JCONES Vol. 11, No. 1 (2022) 21-30
Journal of Control and Network Systems
Situs Jurnal : http://jurnal.dinamika.ac.id/index.php/jcone

Sistem Deteksi Kapasitas Orang Di Dalam Ruangan Menggunakan Metode Faster R-Cnn

Sellina Nuril Laili[1)] **Heri Pratikno**[2)] **Harianto**[3)] **Jusak**[3)]
Program Studi/Jurusan Teknik Komputer
Universitas Dinamika
Jl. Raya Kedung Baruk 98 Surabaya, 60298
Email: 1)18410200057@dinamika.ac.id , 2) heri@dinamika.ac.id, 3) hari@dinamika.ac.id 4) jusak@dinamika.ac.id

Abstrak: Upaya pencegahan penularan virus Covid-19 dilakukan terus menerus oleh pemerintah, salah satunya adalah memberlakukan program PSBB (Pembatasan Sosial Berskala Besar). Program ini bertujuan untuk membatasi kegiatan sosial masyarakat seperti meliburkan tempat kerja. Oleh sebab itu dibuat Keputusan Menteri Kesehatan Republik Indonesia Nomor HK.01.07/MENKES/328/202. Dalam keputusan ini terdapat panduan pencegahan dan pengendalian Covid-19 di tempat kerja perkantoran dan industri dalam mendukung keberlangsungan usaha pada situasi pandemi, salah satunya adalah menerapkan pembatasan jumlah orang dalam suatu ruangan dengan menerapkan sistem *shift* WFH (*Work From Home*) dan WFO (*Work From Office*). Dari permasalahan ini, penulis memberikan solusi untuk membantu meminimalisir penyebaran Covid-19 melalui layanan dan penerapan di bidang teknologi. Pada Penelitian ini, penulis membuat sistem untuk menghitung jumlah orang masuk, keluar, dan jumlah orang yang berada di dalam ruangan secara otomatis. Sistem diproses dengan pengolahan citra digital menggunakan metode Faster R-CNN dengan bahasa *Python*. Metode Faster R-CNN yang digunakan merupakan salah satu metode *deep learning* yang berfungsi mengenali objek pada suatu citra atau gambar yang ditangkap oleh kamera. Dengan menggunakan metode ini sistem hanya dapat mendeteksi *object* orang saja, hasil pengujian yang didapatkan dari penelitian ini menunjukkan bahwa sistem mampu mendeteksi tiga pergerakan keberadaan orang yang berjalan lambat, berjalan normal dan berjalan cepat yang berada di POV (*Point Of* View) kamera dengan tingkat akurasi 77%. Sedangkan tingkat akurasi perhitungan sistem untuk jumlah orang yang masuk, keluar, dan jumlah orang yang berada di dalam ruangan dengan hasil data *reporting* yang disimpan dalam bentuk file *spreadsheet* (*.csv*) memiliki kesesuaian data 100%, data *reporting* pada sistem ini dicatat setiap detik dalam waktu kurang lebih 20 menit. Sistem (model) yang dibuat oleh penulis memiliki nilai *Accuracy* sebesar 100%, memiliki nilai *Precision* sebesar 1, nilai *Recall* sebesar 1, nilai F1 – *Score* sebesar 1, nilai *Rate Accuracy* sebesar 79% dan *Rate Loss* sebesar 21%.

Kata Kunci: *Faster R-CNN, Pembatasan Jumlah Orang, Deteksi Orang, Pembatasan Sosial, Covid-19*

Dari awal pandemic melanda Indonesia sampai saat ini, banyak masyarakat yang belum mempercayai adanya Covid-19, sehingga banyak yang tidak mengikuti aturan pemerintah yang dibuat untuk menekan penyebaran Covid-19 di Indonesia. Pemerintah Indonesia sudah mengerahkan berbagai macam upaya untuk memutus rantai penyebaran Covid-19, salah satunya adalah dengan menerapkan PSBB (Pembatasan Sosial Berskala Besar) yang bertujuan untuk membatasi kegiatan sosial masyarakat Indonesia. Upaya ini dilakukan juga untuk mengetatkan penerapan *social* distancing karena masyarakat masih sulit menerapkannya. Pada lingkungan kantor perusahaan terutama di ruangan rapat yang merupakan tempat berkumpulnya beberapa karyawan untuk berdiskusi. Karena dunia kerja tidak mungkin dilakukan pembatasan secara terus menerus yang berakibat dapat menurunkan ekonomi Indonesia, maka penerapan *social distancing* perlu diketatkan di lingkungan kerja, dengan tujuan meminimalisir penyebaran Covid-19 dalam lingkup yang lebih kecil.

Oleh sebab itu Keputusan Menteri Kesehatan Republik Indonesia Nomor HK.01.07/MENKES/328/2020 membuat panduan pencegahan dan pengendalian Covid-19

di tempat kerja perkantoran dan industri dalam mendukung keberlangsungan usaha pada situasi pandemi. Pada keputusan tersebut dikatakan bahwa upaya untuk melakukan mitigasi dan menyiapkan tempat kerja harus dilakukan seoptimal mungkin, sehingga dapat beradaptasi dengan pola hidup baru selama pandemi (Kemenkes RI, 2020).

