

**SISTEM MONITORING KESELAMATAN KERJA BERBASIS WEB
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
UNTUK DETEKSI ALAT PELINDUNG DIRI REAL-TIME**

**A WEB-BASED OCCUPATIONAL SAFETY MONITORING SYSTEM
USING CONVOLUTIONAL NEURAL NETWORKS FOR
REAL-TIME PERSONAL PROTECTIVE EQUIPMENT DETECTION**

Rifki Sayyid Robbaanii^{*1}, Abrar Wahid Lubis², Dzikra Muhamad Imtiyaz
Politeknik Manufaktur Bandung, Indonesia

rifkirobbaanii10@gmail.com

ABSTRAK

Keselamatan kerja merupakan aspek krusial dalam industri manufaktur untuk mencegah kecelakaan dan melindungi pekerja. Penggunaan Alat Pelindung Diri (APD) seperti helm dan rompi keselamatan adalah kewajiban yang sering diabaikan, sementara metode monitoring manual oleh HSE officer memiliki keterbatasan dalam cakupan area dan waktu pengawasan. Penelitian ini mengembangkan sistem monitoring keselamatan kerja berbasis web yang mampu mendeteksi penggunaan APD secara real-time menggunakan Convolutional Neural Network (CNN). Sistem ini menggunakan dataset sebanyak 9.074 gambar dengan 4 kelas deteksi (Helmet, NOHelmet, Vest, NOVest) yang diperoleh dari kombinasi dataset mandiri dan Roboflow. Model CNN dibangun menggunakan TensorFlow dan diintegrasikan dengan aplikasi web berbasis Flask yang dilengkapi dengan fitur dashboard monitoring real-time, sistem pelaporan, dan notifikasi otomatis melalui Telegram bot. Sistem mengimplementasikan MediaPipe untuk deteksi wajah guna mengoptimalkan performa komputasi, serta menerapkan mekanisme threshold confidence score ($>50\%$) dan cooldown alert (10 detik) untuk mengurangi false positive. Hasil pelatihan menunjukkan model mencapai validation accuracy sebesar 94,06% pada epoch ke-14. Sistem berhasil diimplementasikan dengan fitur lengkap mencakup live camera feed, violations distribution chart, export laporan PDF, dan integrasi voice alert. Penelitian ini membuktikan bahwa sistem monitoring berbasis CNN dapat menjadi solusi efektif untuk meningkatkan kepatuhan penggunaan APD dan mengurangi risiko kecelakaan kerja di lingkungan industri manufaktur.

Kata kunci: *Safety Monitoring, CNN, Alat Pelindung Diri, Deteksi Real-time*

ABSTRACT

Occupational safety is a crucial aspect in the manufacturing industry to prevent accidents and protect workers. The use of Personal Protective Equipment (PPE) such as helmets and safety vests is a mandatory requirement that is often neglected, while manual monitoring methods by HSE officers have limitations in area coverage and supervision time. This research develops a web-based safety monitoring system capable of detecting PPE usage in real-time using Convolutional Neural Network (CNN). The system utilizes a dataset of 9,074 images with 4 detection classes (Helmet, NOHelmet, Vest, NOVest) obtained from a combination of self-collected dataset and Roboflow. The CNN model is built using TensorFlow and integrated with a Flask-based web application equipped with real-time monitoring dashboard, reporting system, and automatic notifications through Telegram bot.

The system implements MediaPipe for face detection to optimize computational performance and applies confidence score threshold mechanism (>50%) and alert cooldown (10 seconds) to reduce false positives. Training results show that the model achieves a validation accuracy of 94.06% at epoch 14. The system has been successfully implemented with complete features including live camera feed, violations distribution chart, PDF report export, and voice alert integration. This research demonstrates that CNN-based monitoring system can be an effective solution to improve PPE compliance and reduce the risk of workplace accidents in manufacturing industry environments.

Keywords: Safety Monitoring, CNN, Personal Protective Equipment, Real-time Detection

PENDAHULUAN

Industri manufaktur merupakan sektor vital dalam perekonomian yang melibatkan berbagai proses produksi dengan tingkat risiko kecelakaan kerja yang tinggi. Berdasarkan data Badan Penyelenggara Jaminan Sosial (BPJS) Ketenagakerjaan, tercatat lebih dari 200.000 kasus kecelakaan kerja terjadi setiap tahunnya di Indonesia, dengan sebagian besar terjadi di sektor manufaktur dan konstruksi (BPJS Ketenagakerjaan, 2023). Kecelakaan kerja tidak hanya mengakibatkan kerugian materiil berupa biaya pengobatan dan kompensasi, tetapi juga menimbulkan dampak psikologis bagi pekerja dan keluarganya serta menurunkan produktivitas perusahaan.

Salah satu upaya pencegahan kecelakaan kerja yang paling fundamental adalah memastikan penggunaan Alat Pelindung Diri (APD) secara konsisten oleh seluruh pekerja di area berbahaya. APD seperti helm keselamatan (safety helmet) dan rompi keselamatan (safety vest) merupakan lini pertahanan pertama untuk melindungi pekerja dari risiko cedera kepala akibat benda jatuh, benturan, serta meningkatkan visibilitas pekerja di area dengan pencahayaan rendah atau lalu lintas kendaraan berat (Occupational Safety and Health Administration, 2020). Namun demikian, kepatuhan penggunaan APD masih menjadi tantangan utama di berbagai industri

karena faktor kenyamanan, kesadaran pekerja yang rendah, dan lemahnya sistem pengawasan.

