

DETEKSI JUMLAH MANUSIA DENGAN ARSITEKTUR *FASTER R-CNN* UNTUK KENDALI SUHU *AIR CONDITIONER (AC)* SENTRAL

HUMAN DETECTION COUNTING WITH FASTER R-CNN ARCHITECTURE FOR CENTRAL AIR CONDITIONER (AC) TEMPERATURE CONTROL

Teja Anggara¹, Budhi Irawan^{2,3}, Casi Setianingsih³

^{1,2,3} Universitas Telkom, Bandung

tejailord@student.telkomuniversity.ac.id¹, budhiirawan@telkomuniversity.ac.id²,
setiacasie@telkomuniversity.ac.id³

Abstrak

Indonesia merupakan negara yang memiliki iklim tropis, dimana ada 2 musim yang terjadi secara bergantian, yaitu musim kemarau dan musim hujan. Dengan terjadinya musim kemarau di Indonesia, maka terjadi peningkatan suhu yang sangat tinggi, terutama di daerah perkotaan. Maka dari itu, masyarakat pun menggunakan Air Conditioner (AC) sebagai alternatif pendingin udara di tempat tinggal mereka, maupun gedung perkantoran. Penggunaan AC secara konstan dan tidak teratur dapat menyebabkan borosnya penggunaan energi listrik. Maka dari itu, diperlukan sistem otomatis yang dapat mengatur penggunaan daya AC berdasarkan jumlah orang didalamnya. Sistem deteksi manusia menggunakan image processing menjadi salah satu cara untuk mendeteksi keberadaan manusia dan menghitung jumlahnya dalam ruangan secara real-time. Sistem yang dirancang menggunakan arsitektur Faster R-CNN dengan backbone Inception v2 sebagai modul CNN nya. Sistem tersebut dirancang dalam bahasa pemrograman Python, yang ditanam dalam mikrokomputer Raspberry Pi 4 yang terhubung dengan webcam USB. Akurasi deteksi yang didapat bervariasi, dan akurasi yang tertinggi didapatkan sebesar 100% dalam beberapa pengujian.

Abstract

Indonesia is a country that has a tropical climate, where there are 2 seasons that occur alternately, namely the dry season and the rainy season. With the onset of the dry season in Indonesia, there is a very high temperature increase, especially in urban areas. Therefore, people also use Air Conditioner (AC) as an alternative to air conditioning in their homes, as well as office buildings. Constant and irregular use of air conditioning can lead to wasteful use of electrical energy. Therefore, an automatic system is needed that can regulate the use of AC power based on the number of people in it. Human detection system using image processing is one way to detect human presence and count the number in the room in real-time. The system is designed using Faster R-CNN architecture with Inception v2 backbone as the CNN module. The system is designed in the Python programming language, embedded in a Raspberry Pi 4 microcomputer connected to a USB webcam. The detection accuracy obtained varies, and the highest accuracy is obtained at 100% in several tests.

Keywords: *Object Detection, CNN, Faster R-CNN*

1. Pendahuluan

Indonesia merupakan negara yang banyak menerima sinaran cahaya matahari, dikarenakan posisinya yang ada di dalam garis khatulistiwa, karena matahari sepanjang tahun berada di atasnya [1]. Maka dari itu, tidak heran jika Indonesia memiliki iklim tropis, yang dimana terdapat dua musim, yaitu: musim kemarau dan musim hujan. Banyak daerah yang memiliki suhu tinggi karena langsung terpapar oleh sinar matahari, terutama perkotaan. Dari kejadian itu, maka banyak orang yang merasa panas saat berada diluar ruangan, dan alternatif yang digunakan masyarakat adalah penggunaan AC pada setiap

gedung. Salah satunya adalah AC Sentral, yang digunakan pada gedung – gedung yang memiliki kapasitas orang yang banyak. Yang menjadi permasalahan adalah penggunaan AC terus menerus dapat meningkatkan penggunaan daya, sehingga berakibat di biaya penggunaan listrik yang membengkak. Hal itu terjadi saat ada/tidaknya orang dalam suatu ruangan, karena AC pada umumnya tidak memerhatikan jumlah orang di dalam ruangan, sehingga penggunaan daya nya konstan.

Dari permasalahan diatas, diperlukan sebuah sistem yang dapat menjadi solusi dari permasalahan tersebut. Sistem otomatis yang dapat mengatur suhu AC dengan melihat jumlah orang yang ada dalam suatu ruangan, dan penggunaan AC dapat disesuaikan dengan kebutuhan, sehingga dapat menghemat energi dan biaya.

Maka dari itu, pada penelitian tugas akhir ini yang berjudul “DETEKSI JUMLAH MANUSIA MENGGUNAKAN ARSITEKTUR FASTER R-CNN UNTUK KENDALI SUHU AC SENTRAL” dilakukan dengan tujuan untuk menjawab permasalahan diatas. Pada penelitian ini, digunakan *library OpenCV* sebagai object detector, dan *Tensorflow* sebagai *library* untuk machine learning, yang nantinya akan digunakan juga algoritma klasifikasi *Faster R-CNN* untuk pendeteksian manusia secara *real-time*, dan tentunya menghitung jumlah orang yang ada di dalam suatu ruangan menggunakan kamera.

