

Pendeteksian Sepeda Motor Yang Melintasi Trotoar Menggunakan Algoritma YOLOv3

Eric Loudwyck¹⁾ Yulia Ery Kurniawati²⁾

Informatika, Fakultas Industri Kreatif Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta 13210

¹⁾ Email: ericloudwyck@gmail.com

²⁾ Email: yulia.kurniawati@kalbis.ac.id

Abstract: Indonesia ranks 10th as the most congested city in the world. To reduce traffic, private vehicle users have to switch to public transportation. When using public transportation, walking on the sidewalk is a common thing. However, the condition of pedestrian sidewalk sometimes become uncomfortable because there is a motorcyclist who crosses it due to traffic jams. This study aims to develop an application that can detect motorbikes crossing the sidewalk. So, it expects to reduce the number of motorcycle riders crossing the sidewalk. The method used in this study is YOLOv3 in the process of identifying motorcycles. The results of this study are software that can detect motorbikes crossing the sidewalk. By using YOLOv3, motorcycle detection can be carried out and produce video output with an average of 4 fps.

Keywords: convolutional neural network, object detection, traffic jam, yolov3

Abstrak: Indonesia menempati urutan ke-10 sebagai kota termacet di dunia. Untuk mengurangi kemacetan, tentu pengguna jalan harus beralih ke transportasi umum. Saat menggunakan transportasi umum, kegiatan berjalan kaki di trotoar adalah hal yang sering terjadi. Namun, kondisi trotoar pejalan kaki terkadang menjadi tidak nyaman karena ada pengendara sepeda motor yang melintasi trotoar karena macetnya jalan raya. Penelitian ini bertujuan untuk menghasilkan aplikasi yang dapat mendeteksi sepeda motor yang melintasi trotoar. Aplikasi ini diharapkan dapat mengurangi sepeda motor yang melintasi trotoar. Metode yang digunakan dalam penelitian ini adalah YOLOv3 dalam proses pendeteksian sepeda motor. Hasil penelitian ini berupa aplikasi yang dapat mendeteksi sepeda motor melintasi trotoar. Dengan menggunakan YOLOv3, pendeteksian sepeda motor dapat dilakukan dan menghasilkan output video dengan rata-rata 4 fps.

Kata kunci: convolutional neural network, kemacetan lalu lintas, pendeteksian objek, yolov3

I. PENDAHULUAN

Kesadaran masyarakat Indonesia akan keselamatan berlalu lintas masih sangat rendah. Menurut sumber data dari Kepolisian Republik Indonesia, rata-rata tiga orang meninggal dunia setiap jam akibat kecelakaan lalu lintas [1]. Berbagai pelanggaran sering dilakukan antara lain menerobos lampu merah, melawan arah, tidak menggunakan atribut lengkap, melintasi trotoar dan masih banyak lagi. Berbagai pelanggaran ini juga dilakukan baik oleh pengendara roda dua, roda empat, dan kendaraan besar lainnya. Hal ini tentunya menimbulkan

ketidaknyamanan berlalu lintas, kemacetan, dan angka kecelakaan yang masih tergolong tinggi.

DKI Jakarta menjadi kota termacet peringkat tujuh dari total 403 kota dalam 56 negara di dunia [2]. Seiring dengan jumlah kendaraan bermotor yang semakin banyak, pengawasan lalu lintas menjadi tidak efektif. Jumlah petugas kepolisian yang perlu dikerahkan juga tidak sedikit jika ingin menjadikan lalu lintas kota tertib seluruhnya. Selain itu, pelanggaran yang dilakukan juga bisa terjadi kapanpun bahkan diluar jam kerja pihak kepolisian. Maka dari itu diperlukan sebuah sistem pengawasan cerdas

untuk menjaga ketertiban lalu lintas agar lebih efektif.

