

# Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan Faster R-CNN

Jonathan Adiwibowo<sup>1</sup>

Program Studi Informatika  
Fakultas Teknologi Industri  
Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya  
60236

Telp. (031) – 2983455, Fax. (031) –  
8417658

jonathanadi46@gmail.com<sup>1</sup>

Kartika Gunadi<sup>2</sup>

Program Studi Informatika  
Fakultas Teknologi Industri  
Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya  
60236

Telp. (031) – 2983455, Fax. (031) –  
8417658

kgunadi@petra.ac.id<sup>2</sup>

Endang Setyati<sup>3</sup>

Program Studi Teknologi Informasi  
Fakultas Sains dan Teknologi  
Institut Sains dan Teknologi Terpadu  
Surabaya

Jl. Ngagel Jaya Tengah No.73-77  
Surabaya, 60284

Telp. (031) – 5027920

endang@stts.edu<sup>3</sup>

## ABSTRAK

Upaya memastikan langkah-langkah keamanan dan keselamatan di zona industri atau area lain yang perlu menggunakan alat pelindung diri merupakan suatu masalah penting. Banyak pekerja yang kerap mengabaikan dan melanggar aturan untuk menggunakan alat pelindung diri di zona mereka bekerja. Oleh karena itu, sebuah program dibuat dengan tujuan membantu mengawasi para pekerja untuk menggunakan alat pelindung diri. Dalam penelitian ini dilakukan percobaan untuk membantu mengenali karakteristik alat pelindung diri, terutama di area kepala. Dalam penelitian terbaru yang telah dilakukan Rifki Dita Wahyu Pradana, et. Al. menggunakan CNN untuk menghasilkan akurasi keseluruhan sebesar 80%. Penelitian ini menggunakan dua metode, yaitu *You Only Look Once* dan *Faster Region-Convolutional Neural Network (Faster R-CNN)*. Metode YOLO digunakan untuk menemukan wilayah kepala pekerja sementara metode F-RCNN yang lebih cepat digunakan untuk mengklasifikasikan peralatan pelindung pribadi yang digunakan oleh pekerja. Hasil klasifikasi R-CNN yang lebih cepat dihitung menggunakan *confusion matrix* untuk mendapatkan akurasi prediksi yang benar. Hasil dari penelitian ini dapat mengidentifikasi pekerja yang menggunakan peralatan pelindung pribadi dalam video. Akurasi rata-rata yang telah diperoleh adalah 93,61%.

**Kata Kunci:** Deteksi, Alat Pelindung Diri, YOLO, Faster R-CNN

## ABSTRACT

*In order to ensure security and safety measures in industrial zones or any other areas that needed to use personal protective equipment are an important matter. Many workers keep disregarding and violating the rule to use personal protective equipment in the area. Therefore, a program was created that could help supervising the workers to use personal protective equipment. In this study an experiment will be conducted to help recognize the characteristics of personal protective equipment, especially in head. In recent studies that have been carried out Rifki Dita Wahyu Pradana, et. Al. using CNN to produce an overall accuracy 80%. This study will be using 2 methods, You Only Look Once and Faster Region-Convolutional Neural Network (Faster R-CNN). The YOLO method is used to find regions of worker's head while Faster R-CNN method is used to classify personal protective equipment used by worker. The results of the Faster R-CNN classification will be calculated using a confusion matrix in order to get the accuracy of the correct prediction. The results from this study will*

*identify workers using personal protective equipment in the video. Average accuracy that has been obtained is 93.61%.*

**Keywords:** *Detection, Personal Protective Equipment, YOLO, Faster R-CNN*

## 1. PENDAHULUAN

Kecelakaan industri secara umum disebabkan oleh 2 hal yaitu perilaku kerja yang berbahaya dan kondisi yang berbahaya. Beberapa hasil penelitian menunjukkan bahwa faktor manusia memegang peranan penting dalam munculnya kecelakaan kerja. Berdasarkan data kecelakaan kerja versi BPJS Ketenagakerjaan pada tahun 2016, kecelakaan yang terjadi dalam bidang industri manufaktur sebesar 31 persen. Menurut Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia no 8 tahun 2010, para pekerja diwajibkan menggunakan Alat Pelindung Diri (APD) saat memasuki tempat kerja [1].

Berkaitan dengan sistem deteksi untuk para pekerja mengenai APD, banyak perusahaan yang masih mengadopsi sistem manual dalam pengecekan APD oleh manusia. Secara umum sistem tersebut tidak layak karena keterlibatan manusia yang efisiensinya menurun dalam durasi yang lama [2]. Otomatisasi menjadi peran penting untuk memantau para pekerja yang tidak menggunakan APD. Dengan adanya otomatisasi mampu mengurangi beban kerja *supervisor* terhadap pekerja yang tidak memakai APD. Solusi yang tepat untuk menghadapi masalah ini adalah menggunakan *object detection* pada video kamera pengintai di zona wajib alat pelindung diri [3].