2. Dasar Teori

2.1. Pengolahan Citra Digital

Pengolahan Citra Digital adalah pengolahan data berbentuk citra dua dimensi pada komputer. Yang diolah adalah Citra Digital, yang berarti sebuah deretan array yang bernilai bilangan real dan maupun kompleks. [2]

Dalam buku ini, penulis menjelaskan bahwa citra dapat direpresentasikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom. X dan Y pada fungsi $f(x,y)$ adalah koordinat spasial, dan intensitas/tingkat keabuan dari citra pada titik tersebut dinamakan amplitudo f, di titik $f(x,y)$. Sebuah citra dapat dikatakan citra digital jika nilai x,y dan nilai amplitudo f secara keseluruhan memiliki nilai diskrit dan berhingga. [2]

Citra digital dapat diwujudkan dalam persamaan matriks berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Persamaan matriks Citra Digital

dimana:

M = baris,

N = kolom, dengan x dan y adalah koordinat spasial.

2.2. Object Detection

Object Detection adalah sebuah metode yang memungkinkan user untuk mendeteksi objek yang terlihat di siaran langsung yang berupa video [3]. Programnya dapat disajikan dalam bahasa pemrograman *Python* menggunakan kamera seperti *Webcam* yang dapat digunakan untuk menyajikan siaran langsung berupa video. Jika metode ini menggunakan model yang sudah dilatih, maka metode ini dapat medeteksi objek tertentu, dapat menghitung objek yang ada pada siaran berupa video.

2.3. Object Counting

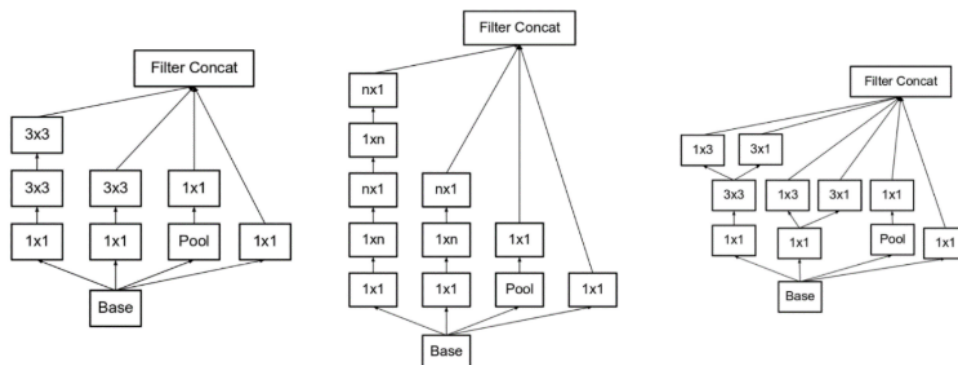
Object Counting merupakan teknik yang sangat penting dalam pemrosesan citra. Teknik ini memungkinkan untuk menghitung jumlah dari objek yang dideteksi. Biasanya, suatu sistem memiliki threshold sebagai syarat agar objek akan dideteksi atau tidak [4].

2.4. Convolutional Neural Network

Convolutional Neural Network atau juga bisa disebut *CNN*, adalah sebuah metode yang biasa digunakan untuk menganalisa citra. Metode ini biasa digunakan untuk memproses data yang berbentuk *multiple array*, contohnya adalah sebuah warna dari citra [5]. Seperti namanya, metode ini memiliki proses matematika yang dinamakan konvolusi pada *layernya* sehingga dinamakan *CNN*.

