

Identifikasi Citra Penggunaan Masker Secara *Real-Time* dengan Arsitektur CNN pada Metode YOLO

1st Aldo Zamaludin FernandoUniversitas Buana Peruangan Karawang
Karawang, Indonesia
if18.aldofernando@mhs.ubpkarawang.ac.id2nd Adi Rizky PratamaUniversitas Buana Peruangan Karawang
Karawang, Indonesia
adi.rizky@ubpkarawang.ac.id3rd Ayu Ratna JuwitaUniversitas Buana Peruangan Karawang
Karawang, Indonesia
ayurj@ubpkarawang.ac.id

Abstract— Dalam masa pandemi masih banyak orang yang kurang tertib dalam mematuhi protokol kesehatan terutama dalam penggunaan masker baik itu aktivitas didalam ruangan maupun diluar ruangan. Oleh karena itu diperlukan sebuah sistem yang dapat mendeteksi atau mengidentifikasi sebuah objek berupa penggunaan masker. Dalam penelitian ini menggunakan Metode YOLO yang memiliki arsitektur dari Algoritma *Convolutional Neural Network* (CNN), dengan menggunakan darknet53 untuk men *training* model pada metode YOLO agar dapat mendeteksi objek secara *real-time*. Dalam mengidentifikasi objek menggunakan metode YOLO terhadap penggunaan masker, sistem dapat berjalan dan dapat mendeteksi objek dengan cukup akurat Dengan mendapatkan nilai *precision* sebesar 74%, nilai *recall* memiliki 50%, dan tingkat akurasi sebesar 74%. sistem dapat dikatakan berhasil dalam mengidentifikasi pemakaian masker dengan menggunakan metode YOLO dengan tingkat keakurasi cukup baik.

Kata Kunci : *Convolutional Neural Network*, YOLO, Deteksi Objek, Citra Digital, Masker.

I. PENDAHULUAN

Dalam kurun waktu dua tahun dimasa pandemi ini masih banyak kasus yang terpapar COVID-19, Dibatasinya mobilitas dan mencegah kerumunan di tempat terbuka serta selalu menerapkan 3M, memakai masker dan mencuci tangan dengan sabun[1]. diharapkan dapat memutus rantai penularan virus COVID-19 ini. Baru-baru ini vaksin COVID-19 sudah di temukan dan sudah banyak yang di suntik vaksin untuk mencegah terpapar dari COVID-19, Meski begitu kita juga harus tetap mematuhi protokol kesehatan yang ada agar terhindar dari orang-orang yang terinfeksi tanpa adanya gejala. Oleh karena itu, di harapkan dengan menerapkan *Physical Distancing* dan penggunaan masker sebelum memasuki ruangan dan berpergian dapat menghindari penyebaran COVID-19. Meskipun masih banyak masyarakat yang abai akan protokol kesehatan dengan tidak menggunakan masker saat beraktivitas

Maka dari itu untuk mencegah penyebaran COVID-19 di era *New Normal* ini peneliti akan membuat sebuah system untuk mendeteksi atau mengidentifikasi penggunaan masker dengan menggunakan metode *Algoritma YOLO*. Metode *Algoritma YOLO* merupakan salah satu *Machine Learning* yang sering kali digunakan sebagai pondasi untuk mendeteksi atau mengenali area wajah dalam bentuk gambar atau video. Metode ini digunakan untuk mengenali area mana saja yang terdeteksi wajah manusia (*Face Recognition*) pada sebuah gambar dengan menggunakan OpenCV. Dengan objek deteksi menggunakan masker, tidak menggunakan masker dengan benar, dan tidak menggunakan masker.

