

TESIS

**DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK Mencari
SUDUT DAN JARAK OPTIMAL PENDETEKSIAN DENGAN METODE
DEEP LEARNING YOLO V3**



Disusun oleh:

Nama : Mutiara Dwi Anggraini

NIM : 20.52.1307

Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2022

TESIS

**DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK MENCARI
SUDUT DAN JARAK OPTIMAL PENDETEKSIAN DENGAN METODE
DEEP LEARNING YOLO V3**

**DETECTION SOCIAL DISTANCE VIDEO DATA TO FIND OPTIMAL
DETECTION ANGLES AND DISTANCES USING THE YOLO V3 DEEP
DEEP LEARNING METHOD**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Mutlara Dwi Anggraini

NIM : 20.52.1307

Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2022

HALAMAN PENGESAHAN

**DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK Mencari
SUDUT DAN JARAK OPTIMAL PENDETEKSIAN DENGAN METODE DEEP
LEARNING YOLO V3**

**DETECTION SOCIAL DISTANCE VIDEO DATA TO FIND OPTIMAL
DETECTION ANGLES AND DISTANCES USING THE YOLO V3 DEEP DEEP
LEARNING METHOD**

Dipersiapkan dan Disusun oleh
Mutiara Dwi Anggraini
20.52.1307

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 2 Januari 2023

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Januari 2023
Rektor

Prof. Dr. M. Suvanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

**DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK Mencari
SUDUT DAN JARAK OPTIMAL PENDETEKSIAN DENGAN METODE DEEP
LEARNING YOLO V3**

**DETECTION SOCIAL DISTANCE VIDEO DATA TO FIND OPTIMAL
DETECTION ANGLES AND DISTANCES USING THE YOLO V3 DEEP DEEP
LEARNING METHOD**

Dipersiapkan dan Disusun oleh
Mutiara Dwi Angraini
20.52.1307

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Senin, 2 Januari 2023

Pembimbing Utama

Prof. Dr. Kusrini, M.Kom.
NIK. 190302106

Pembimbing Pendamping

Hanif Al Fatta, M.Kom.
NIK. 190302096

Anggota Tim Penguji

Andi Sunyoto, M.Kom
NIK. 190302052

Alva H M, S.T., M.Eng., Ph.D
NIK. 190302493

Prof Dr. Kusrini, M.Kom
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Januari 2023
Direktur Program Pascasarjana

Prof Dr. Kusrini, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Mutiara Dwi Anggraini

NIM : 20.52.1387

Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut: **DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK Mencari Sudut dan Jarak Optimal Pendeteksian Dengan Metode Deep Learning YOLO V3**

Dosen Pembimbing Utama : Prof. Dr. Kusrini, M.Kom.

Dosen Pembimbing Pendamping : Nama Dosen

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH dipaparkan untuk mendapatkan gelar akademik, baik di Universitas AMKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Penugasan tugas yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 2 Januari 2023

Yang Menyatakan,



Mutiara Dwi Anggraini

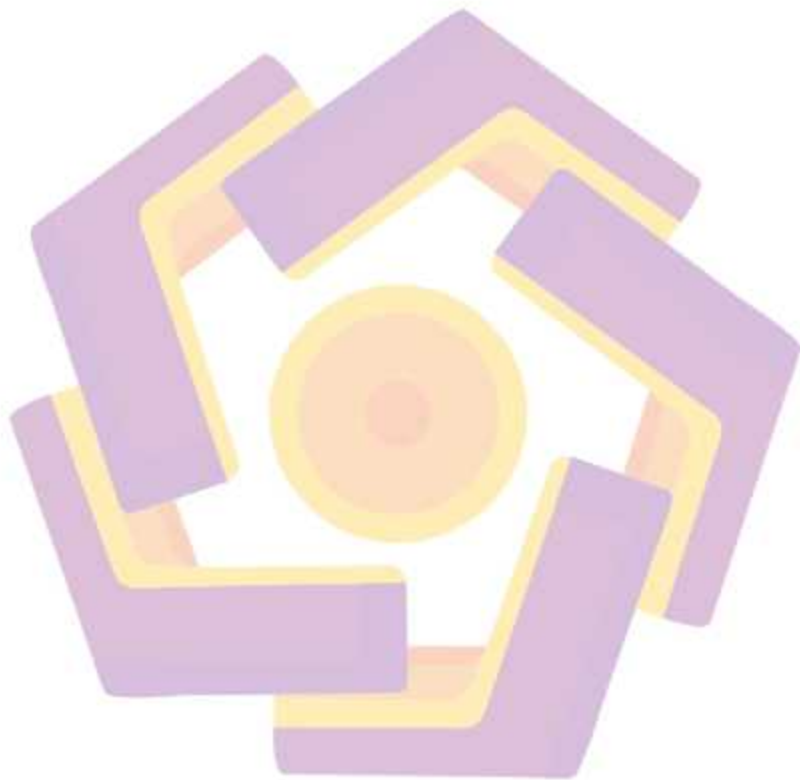
HALAMAN PERSEMBAHAN

Segala puji bagi Allah SWT berkat rahmad dan hidayahnya sehingga saya bisa menyelesaikan thesis saya dengan lancar. Thesis ini saya persembahkan untuk:

1. Orang tua tercinta, berkat kalian berdua memberikan motivasi untuk saya bisa menyelesaikan thesis ini dengan lancar.
2. Rekan dan teman dari semua angkatan magister AMIKOM Yogyakarta yang memberikan masukan dan saran bahan penelitian ini.
3. Rekan kantor Time Excelindo & Qiscus yaitu teman dekat yang selalu menyemangati agar saya dapat menyelesaikan studi magister.

HALAMAN MOTTO

“Kamu tak harus jadi hebat untuk memulai, tapi kamu harus mulai untuk menjadi hebat” - Najwa Shihab



KATA PENGANTAR

Puji dan syukur penulis haturkan kehadiran Allah SWT, karena berkat rahmat dan hidayah-Nya-lah saya dapat menyelesaikan tesis yang berjudul **“DETEKSI SOCIAL DISTANCE DATA VIDEO UNTUK Mencari Sudut dan Jarak Optimal Pendeteksian dengan Metode Deep Learning YOLO V3”**.

Penulis menyadari bahwa tesis masih jauh dari sempurna. Oleh karena itu, penulis berharap dapat belajar lebih banyak lagi dalam mengimplementasikan ilmu yang didapatkan. Tesis ini tentunya tidak lepas dari bimbingan, masukan, dan arahan dari berbagai pihak. Oleh karena itu, pada kesempatan ini saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

Ibu Prof Kusriani sebagai Ketua Program Studi Teknik Informatika yang telah dosen pembimbing pendamping yang telah meluangkan waktu memberikan dukungan, bimbingan dan motivasi dalam menyelesaikan skripsi ini.

Orangtua tercinta, yang telah mendoakan, memberikan dukungan dan memotivasi dalam menyelesaikan tesis ini.

Semua pihak yang telah membantu dan tidak dapat disebutkan satu persatu.

Saya berharap semoga Tuhan Yang Maha Esa mengaruniakan rahmat dan hidayah-Nya kepada mereka semua. Semoga skripsi ini dapat bermanfaat bagi kita semua, Aamiin.

Yogyakarta, 18 November 2022

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
INTISARI.....	xiv
<i>ABSTRACT</i>	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA	6
2.1 Tinjauan Pustaka.....	6
2.2 Keaslian Penelitian.....	8
2.3 Landasan Teori.....	15
BAB III METODE PENELITIAN	19
3.1 Jenis, Sifat, dan Pendekatan Penelitian.....	19
3.2 Metode Pengumpulan Data.....	19
3.3 Metode Analisis Data.....	19
3.4 Alur Penelitian.....	20
3.4.1 Learning Rate.....	22
3.4.2 Max Batches.....	22
3.5 Parameter Performansi.....	22
3.5.1 Akurasi.....	22

3.5.2	Presisi	23
3.5.3	Recall.....	23
3.5.4	F1 Score.....	23
3.5.5	Intersection over Union (IoU).....	23
3.5.6	Confusion Matrix Confusion matrix (error matrix)	24
3.5.7	mean Average Precision (mAP).....	25
BAB IV HASIL PENELITIAN DAN PEMBAHASAN		26
4.1	YOLO v3 Processing	26
4.2	Hasil Training.....	34
4.3	Dataset.....	36
4.4	Skenario Pengujian Sudut.....	38
4.5	Hasil Pengujian Sudut.....	39
4.6	Skenario Pengujian Jarak	51
4.7	Hasil Pengujian Jarak	52
4.8	Tabel Perbandingan.....	63
4.9	Real-time video	65
BAB V PENUTUP.....		66
5.1	Kesimpulan.....	66
5.2	Saran.....	67
DAFTAR PUSTAKA		68

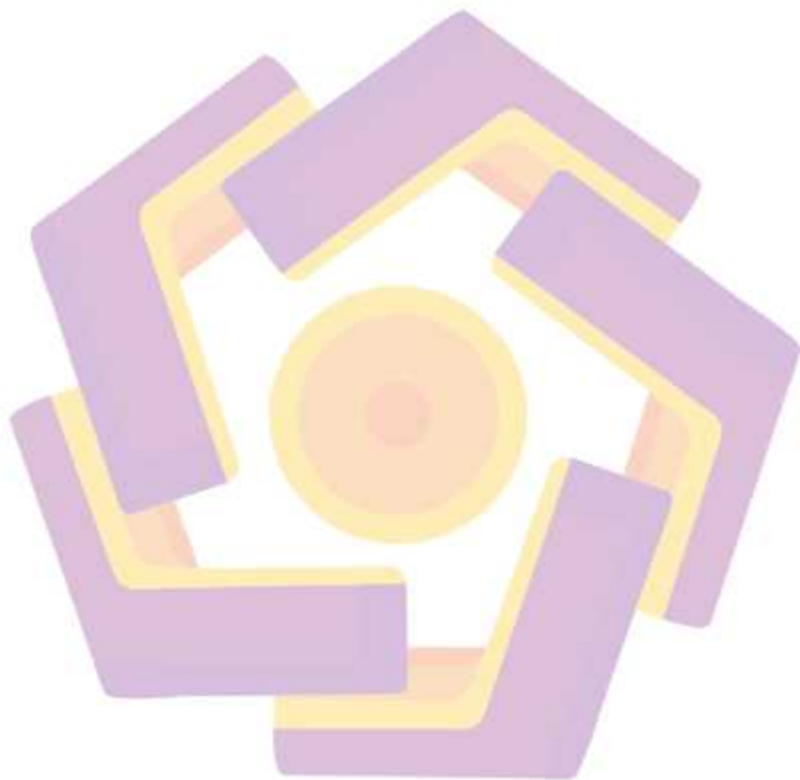
DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	8
Tabel 4.1. Library python use	27
Tabel 4.2. Hasil Training dataset	36
Tabel 4.3 Perhitungan Sudut 90 derajat.....	41
Tabel 4.4 Perhitungan percobaan 60 derajat.....	44
Tabel 4.5 Pengujian 45 derajat.....	46
Tabel 4.6 Perhitungan data 30 derajat.....	48
Tabel 4.7 Perhitungan data 15 derajat.....	50
Tabel 4.8 Perhitungan data jarak 1 meter.....	55
Tabel 4.9 Perhitungan data jarak 2 meter.....	56
Tabel 4.10 Perhitungan data jarak 4 meter.....	59
Tabel 4.11 Perhitungan data jarak 10 meter.....	60
Tabel 4.12 Perhitungan data jarak 20 meter.....	62
Tabel 4.13 Tabel Perbandingan.....	64