2.5. Inception V2

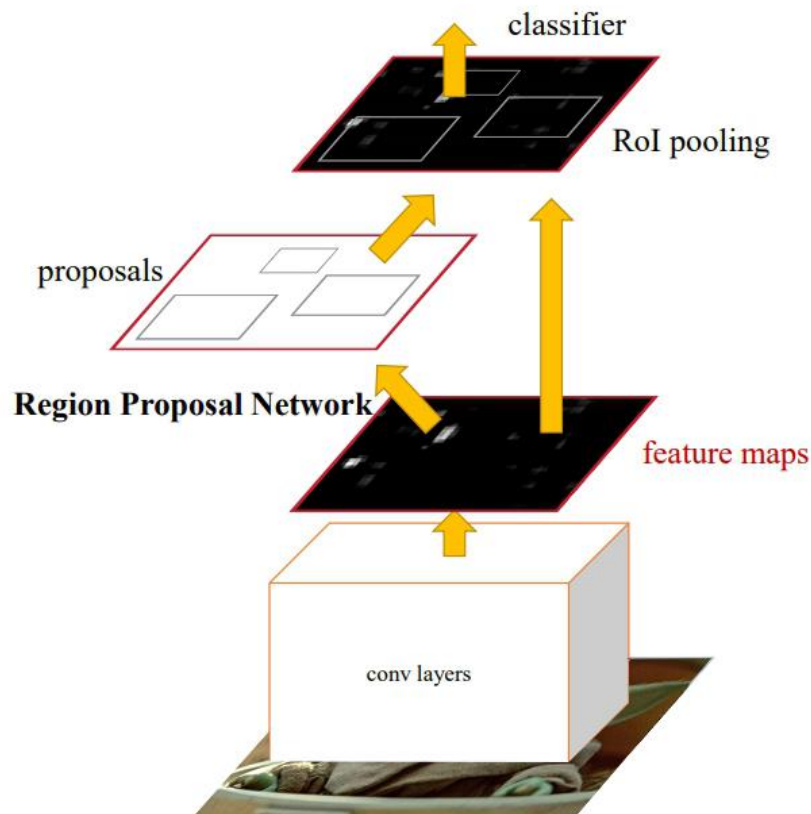
Inception v2 adalah modul yang di desain untuk mengurangi kompleksitas dari jaringan konvolusi [6]. Modul ini membuat jaringan konvolusi menjadi lebih lebar, bukan lebih dalam. Modul *Inception v2* memiliki tiga jenis tipe yang berbeda, yang bernama modul A, B, C.



2.6. Faster R-CNN

Faster R-CNN merupakan suatu sistem pendeteksi objek, yang terdiri dari 2 modul utama, yaitu: *Deep Fully Convolutional Network* yang mengandung *Region Proposed Network* (RPN), dan modul kedua adalah detektornya, yaitu detektor dari *Fast R-CNN* [7][8]. Metode ini merupakan pengembangan dari metode sebelumnya, yaitu *Fast R-CNN*. Di dalam *Faster R-CNN*, fitur *selective search* menggunakan RPN, maka dari itu inilah yang menjadi arsitektur utama *Faster R-CNN* [9].

Gambar Ilustrasi dari Faster R-CNN



2.6.1. Region Proposed Network

Faster R-CNN menggunakan RPN untuk menentukan *boundary box* [10]. RPN mengambil citra dalam ukuran apapun sebagai input dan menghasilkan sebuah *rectangular object proposals*, masing-masing memiliki nilai objek. Proses tersebut disajikan dengan *fully Convolutional network* [8]. *Region Proposal* dapat di ciptakan melalui RPN network, yang menentukan titik yang dimiliki oleh latar depan dan latar belakang melalui *softmax*. Dan sebuah *proposal* yang akurat dapat dicapai dengan menggunakan *bounding box regression correction anchors* [11].

