

# Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

Journal Homepage: <http://journal.lembagakita.org/index.php/jtik>



## Rancang Bangun Aplikasi Deteksi Alat Pelindung Diri (APD) untuk Pekerja Proyek dengan Menggunakan Algoritma Yolov5

Muhamad Alfin Taufiqurrochman <sup>1\*</sup>, Herny Februariyanti <sup>2</sup>

<sup>1,2</sup> Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank, Kota Semarang, Provinsi Jawa Tengah, Indonesia.

### article info

#### Article history:

Received 20 December 2023

Received in revised form

25 February 2024

Accepted 25 March 2024

Available online April 2024.

#### DOI:

<https://doi.org/10.35870/jti.k.v8i2.1960>.

#### Keywords:

Personal Protective Equipment; Artificial Intelligence; Computer Vision; Yolov5.

### abstract

The risk of work accidents that can be experienced by project workers is very high in the world of construction. This can be caused by behavioral factors, one of which is project workers' indiscipline in wearing Personal Protective Equipment (PPE), which can endanger the personal safety of project workers. By utilizing Artificial Intelligence technology, with the computer vision domain, researchers created a PPE detection application using the YoloV5 algorithm. The stage of creating this detection application starts from the process of problem scoping, data acquisition, data exploration, modeling, evaluation and deployment. The dataset used in making this application was taken from Saravana Alagar via Google Drive, covering 4 PPE objects, namely helmets, masks, vests and shoes. By conducting a training dataset of 100 epochs, the percentage results given were very good, namely helmets 96%, vests 96%, masks 95%, and shoes 92%. It is hoped that making this application can minimize cases of work accidents that occur in the project worker area and can increase discipline in using project PPE.

### abstract

Resiko kecelakaan kerja yang dapat dialami oleh pekerja proyek sangatlah tinggi dalam dunia konstruksi. Hal ini dapat disebabkan oleh faktor perilaku, salah satunya adalah ketidakdisiplinan para pekerja proyek dalam memakai Alat Pelindung Diri (APD), yang dimana hal tersebut dapat membahayakan keselamatan diri pekerja proyek. Dengan memanfaatkan teknologi Artificial Intelligence, berbasis domain computer vision, peneliti membuat aplikasi deteksi APD menggunakan algoritma YoloV5. Tahap pembuatan aplikasi deteksi ini dimulai dari proses problem scoping, data acquisition, data exploration, modeling, evaluation, dan deployment. Dataset yang dipakai dalam pembuatan aplikasi ini diambil dari saravana alagar melalui google drive, meliputi 4 objek APD yaitu helm, masker, rompi dan sepatu. Dengan melakukan training dataset sebanyak 100 epoch, hasil persentase yang diberikan sangat bagus, yaitu helm 96%, rompi 96%, masker 95%, dan Sepatu 92%. Diharapkan dari pembuatan aplikasi ini dapat meminimalisir kasus kecelakaan kerja yang terjadi pada area pekerja proyek dan dapat meningkatkan kedisiplinan dalam memakai APD proyek.

\*Corresponding Author. Email: muhamadalfintaufiqurrochman@mhs.unisbank.ac.id <sup>1\*</sup>.

© E-ISSN: 2580-1643.

Copyright @ 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan RISET). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

## 1. Latar Belakang

Faktor perilaku merupakan salah satu permasalahan dalam kepatuhan menggunakan APD (Alat Pelindung Diri) yang dapat mencegah terjadinya kecelakaan kerja. Teori Lawrence Green menjelaskan bahwa perilaku terdiri atas faktor predisposisi, pendukung, dan pendorong [1]. Keselamatan dan kesehatan kerja merupakan suatu pemikiran dan upaya untuk menjamin keutuhan dan kesempurnaan baik jasmani maupun rohani. Dengan keselamatan dan kesehatan kerja maka para pihak pekerja diharapkan dapat melakukan pekerjaan dengan aman dan nyaman [2]. Dari hasil penelitian yang dilakukan oleh Jayanti dkk, hampir seluruh (92.07%) pekerja menggunakan Alat Pelindung Diri secara tidak lengkap dan sebagian kecil (7,93%) pekerja menggunakan Alat Pelindung Diri secara lengkap, dan hampir (90.48%) pekerja tidak sering mengalami kecelakaan kerja, sedangkan sebagian kecil (9,52%) pekerja sering mengalami kecelakaan kerja [3]. Dari persentase tersebut kasus kecelakaan kerja dapat dikatakan cukup tinggi.

Dalam sektor konstruksi, risiko kecelakaan kerja sangat tinggi. Berdasarkan data dari BPJS Ketenagakerjaan, jumlah kasus kecelakaan kerja di Indonesia telah mencapai 234.270 pada tahun 2021, dan angka ini terus meningkat hingga saat ini. Risiko keselamatan di proyek antara lain jatuh dari ketinggian, terjebak di bangunan yang roboh, terbentur kendaraan proyek/alat berat, tersengat aliran listrik, tertimpa benda jatuh, terkena api, dan keracunan [4]. Dalam penelitian Moch. Djaohar dkk, Dimana melakukan Rancang Bangun Pengecekan Alat Pelindung Diri dengan menggunakan Algoritma YOLO. Berdasarkan hasil percobaan pengujian helm safety dari jarak 0–5 meter dapat terdeteksi 100% berhasil. Masker mulai dari 0-5M dapat di deteksi dengan akurasi 97,95%. Kacamata pengaman dari 0-5M berhasil dideteksi 38,77%. Sarung tangan pengaman dari 0-5M dapat dideteksi 100%. Rompi keselamatan dari 0-5M dapat dideteksi 97,95%. Earmuff dari 0-5M berhasil dideteksi 97,95%. Sepatu safety dari 0-5M berhasil dideteksi 100% [5]. Pada penelitian yang dilakukan oleh Susi Susanti, dalam Deteksi Helm Otomatis Untuk Keselamatan Kerja di Tempat Proyek Berbasis Yolo. Hasil implementasi yang dilakukan bertujuan untuk mendeteksi helm menggunakan metode TinyYOLOv3, Dari hasil