Dengan menggunakan arsitektur CNN pada metode Yolo ini sebagai metode yang digunakan untuk mendeteksi objek penggunaan masker. *Convolutional Neural Network* (CNN) merupakan salah satu algoritma yang paling sering digunakan dalam mendeteksi objek[2], arsitektur CNN ini terbilang arsitektur yang cepat dan akurat dalam mendeteksi objek. Metode ini dapat mendeteksi secara *real-time*. Dalam penelitian ini akan mengidentifikasi penggunaan masker dengan menggunakan metode YOLO yang menggunakan darknet53 sebagai metode ekstrasi untuk memproses citra, darknet sendiri memiliki 53 lapisan konvolusional Darknet53 digunakan untuk melatih metode YOLO ini agar dapat mendeteksi penggunaan masker secara *real-time*. Dengan menerapkan *Image Processing* dan membuat sebuah sistem yang dapat mendeteksi penggunaan masker.

II. TINJAUAN PUSTAKA

A. Masker

Masker, juga dikenal sebagai pelindung pernapasan, adalah teknik yang digunakan untuk melindungi orang dari kontaminasi luar seperti zat berbahaya atau ozon. Untuk mencegah terjadinya penyakit pada saluran pernafasan, penggunaan masker ini digunakan untuk melindungi penggunanya. Masker banyak dari mereka digunakan untuk memberikan perlindungan terhadap aerosol dan polusi partikel. Ini dapat menyebabkan sistem pernapasan buatan manusia menjadi tersumbat. Mereka yang tidak menggunakan alat penjamin mandiri, menghilangkan debu dan aerosol dari berbagai sumber Dimensi dan sifat kimia yang berbeda mungkin bermanfaat bagi manusia.

B. Identifikasi

Identifikasi adalah proses mencari, memperoleh, mengumpulkan, mendaftarkan, meneliti, dan menangkap fakta dan informasi dari “kebutuhan” lapangan. Dalam hal ini, kebutuhan dapat dibagi menjadi dua kategori: kebutuhan yang dirasakan yang secara jelas didefinisikan sebagai kebutuhan yang mendesak dan kebutuhan yang tidak diantisipasi yang secara jelas didefinisikan sebagai tidak mendesak.

C. Citra Digital

Citra Digital merupakan sebuah matriks dimana indeks pada baris maupun kolomnya yang mendeklarasikan pada sebuah titik Citra tersebut dan elemen matriks[3], yang dikenal juga sebagai elemen gambar ataupun *pixel* yang menyatakan tingkat keabuan pada titik tersebut. Matriks yang dapat dinyatakan untuk Citra Digital adalah matriks yang berukuran N (baris/tinggi) $\times M$ (kolom/lebar).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, M - 1) \\ f(1,0) & f(1,1) & \cdots & f(1, M - 1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N - 1,0) & f(N - 1,1) & \cdots & f(N - 1, M - 1) \end{bmatrix}$$

Keterangan :
 N = jumlah baris $0 = y = N-1$
 M = jumlah kolom $0 = x = M - 1$
 L = maksimal warna intensitas $0 = f(x,y) = L-1$

D. You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan sebuah algoritma yang dikembangkan untuk dapat melakukan deteksi objek secara *real-time* menggunakan pendekatan yang berbeda dengan algoritma yang sebelumnya. Algoritma YOLO menggunakan pendekatan dengan cara menerapkan jaringan saraf tunggal pada keseluruhan gambar. Pada jaringan ini gambar akan dibagi menjadi wilayah-wilayah yang kemudian akan memprediksi kotak pembatas dan probalitas[4]

E. Convolutional Neural Network (CNN)

Convolutional neural network merupakan salah satu dari neural network yang sering digunakan dalam mengolah data citra. CNN dapat digunakan dalam mendekripsi dan mengenali objek pada sebuah citra. CNN merupakan teknik yang terinspirasi dari sistem penglihatan mamalia-manusia, yang menghasilkan persepsi visual. CNN yang terdiri dari susunan neuron yang mempunyai weight, bias dan activation function. Convolutional layer juga terdiri dari susunan neuron yang sedemikian rupa maka dari itu membentuk sebuah *filter* yang memiliki panjang dan tinggi. CNN Pada dasarnya merupakan neural network dengan memiliki banyak layer[5]