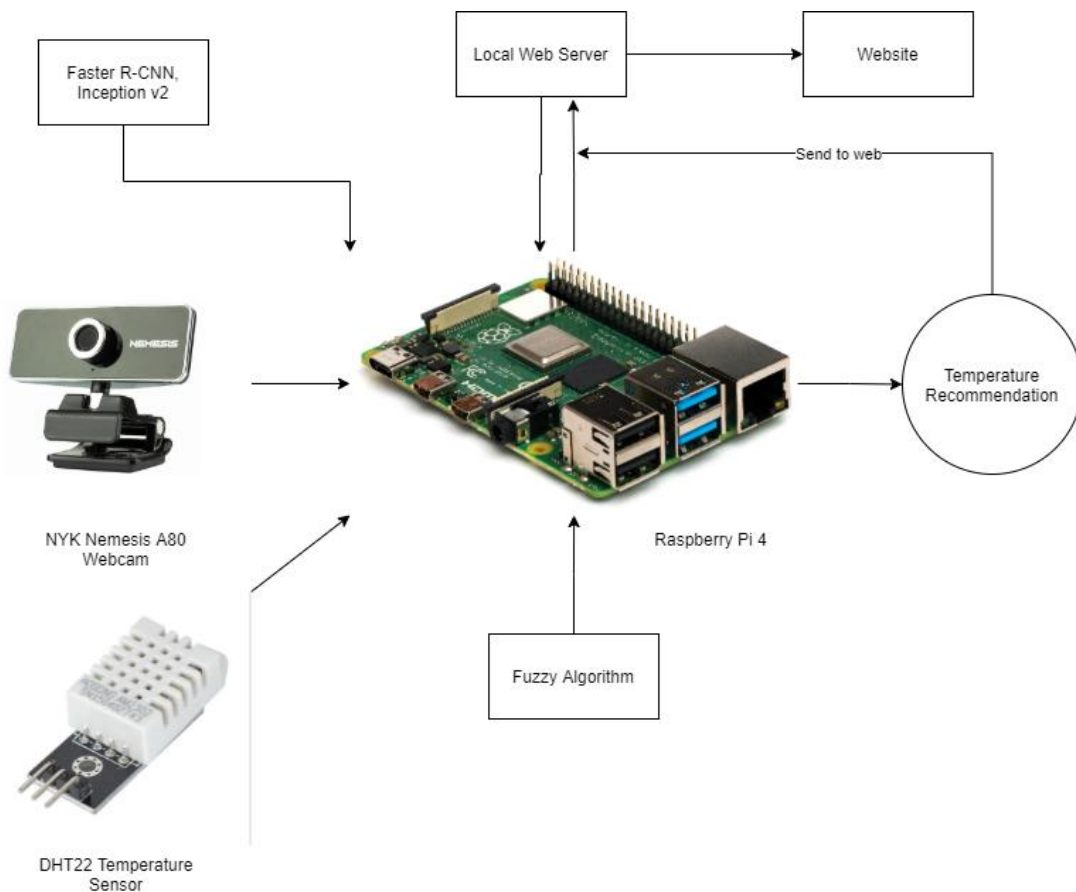
2.6.2. Region of Interest

Proses mengkonversi fitur di region yang diproyeksikan dari citra dalam berbagai ukuran. Misalnya input citra memiliki ukuran $H \times W$, diubah menjadi jendela kecil, $h \times w$. Region dari input dibagi menjadi grid $H \times W$, dan diterapkan *Max Pooling* di setiap grid [10].

3. Perancangan Sistem

3.1. Desain Sistem

Desain sistem yang dirancang untuk mendeteksi jumlah manusia di dalam ruangan dilakukan dengan cara mendeteksi keberadaan manusia didalam stream video yang diambil oleh kamera. Pada desain sistem ini, akan digunakan *USB Webcam*, yang nantinya akan dihubungkan ke Raspberry Pi 4 yang berfungsi untuk melakukan pengolahan gambar. Berikut adalah gambaran sistem yang akan diusulkan:



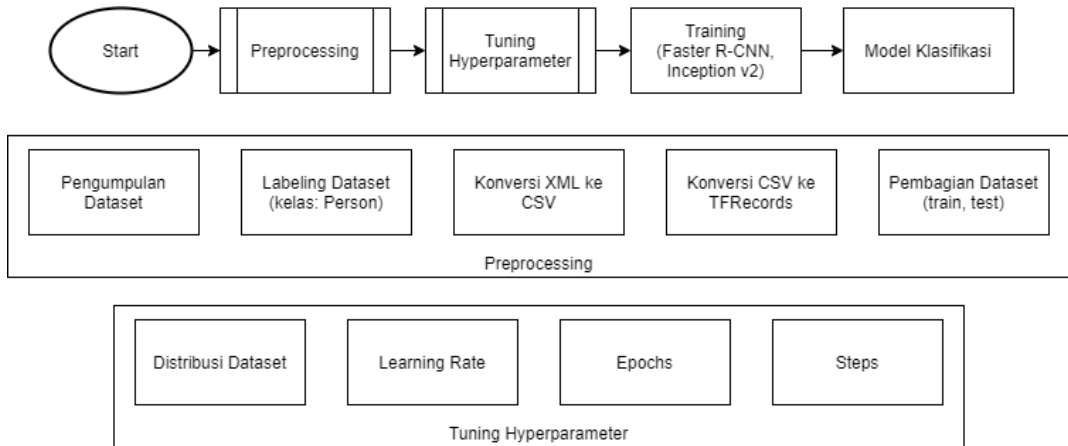
Gambar Desain Sistem

3.2. Blok Diagram

Blok Diagram sistem yang dirancang memerlukan proses training untuk dapat mendeteksi dan menghitung jumlah manusia secara real-time. Untuk melakukan training, hanya diperlukan dataset dengan jumlah 1 kelas saja, yaitu 'person'. Dataset yang digunakan merupakan dataset custom yang berjumlah 500 gambar, yang terdiri dari 10 kategori. Setiap kategori gambar memiliki jumlah sebanyak 50, dan kategori dibuat berdasarkan jumlah (kategori 1 orang, kategori 2 orang, dst).

Setelah melakukan preprocessing terhadap dataset yang digunakan, selanjutnya yang dilakukan adalah proses training, yang menggunakan model Faster R-CNN yang menggunakan backbone Inception V2. Sebelum melakukan proses training, diperlukan tuning hyperparameter sesuai kebutuhan. Parameter – parameter tersebut adalah: Distribusi dataset, Learning Rate, Epochs, dan Steps.

Setelah terbentuknya model klasifikasi dari proses training, maka model tersebut dapat digunakan untuk klasifikasi objek berupa manusia. Sistem pun akan menghitung jumlahnya secara real-time. Pada pengujian, ada parameter - parameter yang perlu di analisa, seperti Akurasi deteksi dan Waktu Komputasi.