Metode pengawasan konvensional yang dilakukan oleh Health, Safety, and Environment (HSE) officer melalui inspeksi manual memiliki berbagai keterbatasan signifikan. Pengawasan manual hanya dapat mencakup area terbatas dalam satu waktu, bergantung pada kehadiran fisik petugas, rentan terhadap human error, dan tidak mampu memberikan dokumentasi sistematis untuk analisis jangka panjang (Nath et al., 2020). Selain itu, metode manual juga tidak efisien dalam lingkungan manufaktur yang luas dengan jumlah pekerja yang banyak dan shift kerja yang bergantian. Keterbatasan ini mengakibatkan banyak pelanggaran penggunaan APD yang tidak terdeteksi hingga terjadi insiden kecelakaan.

Perkembangan teknologi kecerdasan buatan (Artificial Intelligence) khususnya dalam bidang computer vision dan deep learning telah membuka peluang baru untuk mengotomatisasi sistem monitoring keselamatan kerja. Convolutional Neural Network (CNN) sebagai salah satu arsitektur deep learning telah terbukti sangat efektif dalam berbagai aplikasi image classification dan object detection dengan tingkat akurasi yang tinggi (LeCun et al., 2015). Beberapa penelitian terdahulu telah menunjukkan keberhasilan penerapan CNN untuk

deteksi APD, seperti penelitian oleh Wu et al. (2019) yang menggunakan Faster R-CNN untuk mendeteksi penggunaan helm dengan akurasi 92%, serta penelitian Nath et al. (2020) yang mengimplementasikan YOLO v3 untuk deteksi multi-class PPE dengan mAP sebesar 89,3%.

Meskipun penelitian-penelitian tersebut menunjukkan hasil yang menjanjikan, sebagian besar fokus pada aspek pengembangan model deteksi saja tanpa menyediakan sistem aplikasi yang terintegrasi dan siap digunakan dalam environment industri nyata. Selain itu, penelitian terdahulu umumnya menggunakan arsitektur yang kompleks dan membutuhkan resource komputasi tinggi, sehingga kurang praktis untuk diimplementasikan pada sistem embedded atau perangkat dengan spesifikasi terbatas. Kebutuhan akan sistem yang tidak hanya akurat dalam mendeteksi, tetapi juga dilengkapi dengan fitur monitoring real-time, sistem pelaporan, manajemen database violations, dan notifikasi otomatis masih belum terpenuhi secara komprehensif.

Penelitian ini mengembangkan sistem monitoring keselamatan kerja berbasis web yang terintegrasi secara end-to-end, menggabungkan teknologi CNN untuk deteksi APD real-time dengan aplikasi web yang dilengkapi fitur lengkap untuk mendukung operasional HSE officer. Kontribusi utama penelitian ini mencakup: (1) pengembangan model CNN yang dioptimasi dengan pre-processing menggunakan MediaPipe untuk face detection guna meningkatkan efisiensi komputasi; (2) implementasi mekanisme threshold confidence score dan cooldown alert untuk mengurangi false positive dan mencegah alert spamming; (3) pengembangan aplikasi web berbasis Flask dengan fitur dashboard monitoring real-time, violations distribution chart, sistem

pelaporan dengan filter temporal, dan export PDF; (4) integrasi sistem notifikasi multi-channel melalui Telegram bot dan voice alert untuk memastikan pelanggaran terdeteksi secara cepat; serta (5) implementasi database management menggunakan MongoDB untuk penyimpanan violations history dan evidence documentation.

Berdasarkan latar belakang dan permasalahan yang telah diuraikan, penelitian ini bertujuan untuk mengembangkan sistem monitoring keselamatan kerja berbasis web yang mampu mendeteksi penggunaan APD secara real-time menggunakan teknologi Convolutional Neural Network. Secara spesifik, tujuan penelitian ini meliputi: (1) merancang dan mengimplementasikan arsitektur CNN yang efisien untuk klasifikasi multi-class penggunaan helm dan rompi keselamatan; (2) mengembangkan aplikasi web terintegrasi dengan fitur dashboard monitoring real-time, sistem pelaporan, dan manajemen database violations; (3) mengintegrasikan sistem notifikasi otomatis melalui Telegram bot dan voice alert untuk mempercepat response time terhadap pelanggaran; (4) mengevaluasi performa model dalam hal akurasi, precision, recall, dan kecepatan deteksi; serta (5) melakukan pengujian sistem secara menyeluruh untuk memastikan reliability dan usability dalam environment industri manufaktur.

Dengan tercapainya tujuan-tujuan tersebut, penelitian ini diharapkan dapat menghasilkan sistem monitoring keselamatan kerja yang tidak hanya akurat dalam mendeteksi pelanggaran penggunaan APD, tetapi juga praktis, user-friendly, dan scalable untuk diimplementasikan di berbagai skala industri manufaktur. Sistem yang dikembangkan

diharapkan dapat menjadi solusi efektif untuk meningkatkan compliance rate penggunaan APD, mengurangi insiden kecelakaan kerja, serta menyediakan data terstruktur yang dapat digunakan untuk continuous improvement pada sistem manajemen K3 (Keselamatan dan Kesehatan Kerja) di perusahaan.