Dari permasalahan diatas, Penelitian ini memberikan solusi di bidang teknologi dengan membuat sistem deteksi kapasitas orang di dalam ruangan. Sistem ini dibuat dengan menetapkan batas jumlah kapasitas orang yang boleh berada di dalam ruangan tersebut untuk menghindari perkumpulan massa yang berlebihan. Sistem ini menggunakan metode Faster R-CNN untuk mendeteksi objek orang. Selain itu sistem ini juga dapat melakukan perhitungan, yaitu menghitung jumlah orang masuk, jumlah orang keluar, serta jumlah orang yang berada did dalam ruangan tersebut. Sistem ini mengambil inputan data yang berasal dari hasil rekaman kamera atau inputan secara *real* time. Hasil perhitungan yang diambil oleh sistem ini, nantinya disimpan dalam bentuk file *spreadsheet* (.csv).

Metode Faster R-CNN merupakan sebuah algoritma deteksi objek dengan basis wilayah terbaru dengan manampilkan hasil yang luar biasa pada berbagai deteksi objek. Metode ini memiliki tingkat akurasi yang llebih tinggi dan lebih akurat dalam mendeteksi object (Sunario Megawan & Wulan Sri Lestari, 2020).

LANDASAN TEORI

Webcam

Webcam merupakan perangkat keras yang digunakan untuk mengambil gambar atau video, tergolong jenis kamera yang memberikan layanan untuk memenuhi kebutuhan yang berbasis web. Webcam sendiri memiliki cara kerja yang unik yaitu dengan menangkap cahaya melalui lensa dan dibantu dengan detektor cahaya mikrokospik, biasanya berteknologi berteknologi *Charge Couple Device* (CCD) atau CMOS image sensor (Elekomp, 2018).

Pada penelitian ini, webcam yang digunakan adalah Logitech C310 yang menyediakan fitur teknologi *RightLight 2 built-in*, sehingga dapat merekam video serta mengambil gambar sebaik mungkin (Plazakamera.com, n.d.).

Gambar 1. Webcam Logitech C310
(Sumber: https://www.plazakamera.com/shop/logitech-c310-hd-webcam/)

Deep Learning

Deep Learning adalah suatu metode yang membuat computer mampu mempelajari tugas-tugas yang ditugaskan. *Deep Learning* mempunyai kemampuan untuk mengenali, membedakan suatu objek dengan objek lainnya, serta berperan penting dalam mengenali suara seperti suara yang terdapat di Ponsel, Televisi, dan sebagainya (Sena, 2018).

Bahasa Python

Bahasa *Python* merupakan salah satu bahasa pemrograman yang melakukan eksekusi dari beberapa instruksi secara langsung dengan menggunakan metode orientasi objek (*Object Orientation Programming*) serta *semantic dynamic* untuk meningkatkan kemampuan membaca *syntax*. Dalam penelitian ini, penulis menggunakan bahasa *Python*, dengan memanfaatkan beberapa library yang disediakan oleh *Python*.

OpenCV (*Open Source Computer Vision*)

OpenCV merupakan *library* yang berfungsi untuk mengelola gambar dan video, bersifat *opensource* yang artinya gratis, tidak berbayar, dan dapat di download oleh siapapun. Kepanjangan dari "CV" adalah *Computer Vision*, artinya computer yang digunakan untuk mengelola citra/gambar yang ditangkap oleh kamera/webcam kemudian melakukan konversi *analog to digital* dan diolah di dalam computer. Salah satu tujuan pengolahan citra/gambar ini adalah memperbaiki kualitas gambar serta mengidentifikasi gambar (Kandir, 2016).

Numerical Python (Numpy)

Numerical Python merupakan salah satu *library python* yang terarah pada *scientific computic*, aljabar linear, operasi vector (1-d *array*), dan matrix (2-d *array*). NumPy memiliki sebuah

kemampuan untuk membentuk objek N-dimensional *array*. Pada *Python* selain NumPy, terdapat juga *library list* dengan kemampuan yang mirip dengan NumPy. Tetapi NumPy lebih unggul dari *List*, salah satu keunggulannya adalah NumPy mengkonsumsi *memory* lebih kecil, dan *runtime* yang lebih cepat.

Imutils

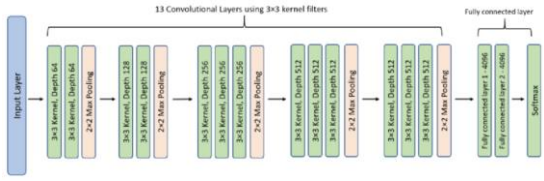
Imutils merupakan salah satu alternatif *library* dari OpenCV. Umumnya *library* imutils ini digunakan untuk menampilkan video. Selain itu *library* imutils juga memiliki berbagai macam kegunaan diantaranya yaitu, melakukan *translation*, *rotation*¸ *resizing* dan *skeletonization*.

