



TESIS - EE184401

**Deteksi Kerumunan untuk Pencegahan Penularan
Virus Corona Berbasis Video Menggunakan
Mask-RCNN**

LUKAS PURBA WISESA
NRP 6022211057

DOSEN PEMBIMBING
Reza Fuad Rachmadi, S.T., M.T., Ph.D
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

PROGRAM MAGISTER
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2022

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik (MT)

di

Institut Teknologi Sepuluh Nopember

Oleh:

Lukas Purba Wisesa

NRP:6022211057

Tanggal Ujian :27 April 2022

Periode Wisuda : September 2022

Disetujui oleh:

Pembimbing:

1. Reza Fuad Rachmadi, S.T., M.T., Ph.D
NIP:198504032012121000

2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP:196806011995121009

Pengaji:

1. Dr. Adhi Dharma Wibawa, ST., MT.
NIP:197605052008121003

2. Eko Setijadi, ST., MT., Ph.D
NIP:197210012003121002

Kepala Departemen Teknik Elektro
Fakultas Teknologi Elektro Dan Informatika Cerdas

Dedet Candra Riawan, ST., M.Eng., Ph.D

Nip:197311192000031001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **Deteksi Kerumunan untuk Pencegahan Penularan Virus Corona Berbasis Video Menggunakan Mask-RCNN** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya 29 April 2022

Lukas Purba Wisesa

Nrp :6022211057

Halaman ini sengaja dikosongkan

Kata Pengantar

Puji syukur kehadirat Tuhan YME. atas segala limpahan berkat dan rahmat-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Deteksi Kerumunan untuk Pencegahan Penularan Virus Corona Berbasis Video Menggunakan Mask-RCNN**.

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Elektro, serta digunakan sebagai persyaratan menyelesaikan pendidikan S2. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak, dan kakak tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian penelitian ini.
2. Bapak Dedet Candra Riawan, ST., M.Eng., Ph.D. selaku Kepala Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC), Institut Teknologi Sepuluh Nopember.
3. Bapak Reza Fuad Rachmadi, ST., MT., Ph. D selaku dosen pembimbing I yang telah memberikan arahan dan saran selama mengerjakan penelitian tugas akhir ini.
4. Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. selaku dosen pembimbing II yang telah memberikan arahan dan saran selama mengerjakan penelitian tugas akhir ini.
5. Bapak Dr. Adhi Dharma Wibawa, ST., MT. dan Bapak Eko Setijadi ST, MT, Ph.D. selaku dosen penguji yang telah memberikan saran dan masukkan yang berguna untuk penulisan penelitian.
6. Bapak-ibu dosen pengajar Departemen Teknik Elektro atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
7. Seluruh teman - teman dari Telematika 2020, Teknik Elektro yang telah mendukung dan menjadi teman seperjuangan saya.

Surabaya, April 2022

Penulis

Halaman ini sengaja dikosongkan

Deteksi Kerumunan untuk Pencegahan Penularan Virus Corona Berbasis Video Menggunakan Mask-RCNN

Nama Mahasiswa : Lukas Purba Wisesa
NRP : 6022211057
Pembimbing : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D
 2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRAK

Pandemi Covid-19 yang berlangsung dari tahun 2020 telah berlangsung selama 2 tahun. Walaupun banyak masyarakat telah menerima vaksin, pemerintah tetap menegaskan masyarakat untuk melakukan protokol kesehatan terutama di tempat umum. Mengikuti saran dari WHO, salah satu protokol kesehatan yang perlu diterapkan adalah menjaga jarak satu sama lain dengan jarak satu meter. Namun, dalam penerapan protokol kesehatan, tentunya banyak masyarakat yang lalai untuk menerapkan hal tersebut. Oleh karena itu, sebuah sistem deteksi yang dapat membantu memudahkan penerapan protokol kesehatan dirasa dibutuhkan.

Penelitian ini memanfaatkan algoritma Mask-RCNN yang digunakan untuk mendeteksi apakah terdapat kerumunan dalam suatu gambar yang diambil. Objek yang dideteksi pada penelitian ini dibagi menjadi tiga kelas yaitu : Crowd, Group, dan Person. Kelas Crowd merupakan kelas dimana terletak 4 orang atau lebih dalam radius 1 m, kelas group merupakan kelas dimana terdapat 2-3 orang dalam radius 1 m sedangkan kelas person merupakan kelas dimana hanya terdapat 1 orang dalam radius 1 m. Mask R-CNN mendeteksi objek dengan cara memberikan beberapa proposal objek yang kemudian diproses dengan layer konvolusi dan *fully connected layer*. Keluaran dari layer konvolusi adalah *mask* yang menunjukkan segmentasi gambar, sedangkan keluaran dari *fully connected layers* adalah memberikan *bounding box* serta menentukan nama kelas dari objek tersebut. Hasil dari penelitian ini adalah model yang dapat mendeteksi adanya kerumunan dengan akurasi sebesar 88.79%.

Kata kunci : Mask R-CNN, Deteksi Kerumunan, *deep learning*, Covid-19.

Halaman ini sengaja dikosongkan

Video-Based Crowd Detection for Preventing Corona Virus Transmission Using Mask-RCNN

By : Lukas Purba Wisesa
Student Identity Number : 6022211057
Supervisor :
1. Reza Fuad Rachmadi, S.T., M.T., Ph.D
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

ABSTRACT

The Covid-19 pandemic that started in 2020 has been going on for 2 years. Although many people have received the vaccine, the government continues to emphasize the public to carry out health protocols especially in public places. Following the advice of WHO, one of the health protocols that need to be implemented is to keep a meter distance from each other. However, in implementing the health protocol, of course, many people tend to neglect the physical distancing practice. Therefore, a detection system that can help facilitate the implementation of health protocols especially physical distancing is needed. Crowd class is a class where there are 4 or more people within a 1 m radius, group class is a class where there are 2-3 people within a radius a meter while the person class is a class where there is only 1 person within a meter radius. This study utilizes the Mask-RCNN algorithm which is used to detect whether there is a crowd in an image taken. The objects detected in this study were divided into three classes, namely: Crowd, Group, and Person. The mask R-CNN algorithm detects objects by providing several object proposals which are then processed with the convolution layer and fully connected layer. The output of the convolution layer is mask which shows the image segmentation, while the output of fully connected layers is to give the bounding box and specify the class name of the object. The result of this study is a model that can detect crowds with an accuracy of 88.79%

Keyword: *Mask R-CNN, Crowd Detection, Deep Learning, Covid-19.*

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
KATA PENGANTAR	vii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
NOMENKLATUR	1
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Batasan Masalah	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
2 KAJIAN PUSTAKA	7
2.1 Covid-19	7
2.1.1 Cara Penularan	8
2.1.2 Epidemiologi	9
2.1.3 Dampak Covid-19	10
2.1.4 Cara Penanggulangan	11
2.2 Machine Learning	12
2.2.1 <i>Supervised Learning</i>	13
2.2.2 <i>Unsupervised Learning</i>	13
2.2.3 <i>Reinforcement Learning</i>	13
2.3 <i>Deep Learning</i>	13
2.4 Artificial Neural Network	14
2.5 <i>Convolutional Neural Network</i>	18
2.5.1 Konvolusi	18
2.5.2 <i>Pooling</i>	19
2.6 Deteksi Objek	20
2.6.1 <i>Two-Stage Object Detectors</i>	22
2.6.2 <i>Single-Stage Object Detector</i>	23
2.7 Algoritma Deteksi Objek Mask R-CNN	24
2.8 Metode Analisa Performa Model	28
2.8.1 <i>Recall</i>	28
2.8.2 <i>Precision</i>	29

2.8.3	mAP@50	30
2.8.4	<i>F1-Score</i>	30
2.8.5	<i>Confusion Matrix</i>	30
3	METODOLOGI	33
3.1	Pembentukan Dataset	34
3.1.1	Pengumpulan Data	35
3.1.2	Anotasi Data	35
3.2	Data <i>Preprocessing</i>	37
3.2.1	Pembuatan <i>Bounding Box</i>	38
3.2.2	Pengubahan Ukuran Gambar	38
3.2.3	Pembentukan Mini Mask	39
3.3	Ekstraksi Fitur	39
3.3.1	Pembuatan <i>Anchor Box</i>	40
3.3.2	Prediksi RPN	41
3.3.3	Pembuatan <i>Feature Map</i>	41
3.4	Proses Training	42
3.5	Proses Pengujian	43
3.6	Analisa Performa	45
4	PENGUJIAN	47
4.1	Hasil Pembuatan Dataset	47
4.2	Hasil <i>Preprocessing</i> Data	49
4.2.1	Hasil Pembuatan <i>Bounding Box</i>	49
4.2.2	Hasil Pengubahan Ukuran Gambar	50
4.2.3	Hasil Pembuatan <i>Mini Mask</i>	51
4.3	Hasil <i>Feature Extraction</i>	52
4.3.1	Hasil Pembuatan <i>Anchor Box</i>	53
4.3.2	Hasil Prediksi RPN dan Pembuatan <i>Feature Map</i>	53
4.3.3	Hasil <i>Training</i>	54
4.3.4	Hasil Pengujian pada Test Dataset	61
5	PENUTUP	69
5.1	Kesimpulan	69
5.2	Saran	70
Daftar Pustaka		71
Biodata Penulis		79

DAFTAR GAMBAR

2.1	<i>Artificial Neural Network</i>	15
2.2	Fungsi Sigmoid	16
2.3	Fungsi Tanh	17
2.4	Fungsi ReLu	17
2.5	Contoh Arsitektur CNN	18
2.6	Konvolusi	19
2.7	Contoh <i>Pooling 2x2</i>	20
2.8	<i>Intersect over Union</i>	21
2.9	Contoh <i>Two Stage Detector</i>	22
2.10	Arsitektur YOLO [1]	24
2.11	Arsitektur Mask R-CNN [2]	25
2.12	Macam - macam <i>Residual Network (ResNet)</i> [3]	26
3.1	Bagan Umum Metodologi Penelitian	34
3.2	Alur Pembentukan Dataset	35
3.3	Konfigurasi Pengambilan Gambar	36
3.4	Software VIA [4]	37
3.5	Contoh Hasil Uji Coba Pembuatan Dataset [5]	37
3.6	Tahap <i>Preprocessing</i>	38
3.7	Proses Ekstraksi Fitur	40
3.8	Bentuk <i>Anchor Box</i> [6]	41
3.9	Hasil <i>non max supression</i> (kanan) [7]	42
3.10	Contoh <i>Feature map</i> pada <i>layer</i> berbeda [5]	42
4.1	Contoh Kelas <i>Person</i>	48
4.2	Contoh Kelas <i>Group</i>	48
4.3	Contoh Kelas <i>Crowd</i>	49
4.4	Contoh Beberapa Hasil Dataset	50
4.5	Contoh Hasil Pembuatan <i>Bounding Box</i>	50
4.6	Contoh Hasil <i>Resizing Image</i>	51
4.7	Contoh Hasil Pembuatan <i>Mini Mask</i>	52
4.8	Arsitektur Mask R-CNN yang diaplikasikan	52
4.9	Hasil <i>Anchor Box</i> yang Dihasilkan	54
4.10	Contoh Hasil <i>Anchor Box</i> yang Menjadi Target RPN	54
4.11	Contoh Hasil <i>Anchor Box</i> yang Tidak Digunakan	55
4.12	Contoh Hasil Prediksi RPN	55
4.13	Contoh Hasil <i>Feature Map</i>	55
4.14	Hasil <i>Loss</i>	58
4.15	Hasil <i>Class Loss</i> RPN	59
4.16	Hasil <i>Bounding Box Loss</i> RPN	59
4.17	Hasil <i>Class Loss</i> Mask RCNN	60

4.18 Hasil <i>Bounding Box Loss</i> Mask RCNN	61
4.19 Hasil <i>Mask Loss</i> Mask RCNN	62
4.20 Hasil Prediksi (1)	63
4.21 Hasil Prediksi dan IoU (1)	63
4.22 Hasil Kurva Prediksi dan Recall (1)	64
4.23 Contoh Hasil Prediksi (2)	65
4.24 Hasil Kurva Prediksi dan Recall (2)	66
4.25 Contoh Hasil Prediksi dan IoU(2)	66
4.26 Beberapa Hasil Deteksi pada Dataset Test	67
4.27 <i>Confusion Matrix</i> dari Hasil Prediksi Test Dataset	67

DAFTAR TABEL

2.1	Contoh <i>Confusion Matrix</i>	30
3.1	Konfigurasi pada proses <i>training</i>	43
3.2	<i>learning rate</i> dinamis	43
4.1	Detail Jumlah Objek Anotasi	49
4.2	Metriks Loss pada Model 124	62

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini memiliki beberapa latar belakang yang disebabkan oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai keluaran dari penelitian.

1.1 Latar Belakang

Virus Corona merupakan varian virus yang dapat menginfeksi manusia, gejala yang ditimbulkan oleh varian virus ini meliputi infeksi paru-paru, pneumonia, demam, dan kesusahan bernapas. Dalam 2 dekade terakhir, Virus Corona telah menyebabkan 3 pandemi besar [8]. Pada tahun 2002, sebuah varian virus corona yang dinamakan *Severe Acute Respiratory Syndrome Corona Virus (SARS-CoV)* menginfeksi manusia untuk pertama kalinya di Cina [9, 10], varian virus ini menginfeksi 8.000 orang dan menyebabkan kematian bagi 750 orang. 10 Tahun kemudian pada tahun 2012, varian baru virus corona yang dinamakan *Middle East Respiratory Syndrome Corona Virus (MERS-CoV)* [11]. MERS-CoV menginfeksi 2.494 pasien dari beberapa negara di Timur Tengah dan menimbulkan korban jiwa sebanyak 866 orang. Kemudian varian Virus Corona yang dinilai paling mematikan muncul pada Desember 2019 dan menyebabkan pandemi besar di Kota Wuhan, Provinsi Hubei, Cina. Virus ini dinamakan SARS-CoV-2.

Penyakit yang ditimbulkan oleh SARS-CoV 2 dinamakan *Corona Virus disease 2019* atau COVID-19, gejala COVID-19 meliputi sakit kepala berat, demam, batuk, dan diare [12]. Pada 29 Oktober 2021, COVID-19 telah menginfeksi lebih dari 450 juta orang dan mengakibatkan kurang lebih 6.000.000 korban jiwa [13]. Walaupun tingkat persentase kematian SARS-CoV 2 lebih rendah dibandingkan SARS-CoV maupun MERS-CoV, kemampuan penyebaran SARS-CoV 2 lebih tinggi dibandingkan varian virus sebelum-sebelumnya [14]. Virus SARS-CoV 2 dapat dengan mudah menyebar melalui *droplet* atau tetesan air yang dikeluarkan dari penderita. Penularan virus

SARS-CoV 2 sering terjadi kepada orang dengan kondisi sehat saat penderita berbicara tanpa menggunakan masker kepada orang tersebut [15].

Tidak hanya kerugian dalam dampak kesehatan, pandemi Covid-19 juga memberikan dampak ekonomi secara negatif dimana bisnis pada berbagai sektor mengalami penurunan keuntungan [16]. Sepanjang pandemi Covid-19, negara di seluruh dunia pun mengalami penurunan pemasukan. Rata-rata penurunan pemasukan tersebut bernilai 4.8% pada negara dengan pemasukan kecil dan penurunan rata-rata 8.9% pada negara dengan pemasukan besar. Salah satu contoh sektor yang mengalami kerugian besar ialah sektor aviasi. Penerbangan sektor internasional menjadi terhambat dikarenakan negara-negara yang melakukan *lockdown* dan menerapkan kebijakan *travel restriction*. Hal ini dibuktikan dengan penerbangan komersial Internasional Amerika yang awalnya 4.000 penerbangan per bulan pada tahun 2019 turun menjadi 1.500 penerbangan per bulannya pada tahun 2020 [17].

Mudahnya penularan virus SARS-CoV 2 serta besarnya efek negatif yang ditimbulkan menjadi alasan bagi pemerintah untuk melakukan pencegahan dan mengurangi penularan virus SARS-CoV 2. *World Health Organization* menyarankan agar masyarakat menjaga jarak dan melakukan *physical distancing* dengan jarak satu meter [18]. Berdasarkan hal tersebut, Indonesia melakukan berbagai kebijakan, beberapa diantaranya ialah pengadaan PPKM yang diatur pada PerPres no 21 tahun 2020 yang meliputi peliburuan sekolah dan tempat kerja, pembatasan kegiatan keagamaan, dan pembatasan kegiatan di tempat umum. Selain kebijakan tersebut Kepala Kepolisian Negara Republik Indonesia juga mengeluarkan maklumat yang melarang adanya aktivitas berkumpulnya massa dalam jumlah banyak, baik di tempat umum maupun di lingkungan sendiri [19].

Dalam melakukan pengawasan kerumunan dan mengurangi penularan virus corona, sering kali petugas melakukan patroli dan peninjauan lokasi secara langsung. Namun patroli tentunya membutuhkan personil untuk datang ke lokasi dan hanya dilakukan pada jam serta hari-hari tertentu. Tidak

hanya itu, patroli yang dilakukan tidak dapat mendata tempat keramaian secara detail untuk analisa lebih lanjut. Dengan permasalahan tersebut, penulis menawarkan solusi untuk melakukan pengawasan kerumunan dengan pembelajaran mesin menggunakan algoritma mask R-CNN. Selain untuk pengawasan kerumunan untuk mengurangi penyebaran Covid-19, data deteksi kerumunan ini juga dapat digunakan untuk desain arsitektur tempat umum seperti ruang tunggu, maupun pintu masuk ruangan yang lebih mendukung gaya hidup pasca Covid-19.

1.2 Rumusan Masalah

Pemantauan kerumunan selama ini masih dijalankan secara pemeriksaan manual. Petugas masih harus melakukan pemantauan pada video cctv atau datang ke lokasi langsung untuk melakukan patroli. Jika kondisi wilayah yang perlu diamati bertambah dan sumber daya manusia pemerintah kota tidak mencukupi dalam melaksanakan pemantauan kerumunan di berbagai wilayah, maka kerumunan tidak dapat diketahui dan berpotensi besar menjadi salah satu media penyebaran Virus Corona. Berdasarkan penjelasan di atas, maka akan dikembangkan sebuah sistem deteksi kerumunan otomatis menggunakan algoritma kecerdasan buatan Mask R-CNN. Harapannya tidak hanya data deteksi kerumunan yang diberikan dapat digunakan untuk mengatasi dan melaporkan situasi kerumunan, data ini juga dapat digunakan dalam pemodelan penularan virus maupun desain ruang yang lebih baik.

1.3 Tujuan

Tujuan utama dari penelitian ini adalah mempersiapkan teknologi untuk melakukan pengamatan secara otomatis terhadap suatu wilayah dalam rangka mengurangi penyebaran virus corona maupun mengambil informasi mengenai kerumunan di beberapa wilayah, serta mengurangi sumber daya manusia yang diperlukan untuk melakukan pengawasan kerumunan. Selain untuk mempersiapkan teknologi masa depan yang dapat memonitoring kerumunan, deteksi kerumunan secara otomatis juga dapat digunakan untuk melihat apakah PPKM yang dilaksanakan berhasil atau tidak dengan mengetahui

jumlah kerumunan yang terjadi. Semua hal ini dilakukan dalam rangka menurunkan kasus Covid-19 dan mengurangi penyebaran Virus Corona.

1.4 Batasan Masalah

Untuk memfokuskan permasalahan yang diangkat, maka dilakukan pembatasan masalah. Batasan - batasan masalah tersebut diantaranya adalah :

1. Data yang diambil merupakan data video dengan kondisi nyata di salah satu pusat perbelanjaan Surabaya, Jawa Timur, Indonesia.
2. Algoritma yang digunakan dalam penelitian ini ialah Mask R-CNN.
3. Pengambilan data dilakukan dari ketinggian dan sudut pengambilan yang akan didefinisikan pada Bab 3.
4. Kelas yang dideteksi dibedakan menjadi tiga kelas yaitu *person*, *group*, *crowd* yang didefinisikan oleh peneliti.
5. Pengambilan data dilakukan pada bulan Januari 2022 di Pasar Atom Surabaya pada jam 1 siang hingga 4 sore.
6. Objek yang dideteksi merupakan objek orang segala usia.
7. Hasil keluaran dari penelitian ini adalah model pembelajaran mesin yang dapat digunakan untuk mendeteksi kerumunan Indonesia.

1.5 Manfaat

Diharapkan penelitian ini dapat mendeteksi kerumunan dalam rangka membantu penanganan Covid-19 maupun pandemi masa depan yang mungkin terjadi. Hasil dari deteksi kerumunan dapat digunakan sebagai sistem *monitoring* atau sebagai penghasil data mengenai perilaku masyarakat pasca pandemi Covid-19, selain itu, diharapkan penelitian ini dapat membantu penanganan pandemi di masa depan.

1.6 Sistematika Penulisan

Laporan penelitian ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. **BAB I Pendahuluan** Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. **BAB II Tinjauan Pustaka**

Bab ini berisi tentang uraian secara sistematis teori - teori yang berhubungan dengan permasalahan yang dibahas pada penelitian ini. Teori - teori ini digunakan sebagai dasar dalam penelitian, yaitu informasi terkait *Deep Learning*, *Object Detection*, *Mask R-CNN*, dan teori - teori penunjang lainnya.

3. **BAB III Desain dan Implementasi Sistem**

Bab ini berisi tentang penjelasan - penjelasan terkait eksperimen yang akan dilakukan, langkah - langkah pengambilan dataset dan proses pembagian kelas kerumunan. Untuk mendukung hal tersebut, maka ditampilkan pula *workflow* agar proses penelitian yang akan dibuat dapat terlihat dan mudah dibaca untuk proses pembuatan pada pelaksanaan tugas akhir.

4. **BAB IV Pengujian dan Analisa**

Bab ini menjelaskan tentang hasil serta analisis yang didapatkan dari pengujian yang dilakukan mulai dari hasil pengujian *precision*, *recall*, *Confusion Matrix* serta rekomendasi penggunaan model.

