Aplikasi Framework YOLO V.3 untuk Deteksi Pelanggaran Penggunaan Masker

Marco Christopher

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav.22,Jakarta
Email: marcochristopher17@gmail.com

Abstract: *This research aims to develop a software that is able to detect violations of mask use such as wearing a mask on the chin or not covering the mouth and nose incompletely using YoloV.3. The software development method is the incremental method. This research is divided into two incremental stages. Incremental one focuses on developing a model that will be evaluated using confusion matrix. The result of incremental one is a model with a mAP value of 77.92%, precision 90.00%, recall 74.00%, and f1-score 81.00%. Incremental two focuses on building a software display which is then evaluated using the black box method. The final result of this research is software that can detect violations of mask usage.*
Keywords: *Mask,Yolo V.3,Darknet, Confusion Matrix,Inkremental Method*

Abstrak: *Penelitian ini bertujuan untuk mengembangkan sebuah perangkat lunak yang mampu mendeteksi pelanggaran penggunaan masker seperti memakai masker di dagu atau tidak menutup mulut dan hidung secara tidak sempurna dengan menggunakan YoloV.3. Metode pengembangan perangkat lunak adalah metode inkremental. Pada penelitian ini dibagi menjadi dua tahap inkremental. Inkremental satu berfokus mengembangkan model yang akan dilakukan evaluasi dengan menggunakan confusion matrix. Hasil inkremental satu adalah model dengan nilai mAP 77,92%, precision 90,00%, recall 74,00%, dan f1-score 81,00%. Inkremental dua berfokus membangun tampilan perangkat lunak yang kemudian dievaluasi menggunakan metode black box. Hasil akhir dari penelitian ini adalah perangkat lunak yang dapat mendeteksi pelanggaran penggunaan masker.*
Kata kunci: *Masker,Yolo V.3,Darknet, Confusion Matrix,Metode Inkremental*

I. PENDAHULUAN

Pada tahun 2019 terdapat virus bernama Covid-19. Virus Covid-19 merupakan virus yang menyerang bagian sistem pernapasan dan dapat menyebabkan infeksi paru-paru penderitanya. Penyebaran virus ini sangat cepat dan menyebar ke indonesia sampai sekarang. Hal ini menyebabkan masyarakat harus menggunakan masker dalam beraktifitas.

Namun seiring dengan penyebaran virus yang masih belum selesai. *World Health Organization* (WHO) menganjurkan untuk tetap menjaga protokol kesehatan. Salah satu protokol kesehatan tersebut diantaranya adalah menggunakan masker dengan benar. Penggunaan masker yang benar menurut WHO adalah masker yang dapat menutup hidung dan mulut secara penuh [1]. Di sisi lain masyarakat mulai abai dengan penggunaan masker yang benar.Tingkat kepatuhan penggunaan masker pun juga menurun [2]. Hal ini dapat menyebabkan penularan virus menjadi tidak terkendali dan dapat meningkatkan kembali kasus aktif covid-19 di Indonesia. Oleh karena itu, walaupun di indonesia sudah memberikan vaksin kepada masyarakat namun perlu diperhatikan bahwa efikasi vaksin tersebut tidak mencapai 100% [3]`. Ini berarti bahwa

belum ada vaksin yang efektif menangkal semua varian mutasi Covid-19. Sehingga masyarakat harus patuh terhadap protokol kesehatan agar kasus aktif Covid-19 tidak bertambah signifikan.

Menurut Jonathan Hui, algoritma *Yolo V.3* menunjukkan bahwa kecepatan proses frame per second algoritma yolo memiliki kecepatan proses yang lebih baik dibandingkan *Faster R-CNN, R-FCN, dan SSD* [4]. Oleh karena itu,pada peneltian ini akan mengembangkan aplikasi yang dapat mendeteksi seseorang yang tidak menggunakan masker dengan benar menggunakan algoritma pendeteksi objek *Yolo V.3*.

