UNIVERSITAS MUHAMMADIYAH SIDOARJO

UMSIDA
DARI SINI PENCERAHAN BERSEMI

Implementasi Algoritma Yolo V4 Untuk Deteksi Kelalaian Penggunaan Prokes Melalui Frame Vidio

Oleh:

Muhammad Chayiya Zumma,

Rohman Dijaya

Progam Studi Informatika

Universitas Muhammadiyah Sidoarjo

Juli, 2025

Pendahuluan

- **Latar Belakang**
- **Pengolah Citra Digital**
- **Google Colab**
- **Yolo v4**

Pendahuluan

- **Latar Belakang**

Virus Corona atau *Severe Acute Respiratory Syndrome Coronavirus* 2 (SARS-CoV-2) adalah virus yang menyerang sistem pernapasan. Penyakit yang disebabkan oleh virus ini disebut Covid-19 merupakan penyakit yang dapat menyebabkan gangguan pada sistem pernapasan, infeksi paru-paru, hingga kematian [1]. Kondisi pandemi yang disebabkan oleh Covid-19 ini menyebabkan pemenuhan dalam Hak Ekonomi, Sosial, dan Budaya terganggu. Salah satu kewajiban dari negara yaitu turut serta dan ikut campur dalam menjamin serta melindungi Hak Ekosob sebagaimana tertuang dalam konstutitusi UUD 1945. Baik melalui peraturan perundangan dan aturan turunan terkait hak tersebut, maupun peran dan perilaku penegak hukum berandil besar pada penyelesaian masalah yang disebabkan wabah ini [2]. Adapum kebijakan yang dikeluarkan olek pemerintah Indonesia sendiri yaitu protokol kesehatan 5M yang terdiri dari : mencuci tangan, memakai masker, mejaga jarak, menjauhi kerumunan dan mengurangi mobilitas.

Dengan diterapkannya prokes 5M yang mengharuskan masyarakat mematuhi protokol kesehatan yang telah ditetapkan oleh pemerintah yang mengharuskan masyarakat untuk mematuhinya, akan tetapi tidak jarang juga banyak masyarakat yang melanggar protokol kesehatan. Selain banyaknya masyarakat yang melanggar atau mengabaikan protokol kesehatan, pejabat negara ataupun pemerintah daerah nampaknya melakukan pelanggaran protokol kesehatan yang telah ditetapkan oleh pemerintah pusat melalui kementerian kesehatan dan gugus tugas penanggulangan covid-19 [3].

Pendahuluan

- **Latar Belakang**

Oleh sebab itu untuk meminimalisir adanya pelanggaran protokol kesehatan kita bisa menggunakan teknologi Pengolahan citra digital, Pengolahan citra digital (Digital Image Processing) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam)[4]. Pengolahan citra digital memiliki beberapa kelebihan, yaitu murah, cepat, dan tidak merusak sampai yang diukur dan mampu mengidentifikasi fisik produk secara obyektif [5].

Menjawab permasalahan diatas, pada penelitian kali ini akan dikembangkan sebuah sistem untuk mendeteksi kelalaian protokol kesehatan salah satunya yaitu tidak menggunakan masker secara real time menggunakan virtual machine Google Colab dan mengunakan algoritma YOLO (You Only Look Once) yang dimana cara kerja sistem menerima input berupa video real time melalui kamera, setelah itu sistem akan melakukan deteksi. Apabila sistem mendeteksi adanya kelalaian protokol kesehatan yaitu tidak menggunakan masker, maka sistem akan mengirim notifikasi ke Bot Telegram bahwa terdapat pelanggaran prokes.

Pendahuluan

- **Pengolah Citra Digital**

Pengolahan Citra Digital (Digital Image Processing) adalah bidang ilmu yang berfokus pada berbagai teknik untuk memanipulasi dan menganalisis gambar. Dalam konteks penelitian ini, "citra" atau "gambar" yang dimaksud secara spesifik mengacu pada gambar statis yang diambil dari sensor visual, seperti webcam.

Pengolahan citra merupakan sebuah bentuk pemrosesan sebuah citra atau gambar dengan proses numerik dari gambar tersebut, dalam hal ini yang diproses adalah masing-masing pixel atau titik dari gambar tersebut. Salah satu teknik pemrosesan citra memanfaatkan komputer sebagai peranti lunak memproses masing-masing pixel dari sebuah gambar. Oleh karena itu muncul istilah pemrosesan citra secara digital atau digital image processing [6].

Pendahuluan

- **Google Colab**

Google Colab atau *Google Colab*oratory adalah layanan yang dikembangkan oleh Google berbasis *Cloud*, *Environment* yang digunakan oleh *Google Colab* adalah *Jupyter Notebook* dengan ekstensi file *.ipynb. *Google Colab* menyediakan processor dengan spesifikasi yang tinggi (GPU dan TPU) yang bertujuan untuk memudahkan para pengguna yang menjalankan program yang membutuhkan spesifikasi tinggi secara online.

Dari sisi perangkat lunak Google Colab telah menyiapkan hampir sebagian besar pustaka (library) yang dibutuhkan. Colab sangat membantu karena sudah menyediakan sebagian besar pustaka (library) yang dibutuhkan, sehingga proses instalasi menjadi jauh lebih sederhana. Bahkan seluruh versi disediakan, misalnya TensorFlow versi 1.x maupun 2.x tersedia, begitu juga versi Python yang mulai dari versi 2.x hingga 3.x.[7]

Pendahuluan

- **Yolo v4**

YOLO *(You Only Look Once)* adalah sebuah pendekatan baru untuk sistem pendeteksian objek, yang ditargetkan untuk pemrosesan secara *real-time*. YOLO membingkai pendeteksian objek sebagai masalah regresi tunggal, dimana dari piksel gambar langsung ke kotak pembatas *(bounding box)* spasial yang terpisah dan probabilitas kelas yang terkait. YOLO melakukan pendeteksian dan pengenalan objek dengan sebuah jaringan syaraf tunggal *(single neural network)*, yang memprediksi kotak-kotak pembatas dan probabilitas kelas secara langsung dalam satu evaluasi [8]. Bagaimana YOLO Memproses Gambar

YOLO *(You Only Look Once)* bekerja dengan cara yang cukup cerdas. Pertama, gambar yang dimasukkan akan dibagi menjadi sebuah *grid*, misalnya berukuran S×S sel. Anggap saja seperti membagi gambar menjadi banyak kotak kecil. Setiap *grid cell* memprediksi B bounding box dan *confidence score* dari tiap kotak, *confidence score* inilah yang merefleksikan seberapa tingkat kepercayaan model bahwa objek di dalam kotak berupa objek yang diprediksikan[9].

