

Deteksi *Social Distancing* Dan Penggunaan Masker Di Restoran Dengan Algoritma *Convolutional Neural Network (CNN)*

1st Ardhana Setyadi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ardhanasetyadi@student.telkomuniversi
ty.ac.id

2nd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

3rd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

Abstrak— *Social distancing* merupakan suatu upaya menjaga jarak, mengurangi kontak fisik antar individu atau kelompok dengan tujuan mencegah penularan penyakit seperti virus Covid-19. Simulasi penelitian dilakukan pada sebuah kamera yang dipasang di ruangan simulasi dari sebuah restoran. Kamera akan mendeteksi penerapan *social distancing* dan penggunaan masker. Kamera akan mendeteksi objek berupa person menggunakan algoritma *You Only Look Once (YOLO)* dan menghitung jarak antar individu yang terdeteksi menggunakan metode *Euclidean Distance*. Sistem dapat mendeteksi wajah tiap individu menggunakan *Haar Cascade* dan melakukan klasifikasi penggunaan masker atau tidak dengan menggunakan algoritma *Convolutional Neural Network (CNN)*. Sistem dapat mendeteksi pelanggaran *social distancing* dan penggunaan masker antar objek person. Hasil training model *YOLOv4* pada rasio 90%:10%, max batches 6000, dan learning rate 0.001 mendapat mAP sebesar 49.02%. Sedangkan untuk CNN didapatkan dengan rasio 90%:10%, batch size 256, dan learning rate 0.0001 mendapat akurasi sebesar 98%. Pada keadaan kamera sejajar dengan objek *YOLO* memiliki tingkat akurasi 100%, pada keadaan letak kamera diatas objek didapatkan akurasi sebesar 77.8%. Sementara CNN dengan range jarak antar wajah dengan kamera 100 cm sampai 300 cm dengan tingkat akurasi sebesar 80.36%.

Kata kunci— *Convolutional Neural Network, Covid-19, Haar Cascade, Social Distancing, YOLO.*

I. PENDAHULUAN

Covid-19 merupakan pandemi yang keberadaannya telah tersebar di seluruh dunia, kemudian mulai memasuki Indonesia pada hari Senin tanggal 2 Maret 2020. Menurut data dari JHU CSSE COVID-19 pada tanggal 6 Oktober 2021, total kasus positif Covid-19 telah mencapai angka 4,22 juta jiwa dengan jumlah kematian sampai saat ini sebesar 142 ribu orang. Kenaikan tajam kasus covid tidak hanya disebabkan oleh varian baru saja, namun karena masyarakat abai akan protokol kesehatan seperti mencuci tangan, memakai masker, menjaga jarak, menjauhi kerumunan dan membatasi mobilisasi.

Pemerintah Indonesia kemudian menerapkan beberapa kebijakan, salah satunya adalah kebijakan PSBB (Pembatasan Sosial Berskala Besar) mengenai *Social Distancing* [1]. *Social Distancing* merupakan kegiatan yang menganjurkan masyarakat untuk tetap berada di dalam rumah

guna menjaga jarak, menghindari keramaian, menghindari kontak fisik dengan orang lain seperti berjabat tangan [2].

Pemerintah telah menerapkan prinsip *social distancing* di berbagai tempat, salah satunya adalah restoran. Restoran merupakan tempat umum sehingga sering terdapat kerumunan di dalam restoran. Namun penerapan *social distancing* saja tidak cukup untuk mencegah rantai penyebaran virus Covid-19. Hal tersebut harus didukung dengan adanya kebijakan menggunakan masker ketika sedang berada di tempat umum. Dengan adanya dua kebijakan tersebut, diharapkan rantai penyebaran virus Covid-19 dapat diminimalkan. Namun pada praktiknya, meskipun pemerintah telah memberikan aturan mengenai *social distancing* dan penggunaan masker, akan sangat sulit untuk memastikan orang-orang dapat mengikuti aturan yang sangat krusial ini. Padahal dengan mengurangi kontak sosial ini dapat mengurangi penularan virus dari orang yang terinfeksi ke orang yang sehat.

Dengan perkembangan teknologi terutama dalam menggunakan metode kecerdasan buatan dalam bidang *Computer Vision* bisa membantu deteksi manusia pada sebuah video digital ataupun citra digital. Beberapa penelitian yang terkait dengan deteksi objek menggunakan algoritma *deep learning* sudah banyak dilakukan. Misalnya, penelitian yang dilakukan oleh Faizal Indaryanto dkk dengan judul “Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis *YOLO* Sebagai Protokol Kesehatan Covid-19”. Pada penelitian ini digunakan *YOLOv3* untuk mendeteksi jumlah dan jarak antar objek manusia pada suatu area [3]. Disisi lain deteksi jarak saja tidak cukup untuk menghambat penyebaran virus ini, masyarakat juga harus dapat melindungi diri dari virus dengan penggunaan masker. Hal ini membuat pemakaian masker dikondisi pandemi covid-19 membuat pengolahan citra digital memiliki manfaat yang baik yaitu untuk mendeteksi kegunaan masker di masa pandemi. Terdapat beberapa penelitian yang telah dilakukan terkait deteksi masker termasuk penelitian mengenai deteksi pemakaian masker menggunakan metode CNN yang dilakukan oleh Restin dkk, untuk mendeteksi pemakaian

masker medis, non medis maupun masker yang tidak benar [4].