Object Detection

Object Detection meerupakan sebuah metode yang digunakan untuk mendeteksi *object* pada suatu gambar atau video. Pada penelitian ini, penulis menggunakan metode ini untuk mendeteksi *object* orang, penulis menggunakan metode ini dengan model yang sudah dilatih, sehingga program yang disajikan hanya mampu mendeteksi *object* orang (Anggara et al., n.d.).

VGG-16

VGG16 merupakan Arsitektur Jaringan Saraf Konvolusi (CNN) sederhana dan banyak digunakan untuk ImageNet, proyek basis data visual besar dan digunakan dalam penelitian perangkat lunak pengenalan objek visual. “VGG” adalah singkatan dari *Visual Geometry Group*, dan '16' menyiratkan bahwa arsitektur ini memiliki 16 *layer* yang terdiri dari 13 *layer* konvolusi, 2 *layer fully connected* dan 1 *layer classifier*.



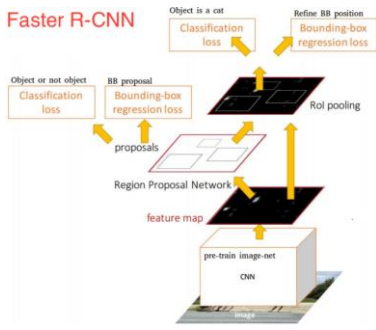
Gambar 2. Arsitektur VGG-16
(Sumber:
https://www.researchgate.net/publication/350115831/figure/fig1/AS:1002342325428227@1615988441584/Gambar-4-Arsitektur-VGG16-9.png/)

Dari gambar 2 di atas dapat diketahui bahwa semua *layer* konvolusi memiliki ukuran kernel 3x3. Jumlah *filter* pada setiap *layer* konvolusi menjadi perbedaan utama dari arsitektur VGG-16. 2 *layer* konvolusi pertama mempunyai jumlah *filter* 64, *layer* 3 dan 4 mempunyai jumlah kernel 128 dan terdapat juga *layer* konvolusi yang mempunyai jumlah *filter* berbeda-beda, yaitu 256 (*layer* 4,5,6) dan 512 (*layer* 7,8,9,10,11,12). 2x2 *max pooling* dilakukan setelah *layer* konvolusi 2, 4, 7, 10 dan 13. *Output* dari *pooling* terakhir dihubungkan ke *fully connected layer* yang nantinya terhubung ke *classifier* untuk menentukan kelas dari sebuah citra (Rismiyati & Luthfiarta, 2021). Pada penelitian ini, penulis menggunakan Arsitektur VGG-16 untuk proses konvolusi dari sebuah gambar, yang kemudian *output* diteruskan untuk proses pendeteksian *object* orang dengan menggunakan metode Faster R-CNN.

Faster R-CNN

Faster R-CNN adalah metode *deep learning* yang berfungsi mengenali objek pada suatu gambar. Objek dapat dikenali dengan menelusuri ciri-ciri objek pada suatu gambar, melalui beberapa *layer* proses konvolusi atau lebih dikenal dengan *Convolutional Neural Network* (CNN) (Alamsyah & Pratama, 2019).

Faster R-CNN merupakan sebuah algoritma deteksi objek dengan 2 modul utama yaitu *Deep Fully Convolutional Network* yang terdapat RPN (*Region Proposed* Network) di dalamanya dan modul yang kedua yaitu detektor dari Faster R-CNN yang berasal dari Fast R-CNN. Metode ini merupakan pengembangan dari Fast R-CNN yang merubah bagian *selective search* menjadi RPN (Sunario Megawan & Wulan Sri Lestari, 2020).



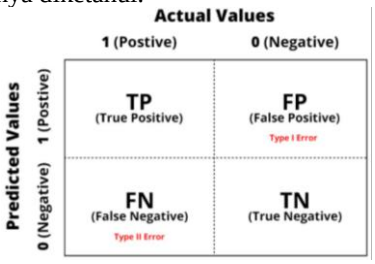
Gambar 3. Bagan Faster R-CNN
(Sumber:
https://medium.com/nodeflux/convolutional-neural-net-untuk-deteksi-objek-f14d72f11ba6/)

Dari gambar 3 diatas, dapat dijelaskan oleh penulis terkait pengertian alur proses yang dilakukan oleh Faster R-CNN:
Saat program dijalankan, program mengambil data *input* berupa gambar atau video, data input tersebut melalui proses konvolusi dan *Pre-train image*

menggunakan arsitektur VGG-16, pada proses ini juga *feature map* dihasilkan. *Feature map* yang dihasilkan ini adalah sebuah *map* yang bertugas untuk mengumpulkan seluruh informasi terkait representasi *vector* dari gambar tersebut (data *input*). Setelah semua informasi terkumpul pada *feature map*, informasi tersebut diolah oleh *Region Proposed Network* (RPN) untuk memprediksi area gambar yang dianggap sebagai sebuah *object* orang dan memprediksi *bounding-box* pada area yang dianggap sebagai *object* orang. Setelah itu, informasi awal yang terkumpul pada *feature map* dan informasi *feature map* yang sudah diolah di RPN diteruskan ke ROI *Pooling* (*Region of Interest*). Pada tahapan RoI *Pooling* ini terjadi proses untuk mengekstraksi semua informasi yang didapat dari *feature map* dan RPN agar dapat diproses lebih lanjut agar dapat mengklasifikasi *object* yang didapat dan memberikan *bounding-box* pada *object* yang terdeteksi. Pada penelitian ini, proses konvolusi menggunakan model *transfer learning* dengan arsitektur VGG-16, sehingga *object* yang di deteksi hanya *object* orang.