pengujian pada total 50 dataset terdapat nilai akurasi tertinggi yaitu 93% pada kondisi skenario 3 dengan waktu proses 9.44 detik [6]. Rescky Marthen Mailoa dan Leo Willyanto Santoso juga melakukan penelitian yaitu Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN. Metode YOLO digunakan untuk mendeteksi bagian tubuh kepala dan badan dari gambar yang di-inputkan. Bagian tubuh yang terdeteksi kemudian dipotong dan diproses dengan metode Convolutional Neural Network (CNN) dengan model ResNet50 untuk diklasifikasikan. Hasil penelitian menunjukkan model YOLO memiliki Tingkat kecepatan mendeteksi yang sangat tinggi dengan akurasi yang baik. Sementara itu model CNN yang dimodifikasi menunjukkan performa yang baik dengan nilai akurasi rata-rata 96% [7]. Agustin Nurfirmansyah dan Rohman Dijaya, dalam penelitian nya membuat Deteksi Kelalaian Alat Pelindung Diri (APD) Pada Pekerja Kontruksi Bangunan dengan menggunakan model deep learning yaitu YOLOv4, Penelitian ini bertujuan untuk pengenalan objek pada citra APD pekerja bangunan menggunakan YOLOv4 Jumlah data citra yang digunakan dalam pelatihan yaitu 500 citra yang terdiri dari 5 jenis APD dan 10 calass. Nila avg loss pada model akhir yang dibangun dengan YOLOv4 yaitu 1,305% dan nilai validasi mAP 99,99%. Berdasarkan hasil pengujian yang dilakukan secara real time Webcam atau CCTV, objek pada citra APD berhasil dikenali dengan akurasi rata-rata 80% [8].

Berdasarkan referensi dari hasil penelitian terdahulu, telah diketahui bahwa banyak terjadi kasus kecelakaan kerja di Indonesia. Tidak sedikit dari mereka mengalami kecacatan, penyakit yang bisa terjadi saat kerja dan bahkan sampai berujung pada kematian. Oleh karena itu, maka perlu adanya usaha dalam melindungi serta mencegah pekerja dari segala bahaya dan risiko yang dapat dialami oleh pekerja konstruksi. Solusi dari masalah yang penulis angkat adalah dengan menerapkan Artificial Intelligence (AI) dalam pembuatan aplikasi deteksi APD untuk pekerja Proyek. Penulis membuat AI dengan domain computer vision dan menerapkan algoritma Yolo pada sistem deteksi. You Only Look Once (YOLO) adalah sebuah algoritma dalam bidang visi komputer yang digunakan untuk mendeteksi dan mengidentifikasi objek dalam gambar atau video secara cepat dan efisien.

Pemilihan algoritma YOLO sebagai metode dalam pembuatan aplikasi pada penelitian ini adalah karena YOLO memiliki kelebihan dimana dapat mengklasifikasikan objek secara langsung dalam satu langkah inferensi (feedforward) melalui jaringan saraf. Hal ini membuatnya lebih cepat daripada beberapa pendekatan deteksi objek lainnya yang memerlukan beberapa langkah atau perhitungan yang lebih rumit. YOLO juga mampu melakukan deteksi objek secara real-time, memiliki tingkat keakuratan yang baik serta memiliki kemampuan mendeteksi berbagai objek dalam satu gambar dengan baik atau biasa disebut dengan multiobjek. Dalam implementasinya, Algoritma YOLO mudah digunakan dan terbilang sederhana untuk pengembangan. Kelebihan-kelebihan dari YOLO ini juga terlihat dalam hasil penelitian terdahulu. Pada penelitian ini, peneliti memilih menggunakan YOLOv5 atau YOLO version 5. YOLOv5 ini adalah versi terbaru dari serangkaian model deteksi objek YOLO yang dikembangkan oleh Ultralytics. YOLOv5 adalah iterasi terbaru dari model-model YOLO dengan perbaikan dan peningkatan performa dari versi-versi sebelumnya. Adapun tujuan dari penelitian ini adalah sebagai upaya dalam meminimalisir kecelakaan kerja yang terjadi di area proyek konstruksi, serta memastikan bahwa pekerja dalam kondisi sesuai standar keselamatan kerja sebelum memasuki area kerja.

## 2. Metode Penelitian

Adapun metodologi penelitian dalam proses mengumpulkan data-data yang dibutuhkan pada penelitian ini adalah sebagai berikut:

### 1) Objek Penelitian

Objek yang digunakan dalam penelitian ini adalah beberapa alat pelindung diri untuk pekerja proyek, dapat meliputi helm, masker, rompi dan sepatu.