Menurut Shiming ye, pendeteksian masker hanya menggunakan metode LLE-CNN untuk mendeteksi wajah bermasker. Kendala yang didapatkan dari penelitian ini adalah jenis dan corak masker yang beragam membuat akurasi mengalami penurunan [4]. Menurut Abu H.M. Rubaiyat, pendeteksian *safety helmet* yang mana menggunakan *Circle Hough Transform* menunjukkan bahwa akurasi paling besar adalah ketika pekerja tidak menggunakan helm daripada akurasi pekerja yang menggunakan helm. Kendala yang ditemukan adalah kesulitan dalam membedakan antara topi dan *safety helmet* serta deteksi yang dilakukan memerlukan seluruh badan [3]. Menurut Pradana, penggunaan metode CNN untuk mendeteksi APD pada bagian kepala. Rata-rata tingkat akurasi yang diperoleh yaitu 80%. Pengambilan data dan *dataset* yang digunakan kebanyakan berupa gambar statis dimana para pekerja yang dicek oleh sistem harus berdiri tetap selama 3 detik. Kendala yang ditemukan adalah pendeteksian masker dapat mendeteksi kacamata juga sehingga menjadi masker dan kacamata [5].

Metode yang diterapkan pada penelitian ini adalah *Faster R-CNN* (*Region Convolutional Neural Network*) dan *YOLO* (*You Only Look Once*). Diharapkan melalui penggabungan dua metode ini dapat menghasilkan akurasi dan kecepatan yang baik dikarenakan metode *YOLO* merupakan *state of the art* dalam mendeteksi secara *real-time* [6] dan metode *Faster R-CNN* baik untuk meningkatkan akurasi dan memiliki kecepatan yang hampir sama dengan *YOLO*.

## 2. DASAR TEORI

### 2.1 Alat Pelindung Diri

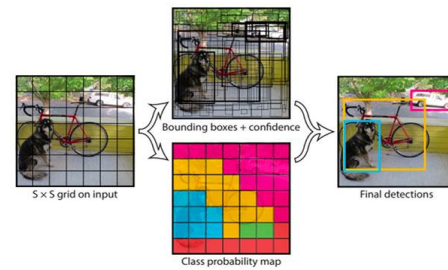
Menurut Tarwaka (2008), Alat Pelindung Diri (APD) adalah seperangkat alat keselamatan yang digunakan oleh pekerja untuk melindungi seluruh atau Sebagian tubuhnya dari kemungkinan adanya pemaparan potensi bahaya lingkungan kerja terhadap kecelakaan dan penyakit akibat kerja [7]. Sedangkan menurut Budiono (2005), Alat Pelindung Diri (APD) adalah seperangkat alat yang digunakan tenaga kerja untuk melindungi sebagian atau seluruh tubuhnya dari potensi kecelakaan kerja. APD tidak secara sempurna dapat melindungi tubuhnya, tetapi dapat mengurangi tingkat keparahan yang mungkin terjadi. Pengendalian ini sebaiknya tetap dipadukan dan sebagai pelengkap pengendalian teknis atau pengendalian administratif [8].

Berdasarkan surat dari Menteri Tenaga Kerja dan Trasmiharsi nomor PER. 08/MENVIII/2010, jenis-jenis alat perlindungan diri antara lain: 1) Alat pelindung kepala, yang terdiri dari helm pengaman (*safety helmet*), topi atau tudung kepala, penutup atau pengaman rambut, dan lain-lain; 2) Alat pelindung mata dan muka, yakni kacamata pengaman (*spectacles*), *goggles*, tameng muka (*face shield*), masker selam, tameng muka dan kacamata pengaman dalam kesatuan (*full face masker*); 3) Alat pelindung telinga, yakni sumbat telinga (*earplug*) dan penutup telinga (*earmuff*); 4) Alat pelindung pernapasan beserta perlengkapannya, yang terdiri dari masker, respirator, katrit, canister, *re-breather*, *airline respirator*, *Continues Air Supply Machine*, *Air Hose Mask Respirator*, tangka selam dan regulator (*Self-Contained Underwater Breathing Apparatus/SCUBA*), *Self-Contained Breathing Apparatus (SCBA)*, dan *emergency breathing apparatus* [1].

### 2.2 You Only Look Once (YOLO)

*You Only Look Once* (YOLO) mengimplementasikan *single convolutional network* secara bersamaan untuk memprediksi area suatu objek. Metode ini dilatih melalui gambar yang sudah disediakan dan langsung mencari cara yang terbaik untuk mengoptimalkan performanya sendiri. YOLO bertugas untuk membagi gambar menjadi *grid* berukuran  $S \times S$ . Apabila suatu objek berada dalam sel *grid*, sel tersebut bertanggung jawab untuk memprediksi objek tersebut.

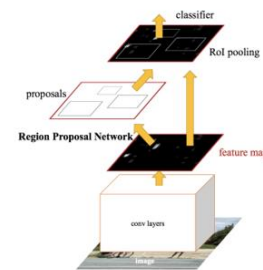
Setiap *grid* sel memprediksi  $B$  *Bound Boxes* dan nilai daripada *bounding boxes* tersebut. Nilai tersebut berfungsi sebagai apakah terdapat model di dalam kotak tersebut dan seberapa tinggi tingkat akurasinya. Setiap *bounding boxes* berisi 5 prediksi yaitu  $x, w, y, h$  dan *confidence score*. Koordinat  $(x, y)$  mewakili pusat berisi dari kotak yang terbatas oleh sel *grid*. Sedangkan  $w$  dan  $h$  merupakan hasil prediksi lebar dan tinggi suatu objek terhadap gambar keseluruhan. *Confidence score* bertugas mewakili *IOU* (*Intersection Over Union*). Setiap sel *grid* juga akan memprediksi 1 set *class* probabilitas  $\Pr(\text{Class}|\text{Object})$ . Skor yang dihasilkan mengkodekan kedua probabilitas *class* tersebut seberapa baik kotak yang diprediksi dan cocok dengan objek [6]. Gambar 1 adalah diagram algoritma YOLO.



