

Implementasi *Computer Vision* Dalam Mendeteksi Pelanggaran Tidak Menggunakan Helm Pada Pengendara Motor

Implementation Of Computer Vision In Detecting Violations Of Not Wearing Helmet On Motorcycles

1st Ananda Ayu Chellsya
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

anandaayu@student.telkomuniversity.ac.id

2nd Suci Aulia
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

suciaulia@telkomuniversity.ac.id

3rd Sugondo Hadiyoso
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

sugondo@telkomuniversity.ac.id

Abstrak—Helm merupakan salah satu safety riding bagi pengendara motor yang berfungsi melindungi kepala. Rendahnya kesadaran pengguna motor untuk menggunakan perangkat keselamatan yang sesuai dengan Undang-Undang. Tingkat kecelakaan di Indonesia di dominasi oleh kendaraan sepeda motor. Ada juga kewajiban mengenai penumpang sepeda motor juga harus menggunakan helm yang memenuhi standar nasional Indonesia sesuai dengan Pasal 106 ayat (8) No. 22/2009. Oleh karena itu dibuatlah sebuah sistem deteksi helm. Salah satu metode yang diciptakan untuk menciptakan object detection yaitu metode You Only Look Once (YOLO) . Cara kerja YOLO yaitu dengan melihat seluruh gambar sekali , kemudian melewati jaringan syaraf sekali langsung mendeteksi object yang ada. Metode untuk pengenalan objek helm pada pengguna sepeda motor tersebut adalah Convolution Neutral Network (CNN). Yang dihasilkan dari pengerjaan Proyek Akhir ini adalah sebuah sistem deteksi helm yang dapat mendeteksi pengendara motor yang menggunakan dan tidak menggunakan helm yang bisa diaplikasikan di jalan raya. Sistem ini dibuat menggunakan metode YOLO Tiny3.

Hasil Implementasi ini bertujuan untuk mendeteksi pengendara motor yang menggunakan helm dan tidak menggunakan helm, dengan hasil keluaran berupa citra yang telah terdeteksi orang yang menggunakan helm atau tidak menggunakan helm dengan adanya *labelling* dan bounding box pada citra yang telah terdeteksi pada pengendara motor di jalan raya. Dari hasil pengujian pada total 15 gambar menunjukkan bahwa sistem dapat memunculkan notifikasi ada atau tidak adanya pelanggaran

pengendara motor yang dilakukan pada 3 skenario terdapat nilai akurasi sebesar 100% dengan waktu proses 0.2 detik. Dan hasil pengujian pada total 75 gambar yang dilakukan pada 3 skenario terdapat rata-rata nilai presisi sistem tertinggi yaitu 88% pada skenario 1 dengan rata-rata waktu proses 0.2 detik dan rata-rata nilai presisi terendah ada pada skenario 2 yaitu 76% dengan rata-rata waktu proses 0.2 detik

Kata kunci — *helm, YOLO, deteksi objek*

Abstract—Helmets are one of the safety riding for motorbike drivers that function to protect the head. Low awareness of motor users to use safety devices in accordance with the Act. The accident rate in Indonesia is dominated by motorcycle vehicles. There is also an obligation regarding motorcycle passengers must also use helmets that meet Indonesian national standards in accordance with Article 106 paragraph (8) No. 22/2009. Therefore, a helmet detection system was created. One of the methods created to create object detection is the You Only Look Once (YOLO) method. The way YOLO works is by looking at the entire image once, then passing through the neural network once directly detecting the object. The method for the recognition of the object of the helmet in the user of such a motorcycle is the Convolution Neutral Network (CNN). What results from the work on this Final Project is a helmet detection system that can detect motorcyclists who use and do not use helmets that can be applied on the road. This system is made using the YOLO Tiny3 method, it is hoped that with the system many motorcyclists are aware of the use of helmets and minimize accidents. The results of this

implementation aim to detect motorcycle riders who use helmets and do not use helmets, with the output in the form of images that have been detected by people who use helmets or do not use helmets with labeling and bounding boxes on images that have been detected on motorcyclists on the highway. From the test results on a total of 15 images, it shows that the system can display notifications whether or not there is a motorcyclist violation committed in 3 scenarios, there is an accuracy value of 100% with a processing time of 0.2 seconds. And the test results on a total of 75 images carried out in 3 scenarios there is an average value of the highest system precision that is 88% in scenario 1 with an average processing time of 0.2 seconds and the lowest average precision value is in scenario 2 which is 76% with an average -average processing time 0.2 seconds

Keyword—: helmet, YOLO, detection object

I. PENDAHULUAN

Helm merupakan salah satu safety riding bagi pengendara motor yang berfungsi melindungi kepala [1]. Kelalaian dalam menggunakan helm adalah salah satu yang paling sering terjadi dan mengakibatkan kesalahan fatal. Rendahnya kesadaran pengguna motor untuk menggunakan perangkat keselamatan yang sesuai dengan Undang-Undang. Tingkat kecelakaan di Indonesia didominasi oleh kendaraan sepeda motor. Bagi pengendara sepeda motor, mengenakan helm merupakan hal yang wajib dilakukan demi keamanan dan keselamatan diri. Alasannya bukan untuk mencegah agar tidak ditilang saja, melainkan untuk melindungi kepala dari benturan jika terjadi kecelakaan. Dengan memakai helm saat berkendara dapat meminimalkan resiko terbenturnya kepala saat mengalami kecelakaan. Helm memang tidak menjamin seluruh organ tubuh akan aman, namun akan melindungi kepala kita. Berdasarkan data dari Korlantas Polri yang dipublikasikan Kementerian Perhubungan, angka kecelakaan lalu lintas di Indonesia mencapai 103.645 Kasus pada tahun 2021. Jumlah tersebut lebih tinggi dibandingkan data tahun 2020 yang sebanyak 100.028 kasus. Berdasarkan jenis kendaraan, keterlibatan kasus kecelakaan lalu lintas yang paling tinggi adalah sepeda motor dengan persentase 73% [2]. Ada juga kewajiban mengenai penumpang sepeda motor juga harus menggunakan helm yang memenuhi standar nasional Indonesia sesuai dengan Pasal 106 ayat (8) No. 22/2009.