5. **BAB V Penutup**

Bab ini berisi penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran mengenai langkah selanjutnya untuk pengembangan lebih lanjut juga dituliskan pada bagian ini.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Covid-19

Covid-19 merupakan salah satu penyakit dengan daya penyebaran paling tinggi yang disebabkan oleh *Severe acute respiratory syndrome (SARS-CoV-2)*. Penyakit Covid-19 telah menyebabkan banyaknya korban jiwa serta kerugian materi di berbagai aspek ekonomi. WHO menetapkan Covid-19 sebagai pandemi global pada tanggal 11 maret 2020. Pada awalnya, dapat dikatakan bahwa pelayanan kesehatan tidak memiliki sistem yang baik dalam penanganan Covid-19. Walaupun sistem pelayanan kesehatan terutama di rumah sakit dalam menangangi Covid-19 semakin lama semakin membaik dan telah mengalami banyak kemajuan, beberapa mutasi yang mengubah karakteristik virus secara drastis dapat menimbulkan semakin banyak korban jiwa maupun kerugian materi. Oleh karena itu pembatasan penyebaran virus dan variannya harus tetap dilakukan.

Hingga saat ini varian SARS-CoV-2 yang disebabkan oleh mutasi virus tetap terus diteliti, beberapa varian yang dianggap membahayakan karena meningkatnya daya penularan dan keagresifan virus diberi nama *Variant of Concern (VOC)*. Pada tanggal 11 Desember 2021, WHO menetapkan adanya lima *variant of concern* yaitu :

1. Alpha (B.1.1.7)

Varian Alpha merupakan *variant of concern* pertama yang muncul di Inggris pada akhir Desember 2020 [20]. Varian ini memiliki 17 mutasi dan mempunyai peningkatan pada daya lekat ke sel *host* [21, 22]. Varian

ini memiliki peningkatan transmisi 43% Hingga 82% dibandingkan varian SARS-CoV-2 sebelumnya [22].

2. Beta (B.1.351)

Varian Beta pertama kali dilaporkan di Afrika Selatan pada Desember 2020 dan menyebabkan adanya gelombang kedua pandemi Covid-19 di Afrika Selatan [23]. Varian ini memiliki 9 mutasi [24] dimana varian ini memiliki kemampuan untuk menurunkan antibodi sehingga dapat menginfeksi orang yang telah divaksin atau menerima plasma konvalesen [25].

3. Gamma(P.1)

Varian Gamma pertama kali dilaporkan di Brazil pada awal Januari 2021 [26]. Varian Gamma memiliki 10 mutasi. Efek dari mutasi ini hampir mirip dengan varian Beta dimana varian ini memiliki kemampuan untuk menurunkan antibodi orang yang terinfeksi [25].

4. Delta (B.1.617.2)

Varian Delta pertama kali dilaporkan di India pada Desember 2020. Varian ini merupakan salah satu yang paling berbahaya karena pasien yang terkena oleh varian delta mempunyai peluang kematian 133% lebih tinggi dibandingkan varian awal dan 235% lebih memungkinkan untuk masuk ke ruang ICU. Vaksin yang diberikan juga mengalami penurunan efektivitas melawan varian delta [27].

5. Omicron (B.1.1.529)

Varian Omicron pertama kali dilaporkan di Afrika Selatan pada November 2021 [28]. Varian Omicron mempunyai 30 mutasi [29] dan 2,8 kali lebih menular dibandingkan varian delta.

2.1.1 Cara Penularan

Penyakit Covid-19 yang disebabkan oleh SARS-CoV-2 dapat menular melalui cara-cara berikut :

- Transmisi SARS-CoV-2 yang paling umum ialah melalui paparan *droplet* yang dikeluarkan oleh individu lain yang telah terinfeksi oleh virus tersebut.
- Melalui material seperti besi dan plastik yang terpapar oleh SARS-CoV-2, SARS-CoV-2 dapat bertahan selama 72 jam pada material besi dan plastik, 24 jam pada material kardus, dan 4 jam pada material tembaga. Tingkat kontaminasi semakin meningkat jika barang tersebut digunakan oleh orang banyak contoh seperti tetikus komputer di rumah sakit [30].
- Melalui transmisi vertikal dari ibu yang hamil kepada bayi yang akan dilahirkan, namun kasus ini jarang terjadi, sebuah penelitian [31] mengungkapkan bahwa dari 936 bayi yang dilahirkan dari ibu yang positif Covid-19, hanya 27 bayi yang dinyatakan positif Covid-19.

2.1.2 Epidemiologi

Dari tulisan sebelumnya, sudah dapat diketahui bahwa Covid-19 dapat menimbulkan kematian, namun terdapat beberapa faktor yang mempengaruhi apakah pasien akan sembuh dari Covid-19 atau tidak. Beberapa faktor tersebut adalah umur, gender, dan penyakit yang diderita oleh pasien. Pasien yang mempunyai umur di atas 65 tahun mempunyai kemungkinan untuk meninggal lebih besar dibandingkan kelompok usia lain [32]. Penelitian lain [33] mengakatakan bahwa pasien-pasien Covid-19 yang memiliki obesitas, penyakit jantung, penyakit kronis ginjal, diabetes, penyakit kronis paru-paru, kanker, dan kebiasaan merokok lebih berpeluang untuk dirawat di rumah sakit (45,4%) dibandingkan yang tidak memiliki kondisi tersebut (7,6%). Selain itu pasien-pasien yang memiliki kondisi tersebut 12x lebih berpotensi untuk tidak sembuh dan meninggal (19,5%) dibandingkan pasien yang tidak memiliki kondisi kesehatan tersebut (1,6%).

Kemudian jika dibedakan berdasarkan gender, berdasarkan data yang diperoleh, pasien pria lebih memiliki risiko yang lebih tinggi untuk menjadi pasien Covid-19 dan meninggal akibat Covid-19 dibandingkan pasien per-

empuan [34]. Peneliti pun juga meniliti apakah ras dan etnik mempunyai hubungan dengan Covid-19. Hasil dari meta-analisis peneliti Amerika dan Inggris mengatakan bahwa orang-orang dengan ras afrika, hispanik (keturunan eropa), dan orang asia memiliki kecenderungan untuk mengalami Covid-19 [35], dimana tingkat kematian tertinggi terjadi pada orang hispanik [36].

2.1.3 Dampak Covid-19

Sangat banyak dampak negatif yang diberikan saat terjadinya pandemi Covid-19, jumlah korban jiwa karena kejadian ini telah mencapai lebih dari 6 juta orang [13]. Pandemi Covid-19 juga membawa dampak kemiskinan di berbagai negara. Di awal pandemi Covid-19, masyarakat di beberapa negara melakukan *panic buying* [37]. Tindakan *panic buying* ini terjadi karena ketakutan dari masyarakat untuk melakukan aktivitas di luar dan kepanikan masyarakat dalam rangka bertahan hidup. Hal ini pun secara otomatis meningkatkan kelangkaan bahan makanan serta harga dari kebutuhan pokok sehari-hari. Tingkat kemiskinan yang naik dan kelangkaan bahan makanan pun dapat dikatakan terjadi karena pandemi Covid-19 [38]. Tidak hanya kesehatan jasmani yang menjadi ancaman, beberapa penelitian juga membahas mengenai dampak Covid-19 pada kesehatan mental masyarakat [39]. Covid-19 memberikan dampak depresi stress, dan trauma terutama pada masyarakat golongan bawah. Diskriminasi terhadpa pasien Covid-19 yang sembuh pun turut mendukung kemungkinan terjadinya stress dan depresi pada orang yang sembuh dari penyakit Covid-19.

Dalam segi ekonomi, Covid-19 telah memberikan penurunan yang drastis pada pasar saham, performa industri, serta menurunkan jumlah pembeli di pasar-pasar tradisional maupun modern. Pada pasar saham, penurunan indeks saham yang terjadi pun berkolerasi erat dengan tingkat kasus Covid-19 yang terjadi [40]. Penurunan nilai saham ini pun dapat dikatakan terjadi berbagai belahan dunia. Performa yang menurun juga dialami oleh berbagai industri mulai dari industri kreatif [41] hingga industri-industri besar [42].

2.1.4 Cara Penanggulangan

Cara penanggulangan (termasuk cara pencegahan) Covid-19 dibedakan menjadi dua macam, yaitu dengan interfensi memprediksi dan non-interfensi medis. Interfensi medis membahas mengenai penanganan Covid-19 yang meliputi penggunaan terapi medis, obat-obatan, dan vaksinasi. Sedangkan non-interfensi medis membahas mengenai regulasi maupun penyuluhan bagi masyarakat.

- **Penanganan Medis**

Penanganan pandemi Covid-19 saat ini telah dilakukan dengan cara vaksinasi dan pemberian *booster* pada masyarakat untuk meningkatkan daya tahan tubuh dan menurunkan efek negatif yang diberikan oleh virus SARS Cov-2 kepada tubuh penerima vaksin. Pemberian vaksin Covid-19 yang dilakukan secara umum telah memberikan efek melindungi tubuh manusia serta memberikan kasiat klinis terhadap SARS-CoV 2 setidaknya 60% bergantung pada jenis vaksin yang diberikan. Salah satu vaksin yang memberikan efek paling baik salah satunya adalah vaksin BNT162b2 yang lebih kita kenal sebagai vaksin pfizer mempunyai efektivitas 95% terhadap penerima vaksin kedua [43].

Penanganan Medis lainnya khususnya penanganan yang digunakan saat pasien terkena Covid-19 adalah terapi plasma konvalesen. Walaupun terapi ini diperbolehkan, terdapat beberapa penelitian yang memberikan data bahwa terapi plasma konvalesen tidak lebih baik dibandingkan dengan terapi pada umumnya [44, 45]. Pemberian antibodi juga dapat membantu pasien khususnya REGN-COV2 dan VIR-7831 sangat dianjurkan. antibodi REGN-COV2 menurunkan jumlah pasien yang meninggal hingga 70% [46], sedangkan antibodi VIR-7831 menurunkan kemungkinan pasien meninggal atau perlu dirawat di rumah sakit hingga 80%.

- **Penanganan Non Medis**

Beberapa penanganan non medis pada pandemi Covid-19 juga telah dilaksanakan, beberapa hal penting yang dilakukan sebagai upaya penanganan non medis adalah penggunaan masker dan menjaga jarak satu sama lain. Penelitian [47] mengatakan bahwa dengan menjaga jarak dan menggunakan masker, menurunkan persentasi prediksi terjadinya korban jiwa 25% hingga 45%. Penanganan ini merupakan hal yang termudah untuk dilakukan dalam menurunkan kasus Covid-19 yang terjadi. Penanganan non medis juga dapat dilakukan dengan cara menutup bandara, membatasi kegiatan pariwisata, dan menutup beberapa tempat pusat kegiatan masyarakat [48].

2.2 Machine Learning

Machine Learning atau Pembelajaran Mesin adalah salah satu cabang dalam kecerdasan buatan dan ilmu komputer yang menggunakan data dan algoritma untuk meniru manusia dalam mempelajari sesuatu [49]. Salah satu hal yang membuat pembelajaran mesin sangat diminati adalah kemampuannya untuk menyelesaikan suatu tugas dengan sedikit intervensi dari manusia.

Sekarang ini, pembelajaran mesin adalah salah satu fokus yang cukup diminati pada bidang *data science*. Dimana dengan menggunakan pembelajaran mesin, diharapkan suatu kecerdasan buatan dapat menyelesaikan beberapa tugas yang bagi komputer cukup rumit seperti misalnya, memberikan prediksi yang akurat berdasarkan data, melakukan klasifikasi pada teks maupun pada gambar, melakukan pemrosesan citra guna mengenali objek di dalam citra tersebut, dan masih banyak lagi.

Untuk prosesnya sendiri, awalnya kita harus mengumpulkan data, data ini dapat kita ambil dari berbagai sumber atau bisa juga menggunakan data yang berasal dari instansi atau pribadi (data yang kita buat sendiri). Selanjutnya adalah proses *training* dimana data akan dimasukkan ke dalam model pembelajaran mesin yang sudah dipilih. Kita dapat merubah beberapa parameter dari model tersebut untuk meningkatkan akurasi dari suatu model pembelajaran mesin. Terakhir adalah melakukan proses *testing*, dimana model

akan melakukan prediksi pada set data yang berbeda dari yang digunakan pada saat proses *training*. Apabila ternyata tingkat akurasi dirasa kurang memadai, dapat dilakukan proses *re-training* sampai tingkat akurasi nya dirasa cukup. Hasil akhir dari proses ini adalah sebuah model pembelajaran mesin yang dapat digunakan walaupun menggunakan data yang berbeda [50].

2.2.1 *Supervised Learning*

Salah satu cabang dalam bidang pembelajaran mesin. Disini data yang dijadikan masukan ke model sudah diberikan label atau struktur tertentu [51]. Berdasarkan dari data berlabel tersebut, sebuah model akan merubah parameter internalnya agar mendekati atau sesuai dengan label yang diberikan [49]. Salah satu contoh model pembelajaran mesin dengan metode pembelajaran seperti ini adalah *Linear Regression*, *Random Forest*, dan sebagainya.

2.2.2 *Unsupervised Learning*

Salah satu cabang dalam bidang pembelajaran mesin. Disini data yang dijadikan masukan ke model tidak diberikan label sama sekali. Nantinya model akan membuat pengelompokan (*clusters*) dan hubungan berdasarkan dari data yang diberikan [50]. Contoh model yang menggunakan metode pembelajaran ini adalah *K-NN clustering*.

2.2.3 *Reinforcement Learning*

Salah satu cabang dalam bidang pembelajaran mesin. Disini model tidak diberikan data awal sama sekali, namun, model dibiarkan melakukan proses percobaan secara mandiri terus-menerus sampai tercapai hasil atau respon yang diinginkan. Apabila terdapat parameter yang menghasilkan respon positif, maka parameter tersebut disimpan dan digunakan sebagai masukan untuk iterasi *training* berikutnya [50].

2.3 *Deep Learning*

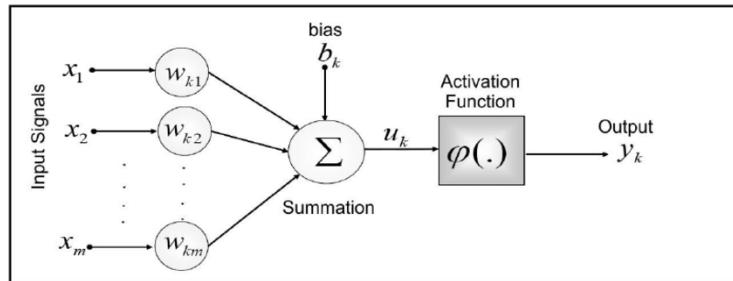
Mirip seperti pembelajaran mesin, *Deep Learning* juga merupakan salah satu bidang dalam bidang kecerdasan buatan. Yang membedakan antara pembelajaran mesin biasa dengan *deep learning* adalah *deep learning* mengacu

pada penggunaan *layer* dan *artificial neural network* hingga mempunyai *multiple hidden layer*, sedangkan *machine learning* lebih mengacu kepada seluruh algoritma yang dapat diterapkan untuk menemukan *rules* dari se rangkaian data. Keuntungan dari model jenis ini adalah model ini dapat memproses masukan yang paling abstrak sekalipun, sehingga menghilangkan proses ekstraksi fitur secara manual [52]. Namun, karena *deep learning* memiliki *layers* yang sangat banyak, maka diperlukan jumlah data yang jauh lebih banyak pula. Karena itu, sebuah model *deep learning* memerlukan daya komputasi yang jauh lebih besar dibandingkan dengan model pembelajaran mesin biasa. Setiap *layer* dapat memiliki fungsi dan tanggung jawabnya masing - masing [50], seperti misal apabila kita menggunakan *deep learning* untuk mendeteksi angka plat nomor di kendaraan bermotor, bisa saja beberapa layer pertama berfungsi untuk mendeteksi letak plat nomor dalam suatu citra, kemudian beberapa layer selanjutnya berfungsi untuk mengambil bentuk dari setiap objek dalam plat nomor tersebut, beberapa layer terakhir berfungsi untuk mengenali bentuk - bentuk dari objek menjadi tulisan teks. Semakin banyak layer yang digunakan, maka semakin tinggi pula kemungkinan kita melakukan sesuatu yang lebih kompleks [50].

2.4 Artificial Neural Network

Dalam pengaplikasianya *Deep learning* menggunakan jaringan saraf neuron yang bekerja layaknya seperti otak manusia. Jaringan neuron ini disebut jaringan neuron buatan atau *artificial neural network* (ANN). ANN disusun oleh neuron buatan (*artificial neuron*) yang dapat menerima input berupa suatu angka, memprosesnya, kemudian menyalurkan output dari proses yang terjadi. Proses yang dilakukan oleh setiap *artificial neuron* adalah mencari nilai *weight* terbaik agar nilai tersebut dapat memberikan *output* dengan error paling kecil setelah diproses dengan sebuah *activation function*. Nilai *weight* pada sebuah neuron buatan dapat berubah dan nilai perubahannya dipengaruhi *learning rate*. *Learning rate* adalah sebuah paramater yang mengatur seberapa besar perubahan yang harus terjadi pada *weight* suatu

neuron dalam mencapai nilai optimal dimana nilai optimal yang dimaksud ialah nilai *weight* yang memberikan error terkecil jika dibandingkan pada data awal.



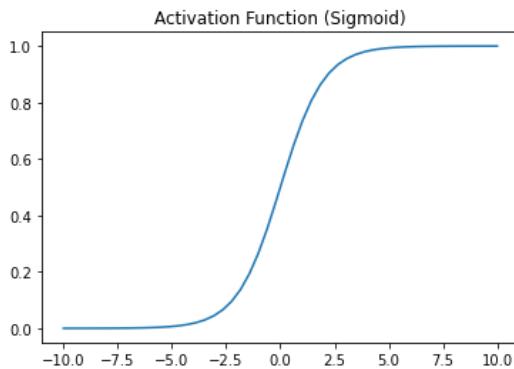
Gambar 2.1. Artificial Neural Network

Nilai sebuah output dari *artificial neural network* juga dipengaruhi oleh bias. Bias pada sebuah model dapat disebabkan oleh dataset awal atau karena kurangnya pemrosesan data. Sebagai contoh, data untuk membedakan gambar kucing dan anjing digunakan untuk membangun sebuah model klasifikasi anjing dan kucing, terdapat 500 gambar data anjing yang digunakan, namun hanya ada 20 gambar kucing yang digunakan. Tentunya dengan dataset ini, model akan cenderung memprediksi atau mengklasifikasikan gambar menjadi kelas anjing. Bias yang sama pun juga dapat terjadi pada sebuah prediksi nilai atau klasifikasi teks. Bias yang dapat terjadi lainnya adalah jika model memiliki data *train* duplikat. Hal ini dapat memberikan *overfitting* karena saat diuji coba dengan gambar yang berbeda, nilai performa model menurun.

Pada sebuah *artificial neural network* terdapat beberapa *activation function* yang dapat diaplikasikan. *Activation function* mengatur apakah sebuah neuron harus diaktifkan dan nilainya digunakan atau tidak. Hal ini berguna dalam memprediksi ataupun mengklasifikasi suatu hal berdasarkan nilai dari input *artificial neural network* dimana dengan sebuah *threshold*, nilai input tertentu dapat memberikan nilai output 0 jika data input dianggap tidak berguna dalam memprediksi. Terdapat tiga *activation function* yang umum digunakan yaitu fungsi sigmoid, ReLu, dan TanH.

- **Sigmoid**

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (2.1)$$



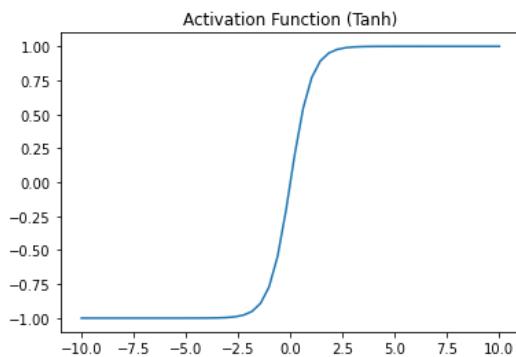
Gambar 2.2. Fungsi Sigmoid

Fungsi aktivasi yang umum digunakan pertama adalah fungsi sigmoid, fungsi sigmoid memberikan nilai dibawah 0,5 pada nilai input negatif, dan nilai 0 pada nilai negatif yang lebih kecil dari -5. Sebaliknya, fungsi ini memberikan nilai diatas 0,5 pada nilai positif dan nilai 1 pada nilai input diatas 5. Jika nilai input 0, maka fungsi ini akan memberikan nilai 0,5. Nilai maksimal dari fungsi ini adalah 1 dan nilai minimalnya adalah 0.

- **Tanh**

$$\tanh x \quad (2.2)$$

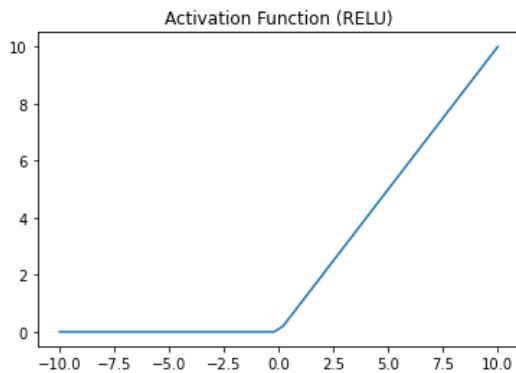
Fungsi aktivasi kedua adalah tanh, jika sebelumnya fungsi sigmoid memiliki nilai minimal 0, maka pada fungsi tanh, nilai minimal yang dimiliki adalah -1 sedangkan nilai maksimal yang dimiliki adalah 1. Fungsi tanh akan memberikan nilai negatif pada input negatif, nilai 0 pada input 0, dan nilai positif pada input positif.



Gambar 2.3. Fungsi Tanh

- **ReLU**

$$Relu(x) = \max(0, x) \quad (2.3)$$

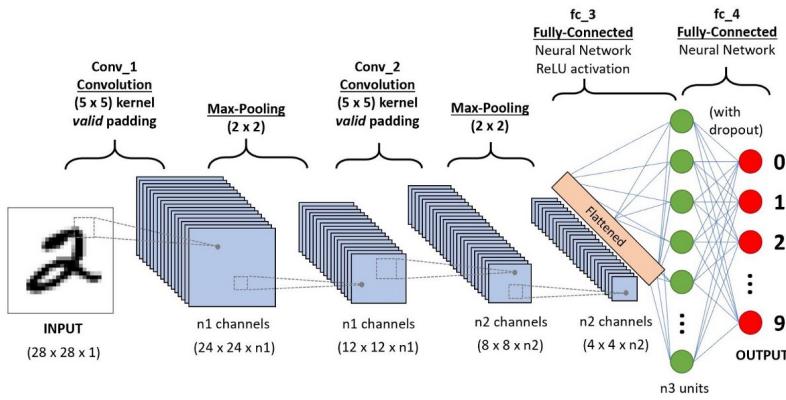


Gambar 2.4. Fungsi ReLu

Fungsi aktivasi ketiga adalah ReLu, fungsi ReLu memiliki nilai minimal 0 namun tidak mempunyai nilai maksimal. ReLu merupakan suatu fungsi aktivasi linear yang sering digunakan dalam mengklasifikasi gambar. Fungsi ReLu akan memberikan nilai 0 pada input 0 dan negatif sedangkan pada input positif, ReLu akan memberikan output nilai input itu sendiri. Terdapat juga fungsi ReLu yang memberikan nilai negatif pada input negatif, fungsi ReLu yang memberikan nilai negatif disebut dengan leaky ReLu.

