



Deteksi Helm Keselamatan Menggunakan Jetson Nano dan YOLOv7

Hadi Supriyanto¹, Sarosa Castrena Abadi², Aliffa Shalsabilah³

^{1,2,3} Jurusan Teknik Otomasi Manufaktur dan Mekatronika Politeknik Manufaktur Bandung, Indonesia

¹hadi@ae.polman-bandung.ac.id, ²sarosa@ae.polman-bandung.ac.id, ³aliffa.shalsabilah@mhs.polman-bandung.ac.id*

Abstract

Increasing awareness of the importance of head personal protective equipment in industrial and construction environments has become a major concern in efforts to improve occupational safety. This research developed an early detection system for the use of computer vision-based head protective equipment using the YOLOv7 model and the Jetson Nano controller. The YOLOv7 algorithm was chosen for its ability for fast and accurate object detection. The YOLOv7 model was trained with a total dataset of 2799 images and iterations of 100 epochs to detect head personal protective equipment with a high degree of accuracy. The system captures imagery, activates a warning alarm, and sends a notification to Telegram when a violation occurs on an object that is not wearing a safety helmet. The test results using the confusion matrix method showed that the developed system was able to detect head personal protective equipment with an accuracy rate of 97.23%, which shows the system's ability to recognize personal protective equipment with very high accuracy. In addition, the system also showed a precision value of 98.71% indicating that all detections performed were correct, and a recall of 95.63% which describes the system's ability to recognize most of the head personal protective equipment available. The average FPS result using GPU with CUDA on Jetson Nano reached 5,723 FPS.

Keywords: PPE, Safety Helmet, YOLOv7, Jetson Nano, Telegram

Abstrak

Peningkatan kesadaran akan pentingnya alat pelindung diri kepala dalam lingkungan industri dan konstruksi telah menjadi perhatian utama dalam upaya meningkatkan keselamatan kerja. Penelitian ini mengembangkan sistem deteksi dini pemanfaatan alat pelindung diri kepala berbasis visi komputer menggunakan model YOLOv7 dan *controller Jetson Nano*. Algoritma YOLOv7 dipilih karena kemampuannya untuk deteksi objek yang cepat dan akurat. Model YOLOv7 dilatih dengan total dataset 2799 gambar dan iterasi sebanyak 100 *epoch* untuk mendeteksi alat pelindung diri kepala dengan tingkat akurasi yang tinggi. Sistem akan menangkap citra, mengaktifkan *alarm* peringatan, dan mengirimkan notifikasi pada *Telegram* ketika terjadi pelanggaran pada objek yang tidak menggunakan helm keselamatan. Hasil pengujian menggunakan metode *confusion matrix* menunjukkan bahwa sistem yang dikembangkan mampu mendeteksi alat pelindung diri kepala dengan tingkat akurasi 97,23%, yang menunjukkan kemampuan sistem dalam mengenali alat pelindung diri dengan akurasi yang sangat tinggi. Selain itu, sistem juga menunjukkan nilai *precision* sebesar 98,71% yang menunjukkan bahwa semua deteksi yang dilakukan adalah benar, dan *recall* sebesar 95,63% yang menggambarkan kemampuan sistem untuk mengenali sebagian besar alat pelindung diri kepala yang ada. Hasil rata-rata FPS menggunakan GPU dengan CUDA pada *Jetson Nano* mencapai 5,723 FPS.

Kata kunci: Alat Pelindung Diri, Helm Keselamatan, YOLOv7, Jetson Nano, Telegram

1. Pendahuluan

Kecelakaan kerja adalah situasi yang tidak dapat diprediksi atau diantisipasi, karena mereka terjadi tanpa sengaja, tanpa perencanaan, dan seringkali berdampak pada kerugian materi dan penderitaan, mulai dari tingkat yang ringan hingga yang sangat serius [1]. Data dari ILO menunjukkan bahwa lebih dari 250 juta insiden kecelakaan kerja dilaporkan tiap tahunnya. Kecelakaan kerja disebabkan oleh risiko atau potensi bahaya di tempat kerja dan perilaku pekerja itu sendiri yang berbahaya, seperti tertimpa benda berat, terluka oleh mesin produksi, atau terkena bahan kimia [2]. Tidak menggunakan alat pelindung diri dengan benar

merupakan penyebab kecelakaan kerja hingga 80% yang terjadi dari kesalahan pekerja itu sendiri [3][4]. Alat Pelindung Diri (APD) adalah suatu perangkat yang bertujuan untuk melindungi individu dengan cara memisahkan sebagian atau keseluruhan tubuhnya dari risiko potensial kecelakaan di lingkungan kerja. Kepentingan untuk melakukan pengawasan terhadap pekerja menjadi faktor krusial dalam upaya untuk mengurangi insiden kecelakaan di tempat kerja. Maka dibutuhkan sebuah deteksi *object detection* penerapan APD khususnya helm keselamatan untuk meminimalisir terjadinya kecelakaan kerja.