Gambar 1. Diagram Algoritma YOLO [6]

### 2.3 Faster R-CNN (Regional-Convolutional Neural Network)

*Faster Region based Convolutional Neural Network* atau biasa disingkat *Faster R-CNN* merupakan sebuah metode perubahan dari *Fast R-CNN*. Perubahan ini dilakukan karena terjadi *bottleneck* terhadap arsitekturnya pada bagian *selective search*, karena *selective search* diperlukan untuk menghasilkan 2000 proposal per gambar. Bagian tersebut merupakan bagian yang paling banyak memakai waktu dalam pelatihan. Di dalam *Faster R-CNN selective search* diubah menggunakan RPN (*Region Proposal Network*) sehingga arsitektur utama dari *Faster R-CNN* adalah *Fast R-CNN* dan RPN [10]. Gambar 2 merupakan diagram arsitektur FRCNN.



Gambar 2. Diagram Arsitektur *Faster R-CNN* [10]

### 2.4 Confusion Matrix

*Confusion matrix* adalah suatu informasi mengenai prediksi dan aktual dari suatu sistem klasifikasi yang memiliki data jawaban benar (*supervise*). *Confusion matrix* menggunakan data dalam bentuk matriks [11]. Dari *table* matriks yang didapatkan akan dihitung akurasi, presisi, *F-Score*, dan *recall* berdasarkan kondisi data yang diprediksi atau diklasifikasi [12]. Isi dari tabel *confusion matrix* ada 4, antara lain: 1) *True Positive* (TP), merupakan kondisi dimana model memprediksi data sebagai ya (*TRUE*) dan jawaban aktualnya adalah ya (*TRUE*); 2) *True Negative* (TN), merupakan kondisi dimana model memprediksi data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah tidak (*FALSE*); 3) *False Positive* (FP), merupakan kondisi dimana model memprediksi data sebagai ya (*TRUE*) dan jawaban aktualnya adalah tidak (*FALSE*); 4) *False Negative* (FN), kondisi dimana model memprediksi data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah ya (*TRUE*).

#### 2.4.1 Precision & Recall

*AP* (*Average Precision*) merupakan pengukuran yang populer di dalam object detection seperti *Faster R-CNN*, *SSD*, dll. *Average precision* menghitung nilai dari 0 sampai 1. Biasanya *AP* digunakan sebagai acuan tingkat akurasi suatu model yang mana diubah menjadi *mAP* (*Mean Average Precision*) [14].

Akurasi merupakan suatu tingkat pengukuran dari rasio benar (positif dan negatif) dengan keseluruhan data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Presi merupakan perbandingan jumlah data yang di prediksi benar postif dengan keseluruhan hasil yang diprediksi positif.

$$Precision = \frac{TP}{FP + TP}$$

Recall merupakan pengukuran pada data dengan klasifikasi positif yang benar.

$$Recall = \frac{TP}{TP + FN}$$

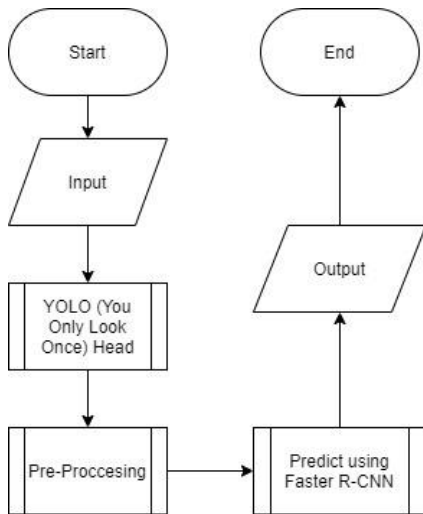
Confusion matrix juga dapat digunakan untuk menghitung *F1 Score* yang bertujuan untuk melakukan perbandingan rata-rata presisi dan recall.

$$F-Score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

### 3. DESAIN SISTEM

#### 3.1 Analisis Sistem

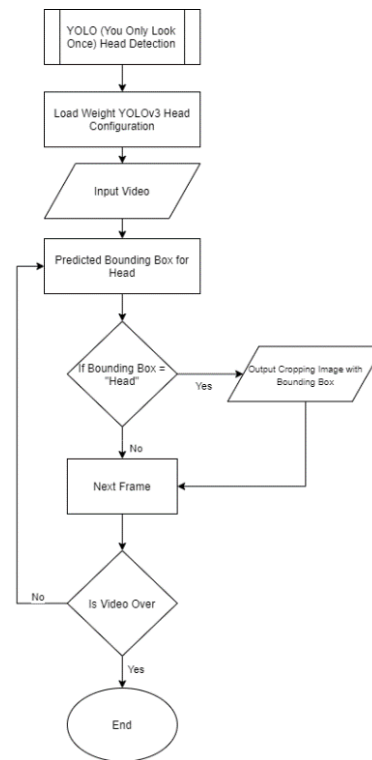
Flowchart pada Gambar Sistem akan menerima inputan berupa video lalu akan diterapkan metode YOLO (*You Only Look Once*) untuk mengetahui lokasi kepala daripada pekerja. Metode YOLO bekerja untuk mengenali lokasi dari kepala pekerja dan mengirimkan gambar tersebut kepada Faster R-CNN. Setelah itu metode Faster R-CNN (*Region Convolutional Neural Network*) akan mengklasifikasikan APD yang digunakan oleh pekerja. Garis besar sistem ditunjukkan pada gambar 3.