DAFTAR GAMBAR

Gambar 2.1	Konsep Yolo	17
Gambar 2.2	Matriks Citra Digital	17
Gambar3.1	Alur Penelitian	21
Gambar3.2	Eror Matrik.....	21
Gambar 4.1	YOLO v3.....	26
Gambar4.2	Konfigurasi coding	29
Gambar 4.3	Class beresiko tinggi	32
Gambar 3.4	Class beresiko sedang	33
Gambar 4.5	Kleas beresiko rendah	34
Gambar 4.6	Yolo Dataset.....	37
Gambar 4.7	Public Dataset Video.....	38
Gambar 4.8	Skenario Sudut	39
Gambar 4.9	Sudut 90 derajat beresiko	40
Gambar 4.10	Sudut 90 derajat tidak beresiko	41
Gambar 4.11	Sudut 60 derajat bersiko.....	43
Gambar 4.12	Sudut 60 derajat tidak beresiko	43
Gambar 4.13	Sudut 45 derajat bersiko.....	45
Gambar 4.14	Sudut 45 derajat tidak beresiko	45
Gambar 4.15	Sudut 30 derajat beresiko.....	47
Gambar 4.16	Sudut 30 derajat tidak beresiko	47
Gambar 4.17	Sudut 15 derajat beresiko.....	49
Gambar 4.18	Sudut 15 derajat tidak beresiko	49
Gambar 4.19	Skenario Pengujian Jarak	51
Gambar 4.20	Alat meteran	52
Gambar 4.21	Jarak 1 m tidak beresiko.....	53
Gambar 4.22	Jarak 1 meter tidak beresiko.....	53
Gambar 4.23	Jarak 2 meter beresiko.....	54
Gambar 4.24	Jarak 2 meter tidak beresiko.....	55
Gambar 4.25	jarak 4 meter beresiko	56
Gambar 4.26	Jarak 4 meter tidak beresiko.....	57
Gambar 4.27	jarak 6 meter beresiko	58

Gambar 4.28	Jarak 6 meter tidak beresiko.....	59
Gambar 4.29	Jarak 20 meter beresiko.....	60
Gambar 4.30	Jarak 20 meter tidak beresiko.....	61



INTISARI

Pandemi Covid-19 memberikan dampak negatif pada ekonomi ke seluruh aspek masyarakat. Salah satu solusi yang hingga saat ini dilakukan oleh pemerintah yaitu menggunakan masker dan jaga jarak atau social distancing. Seiring berjalannya waktu, menjaga jarak di lingkungan masyarakat sulit dikendalikan karena masyarakat telah mengikuti anjuran pemerintah untuk melakukan vaksin sebanyak tiga kali. Salah satu cara untuk mengatasi masalah ini yaitu dengan sistem social distancing detector. Sistem tersebut dapat mendeteksi jumlah dan jarak orang pada suatu area. Penelitian ini bertujuan untuk dapat diterapkan pada area kantor, atau public. Penelitian ini salah satu pengembangan dari aplikasi social distancing detector yang menghasilkan sudut yang optimal dalam pengaplikasian di lingkungan masyarakat. Sehingga program dapat mendeteksi keramaian objek dalam akurasi yang optimal. Sudut sangatlah berpengaruh dalam pengambilan citra yang akan diproses dalam sistem. Pada penelitian ini menggunakan bahasa python dengan library YOLOv3. Penelitian ini mendapatkan hasil terbaik mean average precision di 90%:10% didapatkan dengan learning rate 0.001 dengan nilai mAP 54.11%, akurasi deteksi pada saat pengujian sebesar 100%. Percobaan sudut terdapat 00,150,300, 450,600 dengan total 50 data testing video. Sudut optimal yang didapatkan pada penelitian ini adalah 00,150,300. Hal ini membuktikan bahwa sudut mempengaruhi pengambilan video atau peletakan kamera sistem social distancing.

Kata kunci: Mean average precision, Social distancing, Yolov3, Python

ABSTRACT

The COVID-19 pandemic has had a profound impact on all aspects of society. One of the implementations that the government has so far carried out is using masks and maintaining social distance. Over time, social distancing is difficult to control because people are now getting booster vaccines, but some have not. One way to overcome this problem is with a social distance detector system that detects the number of people and the distance of human objects from one another in an area. This study aims to apply in the office area, or the public. This research is one of the developments of a social distancing detector application that produces an optimal angle in using the application. So the program can detect the entirety of the object with optimal accuracy. Angle is very influential in taking the image to be processed in the system. This study uses the python language with the YOLOv3 library. This study got the best results, and the mean average precision in 90%, 10% didapatkan dengan learning rate 0,001 dengan nilai mAP 54,11%, deteksi pada saat pengujian sebesar 100%. Percobaan sudut terdapat 00.150.300, 450.600 dengan total 50 data video testing. Sudut optimal yang didapatkan pada penelitian ini adalah 00.150.300. Hal ini membuktikan bahwa sudut pengambilan video atau peletakan kamera sistem social distancing.

Keywords: *Mean average precision, Social Distancing, Yolov3, Python*

BAB I

PENDAHULUAN

1.1.Latar Belakang Masalah

Social Distancing atau bisa disebut juga saat ini dengan *Physical Distancing* artinya menjaga jarak aman antara diri sendiri dengan orang lain terutama orang yang tidak satu rumah dengan orang tersebut. Jarak aman yang disebutkan menjadi anjuran pemerintah adalah jarak satu hingga dua meter. Cara ini juga harus diikuti dengan protokol lainnya seperti harus selalu memakai masker, mencuci tangan contohnya

Program pendeteksian pelanggaran *Social Distancing* saat ini sudah pernah diteliti oleh beberapa peneliti pendahulu. Salah satunya di penelitian yang berjudul "Social Distancing Detection with Deep Learning Model". Pada penelitian kali ini algoritma yang digunakan yaitu YOLO V2 yang merupakan salah satu dari pengembangan dari YOLO V1 dalam pendeteksian manusianya.

Maka dari itu, pada penelitian kali ini penulis mengusulkan untuk menggunakan algoritma You Only Look Once (YOLO) yang memiliki performansi yang lebih tinggi dari versi sebelumnya yaitu YOLOv3 dan dengan berbagai sudut pandang yang berbeda contohnya jika objek manusia dari topview memiliki arah yang berbeda. Tujuan dari penelitian ini adalah membuat sistem yang mendeteksi pelanggaran secara real-time dengan efektif dan akurat di lingkungan outdoor maupun indoor.

Salah satu protokol kesehatan adalah menghindari keramaian yaitu dengan menerapkan *social distancing*. Seseorang cenderung sulit mengikuti protokol

kesehatan yang satu ini. Faktor penularan orang satu dengan yang lainnya paling penting adalah karena kontak fisik satu sama lain. Yang diketahui ini kita sedang memasuki pada era Revolusi Industri 4.0 dimana seluruh system ada dalam kehidupan manusia. Di era ini, digitalisasi dan efisiensi sangat diperlukan untuk perkembangan teknologi (Nugroho et al.) Salah satu cara untuk mengetahui batasan keramaian dan social distancing adalah dengan social distancing detector yaitu aplikasi yang dapat menghitung banyaknya jumlah orang dan menganalisis jarak antar orang yang disebut melanggar jarak 1-2 meter di area tertentu dengan memberikan pemberitahuan jika orang di area tersebut telah melebihi batas yang ditentukan dan tidak menerapkan protokol kesehatan social distancing. Dengan memungkinkan komputer untuk mendeteksi jarak sosial, maka akan memudahkan pekerjaan manusia dengan menghitung jumlah orang dalam citra digital yang terdiri dari banyak orang, seperti di jalan raya (Rohman 2018). Pengembangan aplikasi ini menggunakan bahasa pemrograman Python dan library YOLOv3. Alasan yang dianggap akurat menggunakan library YOLOv3 adalah karena kemampuan YOLOv3 dapat mendeteksi objek dengan frame rate yang lebih tinggi (Hammam et al. 2020).

Algoritma YOLO (You Only Look Once) adalah salah satu algoritma deep learning yang dapat di compare dengan jaringan syaraf konvolusional (CNN) dalam mendeteksi sebuah objek. Pada proses CNN, terdapat 3 tahapan yaitu pre-processing, processing, dan classifying (Munadhif et al., 2020). Algoritma membagi gambar ke dalam kisi-kisi ukuran $s \times s$, dan kemudian memprediksi kotak pembatas atau bisa disebut (bounding box) dan peta kelas dari setiap kisi di setiap kisi. Pada satu grid apabila terprediksi objek, maka pada grid tersebut akan

diprediksi yaitu bounding box yang mengelilingi objek tersebut. Nilai confidence yang akan dihitung masing-masing pada bounding box yang kemudian akan diseleksi berdasarkan pada nilai yang didapat. (Nazilly et al. 2020). Sistem deteksi yang diterapkan pada penelitian ini adalah melakukan deteksi dengan menggunakan reuse classifier atau locator (Rachmawati & Widhyaestoeti 2020). Penelitian ini bertujuan untuk mengembangkan aplikasi social distancing detector menggunakan bahasa pemrograman menggunakan Python dengan library YOLOv3.

Pada penelitian ini sesuai saran dari penelitian sebelumnya bahwasanya deteksi social distancing akan sulit menjangkau jika manusia dengan manusia lain saling sejajar dalam penangkapan kamera. Dalam penelitian ini akan membuktikan tingkat akurasi deteksi social distancing dengan berbagai percobaan arah dan sudut diharapkan dapat mempunyai titik kerakurasi yang berbeda dengan faktor sudut dan jarak.

1.2. Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah penelitian ini adalah:

1. Bagaimana hasil training dengan akurasi pada deteksi pelanggaran social distancing menggunakan algoritma YOLOv3 dikolaborasi dengan rumus euclidean distance?
2. Berapakah jarak optimal dan sudut optimal dalam system mendeteksi social distancing?
3. Bagaimana kinerja aplikasi deteksi social distancing ketika dijalankan real time?

1.3. Batasan Masalah

Adapun batasan masalah penelitian ini adalah:

1. Penelitian ini video yang digunakan cctv dengan pengambilan gambar dari atas (topview) dengan dataset yaitu mendapatkan dari 2 sumber, dataset penelitian sebelumnya secara (public) YOLO COCO dan dataset pengambilan video pribadi.
2. Arah dan sudut yang ditentukan sesuai dengan penelitian ini.
3. Device camera yang digunakan camera lensa normal dengan pengambilan video landscape dan portrait.
4. Penelitian ini menggunakan pemrograman python dan library YOLO dengan framework OpenCV.

1.4. Tujuan Penelitian

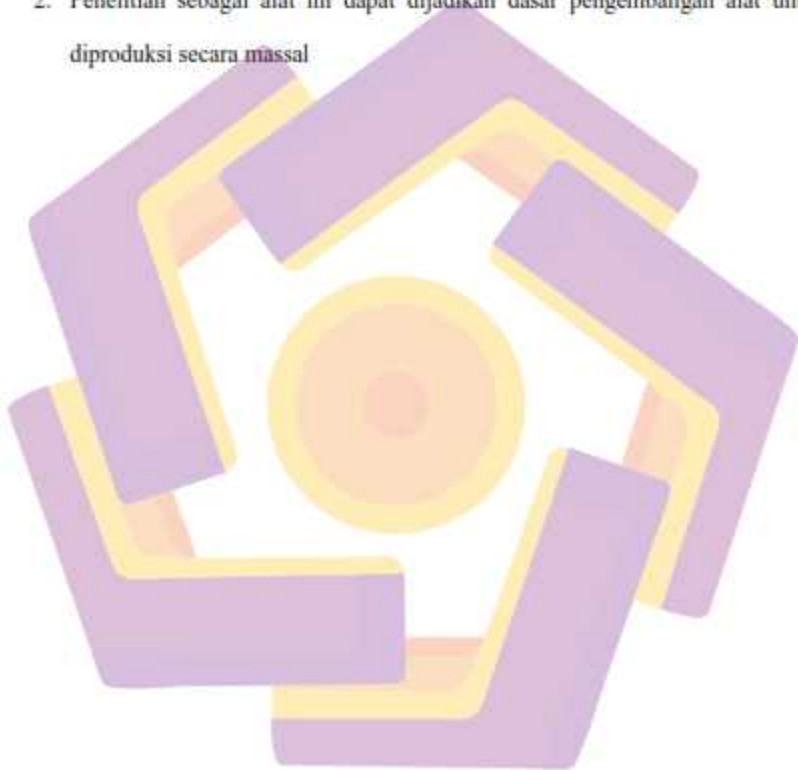
Tujuan pada penelitian ini yaitu:

1. Untuk menghasilkan akurasi terbaik dengan mengintegrasikan data yang dihasilkan dengan sistem social distance berbasis metode deep learning YOLOv3 dengan memanfaatkan rumus euclidean distance.
2. Untuk mengetahui jarak dan sudut yang optimal dihasilkan dengan sistem social distance mendapatkan akurasi tinggi.
3. Untuk menghasilkan sistem deteksi social distance secara realtime dari data video.