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

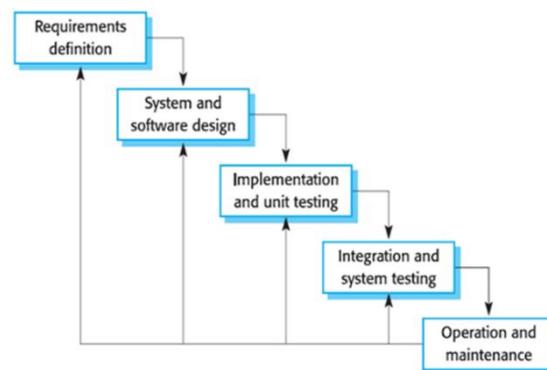
Deep learning banyak digunakan dalam penelitian *computer vision*, salah satu algoritma yang dapat digunakan untuk mendeteksi APD adalah algoritma *You Only Look Once* (YOLO). Algoritma ini merupakan sistem untuk mendeteksi objek secara *real-time* [5]. Penelitian terdahulu dengan menggunakan algoritma YOLOv2 untuk mendeteksi helm keselamatan secara *real time* dan meraih nilai validasi mAP 76.68% [8]. Menggunakan algoritma *K-Means* untuk mengelompokkan objek, lalu menggunakan *multi-scale feature extraction* untuk meningkatkan deteksi objek kecil dan mengoptimalkan model YOLOv3 menggunakan *soft-NMS*, metode ini berhasil mendeteksi helm keselamatan dengan akurasi deteksi 90,7% [9]. Penggunaan algoritma YOLOv4 berdasarkan penelitian deteksi helm keselamatan mendapatkan hasil mAP helm keselamatan hingga 92.89% [10]. Penelitian lainnya, penerapan algoritma YOLOv5 untuk mendeteksi dataset helm keselamatan menunjukkan nilai akurasi rata-rata 94% [11] [12]. Penelitian sebelumnya telah mencoba berbagai algoritma deteksi objek seperti YOLOv2, YOLOv3, YOLOv4, dan YOLOv5 untuk mendeteksi helm keselamatan. Meskipun beberapa penelitian mencapai tingkat akurasi yang cukup baik, masih ada tantangan dalam meningkatkan akurasi dan efisiensi dalam deteksi objek secara *real-time*. YOLOv7 melampaui semua detektor objek yang dikenal dalam hal kecepatan dan akurasi dalam kisaran dari 5 FPS hingga 160 FPS dan memiliki akurasi tertinggi 56,8% AP di antara semua *real-time* detektor objek yang dikenal dengan 30 FPS atau lebih tinggi pada GPU V100 [6]. *Jetson Nano* memiliki arsitektur CPU-GPU yang heterogen, di mana sistem operasi dapat di-*boot* oleh CPU dan dapat diprogram untuk mempercepat tugas *machine learning* yang rumit dari GPU yang mendukung CUDA [7].

Tujuan penelitian ini adalah meningkatkan akurasi dalam deteksi helm keselamatan dibandingkan dengan penelitian-penelitian sebelumnya. Dengan menggunakan algoritma YOLOv7, diharapkan sistem dapat mencapai akurasi deteksi yang lebih tinggi, sehingga pelanggaran keselamatan dapat diidentifikasi dengan lebih tepat. Meskipun banyak penelitian terdahulu telah mengimplementasikan berbagai algoritma deteksi objek, hanya sebagian kecil yang mempertimbangkan perangkat keras khusus untuk akselerasi tugas *machine learning*, seperti *Jetson Nano*. Penggunaan perangkat ini dapat membuka potensi yang belum sepenuhnya dieksplorasi dalam peningkatan kecepatan dan akurasi dalam pengolahan citra dan deteksi objek. Sebagian besar penelitian lebih fokus pada perangkat lunak dan algoritma, dan kurang memperhatikan bagaimana perangkat keras khusus seperti *Jetson Nano* dapat mengambil peran dalam sistem deteksi objek secara *real-time*. Penelitian ini membangun sistem pendeteksian helm keselamatan secara *real-time* dengan algoritma YOLOv7 dan *Jetson Nano* dengan CUDA NVIDIA Tegra X1. Penerapan ini

dilakukan dengan menggunakan sebuah IP kamera untuk menangkap citra pendeteksian objek terlebih dahulu, kemudian objek yang terdeteksi akan dibandingkan dengan dataset yang telah di *training*. Objek yang terdeteksi akan ditandai jika memiliki kemiripan dengan dataset yang telah di *training*. Setelah objek berhasil ditandai, maka saat objek terdeteksi melakukan pelanggaran berupa tidak menggunakan helm keselamatan akan terjadi peringatan dan juga mengirimkan foto pelanggaran dengan memanfaatkan *Telegram* [13] sebagai media notifikasi. Tentunya hal ini akan memudahkan pengurangan beban kerja dalam meningkatkan efisiensi pengendalian pelanggaran helm keselamatan bagi pekerja.

2. Metode Penelitian

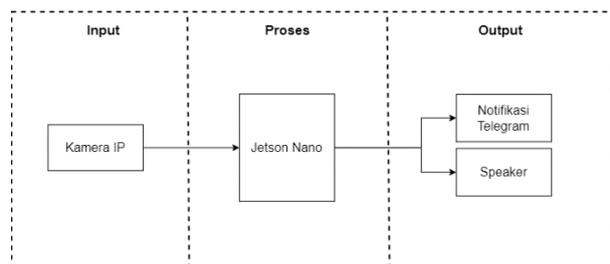
Metode pendekatan *Waterfall* [14] digunakan sebagai panduan untuk melakukan perancangan perangkat. Alur metode *Waterfall* ditampilkan pada Gambar 1.



