

DETEKSI PENGENDARA MOTOR TANPA MENGGUNAKAN HELM DENGAN ALGORITMA DEEP LEARNING YOLO

Khusnul Khoiriyah¹), Mochamad Fauzi Achmad Aji Armawan²)

¹Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma Jakarta

²Prodi Sistem Informasi, Fakultas Teknologi, ITB Swadharma Jakarta

Correspondence author: K. Khoiriyah, khusnul@swadharma.ac.id, Jakarta, Indonesia

Abstract

A helmet is a protective device that is worn on the head and made of metal or other hard materials. Indonesia is a country that requires motorcyclists to wear helmets. Lack of public awareness in driving using a helmet can endanger themselves and others. For this reason, an information technology system is necessary to monitor traffic activities 24 hours a day. This study designed an application to detect the use of helmets and classify motorists using helmets or not. The method used in detecting objects in the head area uses the Convolutional Neural Network (CNN) method with the You Look Only Once (YOLO) Algorithm. This system is able to detect violations committed by motorcycle riders. The accuracy of detecting helmet use with the training dataset as a test method of model evaluation produces predictions with an average accuracy rate of 89.04%, and the avg_loss training yields a rate of 1.2%.

Keywords : helmet detection, yolo algoritm, motorcycle riders

Abstrak

Helm adalah alat perlindungan yang dikenakan di kepala dan biasanya terbuat dari metal atau bahan keras lainnya, Indonesia termasuk negara yang mewajibkan pengendara sepeda motor menggunakan helm. Kurangnya kesadaran masyarakat dalam berkendara menggunakan helm bisa membahayakan diri sendiri dan orang lain. Untuk itu diperlukan sebuah sistem teknologi informasi yang mampu mengawasi aktifitas lalu lintas selama 24 jam. Penelitian ini merancang suatu aplikasi untuk mendeteksi helm dan mengklasifikasikan pengendara bermotor menggunakan helm atau tidak. Metode yang digunakan dalam pendeteksian objek di area kepala menggunakan metode *Convolutional Neural Network* (CNN) dengan Algoritma *You Look Only Once* (YOLO). Sistem ini diharapkan mampu mendeteksi pelanggaran yang dilakukan oleh pengendalian sepeda motor. Keakuratan pendeteksi penggunaan helm dengan dataset training yang merupakan metode pengujian dari evaluasi model menghasilkan prediksi dengan angka akurasi rata rata yaitu 89.04% dan training *avg_loss* menghasilkan angka 1.2%.

Kata Kunci : pendeteksi penggunaan helm, algoritma yolo, pengendara motor

A. PENDAHULUAN

Helm adalah alat perlindungan yang dikenakan di kepala dan biasanya terbuat dari metal atau bahan keras lainnya seperti kevral, serat resin, atau plastik (Hanafi, 2020). Di berbagai negara, helm wajib digunakan bagi pengendara sepeda motor. Indonesia termasuk negara yang mewajibkan pengendara sepeda motor harus memakai helm, untuk menjaga kepala dari benturan keras saat terjadi kecelakaan.

Undang-undang No.22 Tahun 2009 Pasal 57 Ayat 1 dan 2, setiap kendaraan bermotor yang dioperasikan di jalan wajib dilengkapi dengan perlengkapan kendaraan bermotor (1), perlengkapan sebagaimana di maksud Ayat (1) bagi sepeda motor berupa helm standar nasional Indonesia (2). Disebutkan juga dalam undang undang Pasal 106 Ayat 8 bahwa setiap orang yang mengemudikan motor dan penumpang sepeda motor wajib mengenakan helm (Presiden RI, 2009). Paparan undan-undang tersebut dapat di pahami bahwa dalam peraturan tersebut diatur tentang kewajiban seluruh pengendara untuk menggunakan helm. Namun tidak dapat dipungkiri setiap aturan yang dibuat pasti mempunyai tantangan didalam penerapannya, kurangnya kesadaran masyarakat di dalam berkendara tanpa menggunakan helm bisa membahayakan diri sendiri dan orang lain.

Dari permasalahan tersebut diperlukan sebuah sistem teknologi informasi yang mampu mengawasi aktivitas lalu lintas selama 24 jam (Purwanto et al., 2022). Sistem ini diharapkan mampu mendeteksi pelanggaran yang dilakukan oleh pengendara sepeda motor. Oleh karena itu, Dalam penelitian ini akan dibangun program komputer untuk mendeteksi penggunaan helm dan mengklasifikasikan pengendara bermotor apakah terdeteksi menggunakan helm atau tidak.