Teknologi Visi Komputer merupakan bidang kecerdasan buatan atau *Artificial Intelligence* yang melatih komputer untuk

menafsirkan dan memahami sesuatu secara visual. Visi Komputer memanfaatkan gambar digital melalui kamera sebagai masukan serta jaringan *neural* untuk proses mengidentifikasi dan mengklasifikasikan objek dan kemudian mendeteksi apa yang dilihat oleh komputer. Saat ini, Visi Komputer telah banyak diimplementasikan dalam berbagai bidang seperti kesehatan, sosial, administrasi sampai transportasi[3]

. Sementara itu metode untuk pengenalan object detection helm pada pengguna sepeda motor tersebut adalah Convolution Neural Network (CNN). Diharapkan metode ini dapat menghasilkan akurasi dan kecepatan yang baik dikarenakan metode YOLO merupakan state-of-the-art dalam deteksi objek secara real-time dan metode CNN yang sangat baik dalam memproses data special dan mengklasifikasi objek[4].

Pada penelitian sebelumnya [5], telah dilakukan deteksi helm pada pengendara motor. Hasilnya tingkat akurasi sebesar 70,49% yang dilakukan pengujian pada siang hari dan 54,54% yang dilakukan pada malam hari menggunakan metode YOLO. Maka sebagai kelanjutannya, pada Proyek Akhir ini disimulasikan sistem deteksi helm pada pengendara motor di tiga titik poin atau tiga lokasi yang berbeda. Pada tiga titik point akan menampilkan keluaran berupa notifikasi ada atau tidak adanya pelanggaran. Dengan adanya prototype ini, diharapkan dapat menjadi langkah awal untuk selanjutnya dapat dikembangkan dan diimplementasikan sehingga dapat membantu mendeteksi pelanggaran lalu lintas.

II. KAJIAN TEORI

A. Computer Vision

Computer vision adalah sebuah kemampuan sebuah aplikasi agar mampu melihat sebuah object merubahnya dalam format digital dan menampilkan nya dalam bentuk visual,dan telah terbukti komputer mampu handle pekerjaan manusia, pada jurnal ini saya coba membahas dan mengimplementasikan cara kerja komputer vision dalam mendeteksi sebuah objek berdasarkan warna atau colour marker dengan menggunakan kamera secara *realtime*. [6]

B. Helm

Helm merupakan komponen wajib pakai untuk pengendara sepeda motor,di Indonesia helm memiliki standart desain dan pembuatan sesuai dengan Standart Nasional Indonesia

(SNI) [7]. Berdasarkan SNI helm harus memenuhi persyaratan sebagai berikut:[8]

1. Helm harus terdiri dari tempurung yang keras dan memiliki permukaan yang halus, helm juga memiliki tali pengikat ke dagu pengendara.
2. Tinggi helm tidak kurang dari 114 mm yang diukur dari puncak helm ke bidang utamanya, yaitu bagian bawah dari dudukan bola mata dan bidang horizontal melalui lubang telinga.
3. Keliling lingkaran bagian dalam helm adalah S (antara 500 mm– 540 mm), M (540 mm – 580 mm), L (580 mm – 620 mm), XL (lebih dari 620 mm).
4. Tempurung memiliki tekstur keras dan tebal, pelindung tidak menyatu dengan pelindung muka dan mata (khusus helm *half face* bukan untuk helm *full face*).
5. Memiliki peredam beturan yang terdiri dari
9. Helm harus dilengkapi dengan pelindung telinga, penutup leher, pet yang bisadipindahkan, tameng atau tutup dagu.

C. Artificial Intelligence (AI)

Artificial Intelligence (AI) atau kecerdasan buatan adalah ilmu yang dikembangkan dengan kombinasi banyak subyek [9]. Dalam istilah sederhana, kecerdasan buatan adalah memberi mesin dengan kecerdasan manusia, mensimulasikan pemikiran manusia sehingga membantu orang memecahkan masalah dan untuk mewujudkan aplikasi yang lebih canggih seperti produksi dan kehidupan manusia yang dibantu computer [10].

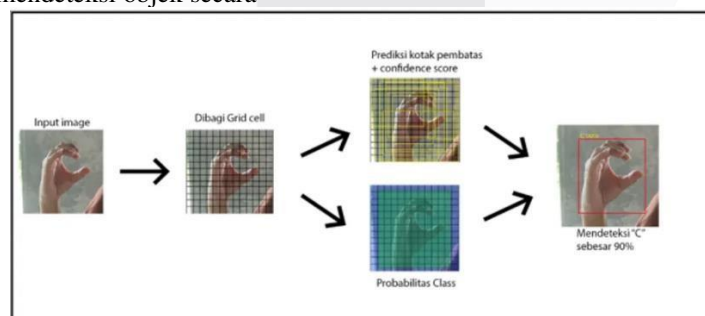
D. You Only Look Once (YOLO Tiny3)

YOLO *You Only Look Once* Version 3-Tiny (YOLOv3-tiny) *You Only Look Once* Versi 3-Tiny (YOLOv3-Tiny) merupakan algoritme yang digunakan untuk mendeteksi objek secara

lapisan peredam kejutan yang dipasang pada bagian dalam tempurung, dengan tebal sekurang-kurangnya 10mm.