Gambar 1. Waterfall

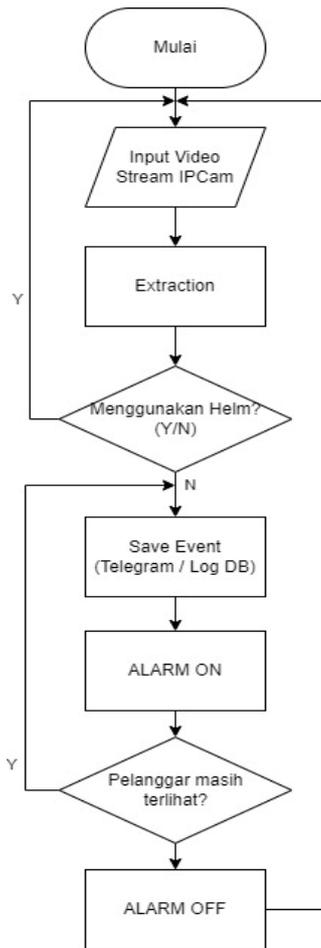
Pada Gambar 1. terdapat skema langkah kerja dari metode penelitian *Waterfall*. Metode ini secara keseluruhan terdiri dari 5 tahapan.

2.1. Perancangan Umum Sistem



Gambar 2. Blok Diagram

Pada blok diagram Gambar 2 menjelaskan secara umum perancangan sistem dalam mendeteksi APD kepala. Kamera IP berfungsi sebagai *monitoring* sistem terhadap objek deteksi. Terdapat 2 *class* objek deteksi yaitu objek yang menggunakan helm keselamatan serta yang tidak menggunakan helm keselamatan. Kemudian citra objek yang ditangkap oleh sistem akan dideteksi melalui pengolahan data *OpenCV* dan *data classification* YOLO yang terpasang pada *Jetson Nano*.



Gambar 3. Diagram Alir Sistem

Gambar 3. menjelaskan ketika terjadi pelanggaran pada objek yang tidak menggunakan helm keselamatan, sistem akan menangkap citra dan mengaktifkan alarm peringatan melalui USB *speaker*. Setiap terjadi ketidakpatuhan penggunaan helm keselamatan, sistem akan mengirimkan notifikasi pada *Telegram* sebagai peringatan dan data-data pelanggaran yang telah ditangkap oleh sistem akan tersimpan dalam *alarm history*.

2.2. Perancangan Alat



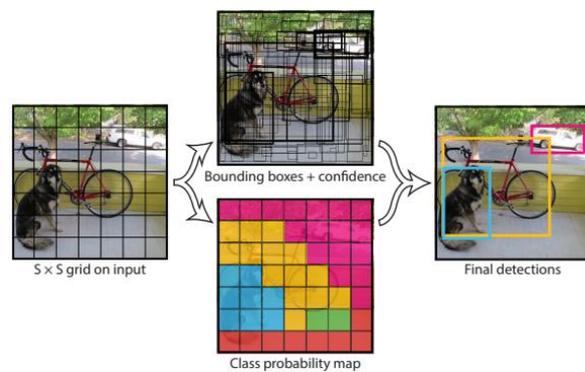
Gambar 4. Perancangan Alat

Pada Gambar 4. Menjelaskan kamera IP terhubung pada *Jetson Nano* melalui kabel *ethernet* sedangkan *USB speaker* akan terhubung dengan *Jetson Nano* melalui

kabel *USB*. Penelitian ini menggunakan *Python 3.8.9*, *OpenCV 4.5.6*, *CUDA 10.2.300*, dan *CuDNN 8.2.1.32*.

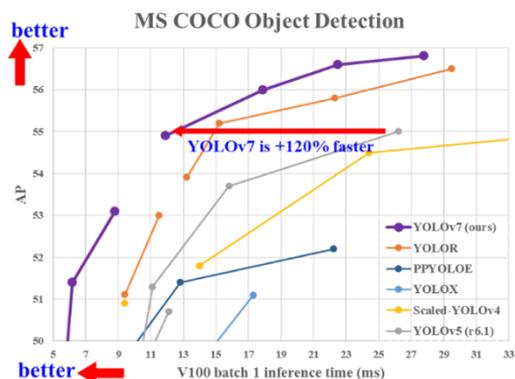
2.3. YOLO

You Only Look Once (YOLO) adalah sebuah algoritma yang dapat digunakan untuk deteksi objek. YOLO mendeteksi objek menggunakan *unified model*, dimana *single convolutional network* secara bersamaan memprediksi beberapa *bounding boxes* (kotak pembatas) dan probabilitas kelas dalam kotak tersebut. Metode ini dipraktikkan menggunakan citra yang diberikan dan kemudian segera menemukan cara terbaik untuk memprosesnya. Tidak seperti pendekatan berbasis *classifier*, YOLO dilatih pada fungsi kerugian yang berhubungan langsung dengan performa deteksi, dan seluruh model dilatih secara bersamaan [15].



Gambar 5. Diagram Algoritma YOLO

Secara umum klasifikasi merupakan proses untuk mengidentifikasi label dari data yang diuji, namun pada YOLO klasifikasinya menggunakan *localization*, dimana terdapat tambahan pemberian lokasi objek dalam bentuk *bounding box*. Terdapat 5 prediksi pada setiap *B bounding box*, yaitu x, w, y, h dan *confidence score*. Koordinat (x,y) mewakili pusat kotak yang dibatasi oleh sel *grid*. Sedangkan w (lebar) dan h (tinggi) merupakan hasil prediksi suatu objek terhadap citra keseluruhan. *Confidence score* bertugas mewakili IOU (*Intersection Over Union*).