II. METODE PENELITIAN

A. Tinjauan Pustaka

1. Kecerdasan Buatan

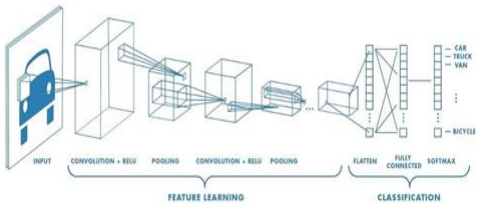
Kecerdasan buatan atau *artificial intelligence* adalah sebuah bidang yang mempelajari komputasi yang dapat mengambil keputusan secara cerdas [5]. Komputasi yang dapat mengambil sebuah keputusan secara cerdas tersebut dapat di definikan sebagai agen. Agen tidak selalu memiliki raga fisik namun dapat berbentuk program yang memiliki bentuk matematis.
Tujuan dari kecerdasan buatan adalah agen dapat mempelajari data yang ada dan memberikan prediksi keputusan berdasarkan data yang diberikan [6].

2. Machine Learning

Machine Learning atau pembelajaran mesin merupakan salah satu cabang pembelajaran dari kecerdasan buatan(AI) dan ilmu komputer. Pembelajaran mesin menggunakan data dan algoritma tertentu agar dapat meniru manusia dalam belajar dan bertindak [7]. Machine learning merupakan salah sagtu cabang dari ilmu kecerdasan buatan.Pada dasarnya machine learning merupakan proses komputer dalam
mempelajari data. Data tersebut diolah menggunakan berbagai algoritma untuk menghasilkan hasil yang terbaik. Tujuan dari machine learning adalah mempelajari pola dari contoh yang dianalisis untuk menentukan jawaban dari pertanyaan selanjutnya

3. *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan suatu jenis *neural network* yang biasanya digunakan pada data yang berbentuk citra. CNN merupakan suatu teknik yang terinspirasi dari cara mamalia menghasilkan persepsi visual. Arstitektur CNN dibagi menjadi 2 bagian besar yaitu *feature Extraction Layer* dan *Fully-Connected Layer* [8]. Pada *Feature Extraction Layer* merupakan suatu proses untuk mengekstrak gambar menjadi bentuk angka-angka. *Feature Extraction layer* dibagi menjadi dua yaitu *convolutional layer* dan pooling layer Sedangkan pada tahap *Fully-Connected Layer* merupakan lapisan yang mengubah multidimensional array menjadi sebuah vektor agar dapat diproses. Arsitektur CNN dapat di lihat pada gambar 1[9]



Gambar 1 Arsitektur CNN

4. *YOLO V.3 (You Only Look Once V3)*

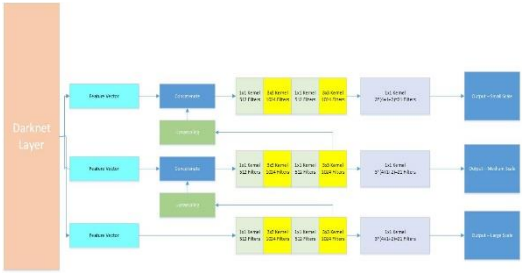
Yolo V.3 merupakan algoritma yang digunakan untuk mendeteksi objek real time. Algoritma *Yolo V.3* menggunakan pendekatan jarigan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasian sebagai objek atau bukan[10]. Penggunaan Yolo V.3 pada penelitian ini karena memiliki waktu pemrosesan yang lebih baik dibandingkan metode lain seperti CNN [4].Arsitektur *Yolo* dibagi menjadi dua bagian besar yaitu *feature extractor* atau *backbone* dan *detector*. Pada *Feature Extractor* dibangun diatas/*Darknet-53*. Hal ini yang sangat membedakan antara Yolo V1 dan V3 dimana kedalaman lapisan darknet yang sebelumnya hanya 19 lapisan menjadi 53 lapisan.