6. Memiliki tali pengikat dagu yang lebarnya minimal 20 mm dan harus benar-benar berfungsi sebagai pengikat helm ketika dikenakan di kepala dan dilengkapi dengan penutup telinga dan tengkuk, sesuai dengan konstruksi helm *half face*.
7. Tempurung helm tidak boleh memiliki tonjolan keluar yang tingginya melebihi 5 mm dari permukaan luar tempurung dan setiap tonjolan harus ditutupi dengan bahan lunak. Helm tidak boleh memiliki bagian tepi yang tajam.
8. Lebar sudut pandang sekeliling sekurang-kurangnya 105 derajat pada tiap sisi dan sudut pandang vertikal sekurang-kurangnya 30 derajat di atas dan 45 derajat di bawah bidang utama.

waktu nyata. Secara garis besar YOLOv3-tiny dalam memprediksi sebuah objek pada citra dilakukan dengan beberapa tahap. Tahap pertama yaitu citra yang akan diprediksi akan dibagi 6 ke dalam beberapa grid cell. Jumlah grid cell ditentukan dengan membagi banyaknya piksel pada citra input dengan nilai stride. Setelah citra dibagi ke dalam beberapa grid cell, tahap selanjutnya yaitu akan YOLOv3-tiny memprediksi anchor box sebanyak B (nilai B dapat di deklarasikan) pada setiap grid cell. Anchor box dapat dilihat pada garis hitam putus putus Gambar 1 Selanjutnya, dari anchor box yang sudah ada YOLOv3-tiny akan memprediksi kotak pembatas atau bounding box pada setiap objek. Dari kotak pembatas sudah ada, YOLOv3-tiny akan memilih kotak pembatas mana yang paling sesuai dalam memprediksi objek. Tahap umum sistem prediksi pada YOLOv3-tiny dapat dilihat pada Gambar 1



GAMBAR 1
ALUR KERJA YOLO TINY 3

Jaringan YOLOv3-tiny akan memprediksi 4 koordinat pada setiap kotak pembatas, yaitu t_x , t_y , t_w , t_h . Parameter p_w , p_h merupakan dimensi anchor box. Jika suatu cell merupakan batas dari sudut atas kiri gambar (c_x , c_y) dan anchor box memiliki lebar dan tinggi p_w , p_h . [11]

E. Image Processing

Pengolahan citra atau *Image Processing* adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun

dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses computer. Serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka image processing tidak dapat dilepaskan dengan bidang computer vision [13]

F. Deteksi objek

Deteksi Objek menentukan keberadaan sebuah objek, ruang lingkungnya, dan lokasi pada gambar. Objek deteksi mengidentifikasi kelas objek yang terdapat pada database yang telah ditraining. Deteksi objek diawali oleh

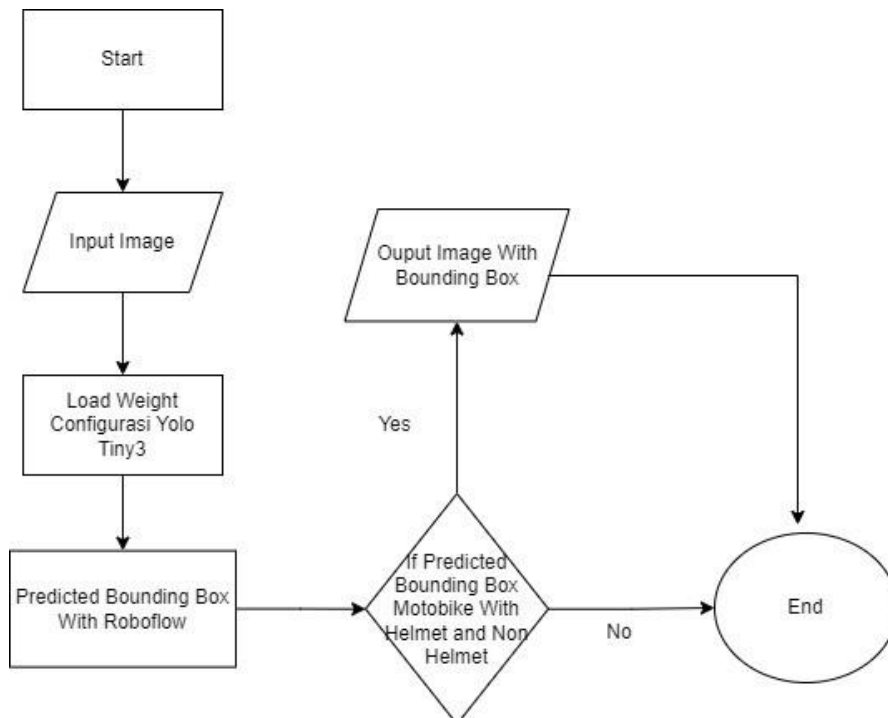


pengenalan suatu objek. Hal ini dapat diperlakukan sebagai pengenalan objek kelas dua, dimana satu kelas mewakili kelas objek dan kelas lain mewakili kelas non-objek. Deteksi objek dibagi menjadi dua, yaitu *soft detection* dan *hard detection*. *Soft detection* hanya dapat mendeteksi keberadaan suatu objek sedangkan *hard 5 detection* mendeteksi keberadaan objek dan lokasi objek pada gambar [14].

III. METODE

A. Tahap Perancangan Sistem

Pada tahap ini dijelaskan tahap perancangan sisten untuk deteksi helm pada pengendara motor di jalan raya berbasis *you only look once* (YOLO) yang digambarkan dalam *flowchart* Gambar 2 dibawah ini :



GAMBAR 2
FLOWCHART SISTEM DETEKSI HELM DENGAN METODE YOLO

You Only Look Once (YOLO) digunakan untuk mendeteksi objek pengendara motor yang menggunakan helm dan tidak menggunakan helm pada gambar. *Load weight* nya itu menggunakan YOLO Tiny3. *Weight* dari *dataset* yang digunakan sudah mampu untuk mengenali objek helm dan *non helmet* dengan akurasi yang cukup tinggi. Dalam hal ini Yolo Tiny3 akan mendapatkan nilai keberadaan *bounding box* dan *confidence score* nya. Setelah itu YOLO akan mengklasifikasi dan menandai

dengan *bounding box* helm dan *non helmet* pada *output* atau keluaran.