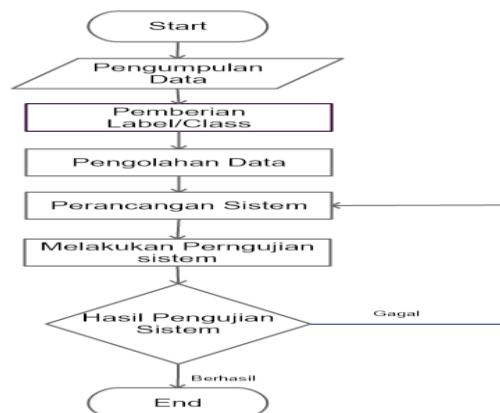
III. HASIL DAN PEMBAHASAN.

A. Hasil Penelitian

Hasil dari penelitian yang telah dilakukan berdasarkan dari hasil uji coba yang dilakukan dengan menggunakan arsitektur CNN yang terdapat pada Model YOLO. Dengan melakukan pengujian system terhadap pendekripsi penggunaan masker yang telah dilakukan untuk menguji seberapa akurat objek tersebut terdeteksi. Dengan menggunakan model pelatihan menggunakan darknet53 untuk tahapan *training* data.

B. Prosedur Penelitian

Pada penelitian yang dilakukan ada beberapa tahapan yang dilakukan untuk memulai proses penelitian pada gambar 1 yang akan dijelaskan tahapan penelitian untuk mendapatkan hasil penelitian yang dilakukan.:



Gambar 1 Prosedur Penelitian

C. Pengumpulan data

Dalam proses ini mengumpulkan data untuk membantu dalam proses penelitian dengan mengumpulkan gambar kemudian dimasukan kedalam dataset untuk kemudian diolah dengan pengolahan citra digital. Data yang akan diambil berupa gambar dengan dibagi kedalam tiga kelas

D. Anotasi Data Atau Pelabelan Data

Sebelum data dapat digunakan untuk melakukan sebuah proses deteksi objek agar dapat melakukan pelatihan pada algoritma dan dapat digunakan untuk mendeteksi sebuah objek. Pertama perlu memasukan sebuah informasi terhadap algoritma yang digunakan agar dapat memproses dan mengirimkan *output* dan *inferensi*. Dengan melakukan anotasi data agar AI dapat mengetahui apakan data yang di inputkan berupa audio, gambar, video, atau teks. Tergantung pada fungsionalitas dan parameter yang ditetapkan, setelah melakukan antotasi data dan model telah ditetapkan kemudian akan mengelasifikasi data dan melanjutkan dengan menjalankan tugasnya. Dalam pelabelan data ini memiliki 3 *class* antara lain sebagai berikut, 1 = *mask* 2 = *badmask* 3 = *nomask*,

Berikut merupakan proses pelabelan citra yang dilakukan sebagai berikut:



Gambar 2 Proses Labeling citra

Setelah melakukan anotasi data atau pelabelan citra didapatkan file txt yang berisi kotak pembatas atau *bounding box* untuk melakukan deteksi dan uji coba pada model yolo berikut hasil labeling pada citra:

0 0.323437 0.555208 0.443750 0.697917

Gambar 3 Hasil Labeling Citra

Pada setiap barisnya terdapat 5 kolom yang merupakan, id *class* dan kotak pembatas dengan memiliki format kordinat pada kotak pembatas id *class*, kordinat x, kordinat y, kordinat w (*width/ lebar*) dan kordinat h (*height/tinggi*).

E. Pengolahan dataset

Pada tahapan selanjutnya melakukan proses *preprocessing* pada dataset yang telah di beri label untuk diolah untuk kemudian data di *training*. Dataset yang sebelumnya telah di anotasi atau diberi label pada setiap *class* disini memiliki 3 *class* terdiri dari *class mask* dengan 697 data, *class badmask* dengan 221, dan *class nomask* dengan 524 data dan total data foto yang digunakan berjumlah 1.422 sebagai dataset untuk diolah. Kemudian dataset dibagi kedalam beberapa bagian antaralain data untuk *training*, data untuk *validation*, dan data untuk *testing*.