### 2) Metode Pengumpulan Data

Dalam penelitian ini, ada beberapa metode untuk memperoleh data yang digunakan peneliti dalam pembuatan aplikasi menggunakan AI untuk mendeteksi kelengkapan APD antara lain:

#### a) Observasi

Metode pengumpulan data dengan melakukan observasi ini dapat dilakukan dengan mengamati pekerjaan di lokasi proyek

untuk memahami kondisi kerja dan situasi di lapangan. Hal ini dapat membantu dalam merinci persyaratan APD yang sesuai dengan situasi nyata.

#### b) Studi Literatur

Pada metode pengumpulan data dengan studi literatur ini dilakukan dengan penelitian literatur tentang topik penelitian yang akan diteliti. Ini melibatkan membaca jurnal, artikel, buku, dan sumber literatur lainnya yang relevan untuk memahami dasar teori AI, jenis APD yang ada, dan tantangan yang dihadapi pekerja proyek. Informasi dari literatur yang telah dipelajari akan menjadi dasar penelitian.

### 3) Metode Pengembangan Sistem

Metode pengembangan sistem adalah langkah-langkah yang akan dilakukan dalam mengembangkan aplikasi mendeteksi APD untuk pekerja proyek yang menggunakan Artificial Intelligence. Tahap-tahap ini mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem, dan hasil yang diharapkan. Langkah – Langkah dalam pengembangan sistem pada penelitian ini sebagai berikut :

#### a) Analisis Kebutuhan

Dalam tahap ini, peneliti akan mengidentifikasi kebutuhan utama dari para pekerja proyek terkait APD, menganalisis ketersediaan teknologi AI, dan melakukan observasi untuk memahami preferensi serta harapan pengguna terhadap aplikasi yang akan dikembangkan. Langkah-langkah ini bertujuan untuk memastikan bahwa aplikasi yang dibuat sesuai dengan kebutuhan pengguna dan teknologi yang tersedia.

#### b) Perancangan Sistem

Pada tahap ini, peneliti akan menentukan arsitektur software dan hardware yang diperlukan, merancang antarmuka pengguna yang sesuai dengan kebutuhan, menetapkan algoritma dan teknik AI untuk deteksi APD, serta mengidentifikasi sumber daya dan teknologi yang diperlukan untuk pengembangan aplikasi. Langkah-langkah ini akan membentuk dasar untuk merancang aplikasi dengan tepat sesuai dengan kebutuhan pengguna dan teknologi yang tersedia.

c) Implementasi

Setelah perancangan selesai, langkah implementasi melibatkan integrasi kode program dalam bahasa Python dan HTML menggunakan Google Colab serta Visual Studio Code. Langkah berikutnya adalah mengintegrasikan komponen kecerdasan buatan (AI) ke dalam aplikasi. Ini penting untuk memastikan aplikasi berjalan efektif sesuai perancangan yang telah ditetapkan sebelumnya.

d) Pengujian Sistem

Pengujian sistem ini dilakukan untuk memastikan bahwa aplikasi dapat berjalan sesuai yang diharapkan. Pengujian yang akan dilakukan dapat dengan melakukan pengujian fungsional dengan memakai atribut alat pelindung diri yang ada untuk menguji apakah semua fitur berjalan dengan benar.

e) Hasil

Tahap akhir pengembangan sistem ini adalah hasil akhir dari pembuatan sistem dengan terciptanya aplikasi pendekripsi alat pelindung diri menggunakan artificial intelligence yang dapat berfungsi dengan baik dan sesuai dengan kebutuhan pengguna.

4) Analisa Penunjang Sistem

a) Analisa Perangkat Lunak

Penunjang sistem dari perangkat lunak dalam konteks rancangan aplikasi APD ini akan fokus pada platform dan teknologi pendukung yang digunakan dalam pengembangan aplikasi. Berikut adalah beberapa contoh analisis penunjang sistem dari perangkat lunak:

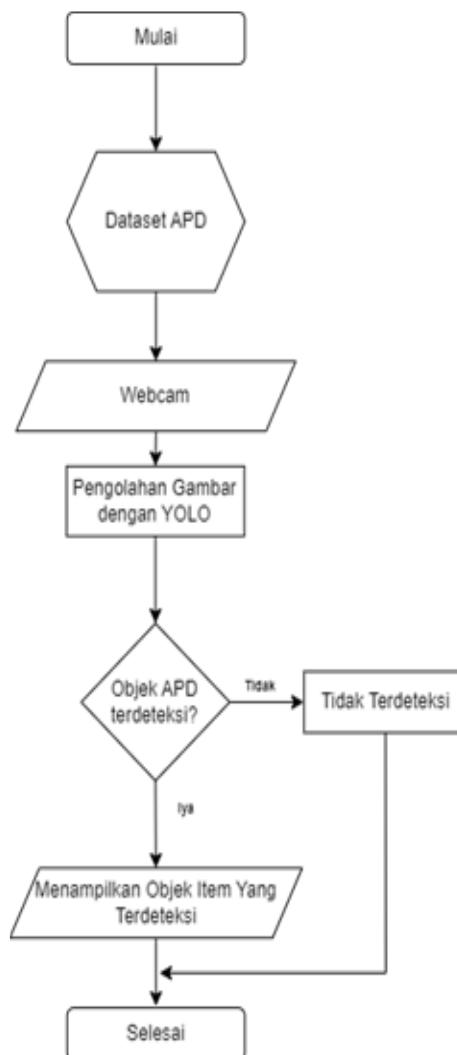
- 1) Operating system: Windows 10
- 2) Bahasa Pemrograman: Python
- 3) Algoritma: YOLO V5
- 4) Database: 4 item APD pekerja konstruksi (Helm, Masker, Rompi, dan Sepatu)
- 5) Antarmuka Pengguna: Website
- 6) Tools editor: Google Colab, Visual Studio Code
- 7) Aplikasi Penunjang: DroidCam