Dari permasalahan yang telah dijelaskan, solusi yang ditawarkan adalah dengan membuat sistem yang dapat mendeteksi pelanggaran social distancing menggunakan algoritma object detection YOLOv4 dengan performansi dan akurasi yang lebih akurat dibandingkan YOLOv3 [5] [6]. Sistem ini kemudian akan dikombinasikan dengan deteksi penggunaan masker menggunakan algoritma Convolutional Neural Network CNN [7]. Hasil akhir dari pengerjaan tugas akhir ini adalah sebuah program yang dapat mendeteksi dua atau lebih orang yang melanggar social distancing atau tidak dengan mempertimbangkan masker. Pengembangan sistem ini dapat berperan sebagai *complementary solution* untuk menunjang solusi *holistic* yang sudah diterapkan saat ini untuk mengatasi dan menanggulangi penyebaran virus Covid-19.

II. KAJIAN TEORI

A. Social Distancing

Social Distancing merupakan suatu tindakan yang menganjurkan masyarakat untuk menjaga jarak dan menghindari kontak fisik dengan sekitar [8]. Social distancing juga harus dikombinasikan dengan tindakan lain yang dapat mengurangi penyebaran virus Covid-19, seperti memakai masker dan selalu mencuci tangan minimal selama 20 detik [8]. Tujuan dari adanya social distancing adalah untuk meminimalkan maupun mencegah rantai penyebaran virus Covid-19. Berdasarkan data dari WHO pada bulan Desember 2020, kasus virus Covid-19 mencapai angka 65 juta dengan total kematian melebihi angka 1,5 juta jiwa [9]. Jarak minimal antar individu yang dianjurkan oleh WHO untuk social distancing adalah satu meter [9].

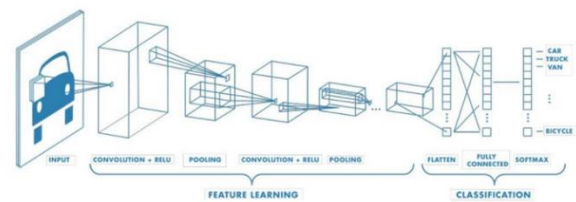
B. Pengolahan Citra

Citra didefinisikan sebagai tiruan atau imitasi yang menggambarkan suatu objek atau benda. Pengolahan citra merupakan suatu pemrosesan citra dengan menggunakan komputer dan teknik-teknik tertentu agar citra yang diproses memiliki kualitas yang lebih baik [10]. Citra dibedakan menjadi dua, yaitu citra analog dan citra digital [10]. Citra analog merupakan citra yang terbentuk oleh sinyal yang bersifat kontinyu dan tidak dapat langsung diolah sehingga harus diubah ke dalam bentuk citra digital, contohnya citra yang tampil pada layar TV atau monitor, dan citra yang dihasilkan dari CT-SCAN [11]. Sedangkan citra digital merupakan citra yang dapat diolah langsung oleh komputer [10]. Citra digital diwakili oleh matriks dua dimensi dengan M kolom dan N baris, dimana terdapat perpotongan antara baris dan kolom tersebut. Perpotongan tersebut dapat disebut sebagai pixel, yaitu elemen terkecil yang terdapat pada sebuah citra [12]. Citra digital dapat digambarkan ke dalam bentuk matrix dibawah ini [9].

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

C. Algoritma CNN (Convolutional Neural Network)

Algoritma CNN atau *Convolutional Neural Network* merupakan salah satu algoritma populer yang terdapat pada *Deep Learning*. CNN biasanya digunakan untuk mendeteksi dan mengklasifikasi sebuah objek pada gambar atau video [13]. CNN mempunyai banyak *layers*, diantaranya ada *input layers*, *output layers*, dan beberapa *hidden layers*. *Layers-layers* tersebut akan belajar untuk mendeteksi fitur yang berbeda dari suatu gambar [14]. Terdapat tiga *layers* yang paling umum adalah *convolution*, *activation/ReLU*, dan *pooling* [15].

