetson Nano Developer Kit B01 (4 GB LPDDR4 RAM, 128 CUDA core Maxwell GPU, Quad-core ARM Cortex-A57). Selain itu, pengujian perbandingan dilakukan pada laptop dengan prosesor AMD Ryzen 5 7535HS dan Radeon Graphics. Skenario ini dirancang untuk mengevaluasi kelayakan implementasi model YOLOv8 yang diusulkan dalam ekosistem IoT berdaya rendah.

Dataset dan Pre-processing

Penelitian ini menggunakan *Human Dataset* yang bersumber dari repositori Kaggle, yang dikurasi menggunakan *toolkit* OIDv4 dari *Google Open Images*. Dataset terdiri dari 17.300 citra resolusi tinggi dengan variasi skala dan pencahayaan, yang dibagi menjadi *subset* pelatihan (13.800 citra) dan validasi (4.000 citra). Seluruh anotasi *bounding box* telah dinormalisasi ke dalam format YOLO.

Hasil Eksperimen Pelatihan Model

Keempat varian model (YOLOv8n, YOLOv8n-p2, YOLOv8n-p6, dan YOLOv8s) dilatih selama 100 *epochs*. Dinamika penurunan *loss* divisualisasikan pada Gambar 3.



Gambar 3. Kurva *loss* selama pelatihan model YOLOv8n, YOLOv8n-p2, YOLOv8n-p6, dan YOLOv8s.

Berdasarkan Gambar 3, seluruh model menunjukkan konvergensi yang stabil. YOLOv8s mencatat nilai *box_loss* dan *cls_loss* terendah pada akhir pelatihan, mengindikasikan kemampuan *fitting* data yang superior. Sebaliknya, YOLOv8n-p6 menunjukkan kurva penurunan *loss* yang lebih lambat pada 20 epoch pertama, yang dapat dikaitkan dengan kompleksitas arsitektur *detection head* tambahan (P6) yang memerlukan lebih banyak iterasi untuk mempelajari fitur spasial objek besar. Tidak ditemukan tanda-tanda *overfitting* yang signifikan, ditandai dengan kurva validasi yang selaras dengan kurva pelatihan.