1.5. Manfaat Penelitian

Adapun manfaat penelitian :

1. Membantu civitas akademik khususnya petugas keamanan dalam menerapkan protokol Kesehatan yaitu social distancing.
2. Penelitian sebagai alat ini dapat dijadikan dasar pengembangan alat untuk diproduksi secara massal



BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Anan Nugroho, Risanuri Hidayat, Hanung Nugroho dan Johan Debayle pada tahun 2021 dengan judul "Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLOv3 Sebagai Protokol Kesehatan COVID-19. Penelitian ini bertujuan untuk melakukan deteksi antar jarak satu dengan jarak lainnya menggunakan algoritma YOLO. Penelitian ini mampu mendeteksi dengan keakurasian. Hal yang menjadi kelemahan dalam penelitian ini dikarenakan tidak adanya menggunakan data set. Dan beberapa percobaan training menggunakan foto.

Penelitian yang dilakukan oleh Harits Hammam Al Asyhar, Suryo Adhi Wibowo dan Gelar Budiman dilakukan tahun 2019 dengan judul Implementasi dan Analisis Perfirmasi Metode You Only Look Once (YOLO) Sebagai Sensor Pornografi pada Video. Penelitian ini bertujuan untuk mendapatkan input video dan mendeteksi apakah film tersebut lolos dari uji pornografi atau tidaknya. Peneliti menggunakan data set dan data latih menghitung mean average performansi.

Penelitian yang dilakukan oleh Muhammad Irfan Hermawan, Iwan Iwut Tritoasmoro dan Nur Ibrahim pada tahun 2019 dengan judul Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo. Penelitian ini bertujuan untuk mengetahui apakah pengaturan lampu lalu lintas telah memenuhi kepadatan atau standar kepadataan. YOLOv2 dirasa cocok untuk deteksi

berikut dan mampu keakurasian cukup tinggi dengan frame rate yang lebih tinggi dibandingkan dengan algoritma lain.

Penelitian yang dilakukan oleh Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam pada tahun 2019 dengan judul Real-Time Object Detection with Yolo. Penelitian ini bertujuan untuk melihat dengan metode YOLO ini apakah waktu deteksi akurat yang dihasilkan atau tidaknya menggunakan objek yaitu mobil dengan mendeteksi plat nomor tersebut sehingga dapat dideteksi. Dan ternyata terbukti YOLO dapat realtime saat mendeteksi objek namun terdapat beberapa masukan saran, dengan jarak tertentu dan saat mobil berada pada tikungan, sistem deteksi tidak bisa mendeteksi plat nomor tersebut.

Penelitian yang dilakukan oleh M. Nishom pada tahun 2019 dengan judul Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis-Chi-Square. Penelitian ini bertujuan untuk melihat berbagai rumus jarak yang ada, dan menghasilkan tingkat paling tinggi pada Euclidean Distance dirasa mendekati jarak sesungguhnya dalam asli dibanding dengan rumus perhitungan jarak lainnya.

Penelitian yang dilakukan oleh Xiaoyue Jiang, Abdenour Hadid, Yanwei Pang, Eric Granger, Xiaoyi Feng pada tahun 2019 dengan judul Deep Learning in Object Detection and Recognition. Penelitian ini bertujuan untuk melihat apakah deep learning metode tepat dalam mendeteksi gambar dan objek yang dari media record. Dan hasilnya penelitian ini membuktikan sebuah objek dapat dikenali cepat dengan deep learning.

2.2 Keaslian Penelitian

Table 2.1. Matriks literatur review dan posisi penelitian
 Deteksi Social Distance Data Video Untuk Mencari Sudut dan Jarak Optimal Pendeteksian Dengan Metode Deep Learning YOLO V3

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLOv3 Sebagai Protokol Kesehatan Covid-19	Anan Nugroho, Risanuri Hidayat, Hariung Nugroho, Johan Debayle, 2021	Menggunakan YOLOv3 dengan pada penelitian ini hanya menggunakan library yolo tanpa terdapat perhitungan jarak.	Berdasarkan hasil pengujian digunakan untuk mendeteksi jarak seseorang. Dapat dikatakan bahwa tujuan dari penelitian ini tercapai karena aplikasi yang dikembangkan memiliki hasil. Ada beberapa faktor yang mempengaruhi akurasi dari aplikasi ini. Semakin Jelas input yang digunakan maka	Saran dalam penelitian ini menggunakan dataset pada YOLOv3 agar lebih akurat, menggunakan rumus untuk perhitungan jarak yang akurasi lebih tinggi dikarenakan data videonya yang kurang cukup. Penelitian ini tidak bisa mendapatkan jarak optimal dalam deteksi yang tepat	Penelitian terdahulu yolo v3 dataset dari sumber penelitian public sedangkan penelitian saya menggunakan data pribadi yang diambil sendiri, Dan perhitungan menggunakan euclidean distance. Penelitian terdahulu tidak bisa menerapkan jarak optimal dan juga sudut optimal system bekerja. Pada saran penelitian ini mempunyai saran yaitu jarak

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				akurasi yang akan didapatkan juga akan lebih baik. Jarak yang kurang tepat menjadi peran utama dalam penelitian ini		dari kamera pada objek manusia kurang pas presisinya.
2	Implementasi Dan Analisis Performansi Metode You Only Look Once (YOLO) Sebagai Sensor Pornografi Pada Video	Harits Hammam Al Asyhar, Suryo Adhi Wibowo, Gelar Budiman, 2020	Yolov3 menggunakan data set dan data latih menghitung mean average performansi (mAP).	Penelitian ini tingkat akurasi yang cukup tinggi dan frame rate yang lebih tinggi dibandingkan dengan algoritma deteksi objek state-of-the-art lainnya. Namun penelitian ini kurang akurat jika kendaraan melaju lebih dari 60km/jam ke atas	Tidak menggunakan alat sebagai media sensor film. Tidak bisa melihat fine tuning learning untuk proses training lebih bagus kedepannya.	Menggunakan dataset yang berbeda. Menggunakan jarak antar objek satu dengan objek lainnya.

Tabel 2.1 Lanjutan

No	Judul	Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				akan sulit dideteksi oleh alat.		
3	Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo	Muhammad Irfan Hermawan, Iwan Iwut Tritoasmoro, Nur Ibrahim, 2021	Yolo v2 algoritma mampu mendeteksi suatu objek dengan tingkat akurasi yang cukup tinggi dan frame rate yang lebih tinggi dibandingkan dengan algoritma deteksi objek state-of-the-art.	Skema dataset yang digunakan sebanyak 624 data latih dan 156 data uji. Parameter performansi yang ditinjau yaitu Average Precision (AP) dan akurasi. Nilai AP tertinggi sebesar 0,89 dengan konfigurasi hyperparameter learning rate 0,0001, epoch 60, dan batch size 4. Hasil pengujian akurasi didapat nilai rata-rata akurasi tertinggi adalah 98,80%	Dengan update nya versi YOLO ke v3 maka akan lebih jauh akurat dalam mendeteksi objek satu kendaraan dan kendaraan lainnya menggunakan ccv dan update versi 3	Mendeteksi tingal kepadatan kendaraan menggunakan YOLOv2, sedangkan untuk mengetahui jarak social distance menggunakan YOLOv3, dan rumus jarak objek manusia yang berbeda dibanding kendaraan lalu lintas.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Geethapriya, S. N. Duraimurugan, S.P. Chokkalingam, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019	Mengetahui seberapa tingkatan akurasi yolo mengetahui dan membaca objek berupa plat nomor mobil. Pengujiannya beberapa gambar plat nomor mobil di deteksi dan sistem akan mengetahui plat nomor tersebut	Yolo membuktikan dapat mendeteksi objek satu dengan yang lain, dapat melihat beberapa bounding box yang di dapatkan, dalam 1 frame mempunyai beberapa bounding box yang ada dalam satu frame video. Dapat terbaca plat nomor, namun beberapa kasus jika plat nomor tersebut dalam jarak di atas 5m kurang tepat hasilnya.	Saran untuk penelitian, menggunakan jarak yang tepat dan memastikan nomor plat tersebut berwarna putih dan masih jelas terbaca. beberapa mobil pada saat belokan maka tidak akan membaca plat tersebut.	Penelitian saya menggunakan video dalam kasus ini, dan dapat menggunakan bounding box yang sama, dengan jarak yang tepat untuk mengukur jarak ideal serta menggunakan sudut yang tepat agar posisi cctv dapat diatur dengan baik kinerja pada sistem deteksi itu sendiri.	

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square	M. Nishom, 2019	Membuktikan mana perhitungan jarak dengan kmeans clustering metode yang terbaik	Perbandingan akurasi metode pengukuran jarak (euclidean, manhattan, dan minkowski) untuk pelabelan klaster status disparitas kebutuhan Gnu telah dilakukan dan memberikan nilai atau tingkat akurasi yang tinggi, yaitu 84.47% (untuk metode euclidean distance), 83.85% (untuk metode manhattan distance), dan 83.85% (untuk metode minkowski). Dengan demikian, dapat disimpulkan bahwa metode euclidean merupakan metode terbaik untuk diterapkan	Saran kedepannya menggunakan Euclidean untuk algoritma selain KMeans dan Clustering	Penelitian kedepannya menggunakan Euclidean pada algoritma CNN

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				dalam algoritma KMeans Clustering.		
6	Deep Learning in Object Detection and Recognition	Xiaoyue Jiang, Abdenour Hadid, Yanwei Pang, Eric Granger, Xiaoyi Feng, 2019	Menceliti apakah objek detection recognition dapat digunakan dengan metode deep learning	Membuktikan bahwa objek dapat terdeteksi sesuai dengan dataset yang diberikan. Setiap objek akan mempunyai objektifitas dan dapat di deteksi benda. Dalam hal ini objek akan dikenali oleh computer lalu dapat user namai benda tersebut, jika system menemukan objek yang serupa dengan sebelumnya maka system akan memanggil ulang data	Mengeset data menggunakan objek selain benda dan apakah dapat di padukan dengan metode lainnya	Penelitian tersebut menggunakan agar computer dapat mendeteksi dan mengenal objek. Sedangkan pada penelitian saya membutuhkan computer agar mendekteksi objek.

Tabel 2.1 Laniutan

No	Judul	Penciti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				yang diberikan sebelumnya.		

2.3 Landasan Teori

You Only Look Once (YOLO)

You Only Look Once atau YOLO adalah Jaringan saraf cerdas untuk deteksi waktu nyata. YOLO menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan dibagi menjadi daerah-daerah, kemudian akan memprediksi kotak pembatas dan probabilitas Untuk setiap kotak di wilayah pembatas, timbang probabilitas untuk mengklasifikasikannya sebagai objek atau bukan. YOLO membagi citra masukan menjadi kisi-kisi berukuran, dimana nilai S adalah 7 dan ukuran citra masukan adalah 448×448 . Untuk mendapatkan kotak pembatas, kita akan menggulung gambar input. Sebuah kotak pembatas memiliki 5 nilai yang harus disimpan: koordinat, koordinat, lebar (*width*), tinggi (*height*), skor kepercayaan (kotak pembatas yang dimaksud memiliki nilai probabilitas objek) (Hermawan et al., 2021).

YOLOv3 memiliki 53 lapisan konvolusion, yang disebut Darknet 53, yang terdiri dari struktur konvolusion dan residual. Di YOLOv3, lapisan konvolusi selalu diikuti dengan normalisasi *batch* dan kebocoran ReLU.