YOLO yang mendasarinya, yang telah meningkatkan akurasi deteksi objek. Melalui eksperimen dan penelitian lebih lanjut, model ini dapat mengenali dan memetakan objek dengan tingkat ketepatan yang lebih tinggi dibandingkan dengan versi sebelumnya.

2.4. Telegram

Telegram adalah aplikasi berbasis *cloud* yang menyediakan sistem enkripsi, termasuk enkripsi *end-to-end*, pesan dengan *self-destruction*, dan infrastruktur *multidata center*. Dengan kemudahan aksesnya yang dapat diakses pada hampir semua *platform*, Telegram mempermudah administrator dalam membangun sistem notifikasi.

Telegram dipilih karena fasilitas *open Application Programming Interface (API)* yang disediakan oleh Telegram melalui *bot* dapat dimanfaatkan untuk mengirimkan pesan secara otomatis. Keberadaan *cloud* pada Telegram memungkinkan proses pengiriman yang lebih cepat, lebih ringan ketika dijalankan. dan kapasitas penyimpanan yang besar.

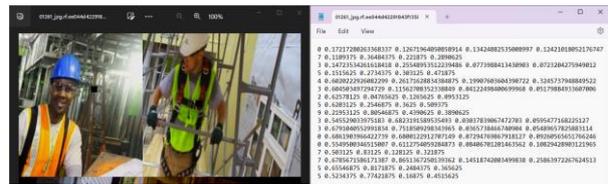
2.5. Training

Dataset yang digunakan dalam penelitian ini diambil dari *platform Roboflow*. Roboflow adalah *platform* yang memfasilitasi pengumpulan dan pengelolaan dataset dalam pemrosesan citra dan *computer vision*. Pemilihan dataset ini didasarkan pada kriteria tertentu yang relevan dengan topik penelitian ini. Dataset dapat diakses pada <https://universe.roboflow.com/roboflow-universe-projects/construction-site-safety/dataset/28>.

Dataset terdiri dari total 2801 gambar dengan 10 kelas objek yang berbeda. Setiap gambar diberi label secara manual oleh para ahli dalam bidangnya untuk memastikan keakuratan data. Dataset ini mencakup objek-objek seperti 'Hardhat', 'Mask', 'NO-Hardhat', 'NO-Mask', 'NO-Safety Vest', 'Person', 'Safety Cone', 'Safety Vest', 'machinery', 'vehicle'. Namun dalam penelitian ini kelas objek yang digunakan adalah 'Hardhat' dan 'NO-Hardhat' saja.

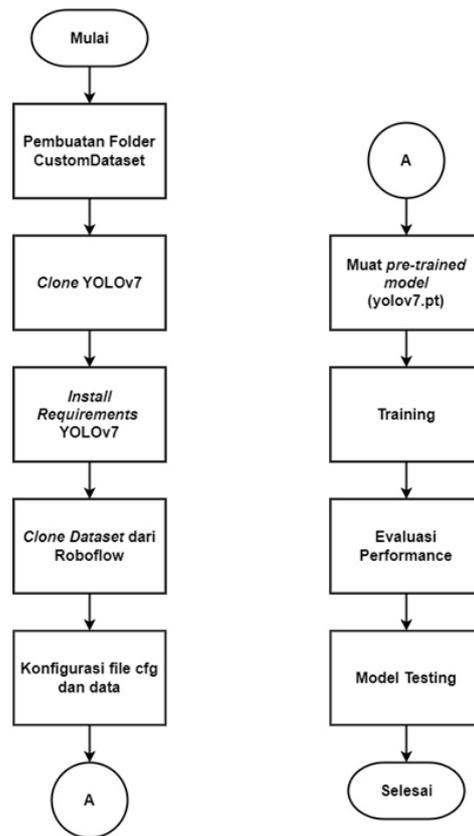
Gambar dan label terbagi menjadi 3 sub-folder yaitu folder *train*, *valid* dan *test*. Folder *train* berisi data yang digunakan selama tahap pelatihan model. Data di sini digunakan untuk mengajar model dalam mengenali pola dan fitur yang ada dalam dataset, terdapat 93% atau 2605 gambar dan label dalam folder *train*. Folder *valid* berisi data yang digunakan untuk mengevaluasi kinerja model saat tahap pengembangan, terdapat 4% atau 114 gambar dan label dalam folder *valid*. Folder *test* berisi data yang tidak pernah dilihat oleh model selama tahap pelatihan atau validasi. Data ini digunakan untuk menguji performa model secara akhir, terdapat 3% atau 82 gambar dan label dalam folder *test*. Komposisi dataset ini dipilih untuk mengakomodasi keterbatasan sumber daya, seperti kebutuhan komputasi yang lebih tinggi untuk melatih model dengan data yang lebih besar. Dalam banyak kasus, penggunaan data yang lebih kecil

untuk validasi dan pengujian dapat membantu menghemat sumber daya komputasi dan waktu yang diperlukan untuk melatih dan menguji model.