B. Implementasi Yolo Tiny 3

Berikut merupakan implementasi Yolo Tiny3 menggunakan bahasa pemrograman *python* dan dilakukan di *google collaboratory*. Berikut tahapan pemograman yang dibuat :

1. Menghubungkan google colab dengan google drive

```
[ ] 1 from google.colab import drive
     2 drive.mount('/content/drive')
```

Mounted at /content/drive

GAMBAR 3
SCRIPT UNTUK MENGHUBUNGAN GOOGLE COLAB DENGAN GOOGLE DRIVE

2. Memasukkan library yang digunakan agar program bisa berjalan


```
[ ] 1 import os
    2 import cv2
    3 import numpy as np
    4 import matplotlib.pyplot as plt
```

GAMBAR 4
SCRIPT UNTUK MEMASUKAN LIBRARY

3. Melihat atau memeriksa OpenCV yang di install

```
[ ] 1 cv2.__version__

'4.6.0'
```

GAMBAR 5
SCRIPT UNTUK MELIHAT OPENCV

4. Mengupload Dataset citra berbentuk zip yang telah di *bounding* box menggunakan label di setiap citra yang didapatkan dari hasil pensimulasian menggunakan *roboflow*

```
[ ] 1 %cd /content
    2
    3 import shutil
    4 from zipfile import Zipfile
    5 from google.colab import files
    6 from os import path
    7
    8 print("Upload `dataset.zip` to colab :")
    9 print("-"*20)
   10
   11 uploaded = files.upload()
   12
   13 for fileName, data in uploaded.items():
   14     with open('dataset.zip', 'wb') as f:
   15         f.write(data)
   16         f.close()
   17     print('saved dataset (.zip) file ' + fileName)
   18
   19 ds = ZipFile(fileName)
   20 ds.extractall()
   21 os.remove(fileName)
   22 print('Extracted zip file ' + fileName)
```

GAMBAR 6
SCRIPT UNTUK MENGUPLOAD DATASET BERBENTUK ZIP

5. Mengunduh model YOLO Tiny3

weight dan *config* yang akan digunakan untuk mendeteksi objek

```
[ ] 1 # Get tiny YOLO v3 weights & config
    2 !mkdir /content/dataset/models/
    3 %cd /content/dataset/models/
    4
    5 !wget https://pjreddie.com/media/files/yolov3-tiny.weights
    6 !wget https://pjreddie.com/media/files/yolov3.weights
```

GAMBAR 7
SCRIPT UNTUK MENGUNDUH YOLO TINY 3 DAN MENCONFIG AGAR MENDETEKSI OBJEK

6. Mengkomplikasi *darknet*

mengcopy versi yang dikompilasi ke drive

```
[ ] 1 # Leave this code uncommented on the very first run of your notebook or if you ever need to recompile darknet again.
    2 # Comment this code on the future runs.
    3 %cd /content/
    4 !git clone https://github.com/AlexeyAB/darknet.git
    5 %cd darknet
    6
    7 # Check the folder
    8 !ls
```

GAMBAR 8
SCRIPT UNTUK MENKOMPLIKASI DARKNET KE DRIVE



7. Menginstall *darknet*

```
[ ] 1 #Compile Darknet
    2 !make
```

GAMBAR 9
SCRIPT UNTUK MENGINSTALL DARKNET

8. Mengkompilasi darknet mengcopy

```
[ ] 1 %cd /content/
    2
    3 import os
    4 from google.colab import files
    5
    6 print("Upload image to Colab!")
    7 uploaded = files.upload()
    8 for name, data in uploaded.items():
    9     with open('image1.jpg', 'wb') as f:
    10        f.write(data)
    11        os.remove(name)
    12        print ('saved file as as image1.jpg')
```

GAMBAR 10
SCRIPT UNTUK MENGCOPY VERSI YANG DIKOMPILASI KE DRIVE

versi yang dikompilasi ke drive

9. Menentukan ukuran gambar yang akan dikeluarkan

```
[ ] 1 def show_image(image_path):
    2     img = cv2.imread(image_path)
    3
    4     plt.figure(figsize=(10,10))
    5     plt.imshow(img[:,:,:-1])
```

GAMBAR 11
SCRIPT UNTUK MENENTUKAN UKURAN GAMBAR

10. Mengupload image ke colab untuk

```
[ ] 1 image_path = '/content/image1.jpg'
    2
    3 show_image(image_path)
```

GAMBAR 12
SCRIPT UNTUK MENGUPLOAD IMAGE KE COLAB

dilakukan test darknet (Inference Tiny Yolo V3)

11. Melihat hasil bahwa image yang di upload telah dapat terdeteksi secara umum

```
[.] 1 %cd /content/darknet/
    2
    3 # Execute darknet using tiny YOLOv3 model with pre-trained weights to detect objects
    4 !./darknet detect cfg/yolov3.cfg /content/dataset/models/yolov3.weights /content/image1.jpg -dont-show
    5
    6 # Show the result using the helper imgShow()
    7 show_image('/content/darknet/predictions.jpg')
```

GAMBAR 13
SCRIPT UNTUK MELIHAT HASIL IMAGE YANG SUDAH TERDETEKSI

12. Mendownload *darknet53.conv.74* atau *yolov3-tiny.conv.11* adalah bobot yolo awal (model dasar) untuk melatih data khusus dan

yolov3.weights atau *yolov3-tiny.weight* adalah bobot terlatih yang dapat digunakan untuk mendeteksi.

```
[ ] 1 mkdir -p /content/dataset/models/
    2 mkdir -p /content/dataset/backup
    3 %cd /content/dataset/models/
    4
    5 # copy config of Yolo V3 & Tiny Yolo V3
    6 !cp /content/darknet/cfg/yolov3-tiny.cfg .
    7 !cp /content/darknet/cfg/yolov3.cfg .
    8
    9 # download tiny yolo v3 pre-trained model from google drive using 'gdown'
    10 !gdown --id 18v36esoXCh-Ps0KvYp2G0WrpYDptDv8Zf
    11
    12 # download yolo v3 pre-trained model from pjreddie.com
    13 !wget https://pjreddie.com/media/files/darknet53.conv.74
```