Object detection menentukan keberadaan suatu objek dan ruang lingkupnya serta lokasi pada sebuah gambar (Setyadi et al., 2022). Hal

ini dapat diperlakukan sebagai pengenalan objek kelas dua, dimana satu kelas mewakili kelas objek dan kelas lain mewakili kelas non-objek. Deteksi objek dapat dibagi lagi menjadi *soft detection* dan *hard detection* (Luthfillah Ahmad et al., 2021). *Soft detection* hanya mendeteksi adanya objek sedangkan *hard detection* mendeteksi adanya objek serta lokasi objek. Pendeteksian objek merupakan suatu tugas yang sangat penting dalam sebuah pengolahan citra digital. Deteksi objek berhubungan dengan penentuan identitas sebuah objek yang sedang diamati dalam sebuah citra dari sekumpulan tanda yang sudah diketahui. Hal ini berarti untuk mendeteksi sebuah objek tertentu dengan akurat, diperlukan proses berulang yang berguna untuk mengeliminasi objek-objek yang tak ingin dideteksi, yang pada akhirnya akan menyisakan gambaran objek yang diinginkan.

Program yang digunakan yaitu *Framework Neural Network Darknet* (Hanafi, 2020) dengan bahasa pemrograman C, *Bounding box* dengan bahasa pemrograman Python (Richo et al., 2021) dengan metode *Convolutional Neural Networks* (Septiana et al., 2020; Setyadi et al., 2022) pada Algoritma YOLO untuk mendeteksi objek pengendara yang tidak menggunakan helm.

Algoritma *You Look Only Once* (YOLO) akan membagi gambar yang dimasukkan menjadi grid berukuran $S \times S$, dimana nilai S adalah 7 dengan Input gambar berukuran 448×448 (Ariyoga et al., 2021). Untuk mendapatkan *bounding box*, akan dilakukan konvolusi dari inputan gambar, sehingga hasil akhirnya akan mendapat ukuran *bounding box* sebesar $S \times S \times B * (5 + C)$ dimana B adalah banyaknya *bounding box* (umumnya 2) dalam 1 grid dan C adalah banyaknya class yang dapat diklasifikasi. Nilai B dikalikan dengan 5 karena sebuah *bounding box* memiliki 5 nilai yang perlu disimpan, koordinat x , koordinat y , lebar (*width*), tinggi (*height*), dan *confidence score* (nilai probabilitas *bounding box* yang

bersangkutan memiliki sebuah objek). *You Only Look Once* (YOLO) merupakan sistem deteksi yang berbasis *Convolutional Neural Network*. Data yang akan digunakan untuk pengujian merupakan rekaman video pengendara selama *day by day* dari tanggal 7 November 2021 sampai dengan 13 November 2021 setiap pukul 15:00 PM sampai dengan 18:00 PM dan akan dilakukan perbandingan akurasi per hari yang didapat dari lokasi penelitian di Jalan Gedong Panjang.

Hasil penelitian terdahulu yang menjadi landasan atau acuan dalam melakukan penelitian ini diantaranya penelitian Yusuf Umar Hanafi, dalam penelitiannya membuat sebuah sistem pendeteksi penggunaan helm pada pengendara bermotor. Metode yang digunakan untuk proses deteksi penggunaan Helm menggunakan YOLO (Hanafi, 2020). Dataset yang digunakan berbagai macam kondisi atribut yaitu menggunakan masker, menggunakan hoodie, menggunakan Jilbab, dan Menggunakan jas Hujan. Lalu data Input menggunakan data video rekaman IP Camera lalu lintas. Lalu dataset di training menggunakan metode CNN (*Convolutional Neural Network*) Dan Hasil deteksi berupa sistem perhitungan dan pengambilan gambar pengendara motor yang tidak menggunakan helm, dan pengujian performa *Mean Average Precision* dengan mengukur nilai ada 2 tipe kondisi yaitu Kondisi Kecerahan tinggi dengan nilai mAP 53.63 % dan Kondisi Hujan 51.19 %. Hasil Grafik *Average Loss* terhadap Iterasi pada Proses Training YOLOv3 yaitu $avg_loss = 0.3035$ terhadap angka iterasi = 6000.

Berikutnya penelitian oleh Dimas Ariyoga, Ridho Rahmadi dan Rian Adam Rajagede dkk Dalam penelitiannya membuat sistem pelanggaran lalu lintas yaitu dengan cara mendeteksi objek kendaraan bersama dengan objek pelanggarannya. Misalnya pada sepeda motor, pelanggaran akan dideteksi dengan mengklasifikasikan pengendara sepeda motor yang menggunakan helm dan pengendara sepeda motor yang tidak

menggunakan helm. Penelitian ini menggunakan YOLOv3 sebagai algoritma untuk deteksi objek. Pendeteksian dilakukan dengan Neural Network dan model Object Detection yang digunakan untuk mengklasifikasikan objek bergerak ke dalam kelas yang berbeda, sehingga klasifikasi kendaraan dan pelanggarannya dapat dilakukan. Proses Metode Deep Learning menunjukkan hasil dengan berbagai Metode. Salah satunya Hasil YOLOv3 menunjukkan bahwa deteksi beberapa pelanggaran lalu lintas dari satu sumber Input dapat diarsipkan. Sistem memiliki akurasi 97.67% untuk deteksi jumlah kendaraan dan akurasi 89.24% untuk mendeteksi kecepatan kendaraan (Ariyoga et al., 2021).