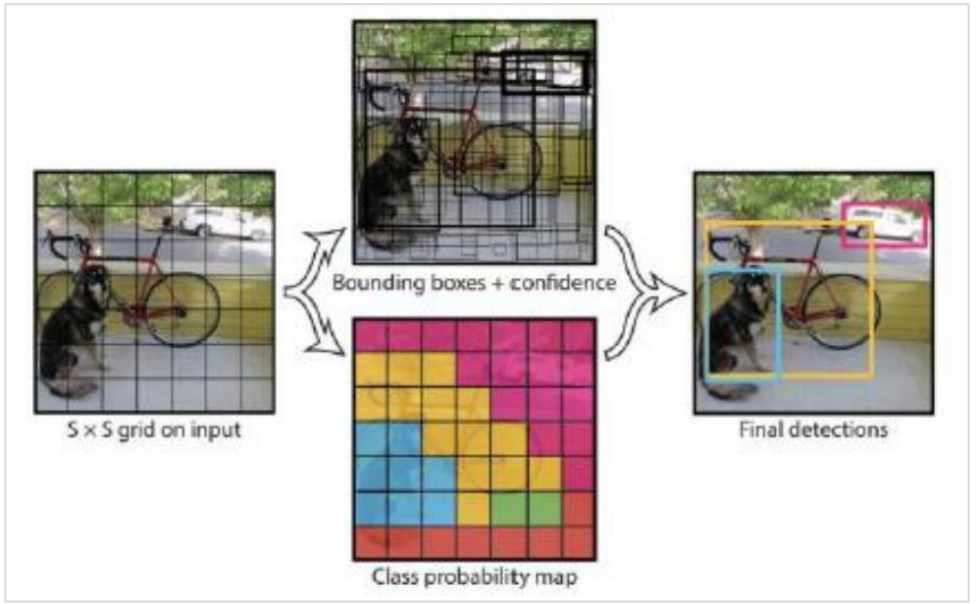
Pendahuluan

- **Yolo v4**

Secara spesifik, YOLO memproses gambar dengan dimensi awal 416×416 piksel. Melalui serangkaian operasi di lapisan konvolusional dengan 32 kanal, ukuran gambar tersebut secara progresif diperkecil hingga menghasilkan peta fitur (feature map) dengan dimensi 13×13. Peta fitur ini kemudian menjadi representasi ringkas dari informasi penting dalam gambar, yang akan digunakan oleh lapisan selanjutnya untuk deteksi objek. Dengan menggunakan kotak jangkar, eksperimen algoritma ini juga dapat meningkatkan akurasi. Eksperimen YOLO hanya dapat memprediksi 98 blok piksel per citra tetapi dengan bantuan tersebut dapat memperkirakan lebih dari 1000 citra [10].

Pendahuluan

- **Yolo v4**

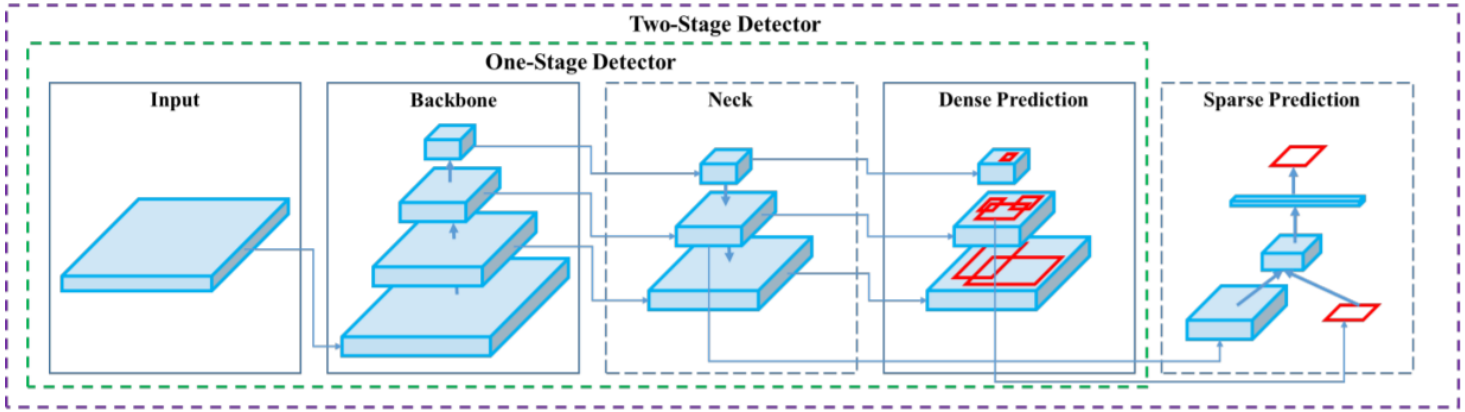


Gambar 1. Cara kerja Yolo

Pendahuluan

- **Yolo v4**

YOLOv4 merupakan versi asli terbaru yang menambahkan sejumlah implementasi tambahan yang meningkatkan akurasi dan efisiensi pendeteksian seperti Weighted Residual Connection , Cross Stage Partial Connections (CSP), Cross Mini-Batch Normalization , Self Adversarial Training, Mosaic Data Augmentation, Drop Block Regularization, dan CioU loss. Arsitektur YOLO v4 menggunakan arsitektur backbone CNN CSPDarknet53 [11]. YOLO v4 berjalan dua kali lebih cepat dari EfficientDet dengan performa yang sama. Dibandingkan dengan YOLOv3, AP dan FPS masing-masing meningkat 10% dan 12%. YOLOv4 memiliki arsitektur seperti gambar 2,



Gambar 2. Arsitektur Yolo V4

Pendahuluan

- **Yolo v4**

Detektor modern biasanya terdiri dari dua bagian, yaitu Backbone yang dilatih di ImageNet sebelumnya dan Head (Neck) yang digunakan untuk memprediksi kategori dan kotak pembatas setiap objek. Untuk pendeteksi performa GPU, jenis backbone yang tersedia adalah VGG, ResNet, dan ResNeXt. Jika hanya CPU yang digunakan, jenis backbone yang tersedia adalah SqueezeNet, MobileNet atau ShuffleNet. Adapun kepala dapat dibagi menjadi dua bagian, yaitu detektor primer dan detektor sekunder. One Stage Detector bisa juga disebut Prediksi Padat, seperti YOLO, SSD dan RetinaNet, dan Two Stage Detector bisa juga disebut Prediksi Jarang, seperti Fast R-CNN, Faster R-CNN dan seri lainnya. Pada dasarnya, Neck adalah bagian dari Backbone dan digunakan untuk meningkatkan ketangguhan fitur yang digunakan. Contoh Neck ini adalah Feature Pyramid Network (FPN), Path Aggregation Network (PAN), BiFPN dan NAS-FPN.[12]

Pertanyaan Penelitian (Rumusan Masalah)

Berdasarkan latar belakang tersebut maka dapat ditarik rumusan masalah sebagai berikut, bagaimana mengimplementasikan algoritma YOLO V4 untuk mendeteksi kelalaian protokol kesehatan penggunaan masker, bagaimana mengembangkan program komputer mendeteksi kelalaian protokol kesehatan penggunaan masker.

Tujuan

Adapun tujuan dari penulisan program ini adalah sebagai berikut, mengimplementasikan algoritma YOLO untuk mendeteksi kelalaian protokol kesehatan penggunaan masker, yanng kedua yaitu mengembangkan program komputer mendeteksi kelalaian protokol kesehatan penggunaan masker

Batasan Masalah

Adapun batasan masalah yang ditentukan sesuai dengan topik yang diambil dalam penelitian ini yaitu "Implementasi Algoritma Yolo V4 Untuk Deteksi Kelalaian Penggunaan Prokes Melalui Frame Vidio" maka penulis memiliki batasan masalah sebagai berikut : Perangkat lunak ini dikembangkan menggunakan virtual machine Google Colab, Pengambilan objek dilakukan di dalam ruangan dan luar ruangan, Mendeteksi kelalaian penggunaan protkol kesehatan yaitu kelalaian penggunaan masker.