1. *Training Set* dengan jumlah data = 991 *Images*
2. *Validation Set* dengan jumlah data = 287 *Images*
3. *Testing Set* dengan jumlah data = 300 *Images*

Selanjutnya tahapan *preprocessing* untuk mengubah gambar berdasarkan data yang dibutuhkan. Disini saya merubah *reize images* menjadi ukuran 416x416 dan *color* pada gambar diubah kedalam *grayscale* atau menjadi keabuan pada seluruh dataset yang digunakan. Berikut merupakan hasil yang di peroleh setelah melakukan *preprocessing* pada gambar berikut:

Hasil dari proses *resize* pada citra.



Gambar 4 Hasil *Resize Images*

Hasil dari proses ke *grayscale*



Gambar 5 Hasil merubah ke *Grayscale*

Setelah mementukan jumlah foto yang akan digunakan pada dataset, dataset telah selesai di *preprocessing* dan siap digunakan untuk ketahapan proses *Training*. Dan berikut hasil dataset yang telah melalui proses *preprocessing*.



Gambar 6 Hasil *Preprocessing* Dataset

F. Training

Pada proses pelatihan data sebelum dapat digunakan melakukan pelatihan terlebih dahulu pada dataset yang digunakan menggunakan darknet untuk melatih model deteksi objek dengan menggunakan algoritma yolo untuk mendapatkan model pelatihan terbaru sesuai deteksi objek yang diinginkan, algoritma yolo yang akan digunakan menggunakan versi yolov3 untuk melatih model deteksi objek tersebut. Dalam pelatihan ini menggunakan google colab untuk melatih data yang akan digunakan. Pertama melakukan konfigurasi terhadap model yang akan digunakan seperti pada gambar sebagai berikut:

```
!cp cfg/yolov3-tiny.cfg cfg/yolov3_training.cfg
!sed -i 's/batch=1/batch=64/' cfg/yolov3_training.cfg
!sed -i 's/subdivisions=1/subdivisions=16/' cfg/yolov3_training.cfg
!sed -i 's/max_batches = 500200/max_batches = 20000/' cfg/yolov3_training.cfg
!sed -i '610 s@classes=80@classes=3@' cfg/yolov3_training.cfg
!sed -i '696 s@classes=80@classes=3@' cfg/yolov3_training.cfg
!sed -i '783 s@classes=80@classes=3@' cfg/yolov3_training.cfg
!sed -i '603 s@filters=255@filters=24@' cfg/yolov3_training.cfg
!sed -i '689 s@filters=255@filters=24@' cfg/yolov3_training.cfg
!sed -i '776 s@filters=255@filters=24@' cfg/yolov3_training.cfg
```

Gambar 7 Mengatur Konfigurasi

Kemudian melanjutkan pelatihan model dengan konfigurasi yang telah diubah tadi sesuai kebutuhan pelatihan. Kemudian memulai pelatihan dengan memasukan hasil konfigurasi tadi yang berada di deriktori darknet berikut merupakan gambar proses *training* model yang dilakukan menggunakan darknet.