Sedangkan penelitian Stephen Ekaputra Limantoro, dalam penelitiannya membuat sistem *Object Detection* yang bisa digunakan di dalam mobil yaitu disebut *Dash Cam*. *Dash Cam* sebelumnya hanya untuk merekam video saja sehingga *Dash Cam* perlu diberi kecerdasan untuk mendeteksi objek pada Pengendara Motor. Penelitian ini menggunakan Metode *Convolutional Neural Network* (CNN) yang mampu untuk mendeteksi Pengendara sepeda motor. CNN digunakan untuk deteksi pengendara sepeda motor dari video *dash cam* yang beresolusi 1080 HD. *Dash cam* ditempelkan pada dashboard mobil dan merekam video di jalan raya Surabaya. Pengambilan citra pengendara sepeda motor dilakukan pada saat posisi pengendara sepeda motor tegak lurus ke depan dan pada saat cerah/siang hari. Metode CNN berhasil mengklasifikasikan objek dengan f1-score 0,94. Sliding window dan heat map digunakan dalam Penelitian ini untuk mencari area/region pengendara sepeda motor. Dua eksperimen dilakukan dalam Penelitian ini. Tujuan dari Penelitian ini adalah mencari kombinasi arsitektur dan parameter CNN yang terbaik. Eksperimen pertama terdiri atas tiga bobot yang sudah dilatih sedangkan eksperimen kedua mengandung satu bobot yang sudah dilatih. Kinerja bobot terhadap data pengujian pada

eksperimen 1 dan eksperimen 2 diukur melalui f1-score masing-masing 0,977, 0,988, 0,989, dan 0,986. Dari hasil percobaan menggunakan sliding window, eksperimen 2 memiliki tingkat error yang lebih rendah untuk memprediksi pengendara sepeda motor dibandingkan dengan eksperimen 1 karena data pelatihan pada eksperimen 1 lebih banyak jumlahnya dan mengandung citra yang lebih variatif.

Tujuan penelitian ini untuk membuat sebuah sistem yang bisa mendeteksi pengendara motor yang tidak menggunakan helm.

B. METODE PENELITIAN

Metode ini digunakan sebagai pedoman dalam pelaksanaan penelitian supaya memperoleh hasil yang sesuai dengan tujuannya. Dalam melakukan penelitian ini menggunakan dua metode pengumpulan data.

Metode pertama yaitu Observasi atau pengumpulan data melalui pengamatan langsung ditempat kejadian atau observasi partisipan, Penelitian dilakukan di Jalan Raya Gedong Panjang. Observasi dilakukan dengan menggunakan rekaman video pengendara yang melintas di lokasi penelitian setiap hari dari tanggal 7 November 2021 sampai dengan 13 November 2021 setiap pukul 15:00 PM sampai dengan 18:00 PM

Tahapan pengumpulan data melalui studi pustaka, penulis mengumpulkan dan mencari referensi yang dibutuhkan untuk mengumpulkan informasi dalam penelitian ini. Pencarian referensi didapat dari jurnal, buku, literatur sejenis, skripsi yang ada dan dari internet.

C. HASIL DAN PEMBAHASAN

Penelitian ini yang dilakukan berkaitan dengan *image & video processing* yang memanfaatkan *Deep learning* untuk mempelajari ciri khusus dari setiap objek

terutama dalam mendeteksi objek pengendara motor yang tidak menggunakan helm. Pada penelitian *object detection* terdapat dua proses yaitu *Train Without Helmet Detector* dan *Apply Without Helmet Detector*. Gambar 1 merupakan diagram *training* dengan menggunakan dataset image pelanggaran pengendara motor, dengan dataset tersebut akan membuat deteksi pengendara yang tidak menggunakan helm. Dataset tersebut nanti akan digunakan untuk memuat melakukan deteksi pelanggaran, lalu mengklasifikasikan setiap kendaraan motor yang tidak menggunakan helm ditunjukkan pada gambar 2.



Gambar 1. *Train Without Helmet Detector*



Gambar 2. *Apply Without Helmet Detector*

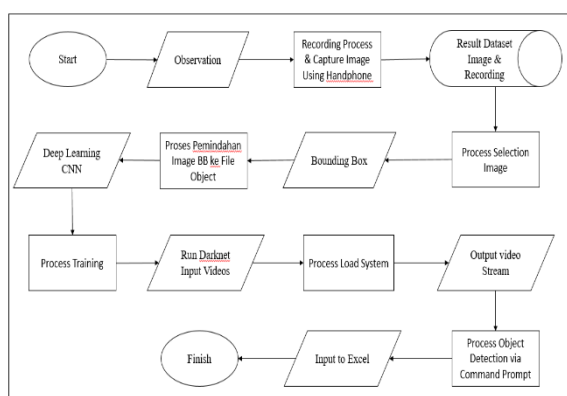
Pada penelitian *deteksi tanpa helm* menggunakan *image processing* dengan sistem *preprocessing* gambar sebelum melanjutkan ketahap berikutnya. Dalam langkah *pre-processing* gambar, akan dilakukan dengan metode proses convolusi yang dari 1 gambar dipecah menjadi gambar-gambar kecil seperti di grid yang kemampuan bisa mengenali sebuah objek, dimanapun posisi objek yang sama tersebut muncul pada sebuah gambar yang disebut menggunakan metode *Training Convolutional Weights*. *Deep Learning Architecture* mempelajari hal penting yaitu fitur *non-linier* dari sampel yang diberikan. Kemudian, mempelajari arsitektur yang digunakan untuk memprediksi sampel yang sebelumnya tidak terlihat. Oleh sebab itu, *Deep Learning Architecture* sangat tergantung pada CNN. Pada *Deep Learning Architecture* ada beberapa tahapan yang harus dilakukan yaitu, *Dataset Collection*, *Training*, dan *Deployment*. Pada *Dataset Collection* merupakan tahapan pengumpulan data dari beberapa sumber yang berbeda, data yang dikumpulkan terdapat 2 jenis yaitu, data pelatihan dan data pengujian model. Dari

