

# Pendeteksian Pelanggaran Jalur Bus Transjakarta Menggunakan Algoritma YOLOv3

Ricardo Hubert Lingga<sup>1)</sup>, Yulia Ery Kurniawati<sup>2)</sup>

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis  
Jalan Pulomas Selatan Kav. 22, Jakarta 13210

<sup>1)</sup> Email: ricardohub681@gmail.com

<sup>2)</sup> Email: yulia.kurniawati@kalbis.ac.id

**Abstract :** Police of Metro Jaya fined about 42,286 Motorcycles that crossed lane of Transjakarta in a period of eleven months in 2014. To reduce the number of bus lane violation, it's very important to take some continues action from the authorities. The purpose of this study is to detect vehicles that break through the bus lanes with the application of computer vision until get the notification of bus lane violation. In this study used training data as much 470 pictures data with three classes of vehicles like motorcycles, bus, and cars. This study used You Only Look Once v3 algorithm with training data that was trained as many as 6000 Iterations which resulted in an map of 99.96% and was succesfully that used to detect vehicles with help of sort method in kalman filter to add counting function to detect vehicles that cross in bus lane with the dividing lane, and creating some new file in .mp4 format for insertion into the GUI.

**Keywords:** bus lane violation, data training, yolo, vehicle detection

**Abstrak:** Kepolisian Metro Jaya menilang sekitar 42.286 kendaraan bermotor yang menerobos jalur Transjakarta dalam kurun waktu sebelas bulan pada tahun 2014. Untuk mengurangi jumlah pelanggaran jalur bus maka perlu adanya penindakan lanjutan dari pihak yang berwajib. Tujuan adanya penelitian ini adalah mendeteksi kendaraan yang menerobos jalur bus dengan penerapan computer vision, hingga mengeluarkan pemberitahuan pelanggaran jalur bus. Pada penelitian ini menggunakan metode pengembangan inkremental. Dalam penelitian ini menggunakan You Only Look Once v3 (YOLO) dengan data yang di training berjumlah 470 data gambar dengan tiga kelas kendaraan yaitu mobil, motor, bus. Penelitian ini menggunakan data training yang dilatih sebanyak 6000 Iterasi yang menghasilkan mAP sebesar 99.96% dan berhasil digunakan untuk mendeteksi kendaraan dan dengan bantuan metode sort yang dilengkapi dengan kalman filter hingga menambahkan fungsi counting pada deteksi kendaraan yang melintasi jalur bus dengan adanya garis pembatas, dan membuat file baru dalam format .mp4 untuk dimasukkan ke dalam GUI.

**Kata kunci:** latih data, pelanggaran jalur bus, pendeteksian kendaraan, yolo

## I. PENDAHULUAN

Kepadatan, kecelakaan, maupun pelanggaran lalu lintas di jalan raya menjadi salah satu permasalahan yang pemerintah hadapi dari dulu hingga saat ini. Padatnya lalu lintas di jalan raya disebabkan oleh banyaknya jumlah kendaraan yang berlebihan, dan bisa juga disebabkan oleh kurangnya kedisiplinan pengendara seperti menerobos jalur bus

yang dapat menyebabkan kecelakaan. Selain sangat berbahaya, menerobos jalur bus biasanya dapat menyebabkan kepadatan [1].

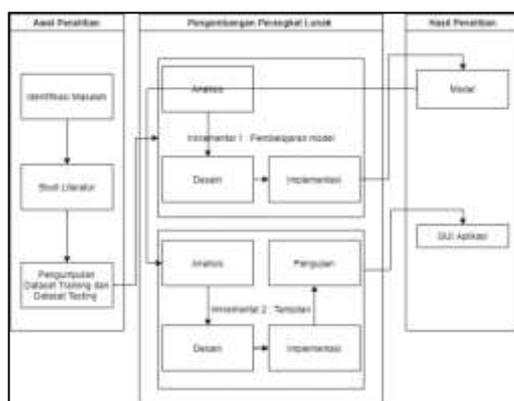
Direktorat lalu lintas Polda Metro Jaya mencatat sebanyak 3.194 pengendara sepeda motor yang terekam kamera *Electronic Traffic Law Enforcement (ETLE)* / tilang elektronik. Dan pelanggaran terbanyak yang dilakukan adalah

menerobos jalur Transjakarta dengan total kasus 1.586 pengendara roda dua yang melanggar [2].