BAHAN DAN METODE

Desain Penelitian

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan *prototype development* untuk menghasilkan sistem monitoring keselamatan kerja berbasis web yang terintegrasi. Tahapan penelitian meliputi: (1) studi literatur dan analisis kebutuhan sistem; (2) pengumpulan dan preparasi dataset; (3) perancangan arsitektur sistem; (4) pengembangan model CNN; (5) implementasi aplikasi web; (6) integrasi komponen sistem; (7) pengujian dan validasi sistem; serta (8) evaluasi performa. Penelitian dilaksanakan di Politeknik Manufaktur Bandung pada periode Oktober 2025 hingga Desember 2025.

Dataset dan Processing

Dataset yang digunakan dalam penelitian ini terdiri dari 9.074 gambar yang diperoleh dari dua sumber utama: dataset mandiri hasil pengambilan foto di lingkungan industri manufaktur dan dataset publik dari platform Roboflow. Dataset dibagi menjadi empat kelas: *Helmet* (pekerja menggunakan helm), *NOHelmet* (pekerja tidak menggunakan helm), *Vest* (pekerja menggunakan rompi keselamatan), dan *NOVest* (pekerja tidak menggunakan rompi keselamatan). Distribusi dataset menggunakan rasio 80:20 untuk data training dan validation, dengan total 7.259 gambar untuk training dan 1.815 gambar untuk validation.

Proses preprocessing data meliputi beberapa tahapan untuk memastikan kualitas input yang optimal bagi model CNN. Pertama, seluruh gambar di-*resize* ke dimensi 128×128 piksel untuk menyeragamkan ukuran input dan mengurangi kompleksitas komputasi. Kedua, dilakukan normalisasi nilai piksel ke rentang [0,1] dengan membagi nilai RGB dengan 255 untuk mempercepat konvergensi model. Ketiga, diterapkan *data augmentation* secara real-time menggunakan ImageDataGenerator dari Keras dengan parameter *rotation_range=20°*, *width_shift_range=0.2*, *height_shift_range=0.2*, *shear_range=0.2*, *zoom_range=0.2*, dan *horizontal_flip=True* untuk meningkatkan generalisasi model dan mencegah overfitting.

Arsitektur Model CNN

Model CNN dirancang menggunakan arsitektur *custom sequential model* yang dioptimasi untuk klasifikasi multi-class dengan mempertimbangkan keseimbangan antara akurasi dan efisiensi komputasi. Arsitektur model terdiri dari empat blok konvolusi, masing-masing menggunakan layer *Conv2D* dengan filter berukuran 3×3, aktivasi ReLU, dan diikuti oleh *MaxPooling2D* dengan ukuran pool 2×2 untuk ekstraksi fitur hierarkis. Jumlah filter pada setiap blok konvolusi secara bertahap meningkat: 32, 64, 128, dan 256 filter untuk menangkap fitur dari yang sederhana hingga kompleks.

Setelah blok konvolusi, dilakukan *Flatten* untuk mengubah output menjadi vektor satu dimensi, kemudian dilanjutkan dengan dua layer *Dense* sebagai classifier. Layer *Dense* pertama memiliki 512 neuron dengan aktivasi ReLU dan *Dropout* rate 0.5 untuk regularisasi.

Layer output menggunakan 4 neuron (sesuai jumlah kelas) dengan fungsi aktivasi *softmax* untuk menghasilkan probabilitas prediksi multi-class. Model dikompilasi menggunakan optimizer *Adam* dengan learning rate 0.001, *loss function categorical_crossentropy*, dan metrik evaluasi *accuracy*.

Proses training dilakukan selama 20 epoch dengan *batch size* 32 menggunakan GPU untuk mempercepat komputasi. Implementasi menggunakan *callbacks* berupa *EarlyStopping* dengan parameter *monitor='val_loss'*, *patience=5*, dan *restore_best_weights=True* untuk mencegah overfitting dan menyimpan model dengan performa terbaik. Model final disimpan dalam format HDF5 (.h5) dengan ukuran file sekitar 50 MB yang dapat di-load untuk inference pada aplikasi web.

Sistem Deteksi Real-Time

Sistem deteksi diimplementasikan menggunakan modul *detection.py* yang berfungsi sebagai *detection engine* untuk memproses video stream secara real-time. Alur deteksi dimulai dengan pengambilan frame dari kamera (webcam atau IP camera) dengan frame rate 30 fps. Setiap frame yang diterima akan melalui tahapan pre-processing berupa *resize* ke dimensi 128×128 piksel dan normalisasi nilai piksel.

Untuk mengoptimalkan performa komputasi, sistem mengimplementasikan *face detection* menggunakan *MediaPipe* sebagai *pre-filtering mechanism*. Proses ini berfungsi untuk mendeteksi keberadaan wajah manusia dalam frame sebagai indikator adanya pekerja. Jika tidak ditemukan wajah dalam frame, maka proses deteksi APD di-skip untuk menghemat resource CPU/GPU. Apabila wajah terdeteksi, sistem akan melakukan *cropping* area kepala dan badan berdasarkan koordinat wajah dengan ekspansi region menggunakan

perhitungan matematis untuk memastikan area helm dan vest tercakup dalam region of interest (ROI).