Berdasarkan permasalahan tersebut, pada penelitian ini diperlukan sebuah solusi pembuatan sistem deteksi pengendara sepeda motor yang melintasi trotoar. Untuk melakukan pendeteksian objek bergerak, ada beberapa algoritma yang dapat dilakukan. Algoritma tersebut antara lain *Faster R-CNN (region-based convolutional neural network)*, *R-FCN (region-based fully convolutional networks)*, *SSD (single shot multibox detector)*, dan *YOLO (you only look once)*. Dalam perbandingan algoritma tersebut, didapatkan hasil akurasi *faster R-CNN* 70,4%, *R-FCN* 77,6%, *SSD* 74,9%, dan *YOLOv3* 76,8% [3]. Untuk FPS (*frames per second*) yang didapat, *faster R-CNN* memiliki 5 FPS, *R-FCN* 6 FPS, *SSD* 22 FPS, dan *YOLOv3* 40 FPS [3]. Dalam skenario lalu lintas, disamping akurasi yang baik, diperlukan FPS yang tinggi pula. Berdasarkan pertimbangan tersebut, maka *YOLOv3* dipilih sebagai algoritma dalam penelitian ini.

Dalam penelitian ini, sistem dilatih untuk dapat mengenali sepeda motor menggunakan algoritma *YOLOv3*. Untuk dapat mengenali kendaraan tersebut, sistem memerlukan *dataset* citra dari sepeda motor. Data tersebut kemudian diolah dengan algoritma *YOLO v3* sehingga menghasilkan *output* berupa model yang kemudian diimplementasikan ke kamera yang diharapkan mampu mengenali sepeda motor tersebut. Jika ada sepeda motor yang terdeteksi berada di trotoar pejalan kaki dan melintasi garis larangan, maka sistem akan memberikan pemberitahuan bahwa ada sepeda motor yang melintasi trotoar.

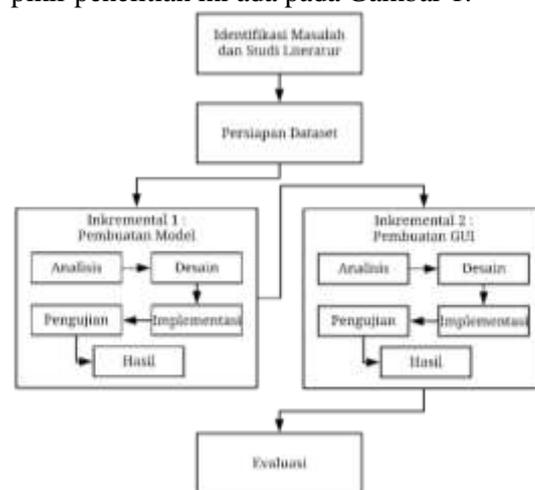
Fokus dalam penelitian ini adalah untuk mengetahui bagaimana cara sistem mendeteksi objek sepeda motor menggunakan algoritma *YOLOv3* dan bagaimana cara sistem menentukan jika ada sepeda motor yang melintasi trotoar. Tujuan dalam penelitian ini adalah mendeteksi pengendara sepeda motor yang melintasi trotoar menggunakan algoritma *YOLOv3* dan menghasilkan sebuah aplikasi

yang dapat memberikan pemberitahuan jika ada sepeda motor yang melintasi trotoar.

II. METODE PENELITIAN

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode inkremental. Model inkremental adalah model pengembangan sistem pada *software development* berdasarkan kebutuhan yang diperlukan yang terbagi menjadi beberapa bagian sehingga pada proses pembangunannya dilakukan secara bertahap. Model inkremental ini dapat membantu untuk memeperskecil kebutuhan biaya dan memeperscepat proses dari pembuatan suatu fungsi perangkat lunak yang akan dibangun. Model inkremental merepresentasikan suatu proses yang fokus utamanya adalah pada kebutuhan sistem dan pengimplementasiannya dalam kerjasama dalam tim [4].