Darknet di usulkan oleh Joseph Redmon pada tahun 2018. Darknet sendiri merupakan *framework opensource* yang ditulis dengan bahasa C dan CUDA. [10]. Dengan adanya 53 lapisan ini dapat menghasilkan 1000 layer yang terhubung sehingga dapat memungkinkan arsitektur yolo dapat mendeteksi lebih banyak class. Gambar input akan melewati arsitektur darknet-53 digambarkan dalam gambar 2. kemudian akan menghasilkan tiga vektor. Tiga vektor tersebut memiliki skala kecil,menengah dan besar. Tiga vektor yang telah didapat akan dimasukkan ke dalam fungsi detector

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

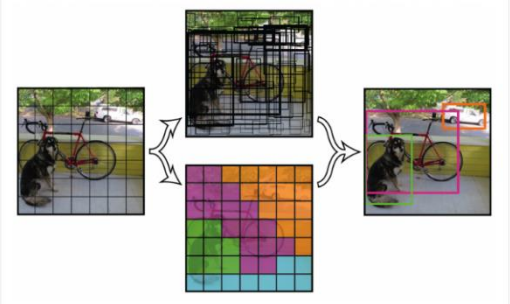
Gambar 2 Darknet-53

Pada bagian *Detector layer*, vektor yang telah didapat kemudian dimasukkan ke dalam layer bersusun yang terdiri dari 1x1 dan 3x3 kernelyang memiliki 512 dan 1024 filters yang kemudian di finalisasi dengan 1x1 kernel dengan jumlah filter yang berjumlah 3*(4+1+ jumlah class).[12]Hasil dari proses *detector layer* adalah sebuah matriks yang berskala besar, menengah dan kecil. Matriks tersebut dapat di kenal sebagai *anchor box. Anchor box* ini yang dapat menjadi *boundary box* pada proses selanjutnya.



Gambar 3 Arsitektur Yolo V.3

Gambar 3 merupakan Arsitektur *Yolo V.3* Arsitektur *Yolo* membagi suatu gambar menjadi menjadi baris dan kolom dengan jumlah yang sama. Hasil dari pembagian ini dapat kita sebut sebagai grid. Setiap grid dapat digunakan oleh beberapa *anchor box*. Akibat *Anchor box* yang banyak maka perlu dilakukan proses eliminasi. Proses eliminasi dimulai dengan melakukan pengecekan dalam anchor box tersebut apakah berisi sebuah objek [11].Terdapat beberapa cara untuk mengeliminasi *anchor box* yaitu dengan nilai skor objektivitas. Jika terdapat nilai skor objektivitas maka akan diyakini sebagai *boundary box. Boundary box* adalah sebuah kotak yang diyakini terdapat suatu objek. Setiap *boundary box* terdapat nilai *confidence score. Grid box* juga memprediksi *conditional class probability* menggunakan *Logistic Classifier.* Dari hasil pengabungan *bounding box* dan *conditional class probability* akan menghasilkan deteksi objek. Skema pemrosesan prediksi dalam yolo dapat dilihat pada gambar 4[12]

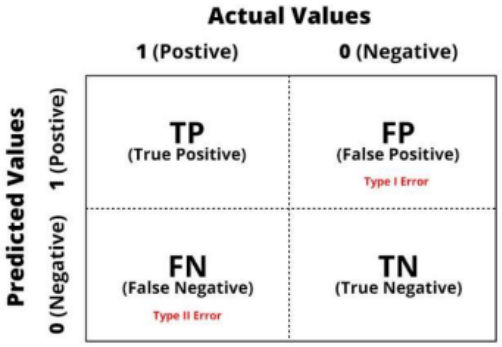


Gambar 4 Alur Pemrosesan Prediksi dalam Yolo

5. *Mean Average Precision (mAP)*

mAP merupakan salah suatu metode evaluasi yang biasanya digunakan dalam deteksi suatu objek. Algoritma pendeteksian objek yang menggunakan evaluasi ini diantaranya *Faster R-CNN,MobileNet SSD*, dan *YOLO*. *Mean Average Precision* merupakan nilai rata rata dari *Average Precision* setiap kelas deteksi[13]. *Average Precision* adalah sebuah nilai rata-rata yang melambangkan akurasi prediksi model dalam setiap satu kelas prediksi. Untuk menghitung Average Precision dibutuhkan nilai precision dan recall. Nilai precision dan recall dapat dihitung menggunakan confusion matrix.