Sambungan blok residual atau pintasan di YOLOv3 diselesaikan dengan menambahkan entri di atas blok residu lapisan konvolusi ke hasil filter lapisan konvolusi 1×1 , lalu melakukan normalisasi *batch* dan penyaringan ReLU, dilanjutkan dengan *convolutional layer filter* 3×3 dengan *batch normalization*, dan pada bagian akhir dilakukan *leaky ReLU*. *Residual layer* pada YOLOv3 didefinisikan sebagai operasi 18 penjumlahan antara input awal dengan output

Hasil sesudah dilakukan *convolutional layer* pada *residual block* (putra et al., 2021). Opencv (*open source computer visionlibrary*) adalah perpustakaan perangkat lunak untuk pemrosesan gambar waktu nyata yang dinamis, dibuat oleh intel, dan sekarang didukung oleh willow garage dan itseez. Opencv dirilis di bawah lisensi bsd yang lebih

liberal daripada GPL, dan memberikan kebebasan penuh untuk penggunaan komersial tanpa mengungkapkan kode sumbernya. Ini juga memiliki antarmuka yang mendukung Bahasa pemrograman C ++, C, Python, dan Java, termasuk sistem operasi Windows, Linux, Mac OS, iOS, dan Android. OpenCV dirancang untuk efisiensi komputasi, dengan fokus pada aplikasi waktu nyata (Zein 2018). *Object Detection* atau deteksi objek merupakan bagian dari *Computer Vision*. *ObjectDetection* mengacu pada kemampuan computer untuk mendeteksi sejumlah objek pada suatu gambar. Hal ini dapat dilakukan dengan cara mengambil *image feature* seperti garis, sudut, kontur dan warna dari sebuah gambar. Deteksi objek merupakan bagian dari *Object Recognition* atau identifikasi objek. Oleh karena itu dapat disimpulkan bahwa untuk deteksi objek pasti harus diidentifikasi terlebih dahulu objek tersebut. Sedangkan pada penelitian ini akan dilakukan deteksi objek kemudian.

You Only Look Once (YOLO) merupakan algoritma *Deep Learning* yang terus dikembangkan sampai sekarang. YOLO juga merupakan pengembangan dari CNN dan diciptakan oleh *Joseph Redmon* pada tahun 2015. Algoritma YOLO menggunakan *single neural network* untuk mendeteksi *bounding box* dan *class probabilities* secara langsung dari keseluruhan gambar dalam sekali evaluasi.

Algoritma ini juga didukung dengan bantuan GPU untuk dapat memproses gambar secara *real-time* hingga 45 fps. Pada gambar di bawah, dijelaskan secara ringkas bagaimana proses algoritma ini dilakukan. Dimulai dengan mengubah *resize* gambar dan proses *convolutional network* dilakukan. Terakhir dengan metode *non-max suppression* menemukan objek dengan *bounding boxes* yang memiliki *confidence score* tertinggi.



Gambar 2.1 Konsep Yolo

Citra

Citra adalah gambaran suatu objek atau kemiripan dari suatu objek barang, manusia, dll. Citra merupakan kombinasi dari titik, garis, bidang, dan warna untuk merepresentasikan suatu objek dari data yang ada.

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

Gambar 2.2 Matriks Citra Digital

Deep Learning

Deep Learning merupakan cabang dari *Machine Learning* dimana pemodelan data bersifat abstrak tingkat tinggi dan memiliki *layer* yang sangat banyak dan

juga kompleks. Pembelajaran di dalam *Deep Learning* bersifat *supervised*, *semi-supervised*, dan *unsupervised*.

Trasfer Learning

Transfer Learning merupakan sebuah metode yang menggunakan suatu model yang telah terbentuk dan dilatih oleh suatu *dataset* dan kemudian model tersebut digunakan kembali untuk permasalahan yang berbeda dengan mengganti atau memodifikasi parameter atau konfigurasinya sesuai dengan *dataset* yang baru.

Euclidean Distance

Dalam menghitung jarak antar orang berdasarkan yang terdeteksi, akan menggunakan metode *Euclidean Distance*. Metode ini akan menghitung jarak antar objek dengan mengukur jarak dua titik di dalam *Euclidean Space*, dan dapat berupa dua dimensi ataupun tiga dimensi, rumus:

$$d(x,y) = |x-y| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \dots\dots(1)$$

d = jarak antara x dan y

x = data pusat kluster

y = data pada atribut

i = setiap data

n = jumlah data,

x_i = data pada pusat kluster ke i

y_i = data pada setiap data ke i

BAB III

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Bagian ini memuat jenis, sifat dan pendekatan penelitian yang digunakan beserta penjelasan secara garis besar mengenai penelitian ini sehingga dapat dikelompokkan dalam penelitian tersebut.

3.2 Metode Pengumpulan Data

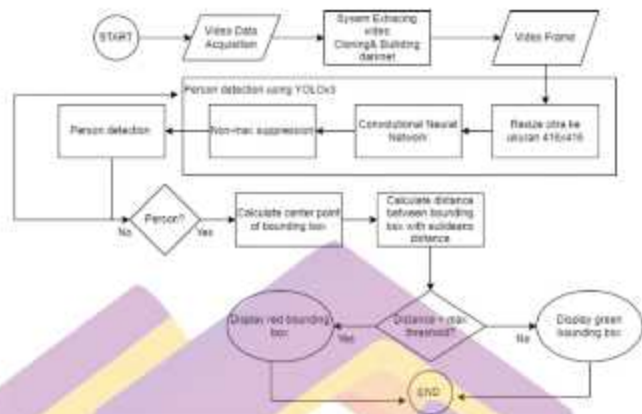
Bagian ini memuat penjelasan secara lengkap dan terinci tentang cara-cara yang digunakan dalam proses pengumpulan data untuk jenis data yang diperlukan. Misalnya melalui observasi, wawancara, eksperimen, atau penyebaran angket. Jika metode penyebaran angket digunakan, maka blangko angket harus dilampirkan dalam proposal. Untuk setiap metode pengumpulan data, harus dijelaskan tentang jenis data yang dikumpulkan dengan metode-metode yang terkait. Bagian ini juga memuat penjelasan secara lengkap dan terinci tentang jenis data yang diperlukan untuk analisis dalam pembahasan.

3.3 Metode Analisis Data

Bagian ini memuat penjelasan secara lengkap dan terinci tentang metode dan alat yang digunakan untuk analisis data.

3.4 Alur Penelitian

Penelitian ini mempunyai proses utama, yaitu saat video data diakuisisi maka sistem akan mengekstraksi video. Proses *building darknet* sebagai arsitektur YOLOv3. Kemudian akan dilakukan penyesuaian frame terhadap *configuration* file dari video. Lalu sistem akan mendeteksi objek dengan YOLOv3 yaitu resize citra ke ukuran 416x416 dilanjutkan CNN dan *non-max suppression*. Sehingga hasil ini akan tersimpan untuk digunakan mendeteksi orang. Jika orang/objek sudah terdeteksi maka sistem akan menentukan *center point* dan *bounding box* setiap objek. Sistem selanjutnya menghitung antara *center point* orang dengan lainnya. Jika jaraknya tidak sesuai dengan *threshold* maka *bounding box* tersebut akan berwarna merah yang berarti jarak kurang dari 1-2 meter. Sementara jika kurang dari *threshold* maka *bounding box* tersebut berwarna hijau yang berarti objek person tersebut dekat dengan 1-2meter. Pada Gambar 2 dijelaskan alur keseluruhan sistem. Dimana video saat di inputkan akan menghasilkan ekstrasi video dilanjutkan memasukan video frame. Maka setelah video input sistem akan mendeteksi YOLO dan kinerja YOLO yaitu membentuk citra ke ukuran 416x416. Selanjutnya setelah citra terbagi ukuran maka CNN akan bekerja dan *non max suppression* dalam YOLO mendeteksi orang. Inputan kinerja YOLO apakah orang terdeteksi atau tidak. Jika tidak maka sistem akan kembali ke pada proses deteksi YOLO.. Namun jika sudah terdeteksi maka sistem akan mengkalkulasi *center point* di dalam *bounding box*. Selanjutnya sistem akan mengkalkulasi bounding box orang tersebut ke orang lainnya. Secara umum berikut cara kerja sistem yang dapat dilihat melalui *flowchart* gambar 2 berikut ini:



Gambar.3.1 Alur Penelitian

Inputan dari video akan di input kedalam sistem. Selanjutnya sistem akan mengekstrak dan *cloning building darknet*. Sistem akan di masukan frame. Maka sistem akan mengatur ulang citra menjadi ukuran 416x416. Dalam mendeteksi orang makan YOLO akan menggunakan *convolutional neural network* dan *non-max suppression*. Jika orang sudah terdeteksi maka akan dikalkulasikan dengan *center point* dari *bounding box* yang di dapatkan. Jika orang masih belum terdeteksi maka sistem akan Kembali pada deteksi YOLO. Setelah mendapatkan *center point*, maka sistem akan dihitung jarak antara *bounding box* satu dengan lainnya. Maka sistem akan menghitung apakah jarak tersebut lebih kecil dari max threshold atau tidak. Jika lebih besar maka akan menampilkan kotak hijau. Namun jika hasilnya lebih kecil maka yang akan tampil berupa kotak merah melambangkan jarak yang berdekatan dibawah 1m.

3.4.1 Learning Rate

Learning rate merupakan penentu seberapa banyak *weight* yang diperbarui dalam proses *backpropagation*. *Learning rate* juga menentukan kecepatan pada iterasi sehingga dapat mencapai *loss function minimum*. Proses *training* berjalan semakin cepat jika nilai *learning rate* semakin tinggi. Namun, nilai *learning rate* yang terlalu tinggi juga dapat menyebabkan nilai *loss function* turun-naik tidak menentu, sehingga dibutuhkan beberapa kali percobaan untuk mendapatkan nilai *learning rate* yang optimal..

3.4.2 Max Batches

Max batches merupakan banyaknya iterasi pada proses *training data*. Semakin tinggi nilai *max batch*, maka sistem akan semakin banyak mempelajari *data training*. Jumlah *training data* tidak boleh lebih dari jumlah *max batches*. Nilai *max batches* perlu disesuaikan dengan jumlah kelas dari objek yang akan dideteksi.

$$\text{Max batches} = \text{number of classes} \times 2000$$

3.5 Parameter Performansi

3.5.1 Akurasi

Akurasi merupakan parameter uji yang menentukan keandalan sistem dalam mengklasifikasikan objek. Parameter akurasi berfungsi untuk menentukan kebenaran sistem dalam mendeteksi orang terhadap data uji. Akurasi dapat ditentukan menggunakan persamaan.

3.5.2 Presisi

Presisi merupakan rasio dari jumlah objek yang terdeteksi dengan benar atau *True Positive* dibandingkan dengan seluruh data yang diprediksi positif. Dari persamaan (4), dapat diketahui bahwa semakin besar nilai *False Positive*, maka presisi semakin rendah dan begitupun kebalikannya.

3.5.3 Recall

Recall merupakan rasio dari jumlah objek yang terdeteksi dengan benar atau *True Positive* dibandingkan dengan seluruh data yang positif. Nilai *recall* yang tinggi menunjukkan bahwa sistem dapat mengklasifikasikan kelas objek dengan benar. *Recall* dapat dikalkulasikan dengan persamaan.

3.5.4 F1Score

F1 Score merupakan perbandingan rata-rata dari nilai presisi dan *recall*. *F1 Score* memiliki nilai tertinggi sebesar 1 dan terendah sebesar 0. Nilai *F1 Score* yang semakin mendekati 1, menunjukkan bahwa kinerja sistem telah baik. Secara matematis, nilai *F1 Score* dapat dilihat pada persamaan.

3.5.5 Intersection over Union (IoU)

IoU merupakan metrik yang mengevaluasi keakuratan sistem dalam mendeteksi objek pada *dataset* yang telah dilatih. IoU membandingkan *ground-*

truth atau objek pada citra dengan *predicted bounding box* dari model. Nilai IoU pada deteksi objek berperan sebagai nilai *threshold*.