Metode

Perancangan sistem deteksi kelalaian protokol kesehatan penggunaan masker digunakan untuk memberikan gambaran secara umum mengenai proses awal hingga mampu menyelesaikan permasalahan yang telah dibuat. Metode Analisa Sistem digunakan pada perencanaan dan pembuatan program komputer sebagai klasifikasi citra untuk mendeteksi kelalaian penggunaan prokes menggunakan metode training meggunakan pendekatan Algorima Yolo V4. Selanjutnya dilakukan *preprocessing* untuk mempersiapkan data yang digunakan untuk melakukan *training* menggunakan Algoritma Yolo V4. Untuk data yang digunakan dalam proses *preprocessing* yaitu sebanyak 300 data, yang terbagi menjadi 2 yaitu data orang yang tidak menggunakan masker dan data orang yang menggunakan masker. Setelah data terkumpul, langkah selanjutnya yaitu melakukan proses *training*. Pada proses *training* ini data akan dilatih untuk mendapatkan nilai koordiat pada data. Setelah itu akan dilakukan akan dilakukan tahap pengujian pada data yang sudah dilakukan *training* untuk melihat seberapa akurat data dalam mengklasifikasi dan mendeteksi objek. Setelah dilakukan pengujian, langkah selanjutnya yaitu mengimplementasikan data tersebut kedalam sistem deteksi kelalaian penggunaan masker.

Metode

- **Analis Kebutuhan**
- **Hasil Analisa**
- **Rancangan Sistem**
- **Preprocessing Data**
- **Proses Training Data**
- **Implementasi**

Metode

- **Analis Kebutuhan**

Untuk menggunakan sistem ini dibutuhkan beberapa data yang perlu digunakan yaitu. Mengumpulkan data citra orang yang menggunakan masker dan tidak menggunakan masker, diperlukan data gambar orang yang menggunakan masker dan tidak menggunakan masker. Data latih yang digunakan dalam penelitian ini adalah dataset publik yang berformat (.jpg) yang bisa di unduh dari internet. Setelah itu data akan diberi pelabelan denga fomat YOLO dan akan disimpan dengan extensi (.txt).

Setelah pengumpulan dan pelabelan data, langkah selanjut yang perlu dilakukan adalah melatih data yang sudah diberi label, disini peneliti menggunakan *Google Colab* untuk melatih data dan *Yolo v4* di gunakan untuk mendeteksi dan mengklasifikasi objek mana yang menggunakan masker dan tidak menggunakan masker. Langkah selanjutnya yaitu melakukan pengujian data, pengujian data dilakukan secara *realtime* menggunakan *webcam.* Setelah itu membuat pengembangan sistem yaitu mengirim notifikasi ke dalam telegram. Apabila terdapat objek yang tidak menggunakan masker maka sistem akan mengirim notifikasi ke dalam telegram bahwa terdapat objek yang melanggar protokol kesehatan tidak menggunakan masker.

Metode

- **Hasil Analisa**

Berdasarkan hasil analisa yang telah didapat diatas maka dapat digunakan sistem pengolahan citra digital yang dapat mengidentifikasi kelalaian penggunaan prokes salah satunya yaitu penggunaan masker menggunakan algoritma YOLO *(You Only Look Once)* membutuhkan operating system seperti *Windows* dan *Machie learning Google Colab* . Untuk kebutuhan sistem yang dibutuhkan dalam inputan sistem ini adalah inputan berupa gambar yang sudah dilakukan training untuk meningkatkan keakuratan dalam mendeteksi dan mengklasifikasifikasikan orang yang memakai masker dan tidak memakai masker.

Metode

- **Rancangan Sistem**

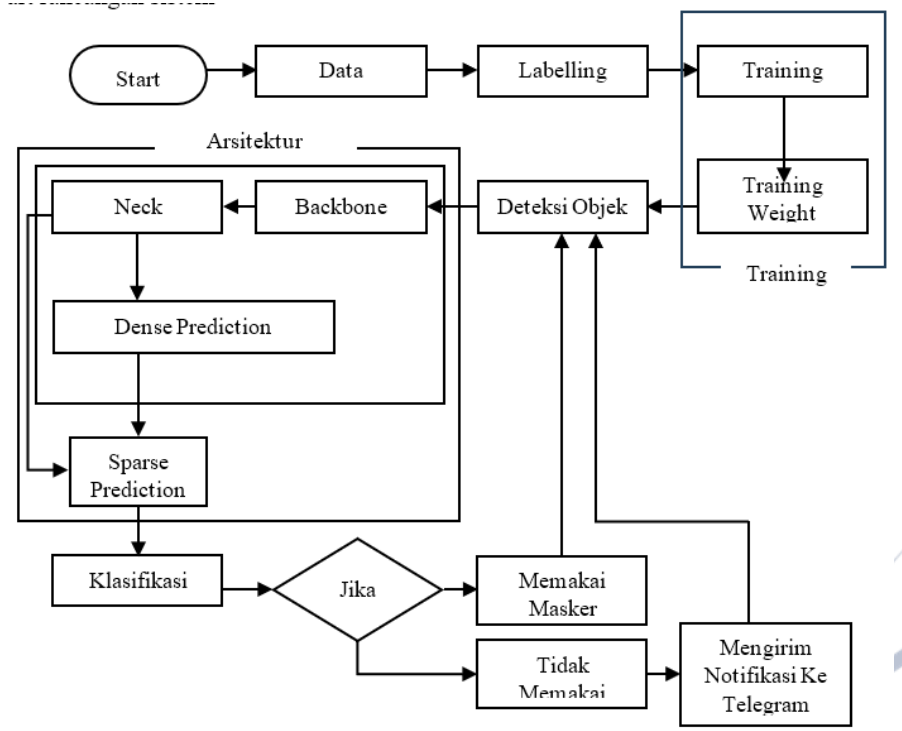
Perancangan sistem deteksi kelalaian penggunaan prokes digunakan untuk memberikan gambaran mengenai proses awal hingga mampu menyelesaikan permasalahan yang telah dibuat.

1. Flowchart rancangan sistem
2. Alur kerja sistem

Metode

- **Rancangan Sistem**

Analisa sistem digunakan pada perencanaan dan pembuatan software Deteksi Kelalaian Penggunaan Prokes Melalui Frame Vidio menggunakan Algoritma YOLO. Kelalaian penggunaan prokes dikelompokkan menjadi 2 kelas yaitu dengan menggunakan masker dan tidak menggunakan masker, dengan proses deteksi mulai dari menginputkan data, *Pre-Processing*, *Training*, Pengembangan Sistem, Hasil. Rancangan sistem dimulai dari input data yaitu mencari data citra atau gambar untuk dilakukan *labelling*, setelah menemukan data yang dicari, langkah selanjutnya yaitu memberi anotasi atau label pada gambar untuk mendapatkan sebuah pola atau bobot. Setelah memberi label pada gambar Langkah selanjutnya yaitu melakukan training pada gambar yang sudah didapatkan bobotnya untuk mendapatkan file *weight* yang akan digunakan untuk mendeteksi objek nantinya.Selanjutnya akan dilakukan deteksi objek, sistem akan melakukan deteksi objek menggunakan *algoritma YoloV4*. Setelah sistem melakukan deteksti menggunakan Yolo maka akan medapatkan hasil klasifikasi, apabila terdapat objek yang tidak memakai masker maka akan mengirim notifikasi ke telegram.