2.5 Convolutional Neural Network

Convolutional neural network adalah sebuah jaringan *neural network* yang memanfaatkan operasi matematika konvolusi dan pooling dalam memproses matriks atau tensor inputnya. *Convolutional neural network* umum digunakan dalam pengolahan citra karena dapat mengekstrak fitur pada gambar dengan baik.



Gambar 2.5. Contoh Arsitektur CNN

2.5.1 Konvolusi

Konvolusi merupakan suatu operasi matematika yang sering dilakukan untuk melakukan pemrosesan gambar. Konvolusi dilakukan untuk mengekstrak fitur-fitur pada gambar agar perbedaan nilai pada fitur yang ingin diekstrak semakin besar dan fitur yang akan diproses menjadi lebih mudah dilihat oleh komputer karena memiliki nilai yang lebih besar serta memori atau ukuran gambar yang lebih kecil. Proses konvolusi dimulai dengan mempersiapkan sebuah matriks kernel yang berukuran sama atau kurang dari ukuran matriks gambar yang ingin diproses. Umumnya matriks kernel ini berukuran 3×3 . Kemudian pada matriks awal akan diambil sebuah bagian dari matriks yang berukuran sama kemudian dikalikan dengan matriks kernel. Hasil dari perkalian tersebut kemudian dijumlahkan menjadi satu nilai yang menjadi satu elemen pada matriks yang baru / matriks dari hasil konvolusi. Pada CNN, nilai kernel matriks ini akan didefinisikan oleh *neural network*.

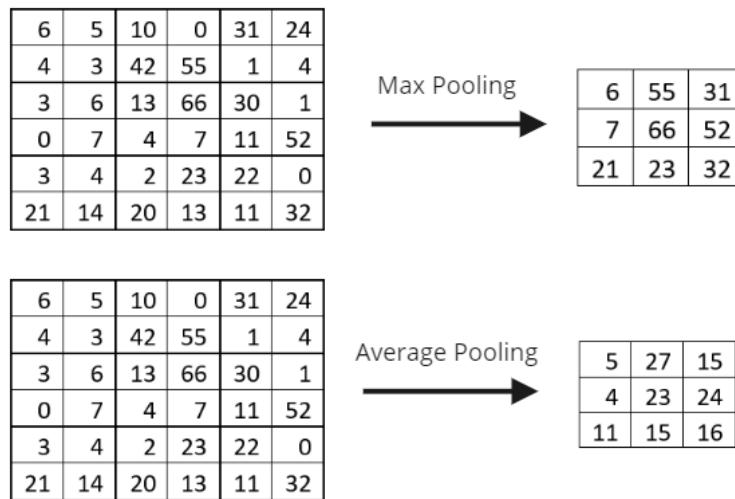
$$\begin{array}{|c|c|c|c|c|} \hline 2 & 0 & 2 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 2 & 2 & 1 \\ \hline 0 & 2 & 0 & 0 & 1 \\ \hline 1 & 1 & 2 & 0 & 2 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 2 & 6 & 4 \\ \hline 3 & 5 & 3 \\ \hline 5 & 4 & 2 \\ \hline \end{array}$$

Gambar 2.6. Konvolusi

2.5.2 *Pooling*

Pooling merupakan suatu operasi matematika yang digunakan untuk memperkecil ukuran matriks dengan memilih atau memproses nilai pada matriks. *Pooling* pada pemrosesan citra umumnya dilakukan pada setiap bagian matriks dengan ukuran 2x2 atau 3x3. Contoh dari operasi *pooling* dapat dilihat pada gambar 2.7. Terdapat beberapa jenis *pooling*, dan beberapa *pooling* yang sering digunakan ialah *max pooling* dan *average pooling*. *Max pooling* mengambil elemen matriks dengan nilai tertinggi pada suatu sekumpulan matriks sesuai dengan ukuran yang sudah didefinisikan. Misalnya jika matriks berukuran 4x4 mengalami proses *pooling* 2x2, maka hasil dari matriks tersebut adalah 1 elemen matriks dari setiap bagian matriks berukuran 2x2 pada matriks 4x4 dimana elemen yang telah diproses tidak diproses kembali. Sehingga hasil dari matriks 4x4 yang mengalami proses *pooling* 2x2 adalah matriks berukuran 2x2.

Sedangkan jenis *pooling* yang sering digunakan lainnya adalah *average pooling* dimana jika sebelumnya pada *max pooling* elemen dengan nilai tertinggi yang dipilih, maka pada *average pooling* maka akan dipilih nilai rata-rata dari *pooling* tersebut. Jika misalnya pada sebuah matriks dilakukan proses *pooling* 2x2, maka rata-rata dari empat elemen dari bagian matriks yang berukuran 2x2 akan menjadi elemen dari matriks hasil *pooling*. Perbedaan jenis *pooling* tidak mempengaruhi ukuran matriks dari hasil pemrosesan *pooling*, namun tentunya mempengaruhi nilai hasil *pooling*.



Gambar 2.7. Contoh *Pooling* 2x2

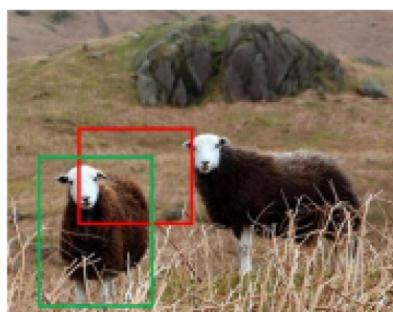
2.6 Deteksi Objek

Deteksi objek (*object detection*) adalah suatu teknik pada *computer vision* dimana suatu komputer dapat mengenali dan mengetahui lokasi dari objek yang menjadi fokus secara otomatis layaknya manusia. *Object detection* dapat dicapai dengan penggunaan *deep learning*. Salah satu kelemahan dalam pembuatan sebuah komputer pendeksi objek adalah, tingginya daya komputasi yang diperlukan dan dalam mendeksi *multiple class* membutuhkan dataset yang banyak sehingga kebutuhan *storage* pun meningkat. Sebagian besar detektor objek beroperasi dengan memilih sekumpulan kandidat region yang mengandung objek fokus dari gambar dan mengklasifikasikannya ke dalam daerah objek *foreground* dan daerah yang tidak memiliki objek (biasanya disebut sebagai latar belakang). Pada dasarnya, pendekatan tersebut dapat mengurangi tugas deteksi objek menjadi tugas klasifikasi gambar yang lebih sederhana. Kandidat daerah yang akan diklasifikasikan dapat diperoleh dengan mudah dengan cara *brute force* dengan menggeser jendela $m \times n$ di atas gambar atau yang sering disebut algoritma *sliding window*. Mengikuti metode ekstraksi fitur dari *sliding window*, terdapat metode dimana setiap *region* akan dinilai berdasarkan kerapatan tepi, ukuran, dan lokasi pada objek dan metode

lain dimana gambar tersebut akan disegmentasi terlebih dahulu menjadi beberapa bagian kemudian mengkombinasikannya berdasarkan kemiripannya. Salah satu metode yang mensegmentasi gambar dan mengelompokkan gambar tersebut adalah *selective search* [53].

Dalam menunjukkan lokasi objek, seringkali sebuah *bounding box* dibentuk untuk menandakan lokasi dari objek yang menjadi fokus. *Bounding box* juga digunakan dalam metode *selective search* untuk menjadi kandidat objek deteksi. Kandidat objek harus meliputi semua objek pada gambar sehingga tidak ada objek yang terlewati dan objek-objek tersebut dapat diklasifikasi. Di sisi lain *bounding box* pada kandidat objek yang terlalu banyak dapat mengakibatkan *bounding box* menimpa satu sama lain pada satu objek yang sama.

Untuk mengatasi hal ini dan mengambil *bounding box* terbaik, maka suatu metode yang dinamakan *non-max suppression* per diaplikasikan. Pertama-tama, semua *bounding box* perlu dicek apakah menimpa objek yang sama atau tidak, hal ini dapat dilakukan dengan menganalisa koordinat dari *bounding box*, kemudian nilai *interference over union* (IoU) pada *bounding box* yang meliputi objek yang sama akan diperiksa, jika nilai IoU pada *bounding box* di atas 0,5 maka *bounding box* tersebut tidak dibuang kemudian jika terdapat beberapa *bounding box* yang memiliki nilai IoU di atas 0,5, maka akan dipilih *bounding box* dengan nilai IoU terbesar [54].



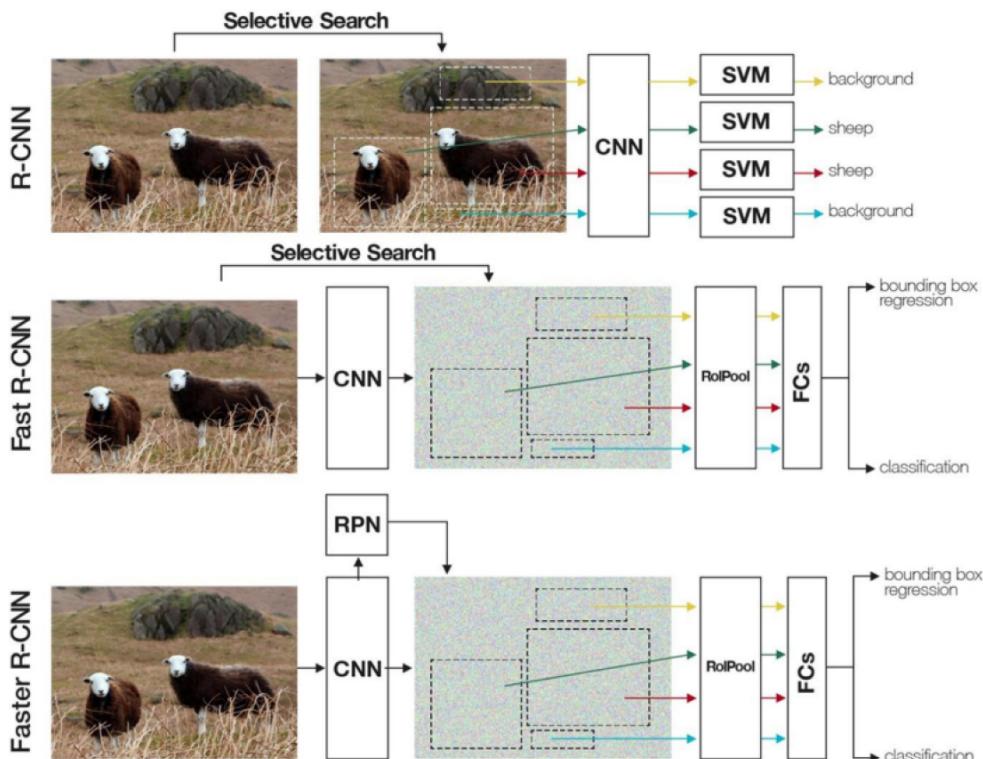
$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

Gambar 2.8. *Intersect over Union*

2.6.1 Two-Stage Object Detectors

Two-stage object detectors adalah suatu prinsip pada deteksi objek dengan mengikuti prinsip mencari daerah kandidat atau daerah proposal. Kemudian daerah proposal tersebut akan dideteksi secara lebih lanjut. Seperti namanya *two stage object detectors* memiliki dua tahapan detektor dimana tahap pertama adalah pemberian proposal dan tahap kedua adalah klasifikasi dan deteksi objek [54].

Salah satu contoh algoritma yang menerapkan *two-stage object detectors* adalah R-CNN. Pada R-CNN, algoritma pencarian selektif mengekstrak *region proposal* dari gambar, yang kemudian akan menghasilkan satu set kandidat objek yang diberi *bounding box*. Setelah memperoleh proposal wilayah, jaringan saraf convolutional digunakan sebagai ekstraktor fitur di atas setiap proposal yang diekstraksi, menghasilkan *feature* gambar dari setiap proposal.



Gambar 2.9. Contoh *Two Stage Detector*

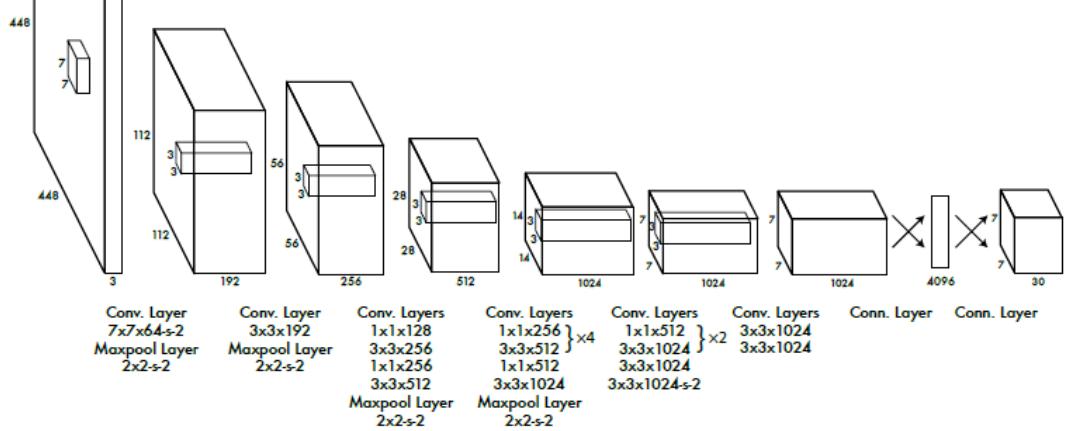
Secara khusus, untuk setiap gambar, pembuatan *region proposal* yang membentuk sebuah *bounding box* suatu objek harus mempunyai IoU melebihi nilai ambang (0,5 umumnya digunakan). Hal ini dianggap sebagai contoh positif untuk kelas objek yang sesuai, sementara proposal yang memiliki nilai IoU lebih rendah dari ambang (0,5) dianggap sebagai contoh negatif untuk semua kelas objek dan dianggap sebagai latar belakang (bukan objek). Setelah proposal dibentuk, maka algoritma lanjutan yang menggunakan SVM akan memprediksi kelas pada objek yang menjadi proposal dan objek proposal yang dipilih adalah proposal yang memiliki IoU tertinggi dari suatu objek yang tumpang tindih.

2.6.2 Single-Stage Object Detector

Selain *two-stage object detector* adapula deteksi objek dengan menggunakan *single stage*. Pada metode ini, *region proposal* tidak dibentuk sehingga objek dideteksi dan diklasifikasi secara langsung. Tentunya karena tidak memerlukan tahapan pembentukan *region proposal* maka algoritma yang menggunakan metode ini memproses gambar lebih cepat dan mendekripsi objek secara lebih cepat. Beberapa contoh dari penerapan *Single-stage object detector* adalah algoritma *Single Shot Detector (SSD)* dan *You Only Look Once (YOLO)* [55].

YOLO menggunakan 24 lapis jaringan konvolusi untuk ekstraksi fitur, diikuti oleh 2 lapisan yang *fully connected*. 20 lapisan konvolusional pertama dilatih pada ImageNet untuk klasifikasi, dan setelah menambahkan lapisan yang tersisa, model selanjutnya dilatih untuk mendekripsi objek. YOLO membagi gambar menjadi sel menjadi ukuran SxS (umumnya 7x7) dan memprediksi kotak pembatas bersama dengan skor *confidence*, serta distribusi probabilitas kelas untuk setiap sel grid. Setiap pembatas kotak diberikan parameter oleh titik pusat kotak relatif terhadap sel grid, yang lebar dan tingginya relatif terhadap keseluruhan gambar [54]. Skor kepercayaan dari kotak pembatas secara formal didefinisikan sebagai IoU antara kotak pembatas yang diprediksi dan kotak kebenaran dasar yang sesuai. Perhatikan bahwa

tidak seperti Faster R-CNN yang memprediksi offset dari kotak jangkar, YOLO memprediksi koordinat kotak pembatas akhir secara langsung.



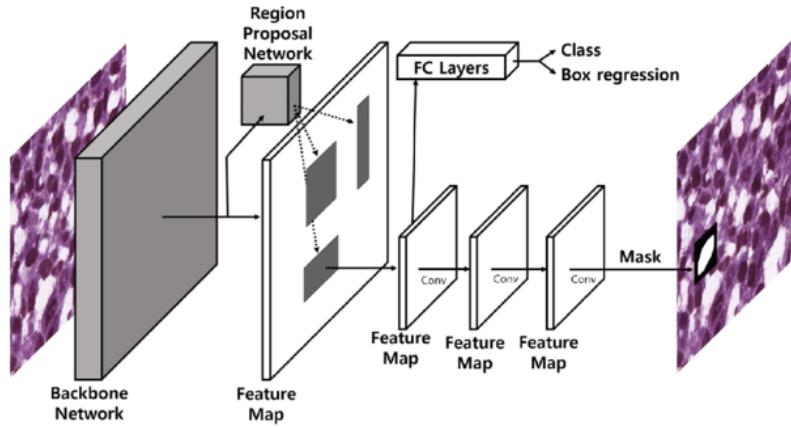
Gambar 2.10. Arsitektur YOLO [1]

2.7 Algoritma Deteksi Objek Mask R-CNN

Mask R-CNN [56] adalah jaringan saraf dalam untuk memecahkan masalah segmentasi instans. Dengan kata lain, dapat memisahkan objek yang berbeda dalam gambar atau video. Dengan memberikan input gambar, Mask R-CNN akan memberikan kotak pembatas objek, kelas, dan *mask*. Mask R-CNN merupakan pengembangan dari *faster R-CNN* dimana sebuah *mask* akan dibentuk dalam mendeteksi objeknya.

Saat ini Mask R-CNN pertama kali dikemukakan pada tahun 2017 dan sejak saat itu menjadi salah satu algoritma deteksi objek yang terkenal dan menarik untuk diimplementasikan. Pengaplikasian Mask R-CNN sudah dilakukan di berbagai penelitian seperti deteksi algae, deteksi sel kanker, deteksi pneumonia, dan masih banyak lagi. Mask R-CNN adalah salah satu algoritma yang memanfaatkan *two stage detector* dalam mendeteksi objek dan dapat mensegmentasi objek hingga level piksel.

Ada dua tahap Mask RCNN. Pertama, menghasilkan proposal tentang wilayah di mana mungkin terdapat objek berdasarkan gambar input. Kedua, memprediksi kelas objek, menyempurnakan kotak pembatas dan menghasilkan



Gambar 2.11. Arsitektur Mask R-CNN [2]

mask di tingkat piksel objek berdasarkan proposal tahap pertama. Tentunya seperti algoritma deteksi objek lainnya, pelabelan kelas dan penunjukkan lokasi objek yang ingin dideteksi dibutuhkan pada data *train*. Mask R-CNN mengambil data kelas dan objek yang telah diberikan *mask* kemudian mengekstrak fitur-fitur pada gambar data *train*.

Dalam mengambil target untuk melatih proses pengekstrakan fitur secara lebih dalam, maka sebuah target perlu dibentuk. Pada hal ini, Mask R-CNN memanfaatkan *anchor box* untuk memberikan perkiraan dimanaakah objek yang perlu dideteksi berada, jika *anchor box* mempunyai IoU lebih dari sama dengan 0,5 dengan *bounding box* awal maka *anchor box* tersebut disimpan untuk menjadi target kecuali jika *bounding box* tersebut *overlap* dengan *anchor box* lain, maka *anchor box* yang mempunyai nilai IoU terbaik yang akan dipilih menjadi target. *Anchor box* yang disimpan akan menjadi target bagi *backbone* untuk mengekstrak fitur menggunakan neural network.

Pengekstrakan fitur dimulai dengan proses konvolusi pada jaringan *backbone*. Umumnya *backbone* yang digunakan pada algoritma ini adalah *Residual Network* atau yang disingkat ResNet. ResNet merupakan sebuah jaringan yang dibentuk oleh berbagai konvolusi, proses *pooling*, dan lapisan neural network. Saat ini terdapat lima layer atau lima jenis ResNet yang dibedakan menurut jumlah layernya. Untuk 18 layer ResNet, jaringan tersebut disusun oleh 1

layer konvolusi 7x7-64 neural network, kemudian proses pooling 3x3 (tidak dihitung sebagai layer), kemudian 4 layer konvolusi 3x3-64 neural network, 4 layer konvolusi 3x3-128 neural network, 4 layer konvolusi 3x3-256 neural network, 4 layer konvolusi 3x3-512 neural network, dan layer terakhir adalah *fully connected layer* atau yang disebut sebagai FC dengan *softmax function* untuk memberikan output dengan berbagai kelas.

Sedangkan pada contoh layer ResNet yang lain, contohnya ResNet 101, layer pada ResNet akan terdiri dari 101 layer dan dapat dilihat pada gambar 2.12. Jaringan pada *backbone* akan menghasilkan sebuah *Region Proposal Network (RPN)* yang merupakan hasil dari *anchor box* yang mempunyai IoU lebih besar dari 0,5 dengan *bounding box* asli. Dan *feature map* yang memuat data hasil dari konvolusi dan pelatihan *neural network* yang berada pada *backbone*.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
				3×3 max pool, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Gambar 2.12. Macam - macam *Residual Network (ResNet)* [3]

Hasil dari ekstraksi fitur oleh jaringan *backbone* berupa sebuah *heatmap* hasil konvolusi dari algoritma ResNet yang disebut *feature map*, hal ini akan memudahkan tahapan memprediksi kelas, ukuran objek, dan pembuatan *mask*. Dalam memprediksi kelas dan memberikan *bounding box* pada objek, *feature map* diproses menggunakan *fully connected layer*. *Fully connected layer* merupakan sebuah lapisan *neural network* yang mengkoneksikan setiap neuron

pada satu layer ke setiap neuron pada layer selanjutnya umumnya digunakan pada lapisan output untuk memberikan kelas. Selain itu, *feature map* juga digunakan untuk membentuk *mask* menggunakan lapisan konvolusi.