Pada gambar 5 *Confusion matrix* berisi 4 komponen diantaranya *True posisitve, False positive,True negative, FalseNegative.*



Gambar 5 Confusion Matrix

Precision sendiri mengambarkan seberapa akurat model dalam mendeteksi class dari semua deteksi yang telah dilakukan. Rumus dari Precision adalah sebagai berikut.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

Selanjutnya diperlukan sebuah nilai Recall yaitu sebuah nilai yang mengambarkan rasio prediksi benar positif terhadap semua data yang benar positif. Rumus dari Recall digambarkan sebagai berikut .

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

F1-Score merupakan rata-rata dari precision dan recall. Rumus pada F1-Score adalah sebagai berikut[34]:

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (3)$$

Setelah mendapat nilai *precision* dan *recall* maka akan di petakan dalam grafik *AP* dapat dihitung dengan persamaan sebagai berikut:

$$AP = \sum_{K=0}^{K=N-1} [Recalls(k) - Recalls(k+1)] * Precisions(k) \quad (4)$$

Maka rumus mAP adalah menghitung nilai rata-rata AP dalam semua kelas deteksi.

Rumus mAP digambarkan sebagai berikut.

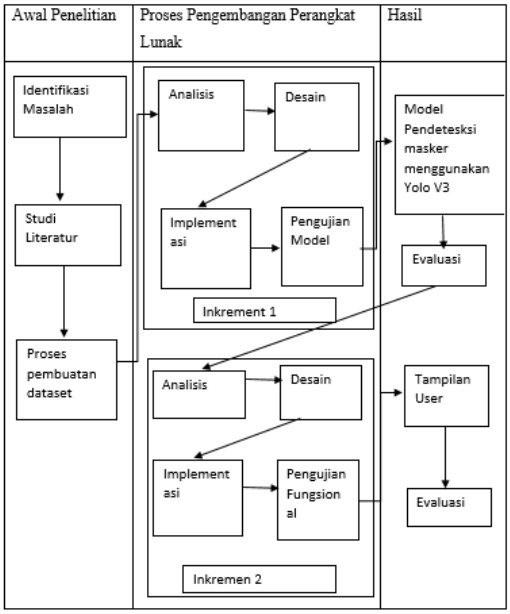
$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (5)$$

B. Kerangka Penelitian

Berdasarkan masalah yang ada dimana tingkat kepatuhan masyarakat dalam menggunakan masker semakin menurun. Pengawasan secara manual sering kali tidak efektif dan cepat. Oleh sebab itu diperlukan sebuah aplikasi yang dapat mendeteksi pelanggaran penggunaan masker. Pada tahap awal,peneltian dimulai dengan mengumpulkan gambar wajah manusia yang menggunakan masker. Gambar wajah manusia akan digunakan sebagai dataset. Gambar wajah manusia akan diklasifikasikan menjadi dua kelas yaitu penggunaan masker yang benar, dan penggunaan masker yang salah.Dataset akan digunakan untuk pembelajaran model dengan bantuan *framework darknet* menggunakan algoritma *Yolo V.3*. Model yang dihasilkan akan dilakukan visualisasi dalam perangkat lunak *desktop*. Hasil yang diharapkan dari penelitian ini adalah aplikasi dapat mendeteksi penggunaan masker yang tidak benar.

1. Proses Penelitian

Pada bagian ini akan dijelaskan tahapan – tahapan yang akan dilakukan dalam pengembangan aplikasi mulai dari awal penelitian sampai hasil penelitian. Gambar 7 merupakan alur pengembangan aplikasi dalam penelitian ini



Gambar 6 Proses Penelitian

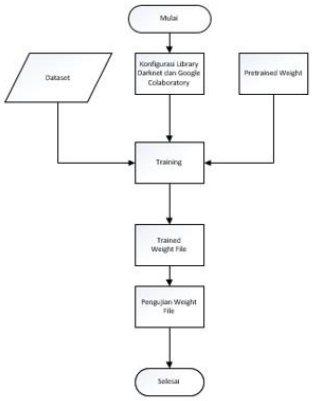
Dari proses penelitian pada gambar 6 . rancangan pengembangan aplikasi dimulai dari identifikasi masalah dan studi literatur. Pada proses pra penelitian , peneliti akan melakukan identifikasi masalah dan mencari studi literatur untuk mendukung proses penelitian yang dilakukan. Masalah yang akan dibahas dalam penelitian ini adalah perancangan aplikasi yang dapat mendeteksi seseorang yang tidak menggunakan masker dengan benar menggunakan algoritma pendeteksi objek *Yolo V.3*. Pengumpulan data yaitu gambar manusia yang dibagi menjadi dua kelas yaitu memakai masker dengan benar dan pelanggaran dengan tidak menggunakan masker atau masker yang tidak menutupi mulut dan hidung. Pada proses pengembangan sistem akan dibagi menjadi dua inkremen. Pada inkremen pertama bertujuan untuk membangun model yang akan digunakan aplikasi dengan menggunakan algoritma *Yolo v.3.* Untuk