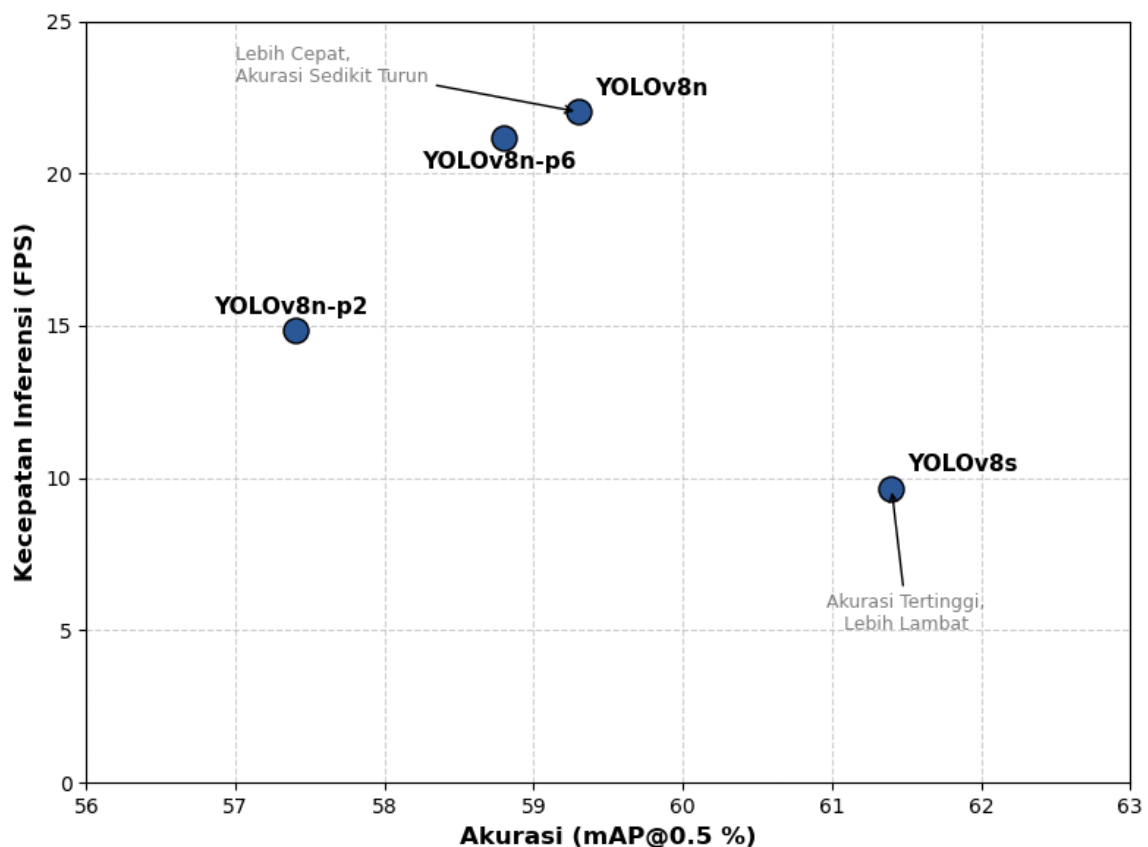
Analisa Performa Komparatif

Evaluasi kuantitatif dilakukan berdasarkan metrik *Precision* (P), *Recall* (R), *mAP@0.5*, serta efisiensi komputasi (Parameter, GFLOPs, dan FPS). Ringkasan hasil disajikan pada Tabel 1.

Tabel 1. Perbandingan Kinerja Model pada *Human Dataset*

Model	P (%)	R (%)	F1-score (%)	mAP@0.5 (%)	Param (M)	FLOPs (G)	FPS
YOLOv8n	100	86	60	59,3	3,01	8,2	22,02
YOLOv8n-p2	100	86	58	57,4	2,93	12,4	14,84
YOLOv8n-p6	100	85	59	58,8	4,78	8,2	21,18
YOLOv8s	100	86	61	61,4	11,14	28,8	9,67

Secara statistik, YOLOv8s mendominasi metrik akurasi dengan *mAP@0.5* sebesar 61,4%, unggul 2,1% dibandingkan YOLOv8n. Namun, keunggulan akurasi ini harus dibayar dengan biaya komputasi yang tinggi; model ini memiliki 11,14 juta parameter dan beban komputasi 28,8 GFLOPs, yang mengakibatkan penurunan *frame rate* signifikan menjadi hanya 9,67 FPS pada Jetson Nano. Di sisi lain, YOLOv8n muncul sebagai solusi paling efisien untuk kebutuhan *real-time*. Dengan hanya 3,01 juta parameter, model ini mencapai kecepatan inferensi tertinggi sebesar 22,02 FPS. Meskipun *mAP@0.5*-nya (59,3%) sedikit di bawah YOLOv8s, selisih performa tersebut masih dalam batas toleransi untuk aplikasi pengawasan umum.



Gambar 4. Grafik Perbandingan FPS vs *mAP@0.5* pada NVIDIA Jetson Nano yang memperlihatkan *trade-off* antara kecepatan (sumbu Y) dan akurasi (sumbu X)

Grafik perbandingan FPS pada Gambar 4 mempertegas posisi YOLOv8n sebagai model dengan latensi terendah, sementara YOLOv8s berada pada spektrum akurasi tertinggi namun dengan latensi yang kurang ideal untuk respon instan pada perangkat *edge* berdaya rendah.