Pada penelitian ini, alur pikir penelitian dibagi menjadi beberapa tahap. Tahap pertama yang dilakukan adalah identifikasi masalah dan studi literatur. Tahap kedua adalah persiapan *dataset training* dan *testing*. Tahap ketiga adalah pembuatan model yang ada pada inkremental 1. Tahap keempat adalah pembuatan GUI yang ada pada inkremental 2. Tahap akhir dilakukan evaluasi terhadap sistem yang telah dibuat. Alur pikir penelitian ini ada pada Gambar 1.



Gambar 1 Alur pikir penelitian

A. Identifikasi Masalah dan Studi Literatur

Identifikasi masalah dilakukan dengan cara mencari fenomena menarik yang dapat diangkat sebagai topik. Dalam penelitian ini, dipilih topik mengenai pengendara sepeda motor yang melintasi trotoar ketika lalu lintas macet. Data menunjukkan bahwa jumlah kendaraan bermotor di Indonesia kian meningkat tiap tahunnya [5]. Peningkatan jumlah kendaraan bermotor ini memicu naiknya tingkat kemacetan di Indonesia [2], yang dapat meningkatkan potensi sepeda motor untuk melintasi trotoar.

Penelitian yang dilakukan oleh J.Redmon dan A.Farhadi [6] yang berjudul “YOLOv3 : An Incremental Improvement”, menjadi acuan utama mengenai algoritma YOLO v3. Dalam penelitian tersebut, dijelaskan secara garis besar tentang perkembangan algoritma YOLO (*You Only Look Once*) v3 dibanding versi sebelumnya. YOLOv3 memiliki mAP (*mean average precision*) 28.2 dengan waktu pemrosesan 22 mili detik. Angka tersebut jika dibandingkan dengan algoritma SSD (*single shot multibox detector*) memiliki akurasi yang setara namun dengan waktu pemrosesan yang 3 kali lebih cepat.

Penelitian serupa lainnya yang dilakukan Espinosa, dkk [7], berfokus pada pendeteksian dan pengklasifikasian sepeda motor berbasis model dalam *faster R-CNN*. Dalam penelitian ini, model pendeteksian objek dibuat oleh peneliti. Penelitian ini menghasilkan akurasi sebesar 75% pada skenario lalu lintas perkotaan yang padat. Pada skenario lalu lintas yang tidak padat, model mampu menghasilkan akurasi sebesar 92%.

YOLO v3 merupakan suatu algoritma yang sangat populer pada topik pendeteksian objek *real time*. Algoritma ini menerapkan *neural networks* pada gambar secara keseluruhan [8]. Dasar dari algoritma ini adalah menggunakan CNN (*Convolutional Neural Network*). YOLO v3 bukan algoritma yang paling baik nilai akurasinya, namun dalam skenario *real-time*, algoritma ini optimal karena memiliki waktu pemrosesan yang cepat. Gambar yang telah dilakukan *preprocessing*, diproses untuk

pengekstraksian fitur menggunakan arsitektur Darknet-53 dan menghasilkan 3 skala fitur. Dari fitur tersebut, kemudian dapat dilakukan pendeteksian objek dan dilakukan pemberian kotak atau *bounding box* pada objek yang terdeteksi.

Convolutional Neural Network (CNN) merupakan salah satu algoritma dari *Deep Learning*. CNN juga merupakan pengembangan dari *Multi Layer Perceptron* (MLP). CNN dirancang untuk mengolah data dalam bentuk *grid*, contohnya seperti citra dua dimensi atau suara. CNN dapat digunakan untuk mengklasifikasikan data berlabel menggunakan metode *supervised learning*. Cara kerja dari *supervised learning* adalah harus terdapat data yang dilatih dan terdapat variabel yang ditargetkan. Tujuan dari *supervised learning* adalah mengelompokkan suatu data baru ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atau melakukan deteksi dan segmentasi objek [9].