data yang telah dikumpulkan diperoleh 500 Dataset *image* dan Data Pengujian model 320.000 *image*.

Training merupakan *processing* yang terfokus untuk memuat dataset pengendara tanpa helm dari penyimpanan dataset, melatih model dengan menggunakan instrument dari *Convolutional Neural Network*. Dari dataset ini, akan membuat serial *Without Helmet Detection* pada penyimpanan dataset.

Deployment merupakan *processing* setelah deteksi tanpa helm yang sudah melalui *training* dataset, selanjutnya akan dapat melanjutkan untuk memuat detektor tanpa helm, melakukan deteksi pengendara dengan memberikan *bounding box* yang akan menyeleksi di area sekitar kepala, kemudian akan terfokus pada *bounding box* yang akan memproses sistem ROI (*Region Of Interest*), kemudian *detect facial landmark* yang akan mendeteksi titik-titik yang ada pada area kepala, setelah proses tersebut maka sistem akan mengklasifikasikan setiap kepala yang terdeteksi tidak menggunakan helm, menggunakan helm, mendeteksi orang dan menggambar *bounding boxes* di sekitar objek deteksi dengan nama kelas yang terdeteksi.

Metode Pengolahan Data



Gambar 3. Proses pengolahan Data

Pada Proses pengolahan Data yang dilakukan dimulai dengan teknik Observasi melakukan pengamatan secara langsung di lapangan sehingga di proses dengan program *object detection*.

Sistem yang dibangun adalah sistem deteksi objek pengendara motor dengan menggunakan metode *Convolutional Neural Network (CNN)*. Pada dasarnya terbagi dalam dua tahapan, yaitu tahap *training* dan tahap *testing*.

Tahap *training* akan menggunakan data input yang berasal dari dataset *train* yang akan di label menggunakan *Bounding Box* yang nantinya akan disimpan pada folder Obj. Data *training* menggunakan 3 *class object* yang dinamakan *No Helmet*, *Helmet* dan *Person* dengan data gambar yang berisi 560 foto kerumunan pengendara bermotor yang tidak menggunakan helm dan menggunakan helm dengan ukuran foto beragam dan foto berekstensi .JPG.

Tahap *testing* akan menggunakan hasil video rekaman *handphone* dengan data *day by day* sebagai uji coba proses *object detection* melalui data gambar yang sudah di proses *training* menggunakan *Convolutional Neural Network (CNN)*. Proses Pengambilan data *training* dilakukan menggunakan gambar yang didapat dari tempat penelitian di daerah Gedong Panjang Jakarta Utara. Pada *dataset* yang dikumpulkan memiliki beberapa atribut tambahan yang tidak menggunakan helm yaitu menggunakan masker, kerudung, topi, kacamata, peci dan kupluk *hoodie*. Dataset yang menggunakan helm dengan jenis nya yaitu helm tanpa kaca, helm *full face*, dan helm *Half Face*.



Gambar 4. Dataset Training No Helmet berbagai macam atribut



Gambar 5. Dataset *Training Helmet* tanpa kaca, *half face*, dan *full face*

Proses Pengambilan data *testing* dilakukan secara langsung yang didapat dari tempat observasi langsung menggunakan kamera handphone sebagai jalur alternatif pengganti cctv untuk melakukan proses rekam aktifitas kendaraan dari hari ke hari dengan file yang berekstensi MOV dan MP4 dengan durasi 3 jam per-harinya.

Name	Date	Type	Size
Senin 8 November ...	11/8/2021 4:53 PM	MOV File	12,843,743 ...
Senin 8 November ...	11/8/2021 3:20 PM	MOV File	15,569,135 ...
Selasa 9 November ...	11/9/2021 4:42 PM	MOV File	15,420,161 ...
Selasa 9 November ...	11/9/2021 3:04 PM	MOV File	15,343,515 ...
Sabtu 10 November...	11/13/2021 5:42 PM	MOV File	2,306,632 ...
Sabtu 10 November...	11/13/2021 4:25 PM	MOV File	13,093,920 ...
Sabtu 10 November...	11/13/2021 2:42 PM	MOV File	15,409,040 ...
Rabu 10 November ...	11/10/2021 4:52 PM	MOV File	10,274,562 ...
Rabu 10 November ...	11/10/2021 3:22 PM	MOV File	15,504,502 ...
Rabu 10 November ...	11/11/2021 1:47 AM	MP4 Video File (VL...	2,076,542 ...
Minggu 14 Novemb...	11/14/2021 4:22 PM	MOV File	15,412,851 ...
Minggu 14 Novemb...	11/14/2021 3:05 PM	MOV File	13,011,987 ...
Minggu 14 Novemb...	11/14/2021 2:44 PM	MOV File	2,397,211 ...
Kamis 11 Novembe...	11/11/2021 4:02 PM	MOV File	15,494,504 ...
Kamis 11 Novembe...	11/11/2021 2:32 PM	MOV File	15,410,596 ...
Jumat 12 November...	11/12/2021 4:14 PM	MOV File	6,448,972 ...
Jumat 12 November...	11/12/2021 2:39 PM	MOV File	16,269,380 ...