Evaluasi Visual Deteksi Objek

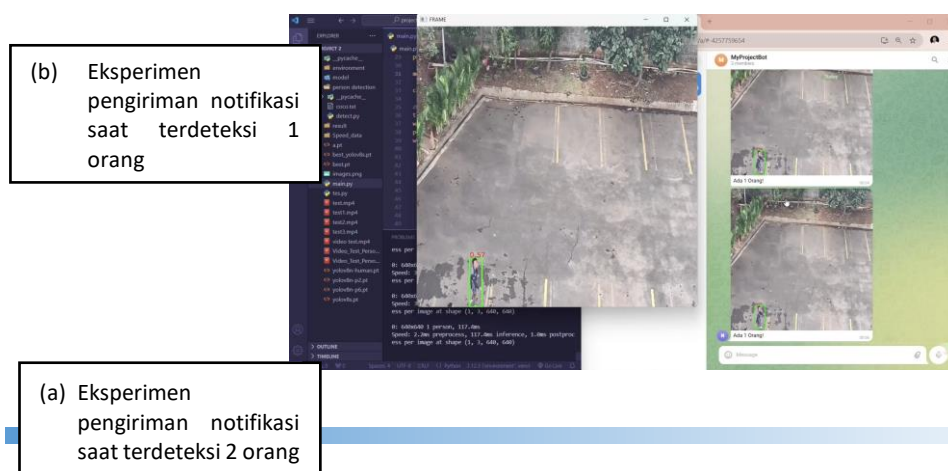
Hasil deteksi visual pada skenario lingkungan kampus ditampilkan pada Gambar 5. Analisis kualitatif pada Gambar 5 menunjukkan bahwa meskipun semua model berhasil mendeteksi keberadaan manusia, YOLOv8s memberikan tingkat kepercayaan (*confidence score*) yang lebih tinggi dan konsisten, terutama pada kondisi oklusi parsial (objek tertutup sebagian). YOLOv8n-p2, yang dirancang dengan *head* P2 resolusi tinggi, menunjukkan sensitivitas yang baik terhadap fitur-fitur detail manusia, namun terkadang menghasilkan *false positives* pada latar belakang yang kompleks.