Pada penelitian ini, studi literatur dilakukan dengan cara membaca jurnal, buku, dan penelusuran di internet untuk mengetahui algoritma apa saja yang dapat mendeteksi objek bergerak. Pada tahap ini, dilakukan analisis mengenai algoritma yang dapat mendeteksi objek bergerak. Ditemukan beberapa algoritma yang cocok dalam penelitian ini seperti *Faster R-CNN*, *R-FCN*, *YOLO*, dan *SSD* [3]. Rincian mengenai performa setiap algoritma dijelaskan pada Tabel 1.

Tabel 1 Perbandingan akurasi [3]

Algoritma	Akurasi	FPS
Faster R-CNN	34,9%	5
R-FCN	31,5%	6
SSD	26,8%	22
YOLOv3	33%	40

Dalam nilai akurasi, *Faster R-CNN* memang memiliki angka yang dapat mengimbangi *YOLO*. Namun pada variabel FPS diketahui bahwa *Faster R-CNN* memiliki *frame per second* yang sangat rendah dibanding *YOLO*. Untuk mendeteksi objek bergerak terutama pada skenario lalu lintas yang cenderung cepat, diperlukan algoritma dengan *frame per second*

yang baik dan akurasi yang mumpuni. Berdasarkan pertimbangan tersebut, YOLO dipilih sebagai algoritma pendeteksian objek dalam penelitian ini karena memiliki perbandingan akurasi dan FPS yang seimbang. YOLOv3 memiliki akurasi yang cukup baik dan juga FPS yang tinggi.

B. Persiapan Dataset

Tahap awal penelitian dilakukan untuk persiapan pengembangan perangkat lunak. Pada tahap pengumpulan data, data yang diperlukan terbagi menjadi dua yaitu *dataset training* dan *dataset testing*. *Dataset training* berupa gambar sepeda motor dari berbagai sisi. Pembuatan *dataset* diperoleh dari *Google images* dan pengambilan gambar dari hasil perekaman layar video CCTV lalu lintas. Kata kunci yang digunakan adalah “sepeda motor” dan “sepeda motor lalu lintas”. Setelah gambar diunduh, dilakukan pemilihan gambar yang memenuhi kriteria yaitu terdapat objek sepeda motor dalam gambar tersebut secara manual. Dalam *dataset* ini berisikan 400 gambar dari berbagai sisi. *Dataset testing* berupa video pada lokasi trotoar pejalan kaki. Pengambilan *dataset testing* ini dilakukan dengan merekam layar video CCTV lalu lintas yang diakses secara *online* pada situs smartcity.jakarta.go.id. Gambar 2 merupakan contoh sebagian dari *dataset* yang digunakan.



Gambar 2 Dataset sepeda motor

C. Inkremental 1

Pada tahap inkremental satu, penelitian berfokus pada pembelajaran dan pengujian terhadap sistem yang dibuat. Di tahap ini, *dataset* yang ada diproses hingga menjadi sebuah model yang dapat mendeteksi pengendara sepeda motor.

1. Analisis

Pada *dataset training*, gambar perlu dilakukan *preprocessing* terlebih dulu sebelum lanjut pada tahap selanjutnya. *Preprocessing* dilakukan dengan cara melakukan *labelling* setiap gambar pada *dataset*. Proses *labelling* yang dilakukan adalah memberikan anotasi koordinat dari letak objek dalam setiap gambar menggunakan aplikasi *LabelImg*. Hasil dari pelabelan gambar adalah koordinat letak objek sepeda motor dalam gambar yang disimpan dalam *file *.txt*. Hal tersebut diperlukan agar program yang dibuat dapat mengetahui lokasi sebenarnya sebuah objek dalam gambar.

2. Desain

Tahapan ini berfokus pada perancangan arsitektur dari algoritma yang digunakan. Dalam pembuatan model dalam penelitian ini, digunakan algoritma YOLO v3 dengan *weight* 416. Pengekstraksian fitur menggunakan *framework* Darknet-53 yang memiliki 53 *layer* konvolusi. Detail dari arsitektur tersebut ada pada Gambar 3.