mengukur keberhasilan model dilakukan evaluasi apakah model dapat mendeteksi masker yang ada pada gambar manusia.Setelah inkremen pertama selesai maka akan dilanjutkan ke tahap inkremen dua yaitu membuat tampilan aplikasi untuk pengguna.Pembangunan tampilan aplikasi akan berbasis *desktop*. Kemudian dilakukan uji *fungsionalitas* aplikasi menggunakan metode blackbox .

2. Proses Pembuatan Dataset

Proses Pembuatan dataset dimulai dengan mengunduh *dataset* yang bernama *face mask detection* pada situs *kaggle*. Pada awalnya *dataset* yang digunakan memiliki 853 gambar dengan tiga kelas yang berbeda. Pada penelitian kali ini akan diubah menjadi dua kelas yaitu kelas pemakaian masker dengan benar dan pelanggaran penggunaan masker. Kemudian dataset di tambahkan 200 gambar acak yang masing-masing penambahannya adalah 100 gambar untuk tiap kelasnya.

3. Inkremen Satu



Gambar 7 Inkremen Satu

dataset akan dibagi menjadi 85% gambar data latih dan 15%gambar sebagai data uji. Data tersebut dimasukkan dalam library Yolo V.3.

Kemudian dilakukan beberapa konfigurasi dimana setiap konfigurasi dalam Tabel 1 memiliki nilai *epoch* dan *batch* yang sama.

Tabel 1 Konfigurasi dari setiap model

Model	Width	Height	Batch	Epoch
Model 1	416	416	64	2000
Model 2	480	480	64	2000
Model 3	512	512	64	2000

Keluaran dari pembelajaran library *Yolo V.3* akan menghasilkan model. Kemudian model akan di evaluasi menggunakan nilai *precision*,*recall* dan *mAP*.

3. Inkremen Dua

Gambar 9 Inkremen Dua

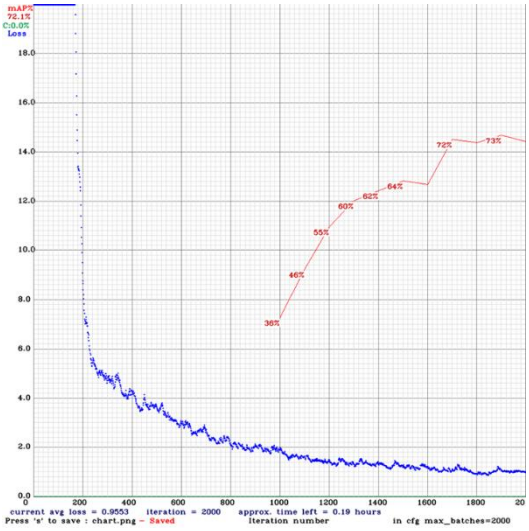
Pada tahap analisis inkremen kedua berfokus pada pembangunan tampilan antarmuka aplikasi dan integrasi model ke dalam aplikasi. Aplikasi ini diharapkan dapat menampilkan jumlah pelanggaran masyarakat yang tidak menggunakan masker atau menggunakan masker secara tidak benar. Tampilan *GUI* akan dibangun berdasarkan bahasa *python* dengan bantuan library *PySimpleGUI* dan *OpenCV* untuk membantu melakukan *processing* gambar.