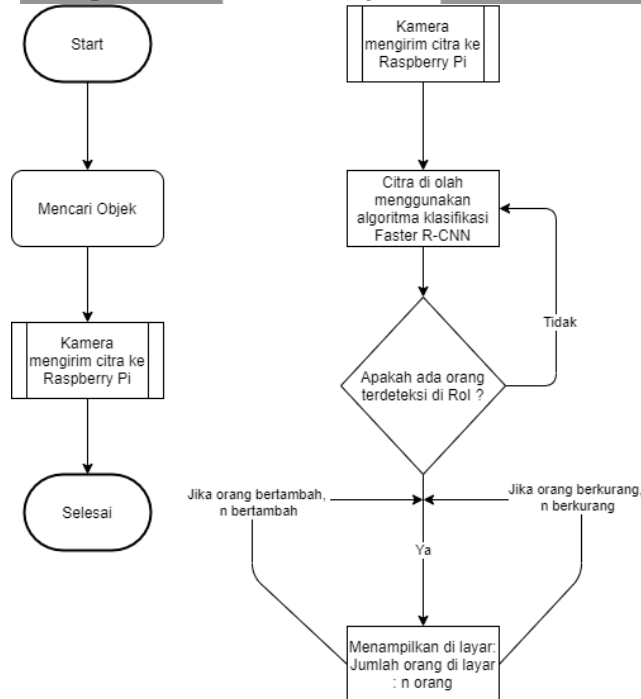
Gambar Diagram Blok

3.3. Diagram Alur

Diagram Alur dimulai dengan sistem mencari objek pada citra yang didapatkan dari webcam yang dihubungkan ke Raspberry Pi. Setiap frame yang didapatkan dari kamera akan langsung dikirim ke Raspberry Pi untuk selanjutnya akan dilakukan proses klasifikasi.

Jika terdapat objek berupa manusia pada citra yang di proses, maka pada layar Raspberry Pi akan ditampilkan jumlah manusia yang terdeteksi saat itu.

Jika tidak terdapat objek atau jumlah objek berkurang pada frame yang di proses, maka nilai jumlah manusia terdeteksi pun akan ikut berkurang.



Gambar Diagram Alur

4. Hasil Pengujian dan Analisis

4.1. Pengujian Akurasi Training

Sebelum dilakukan deteksi objek, diperlukan model klasifikasi yang dapat bekerja secara optimal, baik dari segi akurasi maupun performa, agar dapat melakukan deteksi objek secara real-time pada setiap jenis ruangan dan keramaian. Pengujian ini dilakukan dengan membuat model – model yang sudah dibuat melalui proses training yang memiliki parameternya masing – masing pada setiap pengujian. Ada 4 faktor yang memiliki pengaruh yang besar terhadap akurasi suatu model, yaitu:

4.1.1. Parameter: Dsistribusi Dataset

Pada pengujian distribusi dataset, telah dibuat 6 model dari variasi distribusi yang berbeda. Dimulai dari pembagian dataset 50% data latih dan 50% data tes hingga 95% data latih dan 5% data tes. Berikut adalah tabel dari hasil pengujian Distribusi Dataset.

No	Pengujian								
	Distribusi Dataset	TP	TN	FP	FN	Precision	recall	Accuracy	
1	50% – 50%	2566	74	386	187	87%	93%	82%	
2	60% – 40%	2585	78	359	168	89%	94%	83%	
3	70% – 30%	2608	79	326	145	89%	95%	85%	
4	80% – 20%	2653	73	315	100	89%	96%	87%	
5	90% – 10%	2668	74	340	85	89%	97%	86%	
6	95% – 5%	2667	88	246	86	91%	97%	89%	

4.1.2. Parameter: Learning Rate

Pada pengujian *Learning Rate*, akan diuji *model* terbaik pada pengujian sebelumnya, yaitu *model* dengan partisi *dataset* 95% : 5%. *Model* tersebut akan dijadikan acuan pada pengujian *Learning Rate*.

Pada pengujian *Learning Rate*, telah dibuat 4 model dengan variasi *Learning Rate* 2×10^{-5} hingga 2×10^{-2} . Berikut adalah tabel dari hasil pengujian *Learning Rate*.

No	Pengujian								
	Learning Rate	TP	TN	FP	FN	Precision	recall	Accuracy	
1	2×10^{-5}	2650	78	602	103	81%	96%	79%	
2	2×10^{-4}	2652	73	236	101	91%	96%	89%	
3	2×10^{-3}	2694	64	194	59	93%	98%	92%	
4	2×10^{-2}	0	100	0	2753	-	0		

4.1.3. Parameter: Epochs

Pada pengujian sebelumnya, didapatkan *model* terbaik dengan partisi *dataset* sebesar 95% : 5%, dan *Learning Rate* sebesar 0.002. *Model* tersebut akan dijadikan acuan pada pengujian ini.

Pengujian *Epochs* dilakukan dengan menggunakan 5 nilai *epochs* secara random. Nilai yang digunakan adalah: 1, 5, 10, 50, 100. Nilai – nilai tersebut mempengaruhi akurasi suatu model. Berikut adalah tabel dari hasil pengujian pada keempat model menggunakan parameter *epochs*.