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

Gambar 3 Arsitektur Darknet-53 [6]

3. Implementasi

Pada tahap ini, proses pembelajaran model mulai dilakukan. Seluruh gambar beserta anotasi yang telah dibuat dijadikan *input* dan diolah menggunakan algoritma YOLO v3. Dilakukan perubahan parameter *class* dan *filter* yang disesuaikan dengan kebutuhan penelitian. Parameter *class* diubah menjadi 1 karena objek yang dideteksi hanya sepeda motor. Parameter *filter* diubah menjadi 18 dari hasil perhitungan $3 \times (4 + 1 + \text{jumlah class})$. Angka 3 mewakili jumlah *anchor* pada setiap *grid*, angka 4 mewakili 4 titik dari lokasi objek, angka 1 mewakili nilai *objectness*. Keluaran dari proses ini adalah sebuah model yang dapat mendeteksi sepeda motor. Model yang dihasilkan pada tahap ini akan diuji pada tahap selanjutnya.

4. Pengujian

Dalam tahap pengujian, dilakukan dengan menggunakan input video lalu lintas di trotoar pejalan kaki. Sistem harus mampu untuk mengenali sepeda motor dengan baik lalu dilakukan *tracking* jika sepeda motor tersebut terdeteksi. Bagi sepeda motor yang terdeteksi dan melintasi garis larangan, sistem akan mengeluarkan pemberitahuan bahwa ada sepeda motor yang melintasi trotoar.

5. Hasil

Hasil dari inkremental satu adalah sebuah model yang dapat mendeteksi sepeda motor. Selain itu juga program yang dibangun harus mampu untuk memberikan pemberitahuan jika ada pengendara motor yang melintasi trotoar.

D. Inkremental 2

Pada tahap inkremental dua, penelitian berfokus pada pembuatan *graphical user interface* (GUI). Seluruh kegiatan yang sudah dilakukan pada inkremental satu diimplementasikan penuh pada tahap ini. Dalam pembuatan GUI ini menggunakan *plug-in* PyQt5 sebagai perangkat lunak bantuan. GUI yang dibuat dapat menampilkan *output* berupa video yang dipilih yang sudah dapat melakukan deteksi sepeda motor. Jika sepeda motor yang

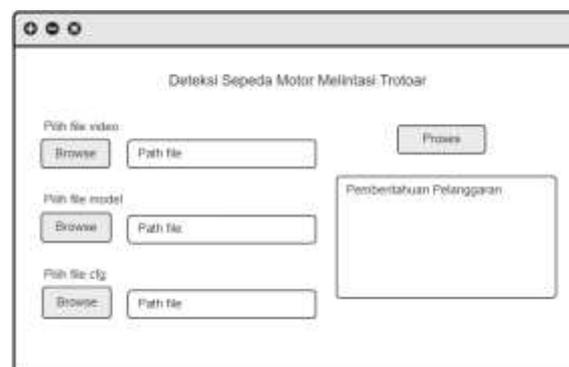
berada di trotoar dan melintasi garis larangan, sistem juga dapat menampilkan pemberitahuan.

1. Analisis

Dalam tahap ini, dilakukan analisis tentang bagaimana cara mengimplementasikan model yang sudah dibuat untuk mendeteksi sepeda motor. Pendeteksian tersebut menggunakan *input testing* video lalu lintas yang terintegrasi secara keseluruhan ke dalam sebuah aplikasi.

2. Desain

Pada tahap ini, dilakukan pembuatan tampilan aplikasi. Dalam aplikasi tersebut, perlu menyajikan judul aplikasi, tampilan *output* video lalu lintas yang sudah dideteksi, dan ruang untuk menampilkan pemberitahuan jika ada sepeda motor yang melintasi trotoar. Gambar 4 merupakan tampilan aplikasi yang akan dibuat.



Gambar 4 Mockup tampilan aplikasi

3. Implementasi

Pada tahap implementasi, mulai dilakukan pembuatan. Seluruh kegiatan pada inkremental satu diintegrasikan secara penuh pada aplikasi. Kegiatan pengkodean dibuat menggunakan bantuan *plugin* PyQt5. Model yang didapat pada inkremental satu, digunakan pada video lalu lintas untuk mendeteksi pengendara motor.