Gambar 7. File Dataset

Dalam *file .txt*, setiap baris mewakili satu gambar dalam dataset. Setiap baris berisi informasi tentang objek yang ada dalam gambar tersebut, termasuk kelas objek dan koordinat *bounding box*. Format yang digunakan dalam *file .txt* dataset *training* YOLO ini adalah format YOLO dengan urutan *<object-class>*, *<x>*, *<y>*, *<width>* *<height>*. Proses pelatihan model YOLOv7 dilakukan menggunakan *Google Colab*, yang merupakan layanan *cloud computing* gratis yang menyediakan lingkungan *Jupyter Notebook* yang kuat. *Google Colab* juga menyediakan akses ke GPU untuk mempercepat pelatihan model.



Gambar 8. Diagram Alir Proses Training YOLOv7

Gambar 8. menunjukkan diagram alir pembuatan model deteksi YOLOv7. Langkah pertama dimulai dengan membuat folder dengan nama "CustomData2". Kemudian mengunduh kode sumber YOLOv7 dan melakukan penginstalan *library Pytorch*, *CUDA*, dan *library* pendukung lainnya yang diperlukan untuk

menjalankan YOLOv7 pada *Google Colab*. Kemudian mengunduh dataset pelatihan yang akan digunakan. Dataset tersebut berisi gambar dan file anotasi yang terbagi menjadi 3 folder yaitu dataset *train*, *valid*, dan *test* sesuai dengan format yang diperlukan oleh YOLOv7. Pastikan juga untuk memodifikasi *file* konfigurasi YOLOv7 *file cfg* dan *data* sesuai dengan jumlah kelas objek yang ada dalam dataset. Setelah itu mengunduh *pre-trained model yolov7.pt* yang akan menjadi sumber acuan dalam pembuatan model baru, lalu melakukan *training* model YOLOv7 dengan menggunakan dataset *training* yang telah disiapkan. Proses *training* akan melakukan iterasi melalui batch gambar dan memperbarui parameter model sesuai dengan dataset yang digunakan. Selama proses *training* berlangsung, peneliti dapat memantau metrik evaluasi seperti *loss*, akurasi, atau *mAP (mean Average Precision)* untuk mengukur kinerja model. Evaluasi ini dapat dilakukan pada dataset validasi atau pengujian. Setelah *training* selesai, model disimpan dalam format yang sesuai (.pt). Model akan melalui proses *testing*, jika nilai yang didapat kurang memuaskan maka model akan melakukan proses *training* kembali hingga model sesuai dengan parameter yang telah ditetapkan.

Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
82/99	9.31G	0.024	0.02302	0.002774	0.04979	431	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.939	0.821	0.867	0.559		
Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
83/99	9.31G	0.02355	0.02255	0.002634	0.04874	299	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.92	0.828	0.869	0.551		
Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
84/99	9.31G	0.02351	0.02182	0.002685	0.04802	253	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.913	0.842	0.869	0.538		
Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
85/99	9.31G	0.02341	0.02201	0.002658	0.04808	345	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.928	0.83	0.865	0.545		
Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
86/99	9.31G	0.0234	0.02202	0.002597	0.04802	317	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.918	0.843	0.867	0.551		
Epoch	gpu_mem	box	obj	cls	total	labels	img_size		
87/99	9.31G	0.02305	0.02152	0.002473	0.04705	350	416:	100%	82/82
Class	Images			P	R	mAP@.5	mAP@.5:.95:		
all	114		697	0.915	0.837	0.867	0.553		

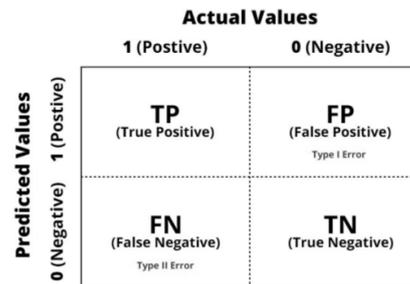
Gambar 9 Proses *Training* Model

2.6. Perancangan Pengujian Sistem

Dalam mengukur akurasi dan nilai kesalahan dari kinerja sistem klasifikasi, terdapat beberapa metode yang dapat digunakan, antara lain *cross validation* dan *confusion matrix*. Metode-metode ini berguna dalam mengevaluasi dan memahami sejauh mana sistem klasifikasi berkinerja secara akurat dan seberapa sering sistem melakukan kesalahan dalam mengklasifikasikan data. *Cross validation* merupakan suatu teknik dalam pengolahan data statistik yang digunakan untuk mengevaluasi kinerja model atau algoritma. Dalam metode ini, data dibagi menjadi dua subset data *training* [16].

Untuk mendapatkan nilai evaluasi kinerja model, dilakukan penggunaan metode *confusion matrix* pada setiap data iterasi *Cross validation*. Dalam penelitian ini,

model terdiri dari dua kelas data (*positive* dan *negative*), maka hanya terdapat 4 kemungkinan output dari sistem yang merepresentasikan elemen pada *confusion matrix* yaitu, *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*.