GAMBAR 14
SCRIPT UNTUK MENDOWNLOAD DARKNET53.CONV.74

13. Membuat *train.txt* dan *test.txt*


```
[ ] 1 #original
2 '''
3 !find /content/dataset/images/test/*.jpg > /content/dataset/data/train.txt
4 !find /content/dataset/images/train/*.jpg > /content/dataset/data/test.txt
5 '''
6 #sca
7 !find /content/test/*.jpg > /content/train.txt
8 !find /content/train/*.jpg > /content/test.txt
```

GAMBAR 15
SCRIPT UNTUK MEMBUAT TRAIN.TXT DAN TEST.TXT

14. Membuat file obj.data dengan membuat folder di data di dalam dataset

```
[ ] 1 !touch /content/dataset/data/obj.data
```

• edit [/content/dataset/data/obj.data](#) with this content,

GAMBAR 16
SCRIPT UNTUK MEMBUAT FILE OBJ.DATA

15. Menginstall pyngrok

```
[ ] 1 !pip install pyngrok
2 from pyngrok import ngrok# Open a HTTP tunnel on port 8090
3 public_url = ngrok.connect(port = '8090')
4
5 public_url
```

GAMBAR 17
SCRIPT UNTUK MENGINSTALL PYNGROK

16. Mentraning atau memproses data

```
[ ] 1 %cd /content/darknet/
2 !./darknet detector train \
3 /content/dataset/data/obj.data \
4 /content/dataset/models/yolov3.cfg \
5 /content/dataset/models/darknet53.conv.74 \
6 -dont_show --mjpeg_port 8090 -map
```

GAMBAR 18
SCRIPT UNTUK MEMPROSES DATA

17. Mengupload image yang akan di dideteksi sesuai kelasnya yaitu helm dan *nonhelmet*

```
[ ] 1 %cd /content/
2 %mkdir test_images -p
3
4 from google.colab import files
5
6 print("Upload image to Colab!")
7 uploaded = files.upload()
8 for name, data in uploaded.items():
9     with open('test_images/' + name, 'wb') as f:
10         f.write(data)
11         print ('saved file', name)
```

GAMBAR 19
SCRIPT UNTUK MENGUPLOAD IMAGE

18. Melihat hasil deteksi objek menggunakan model Yolo Tiny3

```
[ ] 1 %cd /content/darknet/
2
3 # Execute darknet using tiny YOLOV3 model with pre-trained weights to detect objects
4 !./darknet detector test \
5 /content/dataset/data/obj.data \
6 /content/dataset/models/yolov3.cfg \
7 /content/dataset/backup/yolov3_final.weights \
8 /content/test_images/btn_27.jpeg\
9 -dont-show
10
11 # Show the result using the helper imgShow()
12 show_image('/content/darknet/predictions.jpg')
```

GAMBAR 20
SCRIPT UNTUK MELIHAT HASIL DETEKSI OBJEK MENGGUNAKAN MODEL YOLO TINY3

19. Contoh hasil deteksi objek menggunakan model Yolo Tiny3



GAMBAR 21
CONTOH HASIL DETEKSI OBJEK MENGGUNAKAN MODEL YOLO TINY3

IV. HASIL DAN PEMBAHASAN

A. Pengujian Tingkat Akurasi

Pada bab ini dilakukan pengujian tingkat akurasi pada deteksi helm. Tingkat akurasi adalah ukuran seberapa dekat dengan nilai yang sebenarnya, pengujian ini terbagi menjadi 2 kelas yaitu helm dan *non helmet*. Dataset yang digunakan untuk melakukan pengujian di ambil langsung dari tiga titik point dan total dari tiga titik point tersebut ada 15 gambar untuk dilakukan pengujian masing masing skenario berjumlah 5 gambar dengan keluaran berupa memunculkan notifikasi berupa *teks* ada atau tidak adanya pelanggaran.

Pengujian dilakukan dengan 3 skenario yaitu dengan menggunakan dataset di jalan raya Banjaran, jalan raya Soreang, dan jalan raya Batununggal

1. Skenario 1 pertama yaitu pada dataset citra dengan kondisi di jalan raya Banjaran
2. Skenario 2 pertama yaitu pada dataset citra dengan kondisi di jalan raya Soreang
3. Skenario 3 pertama yaitu pada dataset citra dengan kondisi di jalan raya Batununggal

a. Skenario 1

Berdasarkan skenario 1 akan diujikan tingkat akurasi mendeteksi pelanggaran atau tidak ada pelanggaran pada citra hasil deteksi helm dengan YOLO Tiny3.

TABEL 1 HASIL PENGUJIAN SKENARIO 1

No	Dataset	Hasil Deteksi		Status	Waktu Proses
		Sistem	Validasi		
1	Bjr_31	Ada pelanggaran	Ada	Benar	0 . 2
2	Bjr_20	Ada pelanggaran	Ada	Benar	0 . 2
3	Bjr_7	Ada pelanggaran	Ada	Benar	0 . 3

4.	Bjr_25	Tidak Ada Pelanggaran	Tidak	Benar	0 . 2
5.	Bjr_27	Ada Pelanggaran	Ada	Benar	0 . 2

$$\text{Tingkat Akurasi} = \frac{\text{Jumlah Benar}}{\text{Jumlah Data}} * 100\% \quad (4.1)$$

Dari hasil pengujian pada skenario 1 terdapat 4 pelanggar pengendara motor dan 1 tidak ada pelanggaran atau memakai helm diperoleh rata rata tingkat akurasi 100% dengan rata rata proses 0.2 detik.

b. Skenario 2

Berdasarkan skenario 2 akan diujikan tingkat akurasi mendeteksi pelanggaran atau tidak ada pelanggaran pada citra hasil deteksi helm dengan YOLO Tiny3