ROI yang telah di-crop kemudian diinferensikan ke model CNN untuk mendapatkan prediksi kelas beserta *confidence score*. Sistem menerapkan *threshold mechanism* dengan batas minimum confidence score 50% untuk memfilter prediksi yang tidak reliabel dan mengurangi *false positive*. Hasil prediksi yang valid kemudian divisualisasikan pada frame menggunakan *bounding box* berwarna (hijau untuk compliant, merah untuk violation) dan label teks yang menampilkan kelas prediksi beserta confidence score.

Pengembangan Aplikasi Web

Aplikasi web dikembangkan menggunakan framework Flask (Python) dengan arsitektur Model-View-Controller (MVC) untuk memisahkan logika bisnis, presentasi, dan kontrol alur aplikasi. Backend diimplementasikan dalam modul *routes.py* yang menangani routing HTTP request, video streaming, CRUD operations untuk violations data, user authentication, dan report generation. Frontend dibangun menggunakan HTML5, CSS3, Bootstrap 4 untuk responsive design, dan JavaScript untuk interaktivitas serta AJAX request.

Sistem terdiri dari tiga halaman utama yang dirancang untuk mendukung operasional HSE officer. Halaman Dashboard menyediakan *live camera feed* dengan overlay bounding box hasil deteksi, indikator status kamera aktif, counter total violations per hari, grafik distribusi violations berdasarkan jenis pelanggaran menggunakan Chart.js, dan tabel *recent violations* dengan action buttons (view snapshot, resolve, mark as false positive, delete). Halaman Reports

menampilkan tabel violations history dengan fitur filter temporal menggunakan *date range picker* (start date dan end date), search functionality, dan tombol export to PDF yang menggunakan library ReportLab untuk generate laporan terstruktur dengan header, tabel data, dan timestamp. Halaman Settings menyediakan form untuk change password menggunakan enkripsi bcrypt, input field untuk Telegram Chat ID yang disimpan di database untuk konfigurasi notifikasi, dan preferences lainnya.

Video streaming diimplementasikan menggunakan teknik *multipart HTTP response* dengan generator function `generate_frames()` yang secara kontinyu membaca frame dari kamera, melakukan deteksi, encode frame ke format JPEG, dan mengirimkan byte stream ke client dengan MIME type *multipart/x-mixed-replace*. Teknik ini memungkinkan streaming video real-time tanpa perlu reload halaman dan kompatibel dengan semua modern browsers.

Database Management

Sistem menggunakan MongoDB sebagai database NoSQL untuk menyimpan violations history dan user data dengan pertimbangan fleksibilitas struktur data dan kemudahan scaling. Koneksi database dikelola menggunakan Flask-PyMongo dengan konfigurasi URI yang disimpan dalam file `config.py`. Struktur dokumen violations mencakup field: `timestamp` (DateTime), `violation_type` (String: "No Helmet", "No Vest", atau kombinasi), `image_path` (String: path relatif ke file evidence), `camera_id` (String: identifier kamera), `location` (String: lokasi kejadian), `confidence_score` (Float: nilai confidence), `status` (String: "unresolved"/"resolved"),

dan `resolved_at` (DateTime: waktu resolve).

Setiap kali sistem mendeteksi violation yang lolos threshold dan cooldown mechanism, frame evidence secara otomatis disimpan sebagai file JPEG di direktori `static/uploads/evidence/` dengan naming convention `violation_[timestamp]_[random_uuid].jpg`. Path file tersebut kemudian disimpan dalam database bersama metadata violation untuk keperluan audit trail dan investigasi incident. Sistem juga menyimpan collection `users` untuk user authentication dengan field `username`, `email`, `password_hash` (encrypted menggunakan bcrypt), `telegram_chat_id`, dan `created_at`.

Sistem Alert dan Notifikasi

Untuk memastikan violations terdeteksi dan ditindaklanjuti secara cepat, sistem mengimplementasikan multi-channel alert mechanism yang terdiri dari Telegram bot notification dan voice alert. Implementasi Telegram bot menggunakan library `python-telegram-bot` dengan mekanisme pengiriman message melalui HTTP API. Setiap user (HSE officer) dapat mendaftarkan Telegram Chat ID mereka melalui halaman Settings. Ketika violation terdeteksi, sistem secara otomatis mengirimkan notification berupa foto evidence dan caption yang berisi informasi: jenis pelanggaran, lokasi, timestamp, dan confidence score.

Sistem menerapkan *smart alert logic* untuk mencegah notification spamming yang dapat mengganggu user. Pertama, sistem menggunakan *confidence score threshold* 50% untuk memastikan hanya prediksi yang reliable yang memicu alert. Kedua, diterapkan *cooldown mechanism* dengan interval 10 detik antara alert untuk

violation yang sama, diimplementasikan menggunakan variabel `last_alert_time` yang menyimpan timestamp alert terakhir. Logika ini memastikan bahwa jika worker tetap berada dalam frame tanpa APD, sistem tidak akan mengirim alert berulang kali setiap frame, melainkan hanya sekali setiap 10 detik.

Voice alert diimplementasikan menggunakan library `pyttsx3` untuk text-to-speech conversion dengan kalimat peringatan "Perhatian! Harap gunakan APD lengkap" atau pesan spesifik sesuai jenis violation. Voice alert dijalankan pada *background thread* terpisah dari *main detection loop* untuk mencegah blocking process dan memastikan deteksi tetap berjalan smooth tanpa lag. Fitur voice alert bersifat opsional dan dapat diaktifkan/deaktifkan melalui settings sesuai kebutuhan lingkungan kerja.