Gambar 5. Hasil Deteksi Visual pada skenario pengawasan

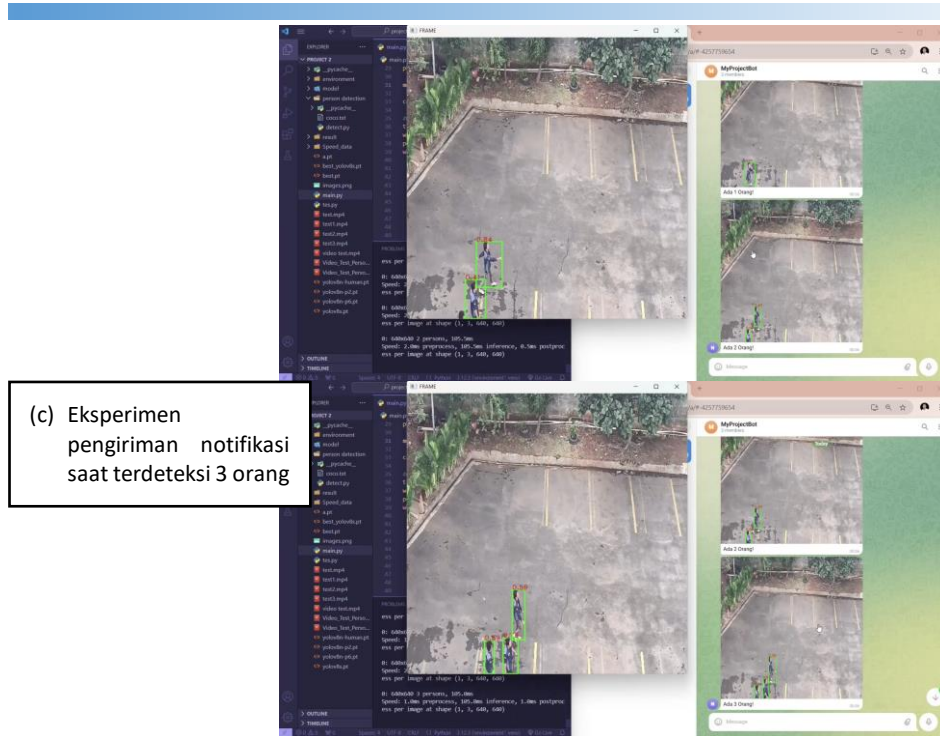
Validasi Sistem Notifikasi IoT

Integrasi dengan Telegram API diuji untuk memvalidasi fungsionalitas pelaporan insiden. Seperti terlihat pada Gambar 6, sistem berhasil mengirimkan peringatan (*alert*) berupa teks jumlah objek terdeteksi dan gambar *snapshot* dengan latensi pengiriman rata-rata di bawah 3 detik (tergantung konektivitas jaringan), yang memadai untuk standar peringatan dini keamanan fisik.



(b) Eksperimen pengiriman notifikasi saat terdeteksi 1 orang

(a) Eksperimen pengiriman notifikasi saat terdeteksi 2 orang



(c) Eksperimen pengiriman notifikasi saat terdeteksi 3 orang

Gambar 6. Contoh notifikasi Telegram yang diterima operator

Pembahasan

Analisis *Trade-off*: Akurasi vs Efisiensi Komputasi

Temuan penelitian ini mengonfirmasi teori dasar *deep learning* mengenai *trade-off* antara kedalaman model dan kecepatan inferensi. YOLOv8s mencapai mAP tertinggi karena kapasitas ekstraksi fiturnya yang lebih dalam (jumlah *layer* dan *channel* lebih banyak), namun hal ini meningkatkan beban komputasi FLOPs hampir 3,5 kali lipat dibandingkan YOLOv8n. Dalam konteks keamanan BTS yang seringkali berlokasi di area terpencil dengan pasokan daya terbatas (mengandalkan baterai atau sel surya), efisiensi energi dan kecepatan pemrosesan menjadi prioritas. Oleh karena itu, YOLOv8n dinilai sebagai kandidat paling seimbang untuk implementasi *edge*, kecuali jika sistem didukung oleh perangkat keras yang lebih kuat seperti Jetson Orin atau Xavier.

Peran Arsitektur P2 dan P6 dalam Deteksi Multi-Skala

Penambahan lapisan deteksi P2 (*high-resolution head*) pada YOLOv8n-p2 secara teoritis bertujuan untuk mempertahankan fitur spasial halus yang sering hilang setelah proses *downsampling* berulang. Hal ini sejalan dengan literatur FPN yang menyatakan bahwa fitur level rendah (P2) krusial untuk mendeteksi objek kecil. Namun, hasil eksperimen menunjukkan bahwa YOLOv8n-p2 justru mengalami penurunan FPS yang signifikan (14,84 FPS) dibandingkan YOLOv8n standar (22,02 FPS) dengan peningkatan akurasi yang marginal. Hal ini mengindikasikan bahwa beban komputasi pemrosesan fitur resolusi tinggi pada perangkat *edge* mungkin tidak sebanding dengan peningkatan akurasinya untuk kasus dataset ini. Sebaliknya, YOLOv8n-p6 yang memanfaatkan fitur level tinggi (P6) lebih efektif untuk objek besar, namun kurang relevan jika target manusia berada pada jarak jauh dari kamera pengawas BTS.

Keterbatasan Penelitian

Meskipun sistem yang diusulkan telah menunjukkan kinerja operasional yang menjanjikan, penelitian ini memiliki beberapa keterbatasan yang perlu digaribawahi