b) Analisa Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengoperasian aplikasi APD untuk pekerja proyek menggunakan AI sebagai berikut:

- 1) Perangkat: Laptop Asus X441 UV
- 2) Random Acces Memory (RAM): 4 GB
- 3) Akses Internet
- 4) Kamera Smartphone Infinix Hot 8

*Desain System*



Gambar 1. Flowchart Desain Sistem Deteksi

Berdasarkan flowchart system deteksi pada Gambar 1 dapat diuraikan sebagai berikut:

- 1) Diawali dengan Terminator Symbol Mulai.
- 2) Setelah itu system menyiapkan dataset APD
- 3) Kemudian system menampilkan webcam yang akan digunakan user untuk mendekripsi item APD

- 4) System akan melakukan proses pengolahan gambar dengan YOLO.
- 5) Lalu system akan memproses apakah item terdeteksi oleh system atau tidak
- 6) Jika system dapat mendeteksi item, maka akan muncul label nama item yang terdeteksi
- 7) Jika tidak terdeteksi, maka label nama item tidak akan muncul
- 8) Proses pendektsian selesai

#### Tahap Pembuatan Sistem Deteksi



Gambar 2. Tahap Pembuatan Sistem Deteksi

#### Problem Scoping

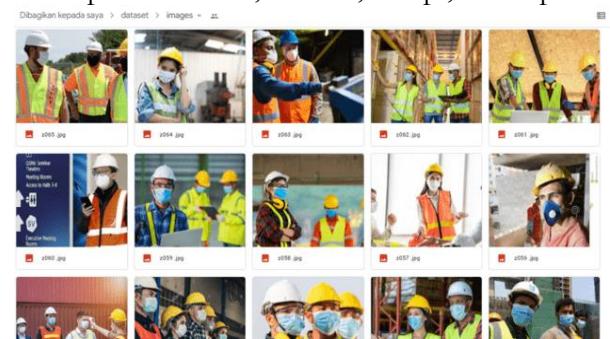
Problem Scoping dapat disebut juga dengan istilah identifikasi masalah. Dalam identifikasi masalah ini, Keselamatan saat bekerja adalah suatu hal yang sangat penting untuk diperhatikan. Terlebih keselamatan kerja bagi mereka yang bekerja dengan tingkat resiko bahaya yang tinggi. Seperti pekerja konstruksi, pekerja pabrik, teknisi listrik dan lain sebagainya. Tidak sedikit adanya kasus kecelakaan saat bekerja, dan mayoritas kasus yang terjadi itu dimana korban tidak menggunakan Alat Pelindung Diri sesuai dengan standar atau ketentuannya. Maka dari itu perlengkapan Alat Pelindung Diri sangatlah penting untuk mengurangi resiko bahaya bagi para pekerja di tempat kerjanya.

Solusi yang diusulkan adalah mengimplementasikan AI berbasis computer vision dengan menggunakan algoritma YOLO untuk mendeteksi APD pada pekerja konstruksi. Hal ini bertujuan untuk mengurangi risiko kecelakaan kerja dengan mendeksi penggunaan APD sesuai standar. Dengan aplikasi ini, diharapkan dapat mencegah kecelakaan, cedera, atau penyakit akibat kerja dengan memastikan bahwa para pekerja telah menggunakan

perlengkapan keselamatan sebelum memasuki area kerja.

#### Data Acquisition

Untuk proses data acquisition, peneliti menggunakan dataset dari Saravana Alagar. Peneliti mengunduh dataset yang terdiri dari 3000 gambar yang mencakup hard hat, masker, rompi (vest), dan sepatu bot. Tujuan dari dataset ini adalah untuk mendeksi perlengkapan APD seperti hard hat, masker, rompi, dan sepatu bot.



Gambar 3. Dataset Yang Digunakan

#### Data Exploration

Setelah memproses dataset di Google Colab, peneliti melakukan pelatihan (training) dengan jumlah epoch sebanyak 100 kali.

#### Modeling

Peneliti mengimplementasikan sistem deteksi perlengkapan APD menggunakan algoritma YOLO (You Only Look Once). YOLO adalah algoritma yang memanfaatkan jaringan saraf convolutional untuk deteksi objek secara real-time. Peneliti menggunakan YOLOv5, sebuah model generasi kelima dari YOLO yang dirilis pada April 2020. Algoritma ini ditulis dalam Python, mempertahankan arsitektur yang mirip dengan generasi sebelumnya.