TABEL 2
HASIL PENGUJIAN SKENARIO 2

No	Dataset	Hasil Deteksi		Status	Waktu Proses
		Sistem	Validasi		
1	Srg_3	Tidak Ada Pelanggaran	Tidak	Benar	0.2
2	Srg_10	Ada pelanggaran	Ada	Benar	0.2
3	Srg_20	Ada pelanggaran	Ada	Benar	0.3
4.	Srg_6	Tidak Ada Pelanggaran	Tidak	Benar	0.2
5.	Srg_52	Tidak Ada Pelanggaran	Tidak	Benar	0.2

$$\text{Tingkat Akurasi} = \frac{\text{Jumlah Benar}}{\text{Jumlah Data}} * 100\% \quad (4.2)$$

Dari hasil pengujian pada skenario 2 terdapat 2 pelanggar pengendara motor dan 3 tidak ada pelanggaran atau memakai helm diperoleh rata rata tingkat akurasi 100% dengan rata rata proses 0.2 detik.

c. Skenario 3

Berdasarkan skenario 3 akan diujikan tingkat akurasi mendeteksi pelanggaran atau tidak ada pelanggaran pada citra hasil deteksi helm dengan YOLO Tiny3.

TABEL 3
HASIL PENGUJIAN SKENARIO 3

No	Dataset	Hasil Deteksi		Status	Waktu Proses
		Sistem	Validasi		
1	Btn_6	Tidak Ada Pelanggaran	Tidak	Benar	0.2
2	Btn_8	Ada pelanggaran	Ada	Benar	0.2
3	Btn_9	Tidak Ada Pelanggaran	Tidak	Benar	0.3
4.	Btn_12	Tidak Ada Pelanggaran	Tidak	Benar	0.2
5.	Btn_15	Tidak Ada Pelanggaran	Tidak	Benar	0.2

$$\text{Tingkat Akurasi} = \frac{\text{Jumlah Benar}}{\text{Jumlah Data}} * 100\% \quad (4.3)$$

Dari hasil pengujian pada skenario 3 terdapat 1 pelanggar pengendara motor dan 4 tidak ada pelanggaran atau memakai helm

diperoleh rata rata tingkat akurasi 100% dengan rata rata proses 0.2 detik.

B. Pengujian Tingkat Presisi

Pada bab ini dilakukan pengujian tingkat presisi pada deteksi helm. Tingkat presisi adalah kedekatan nilai yang

diperoleh atau tingkat konsistensi dari pengamatan yang ditentukan dari besarnya perbedaan dalam nilai data yang dihasilkan, pengujian ini terbagi menjadi 2 kelas yaitu *helmet* dan *non helmet*. Dataset yang digunakan untuk melakukan pengujian di ambil langsung dari tiga titik point dan total dari tiga titik point tersebut ada 75 gambar untuk dilakukan pengujian masing masing skenario berjumlah 25 gambar . Pengujian

dilakukan dengan 3 skenario yaitu dengan menggunakan dataset di jalan raya Banjaran, jalan raya Soreang, dan jalan raya Batununggal

d. Skenario 1

Tabel 4 Perbandingan hasil deteksi helm pengendara motor menggunakan Yolo tiny3 dengan hasil validasi.

TABEL 4
PERBANDINGAN HASIL DETEKSI HELM PENGENDARA MOTOR SKENARIO 1

No	Dataset	Hasil Deteksi Helmet		Status	Hasil Deteksi NonHelmet		Status	Waktu
		Yolo Tiny3	Validasi		Yolo Tiny 3	Validasi		
1	Bjr_1	4	4	Benar	0	0	Benar	0.2
2	Bjr_2	9	9	Benar	1	1	Benar	0.2
3	Bjr_4	1	1	Benar	0	0	Benar	0.3
4.	Bjr_6	3	2	Salah	0	0	Benar	0.2
5.	Bjr_7	1	1	Benar	1	1	Benar	0.2
6.	Bjr_8	0	0	Benar	2	1	Salah	0.2
7	Bjr_9	6	5	Salah	0	0	Benar	0.2
8	Bjr_10	3	3	Benar	0	0	Benar	0.2
9.	Bjr_11	1	1	Benar	0	0	Benar	0.2
10.	Bjr_12	1	1	Benar	1	1	Benar	0.2
11	Bjr_13	0	0	Benar	2	1	Salah	0.2
12.	Bjr_15	2	2	Benar	1	1	Benar	0.2
13.	Bjr_17	2	2	Benar	0	0	Benar	0.2
14.	Bjr_18	2	2	Benar	1	1	Benar	0.2
15.	Bjr_19	1	1	Benar	0	0	Benar	0.2
16.	Bjr_20	1	1	Benar	1	1	Benar	0.2
17.	Bjr_23	3	3	Benar	2	2	Benar	0.3
18.	Bjr_25	5	5	Benar	0	0	Benar	0.2
19.	Bjr_26	2	2	Benar	2	1	Salah	0.2
20.	Bjr_27	0	0	Benar	2	2	Benar	0.2
21.	Bjr_28	5	5	Benar	1	1	Benar	0.2
22.	Bjr_30	0	0	Benar	1	1	Benar	0.2
23.	Bjr_31	1	1	Benar	3	3	Benar	0.2
24.	Bjr_32	5	3	Salah	0	0	Benar	0.2
25.	Bjr_33	9	9	Benar	1	1	Benar	0.2

Tingkat Presisi = Jumlah Benar/Jumlah Data*100%

(4.4)



Berdasarkan Tabel 4 tingkat presisi deteksi helm pengendara motor pada skenario 1 menggunakan yolo