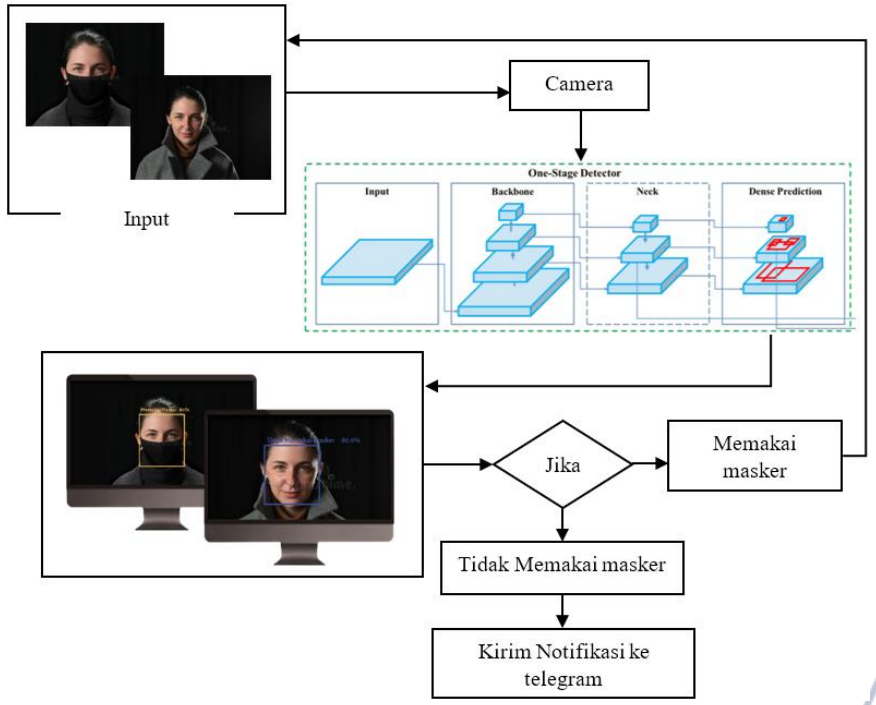


Gambar 3. Flowchart rancangan sistem

Metode

- **Rancangan Sistem**

Pada gambar 4 digambarkan sebuah alur kerja sistem yang dimulai dari input, input disini adalah seorang yang menggunakan masker atau tidak menggunakan masker. Setelah itu sebuah webcam akan menangkap citra orang yang memakai atau tidak memakai masker setelah itu sistem akan mendeteksi objek, apakah objek tersebut memakai masker atau tidak memakai masker. Apabila terdapat objek yang tidak menggunakan masker maka sistem akan mengirim notifikasi ke telegram bahwa terdapat objek yang tidak memakai masker.



Gambar 4. Flowchart alur kerja sistem

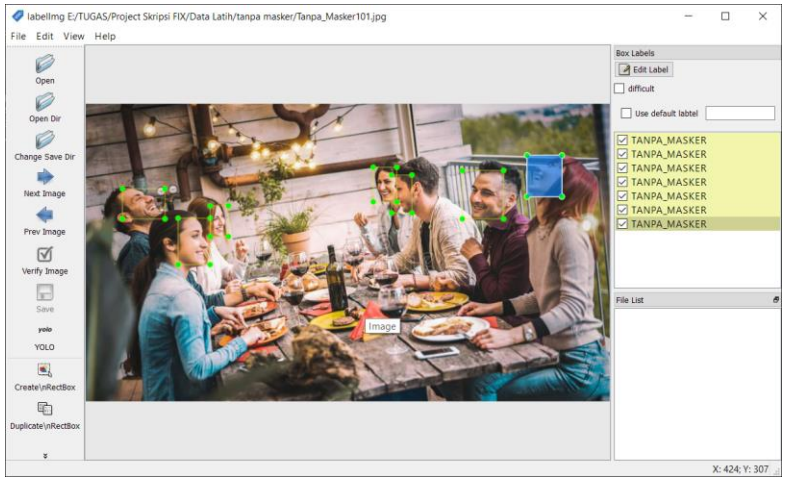
Metode

- **Preprocessing Data**

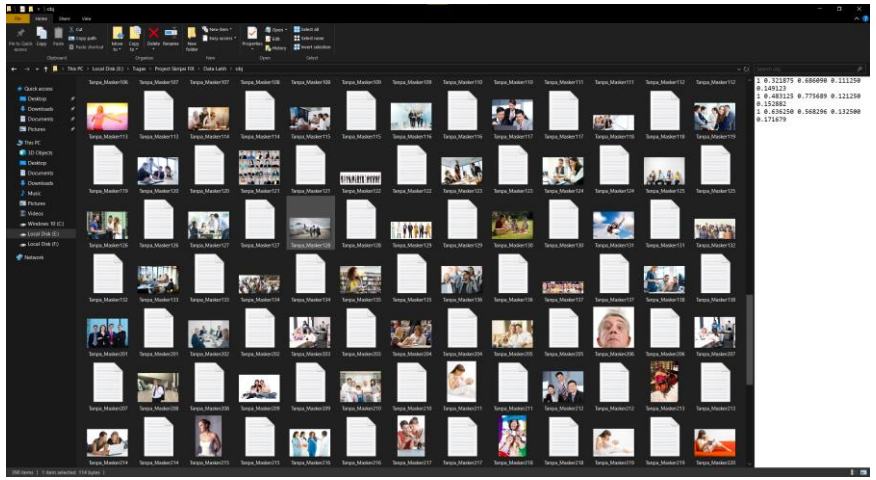
Langkah awal yang dilakukan dimulai dengan melakukan pengumpulan data orang yang meggunakan masker dan tidak menggunakan masker. Kemudian dilakukan pelabelan pada data untuk mendapatkan nilai melalui proses labelling, pelabelan data disini menggunakan aplikasi *labelimg.exe* dapat dilihat pada gambar 5. Data latih yang digunakan oleh peneliti adalah gambar diambil dari website dengan ketentuan orang menggunakan masker dan tidak menggunakan masker. Data yang digunakan sebanyak 300 data latih yang dibagi menjadi 2 kelas yaitu kelas data menggunakan masker dan kelas data tidak menggunakan masker. Setelah melakukan pelabelan maka akan didapat file (.txt) yang berisi koordinat pixel pada gambar, dapat dilihat pada gambar 6. Setelah dilakukan *labelling*. Dataset yang sudah dilabeli akan dilatih sehingga membentuk sebuah pola yang hasilnya berbentuk Bobot. Bobot tersebut akan digunakan untuk mendeteksi objek di dalam citra. Training akan dilakukan menggunakan You Only Look Once (YOLO) dimana metode ini berbasis CNN.[13] dan file gambar dijadikan satu menjadi satu folder yaitu di file *"obj.zip"*, contoh dapat dilihat pada gambar 7.