Terdapat dua nilai *threshold* IoU yang umum digunakan, yaitu 0.5 dan 0.75. Pada penelitian ini, nilai *threshold* yang digunakan adalah 0.5. Jika nilai *threshold* $\text{IoU} \geq 0.5$, maka objek terdeteksi sebagai *True Positive* (TP). Dan ketika *threshold* $\text{IoU} < 0.5$, maka objek terdeteksi sebagai *False Positive* (FP).

3.5.6 Confusion Matrix Confusion matrix (error matrix)

artinya memberikan informasi perbandingan antara hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Confusion matrix ini berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji dimana nilai sebenarnya diketahui.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 3.2 Error Matrik

Tabel pada gambar 9 di atas merupakan 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix, yaitu True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Pada penjelasan berikut 4 istilah dari tabel confusion matrix dengan contoh yang sesuai dengan penelitian Tugas

Akhir ini : a. True Positive (TP), adalah data positif yang diprediksi benar oleh sistem (model). b. True Negative (TN), adalah data negatif yang diprediksi benar oleh sistem (model). c. False Positive (FP), merupakan data negatif akan tetapi sistem (model) mendeteksi sebagai data positif. d. False Negative (FN), merupakan data positif akan tetapi sistem (model) mendeteksi sebagai data negatif. Berikut ini adalah tujuan dan manfaat dari Confusion Matrix : a. Menunjukkan bagaimana sistem (model) yang dibuat membuat suatu prediksi. b. Memberi informasi tentang kesalahan yang dibuat dan jenis kesalahan yang dibuat oleh sistem (model). c. Setiap kolom dari confusion matrix merepresentasikan instance dari kelas prediksi. d. Setiap baris dari confusion matrix mewakili instance dari kelas aktual.

3.5.7 mean Average Precision (mAP)

Nilai mAP merupakan metrik yang mengevaluasi performansi dari model deteksi objek. Sederhananya, mAP merupakan rata-rata dari nilai *average precision* (AP) dan mengukur seberapa bagus performansi dari *weights file* hasil *training data*. Sebelum mengkalkulasikan mAP, perlu dilakukan penyesuaian nilai *threshold* pada IoU untuk menentukan validasi dari objek yang dideteksi. Penelitian ini menggunakan nilai *threshold*

IoU sebesar 0.5, sehingga model menghasilkan mAP50 yang dapat juga disebut mAP@0.5.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 YOLO v3 Processing

Sebelum YOLO-V3, metode YOLO-V2 menggunakan Darknet-19 sebagai feature extractor dengan 19 convolutional layer dan 5 max pooling layer dan softmax layer untuk klasifikasi objek, sedangkan YOLO-V3 menggunakan jaringan Darknet-53 sebagai feature extractor yang memiliki 53 convolutional layer didalamnya serta filter 3x3 dan 1x1 dengan koneksi pintasan. Hal ini jauh lebih dalam dari pada YOLO-V2.



Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
1x Convolutional	32	1 × 1	
1x Convolutional	64	3 × 3	
1x Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
2x Convolutional	64	1 × 1	
2x Convolutional	128	3 × 3	
2x Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
8x Convolutional	128	1 × 1	
8x Convolutional	256	3 × 3	
8x Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
8x Convolutional	256	1 × 1	
8x Convolutional	512	3 × 3	
8x Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
4x Convolutional	512	1 × 1	
4x Convolutional	1024	3 × 3	
4x Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

Gambar 4.1 YOLO v3

Sistem ini menggunakan library deskripsi disini melalui

<https://docs.python.org/id/> :

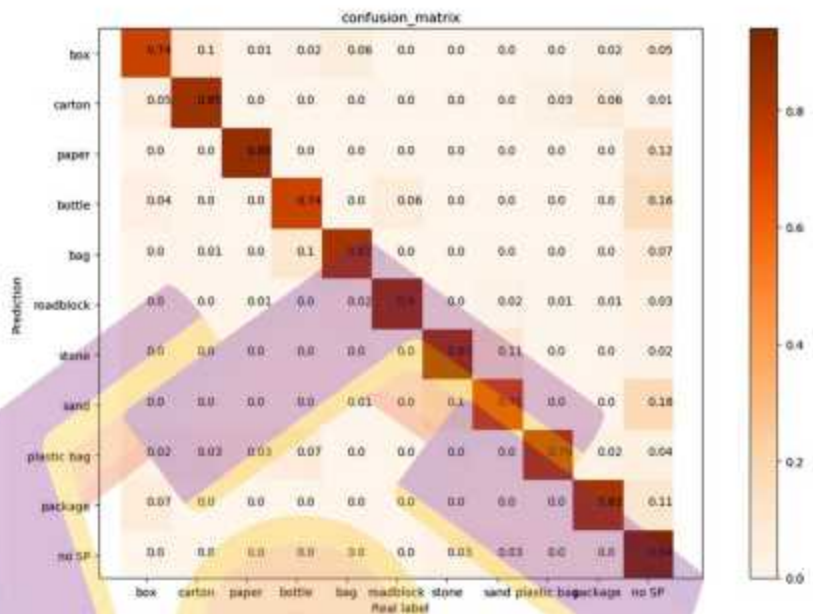
Tabel 4.1. Library python use

No	Lib name	Version	Remaks
1	Argparse	1.4.0	argparse merupakan sebuah pustaka untuk melakukan argument processing. Setiap argumen dapat melakukan trigger dengan aksi yang berbeda, dengan mendefinisikan action argumen pada fungsi <code>add_argument()</code> . Beberapa aksi yang tersedia seperti menyimpan sebuah nilai variabel ataupun konstanta ketika dilakukan action tersebut, tipe data yang disimpan bisa beragam dari boolean, integer, string dan lainnya
2	imutils	0.5.3	Imutils merupakan library untuk melakukan translation, rotation, resizing, and skeletonization.
3	numpy	1.19.2	Numpy merupakan modul untuk komputasi ilmiah dalam Python. NumPy berisi tenyang : <ul style="list-style-type: none"> - Objek N-Dimensional Array - Broadcasting yang cepat - Tools yang terintegrasi dengan C/C++ dan Fortran - Fungsi aljabar linear, fourier transform dan kemampuan pengolahan angka.
4	opencv-python	4.2.0.32	OpenCV merupakan libaty yang digunakan untuk mengolah gambar dan video untuk dapat meng-ekstrak informasi didalamnya. OpenCV dapat berjalan di berbagai bahasa pemrograman seperti C, C++, Java, Python. Contoh dalam penggunaan OpenCV adalah bagaimana kita mendeteksi wajah dalam sebuah

Tabel 4.1 Lanjutan

			gambar.
5	pip	21.1.2	Basic penggunaan python
6	playsound	1.4.1	modul paling sederhana yang digunakan untuk memutar suara. Modul ini bekerja pada Python 2 dan Python 3, dan diuji untuk memutar file wav dan mp3 saja. Ini hanya berisi satu metode, bernamaplaysound(), dengan satu argumen agar Linux mengambil nama file audio untuk diputar.
7	scipy	1.4.1	Mengeksekusi perhitungan matematika dalam code pyhton
8	setuptools	57.0.0	Basic penggunaan pyhton
9	wheel	0.36.2	Basic penggunaan pyhton
10	torch	-	Atau bisa di sebut pyTorch adalah lib yang menjalankan praktisi artificial intelligence (AI)

Cara pendeteksian terdapat dua cara yaitu :



Gambar4.2 Konfigurasi matriks dalam deteksi YOLO

Gambar 4.2 yaitu besaran tingkat YOLO mendeteksi sebuah barang, dalam pendeteksiannya bukan hanya mendeteksi manusia, namun pada case ini hanya untuk dataset manusia saja. Mendeteksi orang dan menghitung jaraknya dari titik tengah bounding box. Cara pertama untuk menghitung jarak antar orang adalah dengan menghitung jarak antar persegi panjang (kotak) yang dikembalikan oleh model, lebih tepatnya jarak antara pusatnya.

Center distance :

```
def center_distance(xyxy1, xyxy2):
    '''Calculate the distance of the centers of the boxes.'''
    a, b, c, d = xyxy1
    x1 = int(np.mean([a, c]))
    y1 = int(np.mean([b, d]))

    e, f, g, h = xyxy2
```

```

x2 = int(np.mean([e, g]))
y2 = int(np.mean([f, h]))

dist = np.linalg.norm([x1 - x2, y1 - y2])
return dist, x1, y1, x2, y2

```

Ketika kami memiliki bingkai video, kami dapat mendeteksi orang-orang di bingkai menggunakan YOLOv5x dan menggambar persegi panjang. Warna persegi panjang menunjukkan jika orang tersebut terlalu dekat dengan orang lain.

```

def detect_people_on_frame(img, confidence, distance):
    '''Detect people on a frame and draw the rectangles and lines.'''
    results = model([img[:, :, :-1]]) # Pass the frame through the
    model and get the boxes

    xyxy = results.xyxy[0].cpu().numpy() # xyxy are the box
    coordinates
    #      x1 (pixels)  y1 (pixels)  x2 (pixels)  y2 (pixels)
    confidence      class
    # tensor([[7.47613e+02, 4.01168e+01, 1.14978e+03, 7.12016e+02,
    6.71210e-01, 0.00000e+00],
    #         [1.17464e+02, 1.96875e+02, 1.00145e+03, 7.11802e+02,
    6.08795e-01, 0.00000e+00],
    #         [4.23969e+02, 4.30401e+02, 5.16833e+02, 7.20000e+02,
    7.77376e-01, 2.70000e+01],
    #         [9.81310e+02, 3.10712e+02, 1.03111e+03, 4.19273e+02,
    2.86890e-01, 2.70000e+01]])

    xyxy = xyxy[xyxy[:, 4] >= confidence] # Filter desired
    confidence
    xyxy = xyxy[xyxy[:, 5] == 0] # Consider only people
    xyxy = xyxy[:, :4]

    colors = ['green'] * len(xyxy)
    for i in range(len(xyxy)):
        for j in range(i+1, len(xyxy)):
            # Calculate distance of the centers

```

```

        dist, x1, y1, x2, y2 = center_distance(xyxy[i],
xyxy[j])
    if dist < distance:
        # If dist < distance, boxes are red and a line is drawn
        colors[i] = 'red'
        colors[j] = 'red'
        img = cv2.line(img, (x1, y1), (x2, y2), (0, 0,
255), 2)
    for i, (x1, y1, x2, y2) in enumerate(xyxy):
        # Draw the boxes
        if colors[i] == 'green':
            color = (0, 255, 0)
        else:
            color = (0, 0, 255)
        img = cv2.rectangle(img, (x1, y1), (x2, y2), color, 2)
    return img

```

Model YOLOv3

```

# load the COCO class labels our YOLO model was trained on
labelsPath = os.path.sep.join([config.MODEL_PATH, "coco.names"])
LABELS = open(labelsPath).read().strip().split("\n")

# derive the paths to the YOLO weights and model configuration
weightsPath = os.path.sep.join([config.MODEL_PATH,
"yolov3.weights"])
configPath = os.path.sep.join([config.MODEL_PATH, "yolov3.cfg"])

```

Model YOLOv5

```

# Load model
model = torch.hub.load('ultralytics/yolov5', 'yolov5x',
pretrained=True, verbose=False)
model.cuda('cuda:0');

```

Menampilkan resiko sedang dan besar yaitu :

```

# draw the total number of social distancing violations on the
output frame

text = "Total resiko tinggi: {}".format(len(serious))
cv2.putText(frame, text, (10, frame.shape[0] - 55),
cv2.FONT_HERSHEY_SIMPLEX, 0.70, (0, 0, 255), 2)

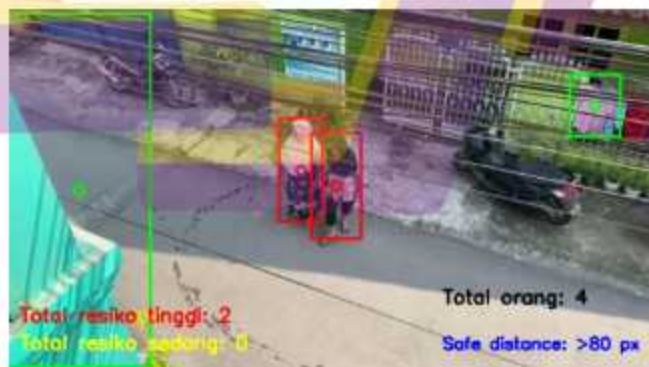
text1 = "Total resiko sedang: {}".format(len(abnormal))
cv2.putText(frame, text1, (10, frame.shape[0] - 25),
cv2.FONT_HERSHEY_SIMPLEX, 0.70, (0, 255, 255), 2)

```

Subclass yang diberikan dalam pendeteksian terdapat 3 kategori, sebagai berikut:

1. Beresiko tinggi

Pendeteksian manusia di gambarkan oleh bounding box antara satu bounding box dengan bounding box lainnya saling bertumpukan atau terkena batas bounding box maka akan dikategorikan sebagai beresiko tinggi dengan pertanda berwarna merah.