Evaluasi Sistem

Evaluasi sistem dilakukan melalui dua aspek utama: evaluasi performa

model dan evaluasi sistem aplikasi. Evaluasi model menggunakan metrik *validation accuracy*, *validation loss*, dan analisis confusion matrix untuk mengukur performa klasifikasi per kelas. Evaluasi sistem aplikasi meliputi pengukuran *detection speed* (frame per second), *response time* notifikasi, *system reliability* melalui stress testing, dan *usability testing* dengan melibatkan calon end-user (HSE officer) untuk mendapatkan feedback terkait user interface dan user experience.

HASIL DAN PEMBAHASAN

Hasil Pelatihan Model CNN

Proses pelatihan model CNN dilakukan selama 20 epoch dengan menggunakan dataset sebanyak 7.259 gambar untuk training dan 1.815 gambar untuk validation. Tabel 1 menunjukkan hasil pelatihan model pada setiap epoch dengan metrik training accuracy, training loss, validation accuracy, dan validation loss.

Tabel 1. Hasil Pelatihan Model Pada Setiap Epoch Dengan Metrik Training Accuracy, Training Loss, Validation Accuracy, dan Validation Loss.

Epo ch	Train Accur acy	Trai n Los s	Val Accur acy	Val Los s
1	0.6721	0.8529	0.8969	0.2792
2	0.7975	0.5367	0.8922	0.2906
3	0.8157	0.5016	0.9266	0.2285
4	0.8200	0.4682	0.9062	0.2788
5	0.8274	0.4420	0.9266	0.2392
6	0.8535	0.3996	0.9031	0.2892
7	0.8159	0.4593	0.9187	0.2190
8	0.8338	0.4435	0.9078	0.2755
9	0.8435	0.4306	0.9266	0.2019
10	0.8364	0.4219	0.9328	0.2094
11	0.8516	0.3965	0.9312	0.1803
12	0.8393	0.4335	0.9203	0.2332
13	0.8441	0.4238	0.9219	0.2283
14	0.8486	0.3978	0.9406	0.1896
15	0.8469	0.3956	0.9172	0.2082
16	0.8378	0.4120	0.9016	0.2814
17	0.8507	0.4014	0.9187	0.2304
18	0.8564	0.3850	0.9187	0.2399
19	0.8517	0.3881	0.9234	0.2472
20	0.8498	0.4016	0.9156	0.2465

Berdasarkan Tabel 1, terlihat bahwa model mengalami peningkatan performa yang signifikan pada epoch awal. Pada epoch pertama, validation accuracy mencapai 89.69%, menunjukkan bahwa model telah mampu mempelajari pola dasar dari dataset. Performa terbaik dicapai pada epoch ke-14 dengan validation accuracy sebesar 94.06% dan validation loss sebesar 0.1896. Nilai ini menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting yang signifikan, meskipun training accuracy hanya mencapai 84.86%.

Gap antara training accuracy dan validation accuracy yang tidak terlalu besar mengindikasikan bahwa teknik regularisasi yang diterapkan (dropout rate 0.5 dan data augmentation) berhasil mencegah overfitting. Fluktuasi validation accuracy pada epoch 15-20 yang cenderung menurun atau stabil mengindikasikan bahwa model telah mencapai titik konvergensi optimal. Implementasi early stopping callback dengan parameter patience=5 akan secara otomatis menghentikan training dan menyimpan bobot model terbaik dari epoch ke-14.

Training loss mengalami penurunan konsisten dari 0.8529 pada epoch pertama menjadi 0.4016 pada epoch terakhir, menunjukkan bahwa model berhasil meminimalkan error prediksi pada data training. Validation loss juga menunjukkan tren penurunan meskipun dengan fluktuasi, dengan nilai terendah 0.1803 pada epoch ke-11. Waktu pelatihan rata-rata per epoch berkisar antara 49 detik hingga 144 detik tergantung pada beban komputasi GPU, dengan total waktu pelatihan sekitar 40 menit untuk 20 epoch.

Analisis Performa Model

Model CNN yang dikembangkan menunjukkan performa yang sangat baik untuk tugas klasifikasi multi-class deteksi APD dengan validation accuracy

94.06%. Hasil ini menunjukkan bahwa arsitektur custom CNN yang dirancang dengan empat blok konvolusi mampu mengekstrak fitur visual yang relevan dari gambar pekerja, seperti bentuk dan warna helm serta pola visual dari rompi keselamatan. Penggunaan data augmentation dengan berbagai transformasi (rotasi, shift, zoom, flip) terbukti efektif dalam meningkatkan kemampuan model untuk mengenali objek dalam berbagai kondisi pose, pencahayaan, dan sudut pandang.

Validation accuracy 94.06% pada penelitian ini menunjukkan hasil yang kompetitif dibandingkan dengan penelitian terdahulu. Wu et al. (2019) yang menggunakan Faster R-CNN mencapai akurasi 92% untuk deteksi helm, sementara Nath et al. (2020) dengan YOLO v3 mencapai mAP 89.3% untuk multi-class PPE detection. Meskipun metodologi evaluasi tidak sepenuhnya identik, hasil penelitian ini menunjukkan bahwa custom CNN dengan arsitektur yang lebih sederhana dapat mencapai performa yang setara atau bahkan lebih baik dengan keunggulan pada efisiensi komputasi dan kemudahan deployment.