#### Evolution

Dalam proyek pembuatan sistem deteksi APD, peneliti menggunakan metrik evaluasi seperti mAP, precision, dan recall yang umumnya digunakan untuk mengevaluasi keberhasilan program deteksi objek berbasis YOLOv5. Metrik mAP menjadi yang utama karena melibatkan perhitungan dari kedua metrik lainnya.

### 3. Hasil dan Pembahasan

#### Implementasi Algoritma YOLOv5

##### Install Algoritma YOLOv5

Instalasi algoritma YOLOv5 sebelum membuat aplikasi deteksi APD bertujuan untuk memanfaatkan model deteksi objek yang telah terlatih untuk mengidentifikasi dan mendeteksi APD pada gambar atau video secara efektif dan akurat. Langkah pertama yang harus dilakukan dalam membuat aplikasi deteksi APD menggunakan YOLOv5 adalah dengan menginstall YOLOv5 terlebih dahulu. Instalasi ini dilakukan menggunakan google colab, dengan perintah seperti pada gambar 4 dibawah ini.

```

Setup
Clone GitHub repository, install dependencies and check PyTorch and GPU.

git clone https://github.com/ultralytics/yolov5 # clone
  ↗ 100% 0KiB/s
Skip install -q requirements.txt # install
  ↗ 100% 0KiB/s
Import torch
Import utils
Display + utils.notebook_init() # checks
  ↗ 100% 0KiB/s

YOLOv5 v6.2-239-gf5372bf Python 3.7.15 torch-1.12.1+cu113 CUDA-11 (Tesla T4, 15GB)
Setup complete [2 CPUs, 12.7 GB RAM, 22.6/78.2 GB disk]

```

Gambar 4. Install Algoritma YOLOv5

#### Mengunduh Dataset

Untuk mengunduh dataset dari Google Drive ke Google Colab, peneliti perlu menghubungkan kedua platform tersebut. Proses ini disebut mounting, yang memungkinkan akses langsung ke file dan folder di Google Drive dari Google Colab. Setelah terhubung, peneliti dapat menggunakan kode Python untuk mengunduh dataset yang diperlukan ke lingkungan kerja Colab. Dengan demikian, data yang dibutuhkan untuk proyek dapat diakses dan dimanfaatkan tanpa perlu mengunduhnya terlebih dahulu ke perangkat lokal. Berikut perintah yang digunakan peneliti untuk mengunduh dataset.

```

!gdown https://drive.google.com/u/1/uc?id=1rhp67vgTT4H4C3pp0e7ghNtKflReport#download
  ↗ 100% 26.9M/26.9M [0:04:06.00, 54.4MB/s]

```

Gambar 5. Perintah Mengunduh Dataset

#### Training dataset

Training Dataset adalah sekumpulan data yang digunakan untuk melatih model dalam machine learning. Tujuannya adalah untuk membantu model

mempelajari pola atau informasi yang diperlukan untuk membuat prediksi yang akurat ketika dihadapkan pada data baru. Proses training pada dataset dengan jumlah epoch sebanyak 100 epoch. Hasil training dataset dapat dikatakan bagus karena memiliki parameter dengan persentase precision sebesar 90%, recall sebesar 90% dan mAP sebesar 93,4%.

Class	Images	Instances	P	R	mAP@50	mAP@95	100% AP@95	80% AP@95	70% AP@95
all	3538	16828	0.9	0.9	0.934	0.597	100%	88%	85%
hardhat	3538	5249	0.935	0.938	0.961	0.628			
vest	3538	4893	0.963	0.931	0.963	0.686			
mask	3538	2819	0.901	0.911	0.936	0.58			
boots	3538	3867	0.861	0.819	0.877	0.495			

Gambar 6. Hasil Training Dataset

#### Testing Dataset

Testing dataset merupakan bagian dari dataset yang digunakan untuk mengevaluasi kinerja suatu model atau algoritma setelah dilatih dengan dataset pelatihan. Tujuannya adalah untuk mengevaluasi performa model pada data baru dan memastikan keandalannya dalam menghadapi situasi dunia nyata. Dalam testing dataset ini melalui proses pengambilan gambar dan uji hasil deteksi gambar.

##### 1) Pengambilan Gambar

###### a) Inisialisasi Pengambilan Gambar

Langkah pertama dalam proses pengambilan gambar adalah dengan melakukan inisialisasi pengambilan gambar terlebih dahulu. Tujuan dari inisialisasi pengambilan gambar ini adalah untuk untuk mengakses sumber gambar yang akan digunakan sebagai input untuk deteksi APD.

```

# Function to convert the Javascript object into an OpenCV Image
def js_to_image(js_reply):
    frame = js_reply['frame']
    img = frame['image'].data
    img = np.frombuffer(img, dtype=np.uint8)
    img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
    return img

# Function to convert OpenCV Bounding Box into a base64 byte string to be overlaid on video stream
# and then to binary image array
def box_to_b64(image):
    bytes = base64.b64encode(image)
    return bytes

# Convert array into PIL Image
def bytes_to_pil(bytes):
    image = BytesIO()
    image.write(base64.b64decode(bytes))
    image.seek(0)
    image = Image.open(image)
    return image

# Convert array into binary image
def bytes_to_binary(bytes):
    image = BytesIO()
    image.write(base64.b64decode(bytes))
    image.seek(0)
    image = np.frombuffer(image.read(), dtype=np.uint8)
    return image