Confuison Matrix

Confusion matrix dapat juga disebut sebagai *error matrix*, artinya memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. *Confusion matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji dimana nilai sebenarnya diketahui.



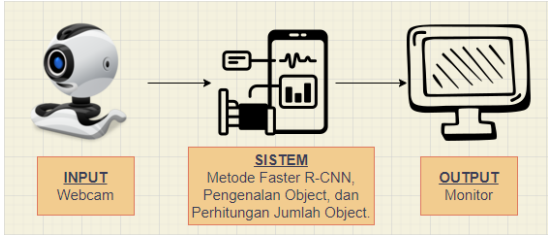
Gambar 4. Tabel *Confusion Matrix*
(Sumber: https://miro.medium.com/max/1400/1*IzN36IDL95ASZcV7g_KRUg.jpeg/)

Dari gambar 4 di atas terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Berikut ini penjelasan 4 istilah dari tabel *confusion matrix* dengan contoh yang sesuai dengan penelitian ini:

a. *True Positive* (TP), merupakan data positif yang diprediksi benar oleh sistem (model). Contohnya terdapat *object* orang dan sistem (model) mendeteksi bahwa *object* tersebut adalah *object* orang.
b. *True Negative* (TN), merupakan data negatif yang diprediksi benar oleh sistem (model). Contohnya terdapat *object* barang dan sistem (model) tidak mendeteksi *object* barang tersebut.
c. *False Positive* (FP), merupakan data negatif tetapi sistem (model) mendeteksi sebagai data positif. Contohnya terdapat *object* barang yang terdeteksi oleh sistem (model) sebagai *object* orang.
d. *False Negative* (FN), merupakan data positif tetapi sistem (model) mendeteksi sebagai data negatif. Contohnya terdapat *object* orang yang terdeteksi oleh sistem (model) sebagai *object* barang.

METODE PENELITIAN

Perancangan Perangkat Keras (Hardware)



Gambar 5. Model perancangan Hardware

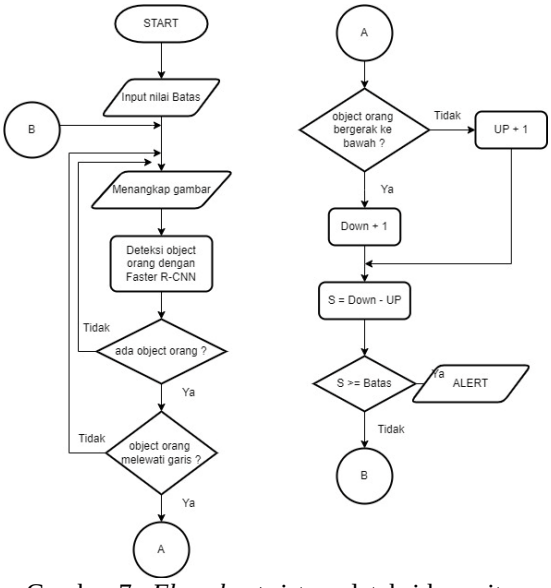
Gambar 5 di atas merupakan perancangan *hardware* dengan inputan berasal langsung dari kamera. Inputan diproses oleh sistem menggunakan metode Faster R-CNN agar dapat mengenali object dan menghasilkan object berupa orang. *Output* yang ditampilkan pada monitor adalah hari, tanggal dan waktu, jumlah orang masuk, keluar, dan total jumlah orang di dalam ruangan.

Gambar 6. Tampilan *output* pada layar monitor

Dilihat dari gambar 6 di atas, dijelaskan bahwa *output* yang ditampilkan sistem pada layar

monitor adalah tanggal dan waktu, Jumlah batas orang di dalam ruangan yang sudah ditentukan, jumlah orang yang masuk, keluar, dan jumlah orang yang berada di dalam ruangan. Selain itu terdapat *center line* berwarna merah pada output tersebut, yang berfungsi sebagai batas sekaligus garis perhitungan orang yang berjalan masuk dan orang yang berjalan keluar ruangan.