Aplikasi yang dibuat dapat menerima dua jenis masukkan. Pertama adalah gambar yang berjenis *JPG* atau *JPEG*. Kedua adalah video yang berasal dari webcam. Setelah pengguna memberi masukkan. Program akan menganalisis semua wajah manusia. Jika terdapat wajah manusia yang tidak menggunakan masker atau tidak menggunakan masker maka aplikasi akan menghitung pelanggaran dan memvisualisasikan ke dalam tampilan.

III. HASIL DAN PEMBAHASAN

A. Hasil Inkremen Satu

1. Hasil dengan Model Konfigurasi 1



Gambar 10 Grafik Akurasi mAP Yolo v3 Model 1

Berdasarkan grafik yang ada pada pembuatan model demgan konfigurasi 3, Nilai *mAP* tertinggi pada pelatihan adalah sebesar 77.92 %. Dengan rata rata *loss* saat pelatihan model adalah sebesar 0.9672.

•Nilai *Average Precision* pada kelas good_mask atau berarti model berhasil mendeteksi penggunaan masker sesuai standar adalah sebesar 83.61 %

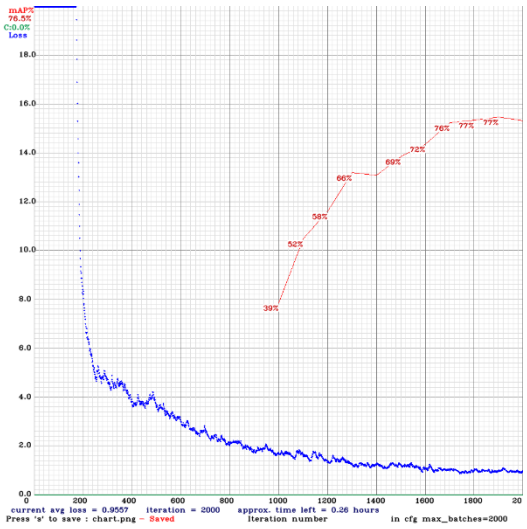
•Nilai *Average Precision* pada kelas bad_mask atau berarti model berhasil mendeteksi pelanggaran masker sebesar 60.51 %

•Gabungan Average Precision pada kedua kelas deteksi menghasilkan nilai *mAP* adalah sebesar 72.06 %

•Dengan nilai *confidence score* sebesar 0.25 menghasilkan nilai *precision* sebesar 0.88, *recall* sebesar 0.72 dan *F1-score* sebesar 0.80

•Model memerlukan waktu sekitar 7 detik dalam melakukan 1 deteksi.

2. Hasil dengan Model Konfigurasi 2



Gambar 11 Grafik Akurasi mAP Yolo v3 Model 2

Berdasarkan grafik yang ada pada pembuatan model demgan konfigurasi 2, Nilai *mAP* tertinggi pada pelatihan adalah sebesar 76.52 %. Dengan rata rata *loss* saat pelatihan model adalah sebesar 0.9557.

•Nilai *Average Precision* pada kelas *good_mask* atau berarti model berhasil mendeteksi penggunaan masker sesuai standar adalah sebesar 86.25 %

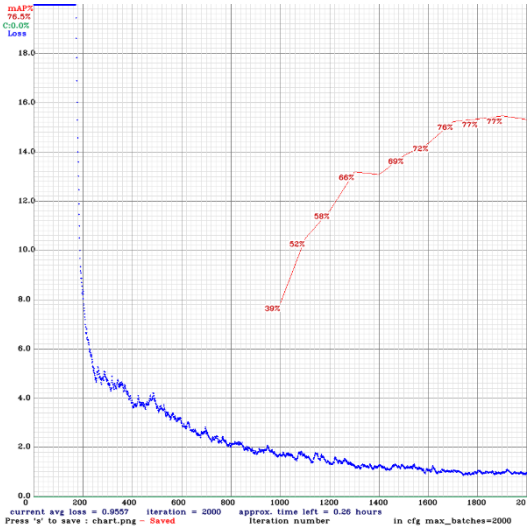
•Nilai *Average Precision* pada kelas *bad_mask* atau berarti model berhasil mendeteksi pelanggaran masker sebesar 66.79 %

•Gabungan *Average Precision* pada kedua kelas deteksi menghasilkan nilai *mAP* adalah sebesar 76.52 %

•Dengan nilai *confidence score* sebesar 0.25 menghasilkan nilai *precision* sebesar 0.89, *recall* sebesar 0.73 dan *F1-score* sebesar 0.81

•Model memerlukan waktu sekitar 7 detik dalam melakukan 1 deteksi.