```

Gambar 7. Inisialisasi Pengambilan Gambar

### b) Testing Pengambilan Gambar

Langkah selanjutnya yang harus dilakukan adalah testing atau uji coba pengambilan gambar. Langkah ini bertujuan untuk memastikan apakah aplikasi deteksi dapat mendeteksi dan mengenali serta memproses gambar dari sumber yang baru atau belum pernah terlihat sebelumnya. Dengan uji coba pengambilan gambar, aplikasi dapat mengevaluasi sejauh mana model dapat mengidentifikasi APD dengan akurat dan responsif terhadap situasi dunia nyata.

```
[5]: from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo(filename='photo.jpg', quality=0.8):
    js = Javascript('''
        async function takePhoto(quality) {
            const div = document.createElement('div');
            const mailt = document.createElement('button');
            mailt.textContent = 'Mail';
            div.appendChild(mailt);
            const video = document.createElement('video');
            video.style.display = 'block';
            const stream = await navigator.mediaDevices.getUserMedia({video: true});

            document.body.appendChild(div);
            div.appendChild(video);
            video.srcObject = stream;
            await video.play();

            // Resize the output to fit the video element.
            google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

            // Wait for capture to be claimed
            await new Promise((resolve) => mailt.onclick = resolve);

            const canvas = document.createElement('canvas');
            canvas.height = video.videoHeight;
            canvas.width = video.videoWidth;
            video.getVideoTracks()[0].stop();
            video.remove();
            document.body.removeChild(div);
            const img = document.createElement('img');
            img.src = canvas.toDataURL('image/jpeg', quality);
        }
    ''')
    display(js)
    binary = b64decode(js.eval_js('takePhoto('+str(quality)+')'))
    with open(filename, 'wb') as f:
        f.write(binary)
        f.close()

take_photo()
```

Gambar 8. Coding Testing Pengambilan Gambar

### c) Uji Hasil Deteksi Gambar

Tujuan dari uji hasil gambar ini adalah untuk mengevaluasi keakuratan dan kinerja model deteksi APD yang telah dibuat. Dengan melakukan uji hasil gambar, kita dapat mengukur seberapa baik model dapat mengenali dan mendeteksi APD dalam gambar.

```
[6]: !python detect.py --weights /content/content/yolov5/runs/train/exp/weights/last.pt --source photo.jpg

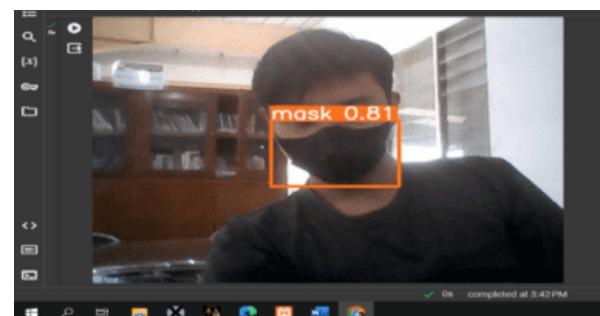
detect: weights:[/content/content/yolov5/runs/train/exp/weights/last.pt], source:photo.jpg, data-data/coco128.yaml, imgsz:[640, 640], conf_thres:0.25, iou_thres:0.45, YOLOv5 v7.0-8-ge596eb Python-3.12.1-torch-2.1.0-cuda10 CUDA-0 [Tesla T4, 15GB]M
Fusing layers...
Model summary: 157 layers, 700003 parameters, 0 gradients, 15.8 GFLOPs
Image 1/1 /content/content/yolov5/photo.jpg: 400x640 1 mask, 46.2ms
Speed: 0.0ms pre-process, 46.2ms inference, 79.4ms NMS per image at shape (1, 3, 640, 640)
Results saved to /run/detect/exp
```

```
[7]: !import glob
!from IPython.display import Image, display

for imagefile in glob.glob('/content/content/yolov5/runs/detect/exp/*.jpg'):
    display(Image(filename=imagefile))
    print("")

!rm -rf /content/content/yolov5/runs/detect/exp
```

Gambar 9. Coding Uji Hasil Gambar

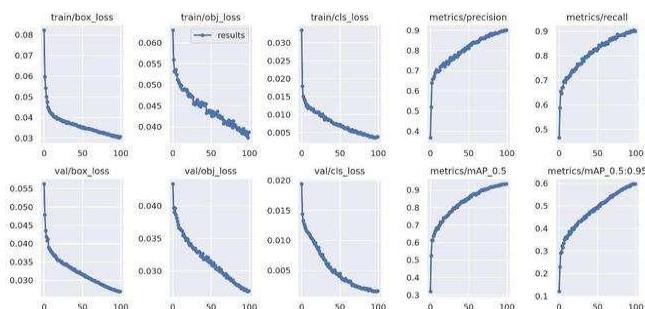


Gambar 10. Hasil Uji Deteksi Gambar

### d) Evaluation

Proses *evaluation* ini dilakukan untuk visualisasi performa deteksi model. Hasil visualisasi akan ditampilkan dalam bentuk scatter plot yang dapat digunakan untuk menampilkan hasil evaluasi berupa metrik seperti precision, recall, atau mAP (mean Average Precision) pada setiap kelas atau objek yang terdeteksi.