Flowchart Sistem



Gambar 7. *Flowchart* sistem deteksi kapasitas orang

Berdasarkan gambar 7 di atas dapat dijelaskan bahwa pengguna sistem wajib memasukkan nilai maksimal jumlah orang dalam suatu ruangan yang disimpan di variable **batas**. Kemudian sistem mengambil data input berupa video dari kamera, yang kemudian diproses oleh metode Faster R-CNN untuk mendeteksi *object* orang. Apabila sistem tidak mendeteksi *object* orang, maka sistem melakukan proses *looping* mulai dari sistem menangkap gambar pada video yang ditangkap, proses ini terus dilakukan hingga sistem mendeteksi adanya *object* orang. Setelah sistem mendeteksi adanya *object* orang, sistem mendeteksi apakah *object* orang telah melewati *center line*. Jika *object* orang tidak melewati *center line*, maka sistem melakukan proses *looping* kembali pada proses menangkap gambar pada video. Tetapi jika *object* orang terdeteksi telah melewati *center line*, sistem melakukan proses *tracking* pada *object* orang yang sudah terdeteksi untuk menentukan apakah *object* orang berjalan ke bawah *line* (masuk ke dalam ruangan) melewati *center line* atau *object* orang berjalan ke atas (keluar dari ruangan). Apabila *object* orang terdeteksi sistem berjalan ke bawah dan telah melewati *center line*, maka secara otomatis nilai dari variabel **Down** bertambah 1 dan seterusnya. Jika sistem mendeteksi *object* orang berjalan ke atas dan telah melewati *center line*, maka nilai dari variabel **UP** bertambah 1 secara otomatis, begitu juga seterusnya.

Setelah itu sistem menjalankan proses perhitungan dengan menghitung jumlah orang yang berada di dalam ruangan dengan cara nilai dari variabel **Down** dikurangi nilai dari variabel **Up** kemudian hasilnya disimpan divariabel **S** dan menampilkannya pada monitor. Kemudian sistem melakukan proses perbandingan apakah nilai dari variabel **S** melebihi nilai variabel **Batas** atau tidak. Jika nila variabel **S** melebihi nilai variabel **Batas,** maka muncul **ALERT** pada monitor. Apabila nilai variabel **S** tidak melebihi nilai variabel **Batas,** maka sistem melakukan *looping* mulai dari proses mendeteksi *object* orang.

HASIL DAN PEMBAHASAN

Pengujian Deteksi Orang

Tujuan pengujian deteksi orang ini dilakukan untuk mengetahui apakah sistem yang dibuat dapat mendeteksi orang yang berada dalam frame tersebut. Pengujian ini dilakukan dengan menjalankan program yang menggunakan metode Faster R-CNN untuk mendeteksi orang, kemudian melakukan pencatatan secara manual apakah orang tersebut terdeteksi atau tidak terdeteksi. Orang dapat dicatat terdeteksi jika terdapat *bounding box* beserta ID yang melekat pada orang tersebut, hal ini dapat dilihat di layar output ketika sistem dijalankan.

Tabel 1. Hasil uji coba deteksi orang

Uji ke-	Orang ke-	Kondisi Pergerakan			Terdeteksi	Tidak Terdeteksi
		Lambat	Normal	Cepat		
1	1	√	-	-	√	-
2	1	-	√	-	√	-
3	1	-	-	√	√	-
4	2	√	-	-	√	-
5	2	-	√	-	√	-
6	2	-	-	√	√	-
7	3	√	-	-	-	√
8	3	-	√	-	√	-
9	3	-	-	√	√	-
10	4	√	-	-	√	-
11	4	-	√	-	√	-
12	4	-	-	√	√	-
13	5	√	-	-	√	-
14	5	-	√	-	-	√
15	5	-	-	√	√	-
16	6	√	-	-	√	-
17	6	-	√	-	√	-
18	6	-	-	√	√	-

Uji ke-	Orang ke-	Kondisi Pergerakan			Terdeteksi	Tidak Terdeteksi
		Lambat	Normal	Cepat		
19	7	√	-	-	-	√
20	7	-	√	-	-	√
21	7	-	-	√	√	-
22	8	√	-	-	√	-
23	8	-	√	-	√	-
24	8	-	-	√	√	-
25	9	√	-	-	√	-
26	9	-	√	-	-	√
27	9	-	-	√	√	-
28	10	√	-	-	√	-
29	10	-	√	-	-	√
30	10	-	-	√	-	√
Tingkat Akurasi					77%	23%

Dari hasil pengujian pada tabel 1 dapat diketahui bahawa sistem yang dibuat dengan menggunakan metode Faster R-CNN, mamapu mendeteksi orang yang berada dalam frame tersebut, dengan tingkat akurasi data 80% yang diperoleh dari perhitungan :

$$\text{Akurasi Data} = \frac{23}{30} * 100\% \quad (1)$$

$$= 77\%$$

Dapat disimpulkan juga, bahwa sistem tidak dapat mendeteksi orang, apabila orang tersebut berjalan terlalu cepat atau setengah berlari sebesar 23%.