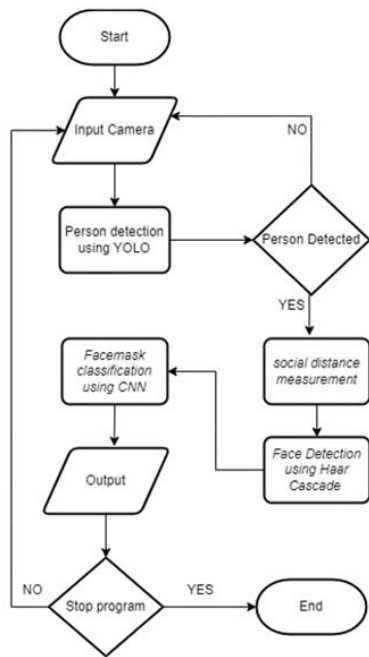


GAMBAR 1
ARSITEKTUR ALGORITMA CNN

Pada gambar 1, dapat dilihat proses-proses yang terjadi pada algoritma CNN. Input berupa citra/gambar akan masuk ke dalam *convolutional layer* untuk menghasilkan gambar baru yang berisi *features* dari gambar yang diinput. Proses konvolusi tersebut akan menghasilkan sebuah output yang biasa disebut *activation map* atau *feature map* yang akan digunakan pada fungsi aktivasi, salah satunya adalah ReLU. Selanjutnya masuk ke *pooling layer*, tujuannya adalah mengurangi dimensi dari *feature map* agar dapat mempercepat proses komputasi dan juga dapat mengatasi *overfitting*.

D. You Only Look Once (YOLO)

YOLO merupakan algoritma untuk mendeteksi objek secara *real time* yang merupakan hasil pengembangan dari CNN (*Convolutional Neural Network*). YOLO diciptakan oleh Joseph Redmon dan tetap dikembangkan hingga saat ini [16]. YOLO menggunakan *single neural network* yang dapat mendeteksi beberapa objek yang terdapat pada suatu gambar secara *real-time*, kemudian dapat menampilkan *bounding boxes* di sekitar objek tersebut [17]. Jika dibandingkan dengan Faster R-CNN yang relatif mirip, YOLO jauh lebih cepat dikarenakan arsitektur dari YOLO lebih sederhana serta dapat mengklasifikasi regresi *bounding boxes* secara bersamaan. YOLO merupakan algoritma berbasis regresi yang dapat memprediksi *class probabilities* dan *bounding boxes* dari sebuah objek secara langsung dari keseluruhan gambar [18].



GAMBAR 5
FLOWCHART SISTEM

Langkah-langkah untuk sistem pendeteksian *social distancing* dan penggunaan masker adalah sebagai berikut:

1. Pendeteksian dimulai ketika menekan tombol *start*, kemudian kamera akan mulai menangkap gambar secara *real time*.
2. Kemudian sistem akan mulai melakukan pendeteksian objek berupa *person* menggunakan arsitektur YOLO.
3. Jika terdapat objek berupa *person* terdeteksi, sistem akan lanjut ke proses selanjutnya. Namun apabila tidak ada *person* yang terdeteksi, sistem akan kembali mengulang proses dari awal sampai sistem mendeteksi *person*.
4. Selanjutnya sistem akan mulai menghitung jarak antara dua *person* atau lebih yang terdeteksi menggunakan *Euclidean Distance*.
5. Setelah mengukur jarak antar individu, sistem akan mulai mendeteksi wajah mereka dengan menggunakan *Haar Cascade*. Apabila wajah terdeteksi, maka sistem akan melanjutkan ke proses klasifikasi penggunaan masker menggunakan *Convolutional Neural Network*.
6. Pada proses output akan menampilkan dua *bounding box*. *Bounding box* pertama untuk kategori deteksi *social distancing*. Ketika jarak dari dua atau lebih *person* yang terdeteksi kurang dari angka yang telah ditentukan, maka *output* berupa tulisan "*Unsafe*". Sebaliknya, ketika jarak sudah melebihi dari angka yang telah ditentukan, maka *output* akan berupa tulisan "*Safe*". *Bounding box* yang kedua untuk kategori penggunaan masker. *Output* berupa tulisan "*Bermasker*" ketika individu yang terdeteksi memakai masker, namun ketika individu yang terdeteksi tidak menggunakan masker, *output* akan berupa tulisan "*Tidak*

Bermasker". Setelah *output* ditampilkan, pengguna dapat berhenti atau melanjutkan pendeteksian dari awal.

C. Data Latih

Data latih yang digunakan terdapat dua jenis. Pada YOLOv4, model yang digunakan sebelumnya sudah dilatih sehingga dapat mendeteksi lebih dari 80 kelas objek yang berbeda. Maka dari itu dilakukan transfer learning agar model hanya dapat mendeteksi satu kelas objek saja, yaitu *person*. Dataset yang digunakan untuk melatih model sehingga hanya dapat mendeteksi *person* diambil dari open image dataset dengan jumlah dataset yang diambil adalah sebanyak 1800 gambar.