Dalam melatih model Mask R-CNN, terdapat beberapa metriks yang perlu diketahui untuk mengetahui seberapa bagus proses *training* yang biasanya dinyatakan dalam metriks loss. dalam Mask RCNN terdapat 5 metriks loss yang dibedakan menjadi loss RPN dan loss MRCNN. Loss RPN merupakan loss yang terjadi pada pembuatan dan pelatihan jaringan *backbone* dalam memberikan *region proposal*. Loss RPN dibedakan menjadi dua yaitu loss RPN *bounding box* dan loss RPN *class*. Loss RPN *bounding box* merupakan loss yang terjadi untuk mengetahui seberapa besar perbedaan *bounding box* pada objek yang diprediksi dengan *bounding box* aslinya. Sedangkan pada Loss RPN *class* merupakan loss yang terjadi saat ada kesalahan prediksi objek menjadi *background* ataupun prediksi *background* menjadi objek.

$$\mathcal{L} = \mathcal{L}_{\text{RPN-cls}} + \mathcal{L}_{\text{RPN-box}} \quad (2.4)$$

$$\mathcal{L}_{\text{RPN-cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i) \quad (2.5)$$

$$\mathcal{L}_{\text{RPN-box}} = \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*) \quad (2.6)$$

Kemudian dalam loss yang sering kali disebut loss MRCNN merupakan loss yang terjadi tahapan setelah pembuatan *feature map* dan digunakan untuk mengetahui seberapa bagus performa dari output pada Mask RCNN. Loss MRCNN dibedakan menjadi tiga jenis yaitu loss *class*, loss *bounding box*, dan loss *mask*. *loss class* adalah loss yang terjadi saat adanya kesalahan prediksi pada kelas objek, jika sebelumnya pada loss rpn, perbedaan kelas hanyalah objek atau tidak (*foreground or background*) pada loss MRCNN, kelas yang diprediksi adalah kelas nama objek yang diteliti. Kemudian pada loss *bounding box* merupakan hal yang hampir sama pada loss rpn *bounding box*, perbedaannya ialah, loss ini merupakan perbandingan output akhir prediksi

dengan data awal. Kemudian loss terakhir adalah loss mask yang merupakan metriks untuk melihat seberapa berbedanya mask yang dibentuk dengan mask data awal. Dibawah ini merupakan persamaan dari loss MRCNN.

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} \quad (2.7)$$

$$\mathcal{L}_{\text{fr}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} \quad (2.8)$$

$$\mathcal{L}_{\text{fr}}(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*) \quad (2.9)$$

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i) \quad (2.10)$$

$$\mathcal{L}_{\text{box}} = \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*) \quad (2.11)$$

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)] \quad (2.12)$$

2.8 Metode Analisa Performa Model

Terdapat beberapa metode yang bisa dilakukan untuk mengetahui apakah suatu model memiliki akurasi yang cukup. Penelitian ini menggunakan beberapa formula yang sudah ditentukan seperti *recall*, *precision*, *f1-score* dan *confusion matrix*.

2.8.1 *Recall*

Recall adalah formula yang harus digunakan ketika kita memiliki data yang tidak seimbang. Berbeda dengan akurasi yang hanya menghitung persentase model memprediksi hasil yang sesuai dengan label secara keseluruhan, *recall* akan menghitung rasio nilai yang diprediksi positif dengan total keseluruhan nilai yang positif [57]. Rumus 2.13 merupakan rumus untuk menghitung *recall*.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.13)$$

2.8.2 *Precision*

Seringnya, kita tidak hanya melihat tingkat akurasi suatu model hanya dengan besaran *recall* maupun tingkat akurasi nya. *Precision* adalah formula untuk menghitung rasio dari prediksi TP (*True Positive*) yang benar dengan keseluruhan prediksi. Apabila prediksi yang dilakukan oleh model kita ternyata memiliki tingkat presisi yang tinggi namun memiliki tingkat *recall* yang rendah, ada kemungkinan model tidak dapat melakukan prediksi pada data yang bersifat negatif [57]. Rumus 2.14 merupakan rumus untuk menghitung nilai dari *precision* suatu model.

$$Precision = \frac{TP}{TP + FP} \quad (2.14)$$

Baik *recall* maupun *precision* merupakan nilai yang cukup penting terutama pada data yang tidak seimbang. Terdapat 3 kelfndisi yang umum terjadi pada saat membandingkan antara *precision* dengan *recall*.

- *Recall* tinggi, *Precision* rendah Sebagian besar data positif dapat diprediksi dengan benar (*False Negative* yang Rendah), namun hanya ada sebagian kecil data negatif yang diprediksi dengan benar (*True Negative* rendah).
- *Recall* rendah, *Precision* tinggi Hasil prediksi model memiliki banyak sekali prediksi negatif (*False Negative* tinggi), namun apabila digunakan untuk memprediksi data positif, maka hasil prediksi sebagian besarnya adalah benar (*False Positive* rendah).
- *Recall* tinggi, *Precision* tinggi Merupakan hasil yang ideal dalam pembuatan model pembelajaran mesin. Disini didapatkan bahwa baik hasil prediksi untuk data positif maupun hasil prediksi untuk data negatif sebagian besarnya adalah benar (*True Positive* dan *True Negative* tinggi).

Tabel 2.1. Contoh *Confusion Matrix*

		Aktual	
		Positif	Negatif
Prediksi	Positif	TP	FP
	Negatif	FN	TN

2.8.3 mAP@50

Dalam mengukur performa sebuah deteksi objek, sebuah threshold IoU 0,5 merupakan parameter yang umum digunakan dimana jika IoU deteksi bernilai lebih dari sama dengan 0,5, maka dapat dikatakan bahwa deteksi yang dihasilkan adalah bagus. Adapula parameter performa yang dinamakan *average precision* yang didefinisikan sebagai luasan dari area di bawah kurva *recall-precision*. Umumnya parameter performa ini ditulis sebagai mAP@50

2.8.4 F1-Score

F1-Score adalah besaran yang berasal dari rata - rata harmonik dari *recall* dan *precision*. Rata - rata harmonik dipilih karena akan menghasilkan nilai rata - rata yang lebih rendah dalam kondisi tidak seimbang apabila dibandingkan dengan rata - rata aritmatis biasa. Dengan rata - rata seperti itu, suatu model dapat menjadi lebih rentan terhadap bias dan memudahkan pada saat pembuatan model [57]. Rumus 2.15 merupakan rumus untuk menghitung f1-score.

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.15)$$

2.8.5 Confusion Matrix

Confusion Matrix adalah tabel kesimpulan yang berisi jumlah prediksi baik yang benar maupun yang salah dan nilai label baik yang benar maupun salah. Biasanya tabel jenis ini digunakan untuk tugas yang bersifat klasifikasi dan berfungsi untuk memvisualisasi bagaimana suatu performa model dalam suatu dataset.

Tabel 2.1 adalah contoh tabel *confussion matrix* untuk prediksi dengan

2 label. Apabila melihat pada tabel tersebut, dapat terlihat bahwa jumlah prediksi dipecah menjadi masing - masing kelas. Diharapkan dengan dipecah menjadi beberapa kelas seperti itu, akan membuat proses pengujian lebih mudah karena akan lebih mudah melihat pada saat model memprediksi jenis data apa yang masih memiliki tingkat akurasi yang kurang bagus. Terdapat beberapa hal yang harus diketahui untuk dapat memahami sebuah tabel *confussion matrix*, yaitu :

- Positif (P) Berisi data yang bernilai positif, baik data tersebut berasal dari hasil prediksi maupun data aktual yang didapat dari dataset.
- Negatif (N) Berisi data yang bernilai negatif, baik data tersebut berasal dari hasil prediksi maupun data aktual yang didapat dari dataset.
- *True Positive* (TP) Suatu kondisi dimana baik hasil prediksi maupun data aktual sama - sama bernilai positif. Semakin tinggi nilai dari TP, semakin akurat pulalah model yang sudah dibuat.
- *False Positive* (FP) Suatu kondisi dimana hasil prediksi adalah positif, namun pada data aktual bernilai negatif. Biasanya, semakin tinggi nilai dari FP ini, maka model semakin memiliki kecenderungan untuk mengeluarkan nilai positif dibanding negatif atau terjadinya bias pada model.
- *True Negative* (TN) Suatu kondisi dimana hasil prediksi dan data aktual bernilai negatif. Semakin tinggi nilai FN berarti semakin akurat model yang sudah dibuat.
- *False Negative* (FN) Suatu kondisi dimana hasil prediksi adalah negatif, namun pada data aktual bernilai positif. Biasanya, semakin tinggi nilai dari TN ini, maka model semakin memiliki kecenderungan untuk mengeluarkan nilai negatif dibanding positif.

Halaman ini sengaja dikosongkan

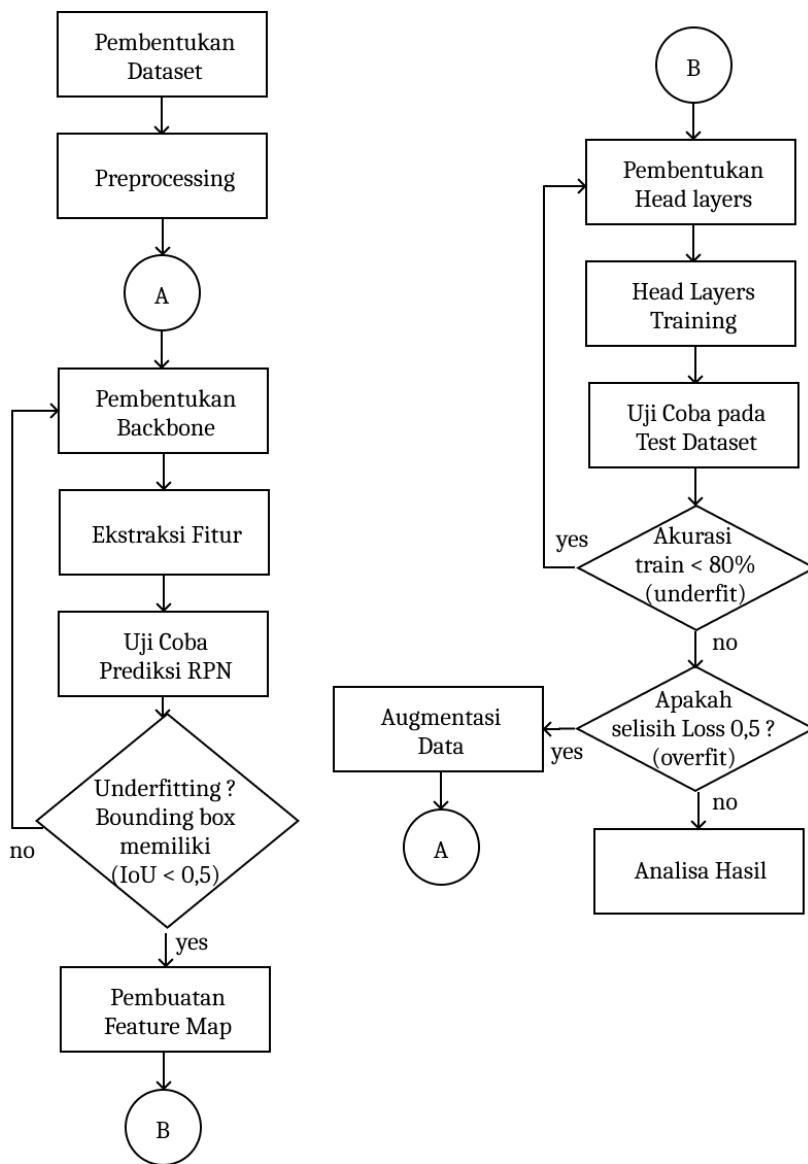
BAB 3

METODOLOGI

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Sedangkan untuk bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

Gambar 3.1 merupakan bagan yang menunjukkan secara garis besar bagaimana proses pembentukan model pada penelitian ini. Penelitian dimulai dengan pembentukan dataset, kemudian dilanjutkan dengan proses *preprocessing*. Setelah pembentukan jaringan *backbone*, maka akan dilanjutkan dengan proses ekstraksi fitur menggunakan jaringan *backbone* yang telah dibentuk. Kemudian jaringan *backbone* tersebut akan menghasilkan *regional proposal network* dan prediksi *bounding box*. Jika model tidak dapat mendeteksi *bounding box* dengan baik dengan kata lain, kemiripan *bounding box* yang diprediksi memiliki irisan (IoU) kurang dari 0,5 dengan *bounding box* asli, maka jaringan *backbone* perlu dikonfigurasi ulang menggunakan model *backbone* yang berbeda. Kemudian jika model *backbone* telah memberikan *bounding box* yang sesuai, maka model *backbone* dapat digunakan untuk menghasilkan *feature map*.

Setelah *feature map* dibentuk, maka *head layers* dibentuk untuk memprediksi kelas dan *bounding box* serta menggenerasi *mask*. Setelah itu lapisan jaringan pada *head* perlu dilatih agar dapat mendeteksi kelas, *bounding box*, Dan menggenerasi *mask*. Setelah proses *training* selesai maka model keseluruhan perlu diuji secara keseluruhan dengan cara mengujinya dengan dataset tes dan dataset validasi. Setelah itu jika terjadi *underfitting*, maka lapisan *head* perlu dimodelkan ulang, namun jika terjadi *overfitting* yaitu dimana loss kumulatif yang terjadi pada *train dataset* memiliki nilai 0,5



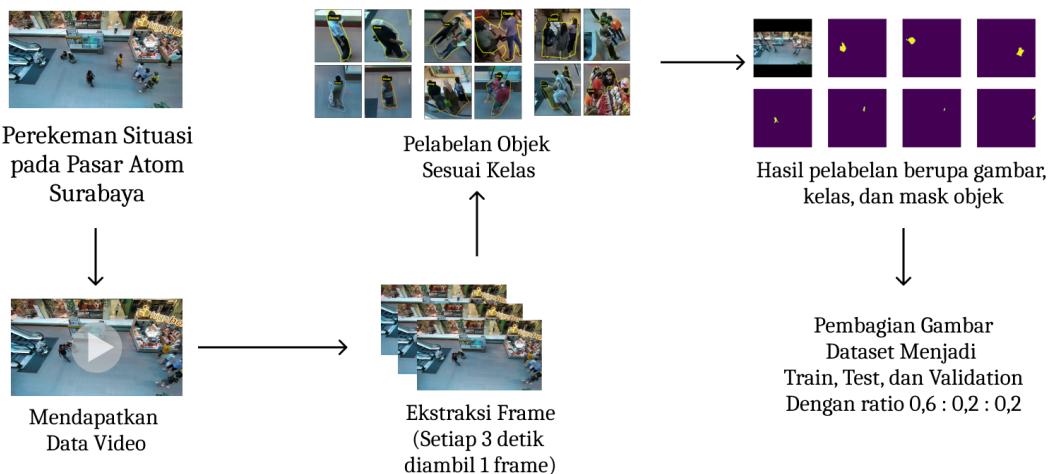
Gambar 3.1. Bagan Umum Metodologi Penelitian

lebih kecil dibandingkan *test dataset* maka augmentasi data dan penambahan regularisasi diperlukan untuk mengurangi selisih *loss* yang terjadi.

3.1 Pembentukan Dataset

Pada proses pembentukan dataset, terdapat dua langkah penting yaitu pengumpulan data serta anotasi data. pengumpulan data dilakukan dengan mengambil video pada lokasi yang menjadi tempat pengamatan, pada penelitian ini, peneliti menetapkan Pasar Atom yang berada pada alamat Jl.

Bunguran No.45, Bongkaran, Kec. Pabean Cantikan, Surabaya. Pasar Atom dinilai sebagai lokasi yang tepat dikarenakan pasar tersebut dinilai modern dan ramai pengunjung.



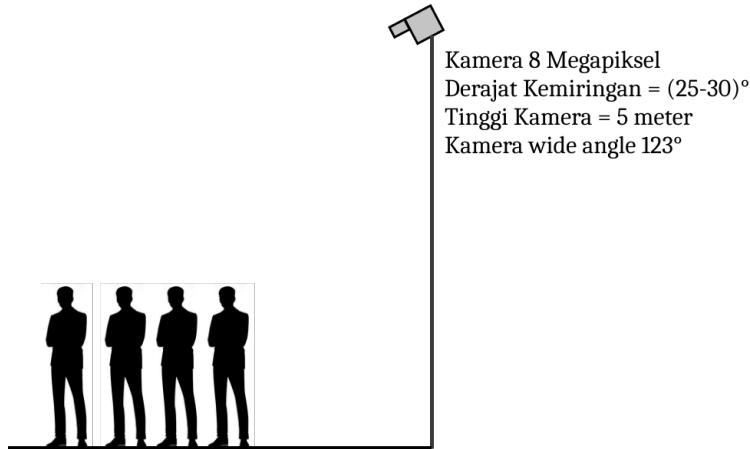
Gambar 3.2. Alur Pembentukan Dataset

3.1.1 Pengumpulan Data

Pada proses pengumpulan data, data mengenai aktivitas pembeli, penjual, dan petugas yang berada pada Pasar Atom akan diambil dalam bentuk video. Data tersebut akan diambil di berbagai lokasi pengamatan yang masih berada pada kawasan Pasar Atom. Data video tersebut kemudian akan diubah menjadi gambar dengan cara mengekstrak satu gambar setiap tiga *frame* pada video. Beberapa lokasi yang menjadi pengamatan ialah pintu masuk utama Pasar Atom, lantai dua Pasar Atom, dan eskalator utama Pasar Atom. Pengumpulan data dilakukan menggunakan kamera 8 Mega Pixel dimana tinggi kamera adalah 5 meter dan mempunyai sudut 30 derajat terhadap sumbu X. Data yang dikumpulkan merupakan data video berukuran 1920 x 1080 px.

3.1.2 Anotasi Data

Pada proses ini, data yang telah berupa gambar tersebut akan dianotasi menjadi tiga kelas sebagai berikut :



Gambar 3.3. Konfigurasi Pengambilan Gambar

- **Kelas *Person***

Kelas *Person* diberikan kepada objek orang yang tidak berada di dekat orang lain dalam jarak radius satu meter.

- **Kelas *Group***

Kelas *Group* diberikan kepada objek kelompok orang yang terdiri dari dua hingga tiga orang dalam radius satu meter.

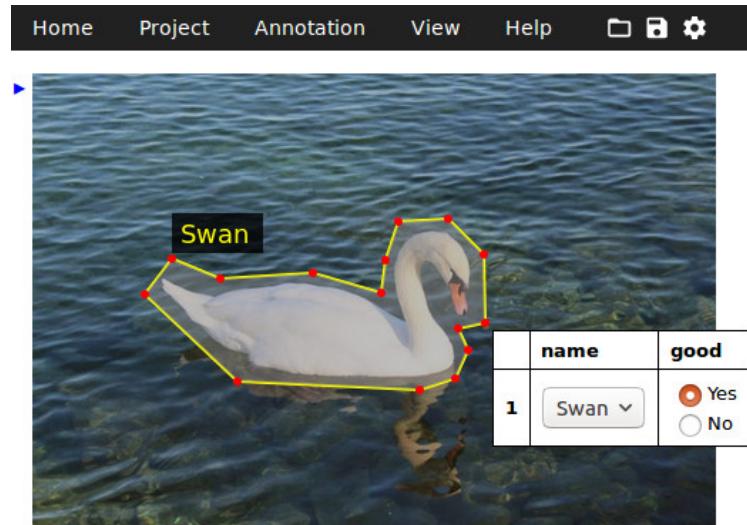
- **Kelas *Crowd***

Kelas *Crowd* diberikan kepada objek kelompok orang yang terdiri dari empat atau lebih orang dalam radius satu meter.

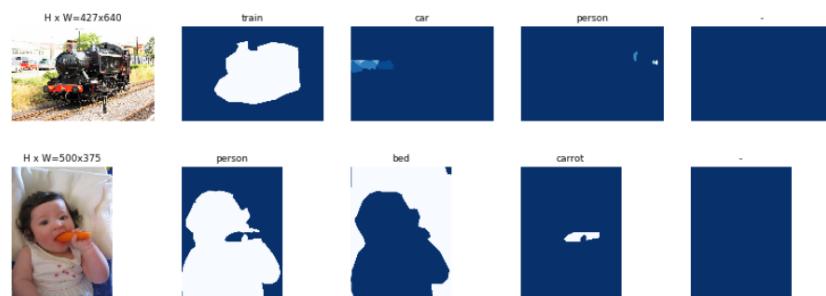
Proses anotasi dilakukan dengan *software VGG Image Annotator (VIA)* [4] dimana pelabelan objek dilakukan dengan cara mensegmentasi objek dengan sebuah *mask*. Proses anotasi menggunakan VIA adalah sebagai berikut.

Kemudian hasil dari pelabelan tersebut akan diubah menjadi format *Java-Script Object Annotation (.Json)* yang berisi mengenai koordinat segmentasi serta nama kelas objek hasil anotasi. Hasil anotasi dan gambar yang telah diambil dari video inilah yang akan menjadi dataset penelitian mengenai deteksi kerumunan. Dataset kemudian akan dibagi menjadi *train set*, *test*

set, dan *val set* dengan proporsi masing-masing secara berurutan 60% – 20% – 20%. *train set* akan digunakan untuk melatih algoritma pembelajaran mesin, *val set* akan digunakan untuk mengukur performa model pada setiap iterasi, sedangkan *test set* akan digunakan untuk memeriksa performa model akhir. *Train set*, *val set*, dan *Test set* memiliki gambar yang berbeda.



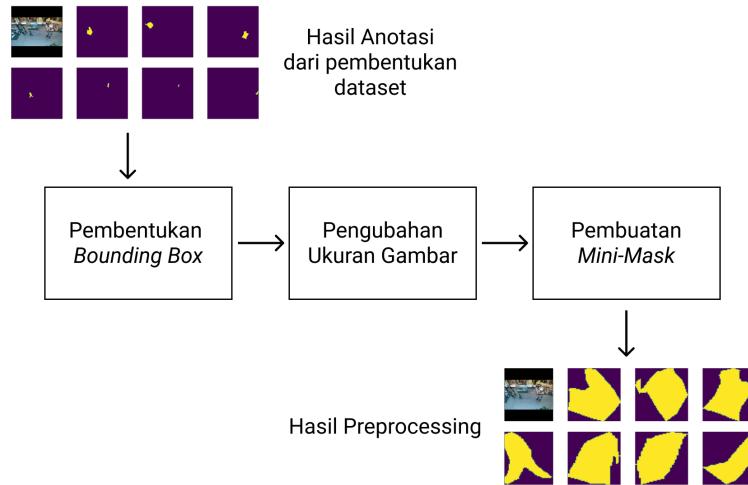
Gambar 3.4. Software VIA [4]



Gambar 3.5. Contoh Hasil Uji Coba Pembuatan Dataset [5]

3.2 Data *Preprocessing*

Pada tahap *preprocessing* dataset yang telah dibentuk pada proses pembuatan dataset akan diolah terlebih dahulu agar ukuran data menjadi lebih kecil dan proses ekstraksi fitur maupun proses *training* dapat dijalani secara lebih cepat dan tidak memerlukan banyak sumber daya komputasi. Seperti



Gambar 3.6. Tahap *Preprocessing*

yang telah tertera pada gambar 3.6, tahap *preprocessing* dibagi menjadi tiga yaitu pembuatan *bounding box*, pengubahan ukuran gambar, dan pembuatan *mini mask*.