Metode

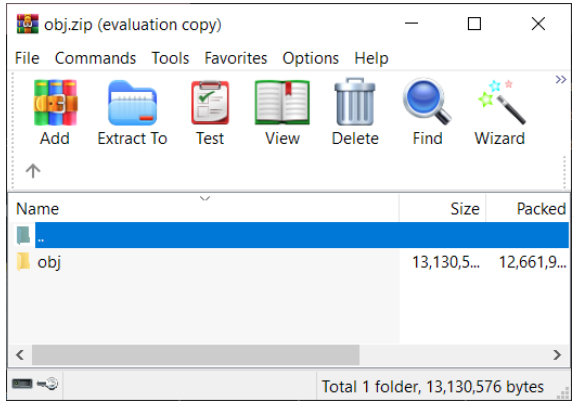
- **Preprocessing Data**



Gambar 5. *Labelling* Data



Gambar 6. Data yang sudah diberi label



Gambar 7. File ZIP *"obj.zip"*

Metode

- **Proses *Training* Data**

Setelah didapat hasil *labelling* dari kelas-kelas objek yang dideteksi, hasil *labelling* tersebut akan diteruskan menuju proses training. Proses ini dilakukan untuk melatih machine learning dengan cara mengolah gambar dan *labelling* data yang sudah dibuat sehingga terbentuk suatu karakteristik dari masing-masng kelas yang akan menjadi Training Weight atau bahan pertimbangan untuk mendeteksi objek. Sebelum dilakukan proses training, sebelumnya akan dilakukan proses upload data yaitu mengupload semua file yang dibutuhkan untuk melakukan training yaitu file *obj.zip, yolov4-custom.cfg, obj.data*, *obj.names* dan *process.py*. Setelah melakukan proses download file *pre-training.weight,* dapat dilihat pada gambar 10. File ini akan dijalankan untuk mendapatkan file *"test.txt"* dan file *"train.txt"*, dapat dilihat pada gambar 11 setelah itu dilakukan proses training.

Metode

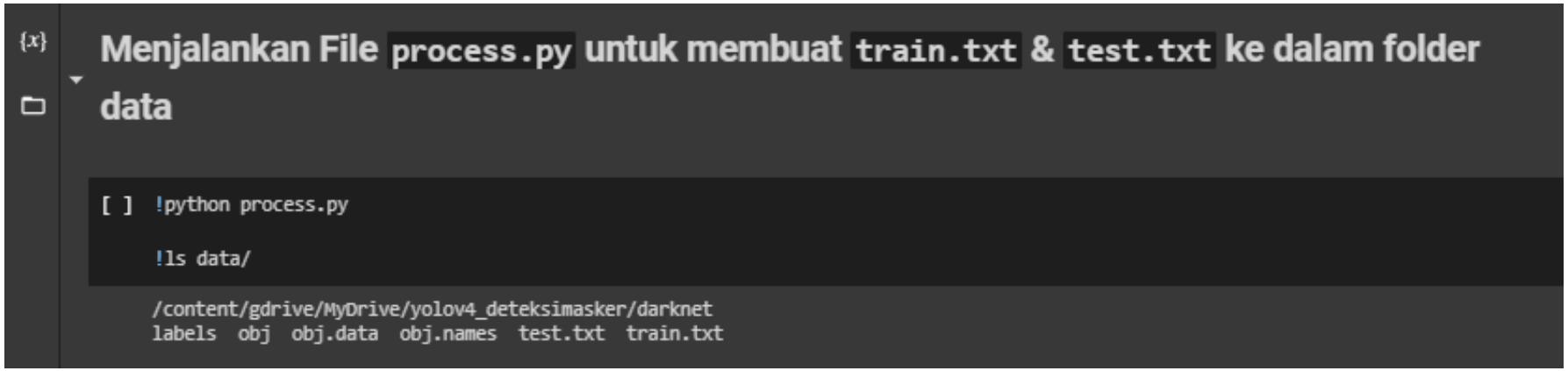
- **Proses *Training* Data**

```
[ ] # Download yolov4 pre-trained weights file

    !wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137

    --2022-01-25 21:56:09--  https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137
    Resolving github.com (github.com)... 140.82.113.3
    Connecting to github.com (github.com)|140.82.113.3|:443... connected.
    HTTP request sent, awaiting response... 302 Found
    Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/75388965/48bfe500-889d-11ea-819e-c4d182fcf0db?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential
    --2022-01-25 21:56:09--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/75388965/48bfe500-889d-11ea-819e-c4d182fcf0db?X-Amz-Algorithm=AWS4-HMAC-SHA256&
    Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.110.133, ...
    Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.108.133|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 170038676 (162M) [application/octet-stream]
    Saving to: 'yolov4.conv.137'

    yolov4.conv.137     100%[==================>] 162.16M  70.8MB/s    in 2.3s

    2022-01-25 21:56:11 (70.8 MB/s) - 'yolov4.conv.137' saved [170038676/170038676]
```

Gambar 10. Download file pre-training

Metode

- **Proses *Training* Data**



Gambar 11. Proses mendapatkan file “test.txt” dan file “train.txt”

Metode

- **Proses *Training* Data**

Proses training ini dilakukan menggunakan algoritma Yolo v4 dengan Framework Neural Network Darknet sebagai load model. Pada proses training ini jumlah batch sangat menentukan jumlah gambar yang akan diproses sebelum network weight. Setelah itu jumlah subdivision digunakan untuk memproses sebagian kecil batch size pada GPU. Selanjutnya max batches, max batches merupakan batas iterasi dalam proses training yang dapat didapatkan dengan persamaan, ketika iterasi mencapai angka yang ditentuka maka secara otomatis proses training akan berhenti. Nilai max batches dapat ditentukan dengan :

max batches = jumlah kelas × 2000

steps = (80% max batches),(90% max batches)

filters = (jumlah kelas+5) ×3

Width dan *Height* merupakan dimensi gambar masukan yang akan dilatih dan *classes* merupakan kelas yang ingin kita deteksi.

Metode

- **Proses *Training* Data**

Tabel 1. Konfigurasi file yolov4-custom.cfg

Jenis Konfigurasi	Keterangan
Batch	64
Subdivision	16
Max Batches	6000
Steps	4800,5400
Width	416
Height	416
Classes	2
Filters	21

Metode

- **Implementasi**

Setelah melakukan proses training dilakukan proses selanjutnya yaitu mengimplementasikan program pada webcam sebagai simulasi untuk mendapatkan citra waja secara *realtime*. Citra wajah yang di deteksi oleh sistem akan diperoses menggunakan *algoritma yolo v4* untuk mendeteksi adanya kelalaian penggunaan masker. Apabila terdeteksi adanya objek yang tidak memakai masker maka sistem akan mengirim notifikasi ke Bot telegram yang sudah dibuat bahwa terdapat kelalaian penggunaan masker. Semua skenario untuk pengimplementasian ini dilakukan di *Google Colab*.

Hasil dan Pembahasan

Pengujian sistem dilakukan secara realtime menggunakan kamera externan agar mendapatkan tangkapan gambar yang lebih bagus. Pengujian dilakukan dengan 5 skenario pengujian yaitu. Skenario pertama pengujian tes performance file yang digunakan untuk mengklasifikasi objek. Skenario kedua pegujian deteksi objek yang tidak memakai masker dan memakai masker. Skenario ketiga pengujian berdasarkan jarak terdekat dan terjauh. Skenario keempat pengujian berdasarkan intensitas pancarana cahaya. Dan skenario kelima pengujian sistem dapat mengirim notifikasi ke telegram.