GAMBAR 6
CONTOH DATA LATIH KELAS *PERSON*

Selain model YOLOv4, algoritma CNN juga menggunakan model sebagai pendeteksinya. Model tersebut didapat dari hasil training dataset dengan jumlah dataset yang digunakan berjumlah 11376 gambar. Setelah melakukan training *dataset*, model akan terbentuk dan dapat mendeteksi dua kelas, yaitu masker dan non-masker.



GAMBAR 7
CONTOH DATA LATIH MASKER DAN NON-MASKER

D. Data Uji

Data uji merupakan data yang akan digunakan untuk proses pengujian/testing. Data uji yang diambil adalah berupa foto atau video yang diambil sendiri. Terdapat beberapa kategori untuk pengambilan data uji, yaitu jarak antar person, jarak kamera, dan sudut/angle kamera.

E. Analisis Kebutuhan Perangkat Keras

Berikut merupakan analisis spesifikasi komponen yang dibutuhkan untuk merancang sistem. Untuk merancang sistem pendeteksian *social distancing* dan pendeteksian masker, dibutuhkan laptop sebagai perangkat keras penunjang dan sebagai alat untuk mengolah program. Laptop yang akan digunakan memiliki spesifikasi sebagai berikut:

1. Processor : AMD Ryzen 7 4800H CPU @ 2.90GHz (16CPUs)
2. RAM : 16 GB
3. Storage : SSD 256 GB and HDD 1 TB

4. GPU : NVIDIA GeForce GTX 1660 TI
5. OS : Windows 10 Home Single Language 64 bit

Selain laptop, *external webcam* juga dibutuhkan sebagai alat pendeteksian *social distancing* dan penggunaan masker. Spesifikasi kamera/webcam yang disarankan adalah sebagai berikut:

1. Resolusi Kamera : 1080 x 1920 atau lebih
2. *Frame rate* : 30 FPS atau lebih

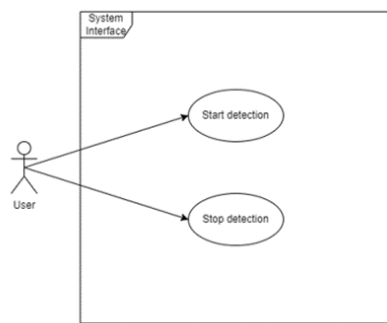
F. Perangkat Lunak

Untuk merancang sistem pendeteksi *social distancing* dan pendeteksi masker, beberapa perangkat keras yang dibutuhkan diantaranya adalah sebagai berikut:

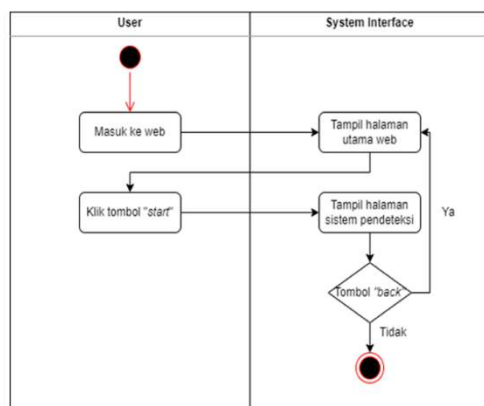
1. Sistem Operasi Windows/Mac OS
2. OpenCV
3. Bahasa Pemrograman Python

G. Perancangan Website

Website yang dirancang berfungsi untuk memudahkan pengguna sistem untuk melakukan pendeteksian tanpa harus menjalankan program secara terus-menerus. Untuk mengetahui cara kerja dan penggunaan dari *website* ini, dapat dilihat pada gambar *Use Case Diagram* dan *Activity Diagram* di bawah ini.



GAMBAR 8
USE CASE DIAGRAM



GAMBAR 9
ACTIVITY DIAGRAM

IV. HASIL DAN PEMBAHASAN

A. Skenario Training Model *Person YOLOv4*

Waktu pengujian dilakukan untuk menguji lamanya waktu selama proses perangkat keras berlangsung dimulai dari penginputan data ke Antares sampai proses eksekusi alarm dan cut off aliran listrik.

Untuk melakukan training model untuk pendeteksi objek *person* dibutuhkan *dataset*. *Dataset* tersebut diambil dari *Open Image Datasets* yang telah disediakan oleh Google. Kemudian *dataset* tersebut akan dibagi menjadi tiga rasio, yaitu rasio 90%:10%, rasio 80%:20%, dan rasio 70%:30%. Proses training dilakukan di *Google Colaboratory*, yaitu *coding environment* bahasa pemrograman *Python* dengan format *notebook* yang berbasis *Cloud*.