sebagai arah pengembangan masa depan. Pertama, distribusi dataset pelatihan saat ini didominasi oleh citra dengan kondisi pencahayaan siang hari yang optimal, sehingga *robustness* model dalam menghadapi kondisi lingkungan yang menantang (seperti pencahayaan rendah di malam hari atau gangguan cuaca ekstrem berupa hujan lebat dan kabut) belum dievaluasi secara ekstensif. Kedua, lingkup deteksi dibatasi secara eksklusif pada kelas "*person*", yang berarti potensi ancaman keamanan fisik lainnya, seperti intrusi kendaraan tak dikenal atau keberadaan peralatan berbahaya, belum terakomodasi dalam cakupan pelatihan model saat ini. Terakhir, validasi inferensi dilakukan terbatas pada perangkat NVIDIA Jetson Nano (4GB) yang merepresentasikan perangkat *edge* kelas *entry-level*; oleh karena itu, profil kinerja latensi dan *throughput* yang dilaporkan mungkin berbeda secara signifikan apabila sistem diimplementasikan pada akselerator AI yang lebih modern dengan kapasitas komputasi yang lebih tinggi.

Implikasi dan Rekomendasi Implementasi

Berdasarkan sintesis temuan eksperimental, penelitian ini merumuskan sejumlah rekomendasi teknis untuk penerapan sistem keamanan cerdas di lingkungan industri. Pertama, terkait pemilihan model, varian YOLOv8n direkomendasikan sebagai solusi paling optimal untuk perangkat *edge* dengan sumber daya terbatas (seperti NVIDIA Jetson Nano atau Raspberry Pi 5) mengingat kemampuannya mempertahankan *frame rate* di atas 20 FPS yang memadai untuk pelacakan pergerakan manusia secara halus; sebaliknya, varian YOLOv8s sebaiknya hanya diadopsi pada infrastruktur yang didukung oleh *on-premise server* berkinerja tinggi. Kedua, guna memenuhi standar efisiensi industri, konversi model ke format TensorRT (.engine) dengan presisi FP16 atau INT8 sangat disarankan, karena strategi optimasi ini berpotensi meningkatkan *throughput* sebesar 30–50% tanpa degradasi akurasi yang signifikan. Terakhir, untuk menjamin efisiensi operasional dan reliabilitas sistem notifikasi, disarankan penerapan arsitektur *deployment* hibrida di mana algoritma deteksi gerak ringan berfungsi sebagai pemicu inferensi YOLO, serta penetapan ambang batas *confidence score* minimal 0.60 untuk pengiriman telemetri Telegram guna memitigasi risiko notifikasi palsu (*false positive*) yang berlebihan.

KESIMPULAN

Penelitian ini berhasil mengembangkan sistem pengawasan keamanan cerdas untuk BTS dengan mengintegrasikan algoritma *deep learning* YOLOv8 pada perangkat *Edge AI* dan sistem telemetri berbasis Telegram. Secara teoretis, penelitian ini memberikan kontribusi dalam mengevaluasi efektivitas modifikasi *detection head* multi-skala (P2 dan P6) pada arsitektur YOLOv8 untuk kasus deteksi manusia di lingkungan infrastruktur kritis dengan sumber daya terbatas. Berdasarkan evaluasi empiris, ditemukan adanya *trade-off* yang signifikan antara kedalaman arsitektur model dan efisiensi komputasi. YOLOv8s mencatatkan akurasi tertinggi dengan mAP@0.5 sebesar 61,4%, namun memiliki keterbatasan pada kecepatan inferensi yang hanya mencapai 9,67 FPS pada NVIDIA Jetson Nano. Sebaliknya, YOLOv8n terbukti menjadi solusi paling optimal untuk penerapan *real-time*, menawarkan keseimbangan terbaik dengan *throughput* 22,02 FPS dan akurasi yang kompetitif (mAP@0.5 sebesar 59,3%). Sementara itu, varian modifikasi YOLOv8n-p2 dan YOLOv8n-p6, meskipun dirancang untuk meningkatkan sensitivitas terhadap variasi skala objek, mengalami penurunan efisiensi komputasi masing-masing menjadi 14,84 FPS dan 21,18 FPS tanpa memberikan peningkatan akurasi global yang signifikan dibandingkan YOLOv8n standar pada dataset ini.