3.2.1 Pembuatan *Bounding Box*

Pada tahap ini, sebuah *bounding box* akan dibentuk pada gambar-gambar yang telah memiliki objek anotasi. *Bounding box* yang dibentuk mengikuti struktur panjang dan lebar segmentasi (*mask*) yang telah dibuat pada tahap pembuatan dataset. *Bounding box* dibentuk mengikuti koordinat panjang dan lebar dari *mask* agar jika gambar atau objek tersebut mengalami transformasi seperti pemutaran gambar, pengubahan ukuran gambar, dan pemotongan gambar, *bounding box* dapat mengikuti koordinat transformasi dari *mask* pada objek yang telah melakukan transformasi. Proses ini tentunya lebih efektif daripada menghitung koordinat *bounding box* setiap kali gambar melakukan transformasi.

3.2.2 Pengubahan Ukuran Gambar

Pada tahap ini, setelah objek pada gambar diberikan *bounding box* gambar akan diubah ukurannya menjadi ukuran yang sama (*uniform*) dimana pada penelitian ini, ditetapkan ukuran gambar adalah 1024 x 1024 karena proses *transfer learning* dari serta proses ekstraksi fitur mendukung ukuran tersebut.

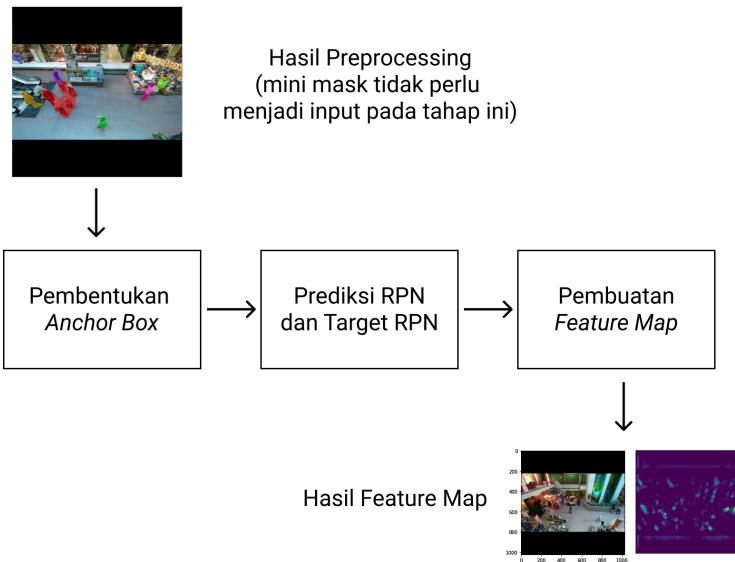
Jika gambar awal tidak memiliki ukuran piksel 1024 pada kedua sisinya, maka sisi dengan nilai piksel terbesar akan diperpanjang hingga bernilai 1024, sedangkan sisi yang lain diperbesar mengikuti ratio awal. Jika salah satu sisi gambar telah bernilai 1024, maka sisi yang lain akan ditambahkan piksel dengan nilai piksel 0 dimana gambar awal akhirnya dipertahankan pada posisi tengah. Piksel bernilai 0 akan memberikan warna hitam pada gambar, sehingga pada proses ekstraksi fitur maupun deteksi citra, piksel bernilai 0 tidak memberikan dampak apapun dan hanya memakan sedikit daya komputasi.

3.2.3 Pembentukan Mini Mask

Pada tahap ini, setelah semua gambar memiliki nilai piksel yang sama, hasil segmentasi atau yang disebut *mask* pada data awal akan diekstrak dan mengalami perubahan ukuran. Pada awalnya *mask* tiap objek memiliki ukuran 1024 x 1024. Jika pada gambar tersebut memiliki banyak objek, misalnya sebagai contoh terdapat 10 objek, maka akan terdapat 11 gambar berukuran 1024 x 1024 piksel yang terdiri dari 10 gambar *mask* dan 1 gambar asli. Tentunya hal ini tidak efektif karena *mask* tersebut akan memakan daya memori komputer saat melakukan proses *training*, karena itu ukuran *mask* akan diperkecil menjadi 56 x 56 piksel dan hanya mengambil piksel *mask*.

3.3 Ekstraksi Fitur

Proses ekstraksi fitur merupakan salah satu proses yang terpenting untuk memberikan data kepada algoritma deteksi objek, dengan fitur-fitur yang diambil pada gambar, algoritma dapat menentukan lokasi serta kelas objek. Pada penelitian ini, ekstraksi fitur menggunakan *backbone* Resnet 101 yang memanfaatkan penggunaan *anchor box*. Proses ekstraksi fitur yang perlu dilakukan pertama kali ialah membentuk *anchor box*, kemudian berdasarkan *anchor box* yang telah dibentuk, maka *region proposal network* dapat dibangkitkan, kemudian Resnet 101 akan digunakan untuk membuat sebuah *feature map* berdasarkan *region proposal network* yang telah dibuat. *Feature map* akan digunakan untuk mendeteksi lokasi, memberikan *mask*,

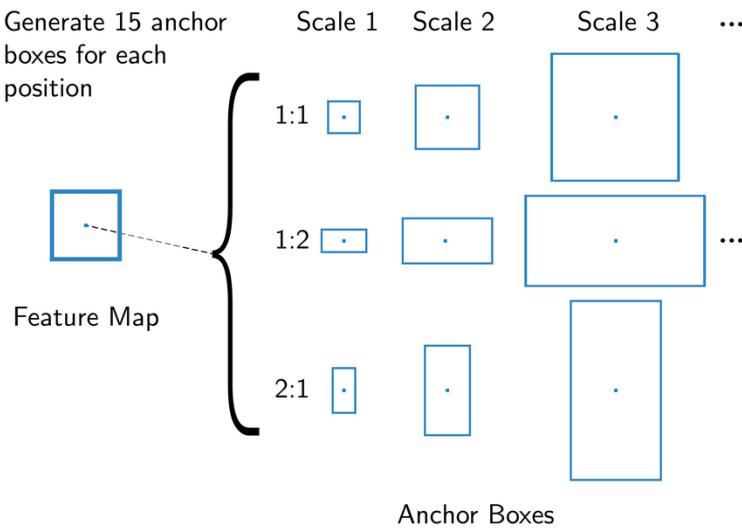


Gambar 3.7. Proses Ekstraksi Fitur

membangkitkan *bounding box* dan mengklasifikasi kelas objek yang dideteksi.

3.3.1 Pembuatan *Anchor Box*

Proses ekstraksi fitur yang pertama ialah pembuatan *anchor box*, *anchor box* digunakan untuk mengetahui lokasi *bounding box* dan menjadi target dalam proses *training* model dalam mendekripsi *regional proposal network* (*RPN*). Terdapat 15 *anchor box* yang dibentuk pada setiap iterasi. Dapat dilihat pada gambar 3.8 bahwa *anchor box* yang dibentuk memiliki lima skala ukuran dan tiga jenis persegi empat yang mempunyai ukuran piksel yang berbeda. 15 belas *anchor box* tersebut akan disusun bertumpuk menyerupai piramid dan akan digeser setiap tiga piksel hingga semua piksel pada gambar telah terlingkupi oleh *anchor box*. *Anchor box* yang memiliki *intersection over union* atau irisan lebih dari sama dengan 70% dengan *bounding box* yang telah dibentuk pada tahap *preprocessing* dataset akan disimpan. Sedangkan *anchor box* yang memiliki irisan kurang dari 70% dengan *bounding box* yang telah dibuat akan dibuang dan tidak digunakan lagi. *Anchor box* yang disimpan dinamakan target *region proposal network* (*RPN*).



Gambar 3.8. Bentuk *Anchor Box* [6]

3.3.2 Prediksi RPN

Pada tahap sebelumnya, beberapa *anchor box* yang mempunyai irisan dengan *bounding box* asli telah disimpan sebagai *region of interest* positif . Target *region proposal network* positif ini akan menjadi data untuk melatih model pada jaringan *backbone*. Pada penelitian ini *backbone* yang digunakan ialah ResNet 101, sehingga jaringan *Convolutional Neural Network* pada ResNet 101 akan dilatih untuk dapat mendekripsi fitur yang ada pada *regional proposal network*. Menggunakan beberapa iterasi maka objek tersebut dapat dideteksi, kemudian suatu *bounding box* hasil prediksi juga dibentuk, tentunya *bounding box* yang dibentuk terkadang dapat menimpa satu sama lain pada satu objek yang sama. Karena itu penerapan algoritma *non maximum suppression* diperlukan agar *bounding box* yang menimpa satu sama lain dapat dihilangkan dan menjadi satu *bounding box* yang optimal.

3.3.3 Pembuatan *Feature Map*

Selain hasil berupa prediksi *region proposal network*, *backbone* ResNet 101 juga menghasilkan sebuah *feature map* yang merupakan hasil dari proses konvolusi dan pooling pada jaringan residual yang digunakan. *Feature map*

ini akan digunakan sebagai data bagi *fully connected layer* dan *layer konvolusi* untuk memprediksi kelas, *bounding box*, dan *mask*. Contoh hasil dari *feature map* dapat dilihat pada gambar 3.10.



Gambar 3.9. Hasil *non max supression* (kanan) [7]



Gambar 3.10. Contoh *Feature map* pada *layer* berbeda [5]

3.4 Proses Training

Proses pembelajaran mesin sudah dilakukan pada tahap ekstraksi fitur, namun dalam mengklasifikasi serta memberikan *bounding box* yang tepat dari berbagai *region proposal network* yang dibentuk dari proses ekstraksi fitur, beberapa *layer neural network* masih dibutuhkan dan perlu dilatih. Hasil keluaran dari *layer convolutional neural network* dan *fully connected layer* inilah yang akan memberikan keluaran akhir dalam bentuk *bounding box*, *mask*, dan kelas objek yang dideteksi. Pada tahap ini sebuah *fully connected layer*

Tabel 3.1. Konfigurasi pada proses *training*

Jenis Konfigurasi	Keterangan
<i>batch</i>	1
<i>learning rates</i>	dinamis dari 0.001 hingga 0.01
<i>epoch</i>	150

Tabel 3.2. *learning rate* dinamis

Epoch	Learning Rate
1 - 20	0.01
21 - 60	0.005
61 - 150	0.001

perlu dibentuk dalam mendeteksi kelas dan *bounding box*. Selain itu, sebuah *convolutional neural network* juga dibentuk untuk menghasilkan *mask* yang sesuai dengan objek.

Pada tahap ini, beberapa *hyperparameter* juga digunakan untuk mengatur jalannya proses *training*. *Hyperparameter* yang daitur pada penelitian ini antara lain *learning rate*, *epoch* dan *batch*. Pada penelitian ini pengaplikasian *learning rate* dinamis juga akan diperkenalkan. *learning rate* dinamis digunakan dengan tujuan untuk mengurangi *overfitting* dan mempercepat proses pembelajaran. Pada *epoch* 1 hingga 20, *learning rate* yang digunakan adalah 0.01, pada *epoch* 21 hingga 60, *learning rate* yang digunakan adalah 0.005, sedangkan pada *epoch* 61 hingga 150, *learning rate* yang digunakan adalah 0.001 yang merupakan nilai terkecil yang digunakan pada penelitian ini, detail *learning rate* yang digunakan dapat dilihat pada tabel 3.2. Sedangkan detail mengenai *hyperparameter* dapat dilihat pada tabel 3.1.

3.5 Proses Pengujian

Proses pengujian model dalam penelitian ini dibagi menjadi 2 bagian, bagian yang pertama dilakukan pada saat model selesai melakukan proses *training* namun masih memiliki iterasi *epoch* yang belum selesai. Proses ini bernama validasi, dimana akurasi dari uji coba mode pada *val set* akan dilakukan. Proses ini sangat vital karena dengan validasi kita dapat

mengetahui apakah model kita mengalami *overfitting* dimana performa model dalam mendeteksi objek di gambar baru dirasa kurang maupun *underfitting* dimana model tidak menunjukkan kemajuan dalam mendeteksi objek pada gambar. Salah satu ciri yang paling mudah yang menandakan kemungkinan terjadinya *overfitting* adalah ketika besaran *training loss* suatu model menjadi semakin kecil, namun besaran *validation loss*-nya malah semakin besar di setiap iterasi. Sedangkan *underfitting* terjadi ketika baik *training loss* maupun *validation loss* memiliki nilai yang terlalu besar. Selain itu pada proses validasi ini, data yang digunakan adalah data yang sama sekali baru dan tidak digunakan selama proses *training* guna menghindari kemungkinan terjadinya bias yang biasa terjadi apabila suatu model diuji pada data yang sama yang digunakan pada saat proses *training*. Berdasarkan data pada proses validasi, algoritma *optimizer* akan memutuskan untuk merubah *weight* dalam pada *neural network* sehingga semakin mendekati titik akurasi tertingginya.

Bagian selanjutnya dari proses pengujian adalah melakukan *test*. Sama seperti pada waktu proses validasi, dataset yang digunakan juga sama sekali berbeda dengan data yang digunakan pada waktu *training* maupun pada waktu *validasi*. Dari proses ini, dapat diambil kesimpulan apakah suatu model tersebut dapat dilakukan perbaikan lagi dengan cara *re-training* dan merubah beberapa parameter maupun konfigurasi yang sudah diatur pada saat proses *training*, ataukah model tersebut dirasa sudah cukup baik dan akan melanjutkan ke proses berikutnya. Adapun beberapa metriks yang dapat diketahui dari proses *training* yaitu metriks *train accuracy*, *loss*, *val-accuracy* dan *val-loss*. *loss function* yang diberikan merupakan jumlah segala *loss* yang terjadi pada tiap tahapan proses pada algoritma Mask R-CNN yang ditetapkan. Persamaan total *loss* dapat dilihat pada persamaan 3.1. Sedangkan persamaan total *mask loss* dapat dilihat pada persamaan 3.6. Untuk persamaan mengenai *loss* yang terjadi pada kelas serta *bounding box*, dapat dilihat pada persamaan 3.2.

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} \quad (3.1)$$

$$\mathcal{L}_{\text{fr}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} \quad (3.2)$$

$$\mathcal{L}_{\text{fr}}(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*) \quad (3.3)$$

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i) \quad (3.4)$$

$$\mathcal{L}_{\text{box}} = \frac{\lambda}{N_{\text{box}}} \sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*) \quad (3.5)$$

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \leq i, j \leq m} [y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k)] \quad (3.6)$$

3.6 Analisa Performa

Setelah melakukan proses pengujian, langkah berikutnya adalah melakuk-an analisa perfforma pada model yang sudah dibuat. Hal ini untuk mengetahui bagaimana kira - kira performa model pada saat sudah diimplementasi. Untuk analisa performa, model akan diuji pada dataset *test*, dimana model tersebut akan mendeteksi gambar-gambar yang belum pernah dilihat sebelumnya. Untuk mengetahui performa model, analisanya akan menggunakan beberapa metode seperti *confusion matrix* untuk mendapatkan pengelompokan berda-sarkan data menjadi *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) dalam proses deteksi. Selain itu, penelitian ini juga akan menggunakan rumus - rumus seperti *Recall*, *Precision* dan *Accuracy* dalam mengetahui performa model.

Halaman ini sengaja dikosongkan

BAB 4

PENGUJIAN

Pada penelitian ini dipaparkan hasil dari proses yang dipaparkan pada bab 3 serta analisa yang dilakukan sesuai dengan desain sistem yang sudah dirancang pada bab sebelumnya. Pembahasan hasil penelitian dilakukan dengan beberapa bagian seperti berikut :

- Hasil dari Pembuatan Dataset
- Hasil *Preprocessing* Data.
- Hasil Ekstraksi Fitur.
- Hasil Pembelajaran Mesin Mask R-CNN.
- Hasil Model Akhir Mask R-CNN.

Bagian pertama ialah membahas mengenai hasil dari pembuatan dataset, pada bagian ini akan dijelaskan mengenai jumlah anotasi objek yang dibentuk. Kemudian pada bagian *preprocessing*, akan dijelaskan hasil mengenai pembuatan *bounding box*, dan *mini mask*. Kemudian pada bagian ekstraksi fitur dan hasil pembelajaran mesin Mask R-CNN akan membahas mengenai *pipeline* dari algoritma *Mask R-CNN* secara keseluruhan. Kemudian terakhir pada bagian hasil model akhir Mask R-CNN akan dijelaskan mengenai

4.1 Hasil Pembuatan Dataset

Pengambilan data video dilakukan di Pasar Atom pada bulan Januari 2022 dimana total video yang dikumpulkan adalah 24 menit dan terdiri dari 6 lokasi. Pengambilan data dilakukan pada jam 1 hingga 2 siang. Setelah pengambilan video dilakukan, 1 frame setiap 90 frame pada video diekstrak sehingga menghasilkan 730 gambar. Gambar-gambar ini kemudian dibagi menjadi tiga set (*train set*, *test set* dan *val set*) dengan perbandingan 0.6 : 0.2 : 0.2. *Train set* memiliki total gambar 436 gambar, *test set* memiliki total 146 gambar, dan sama seperti *test set*, *val set* juga memiliki 146 gambar. Kemudian

pada setiap gambar tersebut, objek manusia dibedakan menjadi tiga kelompok atau kelas. Seperti yang sudah dijelaskan pada bab sebelumnya, kelas tersebut ialah

- Kelas *Person*, dimana objek manusia tersebut berada sendirian pada radius satu meter dapat dilihat pada gambar 4.1.
- Kelas *Group*, dimana terdapat dua hingga tiga orang yang berada pada radius satu meter dapat dilihat pada gambar 4.2.
- Kelas *Crowd*, dimana terdapat empat orang atau lebih yang berada berdekatan pada jarak radius satu meter dapat dilihat pada gambar 4.3.



Gambar 4.1. Contoh Kelas *Person*



Gambar 4.2. Contoh Kelas *Group*

Total objek yang dianotasi pada seluruh 730 gambar adalah 4.980 objek yang terdiri dari 3.382 objek kelas *person*, 1.350 kelas *group*, dan 248 kelas *crowd*. Sedangkan jika dibagi pada tiap set, *Train set* memiliki 2.054 objek kelas *person*, 732 objek kelas *group*, dan 159 objek kelas *crowd*. *Test set* memiliki 687 objek kelas *person*, 321 objek kelas *group*, dan 40 objek kelas *crowd*.



Gambar 4.3. Contoh Kelas *Crowd*

Sedangkan pada *val set* memiliki 641 objek kelas *person*, 297 objek kelas *group*, dan 49 objek kelas *crowd*. Dapat dikatakan bahwa dataset ini merupakan dataset yang tidak seimbang (*imbalance dataset*) namun dalam kondisi Covid-19, objek kerumunan memang lebih jarang ditemukan dibandingkan objek *person* atau orang yang menjaga jarak satu sama lain. Tentunya hal ini juga dapat digunakan untuk melihat perilaku para pengunjung, penjual, maupun petugas saat kondisi Covid-19. Dataset ini juga dapat membantu penelitian lain dalam mengembangkan suatu program deteksi kerumunan terutama dalam pusat perbelanjaan.

Tabel 4.1. Detail Jumlah Objek Anotasi

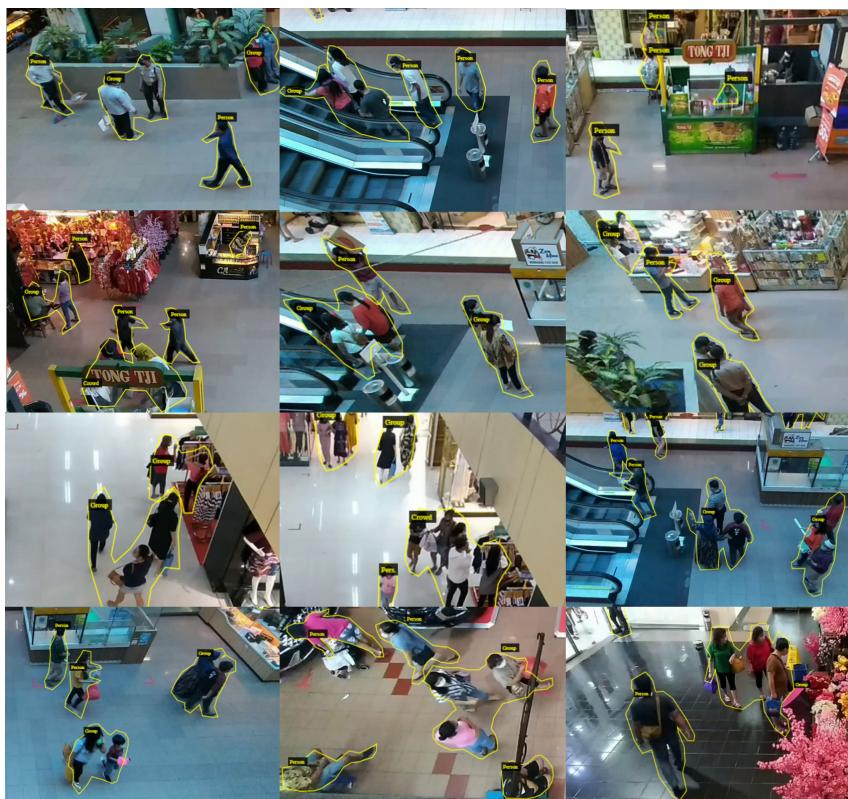
Set	Kelas <i>Person</i>	Kelas <i>Group</i>	Kelas <i>Crowd</i>
Train Set	2.054	732	159
Test Set	687	321	40
Val Set	641	297	49

4.2 Hasil *Preprocessing* Data

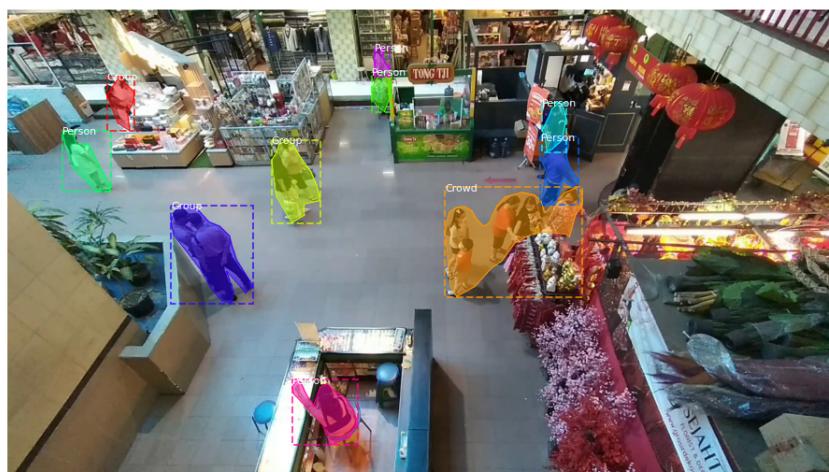
Pada bagian ini, hasil mengenai *preprocessing data pipeline* pada dataset yang telah dibentuk akan dijelaskan secara lebih mendalam.