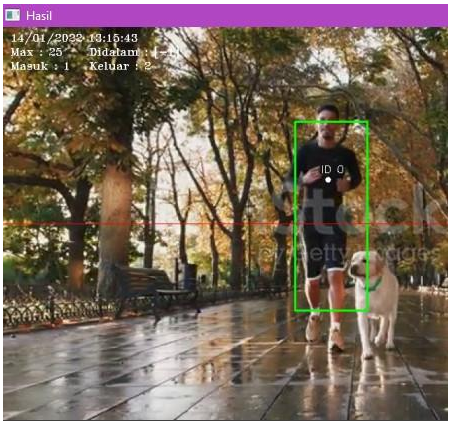
Pengujian Deteksi Object

Tujuan pengujian ini dilakukan untuk mengetahui apakah sistem yang dibuat mampu mendeteksi *object* orang saja atau *object* lain, pengujian ini perlu dilakukan untuk memastikan bahwasannya sistem hanya mendeteksi *object* orang saja, sedangkan *object* lain tidak terdeteksi. Untuk melaksanakan pengujian ini, penulis mengerjakan dengan cara menyiapkan 30 video - untuk media *input* yang sudah didownload dari *link* www.*youtube.com* dan *www.istockphoto.com*, Kemudian menjalankan program yang sudah dibuat menggunakan metode Faster R-CNN untuk mendeteksi *object*. Setelah itu penulis mengamati *layout output* pada sistem untuk mengetahui apakah *object* lain (barang) selain *object* orang dapat terdeteksi atau tidak oleh sistem serta melakukan perhitungan secara manual untuk mendapatkan jumlah *object* orang dan *object* barang yang terdeteksi, *object* orang yang terdeteksi sebagai *object* barang, serta *object* barang yang terdeteksi sebagai *object* orang. Perhitungan dilakukan dengan mengamati jumlah ID *Object* yang terdeteksi pada *layout output*.

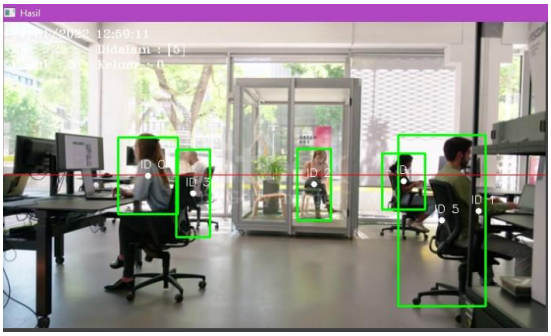
Tabel 2. Hasil uji deteksi *object*

Uji Ke-	Data Uji	Jumlah Orang Pada Video	Hasil Deteksi *Object*				
			Jumlah Orang Terdeteksi	Jumlah Orang Tidak Terdeteksi	Jumlah Barang Terdeteksi	Jumlah Orang Terdeteksi Barang	Jumlah Barang Terdeteksi Orang
1	Video 1	15	14	1	0	0	0
2	Video 2	28	20	8	0	0	0
3	Video 3	45	39	6	0	0	0
4	Video 4	25	18	7	0	0	0
5	Video 5	5	5	0	0	0	0
6	Video 6	20	19	1	0	0	0
7	Video 7	10	9	1	0	0	0
8	Video 8	45	34	11	0	0	0
9	Video 9	82	70	12	0	0	0
10	Video 10	1	1	0	0	0	0
11	Video 11	36	27	9	0	0	0
12	Video 12	5	4	1	0	0	0
13	Video 13	13	9	4	0	0	0
14	Video 14	6	6	0	0	0	0
15	Video 15	1	1	0	0	0	0
16	Video 16	4	4	0	0	0	0
17	Video 17	44	31	13	0	0	0
18	Video 18	31	23	8	0	0	0
19	Video 19	3	3	0	0	0	0
20	Video 20	4	4	0	0	0	0
21	Video 21	18	14	4	0	0	0
22	Video 22	3	3	0	0	0	0
23	Video 23	11	7	4	0	0	0
24	Video 24	17	13	4	0	0	0
25	Video 25	12	9	3	0	0	0
26	Video 26	26	19	7	0	0	0
27	Video 27	14	10	4	0	0	0
28	Video 28	3	2	1	0	0	0
29	Video 29	1	1	0	0	0	0
30	Video 30	25	20	5	0	0	0
Jumlah		553	439	114	0	0	0

Berdasarkan hasil pengujian pada Tabel 2. Hasil Uji Deteksi *Object* diatas, pada baris 1 uji ke-1 dari data uji video 1 menunjukkan bahwa jumlah *object* orang yang terdeteksi ada 14, *object* barang yang terdeteksi ada 0, *object* orang yang terdeteksi sebagai *object* barang ada 0 dan *object* barang yang terdeteksi sebagai *object* orang ada 0, sehingga dapat disimpulkan bahwa sistem yang dibuat penulis mampu untuk mendeteksi *object* orang saja.