3. Hasil dengan Model Konfigurasi 3



Gambar 12 Grafik Akurasi mAP Yolo v3 Model 3

Berdasarkan grafik yang ada pada pembuatan model demgan konfigurasi 2, Nilai mAP tertinggi pada pelatihan adalah sebesar 77.92 %. Dengan rata rata loss saat pelatihan model adalah sebesar 0.9672.

•Nilai *Average Precision* pada kelas good_mask atau berarti model berhasil mendeteksi penggunaan masker sesuai standar adalah sebesar 86.15 %

•Nilai *Average Precision* pada kelas bad_mask atau berarti model berhasil mendeteksi pelanggaran masker sebesar 69.70 %

•Gabungan *Average Precision* pada kedua kelas deteksi menghasilkan nilai mAP adalah sebesar 77.92 %

•Dengan nilai *confidence score* sebesar 0.25 menghasilkan nilai *precision* sebesar 0.90, *recall* sebesar 0.74 dan F1-score sebesar 0.81

•Model memerlukan waktu sekitar 7 detik dalam melakukan 1 deteksi.

4.Pembahasan

Berdasarkan pembelajaran model *Yolo* yang telah dibuat,penggunaan *width* dan *height* yang berbeda pada konfigurasi library darknet akan mempengaruhi tingkat average precision. Pada model dengan konfigurasi 1 memiliki *width* dan *height* sebesar *416 pixel* menghasilkan *average precision* pada kelas *bad_mask* sebesar 60,51%. Pada model konfigurasi kedua digunakan penambahan ukuran *width* dan *height* sebesar 64 pixel menjadi 480 pixel. Hasil dari konfigurasi ke 2 adalah *average precision* setiap kelas deteksi meningkat. Pada model konfigurasi 2 kelas *bad_mask* nilai *average precision* meningkat sebesar 66,79% . Pada model dengan konfigurasi 3 memiliki *width* dan *height* sebesar *512 pixel* menghasilkan *average precision* pada kelas *bad_mask* sebesar 69,70%.

Pada tabel 1 merupakan rangkuman dari pengujian pada inkremental satu menggunakan beberapa parameter nilai *mAP,precision, recall* dan *F1-score*. Berdasarkan hasil pengujian yang ada, model dengan konfigurasi 3 memiliki nilai yang paling optimal. Sehingga model dengan konfigurasi 3 akan digunakan pada tahapan inkremental kedua. Pada tabel 2 merupakan ringkasan hasil evaluasi dari setiap konfigurasi model.

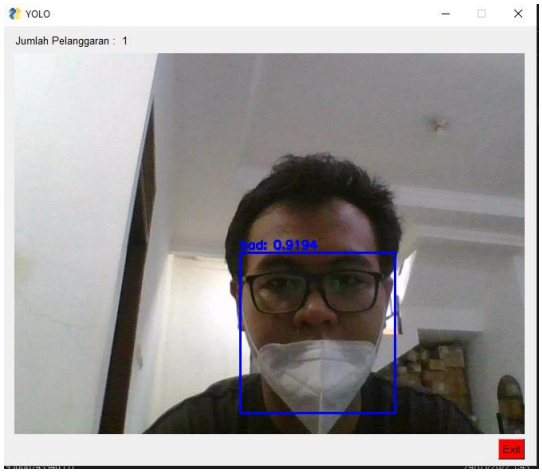
Tabel 2 Rangkuman Hasil Evaluasi dari setiap model

Model	mAP	Precesion	Recall	F1-Score
Model 1	72,06%	0,88	0,72	0,81
Model 2	76,52%	0,89	0,73	0,81
Model 3	77,92%	0,90	0,74	0,81