Sebelum melakukan proses *training*, perlu dilakukan konfigurasi sistem untuk mendapatkan model yang baru, dikarenakan model sebelumnya telah di *training* untuk tujuan tertentu. Konfigurasi sistem yang diubah adalah sebagai berikut:

<i>Batch Size</i>	= 64
<i>Subdivisions</i>	= 16
<i>Learning rate</i>	= 0.001
<i>Max Batches</i>	= 6000
<i>Classes</i>	= 1

Hasil *output* yang didapatkan dari *training* adalah *weight file*. *Weight file* tersebut nantinya akan digunakan untuk sistem pendeteksian objek *person*. Selain *weight file*, beberapa parameter akan didapatkan setelah proses *training* selesai. Parameter-parameter tersebut adalah akurasi, presisi, *recall*, *F1 score*, *IoU*, dan *mAP*.

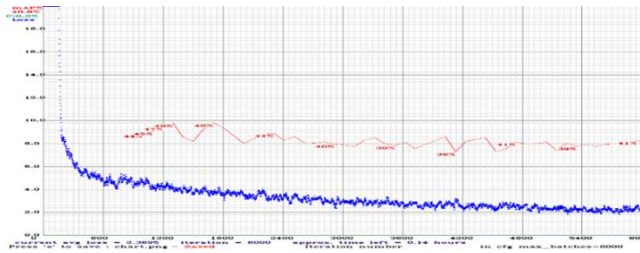
1. Training Berdasarkan Rasio Partisi Data

Pengujian ini dilakukan untuk mengetahui presisi dari sistem dengan cara membagi data menjadi data *training* dan data *testing*. Untuk perbandingan rasio partisi data dapat dilihat pada tabel 1 berikut ini.

TABEL 1
TABEL RASIO PARTISI DATA

Percobaan ke-	Rasio Training:Testing	Durasi training
1	90:10	± 12 Jam
2	80:20	± 12 Jam
3	70:30	± 10 Jam

Berdasarkan hasil pengujian berdasarkan rasio partisi data diatas, training dengan rasio 90:10 menghasilkan model dengan performansi terbaik. Untuk grafik *mAP* yang diperoleh rasio 90:10 dan perbandingan hasil *training* dari ketiga rasio tersebut dapat dilihat pada gambar dan tabel berikut ini.



GAMBAR 10
GRAFIK MAP RASIO 90:10

Terlihat pada gambar 10 di atas, grafik mAP tampil dengan sempurna tanpa adanya garis yang terputus. Hal tersebut dikarenakan selama proses training tidak terjadi kendala.

TABEL 2
HASIL TRAINING RASIO BERBEDA

Rasio	Presisi	Recall	F1 score	Avg.IoU	Avg.Loss	mAP
90:10	42%	61%	49%	32.59%	1	49.02%
80:20	32%	66%	43%	23.58%	1	41.65%
70:30	48%	56%	52%	37.55%	1	44.51%

Dari hasil *training* dengan rasio *dataset* yang berbeda, hasil yang diperoleh dengan rasio *dataset* 90% *data train* dan 10% *data test* menghasilkan model dengan performansi terbaik dan optimal. Kemudian model dengan nilai performansi terbaik dan optimal tersebut akan digunakan dalam pengujian program.

B. Skenario Training Model Masker CNN

Untuk melakukan *training* model untuk pendeteksi masker dan non-masker dibutuhkan *dataset*. *Dataset* tersebut diambil dari Github. *Dataset* tersebut terdapat dua kelas, yaitu bermasker dan tidak bermasker. *Dataset* tersebut akan dibagi menjadi tiga rasio, yaitu rasio 90%:10%, rasio 80%:20%, dan rasio 70%:30%. Proses training dilakukan di Jupyter Notebook, yaitu *coding environment* bahasa pemrograman Python dengan format *notebook*.

Sebelum melakukan proses *training*, perlu dilakukan konfigurasi sistem untuk mendapatkan model yang baru. Konfigurasi awal sistem adalah sebagai berikut:

Learning rate = 0.0001
Batch Size = 256
Epoch = 20

Hasil *output* yang didapatkan dari hasil *training* adalah *model file*. *Model file* tersebut nantinya akan digunakan untuk sistem pendeteksian wajah dengan klasifikasi masker atau tidak bermasker. Selain *model file*, beberapa parameter akan didapatkan setelah proses *training* selesai. Parameter-parameter tersebut adalah akurasi, presisi, *recall*, dan F1 score.

1. Training Berdasarkan Rasio Partisi Data

Training berdasarkan partisi data dilakukan untuk mengetahui seberapa akurat sistem pendeteksi masker yang telah dibuat. Proses *training* dilakukan dengan cara membagi *dataset* ke dalam *data train* dan *data test*. Berikut merupakan tabel persentase partisi *dataset*.