Gambar 10. *Confusion Matrix*

Berdasarkan dari klasifikasi *confusion matrix* maka dapat diukur *performance metrics* kinerja suatu sistem yang telah dibuat seperti *accuracy*, *precision*, *recall*. *Accuracy* adalah pengukuran seberapa dekat kebenaran hasil yang diperoleh dengan hasil yang seharusnya diperoleh. *Precision* adalah tingkat ketepatan informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Untuk mencari nilai dari *performance metrics* dalam persen maka dapat dihitung menggunakan persamaan berikut [17]:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3)$$

Nilai *Accuracy* yang dimaksud adalah nilai seberapa dekat hasil prediksi dengan kebenaran pada objek yang memakai helm keselamatan. Nilai *Precision* yang dimaksud adalah tingkat ketepatan hasil prediksi sistem terhadap objek yang memakai helm keselamatan. Nilai *Recall* yang dimaksud adalah keberhasilan sistem dalam mendeteksi kembali objek yang memakai helm keselamatan.

2.7. Perancangan *User Interface* dan Notifikasi

Perancangan *User Interface* pada sistem dapat diilustrasikan pada Gambar 12 dan 13. Perancangan user Interface dibuat dengan sedeharna yaitu halaman "*Home Page*" yang berisi total objek, total objek tanpa helm, total pelanggaran hari ini, dan tombol *browse folder* foto pelanggaran. Kemudian pada halaman "*Table view*" berisi informasi setiap pelanggaran terjadi.

Pengiriman notifikasi pada *Telegram* dipilih karena *Telegram* merupakan aplikasi *cloud based* dan sistem enkripsi yang menyediakan enkripsi *end-to-end*, *self-destruction messages*, dan infrastruktur *multidata center*. Kemudahan akses yang diberikan *Telegram* yang dapat berjalan di hampir semua *platform* memberikan

kemudahan bagi administrator untuk membangun sistem notifikasi dengan memanfaatkan fasilitas *open Application Programming Interface (API)* yang disediakan oleh *Telegram* melalui bot yang dapat digunakan untuk mengirimkan pesan secara otomatis.

3. Hasil dan Pembahasan

3.1. Training

Penelitian ini menghasilkan sebuah model baru untuk mendeteksi Alat Pelindung Diri pada pekerja menggunakan algoritma YOLOv7 secara *real-time* dengan iterasi sebanyak 100 *epochs* dan *batch size* 32. Tabel 1. menunjukkan bahwa model akhir peringkat validasi mAP sebesar 87,1%, skor tersebut menunjukkan bahwa akurasi deteksi objek model YOLOv2 pada helm keselamatan sudah maksimal, juga mendapat nilai *Precision* sebesar 91,4% dan *Recall* 83,7%.

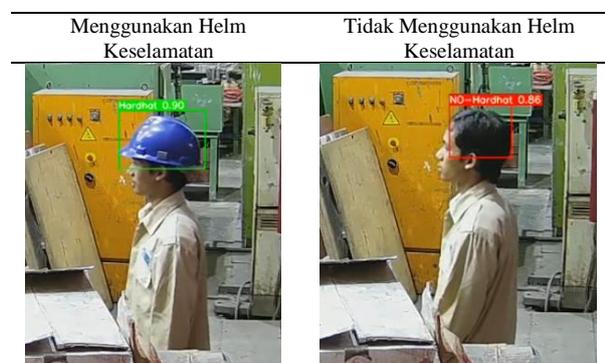
Tabel 1. YOLOv7 Training Result

Epochs	Batch Size	mAP	Precision	Recall
100	32	87,1%	91,4%	83,7%

3.2. Klasifikasi Objek Menggunakan Helm Keselamatan atau Tidak

Pengujian ini mengidentifikasi penggunaan helm keselamatan dan tidak pada objek. Hasilnya menunjukkan bahwa objek menggunakan helm keselamatan dengan benar, sementara objek lainnya tidak menggunakan helm keselamatan sama sekali.

Tabel 2. Pengujian Klasifikasi Objek



Pada Tabel 2. terlihat bahwa sistem yang telah dikembangkan mampu membedakan objek yang menggunakan helm keselamatan dan yang tidak. Jika objek menggunakan helm keselamatan, maka akan ditampilkan kotak berwarna hijau, sedangkan jika tidak menggunakan helm keselamatan, maka akan muncul kotak berwarna merah.

3.3. Pengujian Confusion Matrix

Pengujian ini menggunakan metode *Confusion Matrix* untuk mengevaluasi kinerja sistem deteksi. Data yang berisi sejumlah contoh gambar dengan *label bounding box* diuji pada sistem, dan hasil deteksi kemudian dibandingkan dengan *label* yang sebenarnya.

Tabel 3. Pengujian *Confusion Matrix*

Gambar	TP	FP	FN	TN
1	13	0	0	0
2	12	0	1	0
3	12	0	0	1
4	11	0	0	2
5	11	0	0	2
6	10	0	0	3
7	9	0	0	4
8	8	0	1	4
9	8	0	0	5
10	8	0	0	5
11	6	0	1	6
12	6	1	1	5
13	6	0	0	7
14	6	0	0	7
15	4	0	1	8
16	3	0	2	8
17	4	0	0	9
18	4	0	0	9
19	3	0	0	10
20	3	0	0	10
21	2	0	0	11
22	2	0	0	11
23	1	0	0	12
24	1	0	0	12
25	0	1	0	12
Total	153	2	7	163

		Actual Class	
		Hardhat	No-Hardhat
Predicted Class	Hardhat	TP 153	FP 2
	No-Hardhat	FN 7	TN 163

Gambar 11. Performances Metrics

$$Accuracy = \frac{153 + 162}{153 + 2 + 7 + 163} \times 100\% = 97,23\%$$

$$Precision = \frac{153}{153 + 2} \times 100\% = 98,71\%$$

$$Recall = \frac{153}{153 + 7} \times 100\% = 95,63\%$$

Hasil pengujian menggunakan *Confusion Matrix* menunjukkan bahwa sistem deteksi ini yang dirancang dengan YOLOv7 memiliki performa yang baik. Sistem mampu mendeteksi alat pelindung diri dengan tingkat *accuracy* 97,23%, *precision* 98,71%, *recall* 95,63%.