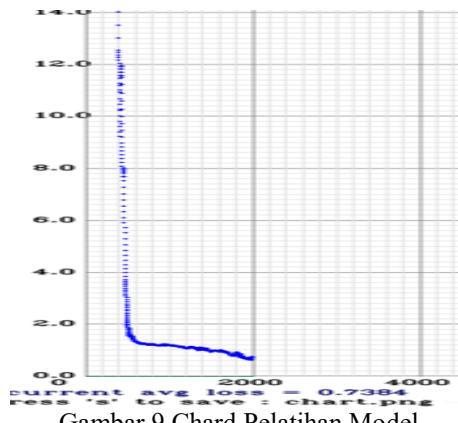
```

1978: 0.661567, 0.678546 avg loss, 0.001000 rate, 0.621033 seconds, 126592 images, 4.355006 hours left
Loaded: 0.026858 seconds
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.688688), count: 5, class_loss = 1.612294, iou_loss = 0.488063,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.000000), count: 1, class_loss = 0.000012, iou_loss = 0.000000,
total_bbox = 125265, rewritten_bbox = 0.000000 %
Can't open label file. (This can be normal only if you use MSCOCO): data/obj/datasetv2/M 165.txt
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.804828), count: 4, class_loss = 0.867168, iou_loss = 0.199137,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.000000), count: 1, class_loss = 0.000020, iou_loss = 0.000000,
total_bbox = 125269, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.847121), count: 4, class_loss = 0.985747, iou_loss = 0.110959,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.000000), count: 1, class_loss = 0.000010, iou_loss = 0.000000,
total_bbox = 125273, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.784012), count: 3, class_loss = 0.726857, iou_loss = 0.158294,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.484470), count: 1, class_loss = 0.394883, iou_loss = 0.240004,
total_bbox = 125277, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.614649), count: 4, class_loss = 1.546136, iou_loss = 0.546886,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.319593), count: 1, class_loss = 0.418218, iou_loss = 0.470546,
total_bbox = 125282, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.711753), count: 4, class_loss = 0.584459, iou_loss = 0.331252,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.000000), count: 1, class_loss = 0.000011, iou_loss = 0.000000,
total_bbox = 125286, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 16 Avg (IOU: 0.668889), count: 4, class_loss = 1.008243, iou_loss = 0.389088,
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.000000), count: 1, class_loss = 0.000024, iou_loss = 0.000000,
total_bbox = 125290, rewritten_bbox = 0.000000 %

```

Gambar 8 Proses Pelatihan Model

Setelah proses pelatihan model dan mendapatkan nilai *avg loss* nya sebesar 0.678546 hasil tersebut cukup bagus dalam hasil pelatihan berikut merupakan gambar hasil dari proses *training* yang di hentikan pada batch 2000.



Gambar 9 Chard Pelatihan Model

Setelah proses pelatihan model selesai maka didapatkan *file model* yang baru sesuai dengan pelatihan model yang di buat dengan *file* yang berbentuk *format weights*. Merupakan *file* model untuk mendeteksi objek.

G. Preprocessing

Dalam proses *processing* disini menggunakan ekstrasi pengubahan warna pada RGB dengan menggunakan HSV untuk merubah warna pada citra yang akan dideteksi berikut merupakan hasil *processing ekstrasi* menggunakan HSV. Dengan ekstrasi citra warna pada objek yang di deteksi dengan menggunakan ambang batas sebesar 100 treshold di dapatkan hasil sebagai berikut :

Tabel 1 Hasil Preprocessing Class Mask

Original	Grayscale	Biner (black and white)

Tabel 2 Hasil Preprocessing Class Nomask

Original	Grayscale	Biner (black and white)

Tabel 3 Hasil Preprocessing Class Badmask

	Original	Grayscale	Biner (black and white)
			

H. Uji Coba Sistem

Dalam tahapan ini akan melakukan pengujian sistem yang telah dibuat serta menguji coba model yang telah dilatih, apakah model dapat berjalan dan dapat mendeteksi sebuah objek apakah tidak, serta mengukur nilai *confidencenya* atau tingkat kepercayaan model pada objek yang dideteksi. Berikut dari hasil uji coba system yang dimasukan kedalam tabel sebagai berikut:

Tabel 4 Hasil Uji Coba

No	Hasil Deteksi	Mask	Badmask	Nomask	Keterangan	Akurasi	Sesuai
1		Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Masker Medis Nilai <i>Confidance</i> = 0.7068 Terdeteksi menggunakan masker	70.68%	Sesuai
2		Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Masker Tidak Rapih Nilai <i>Confidence</i> = 0.5367 Tidak Terdeteksi Badmask	53.67%	Tidak Sesuai
3		Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tanpa Masker Nilai <i>Confidence</i> = 0.8325 Terdeteksi Nomask	83.25%	Sesuai
4		Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Masker Tidak Rapih Nilai <i>Confidence</i> = 0.6135 Terdeteksi Badmask	61.35%	Sesuai
5		Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Masker Kain Nilai <i>Confidence</i> = 0.9245 Terdeteksi Mask	92.45%	Sesuai
6		Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Masker kain Karakter Nilai <i>Confidence</i> = 0.8395 Terdeteksi mask	83.95%	Sesuai
7		Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Masker Kain Karakter Tidak Rapih Nilai <i>Confidence</i> = 0.3629 Terdeteksi Badmask	36.29%	Sesuai
8		Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Masker Kain Hitam Nilai <i>Confidence</i> = 0.8426 Terdeteksi Badmask	84.26%	Sesuai
9		Tidak Terdeteksi	Terdeteksi	Tidak Terdeteks	Masker Kain Hitam Tidak Rapih Nilai <i>Confidence</i> = 0.4010 Terdeteksi	40.10%	Sesuai
...
300		Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Masker kain Karakter	83.95%	Sesuai

Nilai Confidence =
0.8395
Terdeteksi mask

Dari hasil uji coba yang dilakukan sebanyak 300 kali pada masing-masing *class* dimasukan pada table 4 dari beberapa hasil uji coba yang dilakukan ada beberapa hasil prediksi yang tidak sesuai dengan objek yang di deteksi. Pada hasil uji coba dengan nilai *confidence* dan tingkat keakurasi pada setiap *class* nilai tertinggi yang terdeteksi adalah nilai *Confidence* 0.9245 dan 92.45% tingkat keakurasiannya dan nilai terrendah pada setiap *class* yang terdeteksi dengan adalah Nilai *Confidence* 0.4010 dan 40.10% tingkat keakurasiannya.

I. Evaluasi

Confusion Matrix merupakan untuk mengukur tingkat keakurasi dari model yang telah dilatih menggunakan darknet. Berikut merupakan hasil dari evaluasi menggunakan *Confusion Matrix* dalam menghitung keakurasi pada model.

Table 5 Hasil Tabel Prediksi

Prediksi Akurasi	Mask	Badmask	Nomask
Mask	85	5	2
Badmask	0	40	0
Nomask	15	55	98

Keterangan :
 $Mask = (TP = 85), (FP = 15), (FN = 7)$
 $Badmask = (TP = 40), (FP = 60), (FN = 0)$
 $Nomask = (TP = 98), (FP = 2), (FN = 70)$

Mencari nilai *Average Precision* pada pada tabel prediksi di atas dengan masing-masing nilai *class* menggunakan rumus sebagai berikut :

$$precision = \frac{TP}{TP+FP} \times 100\%$$

Nilai *Average Precision* pada *class mask*

$$P(Mask) = \frac{85}{85+15} = 0.85\% \times 100\% = 85\%$$

Nilai *Average Precision* pada *class badmask*.

$$P(Badmask) = \frac{40}{40+60} = 0.4\% \times 100\% = 40\%$$

Nilai *Average Precision* pada *class nomask*.

$$P(Nomask) = \frac{98}{98+2} = 0.98\% \times 100\% = 98\%$$

Menghitung Nilai rata-rata *precision* pada tabel di atas sebagai berikut:

$$precision = \frac{p(mask) + p(badmask) + p(nomask)}{jumlah data uji} \times 100\%$$

$$precision = \frac{85+40+98}{300} = 0.74\% \times 100\% = 74\%$$

Setelah mendapatkan *precision* kemudian menghitung Nilai *Recall* pada masing-masing *class* pada tabel Prediksi dengan menggunakan rumus sebagai berikut:

$$recall = \frac{TP}{TP + FN} \times 100\%$$

Nilai *Average Recall* pada *class mask*.

$$recall = \frac{85}{85+7} = 0.92\% \times 100\% = 92\%$$

Nilai *Average Recall* pada *class badmask*.