Dilihat dari hasil visualisasi, ini memiliki persentase prediksi dengan kelas hardhat (helm) sebesar 96%, vest (rompi) sebesar 96%, mask (masker) sebesar 95%, dan kelas boots (Sepatu) 92%.



Gambar 11. Visualisasi Hasil Training Dataset

### Penggabungan Coding Deteksi Pada Website

Dalam pembuatan aplikasi deteksi APD menggunakan algoritma YOLOv5 berbasis website, peneliti perlu melakukan penggabungan kode atau skrip yang digunakan untuk mendeteksi peralatan pelindung diri (APD) ke dalam struktur atau tampilan situs web. Hal ini memungkinkan deteksi APD berbasis YOLOv5 dapat diakses dan digunakan secara praktis melalui antarmuka web.

Gambar 12. Coding Deteksi Pada Website

## Tampilan Website

Aplikasi deteksi APD yang peneliti buat berbasis pada website, yang dimana dalam website terdapat 6 halaman, yang terdiri dari halaman *home*, *about application*, *hot news*, *detection gallery*, *detection*, *contact*. Pada tampilan halaman *home* ini menampilkan halaman awal berupa nama dari aplikasi yaitu *My Protection*, dan *tagline* aplikasi yaitu *Safety Is Our Top Priority*.



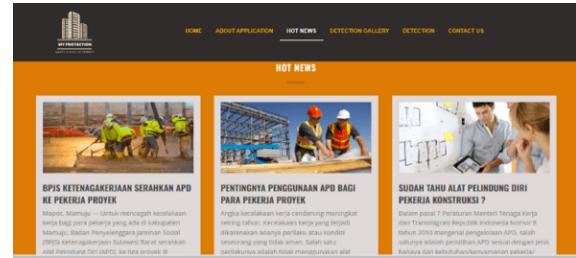
Gambar 13. Halaman *Home*

Halaman *about application* ini menampilkan informasi tentang definisi dari aplikasi *My Protection*, tujuan dari dibuatnya aplikasi dan kegunaan aplikasi ini.



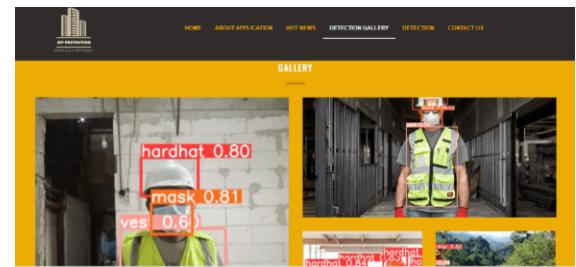
Gambar 14. Halaman *About Application*

Tampilan *hot news* menampilkan berita-berita terkini seputar penggunaan APD dan peraturan-peraturan terkait penggunaan APD.



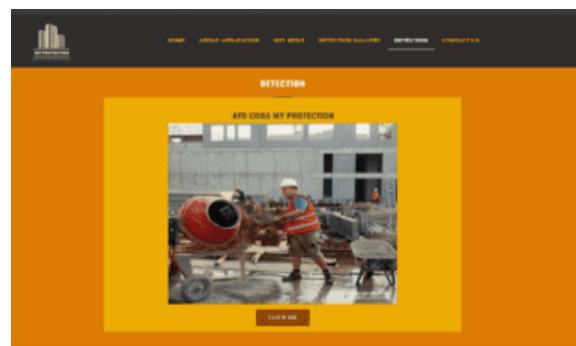
Gambar 15. Halaman *Hot news*

Halaman *detection gallery* menampilkan hasil deteksi berupa *image*.



Gambar 16. Halaman *Detection Gallery*

Tampilan pada halaman detection akan mengarahkan user ke halaman deteksi. Pada halaman detection ini user diharuskan untuk menekan tombol *click me* untuk dapat menampilkan halaman untuk mendeteksi perlengkapan APD secara *real time*.



Gambar 17. Halaman *Detection*

#### **4. Kesimpulan**

Dari penelitian yang dilakukan oleh peneliti, pembuatan aplikasi deteksi yang berbasis website menggunakan algoritma YoloV5, dengan dataset yang terdiri dari 4 objek APD berupa hardhat (helm), mask (masker), vest (rompi), boots (Sepatu). Hasil training dari 4 objek memiliki hasil parameter dengan persentase yang tinggi, yaitu precision sebesar 90%, recall sebesar 90% dan mAP sebesar 93,4%.

Sedangkan untuk persentase prediksi pada masing-masing objek dengan hardhat (helm) sebesar 96%, vest (rompi) sebesar 96%, mask (masker) sebesar 95%, dan kelas boots (Sepatu) 92%. Berdasarkan dari hasil uji coba terlihat bahwa dengan menggunakan algoritma YoloV5, aplikasi dapat mendeteksi setiap objek dengan baik dan akurat. Dalam pembuatan aplikasi deteksi APD untuk pekerja proyek ini, diharapkan dapat meminimalisir kasus kecelakaan kerja yang terjadi pada pekerja proyek serta untuk memastikan bahwa para pekerja proyek telah memakai APD lengkap sesuai dengan standar yang ditentukan.