Gambar 3. Flowchart Garis Besar Sistem

##### 3.1.1 You Only Look Once (YOLO)

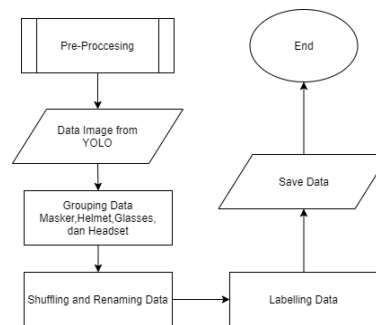
*You Only Look Once* (YOLO) digunakan untuk mendeteksi bagian kepala dari pekerja. *Pre-Trained model* yang digunakan dari YOLO akan dilatih sendiri untuk memprediksi kepala, yang diambil dari *Google Open Images Dataset*. Melalui *training* sendiri menggunakan *Darknet-53* dan *dataset* yang ada. Dalam hal ini YOLOv3 akan mendapatkan nilai keberadaan *bounding box* dan *confidence score*-nya. Setelah itu YOLO akan mengklasifikasi dan memberikan *bounding box* pada bagian kepala pekerja. Gambar 4. merupakan *flowchart* dari penggunaan yolo.



Gambar 4. Alur Penggunaan YOLO

##### 3.1.2 Preprocessing

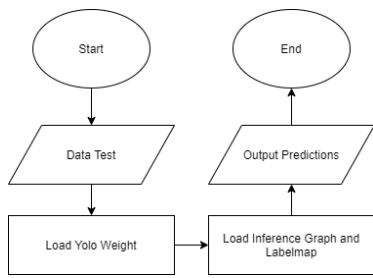
*Preprocessing* dilakukan untuk memilah data yang didapat dari YOLO yang sudah berhasil mendeteksi kepala agar mempermudah proses *Faster R-CNN*. Awal dari *Preprocessing* sendiri dilakukan dengan mengumpulkan data yang menggunakan *Masker*, *Safety Helmet*, *Safety Glasses*, dan *Safety Headset*. Setelah proses pengumpulan data selesai, data akan dijadikan dalam satu *folder* untuk melakukan *shuffling* dan *renaming* data yang beragam agar lebih mudah. Selanjutnya data akan diberikan *labeling* untuk menentukan *annotation* dan *bounding box* pada masing-masing kelas. Setelah semua alur *Pre-Processing* selesai maka data akan disimpan terlebih dahulu untuk membantu pelatihan *Faster-RCNN*. Gambar 5 menunjukkan alur *preprocessing* dari awal sampai akhir.



Gambar 5. Alur Preprocessing

##### 3.1.3 Faster R-CNN

Pada Gambar 6. proses *testing Faster R-CNN* akan dibantu oleh YOLO yang mendeteksi bagian kepala. Setelah kepala terdeteksi *Faster R-CNN* akan mendeteksi APD yang dipakai oleh pekerja berupa *bounding box* dan nama dari APD yang dideteksi. Jika APD yang dideteksi tidak sesuai dengan kriteria yang diberikan, maka akan dipisahkan ke dalam *folder* yang berbeda.



Gambar 6. Alur Testing Faster R-CNN

## 4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada komputer dengan spesifikasi berikut: 1) RAM: 32GB DDR4; 2) Memory: 512 GB SSD + 1T HDD; 3) CPU: Ryzen 5 3600; 4) GPU: NVIDIA GeForce RTX 2070 Super; 5) OS Windows 10 PRO.

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.7 64bit. Adapun beberapa library yang mendukung sistem ini, antara lain: 1) OpenCV 4.2.0; 2) Tensorflow; 3) Darknet; 4) Flask; 5) Numpy.

## 5. ANALISA DAN PENGUJIAN

### 5.1 Pengujian Sistem

Pengujian sistem digunakan untuk menilai performa dari *pre-trained model* yang digunakan. Penilaian performa digunakan untuk melihat tingkat akurasi terhadap hasil training yang dilakukan.