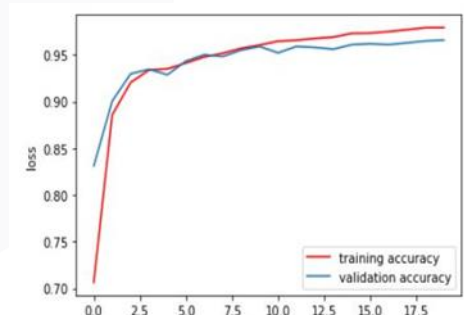
TABEL 3
TABEL PERSENTASE PARTISI DATASET

Percobaan Ke-	Data Train	Data Test	Durasi Latih
1	90%	10%	2 Jam
2	80%	20%	2 Jam
3	70%	30%	2 Jam

Berdasarkan hasil pengujian berdasarkan rasio partisi data di atas, training dengan rasio 90:10 menghasilkan model dengan performansi terbaik. Untuk hasil performansi ketiga rasio tersebut dapat dilihat pada tabel dibawah ini.

TABEL 4
HASIL PERFORMANSI PENGUJIAN MODEL MASKER

Rasio	Parameter	Hasil (%)
90:10	Accuracy	98%
	Precision	99%
	Recall	99%
	F1 Score	98%
80:20	Accuracy	97%
	Precision	97%
	Recall	97%
	F1 Score	97%
70:30	Accuracy	97%
	Precision	96%
	Recall	97%
	F1 Score	97%



GAMBAR 11
GRAFIK AKURASI RASIO 90:10

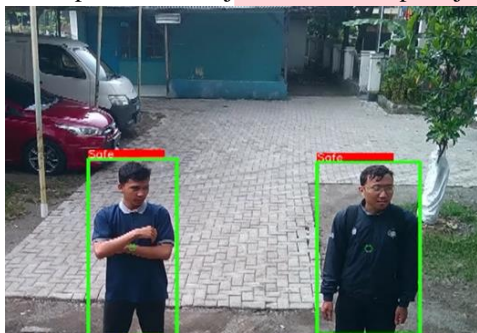
Dapat dilihat bahwa training dengan rasio 90:10 mendapatkan model dengan hasil performansi terbaik. Model dengan performansi terbaik nantinya akan digunakan untuk pengujian program.

C. Pengujian *Social Distancing*

Setelah mendapatkan *weight file* dari hasil *training* YOLOv4, *weight file* tersebut sudah bisa digunakan untuk keperluan pengujian program untuk mendeteksi *person*.

Untuk skenario pengujian program *social distancing* akan dibagi menjadi dua skenario penempatan kamera. Skenario pertama adalah posisi kamera terletak seperti CCTV pada umumnya, kemudian skenario kedua yaitu posisi kamera sejajar dengan *person* yang terdeteksi. Selain skenario penempatan kamera, dilakukan pengujian berdasarkan jarak dan sudut. Jarak yang digunakan adalah 50 cm, 100 cm, dan 150 cm. sedangkan sudut yang digunakan adalah 0°, 30°, dan 60°.

Jarak aman antar *person* agar tidak terdeteksi sebagai pelanggaran *social distancing* adalah satu meter, sedangkan pada program berbeda. Untuk mencari nilai *pixel* yang merepresentasikan jarak satu meter pada jarak asli, perlu menggunakan metode *trial and error*. Nilai *pixel* dicari mulai dari 100 hingga 300 dan diperoleh nilai *pixel* sebesar 150 untuk merepresentasikan jarak satu meter pada jarak asli.



GAMBAR 12
CONTOH PENGAMBILAN JARAK 150 CM SUDUT 0°
SKENARIO CCTV

Berikut merupakan tabel hasil pengujian *social distancing* yang dilakukan adalah sebagai berikut.

TABEL 5
TABEL HASIL PENGUJIAN SKENARIO CCTV

No	Sudut	Jarak (cm)	Hasil Pengujian		Keterangan Gambar
			Berhasil	Tidak	
1	0°	50	v		Gambar 4.8
2	0°	100	v		Gambar 4.11
3	0°	150	v		Gambar 4.14
4	30°	50	v		Gambar 4.9
5	30°	100	v		Gambar 4.12
6	30°	150	v		Gambar 4.15
7	60°	50	v		Gambar 4.10
8	60°	100		v	Gambar 4.13
9	60°	150		v	Gambar 4.16

Dari hasil yang didapatkan diatas, maka nilai akurasi dari sistem pendeteksi pelanggaran *social distancing* skenario CCTV dapat diperoleh dari rumus berikut ini.