No	Pengujian							
	Epochs	TP	TN	FP	FN	Precision	recall	Accuracy
1	1	2675	61	199	78	93%	97%	91%
2	5	2695	66	270	58	91%	98%	90%
3	10	2717	62	285	36	91%	99%	90%
4	50	2683	69	150	70	95%	97%	93%
5	100	2687	66	213	66	93%	98%	91%

4.1.4. Parameter: Steps

Pada pengujian sebelumnya, didapatkan *model* terbaik yaitu *model* dengan partisi data 95% : 5%, *Learning Rate* sebesar 0.002, dan *epochs* sebesar 50. *Model* tersebut akan dijadikan acuan pada pengujian ini.

Pengujian *steps* dilakukan dengan membuat 4 *model* dengan *steps* yang berbeda. Semakin besar *steps* yang digunakan, maka proses pembelajaran mesin akan semakin dalam sehingga estimasi error yang dihasilkan semakin kecil. Berikut adalah hasil dari pengujian *steps*.

No	Pengujian							
	Steps	TP	TN	FP	FN	Precision	recall	Accuracy
1	30000	2703	63	124	50	96%	98%	94%
2	50000	2712	65	118	41	96%	99%	95%
3	100000	2718	63	122	35	95%	99%	95%
4	200000	2707	70	91	46	97%	98%	95%

4.2. Pengujian Akurasi Deteksi

Setelah mendapatkan *model* dengan akurasi terbaik, pengujian selanjutnya adalah pengujian deteksi jumlah orang. Pengujian ini dilakukan dengan cara memasang sistem dengan sebuah web camera, dan menangkap citra langsung dari kamera. Citra yang ditangkap akan diproses secara real-time untuk mencari objek berupa manusia untuk setiap frame yang ditangkap. Pengujian ini bertujuan untuk menguji akurasi *model* yang telah dilatih secara langsung, pada lingkungan ruangan.

Berikut adalah hasil dari pengujian deteksi manusia secara langsung.

4.2.1. Deteksi Jumlah 1 Orang Posisi duduk

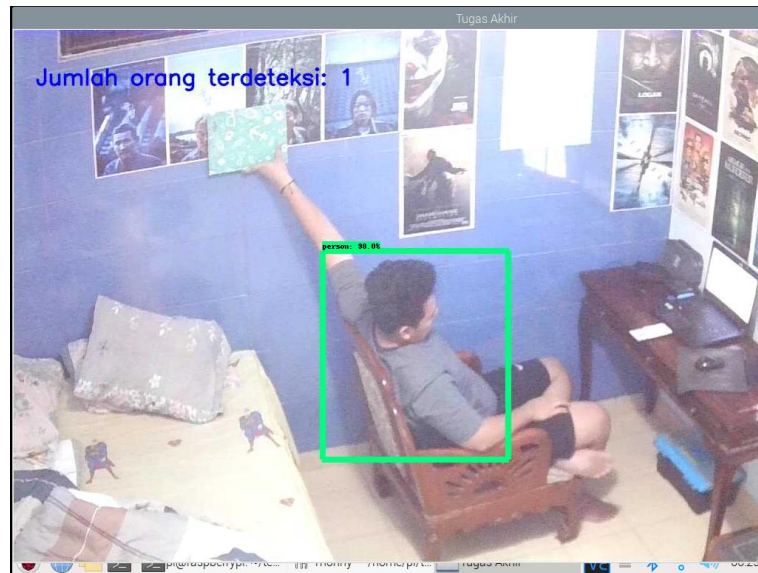
Pengujian dilakukan pada ruangan kamar dengan jarak kamera terhadap objek ± 5 meter. Objek yang dideteksi merupakan manusia utuh dengan pose duduk. Sistem yang digunakan untuk mendeteksi adalah Raspberry Pi yang dihubungkan dengan Web Camera NYK Nemesis A80.

Dari pengujian yang telah dilakukan, berikut adalah hasil dari pengujian sistem saat mendeteksi 1 orang.

No	Model	Hasil Deteksi		
		Jumlah deteksi	Jumlah terdeteksi	Akurasi
1	30000 steps	1	1	100%
2	50000 steps	1	1	100%

No	Model	Hasil Deteksi		
		Jumlah deteksi	Jumlah terdeteksi	Akurasi
3	100000 steps	1	1	100%
4	200000 steps	1	1	100%

Berikut adalah hasil deteksi salah satu model yang digunakan yaitu 100000 steps:



4.2.2. Deteksi Jumlah 1 Orang dalam keadaan Gelap

Pengujian dilakukan pada ruangan tertutup dan intensitas cahaya yang bervariasi, dan pada malam hari. Sumber cahaya pada pengujian ini adalah lampu ruangan dengan daya 8 watt. Untuk mendeteksi cahaya, dilakukan dengan pengukuran intensitas cahaya menggunakan alat Lux Meter.

Berikut adalah hasil dari pengujian deteksi jumlah 1 orang dengan intensitas cahaya sebesar 3 lux.