Gambar 4.3 Class beresiko tinggi

2. Beresiko sedang

Pendeteksian manusia digambarkan oleh bounding box antara satu bounding box dengan bounding box lainnya berdekatan namun tidak saling bertumpukan, dengan estimasi jarak 1 meter maka dikategorikan sebagai beresiko sedang dengan pertanda berwarna kuning.



Gambar 4.4 Class beresiko sedang

3. Tidak beresiko

Pendeteksian manusia digambarkan oleh bounding box antara satu bounding box dengan bounding box lainnya berjauhan atau lebih dari 1 meter maka dikategorikan tidak beresiko dengan pertanda berwarna hijau



Gambar 4.5 Kleas beresiko rendah

4.2 Hasil Training

Hasil dari training data pada sistem sangat mempengaruhi keberhasilan model YOLOv4 dalam mendeteksi objek. Dari proses training data, akan didapatkan parameter akurasi, presisi, recall, F1 score, IoU, dan mAP. Input dari proses ini adalah sejumlah data latih dan data validasi, sementara output dari proses ini yaitu weights file yang akan digunakan saat pengujian sistem.

Sebagian besar dataset yang digunakan diambil dari internet. Hal ini disebabkan kurang kondusifnya situasi luar untuk mengambil data langsung di transportasi publik.

Data latih dan data validasi sepenuhnya diambil dari Open Image Datasets yang dikelola oleh Sementara untuk data uji, hanya sebagian kecil yang

Training data pada skenario pertama digunakan 3000 data latih dan 600 data validasi. Data yang dilatih berasal dari kelas person, di mana sistem akan

mendeteksi orang mulai dari kepala hingga ujung kaki. Konfigurasi sistem yang diatur adalah sebagai berikut:

- | | | |
|----|----------------------|---------|
| 1. | <i>Batch size</i> | = 64 |
| 2. | <i>Sub divisions</i> | = 16 |
| 3. | <i>Channels</i> | = 3 |
| 4. | <i>Learning rate</i> | = 0,001 |
| 5. | <i>Max batches</i> | = 4000 |
| 6. | <i>Random value</i> | = 1 |

Namun, hasil dari *training* pada skenario ini tidak berhasil disebabkan oleh nilai dari *sub divisions* dan *random value* memakan memori cukup banyak saat *training* sehingga proses menjadi tersendat dan membuat program harus di-*running* berulang-ulang. Hal ini mengakibatkan grafik mAP pada training menjadi terputus-putus.

Dari *training* pada skenario pertama ini, karena proses tidak berjalan dengan lancar dan optimal maka parameter performansi yang dihasilkan pun menjadi sangat rendah.

Setelah ditelusuri, 16 *sub divisions* yang juga dapat disebut dengan *mini-batch* ini ternyata kurang mampu untuk diproses pada spesifikasi laptop yang digunakan. *Batch size* sebesar 64 dibagi oleh 16 *sub divisions* menjadi 4. Dalam kata lain, sistem akan melakukan *training* terhadap empat data latih tiap 16 *mini-batch*. Selain itu, nilai *random value* juga turut berpengaruh dalam kegagalan proses training ini

Menggunakan dataset dengan rasio rasio yang berbeda, antara data train : data test yaitu 90%:10%, 80%:20% dan 70%:30%. Dari rasio berikut akan menghasilkan training keseluruhan yaitu pada tabel dibawah ini:

Tabel 4.2. Hasil Training dataset

Dataset Rasio	Presisi	Recall	F1 Score	Average IoU	mAP
90%:10%	61%	62%	63%	44.66%	51.90%
80%:20%	35%	59%	45%	25.39%	32.34%
70%:30%	40%	55%	45%	29.59%	28.20%

4.3 Dataset

Dataset yang digunakan dalam Deteksi Pelanggaran Jarak Social Distancing ini adalah dataset yang dapat diakses oleh semua orang melalui website cocodataset.org. dimana dataset ini memiliki jumlah 120.000 gambar dengan total 880.000 objek berlabel. Sedangkan gambar manusia terdapat 66.808 gambar. Model-model ini di Training untuk memuat 80 jenis objek berbeda dalam kumpulan data ini. Untuk memulai analisis ini, pertama-tama impor pustaka dan tentukan fungsi untuk memplot data menggunakan matplotlib. Bergantung pada data, tidak semua plot akan dibuat. Dataset dalam pengujian menggunakan video.



Gambar 4.6 Yolo Dataset

```

from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
)

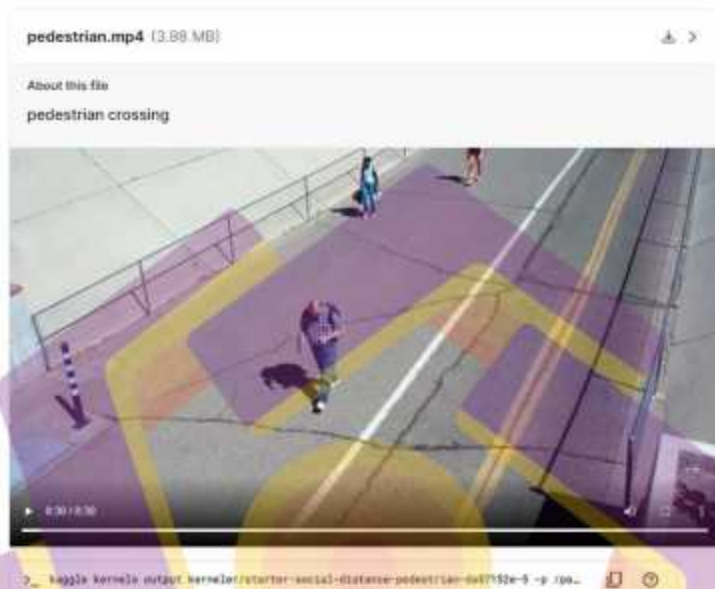
```

Ada 0 file csv dalam versi dataset saat ini:

```

for dirname, _, filenames in os.walk('/coco/dataset/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

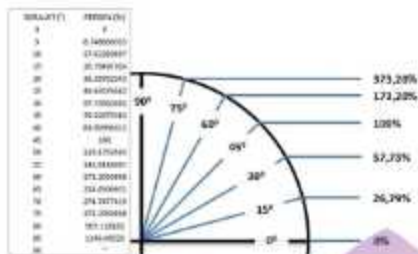
```

Data

Gambar 4. 7Public Dataset Video

4.4 Skenario Pengujian Sudut

Percobaan dilakukan pada beberapa skenario percobaan, dimana pada saat dilakukan percobaan terdapat anomali pada hasil percobaan ditemukan jarak tertentu sistem dapat bekerja dan sudut mempengaruhi sistem mendeteksi person. Dalam menguji, peneliti menggunakan beberapa skenario pengujian untuk mencari optimal maka sistem akan dilakukan pengujian sebagai berikut:



Gambar 4.8 Skenario Sudut

Pada gambar 4.8 sistem akan mendeteksi dengan berbagai variasi sudut yang disediakan yaitu 0° , 15° , 30° , 45° , 60° , 90°

Alasan peneliti mengambil sudut tersebut sebagai standarisasi sudut. Dan penelitian ini diharapkan dapat menghasilkan seberapa pengaruh peletakan kamera dengan sistem yang bekerja.

4.5 Hasil Pengujian Sudut

Melakukan penghitungan sudut menggunakan data yang diambil langsung oleh peneliti. Klasifikasi Beresiko apabila objek/orang berada pada jarak kurang dari 1 m dari yang lainnya, dan ini merupakan kesalahan. Namun jika Tidak beresiko menandakan jarak satu dengan yang lainnya yaitu lebih dari 1 m menunjukkan bahwa sesuai dengan peraturan *social distancing*. Berikut adalah hasil pengujian sudut pada sistem *social distancing* Hasil menggunakan pemrosesan data pribadi:

1. Sudut 90° :

a. Beresiko :



Gambar 4.9 Sudut 90 derajat beresiko

Hasil : Pada gambar 4.9menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko sedang dikarenakan belum banyak berinteraksi dalam 1 frame yang banyak sehingga dikategorikan sedang. Percobaan dilakukan 5 video menghasilkan data 100% berhasil.

b. Tidak Beresiko :



Gambar 4.10 Sudut 90 derajat tidak beresiko

Hasil : Pada gambar 4.10 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Percobaan dilakukan 5 video menghasilkan data 100% berhasil.

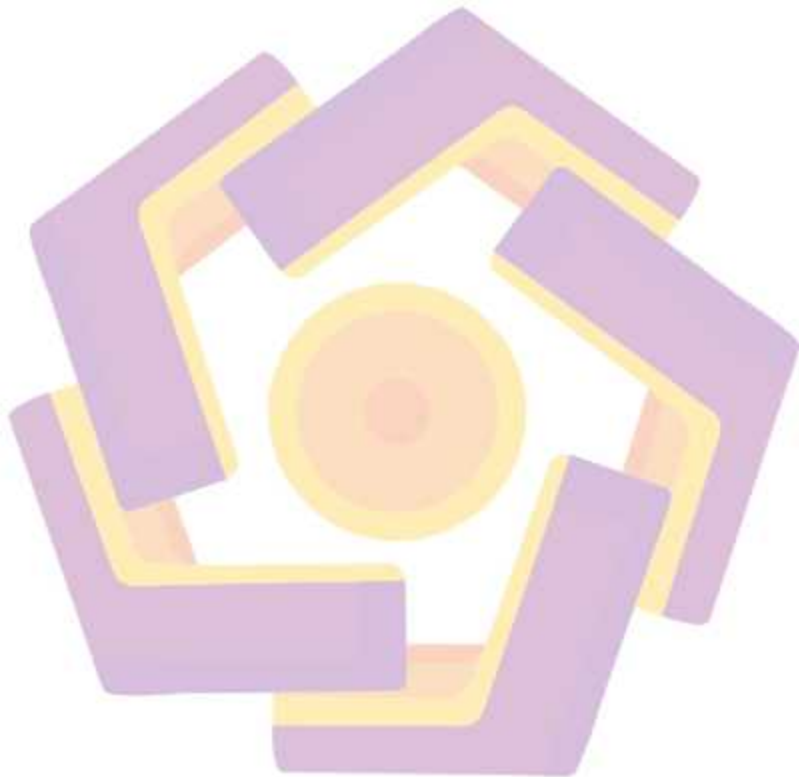
Berikut ini adalah detail percobaan untuk pengujian tingkat akurasi sebagai berikut ini:

Tabel 4.3 Perhitungan Sudut 90 derajat

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	2	2	2	Berdekatan	0
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	6	6	Berdekatan	0
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	2	2	Berjauhan	0
8	9	9	9	Berjauhan	0
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
Total	34	34	0		

Error % 0 data dengan rumus x/y menjadi $34/34$ /maka Akurasi = $100\% - 0\% = 100\%$ Berdasarkan hasil pengujian pada Tabel 4.3 Hasil Uji

Deteksi Pelanggaran Jarak Social Distancing pada sudut 90 derajat menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 100%.