Hasil dan Pembahasan

- **Pengujian tes performance file yang digunakan untuk mengklasifikasi objek**
- **Pegujian deteksi objek yang tidak memakai masker dan memakai masker**
- **Pengujian Berdasakan Jarak**
- **Pengujian Berdasarkan Intensitas Pancaran Cahaya**
- **Pengujian sistem dapat mengirim notifikasi ke telegram**

Hasil dan Pembahasan

- **Pengujian tes performance file yang digunakan untuk mengklasifikasi objek**

Setelah dilakukan proses *training* sebanyak 6000 iterasi dengan data latih 300 data latih objek menggunakan masker dan objek yang tidak meggunakan masker. Disini peneliti menggunakan file *"yolov4-custom_2000.weights"* untuk di uji. Untuk hasil tes dengan ID menggunakan masker mendapat nilai AP 100%, dengan nilai TP 11 dan nilai FP 0, dapat dilihat pada tabel 2. Sedangkan untuk hasil tes denga ID Tanpa Masker mendapatkan nilai AP 99.82% dengan nilai TP 13 dan nilai FP 1, dapat dilihat pada tabel 3. Untuk hasil tes secara keseluruahan pada file *"yolov4-custom_2000.weights"* mendapatkan hasil dengan *precision* 0.98 yang berarti semakin kecil nilai *precision* maka akan semakin akurat sistem dalam mendeteksi. Untuk *recall* medapatakan nilai 1.00, dan TP 44 dengan FP 1 dan FN 0. Untuk *average* IoU sendiri mendapat nilai 74.26 % yang apabila presentase *average* IoU maka akan semakin akurat pula dalam mendeteksi, untuk *average precision* (mAP@0.50) mendapat nilai 99.91 %. Sedangkan *Total Detection Time* yang didapatkan mendapat waktu rata-rata sekitar 1 detik untuk mendeteksi, dapat dilihat pada table 4.

Hasil dan Pembahasan

- **Pengujian tes performance file yang digunakan untuk mengklasifikasi objek**

No	Hasil	Skor
1	AP	100.00%
2	TP	11
3	FP	0

Tabel 2. Hasil tes ID Masker

No	Hasil	Skor
1	AP	99.82%
2	TP	33
3	FP	1

Tabel 3. Hasil tes ID Tanpa Masker

Hasil dan Pembahasan

- **Pengujian tes performance file yang digunakan untuk mengklasifikasi objek**

No	Hasil	Skor
1	*precision*	0.98
2	*recall*	1.00
3	TP	44
4	FP	1
5	FN	0
6	*average* IoU	74.26 %
7	*average precision* (mAP@0.50)	99.91 %
8	*Total Detection Time*	1 *Seconds*

Tabel 4. Hasil performa model YoloV4

Hasil dan Pembahasan

- **Pengujian tes performance file yang digunakan untuk mengklasifikasi objek**

True Positive (TP),True positive mewakili objek yang terdeteksi dan diprediksi benar. *False Positive* (FP), *False Positive* mewakili objek yang terdeteksi namun prediksi salah. *False Negative* (FN), *False Negative* mewakili objek yang tidak terdeteksi dan objek yang bukan menggunakan masker dan menggunakan masker namun terdeteksi sebagai objek menggunakan masker dan menggunakan masker. *Precision*, *Precision* merupakan rasio antara objek yang diprediksi benar berbanding keseluruhan hasil yang diprediksi positif oleh sistem.[14]

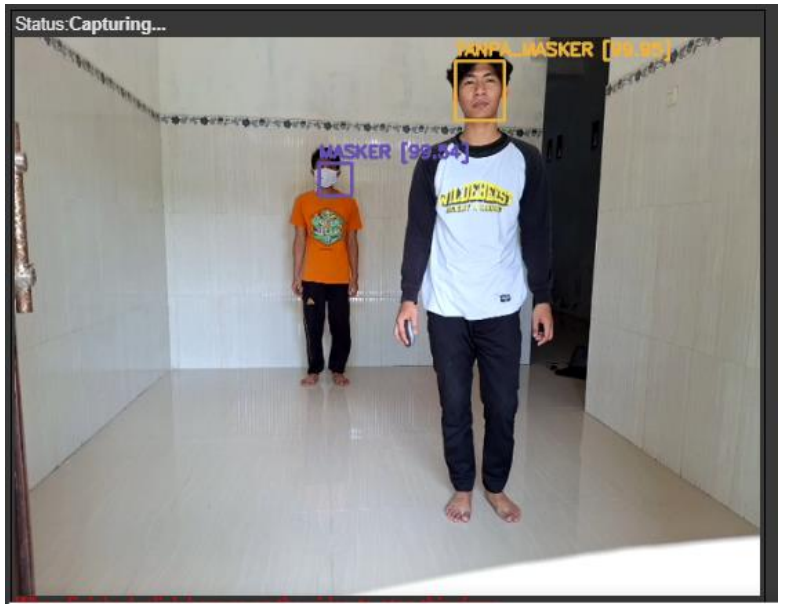
$Precision = \frac{TP}{(TP+FP)}$. *Recall*, *Recall* merupakan rasio antara objek yang diprediksi benar berbanding keseluruhan hasil yang sebenarnya $Recall = \frac{TP}{(TP+FN)}$. *Accuracy*, *Accuracy* merupakan rasio yang diprediksi benar berbanding keseluruhan hasil yang diprediksi oleh sistem , $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$. IOU, IOU adalah rasio tumpang tindih *(Intersection Over Union)* antara kotak prediksi dan kotak ground truth. Semakin besar IOU, maka akan semakin tinggi tingkat akurasinya.[15]

Hasil dan Pembahasan

- **Pegujian deteksi objek yang tidak memakai masker dan memakai masker**

Pengujian deteksi objek dilakukan untuk menguji apakah sistem dapat medeteki objek yang memakai masker dan objek yang tidak memakai masker, pengujian dilakukan dengan 2 objek orang yang memakai masker dan tidak memakai masker.

Pada pengujian di gambar 12 dapat dilihat bahwa sistem dapat mendeteksi adanya objek yang memakai masker dan tidak memakai masker dengan nilai keakuratan objek memakai masker 99.54% dan objek tidak memakai masker dengan keakuratan 99.95%, dapat disimpulkan bahwa sistem dapat mendeteksi objek yang tidak memakai dan memakai masker.



Gambar 12. Pengujian deteksi objek

Hasil dan Pembahasan

- **Pengujian Berdasakan Jarak**

Pengujian berdasarkan jarak dilakukan di dalam dan luar ruangan dengan 2 kondisi yaitu, jarak terdekat yang jarak antara kamera dan objek berkisar 30cm – 200 cm. Sedangkan pengujian berdasarkan jarak terjauh, jarak antara kamera dan objek berkisar 300cm – 500cm.

Hasil dan Pembahasan

- **Pengujian Berdasakan Jarak**

Pengujian berdasarkan jarak dilakukan di dalam dan luar ruangan dengan 2 kondisi yaitu, jarak terdekat yang jarak antara kamera dan objek berkisar 30cm – 200 cm. Sedangkan pengujian berdasarkan jarak terjauh, jarak antara kamera dan objek berkisar 300cm – 500cm.