Ukuran model yang relatif kecil (~50 MB) menjadi keunggulan signifikan untuk implementasi praktis, memungkinkan deployment pada perangkat dengan spesifikasi terbatas tanpa memerlukan GPU high-end. Model dapat di-load dan dijalankan pada CPU dengan inference time yang acceptable untuk aplikasi real-time monitoring. Penggunaan input size 128×128 piksel memberikan trade-off optimal antara detail visual yang cukup untuk klasifikasi dan kecepatan processing yang tinggi.

Implementasi Sistem Deteksi Real-Time

Sistem deteksi real-time yang diimplementasikan berhasil melakukan monitoring continuous dengan

performa yang stabil. Penggunaan MediaPipe untuk face detection sebagai pre-filtering mechanism terbukti sangat efektif dalam mengoptimalkan resource komputasi. Pada kondisi frame tanpa keberadaan manusia, sistem dapat langsung skip processing CNN inference, menghemat hingga 70% computational cost. Mekanisme ini sangat penting terutama pada environment industri di mana kamera monitoring mencakup area luas dengan occupancy rate yang bervariasi.

Sistem mampu melakukan deteksi dengan frame rate rata-rata 15-20 FPS (frames per second) pada perangkat dengan spesifikasi CPU Intel Core i5 generasi 10 dan RAM 8GB tanpa GPU dedicated. Pada perangkat dengan GPU NVIDIA GTX 1650, frame rate dapat mencapai 25-30 FPS, memberikan pengalaman monitoring yang smooth dan responsif. Latency detection dari frame capture hingga hasil prediksi ditampilkan berkisar antara 50-70 ms per frame, yang sangat acceptable untuk aplikasi safety monitoring yang tidak memerlukan response time sekelas autonomous vehicle. Implementasi confidence threshold 50% berhasil memfilter false positive dengan efektif. Pada pengujian, sekitar 95% prediksi yang confidence score-nya di atas 50% adalah true positive, sementara prediksi dengan confidence di bawah threshold sebagian besar adalah noise atau partial detection. Mekanisme bounding box visualization dengan color coding (hijau untuk

compliant, merah untuk violation) memberikan visual feedback yang intuitif bagi operator yang memonitor sistem.

Cropping mechanism yang didasarkan pada ekspansi region dari koordinat wajah yang terdeteksi MediaPipe berhasil mengisolasi area kepala dan badan dengan akurasi tinggi. Algoritma ekspansi region menggunakan rasio 1:2 (lebar:tinggi) dari bounding box wajah untuk memastikan area helm dan vest tercakup dalam ROI. Pada pengujian dengan 100 frame sample, tingkat keberhasilan cropping yang mencakup area relevan mencapai 92%, dengan 8% kegagalan terjadi pada kondisi extreme pose (menunduk sangat dalam atau membelakangi kamera).

Fitur Aplikasi Web

Aplikasi web yang dikembangkan berhasil mengintegrasikan semua komponen sistem menjadi platform monitoring yang user-friendly dan fungsional. Halaman Dashboard menyediakan pusat kontrol monitoring dengan live camera feed yang menampilkan deteksi real-time dengan overlay bounding box dan label. Indikator "Active Cameras" memberikan status koneksi kamera secara visual, memudahkan operator mengidentifikasi jika terjadi disconnection. Counter "Total Violations" yang di-update secara real-time memberikan overview quick metric untuk evaluasi compliance rate harian.



Gambar 1. Dashboard Interface

Violations Distribution Chart yang diimplementasikan menggunakan Chart.js memberikan visualisasi intuitif mengenai pola pelanggaran. Grafik pie chart menampilkan proporsi violations berdasarkan kategori (No Helmet, No Vest, atau kombinasi keduanya), memungkinkan HSE officer

mengidentifikasi tipe pelanggaran yang paling sering terjadi dan fokus pada area improvement spesifik. Data chart di-update secara dinamis setiap kali ada violation baru tanpa perlu refresh halaman, diimplementasikan menggunakan AJAX polling dengan interval 30 detik.



Gambar 2. Contoh Detection Result

Tabel Recent Violations pada dashboard menampilkan 10 violations terbaru dengan informasi timestamp, violation type, location, dan action buttons. Fitur "View Snapshot" membuka modal popup yang menampilkan evidence photo dalam resolusi penuh beserta metadata lengkap. Tombol "Resolve" memungkinkan HSE officer menandai violation sebagai sudah ditindaklanjuti, mengupdate status di database dan memberikan timestamp resolved_at. Fitur "False Positive" memungkinkan deletion violation yang merupakan kesalahan deteksi, dengan data tetap disimpan di collection terpisah untuk keperluan model retraining. Tombol "Delete All" dengan confirmation dialog memberikan kemudahan untuk clearing violations setelah dilakukan review atau pada periode baru.