2. Sudut 75° dan 60° :

a. Beresiko :



Gambar 4.11 Sudut 60 derajat bersiko

Hasil : Pada gambar 4.11 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko sedang dikarenakan belum banyak berinteraksi dalam 1 frame yang banyak sehingga dikategorikan sedang. Percobaan dilakukan 5 video menghasilkan data 60% berhasil..

b. Tidak Beresiko :



Gambar 4.12 Sudut 60 derajat tidak bersiko

Hasil : Pada gambar 4.12 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Percobaan dilakukan 5 video menghasilkan data 60% berhasil.

Berikut ini adalah detail percobaan untuk pengujian tingkat akurasi sebagai berikut ini:

Tabel 4.4 Perhitungan percobaan 75 dan 60 derajat

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	2	2	2	Berdekatan	0
2	2	1	1	Berdekatan	1
3	5	5	5	Berdekatan	0
4	6	4	4	Berdekatan	2
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	1	1	Berjauhan	1
8	9	5	5	Berjauhan	4
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
TOTAL	34	26	76%		

Error 4 data terdapat $26/34=0.76$ menjadimaka Akurasi 76% Berdasarkan hasil pengujian pada Tabel 4.4 Hasil Uji Deteksi Pelanggaran Jarak Social Distancing 75 dan 60 derajat menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 76%

3. Sudut 45⁰:

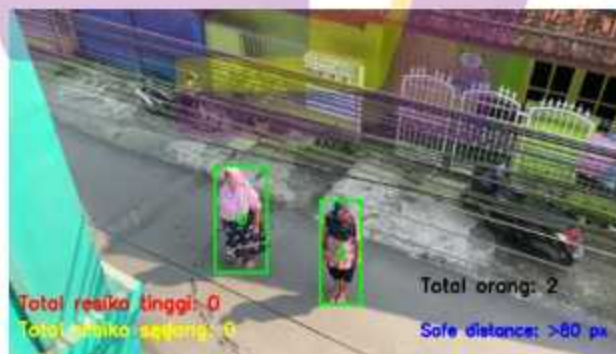
a. Beresiko :



Gambar 4.13 Sudut 45 derajat bersiko

Hasil : Pada gambar 4.13 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko sedang dikarenakan belum banyak berinteraksi dalam 1 frame yang banyak sehingga dikategorikan sedang. Percobaan dilakukan 5 video menghasilkan data 60% berhasil.

b. Tidak Beresiko :



Gambar 4.14 Sudut 45 derajat tidak bersiko

Hasil : Pada gambar 4.14 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Percobaan dilakukan 5 video menghasilkan data 80% berhasil.

Berikut ini adalah detail percobaan untuk pengujian tingkat akurasi sebagai berikut ini:

Tabel 4.5 Pengujian 45 derajat

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	2	2	2	Berdekatan	0
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	4	4	Berdekatan	2
5	2	2	1	Berdekatan	1
6	2	2	2	Berjauhan	0
7	2	2	2	Berjauhan	0
8	9	8	8	Berjauhan	1
9	2	1	2	Berjauhan	0
10	2	1	2	Berjauhan	0
TOTAL	34	24	70%		

Error total 3 terdapat 3/10 jadi maka Akurasi = $100\% - 30\% = 70\%$
 Berdasarkan hasil pengujian pada Tabel 4.3 Hasil Uji Deteksi Pelanggaran Jarak Social Distancing menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 70%..

4. Sudut 30°:

a. Beresiko :



Gambar 4.15 Sudut 30 derajat beresiko

Hasil : Pada gambar 4.15 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Percobaan dilakukan 5 video dengan keakuratan data 60%

b. Tidak Beresiko :



Gambar 4.16 Sudut 30 derajat tidak beresiko

Hasil : Pada gambar 4.16 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Dan percobaan dilakukan 5 video dengan keakurasikan data 60%

Berikut ini adalah detail percobaan untuk pengujian tingkat akurasi sebagai berikut ini:

Tabel 4.6 Perhitungan data 30 derajat

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	2	2	2	Berdekatan	0
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	4	4	Berdekatan	2
5	2	1	1	Berdekatan	1
6	2	1	1	Berjauhan	1
7	2	2	2	Berjauhan	0
8	9	5	5	Berjauhan	1
9	2	1	1	Berjauhan	1
10	2	1	1	Berjauhan	1
TOTAL	34	19	55%		

Error 6 data terdapat 19/34 menjadimaka Akurasi 55% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing 30 derajat menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 55%.

5. Sudut 15⁰:

a. Beresiko :



Gambar 4.17 Sudut 15 derajat beresiko

Hasil :Pada gambar 4.17 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. . Total orang ada 2 dan termasuk kategori resiko sedang dikarenakan belum banyak berinteraksi dalam 1 frame yang banyak sehingga dikategorikan sedang. Percobaan dilakukan 5 video dan menghasilkan 40% keakurasian data.

b. Tidak Beresiko :



Gambar 4.18 Sudut 15 derajat tidak beresiko

Hasil : Pada gambar 4.18 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori aman. Percobaan dilakukan 10 video. 8 video berhasil terdeteksi, 2 video tidak berhasil terdeteksi. Keakurasian data 60%.

Tabel 4.7 Perhitungan data 15 derajat

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	2	2	2	Berdekatan	0
2	2	1	1	Berdekatan	0
3	5	2	2	Berdekatan	3
4	6	4	4	Berdekatan	2
5	2	1	1	Berdekatan	1
6	2	2	2	Berjauhan	0
7	2	2	2	Berjauhan	0
8	9	3	3	Berjauhan	3
9	2	1	1	Berjauhan	1
10	2	0	1	Berjauhan	1
TOTAL	34	18	52%		

Error 6 data terdapat 19/34 menjadimaka Akurasi 52% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing 15 dan 0 derajat menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 52%.

Percobaan dilakukan 10 kali pengambilan video dengan berbagai macam sudut. Hasilnya pada sudut 0, 15,30 dan 45 derajat terdapat kekurangan beberapa kali jika tertutup oleh objek lain maka tidak sesuai,. Namun pada pengambilan sudut 60, 75 dan 90 derajat di hasilkan berbagai percobaan data testing menghasilkan beberapa kali YOLO mendeteksi orang serta jarak yang tepat. Hal ini menjadi tanda bahwa pengaturan sudut sangat berpengaruh pada hasil keakuratan yang di dapatkan. Skenario sudut

0° dan 15° dianggap sejajar dengan orang satu dengan lainnya hal ini akan semakin gagal mendeteksi keakurasian sistem semakin besar sudutnya.

Tingkat keberhasilan dari deteksi untuk perhitungan jarak orang adalah 100%. Dilihat dari tingkat keberhasilannya, aplikasi ini akan berfungsi dengan baik didukung dengan prosesor yang dapat menghasilkan kecepatan tinggi. Perbandingan seluruh hasil analisis yang telah dilakukan pada percobaan sebelumnya.

4.6 Skenario Pengujian Jarak

Percobaan dilakukan pada beberapa skenario percobaan diharapkan dalam penelitian ini ditemukan jarak tertentu sistem dapat bekerja dan jarak mempengaruhi sistem mendeteksi person. Dalam menguji, peneliti menggunakan beberapa skenario pengujian untuk mencari optimal maka sistem akan dilakukan pengujian sebagai berikut:



Gambar 4.19 Skenario Pengujian Jarak

Pada gambar 4.19 sistem akan mendeteksi dengan berbagai variasi sudut yang disediakan yaitu 1 meter, 2 meter, 4 meter, 6 meter, 10 meter. Alasan peneliti mengambil sudut tersebut sebagai standarisasi jarak. Dan penelitian ini diharapkan dapat menghasilkan seberapa pengaruh peletakan kamera dengan sistem yang

bekerja. Untuk keakurasian jarak, penulis menggunakan alat meteran untuk mendukung jarak yang diinginkan seperti pada gambar 4.20.



Gambar 4.20 Alat meteran

4.7 Hasil Pengujian Jarak

Melakukan penghitungan jarak menggunakan data yang diambil langsung oleh peneliti. Klasifikasi Beresiko apabila objek/orang berada pada jarak kurang dari 1 m dari yang lainnya, dan ini merupakan kesalahan. Namun jika Tidak beresiko menandakan jarak satu dengan yang lainnya yaitu lebih dari 1m menunjukkan bahwa sesuai dengan peraturan *social distancing*. Berikut adalah hasil pengujian jarak pada sistem *social distancing* Hasil menggunakan pemrosesan data pribadi:

1. Jarak 1 meter

a. Beresiko



Gambar 4.21 Jarak 1 m tidak beresiko

Hasil : Pada gambar 4.21 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 5 dan termasuk kategori resiko sedang. Percobaan dilakukan 5 video menghasilkan data 50% keberhasilan 5 sesuai dan 5 tidak sesuai secara perhitungan manual.

b. Tidak beresiko



Gambar 4.22 Jarak 1 meter tidak beresiko

Hasil : Pada gambar 4.22 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 5 video menghasilkan data 70%

Berikut ini adalah detail percobaan untuk pengujian tingkat akurasi sebagai berikut ini:

Tabel 4.8 Perhitungan data jarak 1 meter

Video ke	Jumlah Orang/manual	Jumlah orang system	by	Jumlah ketepatan terdeteksi	Kondisi	Error
1	1	1		1	Berdekatan	1
2	2	2		2	Berdekatan	0
3	5	5		5	Berdekatan	0
4	6	6		6	Berdekatan	0
5	2	2		2	Berdekatan	0
6	2	2		2	Berjauhan	0
7	2	2		2	Berjauhan	0
8	0	0		0	Berjauhan	0
9	2	2		2	Berjauhan	0
10	2	2		2	Berjauhan	0
TOTAL	34	33		97%		

Error 1 data terdapat 33/34 menjadimaka Akurat 97% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing pada 1 meter menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 97%.

2. Jarak 2 meter
 - a. Beresiko



Gambar 4. 23 Jarak 2 meter beresiko

Hasil : Pada gambar 4.23 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko sedang. Percobaan dilakukan 5 video menghasilkan data 60% Tidak beresiko



Gambar 4.24 Jarak 2 meter tidak beresiko

Hasil : Pada gambar 4.24 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 10 video menghasilkan data 100% keberhasilan 10 sesuai secara perhitungan manual.

Tabel 4.9 Perhitungan data jarak 2 meter

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Eror
1	1	1	1	Berdekatan	1
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	6	6	Berdekatan	0
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	2	2	Berjauhan	0
8	9	8	5	Berjauhan	1
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
TOTAL	34	32	94%		

Eror 2 data terdapat 32/34 menjadimaka Akurat 94% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing pada 2 meter menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 94%.

3. Jarak 4 meter

a. Beresiko



Gambar 4.25 jarak 4 meter beresiko

Hasil : Pada gambar 4.25 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko tinggi. Percobaan dilakukan 10 video menghasilkan data 100% keberhasilan 10 sesuai secara perhitungan manual.

b. Tidak beresiko



Gambar 4.26 Jarak 4 meter tidak beresiko

Hasil : Pada gambar 4.26menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 10 video menghasilkan data 100% keberhasilan 10 sesuai secara perhitungan manual.

Tabel 4.10 Perhitungan data jarak 4 meter

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	1	1	1	Berdekatan	0
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	6	6	Berdekatan	0
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	1	2	Berjauhan	0
8	0	8	5	Berjauhan	1
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
TOTAL	34	31	91%		

Eror 1 data terdapat 31/34 menjadimaka Akurat 91% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing pada 4 meter menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 91%.

4. Jarak 10 meter

a. Beresiko



Gambar 4.27 jarak 10 meter beresiko

Hasil : Pada gambar 4.27 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori

resiko rendah. Percobaan dilakukan 10 video menghasilkan data 40% keberhasilan 4 sesuai dan 6 tidak sesuai secara perhitungan manual.

b. Tidak beresiko



Gambar 4.28 Jarak 10 meter tidak beresiko

Hasil : Pada gambar 4.28 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 10 video menghasilkan data 100% keberhasilan 10 sesuai secara perhitungan manual.