4. Pengujian

Pengujian aplikasi dilakukan dengan metode *blackbox testing*. Beberapa hal yang diperlukan pengujian antara lain:

- Tampilan *input* video

- Pendeteksian sepeda motor
- Teks pemberitahuan sepeda motor melintasi garis larangan.

5. Hasil

Dalam tahap ini, disajikan hasil dari *blackbox testing* pada tahapan sebelumnya. Tabel 2 merupakan skenario pengujian dengan *blackbox testing*.

Tabel 2 Skenario *blackbox testing*

No	Skenario Pengujian	Ekspektasi Hasil	Berhasil	Tidak Berhasil
1	Menampilkan file browser untuk memilih file input video	Aplikasi dapat menampilkan file browser, dapat memilih file, dan lokasi file tampil pada box yang ada		
2	Menampilkan file browser untuk memilih file model	Aplikasi dapat menampilkan file browser, dapat memilih file, dan lokasi file tampil pada box yang ada		
3	Menampilkan file browser untuk memilih file cfg	Aplikasi dapat menampilkan file browser, dapat memilih file, dan lokasi file tampil pada box yang ada		
4	Memproses video dan menampilkan pemberitahuan saat ada sepeda motor melintasi trotoar	Aplikasi dapat menampilkan window video dan menampilkan pemberitahuan jika ada sepeda motor melintasi trotoar		

E. Evaluasi

Pada tahap ini, dilakukan evaluasi terhadap aplikasi yang sudah dibuat. Evaluasi tersebut meliputi nilai AP (*average precision*) model dan *average loss*. Untuk proses pengujian aplikasi, dilakukan uji *blackbox testing* kepada beberapa

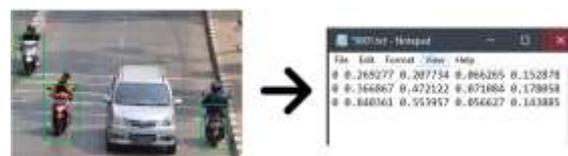
responden untuk mengetahui apakah aplikasi sudah dapat berjalan dengan baik atau tidak.

III. HASIL DAN PEMBAHASAN

Pada pengembangan aplikasi deteksi sepeda motor yang melintasi trotoar ini, dibuat dengan bahasa pemrograman python versi 3.7. Hasil dari proses pendeteksian ini adalah pemberitahuan jika ada motor yang melintasi trotoar beserta gambar saat peristiwa terjadi. Untuk dapat melakukan pendeteksian sepeda motor, perlu dilakukan beberapa langkah seperti persiapan *dataset*, pembuatan model pada inkremental 1, pembuatan GUI pada inkremental 2, dan tahap evaluasi.

A. Persiapan Dataset

Pada tahap ini, semua gambar dari *dataset* yang dimiliki perlu dilakukan *preprocessing* yaitu pelabelan menggunakan aplikasi LabelImg. Hasil dari tahap ini adalah *file* dengan format *.txt yang berisikan koordinat lokasi spesifik dari objek sepeda motor pada tiap gambar. Dalam pelabelan ini, jika ada gambar yang terdapat lebih dari satu objek sepeda motor, maka koordinat letak objek dapat dilakukan sebanyak objek yang ada. Pada Gambar 5 adalah hasil pelabelan gambar.