$$recall = \frac{40}{40+0} = 1\% \times 100\% = 1\%$$

Nilai *Average Recall* pada *class nomask*

$$recall = \frac{98}{98 + 70} = 0.58\% \times 100\% = 58\%$$

Menghitung rata-rata nilai *recall* pada tabel prediksi diatas sebagai berikut:

$$recall = \frac{r(mask) + r(badmask) + r(nomask)}{jumlah data uji} \times 100\%$$

$$recall = \frac{92+1+58}{300} = 0.50\% \times 100\% = 50\%$$

Menghitung tingkat akurasi pada model;

$$accuracy = \frac{TP(mask) + TP(badmask) + TP(badmask)}{total data uji} \times 100\%$$

$$accuracy = \frac{85+40+98}{300} = 0.74 \% \times 100\% = 74\%$$

Dari hasil evaluasi menggunakan *confusion Matrix* maka didapatkan nilai sebagai berikut:

Masing-masing nilai TP dan FP pada tiap *class* ialah

Mask dengan nilai TP = 85 dan FP = 15 dengan memiliki nilai *Ap* dan *Ar* sebesar 85% dan 92%

Badmask dengan nilai TP = 40 dan FP = 60 dengan memiliki nilai *Ap* dan *Ar* sebesar 40% dan 1%

Nomask dengan nilai TP = 98 dan FP 2 dengan memiliki nilai *Ap* dan *Ar* sebesar 98% dan 58%

Dari keseluruhan hasil evaluasi menggunakan *confusion Matrix* maka didapatkan nilai precision sebesar 74% , recall sebesar 50% dan akurasi sebesar 74%.

IV. KESIMPULAN

Penelitian yang dilakukan terhadap pembuatan sistem dengan menggunakan metode YOLO dalam mengidentifikasi penggunaan masker secara *real-time*. sistem dapat dikatakan berhasil setelah melakukan uji coba dan sistem dapat berjalan serta berhasil mengidentifikasi penggunaan masker. Nilai akurasi yang dihasilkan dalam pengujian sistem menggunakan metode YOLO ini dalam mengidentifikasi penggunaan masker secara *real-time*. berhasil mendekripsi dengan cukup akurat dalam hasil uji coba sistem dengan menggunakan metode yolo. Dengan mendapatkan nilai *precision* sebesar 74%, nilai recall memiliki 50% , dan tingkat akurasi sebesar 74%.

PENGAKUAN

Naskah ilmiah ini sebagai bagian dari Tugas Akhir milik Aldo Zamaludin Fernando dengan judul Identifikasi Citra Penggunaan Masker Secara Real-Time Dengan Arsitektur CNN pada Metode Yolo. dibimbing oleh Bapak Adi Rizky Pratama, M.Kom dan Ibu Ayu Ratna Juwita, M.Kom.

DAFTAR PUSTAKA

- [1] Ramdhani, R. R., Adam, R. I., & Ridha, A. A. (2021). Implementasi Deep Learning untuk Deteksi Masker. *Journal of Information Technology and Computer Science (INTECOMS)*, 4(2), 384–391.
- [2] Pamungkas, B. P. G., Nugroho, B., & Anggraeny, F. (2021). Deteksi Dan Menghitung Manusia Menggunakan YOLO-CNN. *jurnal informatika dan sistem informasi*, Vol. 2, No. 1, 27722-130
- [3] Rahman, D., Setianingsih, C., Dirgantara, F. M., & Telkom, U. (2021). *Sistem Deteksi Pelanggaran Social Distancing Di Ruang Terbuka Menggunakan Algoritma You Only Look Once (YOLO) Outdoor Social Distancing Violation System Detection Using You Only Look Once (YOLO) Algorithm*. 8(5), 6678–6683.
- [4] Jupiyandi, S., Saniputra, F. R., Pratama, Y., Dharmawan, M. R., & Cholissodin, I. (2019). Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(4), 413. <https://doi.org/10.25126/jtiik.2019641275>
- [5] Fandisyah, A. F, Irawan, N., & Winahju, W. S. (2021). Deteksi Kapal Di Laut Indonesia Menggunakan YOLOv3. *jurnal sains dan seni ITS*, Vol. 10, No. 1, 2337-3520