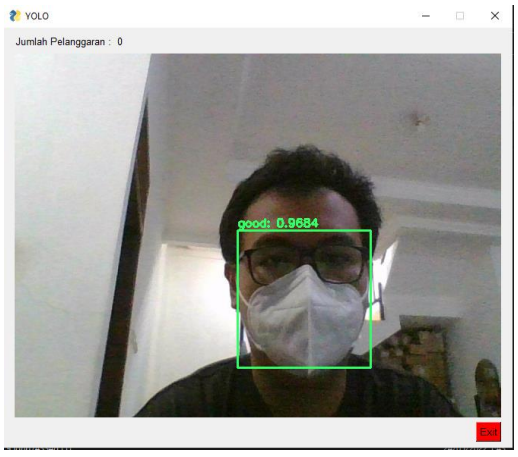
B. Hasil Inkremen Dua

Pada gambar 13, merupakan gambar hasil program, saat berhasil mendeteksi pelanggaran masker yang kemudian

ditampilkan ke dalam aplikasi. Jumlah pelanggaran tersebut dihitung dan ditampilkan diatas gambar hasil deteksi. Sedangkan pada gambar 14 merupakan gambar hasil program saat berhasil mendeteksi tidak adanya pelanggaran. Maka jumlah pelanggaran akan bernilai 0 pada tampilan aplikasi.



Gambar 13 Tampilan Aplikasi saat pelanggaran



Gambar 14 Tampilan Aplikasi saat tidak ada pelanggaran

IV. SIMPULAN

Berdasarkan hasil dari penelitian ini dapat disimpulkan sebagai berikut:

1. Metode Yolo V.3 dapat digunakan untuk melakukan pendeteksian masker.
2. Meningkatkan *width* dan *height* pada konfigurasi *darknet* dapat meningkatkan hasil *mAP* pada model
3. Hasil *mAP* terbaik ada pada model dengan konfigurasi 3 yaitu sebesar 77,92% . Kelas *good_mask* mendapat nilai *AP* sebesar 86,15% dan kelas bad_mask mendapat nilai *AP* sebesar 69,75%.

DAFTAR RUJUKANsa

[1] Pittara, "Virus Corona," 2022. https://www.alodokter.com/virus-corona (accessed Feb. 08, 2022).

[2] H. Prasetyo, "Tingkat kepatuhan memakai masker dan jaga jarak menurun," 2021. https://kesehatan.kontan.co.id/news/tingkat-kepatuhan-memakai-masker-dan-jaga-jarak-menurun?page=all (accessed Nov. 12, 2021).

[3] T. Septiana, "Mengapa harus tetap disiplin prokes meski sudah divaksin Covid-19? Ini penjelasannya," *kesehatan.kontan.co.id*, 2021. https://kesehatan.kontan.co.id/news/mengapa-harus-tetap-disiplin-prokes-meski-sudah-divaksin-covid-19-ini-penjelasannya (accessed Nov. 12, 2021).

[4]J. Hui, "Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)," *Medium.com*, 2018. https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359 (accessed Nov. 12, 2021).

[5] I. Goodfellow, "Deep Learning."

[6] J. Wira and G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," vol. 4, 2020.

[7] IBM Cloud Education, "Machine Learning," *ibm.com*, 2020. https://www.ibm.com/cloud/learn/machine-learning (accessed Feb. 16, 2022).

[8] Q. LINA, "Apa itu Convolutional Neural Network?," *Medium.com*, 2019. https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4 (accessed Feb. 16, 2022).

[9] A. Yanuar, "YOLO (you only look once)," 2018. https://machinelearning.mipa.ugm.ac.id/2018/08/05/yolo-you-only-look-once/ (accessed Nov. 12, 2021).
[10] J. Redmon, "Darknet: Open Source Neural Networks in C," 2016. https://pjreddie.com/darknet/ (accessed Feb. 16, 2022).
[11] E. Y. Li, "Dive Really Deep into YOLO v3: A Beginner's Guide," *Towards Data Science*, 2019. https://towardsdatascience.com/dive-really-deep-into-yolo-v3-a-beginners-guide-9e3d2666280e.
[12] M. Hollemans, "Real-time object detection with YOLO," *machinethink.net*, 2017. https://machinethink.net/blog/object-detection-with-yolo/ (accessed Feb. 23, 2022).
[13] S. Yohanandan, "mAP (mean Average Precision) might confuse you!" https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2.