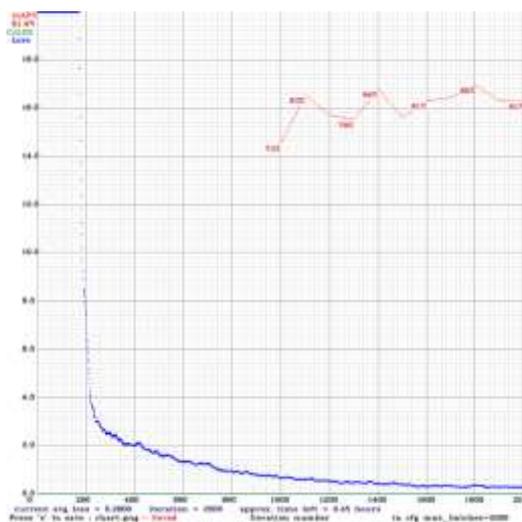


Gambar 5 Hasil pelabelan gambar

B. Hasil Inkremental 1

Pada tahap ini, dilakukan proses *training* dari *dataset* yang dimiliki untuk menjadi sebuah model. Proses *training* ini dilakukan menggunakan Google Colab dengan GPU. Hal ini dilakukan untuk mempercepat proses *training*. Hal yang perlu dilakukan sebelum *training* adalah melakukan konfigurasi agar program dieksekusi menggunakan GPU. Lalu dilakukan *cloning* Darknet beserta *file weights*. Darknet merupakan algoritma untuk melatih

neural network yang ditulis dalam bahasa C dan CUDA dan berfungsi sebagai basis untuk YOLO. Selanjutnya dilakukan *importing dataset* sepeda motor yang sebelumnya sudah dibuat. Setelah semua data telah siap, dapat dilakukan proses pembelajaran atau pembuatan model untuk mendeteksi sepeda motor. Dengan mengubah parameter *class* dan *filter* menjadi 1 dan 18, dihasilkan model dengan AP 81,38% dan *average loss* 0,28% seperti pada Gambar 6.



Gambar 6 Visualisasi AP dan *average loss* model

C. Hasil Inkremental 2

Pada tahap ini dipilih PyQt5 sebagai alat bantu dalam pembuatan GUI pada penelitian ini. Aplikasi yang dibuat dalam penelitian ini memiliki menu untuk memilih *file* video *input* dari media penyimpanan lokal maupun *live streaming*, memilih *file* model, memilih *file* *cfg*, dan menu bantuan. Menu tersebut disajikan dalam bentuk tombol untuk memilih *file* dan *list text* sebagai penampil lokasi *file* pada penyimpanan. Kemudian untuk memulai pendeteksian, disajikan dalam bentuk tombol bertuliskan proses yang *output* pemberituannya dalam bentuk *list text*. Hasil dari tahap ini merupakan tampilan aplikasi seperti pada Gambar 7.



Gambar 7 Tampilan aplikasi

Untuk pengujian aplikasi menggunakan *input* video CCTV ada pada Gambar 8. Jika pemrosesan video dilakukan, akan tampil video beserta pendeteksiannya. Untuk *file input* dapat menggunakan *file* lokal, kamera, dan juga *live streaming*. Dalam prosesnya, aplikasi akan mendeteksi adanya motor yang melintasi trotoar jika ada objek terdeteksi menyentuh garis larangan yang sebelumnya sudah dibuat. Garis larangan memiliki posisi yang berbeda-beda, disesuaikan dengan kebutuhan dimana letak trotoar tersebut berada.



Gambar 8 Pengujian aplikasi

D. Evaluasi

Setelah aplikasi dapat dijalankan, diperoleh hasil seperti pada Gambar 8. Jika terdeteksi ada sepeda motor yang melintasi trotoar, pada kotak teks pemberitahuan akan mengeluarkan tulisan berupa waktu terjadinya peristiwa dan jumlah peristiwa yang terjadi. Selain itu gambar saat ada sepeda motor melintasi trotoar juga tersimpan pada folder *output*. Untuk mengetahui keberhasilan fungsi aplikasi, dilakukan *blackbox testing* kepada beberapa responden dengan hasil seperti pada Tabel 3.