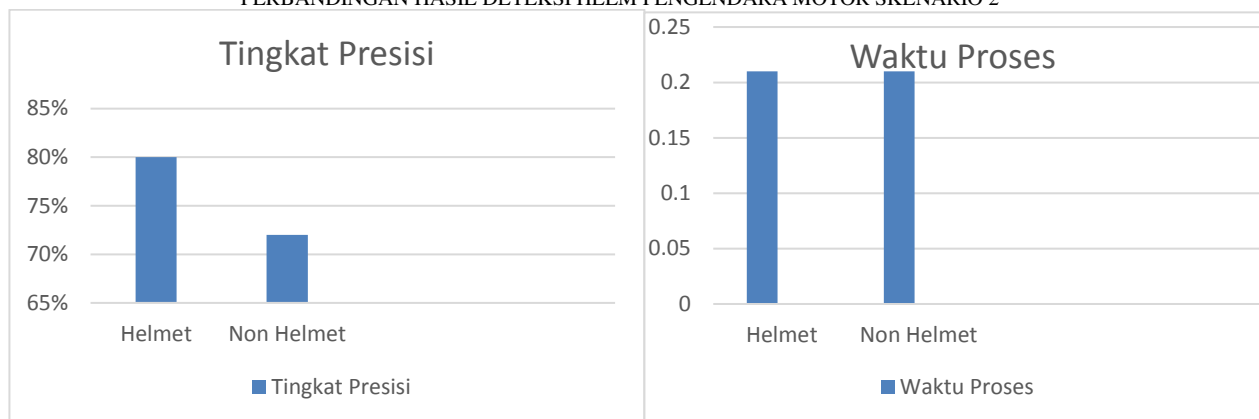
tiny3 diperoleh 88%, dengan rata rata waktu proses deteksi 0.20 detik. Berdasarkan Tabel 4 tingkat presisi deteksi *non helmet* pengendara motor pada skenario 1 menggunakan Yolo tiny3 diperoleh 88%, dengan rata rata waktu proses

deteksi 0.20 detik.

e. Skenario 2

Diagram Tabel 5 Perbandingan hasil deteksi helm pengendara motor menggunakan Yolo Tiny3 dengan hasil validasi.

TABEL 5
PERBANDINGAN HASIL DETEKSI HELM PENGENDARA MOTOR SKENARIO 2



Tingkat Presisi = Jumlah Benar/Jumlah Data*100%
(4.5)

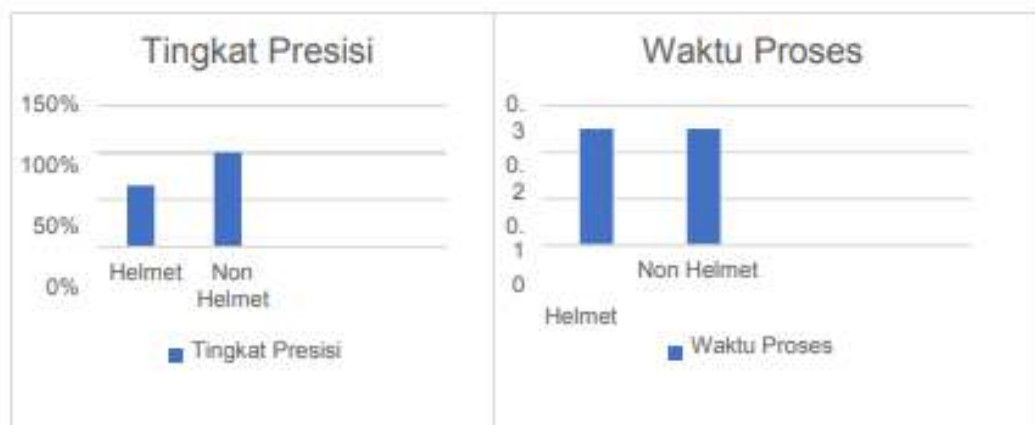
tiny3 diperoleh 72%, dengan rata rata waktu proses deteksi 0.21 detik.

Berdasarkan Diagram Tabel 4.5. tingkat presisi deteksi helm pengendara motor pada skenario 2 menggunakan yolo tiny3 diperoleh 80%, dengan rata rata waktu proses deteksi 0.21 detik. Berdasarkan Diagram Tabel 4.5 tingkat presisi deteksi *non helmet* pengendara motor pada skenario 2 menggunakan yolo

c. Skenario 3

Diagram Tabel 6 Perbandingan hasil deteksi helm pengendara motor menggunakan Yolo Tiny3 dengan hasil validasi.

Tabel 6 Perbandingan Hasil Deteksi Helm Pengendara Motor Skenario 3



Tingkat Presisi = Jumlah Benar/Jumlah Data*100%
(4.6)

Berdasarkan Diagram Tabel 4.6 tingkat presisi deteksi *non helmet* pengendara motor pada skenario 1 menggunakan Yolo tiny3 diperoleh 100%, dengan rata rata waktu proses deteksi 0.25 detik.

Berdasarkan Diagram Tabel 6 tingkat presisi deteksi helm pengendara motor pada skenario 3 menggunakan Yolo tiny3 diperoleh 65%, dengan rata rata waktu proses deteksi 0.25 detik.

C. Analisa

Berdasarkan pengujian tingkat akurasi dari ke tiga skenario pada Tabel 4.1.1, Tabel 4.1.2, Tabel 4.1.3 diperoleh dari total 15 gambar tingkat akurasi sistem ada pelanggaran atau tidak adanya pelanggaran pengendara motor dengan menggunakan metode Yolo Tiny3 sebesar 100% dengan rata rata waktu proses 0.2 detik.