Gambar 6. Dataset rekaman

Pada Gambar 6. merupakan Dataset rekaman yang akan dilakukan sebagai data testing untuk melakukan object detection.

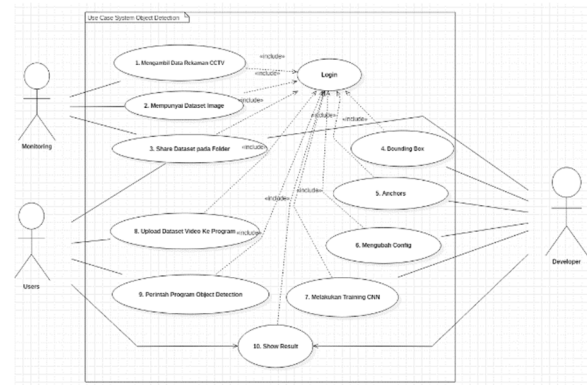
Metode Pemrosesan Data

Data *Preparation* dan proses model dilakukan melalui dengan skema seperti berikut:



Gambar 7. Model Processing Data

Dalam Pembuatan sistem *Object Detection* dengan metode *Convolutional Neural Network* (CNN) diperlukan model interaksi, untuk memberikan gambaran umum kepada *user* setelah Perancangan *Object Detection* dapat di implementasikan di cctv dan Perancangan tersebut menggunakan *Use Case Diagram*.



Gambar 8. Use Case Diagram

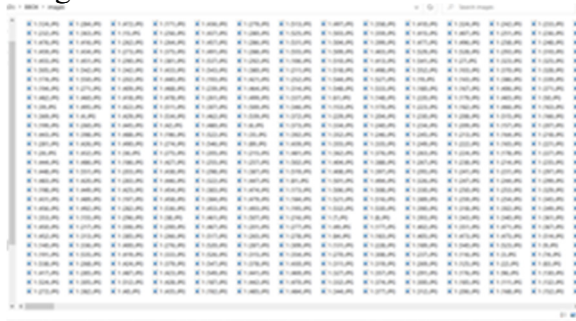
Library Preparation merupakan sekumpulan kode yang memiliki fungsi-fungsi tertentu dan dapat dipanggil kedalam program lain. *Library* dibuat untuk mempermudah dalam membangun sebuah aplikasi. *Library* yang harus disiapkan untuk melakukan install dalam membantu proses pengembangan aplikasi object detection yaitu :

OpenCV_CUDA adalah Bagian penting dari *Computer Vision* adalah pemrosesan gambar, area yang awalnya dirancang untuk akselerator grafis. Digunakan untuk mengandaikan komputasi paralel besar-besaran dan sering kali secara alami memetakan ke arsitektur GPU.

Numpy adalah *library* yang digunakan untuk pendukung *Bounding Box*, yang ditugaskan untuk membuat *N-Dimensional Array*.

Pandas adalah *library* yang dibutuhkan untuk pendukung *Bounding Box*, yang ditugaskan untuk membuat *labelling*, membuat tabel, membuat struktur data, dan membuat *ekstension* file dalam berbagai format, dan lain lain. *Data Preparation* adalah suatu proses yang dilakukan untuk

menyiapkan data. Diantaranya yaitu menempatkan data pada *folder image*. Dan juga menyiapkan *library* yang digunakan untuk pemrosesan data. Pertama *dataset* disimpan di dalam *folder bounding box* yang akan di proses *labelling* didalam folder *image*:



Gambar 9. Dataset Image Preparation

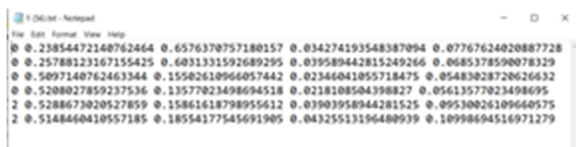
Project Workflow (AlurKerja Proyek)

Load Dataset, Dataset pada gambar yang sudah disiapkan / *Data Preparation* akan dipindahkan ke dalam *Bounding Box* pada folder *images*.