$$\text{Akurasi Social Distancing} = \frac{\text{Total Pengujian Berhasil}}{\text{Total Pengujian}} \times 100\% \quad (3)$$

$$= \frac{7}{9} \times 100\%$$

$$= 77.8 \%$$

TABEL 6
TABEL HASIL PENGUJIAN SKENARIO SEJAJAR

No	Sudut	Jarak (cm)	Hasil Pengujian		Keterangan Gambar
			Berhasil	Tidak	
1	0°	50	v		Lampiran A
2	0°	100	v		Lampiran A
3	0°	150	v		Lampiran A
4	30°	50	v		Lampiran A
5	30°	100	v		Lampiran A
6	30°	150	v		Lampiran A
7	60°	50	v		Lampiran A
8	60°	100	v		Lampiran A
9	60°	150	v		Lampiran A

Dari hasil yang didapatkan diatas, maka nilai akurasi dari sistem pendeteksi pelanggaran *social distancing* skenario CCTV dapat diperoleh dari rumus berikut ini.

$$\text{Akurasi Social Distancing} = \frac{\text{Total Pengujian Berhasil}}{\text{Total Pengujian}} \times 100\% \quad (4)$$

$$= \frac{9}{9} \times 100\%$$

$$= 100 \%$$

D. Pengujian Masker

Berdasarkan hasil pengujian sebelumnya, maka pada pengujian ini menggunakan partisi data 90% *data train* dan 10% *data test*, *learning rate* bernilai 0.0001, dan *batch size* sebesar 256. Setelah mendapatkan *model file* dari hasil *training* tersebut, *model file* tersebut sudah bisa digunakan untuk keperluan pengujian program untuk mendeteksi masker dan tidak bermasker. Untuk skenario pengujian program masker akan menggunakan skenario jarak dari kamera terhadap objek wajah yang terdeteksi.

Pengujian berdasarkan algoritma CNN dilakukan untuk memperoleh akurasi yang dihasilkan sistem dengan algoritma CNN. Perhitungan akurasi dilakukan dengan menghitung *error* terlebih dahulu kemudian didapatkan hasil akurasi tersebut, dengan rumus:

$$\text{Error} = \frac{\text{Hasil real} - \text{Hasil sistem}}{\text{Hasil real}} \times 100\% \quad (5)$$

Sehingga didapatkan nilai akurasi dengan rumus:

$$\text{Akurasi} = 100\% - \text{Error} \quad (6)$$

Berikut merupakan tabel hasil pengujian masker yang dilakukan adalah sebagai berikut.

TABEL 7
TABEL HASIL PENGUJIAN AKURASI PENDETEKSI MASKER

Jarak	Hasil Sistem		Hasil Real		Error	Akurasi
	Mask	Non-mask	Mask	Non-mask		
100	2	-	2	-	0%	100%
100	1	1	1	1	0%	100%
100	-	2	-	2	0%	100%
150	4	-	4	-	0%	100%
150	2	2	2	2	0%	100%
150	-	4	-	4	0%	100%
200	1	3	-	4	25%	75%
200	2	-	4	-	50%	50%
200	-	2	-	2	0%	100%
300	-	2	-	4	50%	50%
300	2	-	4	-	50%	50%
300	1	1	1	1	0%	100%
300	1	-	1	1	100%	0%
300	1	1	1	1	0%	100%
Rata-rata					19.64%	80.36%

Hasil pengujian pendeteksi masker pada tabel 7 dapat disimpulkan bahwa berdasarkan rumus diatas didapatkan akurasi sistem sebesar 80.36%. Pada tabel diatas terlihat pada jarak tertentu, orang yang memakai kacamata akan terdeteksi sebagai pengguna masker atau tidak terdeteksi sama sekali. Hal tersebut dikarenakan isi *dataset* tidak terlalu banyak orang yang menggunakan kacamata sehingga algoritma tidak terlalu mempelajari orang yang memakai kacamata. Selain itu, pada tabel diatas terlihat ada beberapa orang yang wajahnya tidak terdeteksi. Hal tersebut dikarenakan wajah terlalu gelap sehingga algoritma tidak mengenal dan mendapatkan struktur wajah pada orang tersebut.

V. KESIMPULAN

Sistem dapat mendeteksi pelanggaran *social distancing* menggunakan algoritma YOLO antar objek *person* dengan *range* jarak 50-100 cm dengan tingkat akurasi 100% pada keadaan kamera sejajar dengan objek dan mendapatkan akurasi sebesar 77.8% pada keadaan letak kamera diatas objek menyerupai CCTV. Sistem juga dapat mendeteksi penggunaan masker menggunakan algoritma CNN dengan *range* jarak antar wajah dengan kamera 100 cm sampai 300 cm dengan tingkat akurasi sebesar 80.36%.