No	Model	Hasil Deteksi		
		Jumlah deteksi	Jumlah terdeteksi	Akurasi
1	30000 steps	1	1	100%
2	50000 steps	1	1	100%
3	100000 steps	1	1	100%
4	200000 steps	1	1	100%

Berikut adalah hasil deteksi salah satu model yang digunakan yaitu 100000 steps:

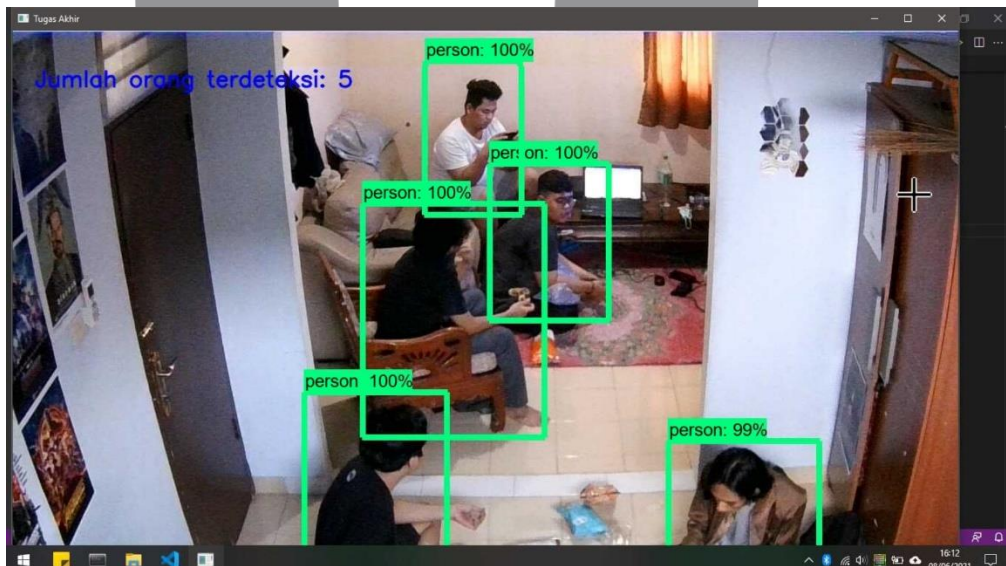


4.2.3. Deteksi Jumlah 5 Orang

Dari pengujian yang telah dilakukan, berikut adalah hasil dari pengujian sistem saat mendeteksi 5 orang.

No	Model	Hasil Deteksi		
		Jumlah deteksi	Jumlah terdeteksi	Akurasi
1	30000 steps	5	5	100%
2	50000 steps	5	5	100%
3	100000 steps	5	5	100%
4	200000 steps	5	5	100%

Berikut adalah hasil deteksi salah satu model yang digunakan yaitu 100000 steps:



4.2.4. Deteksi Jumlah 27 Orang

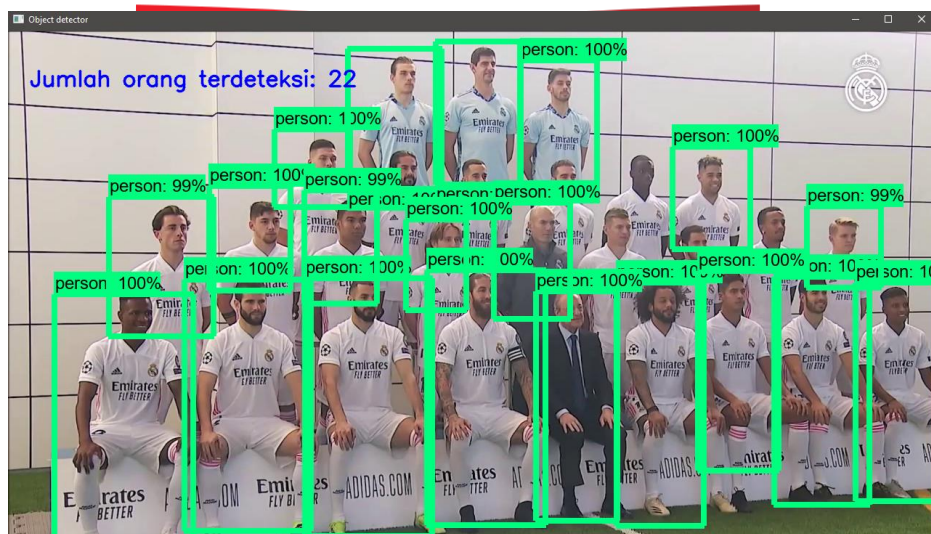
Pengujian dilakukan dengan melakukan deteksi pada video yang diambil dari Youtube. Video yang digunakan adalah hasil cropping (pemotongan) dari video aslinya, dengan tujuan hanya mengambil bagian video yang berisi jumlah orang yang ingin di uji. Durasi video yang

digunakan untuk pengujian adalah 2 detik, dari hasil pemotongan video aslinya yang berdurasi 01:08. Link video yang digunakan: <https://www.youtube.com/watch?v=XJf4FnuE4Wo>.

Berikut adalah hasil dari pengujian deteksi dengan jumlah 27 orang pada video.