Gambar 8. Hasil deteksi *object* sistem pertama



Gambar 9. Hasil deteksi *object* sistem kedua

Pengujian Tingkat Akurasi Perhitungan Data Report

Tujuan pengujian ini dilakukan dengan membandingkan perhitungan yang dilakukan oleh sistem denga data report yang disimpan oleh sistem berbentuk file *spreadsheet* dalam kurung waktu kurang lebih 20 menit dengan pengujian dilakukan sebanyak 30x. Pengujian ini dilakukan dengan cara menjalankan program yang sudah dibuat dengan menggunakan metode Faster R-CNN dengan file berekstensi *.py*. Setelah itu merekam layar output sistem saat program dijalankan menggunakan aplikasi OSBS Studio. Setelah itu melakukan perbandingan hasil perhitungan dari video rekaman dengan hasil report yang sudah tersimpan dalam bentuk file *spreadsheet* (.csv). Hasil Pengujian Tingkat Akurasi Perhitungan Data Report dapat dilihat pada table dibawah ini.

Tabel 3. Hasil uji akurasi data report

Uji Ke-	Hasil perhitungan sistem			Data Report		
	Masuk	Keluar	Di dalam	Masuk	Keluar	Di dalam
1	2	1	1	2	1	1
2	3	1	2	3	1	2
3	8	3	5	8	3	5
4	7	3	4	7	3	4
5	9	3	6	9	3	6
6	14	4	10	14	4	10
7	13	3	10	13	3	10
8	10	3	7	10	3	7
9	11	3	8	11	3	8
10	12	3	9	12	3	9
11	13	4	9	13	4	9
12	7	4	3	7	4	3
13	8	3	5	8	3	5
14	9	4	5	9	4	5
15	5	2	3	5	2	3
16	5	3	2	5	3	2
17	7	3	4	7	3	4
18	2	0	2	2	0	2
19	4	0	4	4	0	4
20	4	1	3	4	1	3
21	6	2	4	6	2	4
22	6	3	3	6	3	3
23	6	1	5	6	1	5
24	10	4	7	10	4	7
25	10	0	10	10	0	10
26	8	5	3	8	5	3
27	7	2	5	7	2	5
28	7	4	3	7	4	3
29	11	5	6	11	5	6
30	13	7	6	13	7	6

Dari hasil pengujian akurasi data report yang ditampilkan di tabel 3 di atas, dapat disimpulkan bahawa perhitungan yang tercatat di aplikasi dan perhitungan yang terdapat pada data report file *spreadsheet* (.csv) mempunyai nilai yang sama tanpa adanya angka selisih, sehingga dapat disimpulkan bahwa tingkat akurasi data report adalah 100%.

Masuk	Keluar	Di Dalam
1	1	3
2	2	
3	3	
4		
5		
6		

Masuk	Keluar	Di Dalam
1	1	3
2	2	
3	3	
4		
5		
6		

Gambar 10. Contoh tampilan hasil data report

Perhitungan Confusion Matrix

Perhitungan *Confusion Matrix* ini dilakukan dengan tujuan untuk mendapatkan nilai *Rate accuracy, Rate loss, Precission, Recall,* F1-*Score*. Prosedur perhitungan *confusion matrix* dilakukan dengan cara melakukan perhitungan berdasarkan pada tabel 2. Hasil Pengujian Deteksi *Object*, memanfaatkan *Microsoft Excel*, membuat tabel *Confusion Matrix* dan menghitung nilai *Performance Metric* dari *Confusion Matrix* yaitu, *Rate accuracy, Rate loss, Precission, Recall,* F1-*Score* berdasarkan data pada tabel 2. Hasil Pengujian Deteksi *Object*.

Tabel 4. Confusion Matrix

	CONFUSION MATRIX	
	Actual Values	
Predict Values	TP	FP
	439	0
	FN	TN
	0	0

Keterangan:

- **TP** = Jumlah *object* orang terdeteksi.
- **TN** = Jumlah *object* barang terdeteksi.
- **FP** = Jumlah *object* barang terdeteksi sebagai *object* orang.
- **FN** = Jumlah *object* orang terdeteksi sebagai *object* barang.

Tabel 4 di atas adalah tabel *confusion matrix* yang sudah dibuat oleh penulis dengan berdasarkan data pada Tabel 2 Hasil Pengujian Deteksi *Object*. Dapat dilihat dari Tabel 4.4 diatas bahwa TP bernilai 439 sedangkan TN, FP dan FN bernilai 0. Berikut ini hasil perhitungan *performance metric* dari Tabel 4 di atas:

- *Accuracy*

  Untuk mendapatkan nilai *accuracy*, penulis menghitung menggunakan rumus (2), sehingga didapatkan hasil:

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100\% \quad (2)$$

$$= \frac{439 + 0}{439 + 0 + 0 + 0} \times 100\%$$

$$= 100\%$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis mampu mendeteksi *object* orang saja dengan nilai keakuratan sebesar 100%.

- *Precision*

  Untuk mendapatkan nilai *precision*, penulis menghitung menggunakan rumus (3), sehingga didapatkan hasil:

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (3)$$

$$= \frac{439}{439 + 0}$$

$$= 1$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *precision* sebesar 1.

- *Recall*

  Untuk mendapatkan nilai *recall*, penulis menghitung menggunakan rumus (4), sehingga didapatkan hasil:

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

$$= \frac{439}{439 + 0}$$

$$= 1$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *recall* sebesar 1.