Tabel 4.11 Perhitungan data jarak 10 meter

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Error
1	1	1	1	Berdekatan	1
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	6	6	Berdekatan	0
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	1	1	Berjauhan	1
8	9	8	6	Berjauhan	3
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
TOTAL	34	31	91%		

Error 3 data terdapat 31/34 menjadimaka Akurat 91% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing

pada 10 meter menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 91%.

5. Jarak 20 meter

a. Beresiko



Gambar 4.29 Jarak 20 meter beresiko

Hasil : Pada gambar 4.29 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 10 video menghasilkan data 100% keberhasilan 10 sesuai secara perhitungan manual.

b. Tidak beresiko



Gambar 4.30 Jarak 20 meter tidak beresiko

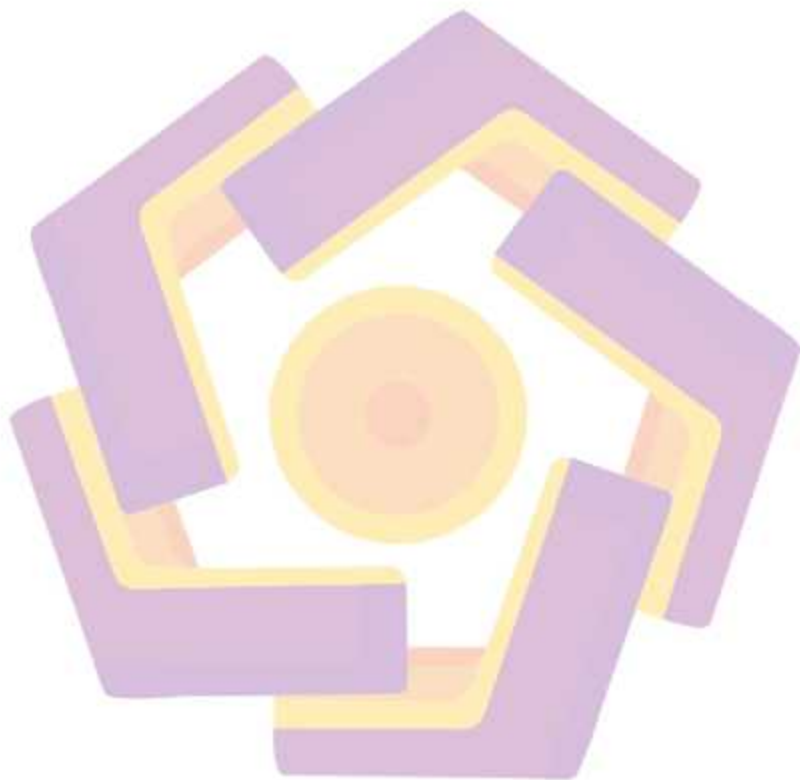
Hasil : Pada gambar 4.30 menandakan sistem berhasil mendeteksi orang serta jarak yang ditentukan. Total orang ada 2 dan termasuk kategori resiko rendah. Percobaan dilakukan 10 video menghasilkan data 90% keberhasilan 9 dan 1 tidak sesuai secara perhitungan manual.

Tabel 4.12 Perhitungan data jarak 20 meter

Video ke	Jumlah Orang/manual	Jumlah orang by system	Jumlah ketepatan terdeteksi	Kondisi	Eror
1	1	1	1	Berdekatan	0
2	2	2	2	Berdekatan	0
3	5	5	5	Berdekatan	0
4	6	5	5	Berdekatan	1
5	2	2	2	Berdekatan	0
6	2	2	2	Berjauhan	0
7	2	1	1	Berjauhan	1
8	9	9	9	Berjauhan	3
9	2	2	2	Berjauhan	0
10	2	2	2	Berjauhan	0
TOTAL	34	31	91%		

Eror 3 data terdapat 31/34 menjadimaka Akurat 91% Berdasarkan hasil pengujian pada Hasil Uji Deteksi Pelanggaran Jarak Social Distancing

pada 10 meter menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 91%.



4.8 Tabel Perbandingan

Tabel 4.13 Tabel Perbandingan

Uji ke-	Video	Sudut /jarak	Durasi waktu video (detik)	Durasi waktu deteksi (detik)	Data Error	Akurasi (%)
1		90 ⁰	3d	6d	0	100%
2		75 ⁰	3d	6d	4	76%
3		60 ⁰	3d	6d	4	76%
4		45 ⁰	4d	9d	4	70%
5		30 ⁰	4d	10d	6	55%
6		15 ⁰	4d	8d	6	52%
7		0 ⁰	3d	5d	6	52%
8		1m	4d	8d	1	97%
9		2m	3d	6d	2	94%
10		4m	3d	7d	1	91%
11		10m	3d	7d	3	91%
12		20m	3d	7d	3	91%

Data perbandingan diatas secara keseluruhan dari pengujian yang telah dilakukan menjadimaka Akurasi dengan rata-rata 85,6% Berdasarkan hasil pengujian pada Tabel 4.13 Hasil Uji Deteksi Pelanggaran Jarak Social Distancing Rata-Rata menunjukkan bahwa sistem dapat mendeteksi pelanggaran jarak social distancing dengan tingkat akurasi sebesar 85,6%, sedangkan 14,7% pelanggaran jarak social distancing tidak terdeteksi oleh sistem atau error. Tingkat 14,7% error dikarenakan metode YOLO yang tidak tepat dalam

pendeteksiannya dikarenakan metode YOLO yang beberapa kali gagal/eror dalam pendeteksiannya sehingga looping terus terjadi hingga system dapat mendeteksi manusia.



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Error (\%) = \frac{Error}{Jumlah\ pelanggaran\ perhitungan\ manual} \times 100\%$$

$$Rata - Rata\ error (\%) = \frac{\sum error (\%)}{Banyaknya\ Percobaan}$$

4.9 Real-time video

Percobaan data real time dari device laptop secara langsung, cukup akurat. Namun terdapat perlambatan video saat dijalankan. Hal ini dikarenakan oleh YOLO v3 kurangnya efektif dijalankan di device penulis, yang membuat kecepatan menurun. Namun jika dilihat dari segi validasi, sesuai bahwa seseorang berjarak lebih dari 1 meter maka tergolong tidak beresiko, namun jika seseorang dekat dari 1 meter maka akan dinyatakan beresiko sedang atau tinggi.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, maka peneliti menyimpulkan sebagai berikut:

1. Hasil *training* yang didapatkan dengan rasio 90%:10% *learning rate* 0.001 dengan nilai mAP 54.11% adalah *training* terbaik.
2. Data uji coba *testing* menggunakan data pribadi yang dimasukkan lebih tinggi dibanding dengan dataset YOLO COCO. Hasil akurasi pengujian sistem deteksi pelanggaran *social distancing* adalah 85,6% dengan sudut pengambilan 90°, 75°, 60° di antara ketiga sudut tersebut sudut 90 derajat mempunyai akurasi lebih tinggi, namun dikarenakan kesulitannya peletakan camera pada 90 derajat maka posisi yang di recommendasikan adalah 75 derajat masih dapat system mendeteksi YOLO dengan euclidean distance, namun pada sudut 0° dan 15° system sudah tidak akurat dalam pendeteksian system. Jarak pada 1-20 meter masih valid untuk mendeteksi. Jarak antara kamera tidak mempengaruhi jarak deteksi pelanggaran *social distancing*. Tingkat akurasi hasil dari penelitian ini dalam mendeteksi pelanggaran jarak *social distancing* berdasar visualisasi perspective view images 90° sebesar 85,6%, sedangkan 14,7% pelanggaran jarak *social distancing* tidak terdeteksi pada sudut 0° dan 15°. Error 14,7% dikarenakan metode YOLO tidak dapat mendeteksi manusia secara akurat menyebabkan ada benda lain yang ikut terdeteksi.

3. Hasil real time detection masih valid dari segi kecepatan pendeteksian sama seperti inputan video mengalami penurunan kecepatan 2x lebih lama dari video asli.

5.2 Saran

Terdapat saran untuk penelitian selanjutnya yaitu

1. Camera yang diujicobakan dapat lebih dari 1 model camera, seperti widecam.
2. Dengan kesimpulan 90 derajat adalah keakurasian tertinggi maka penulis memberikan saran program sistem deteksi bird's eye view 90^o, sehingga program menjadi sebuah program perspektif yang berbeda. Dengan perhitungan 0^o dan 15^o masih terjadi layering objek dimana bisa menjadi solusi pada penelitian selanjutnya.
3. Menggunakan prosesor dan device camera deep yang memadai untuk running sistem YOLO.

DAFTAR PUSTAKA

PUSTAKA BUKU

WHO Indonesia, "Coronavirus Disease Situation Report World Health Organization," 2020

Pulung Nurtantio, Andono, T.Sutojo, Muljono. 2017. Pengolahan Citra Digital. Yogyakarta: ANDI.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

M. Pawar, "The Global Impact of and Responses to the COVID-19 Pandemic," *The International Journal of Community and Social Development*, vol. 2, no. 2, pp. 111–120, 2020, doi: 10.1177/2516602620938542.

Adhinata., Faisal Dharna, Rakhmadani., Diavianto Putra., Alon Jala Tirta Segara. "YOLO Algorithm for Detecting People in Social Distancing System". *TRANSFORMATIKA*, ISSN: 1693-3656, Vol 19, No.1, July 2021, pp. 1-7.

A. H. Ahamad, N. Zaini, and M. F. A. Latip, "Person Detection for Social Distancing and Safety Violation Alert based on Segmented ROI," *Proceedings - 10th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2020*, no. August, pp. 113–118, 2020, doi: 10.1109/ICCSCE50387.2020.9204934.

J. Redmon and A. Farhadi, "Yolov3 : An incremental improvement", *arXiv Prepr. arXiv200.10934*, 2020.

Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam, "Real-Time Object Detection with Yolo" *International Journal of Engineering and Advanced Technology (IJEAT)* ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019.

Hammam, H., Asyhar, A., Wibowo, S. A., & Budiman, G. "Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video". *E-Proceeding of Engineering* :ISSN: 2355-9365, Vol 7, No.2 Agustus 2020 Page 3631.

- Hermawan, M. I., Tritoasmoro, I. I., Ibrahim, N. "Pengaturan Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Metode Yolo". *Jurnal Edukomputika*. V.811.47837. 2021
- Jupiyandi, S., Saniputra, F. R., Pratama, Y., Dharmawan, M. R., & Cholissodin, I. "Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO". *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(4), 413. <https://doi.org/10.25126/jtiik.2019>
- Tsabita Al Asshifa Hadi Kusuma, Koredianto Usman, Sofia Saidah. 2021. "People Counting for Public Transportations Using You Only Look Once Method". *Jurnal Teknik Informatika (JUTIF)*, Vol.2 No.1 Juni 2021, hlm 57-66.
- J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, realtime object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779-788, 2016, doi: 10.1109/CVPR.2016.91
- M. Junker, R. Hoch, and A. Dengel, "On the evaluation of document analysis components by recall, precision, and accuracy," *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 717-720, 1999, doi: 10.1109/ICDAR.1999.791887.
- B. Benfold and I. Reid, "Stabel multi-target tracking in real-time surveillance video," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3457-3464, 2011, doi: 10.1109/CVPR.2011.5995667.
- Fardiasyah., Adam Fahmi, Nur Iriawan dan Wiwiek Setya Winahju. "Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3". *Jurnal Sains dan Seni ITS*. Vol.10 No.1 (2021), 2337-3520(2301-928X Print)
- M. Nishom, "Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square", *Jurnal Informatik: Jurnal Pengembangan IT (JPIT)*, Vol.04, No.01, Januari 2019, pp. 20-21, 2019.
- Xiaoyue Jiang, Abdenour Hadid, Yanwei Pang, Eric Granger, Xiaoyi Feng, "Deep Learning in Object Detection and Recognition", pp. 1-3, 2019.
- Adarsh, P., Rathi, P., Kumar, M., YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020 687-694.

PUSTAKA LAPORAN PENELITIAN

Kurniawan, M. P., 2011, Teknologi Motion Capture dengan Multi Kamera pada Pembuatan Animasi 3D, Tesis, S2 Teknik Informatika, STMIK AMIKOM, Yogyakarta