No	Model	Hasil Deteksi		
		Jumlah deteksi	Jumlah terdeteksi	Akurasi
1	30000 steps	27	20	74%
2	50000 steps	27	21	77%
3	100000 steps	27	22	81%
4	200000 steps	27	20	74%

Berikut adalah hasil deteksi salah satu model yang digunakan yaitu 100000 steps



5. Kesimpulan dan Saran

5.1. Kesimpulan

1. Sistem yang dirancang menggunakan bahasa pemrograman Python, dengan library-library pendukung pembelajaran mesin seperti Tensorflow dan OpenCV, dan juga menggunakan Tensorflow Object Detection API. OpenCV digunakan untuk membuat tampilan visual dari hasil deteksi objek. Proses perhitungan manusia berjalan dengan baik. Jumlah manusia dihitung berdasarkan jumlah bounding box yang nampak pada layar.
2. Proses training dilakukan pada dua tempat, yaitu di laptop dan di Google Colab. Google Colab digunakan untuk beberapa parameter tertentu. Proses training bertujuan untuk menghasilkan model dengan akurasi yang baik. Untuk mendapatkan akurasi yang baik, diperlukan sebuah pengujian yang berfokus pada akurasi. Pada pengujian akurasi, terdapat 4 parameter yang menjadi dasar pengujian untuk mencari akurasi terbaik. Keempat parameter tersebut adalah: Distribusi dataset, Learning Rate, Epochs, dan jumlah Steps. Keempat parameter tersebut memberi pengaruh yang besar terhadap akurasi yang dihasilkan oleh model yang dilatih. Setelah diuji, model-model yang telah dilatih menghasilkan akurasi yang berbeda setiap kali parameter tersebut diubah, sehingga ditemukan model terbaik, yaitu model dengan distribusi dataset sebesar 95% training – 5% testing, learning rate sebesar 2×10^{-3} (0.002), epochs sebesar 50, dan jumlah steps sebesar 200000. Model tersebut telah cukup dalam saat menjalani proses training sehingga menghasilkan akurasi terbaik, yaitu 95%. Maka dari itu, proses training telah dilakukan dengan baik dan menghasilkan model dengan akurasi yang baik.

5.2. Saran

Berdasarkan hasil perancangan dan pengujian sistem yang telah dilakukan pada tugas akhir ini, maka berikut adalah saran-saran yang diusulkan untuk penelitian lebih lanjut, yaitu:

1. Menambah jumlah dataset, agar model yang dilatih akan semakin baik.
2. Menambah dataset objek berupa orang yang ada di dalam ruangan kantor, dengan jumlah yang seimbang.
3. Melakukan tuning hyperparameter kembali sehingga akurasi yang dihasilkan akan semakin tinggi, dan menghindari overfitting maupun underfitting pada model yang dilatih.
4. Setelah pandemi Covid-19 berakhir, diharapkan seluruh pengujian dapat dilakukan secara langsung.

REFERENSI

- [1] S. Wirjohamidjojo and Y. Swarinoto, *Iklm Kawasan Indonesia (Dari Aspek Dinamik - Sinoptik)*. Jakarta: Badan Meteorologi Klimatologi dan Geofisika (BMKG), 2010.
- [2] D. Putra, *Pengolahan Citra Digital*, 1st ed. Andi Publisher, 2010.
- [3] S. Sachan, "Object Detection using Raspberry Pi," *CIKITUSI J. Multidiscip. Res.*, vol. 6, no. 10, pp. 39–42, 2019.
- [4] J. Manasa, J. T. Pramod, S. A. K. Jilani, and S. J. Hussain, "Real time object counting using Raspberry pi," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 7, pp. 540–544, 2015, doi: 10.17148/IJARCCCE.2015.47122.
- [5] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [6] D. Alamsyah and M. Fachrurrozi, "Faster R-CNN with inception v2 for fingertip detection in homogenous background image," *J. Phys. Conf. Ser.*, vol. 1196, no. 1, 2019, doi: 10.1088/1742-6596/1196/1/012017.
- [7] Y. Liu, "An Improved Faster R-CNN for Object Detection," *Proc. - 2018 11th Int. Symp. Comput. Intell. Des. Isc. 2018*, vol. 2, pp. 119–123, 2018, doi: 10.1109/ISCID.2018.10128.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [9] J. Adiwibowo, K. Gunadi, and E. Setyati, "Deteksi Alat Pelindung Diri Menggunakan Metode YOLO dan Faster R-CNN," *J. Infra*, vol. 18, no. 2, pp. 106–112, 2020.
- [10] R. M. Awangga, R. Andarsyah, and E. C. Putro, *Tutorial Object Detection With Faster Region-Based Convolutional Neural Network (Faster R-CNN)*. Kreatif Industri Nusantara, 2020.
- [11] B. Liu, W. Zhao, and Q. Sun, "Study of object detection based on Faster R-CNN," *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 6233–6236, 2017, doi: 10.1109/CAC.2017.8243900.