Gambar 10. Bounding box labeler

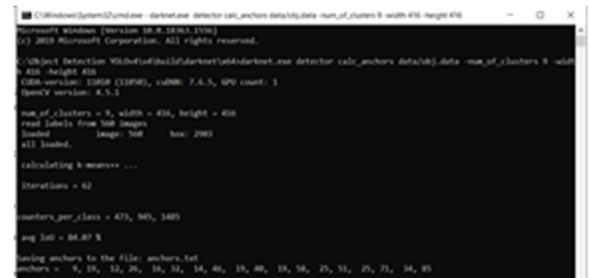
Proses pembuatan kotak pembatas / *labelling* pada gambar untuk melakukan *custome object detection*, dengan menggunakan 3 *class object* yaitu *class 0: No Helmet, class 1: Helmet, Class 2: Person*.



Gambar 11. Koordinat Image Labeler

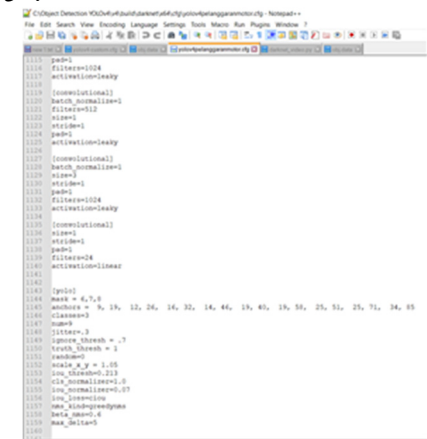
Dibalik proses *selection object* pada *bounding box*, akan tersimpan otomatis yang

berisi koordinat ($\langle x_center \rangle \langle y_center \rangle \langle width \rangle \langle height \rangle$) di dalam ekstensi file nama_gambar.txt dengan masing-masing dataset. Untuk Memproses data *image* yang sudah dilakukan proses *labelling*, akan dipindahkan ke *file directory program* aplikasi *darknet* untuk melakukan proses tahap selanjutnya.



Gambar 12. Anchors Calculation

Anchors merupakan salah satu parameter terpenting yang disesuaikan untuk meningkatkan kinerja pada dataset. *Processing Anchors* yaitu menghitung dan membaca mengenai dataset yang sudah di label untuk melakukan proses pengukuran jangkar kotak pembatas. Dengan menggunakan ukuran dimensional 416 x 416 pixel, Jumlah dataset *image* yaitu 560, dan total image yang sudah di label sebanyak 2903. Lalu ada metrik evaluasi mengukur nilai rata-rata pada dataset sebelum melakukan training yaitu Avg IoU = 84.07%. Ukuran kotak jangkar pada dataset dari kecil hingga besar yaitu “9, 19, 12, 26, 16, 32, 14, 46, 19, 40, 19, 58, 25, 51, 25, 71, 34, 85”.



Gambar 13. Configuration Model YOLO

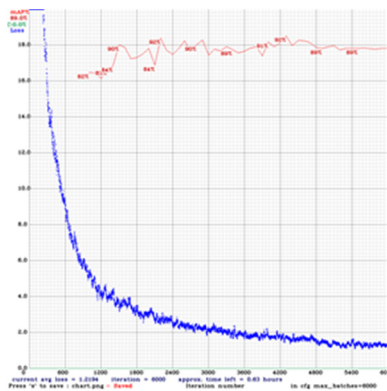
Pada Gambar 13, yaitu memodifikasi *file* *cfg* untuk menyesuaikan konfigurasi model yang terkait dengan mengubah kelas, atau lainnya yang berasal darinya dan juga mengaktifkan flag acak untuk proses persiapan pengujian dalam membangun *Convolutional Neural Network*.

```

C:\Windows\system32\cmd.exe - darknet.exe detector train data/obj.data/obj.yolo4pelanggaranmotor.cfg yolov4.conv.137
[... code for running darknet.exe ...]
[... output showing training progress ...]
[... layer calculations ...]
  
```

Gambar 14. Start Training CNN Method

Membangun Model dengan menggunakan *Convolutional Neural Network* (CNN) melalui sistem perintah yaitu “darknet.exe detector train data/obj.data/cfg/yolov4pelanggaranmotor.cfg yolov4.conv.137” yang bertujuan untuk melatih model YOLOv4 dalam kerangka kerja darknet dengan file *weight* “yolov4.conv.137”. Proses training tersebut membutuhkan waktu hingga 48 Jam *Full Time* untuk selesai melakukan pelatihan model.



Gambar 15. Grafik Perbandingan *Iteration* dan *Loss*

Proses pelatihan dilakukan sebanyak 6.000 batch yang menghasilkan sebuah grafik diatas merupakan hasil dari training pada dataset yaitu nilai perbandingan *Loss* dengan nilai *Iteration*, bahwa semakin turun *loss* maka semakin bagus hasil kinerja dataset untuk di *detection*. Model akhir yang dibangun memiliki nilai *avg_loss* 1.2% dan nilai validasi mAP (*mean average precision*) yaitu 89%. Nilai mAP menunjukkan bahwa tingkat akurasi pengenalan objek pada pengendara motor menggunakan model YOLO telah dilatih sebesar 89%.

```

(next mAP calculation at 6000 iterations)
[... accuracy statistics ...]
[... mAP calculation ...]
[... detection layer results ...]
[... class statistics ...]
[... IOU threshold ...]
[... mean average precision ...]
[... total detection time ...]
[... save points ...]
  