#### 5.1.1 Pengujian Training YOLO

Pengujian training YOLO akan mengevaluasi akurasi dari YOLO yang dilatih pada masing-masing ukuran yang ditentukan. Konfigurasi yang digunakan ada lima yaitu 320, 416, 512, 608, dan 800. Akurasi tertinggi didapatkan pada ukuran 800 dengan 73.83% pada dataset milik Open Images Dataset. Tabel 1 menunjukkan hasil akurasi YOLO pada threshold 50%

Tabel 1. Akurasi YOLO pada Threshold 50%

Weight	320	416	512	608	800
Last51	65.44%	68.31%	<b>69.95%</b>	68.31%	69.15%
Last52	64.89%	68.30%	69.44%	69.35%	72.60%
Last53	<b>66.82%</b>	68.42%	69.13%	70.96%	<b>73.83%</b>
Last54	64.72%	<b>68.55%</b>	69.71%	69.44%	73.57%
Last55	64.68%	68.12%	69.61%	<b>71.11%</b>	72.43%

#### 5.1.2 Pengujian Training Faster R-CNN

Pengujian training FRCNN akan mengevaluasi tingkat akurasi dan loss pada *pre-trained model* yang digunakan. Dataset yang digunakan ada dua jenis yaitu, dataset heterogen campuran dari YOLO dan Google, dan dataset homogen yaitu YOLO dan Google. *Pre-trained model* yang digunakan adalah FRCNN Inception V2, FRCNN Resnet50, dan FRCNN Resnet101. Tabel 2 menunjukkan jumlah objek masing-masing dataset.

Tabel 2. Jumlah objek masing-masing dataset

Dataset	Masker	Helmet	Headset	Glasses	Total
Campuran	482	417	509	454	1862
YOLO	506	251	447	381	1585
Google	297	283	188	320	1088

#### 5.1.2.1 Pre-Trained Faster R-CNN

Pengujian ini akan mengevaluasi hasil dari *training* FRCNN dengan konfigurasi awal yang tidak diubah sama sekali. Tabel 3, 4, dan 5 menunjukkan Perbandingan Step, mAP dan Loss masing-masing model pada dataset pertama, kedua dan ketiga.

Tabel 3. Perbandingan Step, mAP, dan Loss

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	Loss	mAP	loss
2000	88.95	0.7312	82.65	0.6437	82.53	0.7063
3000	90.83	0.6976	86.88	0.6561	83.36	0.7527
4000	91.8	0.6942	89.58	0.6166	86.25	0.7588
5000	92.71	0.6903	92.08	0.6367	86.46	0.7047
6000	93.73	0.6852	93.5	0.6485	88.65	0.7655
7000	94.03	0.6863	93.96	0.6429	<b>90.75</b>	<b>0.736</b>
8000	<b>94.22</b>	<b>0.6785</b>	<b>94.62</b>	<b>0.6338</b>	92.42	0.759
9000	93.87	0.6752	94.73	0.649	92.62	0.7534
10000	93.88	0.6698	94.95	0.6748	92.84	0.751

Tabel 4. Perbandingan Step, mAP, dan Loss

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	loss	mAP	loss
2000	88.31	0.8523	84.22	0.7944	82.92	0.8225
3000	91.63	0.7957	87.72	0.7912	88.20	0.8331
4000	92.07	0.7855	91.73	0.7579	89.66	0.8152
5000	93.39	0.7857	92.03	0.7555	90.67	0.8274
6000	93.97	0.7872	92.89	0.8229	90.13	0.8584
7000	94.78	0.7776	92.86	0.8058	91.13	0.8544
8000	95.31	0.7993	93.77	0.7696	<b>92.45</b>	<b>0.8638</b>
9000	<b>95.59</b>	<b>0.7783</b>	<b>94.36</b>	<b>0.7814</b>	92.03	0.8225
10000	95.57	0.7839	94.26	0.7901	92.00	0.8284

Tabel 5. Perbandingan Step, mAP, dan Loss

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	loss	mAP	loss
2000	95.95	0.9355	92.80	0.8449	88.10	0.8538
3000	96.62	0.8800	96.07	0.7728	91.49	0.8232
4000	96.85	0.9002	96.61	0.7573	94.46	0.8140
5000	96.90	0.9259	97.08	0.7839	94.47	0.8156
6000	97.06	0.9127	97.02	0.7719	95.52	0.8129
7000	97.21	0.9334	<b>97.83</b>	<b>0.8234</b>	96.60	0.8278
8000	97.44	0.9352	97.60	0.8285	95.96	0.8297
9000	97.49	0.9497	96.89	0.8176	96.17	0.8101
10000	<b>97.48</b>	<b>0.9453</b>	97.63	0.8430	<b>96.17</b>	<b>0.8003</b>

### 5.1.2.2 Pre-Trained Faster R-CNN dengan Learning Rate 0,0001

Pengujian ini sama seperti pengujian sebelumnya, namun *learning rate* dari masing-masing *pre-trained model* disamakan menjadi 0,0001. Hal ini bertujuan untuk membuktikan apakah *learning rate* memberikan pengaruh terhadap mAP dan kestabilan dari loss. Tabel 6,7,8 adalah hasil pelatihan FRCNN dengan Learning Rate 0,0001 pada dataset pertama, kedua, dan ketiga.