- F1 – *Score*
  Untuk mendapatkan nilai F1 - *Score*, penulis menghitung menggunakan rumus (5), sehingga didapatkan hasil:

$$\text{F1} - Score = 2 \times \frac{Precission \times Recall}{Precission + Recall} \quad (5)$$

$$= 2 \times \frac{1 \times 1}{1 + 1}$$

$$= 1$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai F1 - *Score* sebesar 1.

- *Rate Accuracy*
  Untuk mendapatkan nilai *rate accuracy*, penulis menghitung menggunakan rumus (6), sehingga didapatkan hasil:

$$Rate\ Accuracy = \frac{\text{Jumlah } object \text{ orang terdeteksi}}{\text{Jumlah } object \text{ orang keseluruhan pada video}} \times 100\% \quad (6)$$

$$= \frac{439}{553} \times 100\%$$

$$= 79\%$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *rate accuracy* sebesar 79% dalam mendeteksi *object* orang.

- *Rate Loss*
  Untuk mendapatkan nilai *rate loss*, penulis menghitung menggunakan rumus (7), sehingga didapatkan hasil:

$$Rate\ Loss = 100\% - Rate\ Accuracy \quad (7)$$

$$= 100\% - 79\%$$

$$= 21\%$$

  Dari hasil perhitungan diatas dapat diketahui bahawa sistem yang dibuat oleh penulis memiliki nilai *rate loss* sebesar 21% dalam mendeteksi *object* orang.

KESIMPULAN

Dari hasil pengujian yang dilakukan oleh penulis untuk penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Hasil pengujian deteksi *object* pada penelitian ini membuktikan bahwa sitem yang dibuat sudah mampu untuk mendeteksi *object* orang saja.
2. Hasil pengujian sistem yang dibuat pada penelitian ini mampu mendeteksi tiga macam pergerakan orang, yaitu: berjalan lambat, berjalan normal dan berjalan cepat dengan tingkat akurasi 77%, sedangkan sistem tidak dapat mendeteksi beberapa pergerakan orang pada kondisi berjalan lambat dan berjalan cepat sebasar 23%.
3. Tingkat akurasi perhitungan sistem untuk jumlah orang yang masuk, keluar dan jumlah orang yang berada di dalam ruangan dengan hasil data *reporting* yang disimpan dalam bentuk file *spreadsheet* (*.csv*) memiliki kesesuaian data 100%.
4. Data *reporting* pada sistem ini dicatat setiap detik dalam waktu kurang lebih 20 menit.
5. Sistem (model) yang dibuat oleh penulis memiliki nilai *Accuracy* sebesar 100%, memiliki nilai *Precision* sebesar 1, nilai *Recall* sebesar 1, nilai F1 – *Score* sebesar 1, nilai *Rate Accuracy* sebesar 79% dan *Rate Loss* sebesar 21%.

DAFTAR PUSTAKA

Alamsyah, D., & Pratama, D. (2019). Deteksi Ujung Jari menggunakan Faster-RCNN dengan Arsitektur Inception v2 pada Citra Derau. *JuSiTik : Jurnal Sistem Dan Teknologi Informasi Komunikasi*, *2*(1), 1. https://doi.org/10.32524/jusitik.v2i1.435

Anggara, T., Irawan, B., Setianingsih, C., & Telkom, U. (n.d.). *HUMAN DETECTION COUNTING WITH FASTER R-CNN ARCHITECTURE FOR*.

Elekomp. (2018). *Pengertian Webcam Beserta Fungsi dan Cara Kerjanya - Elektronika dan Komputer*. Elekkomp. https://elekkomp.blogspot.com/2018/10/pengertian-webcam-beserta-fungsi-dan.html

Kandir, N. (2016). *Mengenal OpenCV dan Python serta Kaitan Keduanya – Oase Ilmu Multimedia dan Keislaman*. Norkandirblog. https://norkandirblog.wordpress.com/2016/12/23/mengenal-opencv-dan-python/

Kemenkes RI. (2020). KMK Nomor Hk.01.07/Menkes/328/2020 Tentang Panduan Pencegahan Dan Pengendalian Corona Virus Disease 2019 (Covid-19) Di Tempat Kerja. In *Menteri Kesehatan Republik Indonesia* (Vol. 2019, pp. 1–39).

Plazakamera.com. (n.d.). *Logitech C270 HD Webcam Harga Murah dan Spesifikasi*. Plazakamera.Com. Retrieved September 19, 2021, from https://www.plazakamera.com/shop/logitech-c270-hd-webcam/

Rismiyati, R., & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, *18*(1), 37. https://doi.org/10.31315/telematika.v18i1.4025

Sena, S. (2018). *Pengenalan Deep Learning Part 8 : Gender Classification using Pre-Trained Network (Transfer Learning) | by Samuel Sena | Medium*. Medium.Com. https://medium.com/@samuelsena/pengenalan-deep-learning-part-8-gender-classification-using-pre-trained-network-transfer-37ac910500d1

Sunario Megawan, & Wulan Sri Lestari. (2020). Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *9*(3), 261–267. https://doi.org/10.22146/.v9i3.231