4.2.1 Hasil Pembuatan *Bounding Box*

Seperti yang telah dijelaskan sebelumnya pada bab metodologi, pembuatan *bounding box* dilakukan dengan mengikuti ukuran dari *mask*. Jumlah *bounding box* pada tahap ini mempunyai nilai yang sama dengan jumlah objek pada dataset yaitu 4.970 objek.



Gambar 4.4. Contoh Beberapa Hasil Dataset

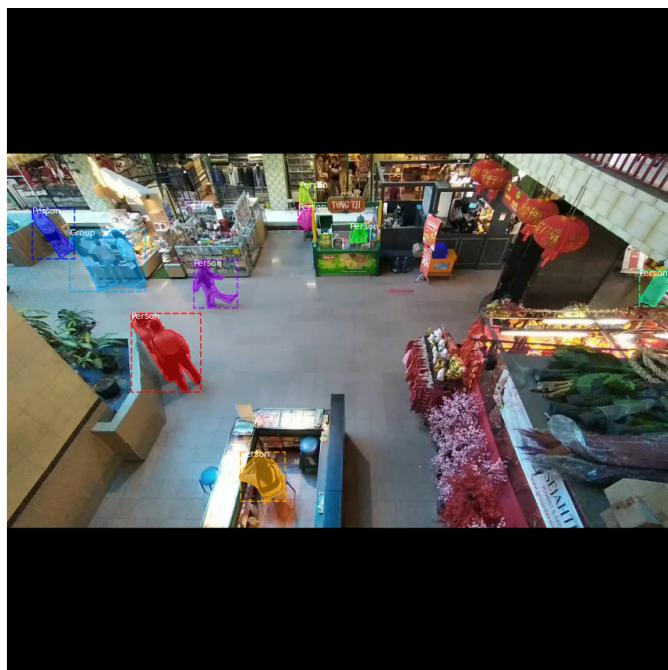


Gambar 4.5. Contoh Hasil Pembuatan *Bounding Box*

4.2.2 Hasil Pengubahan Ukuran Gambar

Setelah koordinat *bounding box* didapatkan dan *bounding box* dapat dibentuk, maka ukuran gambar akan disamakan sesuai dengan input pada

model dimana pada penelitian ini ialah 1024 x 1024 piksel. Ukuran gambar yang awalnya berukuran 1920 x 1080 dirubah menjadi 1024 x 1024 dengan cara memperkecil ukuran gambar dan memberikan piksel hitam pada bagian atas dan bawah gambar. Hasil dari *preprocess* ini dapat dilihat pada gambar 4.6. Dimensi gambar pun berubah menjadi 1024 x 1024 x 3. Pada penelitian ini fitur warna tetap digunakan sebagai fitur, walaupun mungkin dalam kenyataannya, dimensi warna ini dapat dirubah menjadi satu jenis saja.



Gambar 4.6. Contoh Hasil *Resizing* Image

4.2.3 Hasil Pembuatan *Mini Mask*

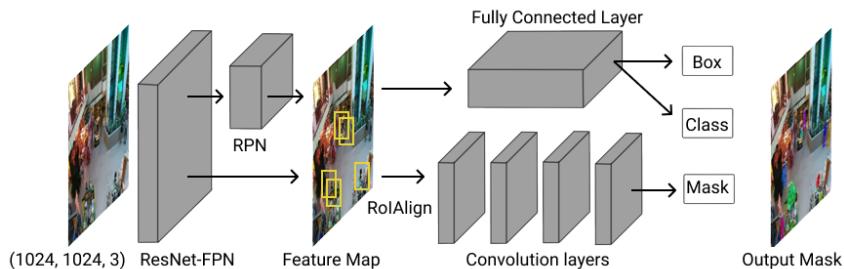
Setelah gambar-gambar dataset mempunyai ukuran yang sama dan sesuai dengan input dari model, maka hal selanjutnya yang perlu dilakukan ialah pembentukan *mini mask*. Sebelumnya, *mask* yang terbentuk pada dataset juga mempunyai ukuran yang sama pada gambar, tentunya hal ini tidak efektif karena *mask* tersebut dibedakan sesuai objeknya dan mempunyai banyak nilai null. Karena itu, bentuk *mask* diubah menjadi ukuran 56 x 56 piksel.



Gambar 4.7. Contoh Hasil Pembuatan *Mini Mask*

4.3 Hasil *Feature Extraction*

Pada bagian ini, hasil mengenai *feature extraction pipeline* pada dataset yang telah di-*preprocess* akan dijelaskan secara lebih mendalam. Proses *feature extraction* yang dilakukan pada tahap ini sudah menggunakan algoritma Mask-RCNN. Namun hasil yang dikemukakan pada bagian ini adalah hasil *feature map*. Sedangkan hasil mengenai proses Mask R-CNN yang memberikan klasifikasi kelas, *bounding box*, dan membangkitkan *mask*.



Gambar 4.8. Arsitektur Mask R-CNN yang diaplikasikan

Sesuai dengan proses algoritma Mask R-CNN yang diilustrasikan pada gambar 4.8. Proses Mask R-CNN dimulai dengan ekstraksi fitur pada jaringan *backbone* dimana pada penelitian ini *backbone* yang digunakan ialah ResNet 101. Kemudian hasil dari ResNet akan menghasilkan prediksi RPN *Regional Proposal Network* serta *feature map* yang merupakan hasil dari konvolusi dan

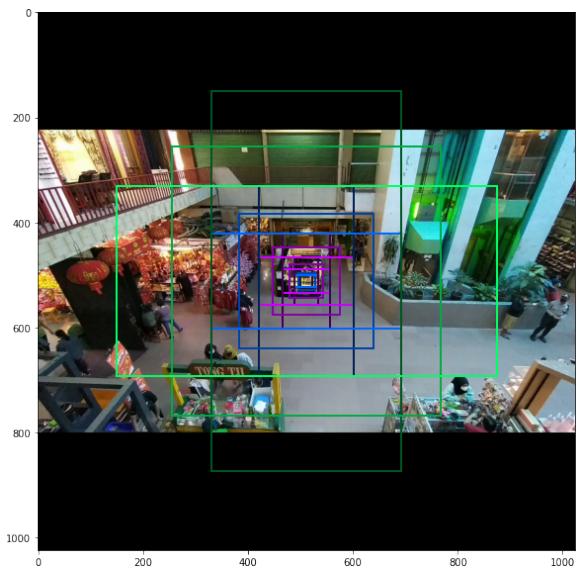
pooling pada jaringan ResNet, kemudian pada *feature map* tersebut akan diproses oleh *fully connected layer* untuk melakukan prediksi mengenai *bounding box* pada objek serta kelas daripada objek tersebut. Selain itu *convolution layer* akan menggunakan fitur-fitur pada *feature map* untuk membentuk *mask* dari objek. Hasil *training* dari pemrosesan oleh *Fully Connected Layer* dan *Convolution Layer* akan dijelaskan pada subbab berikutnya.

4.3.1 Hasil *Pembuatan Anchor Box*

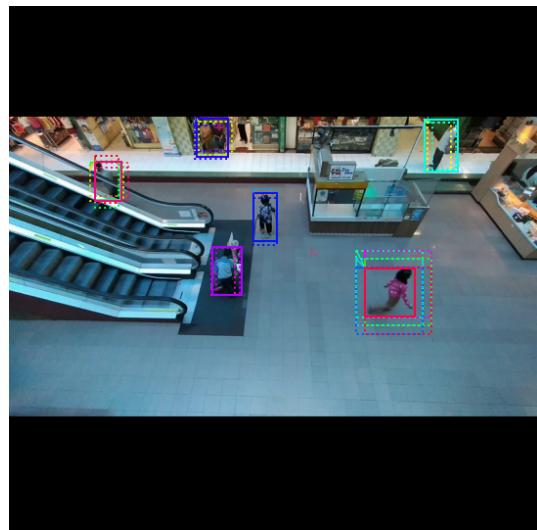
Bagian pertama pada proses *feature extraction* adalah membuat *anchor box* yang akan digunakan untuk target RPN. Untuk penentuan *anchor box* yang akan menjadi target RPN, maka digunakan metriks IoU *Intersect over Union*, jika *bounding box ground* atau *bounding box* asli memiliki IoU lebih dari sama dengan 0.7 dengan *anchor box*, maka *anchor box* tersebut digunakan sebagai target RPN. Hasil dari pembuatan *anchor box* ditunjukkan pada gambar 4.9. Sedangkan contoh *anchor box* yang digunakan sebagai target RPN dapat dilihat pada gambar 4.10. Untuk menunjukkan adanya perbedaan mengenai *anchor box* yang digunakan sebagai target dan dengan yang tidak maka hasil *anchor box* yang tidak digunakan pada proses melatih jaringan *backbone* dalam memprediksi RPN juga ditampilkan dan dapat dilihat pada 4.11.

4.3.2 Hasil Prediksi RPN dan Pembuatan *Feature Map*

Setelah jaringan *residual network* dilatih menggunakan target RPN, maka prediksi RPN pun dapat dilakukan oleh ResNet. Prediksi RPN dilakukan dengan menggunakan *train-set* dan *val-set* sehingga performa model dapat diketahui secara lebih dalam terutama dalam mengidentifikasi adanya *overfitting* maupun *underfitting*. RPN ini juga akan turut serta membantu dalam membentuk *feature map* karena jaringan ResNet ingin memperlihatkan fitur-fitur pada *regional proposal network* yang telah diprediksi dengan konvolusi dan *pooling*. Pada tahap ini, RPN juga mendeteksi kelas pada objek di dalam *bounding box* namun deteksi kelas yang dimaksud ialah deteksi apakah objek tersebut termasuk *foreground* atau *background*. Dalam memprediksi



Gambar 4.9. Hasil *Anchor Box* yang Dihasilkan



Gambar 4.10. Contoh Hasil *Anchor Box* yang Menjadi Target RPN

kelas objek, *loss* yang terjadi pada rpn umumnya bernilai kecil karena hanya memprediksi dua nilai saja yaitu *foreground* atau *background*.

4.3.3 Hasil *Training*

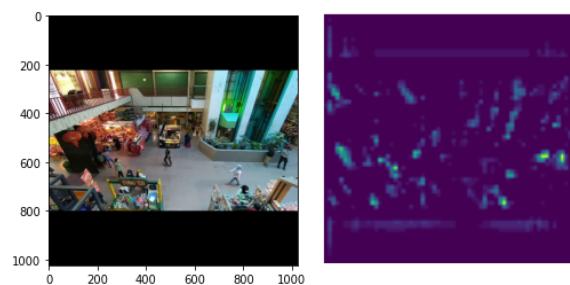
Pada bagian ini akan dijelaskan secara lebih mendalam mengenai hasil dari model yang telah dibentuk. Beberapa metriks *loss* yang dipaparkan adalah sebagai berikut.



Gambar 4.11. Contoh Hasil *Anchor Box* yang Tidak Digunakan



Gambar 4.12. Contoh Hasil Prediksi RPN



Gambar 4.13. Contoh Hasil *Feature Map*

- ***Loss***

Metriks *loss* merupakan metriks yang menunjukkan jumlah dari segala *loss* yang terjadi pada pemodelan, sedangkan *val loss* adalah jumlah *loss* yang terjadi saat model diuji coba pada dataset validasi.

- ***RPN Bounding Box Loss***

Metriks *loss* mengenai *bounding box* yang terjadi pada RPN menunjukkan adanya kesalahan prediksi pada jaringan *backbone* dalam memprediksi *bounding box* yang sesuai seperti target RPN. Kesalahan prediksi ditunjukkan dengan jauhnya jarak *anchor box* asli dengan *bounding box* yang diprediksi. Hasil *RPN Bounding Box Loss* dari proses *training* dapat dilihat pada gambar 4.16.

- ***RPN Class Loss***

Sedangkan Metriks *RPN Class loss* menunjukkan adanya kesalahan prediksi pada RPN untuk membedakan *foreground* dan *background* pada objek atau citra yang berada pada di dalam *bounding box* yang telah diprediksi, nilai *loss* ini umumnya bernilai lebih kecil daripada *loss* yang lain karena hanya kesalahan prediksi untuk membedakan dua kelas. Hasil *RPN Class Loss* dari proses *training* dapat dilihat pada gambar 4.15.

- ***Mask RCNN Bounding Box Loss***

Metriks *Mask RCNN Bounding Box Loss* mengenai *bounding box* yang terjadi pada *fully connected layer*. Hampir sama dengan metriks *RPN Bounding Box Loss*, kesalahan prediksi ditunjukkan dengan jauhnya jarak *anchor box* asli dengan hasil akhir *bounding box* yang diprediksi. Hasil *Mask RCNN Bounding Box Loss* dari proses *training* dapat dilihat pada gambar 4.18.

- ***Mask RCNN Class Loss***

Metriks *Mask RCNN Class Loss* memberikan penilaian mengenai kesalahan prediksi kelas, dimana pada penelitian ini, *Mask RCNN Class*

Loss akan bertambah jika Mask RCNN memberikan kesalahan klasifikasi antara tiga kelas objek. Hasil *Mask RCNN Class Loss* dari proses *training* dapat dilihat pada gambar 4.17.

- ***Mask RCNN Mask Loss***

Metriks *Mask RCNN Mask Loss* memberikan penilaian mengenai kesalahan pembentukan *mask* objek. Jika *mask* yang dibentuk oleh model mempunyai selisih yang jauh dengan *mask* yang dibentuk di awal, maka nilai *Mask RCNN Mask Loss* akan bertambah. Hasil *Mask RCNN Class Loss* dari proses *training* dapat dilihat pada gambar 4.19.

Dalam pemilihan model terbaik yang dapat diaplikasikan, metriks *val loss* merupakan yang terbaik karena metriks tersebut mencerminkan bagaimana model Mask R-CNN bekerja pada data yang belum pernah dilihat sebelumnya. Dari grafik-grafik *loss* yang dipaparkan, dapat diperhatikan bahwa pada epoch 0 hingga 60, nilai *loss* dan *val loss* menunjukkan penurunan yang signifikan, sedangkan pada epoch 60 hingga 150, nilai *loss* tetap menurun, namun nilai *val loss* tidak menunjukkan adanya perubahan yang signifikan. Hal ini dapat dikarenakan kecilnya *learning rate* yang diaplikasikan atau karena model telah mencapai nilai optimal. Nilai *val loss* mencapai titik terendah pada epoch 124 dengan nilai 0,772 dan nilai *loss* 0,5877. Karena nilai *val loss* pada model epoch 124 mempunyai nilai terkecil, maka model inilah yang akan diteliti lebih lanjut dan diuji coba pada test dataset. Namun sebelum itu, parameter *loss* yang lain perlu dilihat untuk memastikan apakah proses *training* berjalan dengan baik.

Selain memberikan informasi mengenai nilai *val loss* terendah, grafik *loss* pada gambar 4.14 menunjukkan adanya *overfitting*. *Overfitting* dapat diakibatkan oleh banyak faktor seperti arsitektur *neural network*, penggunaan *hyperparameter*, dan kurangnya dataset yang diambil. Dalam penelitian ini, kemungkinan besar hal ini terjadi karena adanya ketidak seimbangan antara kelas dataset dan kurang dataset kelas *crowd*. Karena itu *overfitting* terjadi,

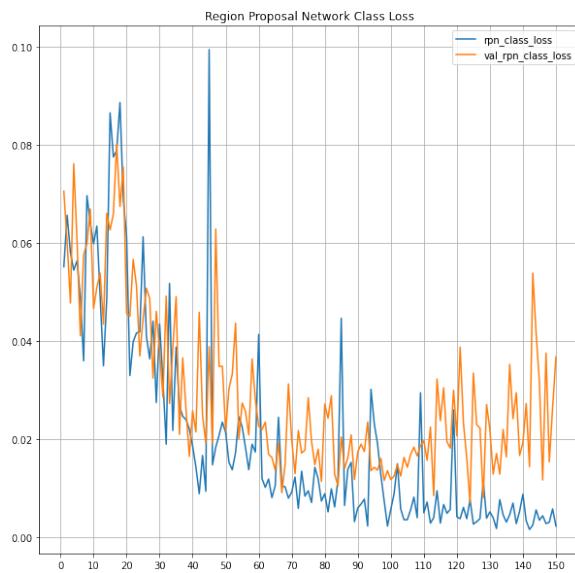


Gambar 4.14. Hasil *Loss*

namun kualitas dari model dapat dilihat terlebih dahulu sehingga dapat diketahui apakah data yang lebih banyak memang diperlukan, atau performa model dapat diterima walau mengalami *overfitting*. Selain itu komponen parameter *loss* apakah yang menyebabkan nilai *val loss* tidak mengalami penurunan juga dapat diketahui jika melihat grafik setiap parameter *loss*.

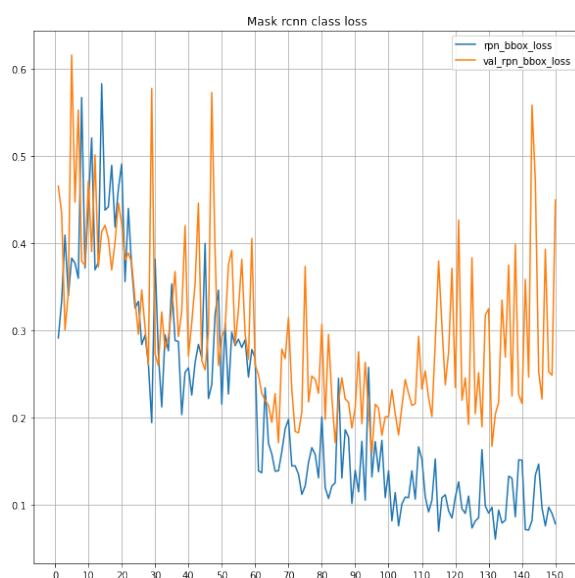
Selanjutnya pada metriks *RPN Class Loss* (gambar 4.15) dapat dilihat bahwa nilai *loss* yang terjadi di awal sudah kecil dibandingkan metriks *loss* yang lain. Nilai awal pada *RPN class loss* adalah 0,07 kemudian mempunyai nilai pada epoch 124 dengan nilai 0,0069. Sehingga dapat dikatakan bahwa model pada epoch 124 dapat membedakan objek yang menjadi fokus (*foreground*) dan objek yang menjadi latar belakang atau *background*. Proses pembelajaran *ResNet* dalam membedakan *foreground* maupun *background* dapat terbilang cukup baik karena memiliki nilai *loss* yang menurun pada epoch 0 hingga 100, namun nilai *loss* mulai berangsur naik saat epoch lebih dari 100. Di saat inilah *overfitting* mengenai prediksi *foreground* dan *background* terjadi.

Kemudian pada metriks *RPN Bounding Box Loss* (gambar 4.16) dapat dilihat bahwa nilai awal pada metriks *loss* adalah 0,45 dan mencapai nilai



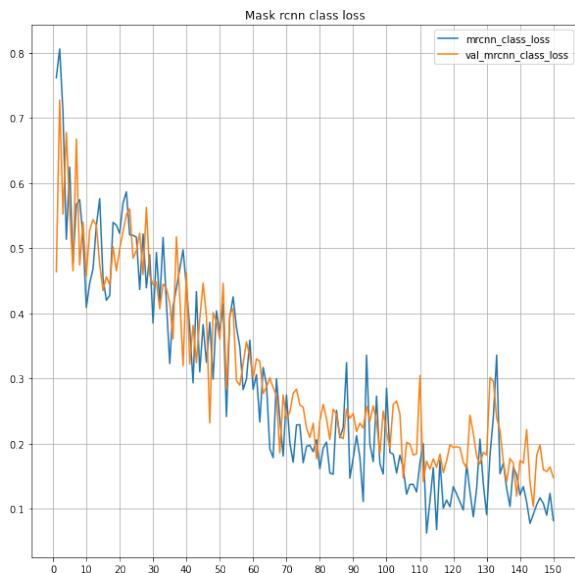
Gambar 4.15. Hasil *Class Loss* RPN

minimal 0,1591 pada epoch 95. Sedangkan pada epoch 124 yang merupakan model dengan total val loss paling rendah memiliki *val rpn bounding box loss* adalah 0,1921. Walaupun bukan nilai terendah, namun selisih nilai *val rpn bounding box loss* epoch 124 mempunyai selisih yang cukup kecil dengan epoch 95.



Gambar 4.16. Hasil *Bounding Box Loss* RPN

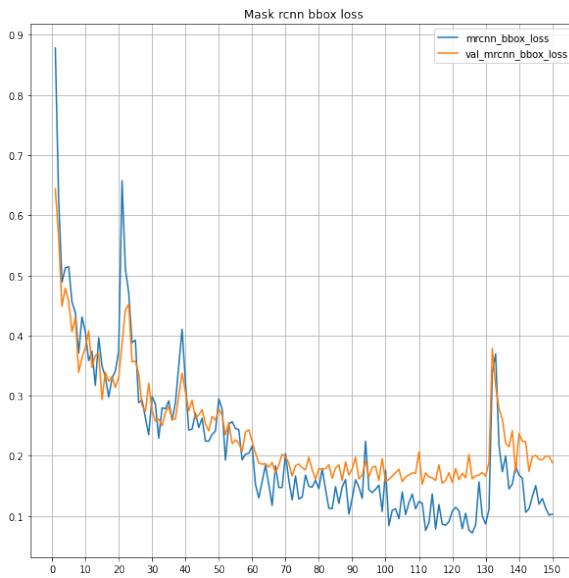
Selanjutnya pada metriks *class loss Mask RCNN* (gambar 4.17) dapat dilihat bahwa nilai awal pada metriks *val loss* berada di sekitar 0,7 dan mencapai nilai minimal 0,1032 pada epoch 144. Metriks *class loss Mask RCNN* merupakan parameter yang menunjukkan apakah Mask R-CNN dapat memprediksi kelas objek dengan benar atau tidak. Dalam melatih kemampuan Mask R-CNN dalam mendeteksi kelas objek, dapat dilihat bahwa nilai loss kelas memiliki trend yang menurun. Seperti metriks loss lainnya, pada epoch 0 hingga 60, penurunan nilai loss dapat dianggap drastis kemudian dengan learning rate yang lebih kecil, penurunan nilai loss juga semakin kecil.