**Tabel 6. Perbandingan Step, mAP, dan Loss**

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	loss	mAP	loss
2000	77.64	0.7985	62.5	0.7006	53.77	0.804
3000	82.4	0.7395	67.79	0.6786	61.16	0.7288
4000	85.99	0.7323	74.38	0.6617	69.49	0.6948
5000	88.97	0.7287	81.12	0.6374	80.41	0.692
6000	90.62	0.705	83.88	0.6181	80.84	0.6968
7000	91.6	0.7075	85.36	0.6207	82.63	0.7165
8000	92.2	0.7061	86.03	0.6364	83.13	0.698
9000	92.67	0.6966	<b>86.15</b>	<b>0.6125</b>	<b>83.69</b>	<b>0.6918</b>
10000	<b>92.69</b>	<b>0.6927</b>	86.14	0.612	83.7	0.6954

**Tabel 7. Perbandingan Step, mAP, dan Loss**

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	loss	mAP	loss
2000	82.84	0.8815	72.94	0.8279	62.24	0.8780
3000	87.37	0.8388	76.81	0.8105	68.47	0.8921
4000	89.41	0.8129	80.06	0.7707	74.73	0.8185
5000	90.26	0.8117	82.86	0.7386	79.19	0.7868
6000	91.85	0.8140	84.25	0.7290	83.73	0.7883
7000	92.41	0.7906	86.10	0.7103	83.36	0.8025
8000	92.95	0.7862	86.62	0.7318	83.44	0.8099
9000	<b>92.97</b>	<b>0.7785</b>	<b>87.66</b>	<b>0.7095</b>	84.95	0.7932
10000	92.98	0.7823	87.66	0.7106	<b>84.96</b>	<b>0.7930</b>

**Tabel 8. Perbandingan Step, mAP, dan Loss**

Steps	FRCNN Inception V2		FRCNN Resnet 50		FRCNN Resnet 101	
	mAP	loss	mAP	loss	mAP	loss
2000	90.18	0.9735	86.45	0.9146	77.61	0.9521
3000	94.21	0.9171	90.98	0.8900	83.12	0.9115
4000	95.53	0.9037	93.30	0.8275	85.28	0.8570
5000	96.43	0.9167	93.88	0.8012	88.86	0.8277
6000	96.96	0.9063	93.81	0.7827	89.95	0.8268
7000	97.62	0.9283	94.87	0.7883	89.97	0.8036
8000	97.21	0.9187	95.01	0.7758	89.73	0.8007
9000	<b>97.33</b>	<b>0.9036</b>	95.24	0.7744	90.91	0.7905

10000	97.28	0.9042	<b>95.26</b>	<b>0.7697</b>	<b>90.97</b>	<b>0.7830</b>
-------	-------	--------	--------------	---------------	--------------	---------------

Pengujian di atas membuktikan bahwa *learning rate* mempengaruhi terhadap fluktuasi dari *loss* dan model Resnet50 serta Resnet101 memiliki mAP dibawah 90 kecuali pada dataset ketiga. Hal ini dikarenakan jumlah objek pada dataset ketiga jauh lebih sedikit dibandingkan dataset pertama maupun kedua. Dataset pertama dan kedua akan menggunakan model InceptionV2 dengan akurasi sebesar 94.22%, dataset kedua sebesar 95.59% dan dataset ketiga akan menggunakan Resnet50 dengan akurasi 97.83%.

### 5.1.3 Pengujian Terhadap Ukuran Bounding Box

Pengujian ini bertujuan untuk mencari ukuran bounding box pada kepala agar gambar yang didapatkan tidak terpotong.

**Tabel 9. Hasil Penambahan Ukuran Bounding Box (Bb)**

Bb+(Bb*N)	30%	40%	Kelemahan
Image			

Tabel 9 merupakan hasil dari penambahan ukuran terhadap *bounding box*. Dari hasil pengujian dapat ditentukan bahwa menambah 40% dari ukuran *bounding box* mampu memberikan gambar yang tidak terpotong ketika memakai APD secara keseluruhan. Metode ini memiliki kelemahan yaitu pada saat pekerja hanya memakai APD seperti masker atau kacamata maka background yang dihasilkan akan terlalu besar.

## 5.2 Pengujian Program

Pengujian ini akan menggunakan metode *confusion matrix* untuk menentukan *F1-Score*, *recall*, *accuracy*, dan *precision*. Sebanyak 219 gambar didapatkan dalam pengujian ini. Perincian *confusion matrix* dataset pertama, kedua, dan ketiga dapat dilihat pada tabel 10,11 dan 12.

**Tabel 10. Confusion Matrix Dataset Pertama**

Jenis	TP	TN	FP	FN
Lengkap	19	179	5	16
Helm	102	112	3	2
Headset	81	130	7	1
Glasses	90	107	4	18
Masker	193	11	2	13
Masker Glasses	88	103	9	19
Masker Headset	73	131	15	0
Glasses Headset	21	182	6	10
Masker Helm	94	112	6	7
Glasses Helm	21	181	7	10
Headset Helm	69	146	4	0
Masker Glasses Headset	19	185	5	10
Helm Masker Glasses	19	183	7	10
Masker Headset Helm	67	149	3	0
Glasses Headset Helm	19	189	1	10

TP = True Positive

FP = False Positive

TN = True Negative

FN = False Negative

**Tabel 11. Confusion Matrix Dataset Kedua**

Jenis	TP	TN	FP	FN
Lengkap	10	193	1	15
Helm	88	119	7	5
Headset	81	123	5	10
Glasses	94	90	10	25
Masker	181	10	21	7
Masker Glasses	88	91	12	28
Masker Headset	74	120	5	10
Glasses Headset	14	185	5	15
Masker Helm	80	120	15	4
Glasses Helm	11	183	12	13
Headset Helm	60	148	9	2
Masker Glasses Headset	12	186	4	17
Helm Masker Glasses	10	181	12	16
Masker Headset Helm	56	140	10	13
Glasses Headset Helm	12	185	6	16