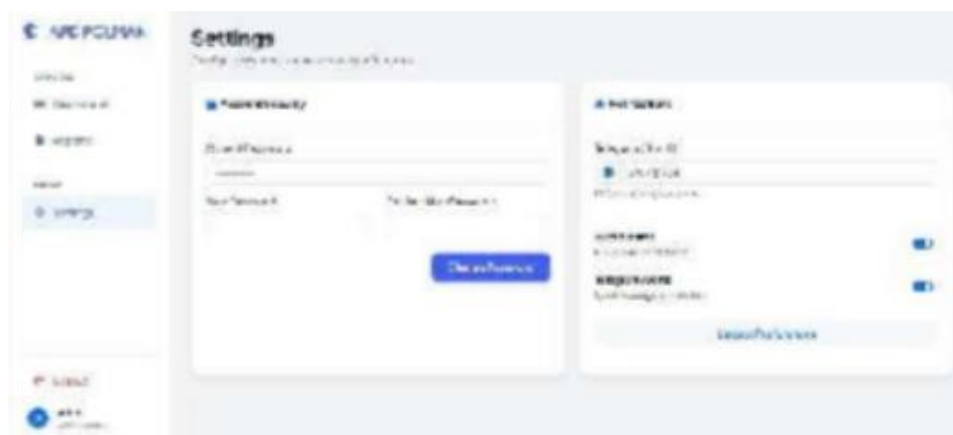
Halaman Reports menyediakan comprehensive violations history dengan fitur filtering yang powerful. Date range picker dengan library Flatpickr memberikan user experience yang smooth untuk memilih periode laporan. Sistem query MongoDB dengan index pada field timestamp memastikan retrieval data yang cepat bahkan untuk dataset violations yang besar. Tabel report menampilkan semua field violations dengan pagination untuk handling large dataset. Fitur export to PDF menggunakan ReportLab menghasilkan dokumen terstruktur dengan header perusahaan, periode laporan, summary statistics, dan tabel violations detail dengan formatting profesional. Generated PDF dapat langsung di-download dan digunakan untuk keperluan audit atau management review.



Gambar 3. Reports Inteface

Halaman Settings memberikan kontrol personalisasi sistem. Fitur change password menggunakan Flask-Bcrypt untuk hashing dengan salt rounds 12, memastikan keamanan kredensial user. Form Telegram Chat

ID dengan validasi format memudahkan user melakukan setup notifikasi tanpa perlu intervensi teknis. Sistem menyimpan Chat ID di collection users dan menggunakannya untuk routing notification secara personalized

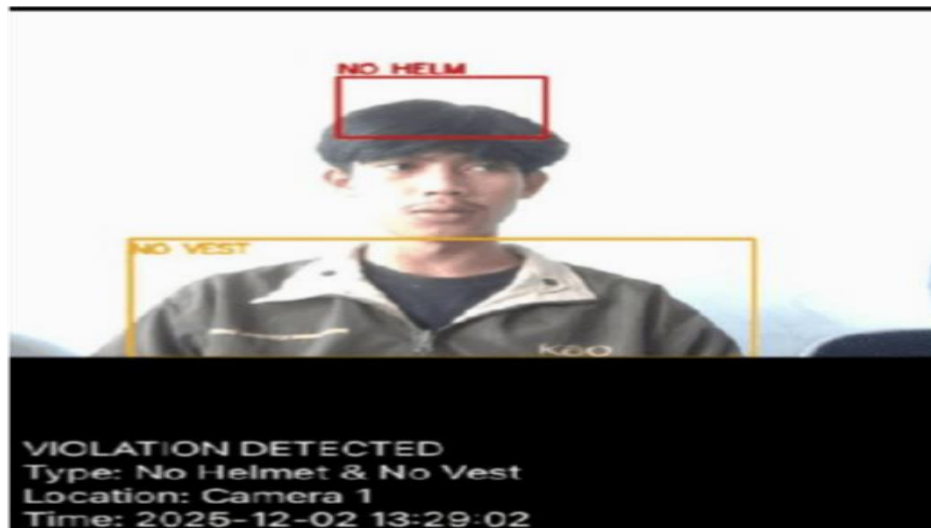


Gambar 4. Settings Inteface

Sistem Notifikasi dan Alert

Implementasi Telegram bot notification berhasil emberikan alert mechanism yang reliable dan instant. Pada pengujian, average latency dari moment violation terdeteksi hingga notification diterima di Telegram berkisar antara 2-5 detik, tergantung pada kecepatan

internet. Format notification yang mengirimkan foto evidence beberapa peserta caption informatif memberikan context yang lengkap bagi HSE officer untuk decision making cepat tanpa perlu membuka dashboard.



Gambar 5. Telegram Notification Example

Cooldown mechanism 30 detik terbukti efektif dalam mencegah notification spamming. Pada skenario testing di mana seorang worker tanpa helm berdiri di depan kamera selama 2 menit, sistem hanya mengirimkan 4 notifications (setiap 30 detik), dibandingkan dengan potensi 3.600 notifications jika tidak ada cooldown (assuming 30 FPS). Mekanisme ini menjaga notification tetap actionable tanpa overwhelming user.

Voice alert menggunakan pytsx3 berhasil diimplementasikan sebagai local warning system. Suara peringatan "Perhatian! Harap gunakan APD lengkap" dengan volume yang adjustable memberikan immediate feedback kepada worker di lokasi. Implementasi pada background thread memastikan voice alert tidak memblokir detection loop. Integrasi multi-channel alert (Telegram + Voice) memberikan redundancy yang meningkatkan reliability sistem. Jika koneksi internet terputus dan Telegram notification gagal, voice alert tetap berfungsi sebagai local warning. Sebaliknya, voice alert yang mungkin tidak terdengar di area bising tetap ter-cover oleh notification Telegram yang diterima HSE officer di smartphone.