3.4. Pengujian FPS Jetson Nano

Tabel 4. Pengujian FPS *Jetson Nano*

Pengujian ke-	FPS
1	5,7
2	5,64
3	5,66
4	5,71
5	5,6
6	5,84
7	5,77
8	5,89
9	5,73
10	5,69
Rata-rata FPS	5,723

Hasil pengujian menunjukkan bahwa penggunaan GPU CUDA pada Jetson Nano mendapatkan rata-rata 5,723 FPS.

3.5. Implementasi *Interface* Pada Aplikasi *Desktop* dan *Telegram*

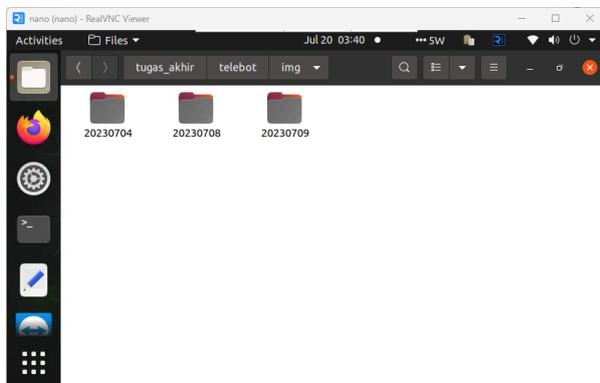
Pengujian ini berhasil mengintegrasikan sistem deteksi dengan *interface* yang dibuat dan berfungsi dengan baik. Pada pengujian ini, sistem deteksi akan mengirimkan data hasil deteksi, dan menampilkan informasi yang relevan pada *interface* yang telah dibuat.



Gambar 12. Tampilan *Interface* Home Page



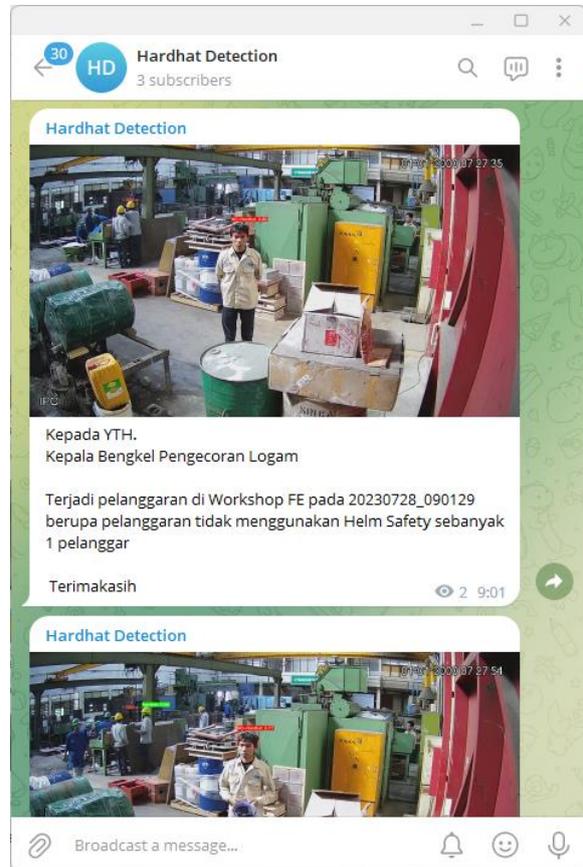
Gambar 13. Tampilan *Interface* Tabel Pelanggaran



Gambar 14. Folder Pelanggaran per Hari

Ketika tombol *browse folder* foto pelanggaran pada Gambar 12. Sistem akan membuka *file folder*

penyimpanan foto pelanggaran secara otomatis. Gambar 14. Menjelaskan *folder* yang dibuat telah diatur melalui pemrograman untuk tujuan penyimpanan pelanggaran per hari. Dengan demikian, struktur *folder* ini memastikan bahwa foto-foto pelanggaran tidak akan tersebar atau tercampur, melainkan akan diorganisir berdasarkan harinya masing-masing.



Gambar 15. Pengiriman Notifikasi Pada *Telegram*

Pengujian pada gambar 15. Ini telah berhasil mengembangkan sistem yang dapat mengirimkan foto yang ditangkap dalam format .png beserta deskripsi pelanggaran terkait penggunaan helm keselamatan melalui layanan pesan *Telegram*. Setiap kali terjadi pelanggaran, sistem secara otomatis mengirimkan informasi yang relevan melalui *Telegram*.

4. Kesimpulan

Kesimpulan dari penelitian ini adalah hasil pengujian performa menunjukkan bahwa deteksi dini yang dirancang dengan YOLOv7 memiliki performa yang baik, dengan tingkat *accuracy* mencapai 97,23%, *precision* sebesar 98,71%, dan *recall* sebesar 95,63%. Penggunaan *controller Jetson Nano* juga memberikan hasil rata-rata FPS mencapai 5,723 FPS saat menggunakan GPU dengan CUDA. Hal ini menandakan bahwa sistem deteksi dini alat pelindung diri kepala masih kompatibel dan dapat berjalan menggunakan *controller Jetson Nano*.