TP = True Positive      FP = False Positive  
 TN = True Negative      FN = False Negative

**Tabel 12. Confusion Matrix Dataset Ketiga**

Jenis	TP	TN	FP	FN
Lengkap	3	174	0	42
Helm	92	115	8	4
Headset	77	120	7	15
Glasses	82	97	3	37
Masker	197	10	7	5
Masker Glasses	79	98	2	40
Masker Headset	70	126	7	16
Glasses Headset	4	183	8	24
Masker Helm	88	110	9	12
Glasses Helm	3	189	5	22
Headset Helm	64	151	1	4
Masker Glasses Headset	4	184	7	24
Helm Masker Glasses	3	189	4	23
Masker Headset Helm	62	152	1	4
Glasses Headset Helm	3	193	0	23

TP = True Positive      FP = False Positive  
 TN = True Negative      FN = False Negative

**Tabel 13. Hasil Kalkulasi Confusion Matrix Dataset Pertama**

Jenis	ACC	PRE	REC	F1
Lengkap	90,41	79,17	54,29	64,41
Helm	97,72	97,14	98,08	97,61
Headset	96,35	92,05	98,78	95,29
Glasses	89,95	95,74	83,33	89,11

Masker	93,15	98,97	93,69	96,26
Masker Glasses	87,21	90,72	82,24	86,27
Masker Headset	93,15	82,95	100	90,68
Glasses Headset	92,69	77,78	67,74	72,41
Masker Helm	94,06	94	93,07	93,53
Glasses Helm	92,24	75	67,74	71,19
Headset Helm	98,2	94,52	100	97,2
Masker Glasses Headset	93,15	79,17	65,52	71,70
Helm Masker Glasses	92,24	73,08	65,52	69,09
Masker Headset Helm	98,63	95,71	100	97,81
Glasses Headset Helm	94,98	95	65,52	77,55
Average	<b>93,61</b>	<b>88,07</b>	<b>82,37</b>	<b>84,67</b>

ACC (%) = Accuracy      REC (%) = Recall  
 PRE (%) = Precision      F1 (%) = F1-Score

**Tabel 14. Hasil Kalkulasi Confusion Matrix Dataset Kedua**

Jenis	ACC	PRE	REC	F1
Lengkap	92,69	90,91	40	55,56
Helm	94,52	92,63	94,62	93,62
Headset	93,15	94,19	89,01	91,53
Glasses	84,02	90,38	78,99	84,3
Masker	87,21	89,6	96,28	92,82
Masker Glasses	78,08	80	74,07	76,92
Masker Headset	92,82	93,67	88,10	90,8
Glasses Headset	90,87	73,68	48,28	58,33
Masker Helm	91,32	84,21	95,24	89,39
Glasses Helm	88,58	47,83	45,83	46,81
Headset Helm	94,98	86,96	96,77	91,6
Masker Glasses Headset	90,41	75,00	41,38	53,33
Helm Masker Glasses	87,21	45,45	38,46	41,67
Masker Headset Helm	89,5	84,85	81,16	82,96
Glasses Headset Helm	89,95	66,67	42,86	52,17
Average	<b>89,69</b>	<b>79,74</b>	<b>70,07</b>	<b>73,45</b>

ACC (%) = Accuracy      REC (%) = Recall  
 PRE (%) = Precision      F1 (%) = F1-Score

**Tabel 15. Hasil Kalkulasi Confusion Matrix Dataset Ketiga**

Jenis	ACC	PRE	REC	F1
Lengkap	80,82	100	6,67	12,5
Helm	94,52	92	95,83	93,88
Headset	89,95	91,67	83,7	87,50
Glasses	81,74	96,47	68,91	80,39
Masker	94,52	96,57	97,52	97,04
Masker Glasses	80,82	97,53	66,39	79
Masker Headset	89,5	90,91	81,4	85,89
Glasses Headset	85,39	33,33	14,29	20

Masker Helm	90,41	90,72	88	89,34
Glasses Helm	87,67	37,5	12	18,18
Headset Helm	97,73	98,46	94,12	96,24
Masker Glasses Headset	85,84	36,36	14,29	20,51
Helm Masker Glasses	87,67	42,86	11,54	18,18
Masker Headset Helm	97,72	98,41	93,94	96,12
Glasses Headset Helm	89,5	100	11,54	20,69
Average	<b>88,92</b>	<b>80,19</b>	<b>56,01</b>	<b>61,03</b>

ACC (%) = Accuracy

REC (%) = Recall

PRE (%) = Precision

F1 (%) = F1-Score

Tabel 13,14 dan 15 menunjukkan hasil kalkulasi dari *confusion matrix dataset* pertama, kedua, dan ketiga. Dalam pengujian ini dapat disimpulkan bahwa *dataset* yang tidak homogen memiliki tingkat akurasi lebih tinggi sebesar 93.61% sedangkan *dataset* kedua dan ketiga yang homogen memiliki tingkat akurasi sebesar 89.69% dan 88.92%.