Pada pengujian jarak terdekat peneliti mendapat beberapa hasil yaitu, pengujian dengan jarak 30 cm dari *webcam* sistem dapat mendeteksi objek dengan waktu deteksi kurang dari 1 detik, untuk pengujian dengan jarak 50cm sampai 100 cm sistem dapat mendeteksi objek dengan waktu kurang dari 1 detik sedangkan untuk pengujian pada jarak 200 cm sistem dapat mendeteksi objek dengan waktu kurang dari 1 detik.

No	Jarak	Gambar	Waktu deteksi	Hasil
1	30 cm		< 1 detik	Terdeteksi
2	50 cm – 100 cm		< 1 detik	Terdeteksi
3	200 cm		< 1 detik	Terdeteksi

Pengujian bedasarkan jarak terdekat

Hasil dan Pembahasan

- **Pengujian Berdasakan Jarak**

Pada pengujian jarak terajuh peneliti melakukan pengujian di luar ruangan dengan jarak dimulai dari jarak 300 cm hingga jarak 500 cm. Di pengujian pada jarak 300 cm sistem dapat mendeteksi objek dengan baik dengan waktu kurang dari 1 detik, pada jarak 400 cm sistem dapat mendeteksi objek dengan waktu kurang dari 1 detik, dan pada pengujian di jarak 500 cm sistem dapat mendeteksi objek akan tetapi membutuhan waktu lebih lama yaitu lebih dari 1 detik.

No	Jarak	Gambar	Waktu deteksi	Hasil
1	300 cm		< 1 detik	Terdeteksi
2	400 cm		< 1 detik	Terdeteksi
3	500 cm		> 1 detik	Terdeteksi

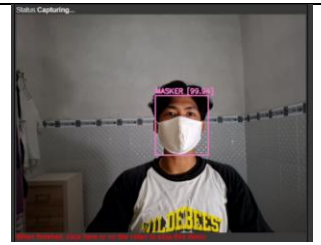
Pengujian berdasarkan jarak terjauh

Hasil dan Pembahasan

- **Pengujian Berdasarkan Intensitas Pancaran Cahaya**

Pengujian berdasarkan intensitas pancaran Cahaya dilakukan dengan 3 kondisi yaitu kondisi Cahaya normal, terang dan gelap.

Di pengujian berdasarkan intensitas cahaya pada kondisi cahaya normal dengan cahaya lampu ruangan dengan jarak kurang lebih 30 cm sampai 40 cm dari *webcam* sistem dapat mendeteksi objek dengan waktu kurang dari 1 detik, pada pengujian dengan intensitas cahaya terang dengan penambahan cahaya 30% dari kondisi cahaya normal, sistem dapat mendeteksi objek dengan waktu kurang dari satu detik. Dan pada pengujian denga cahaya gelap dengan pengurangan cahaya 30% dari cahaya normal sistem dapat mendeteksi objek dengan waktu lebih dari 2 detik.

No	Kondisi Cahaya	Gambar	Waktu deteksi	Hasil
1	Normal		< 1 detik	Terdeteksi
2	Terang, panambahan cahaya 30%		< 1 detik	Terdeteksi
3	Gelap, pengurangan Cahaya 30%		> 2 detik	Terdeteksi

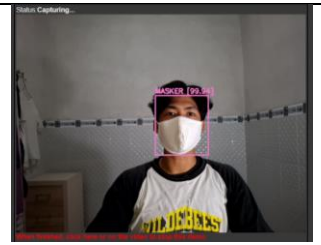
Pengujian berdasarkan jarak terjauh

Hasil dan Pembahasan

- **Pengujian Berdasarkan Intensitas Pancaran Cahaya**

Pengujian berdasarkan intensitas pancaran Cahaya dilakukan dengan 3 kondisi yaitu kondisi Cahaya normal, terang dan gelap.

Di pengujian berdasarkan intensitas cahaya pada kondisi cahaya normal dengan cahaya lampu ruangan dengan jarak kurang lebih 30 cm sampai 40 cm dari *webcam* sistem dapat mendeteksi objek dengan waktu kurang dari 1 detik, pada pengujian dengan intensitas cahaya terang dengan penambahan cahaya 30% dari kondisi cahaya normal, sistem dapat mendeteksi objek dengan waktu kurang dari satu detik. Dan pada pengujian denga cahaya gelap dengan pengurangan cahaya 30% dari cahaya normal sistem dapat mendeteksi objek dengan waktu lebih dari 2 detik.

No	Kondisi Cahaya	Gambar	Waktu deteksi	Hasil
1	Normal		< 1 detik	Terdeteksi
2	Terang, panambahan cahaya 30%		< 1 detik	Terdeteksi
3	Gelap, pengurangan Cahaya 30%		> 2 detik	Terdeteksi

Pengujian berdasarkan jarak terjauh

Hasil dan Pembahasan

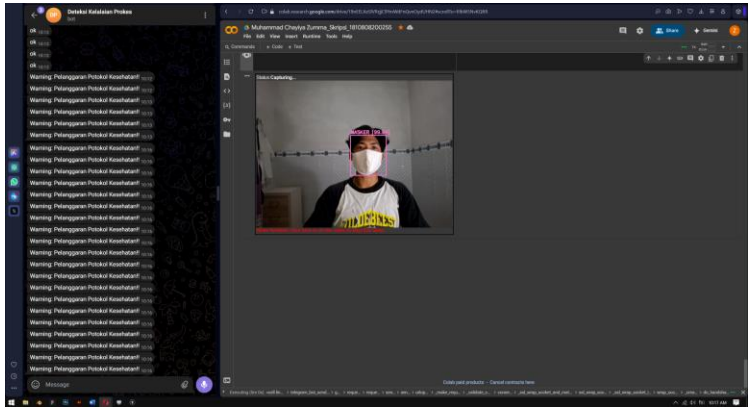
- **Pengujian sistem dapat mengirim notifikasi ke telegram**

Apabila sistem mendeteksi adanya objek yang tidak memakai masker maka maka sistem akan mengirim notifikasi ke *bot telegram* yang telah di buat atau pemberitahuan melalui telegram bahwa terdapat objek yang tidak memakai masker dapat dilihat pada gambar 13. Dan apabila sistem mendeteksi adanya objek yang memakai masker maka sistem tidak akan mengirim notifikasi ke bot atau pemberitahuan, dapat dilihat pada gambar 14.

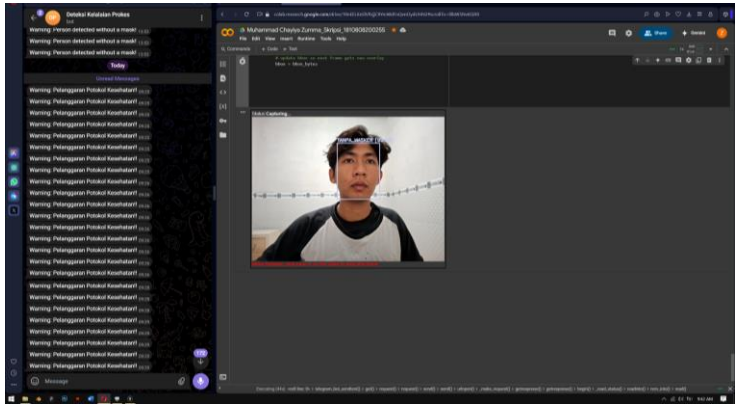
Pada gambar 13 sistem mendeteksi adanya objek yang tidak memakai masker dan sistem mengirim notifikasi ke *bot telegram* yang telah dibuat denga peringatan terdapat objek yang melanggar Protokol Kesehatan seperti pada gambar 15. Dan pada gambar 14 sistem juga medeteksi objek yang memakai masker tetapi tidak mengirim notifikasi ke *bot telegram* bahwa terdapat objek yang melanggar Protokol Kesehatan.