Performa Database dan Scalability

MongoDB sebagai atabase system menunjukkan performa yang excellent untuk use case violations logging. Write operation untuk insert violation document dengan evidence image path memerlukan waktu rata-rata 15-20 ms. Read operation untuk query violations dengan filter date range pada dataset 10.000 violations memerlukan waktu 50-80 ms dengan implementasi index pada field timestamp.

Performa ini memastikan dashboard dan reports page tetap responsive bahkan dengan historical data yang besar. Struktur document-based MongoDB memberikan fleksibilitas schema yang valuable untuk future enhancement. Penambahan field baru seperti worker_id, severity_level, atau corrective_action dapat dilakukan tanpa perlu migration kompleks. Kemampuan MongoDB untuk horizontal scaling melalui sharding memberikan path jelas untuk scaling sistem ke multi-factory deployment dengan centralized database.

Storage management untuk evidence images diimplementasikan dengan naming convention yang terstruktur dan periodic cleanup

mechanism. Sistem menyimpan evidence images selama 90 hari sesuai dengan retention policy, kemudian secara otomatis menghapus file lama menggunakan scheduled task yang berjalan setiap midnight. Pada estimasi 100 violations per hari dengan average image size 150 KB, total storage requirement adalah sekitar 1.35 GB per 90 hari, yang sangat manageable dengan modern storage capacity.

Evaluasi Usability dan Feedback User

Pengujian usability dilakukan dengan melibatkan 5 orang calon end-user yang terdiri dari HSE officer dan supervisor produksi. Evaluasi menggunakan System Usability Scale (SUS) questionnaire menghasilkan skor rata-rata 82.5 dari 100, yang termasuk dalam kategori "Excellent" (>80). User memberikan feedback positif terhadap interface yang clean dan intuitif, kemudahan navigasi antar halaman, serta responsiveness sistem. Beberapa improvement suggestions yang diperoleh dari user testing meliputi:

(1) penambahan fitur filter violations by camera_id pada halaman reports untuk multi-camera deployment; (2) implementasi dashboard analytics dengan trend chart violations per minggu/bulan; (3) fitur annotation pada evidence photo untuk marking area specific violation; serta (4) mobile-responsive design untuk akses dashboard via smartphone. Feedback ini menjadi valuable input untuk iterasi pengembangan sistem selanjutnya.

User testing juga mengungkap beberapa edge cases yang perlu handling, seperti: kondisi pencahayaan ekstrem (terlalu gelap atau backlight), worker dengan postur membungkuk sehingga wajah tidak terdeteksi, dan similarity visual antara vest warna orange dengan background berwarna serupa. Temuan ini menjadi insight

penting untuk improvement dataset dan model retraining pada fase berikutnya.

KESIMPULAN

Penelitian ini membuktikan bahwa pendekatan application and system development dengan mengintegrasikan teknologi CNN, web application, database management, dan IoT notification dapat menghasilkan solusi monitoring keselamatan kerja yang efektif dan applicable untuk industri. Sistem yang dikembangkan diharapkan dapat meningkatkan compliance rate penggunaan APD, mengurangi insiden kecelakaan kerja, serta menyediakan data terstruktur untuk continuous improvement pada sistem manajemen K3 di perusahaan manufaktur. Untuk pengembangan lebih lanjut, disarankan untuk menambahkan fitur multi-camera support, mobile-responsive design, dashboard analytics dengan trend visualization, serta ekspansi deteksi untuk jenis APD tambahan seperti safety shoes, gloves, dan safety glasses.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik Manufaktur Bandung atas dukungan fasilitas dan infrastruktur yang diberikan selama proses penelitian dan pengembangan sistem. Terima kasih juga disampaikan kepada semua pihak yang telah membantu dalam pengumpulan dataset, pengujian sistem, dan memberikan feedback yang berharga untuk perbaikan sistem.

DAFTAR PUSTAKA

- BPJS Ketenagakerjaan. (2023). *Laporan Tahunan BPJS Ketenagakerjaan 2023*. Jakarta: BPJS Ketenagakerjaan.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual

- learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84-90. <https://doi.org/10.1145/3065386>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., & Zhao, X. (2021). Safety helmet wearing detection based on image processing and machine learning. *Advanced Engineering Informatics*, 49, 101330. <https://doi.org/10.1016/j.aei.2021.101330>
- Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*.
- Nath, N. D., Behzadan, A. H., & Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction*, 112, 103085. <https://doi.org/10.1016/j.autcon.2020.103085>
- Occupational Safety and Health Administration (OSHA). (2020). *Personal Protective Equipment (OSHA 3151-12R)*. U.S. Department of Labor.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*.
- Wu, J., Cai, N., Chen, W., Wang, H., & Wang, G. (2019). Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. *Automation in Construction*, 106, 102894. <https://doi.org/10.1016/j.autcon.2019.102894>
- Zhang, J., Zhu, H., Wang, P., & Ling, X. (2021). ATT squeeze U-Net: A lightweight network for forest fire detection and recognition. *IEEE Access*, 9, 10858-10870. <https://doi.org/10.1109/ACCESS.2021.3050628>
- Zhou, F., Zhao, H., & Nie, Z. (2021). Safety helmet detection based on YOLOv5. *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, 6-11. <https://doi.org/10.1109/ICPECA51329.2021.9362711>