Gambar 4.17. Hasil *Class Loss* Mask RCNN

Sedangkan pada metriks *Loss Bounding Box Mask R-CNN* (gambar 4.18) awalnya memiliki nilai loss 0,63 namun perlahan lahan turun mendekati nilai stabil pada 0,17 setelah 60 epoch. Parameter loss ini memiliki nilai terendah pada epoch 111 dengan nilai 0,1528. Tanda-tanda *overfitting* juga mulai terlihat setelah epoch 100, walaupun begitu nilai selisih antara loss *train* dan *validation* tidak terlalu jauh hanya sekitar 0,05.

Terakhir, metriks *Loss Mask* digambarkan pada (gambar 4.19). Parameter ini merupakan parameter yang tercepat mencapai nilai stabil. Namun para-



Gambar 4.18. Hasil *Bounding Box Loss* Mask RCNN

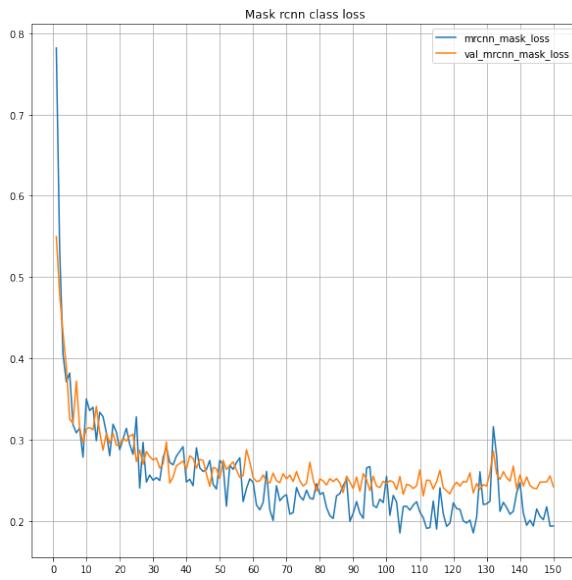
meter loss ini memiliki loss yang paling tinggi. Hal ini dikarenakan pembuatan mask diharapkan seakurat piksel segmentasi dataset, namun untuk melakukan segmentasi secara akurat pada dataset pun merupakan hal yang sangat sulit dilakukan. Nilai loss yang besar ini dapat memberikan masalah jika mask yang dibentuk sangat tidak sesuai dengan objek, karena itu pengujian dengan test dataset sangat diperlukan untuk melihat performa model secara keseluruhan.

Berdasarkan parameter loss yang telah dipaparkan, maka peneliti memutuskan untuk mengambil model pada epoch 124 karena memiliki nilai *validation loss* terendah dari proses pelatihan. Model pada epoch 124 memiliki detail *loss* sebagai berikut.

4.3.4 Hasil Pengujian pada Test Dataset

Pengujian pada model epoch ke 124, dilakukan menggunakan test dataset yang mempunyai 134 gambar. Namun sebelum model diuji secara keseluruhan, model akan diuji pada beberapa gambar. Hal ini dilakukan jika hasil pada pengujian pada beberapa gambar menunjukkan nilai yang buruk, maka model tidak perlu diuji dengan keseluruhan gambar.

Gambar 4.21 menunjukkan objek yang diprediksi dan *ground truth* dimana

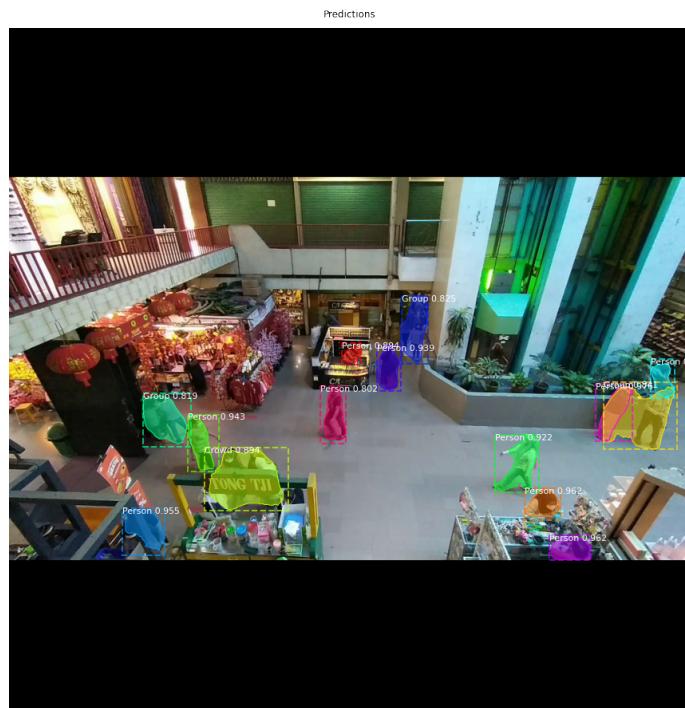


Gambar 4.19. Hasil Mask Loss Mask RCNN

Tabel 4.2. Metriks Loss pada Model 124

Parameter Model 124	Nilai
<i>loss</i>	0,5877
<i>rpn class loss</i>	0,0076
<i>rpn bounding box loss</i>	0,1102
<i>mrcnn class loss</i>	0,1683
<i>mrcnn bounding box loss</i>	0,1045
<i>mrcnn mask loss</i>	0,1971
<i>val loss</i>	0,772
<i>val rpn class loss</i>	0,0069
<i>val rpn bounding box loss</i>	0,1921
<i>val mrcnn class loss</i>	0,1620
<i>val mrcnn bounding box loss</i>	0,1634
<i>val mrcnn mask loss</i>	0,2476

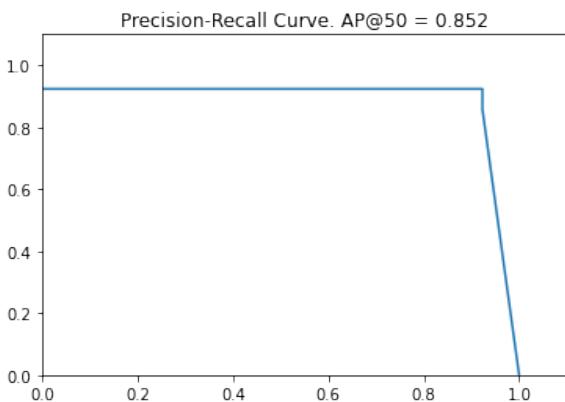
nilai prediksi dianggap benar jika IoU diatas 0,5. Dapat dilihat bahwa pada pengujian ini, 12 dari 14 objek dapat diprediksi secara benar dan dengan *confidence* di atas 70%. Nilai *confidence* dapat dilihat pada label sebelah kiri, sebagai contoh Person(0,96) berarti, model yang diaplikasikan mendeteksi suatu objek dengan nilai *confidence* 96% bahwa objek tersebut adalah objek kelas person. Sedangkan nilai yang berwarna biru adalah nilai dari *intersect over union* dimana jika nilai tersebut berada di atas nilai 0,5 maka dianggap



Gambar 4.20. Hasil Prediksi (1)

	Group	Group	Person	Crowd	Person	Group	Person	Person	Person						
Predictions															
Person (0.98)	0.000	0.216	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.96)	0.000	0.000	0.000	0.609 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.96)	0.000	0.000	0.000	0.000	0.840 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.95)	0.000	0.000	0.000	0.000	0.000	0.588 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.94)	0.000	0.000	0.000	0.000	0.000	0.000	0.717 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.94)	0.000	0.000	0.000	0.000	0.000	0.000	0.844 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.92)	0.000	0.000	0.811 match	0.000	0.000	0.000	0.000	0.558 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.89)	0.000	0.000	0.000	0.000	0.000	0.000	0.708 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Crowd (0.89)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.713 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Group (0.84)	0.599 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.814 match	0.000	0.000	0.000	0.000	0.000	0.000
Group (0.83)	0.120	0.000	0.000	0.000	0.000	0.000	0.841 match	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Person (0.80)															
Person (0.74)															

Gambar 4.21. Hasil Prediksi dan IoU (1)



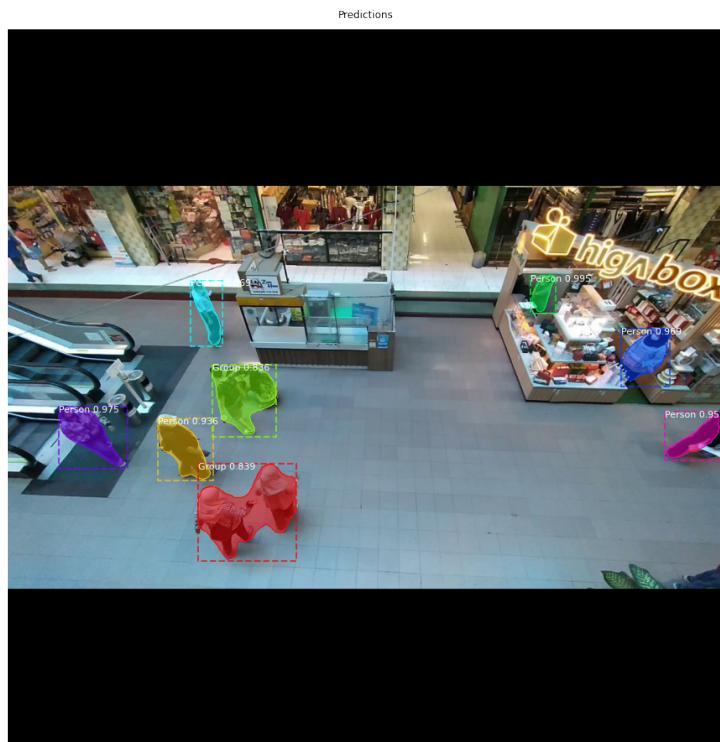
Gambar 4.22. Hasil Kurva Prediksi dan Recall (1)

prediksi benar.

Sedangkan grafik pada gambar 4.22 menunjukkan grafik hubungan antara presisi dan *recall* pada gambar yang diprediksi. Nilai pada x-axis menunjukkan nilai *recall* sedangkan nilai pada y-axis menunjukkan nilai presisi dan nilai *threshold*. Gambar 4.22 menunjukkan bahwa, saat diuji coba pada gambar tersebut, dengan threshold deteksi (*confidence*) 0,852, nilai presisinya ialah 0,852 dan nilai *recall* adalah 0,92. Model ini dapat dikatakan model yang bagus dalam mendeteksi gambar tersebut.

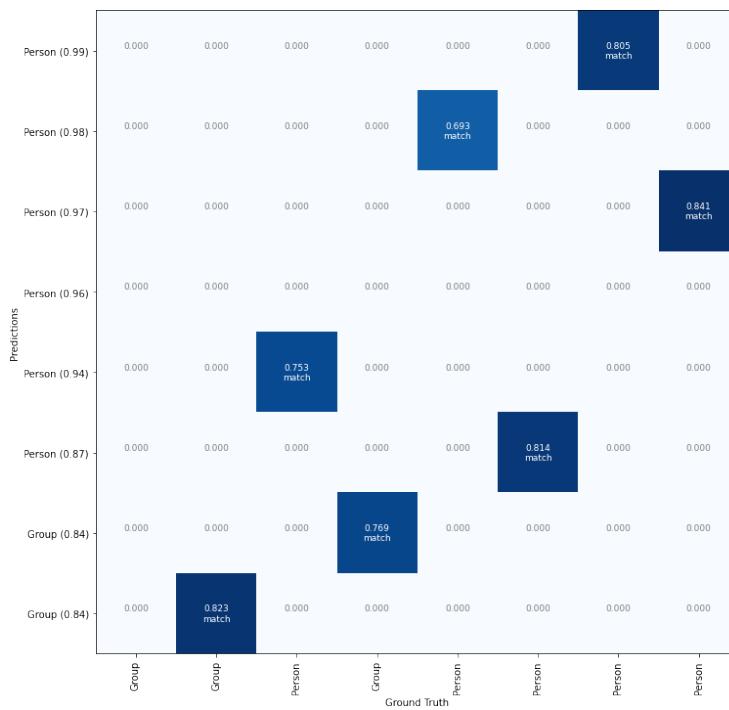
Kemudian saat diuji coba dengan gambar kedua, dapat dilihat bahwa model mendeteksi objek pada gambar dengan baik. dengan *threshold* 0,812, nilai presisi dan *recall* pada model pun cukup bagus. Deteksi yang dihasilkan pada uji coba gambar kedua ini pun dapat dihitung cukup memuaskan karena nilai *confidence* di atas nilai 0,8. Walaupun ada satu objek kelas *group* yang tidak terdeteksi, namun dari 8 objek yang ada, model dapat mendeteksi 7 objek dengan kelas yang benar dan mayoritas objek yang dideteksi mempunyai nilai *interference over union* di atas 0,7. terdapat satu objek kelas yang memiliki *interference over union* 0,693, namun saat dilihat pada gambar, objek tersebut memiliki *mask* yang cukup baik. Kemudian dapat dilihat juga bahwa model dapat memprediksi objek *person* yang hanya terlihat bagian kakinya, tentunya ini dapat dinilai sebagai model yang cukup cerdas dalam memperhatikan fitur

manusia. Hasil prediksi pada gambar kedua dapat dilihat pada gambar 4.23. Sedangkan untuk nilai *confidence* dan nilai IoU dapat dilihat pada gambar 4.24 dan kurva *precision-recall* dapat dilihat pada gambar 4.25. Dengan hasil ini dapat disimpulkan secara sementara bahwa model yang dibentuk memiliki kemampuan deteksi yang baik.

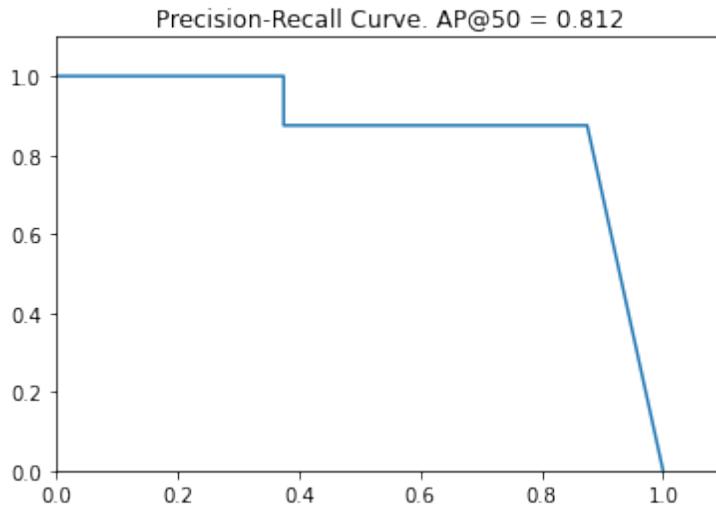


Gambar 4.23. Contoh Hasil Prediksi (2)

Setelah pengujian dengan beberapa gambar, maka pengujian dilakukan menggunakan seluruh test dataset. Hasil dari pengujian divisualisasikan dengan *confusion matrix* pada gambar 4.27. Perlu diketahui bahwa parameter deteksi adalah dimana model memiliki *confidence* lebih dari 70% pada sebuah deteksi objek dan memiliki *interference over union* lebih dari 50%. Pada *confusion matrix* kita dapat mengetahui bahwa akurasi secara menyeluruh dari model yang dihasilkan adalah 88,79%. Dari total 591 objek kelas *person*, model dapat mendekksi 531 objek kelas person yang benar. Sedangkan dari total 296 objek dengan kelas *group*, model dapat mendekksi 261 objek kelas



Gambar 4.24. Hasil Kurva Prediksi dan Recall (2)



Gambar 4.25. Contoh Hasil Prediksi dan IoU(2)

group secara benar. Terakhir, dari total 32 objek kelas *crowd*, terdapat 24 objek kelas *crowd* yang dideteksi dengan benar. Selain itu, tidak ada satupun objek dengan kelas *person* dideteksi sebagai objek kelas *crowd*

Dapat dilihat juga bahwa *recall* pada objek kelas *person* adalah 89,85%



Gambar 4.26. Beberapa Hasil Deteksi pada Dataset Test

dan memiliki presisi sebesar 95,85%. Selanjutnya pada objek kelas *group*, *recall* yang dihasilkan adalah 88,17% dengan presisi sebesar 79,33%. Terakhir pada objek kelas *crowd*, model memiliki performa deteksi yang paling rendah dibandingkan kelas yang lainnya yaitu 75% *recall* dan 66,67% pada presisi. Namun semua kelas *crowd* terdeteksi sebagai kelas *crowd* dan *group*. Tidak ada kelas *crowd* yang terdeteksi sebagai kelas *person* sehingga kelas ini dapat dianggap sebagai terdeteksi sebagai parameter yang berbahaya dalam masa pandemi Covid-19.

Predicted	Ground Truth			Precision
	Person	Group	Crowd	
Person	531	23	0	95.85 %
	60	261	8	79.33%
Crowd		12	24	66.67%
Recall	89.85%	88.17%	75%	Accuracy = 88.79%

Gambar 4.27. *Confusion Matrix* dari Hasil Prediksi Test Dataset

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Pada penelitian ini, sebuah dataset mengenai aktivitas jual beli pada suatu pusat perbelanjaan Di Surabaya berhasil dibentuk dengan total gambar 730 dan total objek 4.980. Dataset yang dibentuk memiliki total objek kelas *person* sebanyak 3.382 objek, objek kelas *group* sebanyak 1.350 objek, dan objek kelas *crowd* sebanyak 248 objek. Dataset tersebut dapat diakses pada . Selain dataset, penelitian ini juga memberikan kontribusi berupa alur kerja penerapan deteksi objek kerumunan pada pusat perbelanjaan, dan memberikan model yang dapat mendeteksi kerumunan dengan baik. Berdasarkan hasil penilitian ini penggunaan algoritma Mask R-CNN pada dataset ini terbukti dapat membedakan objek kelas *person* dimana objek kelas *person* adalah orang yang tidak berdekatan dengan orang lain dalam jarak kurang lebih satu meter dengan objek kelas *group* dan kelas *crowd* yang merupakan objek orang yang saling berdekatan dalam jarak satu meter. Model Mask R-CNN yang dibentuk dengan *backbone* ResNet 101 memiliki akurasi 88,79% dalam mendeteksi objek pada dataset dengan parameter *confidence* lebih dari sama dengan 70% dan *intersect over union* lebih dari sama dengan 50% dengan *mask* asli meskipun terdapat ketidak seimbangan data. Model yang dibentuk masih memiliki nilai *recall* dan presisi yang kecil dalam mendeteksi objek kelas *crowd* karena data kelas *crowd* hanyalah 5% dari total seluruh objek yang ada. Secara umum, model yang dibentuk dapat memprediksi adanya *group* dan *crowd*. Namun apakah model ini dapat digunakan dalam tempat pusat perbelanjaan lain masih perlu diteliti lebih lanjut.

5.2 Saran

Walaupun model ini secara umum sudah dapat mendeteksi apakah terdapat kerumunan atau tidak, namun nilai *recall* dan presisi saat mendeteksi objek *crowd* masih dinilai rendah. Hal ini disebabkan oleh tidak seimbangnya dataset yang telah dikumpulkan. Dikarenakan pengumpulan dataset dilakukan pada masa pandemi Covid-19, maka jumlah kerumunan yang terjadi pun tidak banyak. Dataset dan model dapat diperbaiki dengan menambah data terutama objek kelas *crowd*.

Daftar Pustaka

- [1] N. Artamonov and P. Yakimov, “Towards real-time traffic sign recognition via yolo on a mobile gpu,” *Journal of Physics: Conference Series*, vol. 1096, p. 012086, 09 2018.
- [2] H. Jung and B. Lodhi, “An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images,” *BMC Biomedical Engineering*, vol. 1, 10 2019.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [4] A. Dutta and A. Zisserman, “The VIA annotation software for images, audio and video,” in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM ’19. New York, NY, USA: ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3343031.3350535>
- [5] W. Abdulla, “Mask r-cnn for object detection and instance segmentation on keras and tensorflow,” https://github.com/matterport/Mask_RCNN, 2017.
- [6] M. Ferguson, R. ak, Y.-T. Lee, and K. Law, “Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning,” vol. 2, 09 2018.
- [7] V. Prisacariu and I. Reid, “fasthog-a real-time gpu implementation of hog,” 08 2009.
- [8] J. Guarner, “Three Emerging Coronaviruses in Two Decades: The Story of SARS, MERS, and Now COVID-19,” *American Journal of Clinical Pathology*, vol. 153, no. 4, pp. 420–421, 02 2020. [Online]. Available: <https://doi.org/10.1093/ajcp/aqaa029>
- [9] C. Drosten, S. Günther, W. Preiser, S. van der Werf, H. R. Brodt, S. Becker, H. Rabenau, M. Panning, L. Kolesnikova, R. A. Fouchier, A. Berger, A. M. Burguière, J. Cinatl, M. Eickmann, N. Escriou, K. Grywna, S. Kramme, J. C. Manuguerra, S. Müller, V. Rickerts, M. Stürmer, S. Vieth, H. D. Klenk, A. D. Osterhaus, H. Schmitz, and H. W. Doerr, “Identification of a novel coronavirus in patients with severe acute respiratory syndrome,” *N Engl J Med*, vol. 348, no. 20, pp. 1967–1976, May 2003.
- [10] T. G. Ksiazek, D. Erdman, C. S. Goldsmith, S. R. Zaki, T. Peret, S. Emery, S. Tong, C. Urbani, J. A. Comer, W. Lim, P. E. Rollin, S. F.