Berdasarkan pengujian tingkat presisi dari ke tiga skenario pada Tabel 4.4 , Tabel 4.5, dan Tabel4.6, diperoleh dari total 75 gambar tingkat presisi tertinggi deteksi helm yang didapatkan sebesar 88% skenario 1 di Jalan Raya Banjaran. Dan

untuk *non helmet* diperoleh tingkat presisi tertinggi deteksi *non helmet* yang didapatkan sebesar 100% skenario 3 di Jalan Raya Batununggal karena ada 1 pengendara motor saja yang terdeteksi tidak menggunakan helm dari 25 gambar. Berdasarkan pengujian tingkat presisi dari 3 skenario pada tabel dibawah ini diperoleh dari total 75 gambar rata-rata tingkat presisi tertinggi deteksi helm dan *non helmet* diperoleh rata-rata presisi sistem tertinggi 88% dan untuk rata-rata tingkat presisi terendah deteksi helm dan *non helmet* diperoleh 76%

TABEL
7 TOTAL AKURASI PENGUJIAN 3 SKENARIO HELMET DAN NON HELMET

No	Skenario Pengujian	Tingkat Presisi		Total Presisi Sistem
		Helmet	Non Helmet	
1	Skenario 1	88%	88%	88%
2	Skenario 2	80%	72%	76%
3	Skenario 3	65%	100%	82.5%

$$\text{Rata-Rata Presisi Sistem} = \text{Tingkat presisi helm} + \text{Tingkat presisi non helm} / 2 \quad (4.7)$$

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil perancangan, pengujian, dan analisa yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Metode Yolo Tiny3 (*You Only Look Once*) dapat digunakan untuk mendeteksi pengendara motor yang menggunakan helm dan tidak menggunakan helm dengan baik. Akan tetapi Yolo Tiny3 yang digunakan masih mengalami kesalahan ketika gambar sedikit buram atau objek terlalu jauh dan tidak dapat mendeteksi objek helm dan *non helmet*. Dan juga mengalami kesalahan ketika 2 objek berdekatan, karena tidak dianggap dalam 1 region helm
2. Dari 3 titik pengujian atau skenario dengan dataset yang digunakan berjumlah 15 citra didapatkan tingkat akurasi sistem helm dan *non helmet* pengendara motordengan menggunakan metode Yolo Tiny3 sebesar 100% dengan rata rata waktu proses 0.2 detik.
3. Dari 3 titik pengujian atau skenario didapatkan tingkat presisi sistem helm dan *non helmet* pengendara motor dengan menggunakan metode Yolo Tiny3 sebesar 88% dengan rata rata waktu proses 0.2 detik.

4. Pengambilan angle kamera berpengaruh

B. Saran

Berdasarkan hasil pembangunan Proyek Akhir ini, dapat disampaikan beberapa saran untuk pengembangan selanjutnya yaitu :

1. Mengembangkan metode / menggunakan metode lain agar dapat mengatasi objek yang kecil atau berdempetan
2. Mengembangkan sistem yang mampu mendeteksi pelanggar serta mengenali plat nomer kendaraan pengendara sepeda motor
3. Penambahan dataset yang lebih banyak dan bervariasi untuk mendapatkan filter *bounding box* yang lebih baik

REFERENSI

- [1] H. Bahtiar and I. K. Somawirata, "Sistem Pendeteksi Helm Yang Dikenakan Pengendara Sepeda Motor Untuk Safety Riding Berbasis Raspberry Pi," *Jurnal Informatika*, vol. 1, no. 1, pp. 1-9, 2020.
- [2] . I. J. R. Zahir, MSc and I. B. Ernani, MURP, Helm Manual keselamatan jalan untuk pengambil keputusan dan praktisi, Jakarta: Global Road Safety Partnership, 2014.

- [3] 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)," *Majalah Ilmiah UNIKOM*, vol. 17, no. 1, pp. 69-76, 2020.
- [4] H. H. Al Asyhar, S. A. Wibowo and G. Budiman, "Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video," *e-Proceeding of Engineering*, vol. 7, no. 2, pp. 3631-3638, 2020.
- [5] A. K. Gunadi and E. Setyati, "Deteksi Helm pada Pengguna Sepeda Motor dengan Metode Convolutional Neural Network," *Jurnal Informatika*, vol. 1, no. 1, pp. 1-7, 2020.
- [6] S. Eliyas, K. Swaathi, D. Ranjana and A. Harshavardhan, "Helmet, Violation, Detection Using Deep Learning," *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 2, pp. 5173-5178, 2020.
- [7] E.H. Purwanto, "Signifikansi Helm Sni Sebagai Alat Pelindung Pengendara Sepeda Motor Dari Cedera Kepala," *Jurnal Standardisasi*, vol. 17, no. 1, pp. 31-46, 2015.
- [8] H. BSN, "Berlaku di Indonesia, Ini Standar Helm yang Sesuai SNI," Badan Standarisasi Nasional, Februari 10 2020. [Online]. Available: <https://bsn.go.id/main/berita/detail/10868/berlaku-di-indonesia-ini-standar-helm-yang-sesuai-sni#:~:text=persyaratan%20sebagai%20berikut%20%3A-1.,bawah%20dari%20dudukan%20bola%20mat>. [Accessed 27 Juli 2022].
- [9] T. Susim and C. Darujati, "Pengolahan Citra Untuk Pengenalan Wajah (Face Recognition) Menggunakan Opencv," *Sosial Teknik*, vol. 2, no. 3, pp. 1-5, 2021.
- [10] M. Dasgupta, O. Bandyopadhyay and S. Chatterji, "Automated Helmet Detection for Multiple Motorcycle Riders using CNN," *Conference on Information and Communication Technology (CICT)*, vol. 1, no. 1, pp. 1-4, 2019.
- [11] A. Penggunaan YOLO, Bandung: Telkom University, 2020.
- [12] D. Ramadhan, *Klasifikasi Tumor Otak Menggunakan Convolutional Neural Network (CNN)*, Malang: Universitas Muhammadiyah Malang, 2020.
- [13] H. Mulyawan, M. . Z. H. Samsono and S. , "Identifikasi Dan Tracking Objek Berbasis Image Processing Secara Real Time," *Jurnal Elektronika*, vol. 1, no. 1, pp. 1-5, 2020.
- [14] F. Jalled and I. Voronkov, "Object Detection Using Image Processing," *Jurnal Informatika*, vol. 1, no. 1, pp. 1-6, 2016.
- [15] "A Hybrid Approach for Helmet Detection for Riders Safety using Image Processing, Machine Learning, Artificial Intelligence," *International Journal of Computer Applications*, vol. 182, no. 37, pp. 50-55, 2019.