Secara praktis, sistem yang diusulkan telah membuktikan keandalannya dalam mengirimkan notifikasi deteksi beserta bukti visual secara *real-time* melalui Telegram,

memungkinkan respons keamanan yang cepat tanpa memerlukan pengawasan manual terus-menerus. Temuan ini merekomendasikan penggunaan arsitektur model yang ringan (YOLOv8n) untuk implementasi *edge computing* pada infrastruktur telekomunikasi, serta membuka peluang penelitian lanjutan terkait optimasi model melalui teknik kuantisasi (TensorRT) untuk meningkatkan performa pada perangkat keras berdaya rendah.

REFERENSI

- [1] I. T. U. (ITU), "Cyber Security Resilience Assessment Toolkit for DFS Critical Infrastructure," Geneva, Switzerland, 2023. [Online]. Available: https://www.itu.int/en/ITU-T/dfs/Documents/ITU_Cyber_Security_Resilience_Assessment_toolkit_for_DFS_Critical_Infrastructure.pdf
- [2] F. A. Bukhsh, E. Vriezokolk, H. Wiene, and R. Wieringa, "Availability Incidents in the Telecommunication Domain: A Literature Review," 2020. [Online]. Available: https://scs.ewi.utwente.nl/research/r_cybersecurity/LINC/
- [3] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision," *ArXiv*, vol. abs/2407.0, 2024, [Online]. Available: <https://api.semanticscholar.org/CorpusID:270924280>
- [4] M. Bakirci, "Real-time vehicle detection using YOLOv8-nano for intelligent transportation systems," *Trait. du Signal*, vol. 41, no. 4, pp. 1727–1740, 2024, doi: 10.18280/ts.410407.
- [5] Y. Tan, J. Song, and C. Chu, "Detection of Small Object based on Improved-YOLOv8," *Front. Comput. Intell. Syst.*, vol. 10, no. 3 SE-Articles, pp. 79–85, doi: 10.54097/n5rtnt71.
- [6] B. Khalili and A. W. Smyth, "SOD-YOLOv8—Enhancing YOLOv8 for Small Object Detection in Aerial Imagery and Traffic Scenes," *Sensors*, vol. 24, no. 19, 2024, doi: 10.3390/s24196209.
- [7] Jarot Dian Susatyono, Febryantahanuji Febryantahanuji, and Arsito Ari Kuncoro, "Pengembangan Sistem Keamanan Rumah Berbasis IoT dengan Deteksi Intrusi Real-Time Menggunakan Sensor PIR dan Kamera, serta Notifikasi Otomatis melalui Aplikasi Mobile," *J. Publ. Ilmu Komput. dan Multimed.*, vol. 4, no. 2 SE-Articles, pp. 30–41, May 2025, doi: 10.55606/jupikom.v4i2.3856.
- [8] M. A. Firdaus and N. Budhisantosa, "Rancang Bangun Sistem Pemantauan Real-Time Ruang Server dengan Integrasi Telegram Bot dan Esp32," *J. Pendidik. Tambusai*, vol. 9, no. 1 SE-Articles of Research, pp. 10320–10335, Apr. 2025, doi: 10.31004/jptam.v9i1.26271.
- [9] F. Feng, Y. Hu, W. Li, and F. Yang, "Improved YOLOv8 algorithms for small object detection in aerial imagery," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 6, 2024, doi: 10.1016/j.jksuci.2024.102113.
- [10] J. Chen, Y. Xie, Z. Qian, K. Chen, M. Zhen, and X. Hu, "Power Adapter Appearance Defect Detection Based on Task Feature Decoupling YOLOv8n," *IEEE Access*, vol. 12, no. May, pp. 60070–60080, 2024, doi: 10.1109/ACCESS.2024.3392638.
- [11] S. V. Jansi Rani, I. Ioannou, R. Swetha, R. M. Dhivya Lakshmi, and V. Vassiliou, "A novel automated approach for fish biomass estimation in turbid environments through deep learning, object detection, and regression," *Ecol. Inform.*, vol. 81, no. June, p. 102663, 2024, doi: 10.1016/j.ecoinf.2024.102663.
- [12] Y. Zhao, Y. Li, and X. Xu, "Object Detection in High-Resolution UAV Aerial Remote Sensing Images of Blueberry Canopy Fruits," *Agric.*, vol. 14, no. 10, 2024, doi: 10.3390/agriculture14101842.
- [13] Y. Saragih *et al.*, "Prototype of Radio Frequency Identification Technology Utilization for Monitoring of BTS Room Using of IoT (Internet of Things) System," *2019 Int. Conf. Comput. Sci. Inf. Technol. ICoSNIKOM 2019*, no. November, 2019, doi: 10.1109/ICoSNIKOM48755.2019.9111585.
- [14] Yuhfizar, A. Nasution, R. Putra, E. Asri, and D. Satria, "IoT: Heart Rate Monitoring Tool Using Android with Alert Messenger Telegram System," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 846, no. 1, 2020, doi: 10.1088/1757-899X/846/1/012014.
- [15] M. I. S. Siregar, S. Suwarno, and S. M. Putri, "Perancangan Peralatan Sistem Keamanan Elektronik di SHELTER BTS Secara Real Time Melalui SMS Berbasis MIKROKONTROLLER ATmega16 dan Module GSM," *J. Electr. Syst. Control Eng.*, vol. 2, no. 2, 2019, doi: 10.31289/jesce.v2i2.2357.
- [16] H. M. Chiam, Y. C. Wong, R. S. Sarban Singh, and T. J. S. Anand, "Energy Optimized YOLO: Quantized Inference for Real-Time Edge AI Object Detection," *J. Telecommun. Electron. Comput. Eng.*, vol. 17, no. 1 SE-Articles, pp. 19–28, Mar. 2025, doi: 10.54554/jtec.2025.17.01.003.
- [17] J. L. Mela and C. G. Sánchez, "Yolo-based power-efficient object detection on edge devices for USVs," *J. Real-Time Image Process.*, vol. 22, no. 3, p. 108, 2025, doi: 10.1007/s11554-025-01682-2.
- [18] R. Hakani and A. Rawat, "Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano," *Drones*, vol. 8, no. 11, p. 680, 2024. doi: 10.3390/drones8110680.
- [19] D. K. Alqahtani, M. A. Cheema, and A. N. Toosi, "Benchmarking Deep Learning Models for Object Detection on Edge Computing Devices BT - Service-Oriented Computing," W. Gaaloul, M. Sheng, Q. Yu, and S. Yangui, Eds., Singapore: Springer Nature Singapore, 2025, pp. 142–150.