Meskipun performa sistem saat ini cukup baik, masih ada ruang untuk meningkatkan kinerja. Melakukan optimasi lebih lanjut pada algoritma deteksi objek dan pengaturan perangkat keras dapat meningkatkan akurasi dan kecepatan deteksi. Dapat mengembangkan fitur tambahan, seperti pengenalan objek lain yang relevan dengan keselamatan di tempat kerja (seperti sepatu pelindung, sarung tangan, dll.), atau integrasi dengan sistem notifikasi keamanan yang lebih luas.

Daftar Rujukan

- [1] D. Nuraeni and P. Hargiyarto, "Pemahaman Penggunaan Alat Pelindung Diri (APD) dan Sikap Keselamatan dan Kesehatan Kerja (K3) Terhadap Perilaku K3 di Bengkel Bubut," *J. Pendidik. Vokasional Tek. Mesin*, vol. 7, no. 3, pp. 195–202, 2019.
- [2] M. E. Laily, F. Nur, and G. Q. O. Pratamasunu, "Deteksi Penggunaan Alat Pelindung Diri (APD) Untuk Keselamatan dan Kesehatan Kerja Menggunakan Metode Mask Region Convolutional Neural Network (Mask R-CNN)," *J. Komput. Terap.*, vol. 8, no. 2, 2022, [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jkt/>
- [3] F. Edigan, L. R. Purnama Sari, and R. Amalia, "Hubungan Antara Perilaku Keselamatan Kerja Terhadap Penggunaan Alat Pelindung Diri (APD) Pada Karyawan PT Surya Agrolika Reksa Di Sei. Basau," *J. Sainis*, vol. 19, no. 02, p. 61, 2019, doi: 10.25299/sainis.2019.vol19(02).3741.
- [4] A. Septianto and A. R. Wardhani, "Penerapan Analisis Resiko Terhadap Kesehatan Dan Keselamatan Kerja(K3) Pada Pt. X," *J. Apl. Dan Inov. Ipteks "Soliditas"*, vol. 3, no. 1, p. 7, 2020, doi: 10.31328/js.v3i1.1385.
- [5] R. Ac, "Real Time Object Detection System with YOLO and CNN Models: A Review," no. July, 2022, doi: 10.37896/JXAT14.07/315415.
- [6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," pp. 1–17, 2022, [Online]. Available: <http://arxiv.org/abs/2207.02696>
- [7] T. Peng, D. Zhang, R. Liu, V. K. Asari, and J. S. Loomis, "Evaluating the Power Efficiency of Visual SLAM on Embedded GPU Systems," *Proc. IEEE Natl. Aerosp. Electron. Conf. NAECON*, vol. 2019-July, pp. 117–121, 2019, doi: 10.1109/NAECON46414.2019.9058059.
- [8] M. Hatami, F. Nurapriani, W. Andriani, U. Buana, P. Karawang, and S. B. Saleh, "Deteksi Helmet Dan Vest Keselamatan Secara Realtime Menggunakan Metode Yolo Berbasis Web FLASK," vol. 10, no. 1, pp. 221–233, 2023.
- [9] P. Wen, M. Tong, Z. D. B, and Q. Qin, *Method Based on YOLOv3*, vol. 1, no. 2019. Springer International Publishing, 2020. doi: 10.1007/978-3-030-57884-8.
- [10] D. Benyang, L. Xiaochun, and Y. Miao, "Safety helmet detection method based on YOLO v4," pp. 155–158, 2020, doi: 10.1109/CIS52066.2020.00041.
- [11] Z. P. Xu, Y. Zhang, J. Cheng, and G. Ge, "Safety Helmet Wearing Detection Based on YOLOv5 of Attention Mechanism," 2022, doi: 10.1088/1742-6596/2213/1/012038.
- [12] F. Zhou, H. Zhao, and Z. Nie, "Safety Helmet Detection Based on YOLOv5," in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, Jan. 2021, pp. 6–11. doi: 10.1109/ICPECA51329.2021.9362711.
- [13] F. Fitriansyah and Aryadillah, "Penggunaan Telegram Sebagai Media Komunikasi Dalam Pembelajaran Online," *J. Hum. Bina Sarana Inform.*, vol. 20, no. Cakrawala-Jurnal Humaniora, p. 113, 2020, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/cakrawala>
- [14] C. Trisianto, "Penggunaan Metode Waterfall Untuk Pengembangan Sistem Monitoring Dan Evaluasi Pembangunan Pedesaan," *J. Teknol. Inf. ESIT*, vol. XII, no. 01, pp. 7–21, 2018, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/ESIT/article/view/18174/9335>
- [15] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, "Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA Dan Modified YOLO," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, pp. 413–419, 2019, doi: 10.25126/jtiik.201961275.
- [16] A. Pamuji, "Prediksi Otorisasi Pengguna Sistem Berkas pada Algoritma Klasifikasi dengan Teknik Naïve Bayes," *Infomatek*, vol. 24, no. 1, pp. 35–44, 2022, doi: 10.23969/infomatek.v24i1.4604.
- [17] Karsito and S. Susanti, "Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia," *J. Teknol. Pelita Bangsa*, vol. 9, pp. 43–48, 2019.