```

Gambar 16. *Result Modelling Convolutional Neural Network*

Setelah Peneliti selesai melakukan pelatihan model, maka hasil pelatihan bobot *darknet* secara otomatis tersimpan di dalam folder *weights* di setiap 1.000 iterasi, *darknet* juga menyimpan model terbaik dengan ekstensiyolov4pelanggaranmotor_final.weights. T-nya didasarkan pada kinerja set validasi.

Hasil metode pengujian dari Evaluasi model yaitu : mAP (*mean Average Precision*) dengan bobot *last iteration* yaitu 89.04%, dan angka rata-rata terbaik 92.45%. AP (*Average Precision*) di setiap kelas *object* yaitu *No Helmet* 85,31%, *Helmet* 82,31% *Person* 99,27% . Hasil Perbandingan *Recall* dan *Precision* di setiap kelas objek yaitu (*No Helmet* TP = 462, FP 219), (*Helmet* TP = 933, FP = 478), (*Person* TP = 1475, FP = 246).

Dataset Testing



Gambar 17. Testing Deteksi Objek

Setelah selesai melakukan proses *training*, langkah *testing* merupakan langkah terakhir untuk melakukan tes. Data *testing* merupakan data video rekaman aktifitas kendaraan yang sudah dipersiapkan untuk di deteksi, Lalu data rekaman tersebut di proses dan melakukan analisa *day by day* untuk mengetahui di hari apa saja yang paling banyak pelanggaran berdasarkan FPS (*Frame Per Second*) yang terdeteksi di bagian output sistem perintah.



Gambar 18. Hasil Perbandingan Tanpa Helm dan Menggunakan Helm

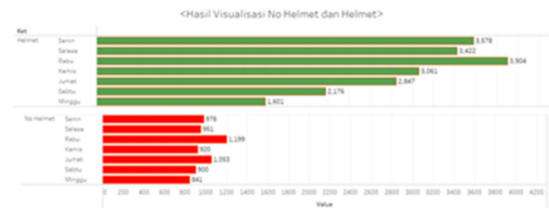
Pada Gambar 18, hasil perbandingan yang sudah dilakukan proses *Cleansing* data dan dijadikan data preparation per-harinya agar dapat di visualisasi, yaitu dengan total rincian sebagai berikut:

Data *Helmet*

Senin = 210,745 *Frame Per Second*
 Selasa = 189,113 *Frame Per Second*
 Rabu = 266,074 *Frame Per Second*
 Kamis = 134,708 *Frame Per Second*
 Jumat = 145,104 *Frame Per Second*
 Sabtu = 80,431 *Frame Per Second*
 Minggu = 33,060 *Frame Per Second*

Data *No Helmet*

Senin = 38,972 *Frame Per Second*
 Selasa = 37,619 *Frame Per Second*
 Rabu = 53,025 *Frame Per Second*
 Kamis = 34,798 *Frame Per Second*
 Jumat = 47,192 *Frame Per Second*
 Sabtu = 27,626 *Frame Per Second*
 Minggu = 16,455 *Frame Per Second*



Gambar 19. Hasil Perbandingan *No Helmet* dan *Helmet* menggunakan alat *Counting*

Pada Gambar 19. Merupakan hasil perbandingan *No Helmet* dan *Helmet* dengan menggunakan alat *Counting* dengan rincian sebagai berikut:

Data *Helmet*

Senin = 3,578 menggunakan helm
 Selasa = 3,422 menggunakan helm
 Rabu = 3,904 menggunakan helm
 Kamis = 3,061 menggunakan helm
 Jumat = 2,847 menggunakan helm
 Sabtu = 2,176 menggunakan helm
 Minggu = 1,601 menggunakan helm

Data *No Helmet*

Senin = 978 tidak pakai helm
 Selasa = 961 tidak pakai helm
 Rabu = 1,199 tidak pakai helm
 Kamis = 920 tidak pakai helm
 Jumat = 1,053 tidak pakai helm
 Sabtu = 900 tidak pakai helm
 Minggu = 841 tidak pakai helm

Dapat dinyatakan bahwa perbandingan dengan data berdasarkan *Counting* dan *Frame Per Second* mengenai *No Helmet* dan *Helmet* dari tanggal 8 November 2021 sampai dengan 14 November 2021 yaitu masih banyak yang menggunakan helm di setiap harinya daripada yang tidak menggunakan helm. Namun dengan data *No Helmet*, masih banyak yang melakukan

pelanggaran di hari senin sampai sabtu, tetapi kita dapat menyimpulkan bahwa banyak yang melakukan pelanggaran tanpa helm di hari aktifitas kerja daripada di hari libur.

Pendekatan yang diusulkan untuk *helmet detection* dinyatakan dengan berbagai langkah implementasi yang terlibat dalam pendekatan ini. Diagram alir untuk ini disajikan dibawah yang menyatakan aliran keseluruhan dari pendekatan tersebut.