Tabel 3 Skenario *blackbox testing*

Responden	Jumlah Pengujian Berhasil / Total Pengujian
Responden ke-1	4/4
Responden ke-2	4/4
Responden ke-3	4/4
Responden ke-4	4/4

IV. SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat diambil simpulan sebagai berikut:

1. Pada penelitian ini berhasil dilakukan pendeteksian sepeda motor menggunakan algoritma YOLO v3. Dengan menggunakan *plug-in* PyQt5, dihasilkan program dengan GUI yang dapat memberikan *output* pemberitahuan saat ada sepeda motor yang melintasi trotoar.
2. Dalam penelitian ini dilakukan perubahan parameter *class* dan *filter*. Parameter *class* diubah menjadi 1 dan *filter* diubah menjadi 18.
3. Algoritma YOLOv3 cocok digunakan untuk pendeteksian objek terutama pada skenario yang memerlukan *frames per second* yang tinggi.
4. Kesesuaian *dataset* dengan data pengujian berperan sangat penting dalam proses pendeteksian objek. Dalam penelitian ini, digunakan *dataset* motor yang beredar di Indonesia dan data pada sudut pengambilan gambar dari CCTV untuk menyesuaikan dengan objek yang akan dideteksi.
5. Dalam memproses video *realtime* menggunakan algoritma YOLOv3, diperlukan perangkat keras komputer yang mumpuni untuk mendukung kualitas *output* yang dihasilkan.
6. Pada proses pendeteksian, semakin banyak *layer* konvolusi yang digunakan, semakin akurat hasil yang diperoleh. Namun komputasi yang dijalankan juga semakin berat.

Dalam penelitian ini, masih terdapat kekurangan yang ditemui, saran yang dapat dilakukan untuk penelitian selanjutnya sebagai berikut:

1. Memperbanyak *dataset* sepeda motor yang berada di Indonesia dalam berbagai kondisi dan sudut pengambilan gambar untuk meningkatkan akurasi model yang dihasilkan.
2. Dilakukan penambahan pengenalan plat nomor kendaraan agar aplikasi dapat langsung mendapatkan data pelaku pelanggaran lalu lintas.
3. Dilakukan pendeteksian seluruh area trotoar agar sepeda motor yang berada dalam area trotoar dapat terdeteksi sebagai area pelanggaran.

DAFTAR RUJUKAN

- [1] Marroli, "Rata-rata Tiga Orang Meninggal Setiap Jam Akibat Kecelakaan Jalan," 2017. [Online]. Available: https://kominform.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr. [Accessed: 11-Mar-2020].
- [2] "Jakarta traffic report," 2019. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/jakarta-traffic. [Accessed: 31-Oct-2019]. F. M. White. *Fluid Mechanics*, 3rd edition. New York: McGraw-Hill, Inc., hlm 30-32, 1994.
- [3] J. Hui, "Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)," 2018. [Online]. Available: https://medium.com/@jonathan_hui/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359. [Accessed: 07-Apr-2020].
- [4] R. T. TEDJA, "Software Development Model: Incremental Model," 2019. [Online]. Available: <https://sis.binus.ac.id/2019/07/02/software-development-model-incremental-model/>. [Accessed: 24-Mar-2020].
- [5] H. Widowati, "Berapa Jumlah Kendaraan di DKI Jakarta?," 2019. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2019/08/02/berapa-jumlah-kendaraan-di-dki-jakarta>. [Accessed: 31-Oct-2019].
- [6] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." 2018. [Online]. Available: <https://pjreddie.com/yolo>. [Accessed: Apr. 8, 2020].
- [7] J. E. Espinosa, S. A. Velastin, and J. W. Branch, "Motorcycle detection and classification in urban

- scenarios using a model based on faster R-CNN,” IET Conf. Publ., vol. 2018, no. CP745, pp. 91–96, 2018.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” 2016. [Online]. Available: <http://pjreddie.com/yolo/>. [Accessed: 23- Jun- 2020].
- [9] T. F. Kusumaningrum, “Implementasi Convolution Neural Network (CNN) untuk Klasifikasi Jamur Konsumsi di Indonesia Menggunakan Keras,” *Univ. Islam Indones.*, 2018.