Hasil *training* model YOLOv4 terbaik yang didapatkan adalah dengan rasio 90%:10% dengan *max batches* sebesar 6000 dan *learning rate* 0.001 sehingga mendapat mAP sebesar 49.02%. Sedangkan untuk *training* model CNN terbaik yang didapatkan adalah dengan rasio 90%:10% dan nilai *batch size* sebesar 256 serta *learning rate* dengan nilai 0.0001 sehingga mendapat akurasi sebesar 98%.

REFERENSI

- [1] A. A. Pradana, C. and N. , "Pengaruh Kebijakan Social Distancing Pada Wabah COVID-19 Terhadap Kelompok Rentan di Indonesia," *Jurnal Kebijakan Kesehatan Indonesia : JKKI*, vol. 9, pp. 61-67, 2020.
- [2] T. V. Radhitya, N. Nurwati and M. Irfan, "Dampak Pandemi COVID-19 Terhadap Kekerasan dalam Rumah Tangga," *Jurnal Kolaborasi Resolusi Konflik*, vol. 2, pp. 111-119, 2020.
- [3] F. Indaryanto, A. Nugroho and A. Faridh Suni, "Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol," *Edu Komputika Journal*, pp. 31-32, 2021.
- [4] R. Alfinda Zai and F. Astuti Hermawati, "Sistem Deteksi Pemakaian Masker Menggunakan Metode Convolutional Neural Networks (CNN)," *Proceeding KONIK (Konferensi Nasional Ilmu Komputer)*, vol. 5, pp. 182-187, 2021.
- [5] A. F. Fandisyah, N. Iriawan and W. S. Winahju, "Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3," *Jurnal Sains dan Seni ITS*, vol. 10, pp. D25-D32, 2021.
- [6] F. Rofli, G. Priyandoko, M. I. Fanani and A. Suraji, "Vehicle Counting Accuracy Improvement By Identity Sequences Detection Based on Yolov4 Deep Neural Networks," *Teknik*, vol. 42, no. 2, pp. 169-177, 2021.
- [7] I. W. Suartika, A. Y. Wijaya and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, pp. A65-A69, 2016.
- [8] S. Srinivasan, R. S. R, R. R. Biradar and R. S. A, "COVID-19 Monitoring System using Social Distancing and Face Mask Detection on Surveillance video datasets," in *International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, 2021.
- [9] S. Madane and D. Chitre, "Social Distancing Detection and Analysis through Computer Vision," in *6th International Conference for Convergence in Technology (I2CT)*, Pune, 2021.
- [10] A. S. Irtawaty and R. Jayanti, "Implementasi Pengolahan Citra Pada Analisis Ciri Bakteri Yogurt," *Jurnal Sains Terapan*, vol. 2, pp. 83-87, 2016.
- [11] D. I. Surya Saputra, T. B. Pranata and S. W. Handani, "Prototype Aplikasi Pengolah Citra Invert Sebagai Media Pengolah Klise Foto," in *Conference on Information Technology, Information System and Electrical Engineering*, Yogyakarta, 2016.

- [12] R. Kusumanto and A. N. Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB," in *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, Palembang, 2011.
- [13] A. T. Vo, H. S. Tran and T. H. Le, "Advertisement Image Classification Using Convolutional Neural Network," in *9th International Conference on Knowledge and Systems Engineering(KSE)*, Ho Chi Minh, 2017.
- [14] F. J. Wijaya, S. P. Sutra, P. W. Kosasih and P. Sirait, "Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun," *Jurnal SIFO Mikroskil*, vol. 21, pp. 1-10, 2020.
- [15] MathWorks, "Convolutional Neural Network," MathWorks, [Online]. Available: <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>. [Accessed 14 Desember 2021].
- [16] T. Q. Vinh and N. T. Ngoc Anh, "Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier," in *International Conference on Advanced Computing and Applications (ACOMP)*, Ho Chi Minh, 2020.
- [17] D. Jeong, "Road Damage Detection Using YOLO with Smartphone Images," in *IEEE International Conference on Big Data (Big Data)*, Odense, 2020.
- [18] D. Benyang, L. Xiaochun and Y. Miao, "Safety Helmet Detection Method based on YOLO v4," in *16th International Conference on Computational Intelligence and Security (CIS)*, Guilin, 2020.
- [19] R. Suwanda, Z. Syahputra and E. M. Zamzami, "Analysis of Euclidean Distance and Manhattan Distance in the K-Means Algorithm for Variations Number of Centroid K," in *ICCAI*, Medan, 2019.
- [20] A. Ghorai, S. Gawde and D. Kalbande, "Digital Solution for Enforcing Social Distancing," in *International Conference on Innovative Computing and Communication (ICICC)*, Mumbai, 2020.