Dimulai dari persiapan dataset *image* dan *videos* yang didapatkan dari tempat penelitian kemudian disimpan dan di *load* dataset untuk melakukan proses *bounding box* dengan *library* yang sudah disediakan lalu memproses *custome detection* memilih area kepala dengan membedakan yang tidak memakai helm dan memakai helm, kemudian dari hasil tersebut mendapatkan file dengan ukuran koordinat gambar dan dilakukan proses *anchors* lalu data gambar akan dilakukan proses *training* dengan membuat model melalui *setting configuration* YOLO pada file tersebut.

Kemudian Model dimulai dengan pemuatan dataset untuk *helmet detection* dan *preprocessing* data dilakukan dengan algoritma *Convolutional Neural Networks* (CNN). Kemudian setelah dilakukan pemodelan data dilakukan *training* model dengan 6000 *batch* menggunakan 3 kelas objek dan membutuhkan waktu 48 jam *full time*, dan model tersebut keluar hasil grafik yang merupakan kinerja dari sebuah dataset gambar *helmet detection*. Fase pertama dari model ini adalah seperti yang dinyatakan dalam pernyataan di atas dan proses pelatihan dilakukan dengan menggunakan *framework* C dan *opencv*.

Setelah mendapatkan hasil model dari *helmet detection*, kemudian dataset video yang sudah dipersiapkan akan dilakukan proses *testing* untuk mengukur kinerja pada program dan dataset video sudah dipersiapkan dengan data 1 minggu untuk melakukan proses analisa dari *day by day* untuk mengetahui di hari apa saja yang banyak melakukan pelanggaran yang tidak

menggunakan helm dengan data FPS (*Frame Per Second*) dan membuat program aplikasi *helmet detector* agar bisa digunakan untuk mendeteksi pelanggaran helm pada pengendara motor.

Pengembangan dilakukan dengan mengacu kepada metode *Deep Learning* yang diakui secara internasional, sehingga dapat dikembangkan dan dioperasikan dengan baik secara operasional sudah layak memenuhi standar.

Kelayakan hukum dari aplikasi ini merupakan pengembangan Cabang AI yaitu salah satunya *Computer Vision* dan skrip pemrograman yang dibuat tidak mengandung unsur-unsur yang membahayakan pada perangkat keras maupun perangkat lunak, dan tidak melanggar hukum yang berlaku yang ditetapkan oleh pemerintah.

D. PENUTUP

Pendeteksian objek di area kepala dengan menggunakan metode *Convolutional Neural Network* (CNN) memiliki kemampuan dan kinerja yang sangat baik, hal ini terlihat berdasarkan pengujian yang dilakukan terhadap identifikasi objek tanpa helm dalam kondisi yang memungkinkan dapat mempengaruhi hasil pendeteksian helm dan tanpa helm. Keakuratan *helmet detection* dengan dataset training yang merupakan metode pengujian dari evaluasi model menghasilkan angka akurasi rata rata yaitu 89.04% dan training avg_loss menghasilkan angka 1.2%

Program *Darknet* ditambahkan dengan fitur *Counting People Object* untuk mengetahui jumlah objek berapa banyak orang yang melanggar dan tidak melanggar pada pengendara motor. Penelitian ini dapat dikembangkan dengan algoritma YOLO dengan versi yang berbeda atau dengan algoritma lainnya.

E. DAFTAR PUSTAKA

Ariyoga, D., Rahmadi, R., & Rajagede, R. A.

- (2021). Penelitian Terkini Tentang Sistem Pendeteksi Pelanggaran Lalu Lintas Berbasis Deep Learning : Sebuah Kajian Pustaka. *Automata*, 2(1).
- Hanafi, Y. U. (2020). *Deteksi Penggunaan Helm Pada Pengendara Bermotor Berbasis Deep Learning*.
- Luthfillah Ahmad, F., Nugroho, A., & Alfa Faridh Suni. (2021). Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19. *Edu Elekrika Journal*, 10(1), 13–18.
- Presiden RI. (2009). Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 Tentang Lalulintas Dan Angkutan Jalan. In <https://www.dpr.go.id>.
- Purwanto, E., Utomo, B. P. C., & Permatasari, H. (2022). Prototype sistem informasi monitoring penjualan. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(4), 761–768. <https://doi.org/10.25126/jtiik.202294880>
- Richo, R. A., Swastika, I. P. A., Permana S, P. T. H., & Dharma, E. M. (2021). Rancang Bangun Aplikasi Point of Sales Kasirin Dengan Terintegrasi Payment Gateway. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 10(2), 319. <https://doi.org/10.35889/jutisi.v10i2.657>
- Septiana, T., Puspita, N., Fikih, M. Al, & Setyawan, N. (2020). Face Mask Detection Covid-19 Using Convolutional Neural Network (Cnn). *Seminar Nasional Teknologi Dan Rekayasa (SENTRA) 2020*, 3, 27–32.
- Setyadi, A., Kallista, M., & Setianingsih, C. (2022). *Skripsi : Deteksi Social Distancing Dan Penggunaan Masker Di Restoran Dengan Algoritma Convolutional Neural Network (CNN)*. Bandung : Universitas Telkom.