## 6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- *Pre-trained model* milik YOLOv3 yang sudah dilatih ulang untuk mendeteksi hanya kepala saja mendapatkan tingkat akurasi sebesar 73.83% menggunakan dataset dari *Open Images Dataset*, dimana dataset tersebut sangat beragam.
- *You Only Look Once* (YOLO) khususnya YOLOv3 memiliki hasil yang bagus untuk mendeteksi kepala pekerja di dalam video berukuran 720x480. Akan tetapi YOLOv3 masih memiliki kelemahan yaitu susah mendeteksi saat 2 atau lebih kepala saling berhimpitan sehingga dianggap menjadi 1 *region*.
- *Faster R-CNN* (FRCNN) memiliki hasil yang cukup memuaskan dengan rata-rata dari *confusion matrix* sebesar 93.61% untuk *accuracy*, 88.07% untuk *precision*, 82.37% untuk *Recall* dan 84.67% untuk *F1-Score* nya.
- Penambahan ukuran *bounding box* sebesar 40% memiliki hasil yang baik saat pekerja memakai *safety helmet* dan *safety headset*. Hal ini juga berpengaruh terhadap pekerja yang tidak memakai *safety helmet* dan *safety headset*, karena *background* yang diambil menjadi terlalu besar sehingga saat ada pekerja yang memakai APD di belakang pekerja yang tidak memakai *safety helmet* maupun *safety headset*, pekerja yang berada di belakangnya akan ikut terdeteksi.
- Penggunaan dataset yang heterogen lebih memberikan hasil yang bagus dibandingkan dengan penggunaan dataset yang homogen. Hal ini dikarenakan penggabungan dataset dari YOLO dan Google mampu memberikan variasi yang berbeda terhadap objek yang dipelajari.

Saran untuk pengembangan kedepannya adalah:

- Penambahan *dataset* yang lebih bervariasi khususnya untuk masker, karena masker memiliki banyak perbedaan mulai dari bentuk dan warna.
- Perbanyak pelatihan untuk *glasses* dan *safety helmet* dikarenakan saat pekerja memakai keduanya secara bersamaan maka *glasses* dapat terkena bayangan dari *safety helmet* yang akhirnya membuat FRCNN gagal mendeteksi *glasses* karena terlalu gelap.

- Menguji konfigurasi lainnya untuk YOLO agar dapat meningkatkan akurasi.

## 7. DAFTAR PUSTAKA

- [1] Kementerian Tenaga Kerja dan Transmigrasi Republik Indonesia. 2010. *Peraturan menteri tenaga kerja dan transmigrasi republik indonesia nomor PER.08/MEN/VII/2010 tentang alat pelindung diri*. Jakarta: Kementerian Tenaga Kerja dan Transmigrasi Republik Indonesia.
- [2] Adam, Amir et al. 2008. *Robust real-time unusual event detection using multiple fixed-location monitors*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, 2008, pp. 555–560., doi:10.1109/tpami.2007.70825.
- [3] Rubaiyat, Abu H. M., et al. 2016. *Automatic detection of helmet uses for construction safety*. Morgantown: West Virginia University, doi: 10.1109/WIW.2016.045.
- [4] Ge, S., Li, J., Ye, Q., Luo, Z., 2017. *Detecting masked faces in the wild with LLE-CNNs*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2682-2690. doi: https://doi.org/10.1109/CVPR.2017.53.
- [5] Pradana, R. D. W., et al., 2019. *Identification system of personal protective equipment using convolutional neural network (CNN) method*. 2019 International Symposium on Electronics and Smart Devices.
- [6] Redmon, et al. 2016. *You only look once: Unified, real-time object detection*. Washington: University of Washington.
- [7] Tarwaka. 2008. *Keselamatan dan kesehatan kerja*. Surakarta: Harapan Press.
- [8] Budiono SA. 2005. *Bunga rampai hiperkes & kk*. Semarang: Universitas Negeri Diponegoro.
- [9] Mathworks. 2019. *What is object detection?* Retrieved Mei 12, 2020, from https://www.mathworks.com/discovery/object-detection.html.
- [10] Pawangfgf. n.d. *R-CNN vs Fast R-CNN vs Faster R-CNN / ML*. Retrieved Juli 1, 2020 from GeeksforGeeks: https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/.
- [11] Hamilton, H. 2018. *Computer science*. Retrieved Februari 19, 2020, from Knowledge Discovery in Databases: http://www2.cs.uregina.ca/~dbd/cs831/index.html.
- [12] Dataschool. 2014. *Simple guide to confusion matrix terminology*. Retrieved Februari 19, 2020, from dataschool: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/.
- [13] Narkhede, S. 2018. *Understanding Confusion Matrix*. Retrieved Mei 12, 2020, from Understanding Confusion Matrix: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.
- [14] Hui, Jonathan. 2018. *mAP (mean average precision) for object detection*. Retrieved Februari 20, 2020, from https://medium.com/@jonathan\_hui/map-mean-average-precision-for-object-detection-45c121a31173.
- [15] Redmon, J., & Farhadi, A. 2018. *YOLOv3: An Incremental Improvement*.