## 5. Daftar Pustaka

- [1] Azizah, D. N., Pulungan, R. M., Utari, D., & Amrullah, A. A. (2021). Faktor-Faktor yang Berhubungan dengan Kepatuhan Menggunakan Alat Pelindung Diri (APD) pada Pekerja Proyek Pembangunan PLTGU Muara Tawar (Persero). *Jurnal Ilmiah Kesehatan Masyarakat: Media Komunikasi Komunitas Kesehatan Masyarakat*, 13(3), 141-150.
- [2] Sofyan, M. T. (2023). Hubungan Alat Pelindung Diri Terhadap Kecelakaan Kerja; Literature Riview. *JOURNAL SCIENTIFIC OF MANDALIKA (JSM)* e-ISSN 2745-5955 | p-ISSN 2809-0543, 4(5), 71-75. DOI: <https://doi.org/10.36312/10.36312/vol4iss5> pp71-75.
- [3] JAYANTI, U., ALI, H., REFLIS, R., RAMDHON, M., UTAMA, S., ADEKO, R., ... & SISWAHYONO, S. (2023). ANALISIS PENGGUNAAN ALAT PELINDUNG DIRI DAN KECELAKAAN KERJA PADA PEKERJA PABRIK KELAPA SAWIT DI PT. PALMA MAS SEJATI KABUPATEN BENGKULU TENGAH. *Journal of Nursing and Public Health*, 11(1), 272-278. DOI: <https://doi.org/10.37676/jnph.v11i1.4138>.
- [4] Adi, Y. K., & Kushartomo, W. (2023). Analisis Penerapan Keselamatan Dan Kesehatan Kerja (K3) Pada Proyek X Di Jakarta Pusat. *Jmts: Jurnal Mitra Teknik Sipil*, 589-594.
- [5] Djaohar, M., & Sunawar, A. (2022). RANCANG BANGUN PENGECEKAN ALAT PELINDUNG DIRI MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE (YOLO). *Journal of Electrical Vocational Education and Technology*, 7(1), 12-18.
- [6] Susanti, S., Aulia, S., & Irawati, I. D. (2023). Deteksi Helm Otomatis Untuk Keselamatan Kerja di Tempat Proyek Berbasis Yolo. *eProceedings of Applied Science*, 9(1).
- [7] Mailoa, R. M., & Santoso, L. W. (2022). Deteksi Rompi dan Helm Keselamatan Menggunakan Metode YOLO dan CNN. *Jurnal Infra*, 10(2), 49-55.
- [8] Nurfirmansyah, A., & Dijaya, R. (2022, August). DETEKSI KELALAIAN ALAT PELINDUNG DIRI (APD) PADA PEKERJA KONTRUKSI BANGUNAN. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 6, No. 1, pp. 058-063). DOI: <https://doi.org/10.29407/inotek.v6i1.2452>.
- [9] Ashar, M. H., & Suarna, D. (2022). Implementasi Algoritma YOLOv5 dalam Mendeteksi Penggunaan Masker Pada Kantor Biro Umum Gubernur Sulawesi Barat. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 3(3), 298-302. DOI: <https://doi.org/10.30865/klik.v3i3.559>.
- [10] Riva, L. S., & Jayanta, J. (2023). Deteksi Penyakit Tanaman Cabai Menggunakan Algoritma YOLOv5 Dengan Variasi Pembagian Data. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(3), 248-254. DOI: <https://doi.org/10.30591/jpit.v8i3.5679>.
- [11] Asmara, R. A., Samudra, M. R., & Wibowo, D. W. (2022). Identifikasi Person Pada Game First Person Shooter (Fps) Menggunakan Yolo Object Detection Dan Diimplementasikan Sebagai Agent Cerdas Automatic Target Hit. *Jurnal Teknik Ilmu Dan Aplikasi*, 3(2), 174-178.

- [12] Guntara, R. G. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendekripsi Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 55-60. DOI: <https://doi.org/10.47233/jteksis.v5i1.750>.
- [13] Widharma, I. G. S., Santyary, P. A. W., Sunaya, I. N., Darminta, I. K., Sangka, I. G. N., & Widiatmika, P. A. W. (2022). Deteksi api kebakaran berbasis computer vision dengan algoritma YOLO. *Journal of Applied Mechanical Engineering and Green Technology*, 3(2), 53-58. DOI: <https://doi.org/10.31940/jametech.v3i2.53-58>.
- [14] Liunanda, C. N., Rostianingsih, S., & Purbowo, A. N. (2020). Implementasi Algoritma YOLO pada Aplikasi Pendekripsi Senjata Tajam di Android. *Jurnal Infra*, 8(2), 235-241.
- [15] Adinata, Y., Gunadi, K., & Sugiarto, I. (2022). Aplikasi Deteksi Jumlah Orang pada Area Indoor Untuk Mendukung Pelaksanaan PPKM dengan Metode YOLO. *Jurnal Infra*, 10(1), 129-135.
- [16] Indaryanto, F., Nugroho, A., & Suni, A. F. (2021). Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19. *Edu Komputika Journal*, 8(1), 31-38. DOI: <https://doi.org/10.15294/edukomputika.v8i1.47837>.
- [17] Prasanta, M. R., Pranata, M. Y., Firnanda, M. A., & Sendari, S. (2022). Rancang Bangun Quadcopter Drone Untuk Deteksi Api Menggunakan YOLOv4. *CYCLOTRON*, 5(1). DOI: <https://doi.org/10.30651/cl.v5i1.10013>.