Hasil dan Pembahasan

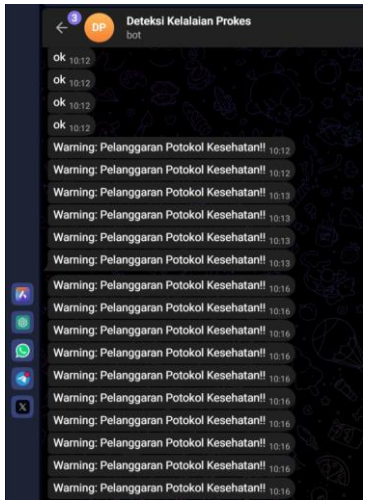
- **Pengujian sistem dapat mengirim notifikasi ke telegram**



Gambar 13. Pengujian pada objek tidak memakai masker



Gambar 14. Pengujian pada objek memakai masker



Gambar 15. Notifikasi terdeteksi tidak memakai masker

Simpulan

Setelah melakukan beberapa skenario pengujian, peneliti mendapatkan beberapa kesimpulan yaitu, untuk mendapatkan nilai presisi dan keakuratan yang tinggi, dibutuhkan data latih yang lebih banyak. Kedua, berdasarkan pengujian sistem dapat mengenali objek yang menggunakan masker dan tidak menggunakan masker. Ketiga, berdasarkan pengujian sistem dapat mendeteksi objek dari jarak terdekat 30cm – 200cm dari webcam dengan waktu kurang dari 1 detik. Keempat, Sistem dapat mendeteksi objek dari jarak terjauh 300cm - 500cm dari webcam, dengan waktu kurang dari 1 detik untuk jarak 300cm – 400cm dan waktu lebih dari 1 detik pada jarak 500cm. Selanjutnya penguji menguji berdasarkan jarak objek dari webcam untuk mendeteksi objek mempengaruhi kecepatan dan keakuratan dalam mendeteksi objek. Hasil pengujian sistem berdasarkan intensitas pancaran Cahaya, Cahaya sangat mempengaruhi sistem untuk dapat mendeteksi objek. Dalam kondisi Cahaya normal sistem dapat mendeteksi objek dengan baik dan akurat, dalam kondisi gelap sistem kesulitan untuk dapat mendeteksi objek. Sistem dapat mengirim notifikasi ke telegram saat mendeteksi adanya objek yang melanggar protokol Kesehatan atau tidak menggunakan masker.

Pada penelitian berikutnya diharapkan dapat lebih mengembangkan sistem ini dengan lebih banyak data latih dan lebih banyak variasi pelanggaran protokol kesehatan. Dan peneliti berharap untuk tampilan *Interface* pada sistem ini dapat di kembangkan lebih baik. Peneliti juga berharap agar sistem ini dapat dikombinasikan dengan notifikasi atau pemberitahuan yang lebih akurat dan tertuju pada pihak yang lebih spesifik.

Referensi

[1] A. N. Fatyandri, N. Hadiyati, and K. Rusyen, "Pemberian Sarana Dan Panduan Singkat Prosedur Kesehatan Untuk Mengatasi Ramainya Pengunjung Di Tengah Kondisi Pandemi Covid-19 BagiToko Mulya Jaya," vol. 3, pp. 777–781, 2021.

[2] A. Cahyani and A. S. Putri, "Meninjau Respon Masyarakat Terkait Pemenuhan Hak Ekosob Melalui Kebijakan Protokol Kesehatan di Masa Pandemi," *Semin. Nas. Huk. Univ. ...*, vol. 7, no. 1, 2021, [Online]. Available: https://proceeding.unnes.ac.id/index.php/snh/article/view/703.

[3] I Kadek Arya Andika, I Nyoman Sugiartha, and I Nyoman Sutama, "Sanksi Hukum terhadap Pejabat Negara yang Melanggar Protokol Kesehatan di Masa Pandemi Covid-19," *J. Interpret. Huk.*, vol. 2, no. 2, pp. 308–314, 2021, doi: 10.22225/juinhum.2.2.3432.308-314.

[4] O. N. Shpakov and G. V. Bogomolov, "Technogenic activity of man and local sources of environmental pollution," *Stud. Environ. Sci.*, vol. 17, no. C, pp. 329–332, 1981, doi: 10.1016/S0166-1116(08)71924-1.

[5] M. Effendi, F. Fitriyah, and U. Effendi, "Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan," *J. Teknotan*, vol. 11, no. 2, p. 67, 2017, doi: 10.24198/jt.vol11n2.7.

[6] P. M. Konvolusi, "Penerapan Metode Konvolusi ..... (Wikaria Gazali ; dkk) PENERAPAN METODE KONVOLUSI DALAM."

[7] R. T. Handayanto and H. Herlawati, "Prediksi Kelas Jamak dengan Deep Learning Berbasis Graphics Processing Units," *J. Kaji. Ilm.*, vol. 20, no. 1, pp. 67–76, 2020, doi: 10.31599/jki.v20i1.71.

Referensi

[8] J. S. W. Hutauruk, T. Matulatan, and N. Hayaty, "Deteksi Kendaraan secara Real Time menggunakan Metode YOLO Berbasis Android," *J. Sustain. J. Has. Penelit. dan Ind. Terap.*, vol. 9, no. 1, pp. 8–14, 2020, doi: 10.31629/sustainable.v9i1.1401.

[9] C. N. Liunanda, S. Rostianingsih, and A. N. Purbowo, "Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android," *J. Infra*, vol. Vol 8, No., pp. 1–7, 2020.

[10] A. R. Wasril, M. S. Ghozali, and M. B. Mustafa, "Pembuatan Pendeteksi Obyek Dengan Metode You Only Look Once (Yolo) Untuk Automated Teller Machine (Atm)," *Maj. Ilm. UNIKOM*, vol. 17, no. 1, pp. 69–76, 2019, doi: 10.34010/miu.v17i1.2240.

[11] R. N. Dhiaegana, "Penerapan Convolutional Neural Network untuk Deteksi Pedestrian pada Sistem Autonomous Vehicle," *Inst. Teknol. Bandung*, 2020.

[12] I. H. Al amin and A. Aprilino, "Implementasi Algoritma Yolo Dan Tesseract Ocr Pada Sistem Deteksi Plat Nomor Otomatis," *J. Teknoinfo*, vol. 16, no. 1, p. 54, 2022, doi: 10.33365/jti.v16i1.1522.

[13] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.

[14] A. Kanisius, E. Lapian, S. R. U. A. Sompie, and P. D. K. Manembu, "You Only Look Once (YOLO) Implementation For Signature Pattern Classification," vol. 16, no. 3, pp. 337–346, 2021.

Referensi

[15] G. N. Rizkatama, A. Nugroho, and F. Suni, “Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan,” vol. 8, no. 2, pp. 91–99, 2021.


UMSIDA
DARI SINI PENCERAHAN BERSEMI