- Dowell, A. E. Ling, C. D. Humphrey, W. J. Shieh, J. Guarner, C. D. Paddock, P. Rota, B. Fields, J. DeRisi, J. Y. Yang, N. Cox, J. M. Hughes, J. W. LeDuc, W. J. Bellini, and L. J. Anderson, “A novel coronavirus associated with severe acute respiratory syndrome,” *N Engl J Med*, vol. 348, no. 20, pp. 1953–1966, May 2003.
- [11] A. M. Zaki, S. van Boheemen, T. M. Bestebroer, A. D. Osterhaus, and R. A. Fouchier, “Isolation of a novel coronavirus from a man with pneumonia in Saudi Arabia,” *N Engl J Med*, vol. 367, no. 19, pp. 1814–1820, Nov 2012.
- [12] H. Kimambo, J. H. Chin, M. Mnacho, P. Punatar, D. Msilanga, and A. C. Chagula, “Severe headache as the sole presenting symptom of COVID-19 pneumonia: A case report,” *Interdiscip Neurosurg*, vol. 22, p. 100882, Dec 2020.
- [13] “Coronavirus disease (covid-2019) situation reports,” <https://covid19.who.int/>, 2022, accessed: 2022-03-08.
- [14] Y. Yi, P. N. P. Lagniton, S. Ye, E. Li, and R. H. Xu, “COVID-19: what has been learned and to be learned about the novel coronavirus disease,” *Int J Biol Sci*, vol. 16, no. 10, pp. 1753–1766, 2020.
- [15] S. E. Eikenberry, M. Mancuso, E. Iboi, T. Phan, K. Eikenberry, Y. Kuang, E. Kostelich, and A. B. Gumel, “To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic,” *Infect Dis Model*, vol. 5, pp. 293–308, 2020.
- [16] M. Hyman, C. Mark, A. Imteaj, H. Ghiaie, S. Rezapour, A. M. Sadri, and M. H. Amini, “Data analytics to evaluate the impact of infectious disease on economy: Case study of COVID-19 pandemic,” *Patterns (N Y)*, vol. 2, no. 8, p. 100315, Aug 2021.
- [17] D. Truong, “Estimating the impact of covid-19 on air travel in the medium and long term using neural network and monte carlo simulation,” *Journal of Air Transport Management*, vol. 96, p. 102126, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0969699721001083>
- [18] “Covid-19: physical distancing,” <https://www.who.int/westernpacific/emergencies/covid-19/information/physical-distancing>, 2022, accessed: 2022-03-19.
- [19] R. Djalante, J. Lassa, D. Setiamarga, A. Sudjatma, M. Indrawan, B. Haryanto, C. Mahfud, M. S. Sinapoy, S. Djalante, I. Rafliana, L. A. Gunawan, G. A. K. Surtiari, and H. Warsilah, “Review and analysis of current responses to covid-19 in indonesia: Period of january to march 2020,” *Progress in Disaster Science*, vol. 6, p. 100091, 2020.

- [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590061720300284>
- [20] S. E. Galloway, P. Paul, D. R. MacCannell, M. A. Johansson, J. T. Brooks, A. MacNeil, R. B. Slayton, S. Tong, B. J. Silk, G. L. Armstrong, M. Biggerstaff, and V. G. Dugan, “Emergence of SARS-CoV-2 B.1.1.7 Lineage - United States, December 29, 2020-January 12, 2021,” *MMWR Morb Mortal Wkly Rep*, vol. 70, no. 3, pp. 95–99, Jan 2021.
 - [21] K. Wu, A. P. Werner, J. I. Moliva, M. Koch, A. Choi, G. B. E. Stewart-Jones, H. Bennett, S. Boyoglu-Barnum, W. Shi, B. S. Graham, A. Carfi, K. S. Corbett, R. A. Seder, and D. K. Edwards, “mRNA-1273 vaccine induces neutralizing antibodies against spike mutants from global SARS-CoV-2 variants,” *bioRxiv*, Jan 2021.
 - [22] N. G. Davies, S. Abbott, R. C. Barnard, C. I. Jarvis, A. J. Kucharski, J. D. Munday, C. A. B. Pearson, T. W. Russell, D. C. Tully, A. D. Washburne, T. Wenseleers, A. Gimma, W. Waites, K. L. M. Wong, K. van Zandvoort, J. D. Silverman, K. Diaz-Ordaz, R. Keogh, R. M. Eggo, S. Funk, M. Jit, K. E. Atkins, and W. J. Edmunds, “Estimated transmissibility and impact of SARS-CoV-2 lineage B.1.1.7 in England,” *Science*, vol. 372, no. 6538, 04 2021.
 - [23] H. Tegally, E. Wilkinson, M. Giovanetti, A. Iranzadeh, V. Fonseca, J. Giandhari, D. Doolabh, S. Pillay, E. J. San, N. Msomi, K. Mlisana, A. von Gottberg, S. Walaza, M. Allam, A. Ismail, T. Mohale, A. J. Glass, S. Engelbrecht, G. Van Zyl, W. Preiser, F. Petruccione, A. Sigal, D. Hardie, G. Marais, N. Y. Hsiao, S. Korsman, M. A. Davies, L. Tyers, I. Mudau, D. York, C. Maslo, D. Goedhals, S. Abrahams, O. Laguda-Akingba, A. Alisoltani-Dehkordi, A. Godzik, C. K. Wibmer, B. T. Sewell, J. Lourenço, L. C. J. Alcantara, S. L. Kosakovsky Pond, S. Weaver, D. Martin, R. J. Lessells, J. N. Bhiman, C. Williamson, and T. de Oliveira, “Detection of a SARS-CoV-2 variant of concern in South Africa,” *Nature*, vol. 592, no. 7854, pp. 438–443, 04 2021.
 - [24] M. Mwenda, N. Saasa, N. Sinyange, G. Busby, P. J. Chipimo, J. Hendry, O. Kapona, S. Yingst, J. Z. Hines, P. Minchella, E. Simulundu, K. Changuula, K. S. Nalubamba, H. Sawa, M. Kajihara, J. Yamagishi, M. Kapin’aa, N. Kapata, S. Fwoloshi, P. Zulu, L. B. Mulenga, S. Agolory, V. Mukonka, and D. J. Bridges, “Detection of B.1.351 SARS-CoV-2 Variant Strain - Zambia, December 2020,” *MMWR Morb Mortal Wkly Rep*, vol. 70, no. 8, pp. 280–282, Feb 2021.
 - [25] P. Wang, R. G. Casner, M. S. Nair, M. Wang, J. Yu, G. Cerutti, L. Liu, P. D. Kwong, Y. Huang, L. Shapiro, and D. D. Ho, “Increased Resistance of SARS-CoV-2 Variant P.1 to Antibody Neutralization,” *bioRxiv*, Apr 2021.

- [26] N. R. Faria, T. A. Mellan, C. Whittaker, I. M. Claro, D. D. S. Candido, S. Mishra, M. A. E. Crispim, F. C. Sales, I. Hawryluk, J. T. McCrone, R. J. G. Hulswit, L. A. M. Franco, M. S. Ramundo, J. G. de Jesus, P. S. Andrade, T. M. Coletti, G. M. Ferreira, C. A. M. Silva, E. R. Manuli, R. H. M. Pereira, P. S. Peixoto, M. U. Kraemer, N. Gaburo, C. D. C. Camilo, H. Hoeltgebaum, W. M. Souza, E. C. Rocha, L. M. de Souza, M. C. de Pinho, L. J. T. Araujo, F. S. V. Malta, A. B. de Lima, J. D. P. Silva, D. A. G. Zauli, A. C. de S Ferreira, R. P. Schnakenberg, D. J. Laydon, P. G. T. Walker, H. M. Schlüter, A. L. P. Dos Santos, M. S. Vidal, V. S. Del Caro, R. M. F. Filho, H. M. Dos Santos, R. S. Aguiar, J. L. P. Modena, B. Nelson, J. A. Hay, M. Monod, X. Misouridou, H. Coupland, R. Sonabend, M. Vollmer, A. Gandy, M. A. Suchard, T. A. Bowden, S. L. K. Pond, C. H. Wu, O. Ratmann, N. M. Ferguson, C. Dye, N. J. Loman, P. Lemey, A. Rambaut, N. A. Fraiji, M. D. P. S. S. Carvalho, O. G. Pybus, S. Flaxman, S. Bhatt, and E. C. Sabino, “Genomics and epidemiology of a novel SARS-CoV-2 lineage in Manaus, Brazil,” *medRxiv*, Mar 2021.
- [27] S. Shiehzadegan, N. Alaghemand, M. Fox, and V. Venketaraman, “Analysis of the Delta Variant B.1.617.2 COVID-19,” *Clin Pract*, vol. 11, no. 4, pp. 778–784, Oct 2021.
- [28] A. Vaughan, “Omicron emerges,” *New Sci*, vol. 252, no. 3363, p. 7, Dec 2021.
- [29] H. Gu, P. Krishnan, D. Y. M. Ng, L. D. J. Chang, G. Y. Z. Liu, S. S. M. Cheng, M. M. Y. Hui, M. C. Y. Fan, J. H. L. Wan, L. H. K. Lau, B. J. Cowling, M. Peiris, and L. L. M. Poon, “Probable Transmission of SARS-CoV-2 Omicron Variant in Quarantine Hotel, Hong Kong, China, November 2021,” *Emerg Infect Dis*, vol. 28, no. 2, pp. 460–462, 02 2022.
- [30] Z. D. Guo, Z. Y. Wang, S. F. Zhang, X. Li, L. Li, C. Li, Y. Cui, R. B. Fu, Y. Z. Dong, X. Y. Chi, M. Y. Zhang, K. Liu, C. Cao, B. Liu, K. Zhang, Y. W. Gao, B. Lu, and W. Chen, “Aerosol and Surface Distribution of Severe Acute Respiratory Syndrome Coronavirus 2 in Hospital Wards, Wuhan, China, 2020,” *Emerg Infect Dis*, vol. 26, no. 7, pp. 1583–1591, 07 2020.
- [31] A. M. Kotlyar, O. Grechukhina, A. Chen, S. Popkhadze, A. Grimshaw, O. Tal, H. S. Taylor, and R. Tal, “Vertical transmission of coronavirus disease 2019: a systematic review and meta-analysis,” *Am J Obstet Gynecol*, vol. 224, no. 1, pp. 35–53, 01 2021.
- [32] P. Gautret, M. Million, P. A. Jarrot, L. Camoin-Jau, P. Colson, F. Fenollar, M. Leone, B. La Scola, C. Devaux, J. Y. Gaubert, J. L. Mege, J. Vitte, C. Melenotte, J. M. Rolain, P. Parola, J. C. Lagier, P. Brouqui, and D. Raoult, “Natural history of COVID-19 and therapeutic options,” *Expert Rev Clin Immunol*, vol. 16, no. 12, pp. 1159–1184, 12 2020.

- [33] E. K. Stokes, L. D. Zambrano, K. N. Anderson, E. P. Marder, K. M. Raz, S. El Burai Felix, Y. Tie, and K. E. Fullerton, “Coronavirus Disease 2019 Case Surveillance - United States, January 22-May 30, 2020,” *MMWR Morb Mortal Wkly Rep*, vol. 69, no. 24, pp. 759–765, Jun 2020.
- [34] C. Gebhard, V. Regitz-Zagrosek, H. K. Neuhauser, R. Morgan, and S. L. Klein, “Impact of sex and gender on COVID-19 outcomes in Europe,” *Biol Sex Differ*, vol. 11, no. 1, p. 29, 05 2020.
- [35] S. Sze, D. Pan, C. R. Nevill, L. J. Gray, C. A. Martin, J. Nazareth, J. S. Minhas, P. Divall, K. Khunti, K. R. Abrams, L. B. Nellums, and M. Pareek, “Ethnicity and clinical outcomes in COVID-19: A systematic review and meta-analysis,” *EClinicalMedicine*, vol. 29, p. 100630, Dec 2020.
- [36] F. B. Ahmad, J. A. Cisewski, A. Miniño, and R. N. Anderson, “Provisional Mortality Data - United States, 2020,” *MMWR Morb Mortal Wkly Rep*, vol. 70, no. 14, pp. 519–522, 04 2021.
- [37] T. Chen, Y. Jin, J. Yang, and G. Cong, “Identifying emergence process of group panic buying behavior under the covid-19 pandemic,” *Journal of Retailing and Consumer Services*, vol. 67, p. 102970, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0969698922000637>
- [38] D. Headey, S. Goudet, L. Isabel, E. M. Maffioli, T. Z. Oo, and T. Russell, “Poverty and food insecurity during covid-19: Phone-survey evidence from rural and urban myanmar in 2020,” *Global Food Security*, p. 100626, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2211912422000177>
- [39] A. Campo-Arias, J. C. Pedrozo-Pupo, and C. C. Caballero-Domínguez, “Relation of perceived discrimination with depression, insomnia and post-traumatic stress in covid-19 survivors,” *Psychiatry Research*, vol. 307, p. 114337, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165178121006314>
- [40] E. A. Udeaja and K. O. Isah, “Stock markets’ reaction to covid-19: Analyses of countries with high incidence of cases/deaths in africa,” *Scientific African*, vol. 15, p. e01076, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S246822762100377X>
- [41] O. Khlystova, Y. Kalyuzhnova, and M. Belitski, “The impact of the covid-19 pandemic on the creative industries: A literature review and future research agenda,” *Journal of Business Research*, vol. 139, pp. 1192–1210, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0148296321007141>

- [42] W. Xu, A. Li, and L. Wei, “The impact of covid-19 on china’s capital market and major industry sectors,” *Procedia Computer Science*, vol. 199, pp. 87–94, 2022, the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 and 2021): Developing Global Digital Economy after COVID-19. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050922000114>
- [43] F. P. Polack, S. J. Thomas, and a. Kitchin, N. et, “Safety and efficacy of bnt162b2 mrna covid-19 vaccine,” *N Engl J Med*, vol. 383, no. 27, pp. 2603–2615, 12 2020.
- [44] V. A. Simonovich, L. D. Burgos Pratx, P. Scibona, M. V. Beruto, M. G. Vallone, C. Vázquez, N. Savoy, D. H. Giunta, L. G. Pérez, M. D. L. Sánchez, A. V. Gamarnik, D. S. Ojeda, D. M. Santoro, P. J. Camino, S. Antelo, K. Rainero, G. P. Vidiella, E. A. Miyazaki, W. Cornistein, O. A. Trabadelo, F. M. Ross, M. Spotti, G. Funtowicz, W. E. Scordo, M. H. Losso, I. Ferniot, P. E. Pardo, E. Rodriguez, P. Rucci, J. Pasquali, N. A. Fuentes, M. Esperatti, G. A. Speroni, E. C. Nannini, A. Matteaccio, H. G. Michelangelo, D. Follmann, H. C. Lane, W. H. Bellos, V. A. Simonovich, L. D. Burgos Pratx, P. Scibona, M. V. Beruto, M. G. Vallone, C. Vázquez, N. Savoy, D. H. Giunta, L. G. Pérez, M. Del L Sánchez, A. V. Gamarnik, D. S. Ojeda, D. M. Santoro, P. J. Camino, S. Antelo, K. Rainero, G. P. Vidiella, E. A. Miyazaki, W. Cornistein, O. A. Trabadelo, F. M. Ross, M. Spotti, G. Funtowicz, W. E. Scordo, M. H. Losso, I. Ferniot, P. E. Pardo, E. Rodriguez, P. Rucci, J. Pasquali, N. A. Fuentes, M. Esperatti, G. A. Speroni, E. C. Nannini, A. Matteaccio, H. G. Michelangelo, D. Follmann, H. C. Lane, W. H. Bellos, M. B. Bonella, L. Sujodoles Gazzero, F. Warley, M. A. Marco, E. A. Ko, A. M. Comisario, F. A. Serra Frías, G. Colucci, M. S. Osorno, M. S. Odstrcil Bobillo, E. F. Huaier Arriazu, T. Caccavo, R. C. Moreno Rodriguez, H. M. Recchioni, P. E. Guantay, J. B. Blanco, F. J. Vázquez, F. L. Cilenti, J. Rappi, M. Carlone, A. J. Scherling, S. Adra, B. Amarilla, V. L. Valiente, D. Montoya, L. Delgado, V. Carolina, D. N. Aramberri, F. Veloso, G. Montoya, N. E. Pons, C. V. Gamboa, G. P. Grottola, M. Lucas, H. J. Salamone, M. S. Venuti, M. J. Las Heras, V. C. Stanek, M. De Paz, N. Y. Brun, F. Seoane, A. Vernengo, D. Arrigo, M. I. Giménez, L. Molina, J. Larriba, M. Tambutti, M. Martínez von Scheidt, L. L. Bustamante, V. Sylvestre, M. B. Peralta, J. E. San Roman, M. M. Gonzalez Lopez Ledesma, L. N. Sanchez, G. S. Costa Navarro, H. M. Pallares, S. M. Villordo, D. E. Alvarez, J. J. Caramelo, J. Carradori, M. J. Yanovsky, P. Notrica, A. Acuña Elías, A. L. Tortoriello, C. A. Medina, E. L. Del M Romera, C. N. Mahler, A. Gamba, P. Galuppo, R. Baez, F. M. Rivero, M. A. Masciocchi, M. M. Cortiñas Chudoba, C. Ramirez, M. A. Malvicini, M. L. Pereyra, A. Rios, V. Marquevich, M. M. Lynch Garay, A. Espejo, N. Marcolini, A. Seresi, P. Brenzoni, P. Pratesi, M. Tisi Baña, F. Palizas, B. Lattanzio, M. Casanova, M. Gordóvil, E. Gándara,

- M. E. González, C. Moya, S. Díaz, A. Villoldo, M. Olmos, G. Lonegro, G. Santamaría, F. Tortosa, J. J. Toibaro, R. Fernandez Deud, C. Delgado, F. Masciotra, S. R. Caimi, V. Pachioli, A. M. Muñoz, P. Paulin, L. E. Epstein, S. Giannasi, J. D. Benso, M. A. Prieto, E. A. Herlein, L. A. Ducatenzeiler, J. A. Valverde, F. B. Libertella, L. G. Fernandez Otero, J. Méndez, M. Lahitte, M. Fenés, A. Zárate, V. Damonte, S. Villada, and G. Visintin, “A Randomized Trial of Convalescent Plasma in Covid-19 Severe Pneumonia,” *N Engl J Med*, vol. 384, no. 7, pp. 619–629, 02 2021.
- [45] L. Li, W. Zhang, Y. Hu, X. Tong, S. Zheng, J. Yang, Y. Kong, L. Ren, Q. Wei, H. Mei, C. Hu, C. Tao, R. Yang, J. Wang, Y. Yu, Y. Guo, X. Wu, Z. Xu, L. Zeng, N. Xiong, L. Chen, J. Wang, N. Man, Y. Liu, H. Xu, E. Deng, X. Zhang, C. Li, C. Wang, S. Su, L. Zhang, J. Wang, Y. Wu, and Z. Liu, “Effect of Convalescent Plasma Therapy on Time to Clinical Improvement in Patients With Severe and Life-threatening COVID-19: A Randomized Clinical Trial,” *JAMA*, vol. 324, no. 5, pp. 460–470, 08 2020.
- [46] D. M. Weinreich, S. Sivapalasingam, T. Norton, S. Ali, and e. a. Gao, H., “Regn-cov2, a neutralizing antibody cocktail, in outpatients with covid-19,” *N Engl J Med*, vol. 384, no. 3, pp. 238–251, 01 2021.
- [47] K. Hanthanan Arachchilage and M. Y. Hussaini, “Ranking non-pharmaceutical interventions against covid-19 global pandemic using global sensitivity analysis—effect on number of deaths,” *Chaos, Solitons, Fractals*, vol. 152, p. 111458, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077921008122>
- [48] Y. Ge, W.-B. Zhang, H. Liu, C. W. Ruktanonchai, M. Hu, X. Wu, Y. Song, N. W. Ruktanonchai, W. Yan, E. Cleary, L. Feng, Z. Li, W. Yang, M. Liu, A. J. Tatem, J.-F. Wang, and S. Lai, “Impacts of worldwide individual non-pharmaceutical interventions on covid-19 transmission across waves and space,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102649, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0303243421003561>
- [49] I. C. Education, “Machine learning,” 06 2020, diakses 23 Maret 2022. [Online]. Available: <https://www.ibm.com/cloud/learn/machine-learning>
- [50] S. Brown, “Machine learning, explained,” 04 2021, diakses 23 Maret 2022. [Online]. Available: <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained>
- [51] M. A. C. Team, “What is machine learning ?” diakses 23 Maret 2022. [Online]. Available: <https://azure.microsoft.com/en-us/overview/what-is-machine-learning-platform/>

- [52] M. Team, “What is deep learning ?” diakses 23 Maret 2022. [Online]. Available: <https://www.mathworks.com/discovery/deep-learning.html>
- [53] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013. [Online]. Available: <https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013>
- [54] I. Alexandros and A. Tefas, *Deep Learning for Robot Perception and Cognition*. Academic Press 2022, 2022.
- [55] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2015. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [56] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” 2017. [Online]. Available: <https://arxiv.org/abs/1703.06870>
- [57] banerjeer2611, “Understanding accuracy, recall, precision, f1 scores, and confusion matrices,” 01 2021, diakses 23 Maret 2022. [Online]. Available: <https://www.realpythonproject.com/understanding-accuracy-recall-precision-f1-scores-and-confusion-matrices/>

BIODATA PENULIS



Identitas Diri

Nama : Lukas Purba Wisesa
Tempat Lahir : Rembang
Tanggal Lahir : 24 November 1999
Alamat : Jalan Diponegoro 28

Riwayat Pendidikan

2020-sekarang : Program Master (S2), Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember

2017-2021 : Program Sarjana (S1), Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember

Tentang Penulis

Lukas Purba Wisesa, merupakan seseorang mahasiswa yang berasal dari Institut Teknologi Sepuluh Nopember departemen Teknik Elektro. Penulis merupakan lulusan S1 Institut Teknologi Sepuluh Nopember. Dalam masa kuliah, penulis tertarik pada bidang pengembangan *artificial intelligence* dan pembelajaran mesin. Selain itu, penulis juga aktif dalam melakukan analisis sebagai hobi dimana pekerjaan penulis dapat diakses di github.com/lukaspurbaw. Penulis juga aktif dalam mengikuti kompetisi pengembangan model data science di ajang NDSC 2020. Bagi pembaca yang memiliki kritik, saran, atau pertanyaan mengenai laporan thesis ini dapat menghubungi penulis melalui surel lukaspurbaw@gmail.com.

Halaman ini sengaja dikosongkan