-
- [20] U. Ali, M. A. Ismail, R. A. Ariyaluran Habeeb, and S. R. Ali Shah, "Performance Evaluation of YOLO Models in Plant Disease Detection," *J. Informatics Web Eng.*, vol. 3, no. 2, pp. 199–211, 2024, doi: 10.33093/jiwe.2024.3.2.15.
- [21] Q. Zhou, Z. Wang, Y. Zhong, F. Zhong, and L. Wang, "Efficient Optimized YOLOv8 Model with Extended Vision," *Sensors*, vol. 24, no. 20, pp. 1–24, 2024, doi: 10.3390/s24206506.
- [22] Y. Lan, Y. Lv, J. Xu, Y. Zhang, and Y. Zhang, "Breast mass lesion area detection method based on an improved YOLOv8 model," *Electron. Res. Arch.*, vol. 32, no. 10, pp. 5846–5867, 2024, doi: 10.3934/era.2024270.
- [23] C. Hao and H. Gu, "Semantic Web-Based Image Super-Resolution for Enhanced Steel Surface Defect Detection," *Int. J. Semant. Web Inf. Syst.*, vol. 20, no. 1, pp. 1–32, 2024, doi: 10.4018/IJSWIS.358452.
- [24] S. Liu, M. Zhu, R. Tao, and H. Ren, "Fine-Grained Feature Perception for Unmanned Aerial Vehicle Target Detection Algorithm," *Drones*, vol. 8, no. 5, 2024, doi: 10.3390/drones8050181.
-