Penilangan *online* merupakan jalan yang tepat dalam upaya menertibkan lalu lintas seperti penerobos jalur bus, akan tetapi pihak kepolisian harus terus menerus memantau layar CCTV yang menyebabkan banyaknya pembiaran pelanggaran lalu lintas yang dapat berpotensi meningkatkan adanya kecelakaan lalu lintas, sehingga hal tersebut dianggap kurang efisien dalam penanganan pelanggaran lalu lintas khususnya penyerobotan jalur bus. Penelitian ini berfokus untuk mendeteksi kendaraan pelanggar jalur bus dengan penerapan *computer vision* dengan menggunakan algoritma YOLO, hingga mengeluarkan pemberitahuan terjadinya pelanggaran jalur bus.

## II. METODE PENELITIAN

Di bawah ini merupakan alur dari proses penelitian yang menjadi gambaran dalam pembuatan aplikasi pendeteksi pelanggaran jalur bus transjakarta dalam bentuk *video* yang terbagi atas tiga kelas objek yaitu motor, mobil, dan bus menggunakan YOLOv3.



Gambar 1 Proses penelitian

Pada Gambar 1 merupakan alur proses penelitian yang terbagi atas tiga tahap yaitu awal penelitian,

pengembangan perangkat lunak, dan hasil penelitian.

### A. Awal Penelitian

Pada tahap awal penelitian, terdapat langkah-langkah berupa identifikasi masalah, studi literatur, dan pengumpulan data. Awal penelitian dilakukan untuk melakukan persiapan dalam melakukan pengembangan perangkat lunak.

- **Identifikasi Masalah**  
Identifikasi masalah dilakukan untuk mencari suatu kejadian yang dapat dijadikan topik penelitian, Pada penelitian ini dipilih topik mengenai kendaraan yang melintasi jalur bus. Data menunjukkan tingkat pelanggaran perlintasan jalur bus yang tinggi [2].
- **Studi Literatur**  
Studi literatur ini dilakukan berdasarkan penelitian yang telah dilakukan oleh peneliti-peneliti sebelumnya. Studi literatur ini dapat berfungsi untuk memvalidasi metode yang akan digunakan dalam melakukan pengembangan aplikasi pelanggaran jalur bus transjakarta dengan menggunakan algoritma YOLO. Selain itu terdapat juga metode pelacakan *kalman filter* yang ikut berperan dalam penelitian ini. *Kalman filter* merupakan algoritma yang memberikan perkiraan beberapa variabel yang tidak diketahui berdasarkan pengukuran yang diamati dari waktu ke waktu [3].
- **Pengumpulan Data (Observasi)**  
Pada penelitian ini menggunakan metode observasi dalam mengumpulkan informasi pada data yang dibutuhkan dalam proses penelitian.

Data yang dibutuhkan dalam penelitian ini terbagi atas himpunan data latih, dan himpunan data uji. Himpunan data uji berupa gambar dari motor, mobil, dan bus. Pembuatan himpunan data

diambil dari hasil foto kamera *handphone*.

## B. Pengembangan Perangkat Lunak

Pada penelitian ini menggunakan pengembangan perangkat lunak dengan metode inkremental. Model inkremental berfokus terhadap pembawaan produk operasional terhadap setiap inkremen [4]. Pengembangan inkremental digunakan untuk menyesuaikan waktu ke waktu yang sesuai dengan kondisi, dan juga dapat dijadikan sebagai suatu prinsip dalam *Product Software Knowledge Infrastructure (PSKI)* [5]. Tahap konsep pengembangan inkremental terdiri dari *Requirment* yang menganalisis/menentukan kebutuhan, *Specification* yang merupakan proses yang mengacu pada analisis kebutuhan, *Architecture design* sebagai perancangan software, *code* merupakan proses implementasi, dan *test* sebagai tahap pengujian [6].

- Inkremental Satu

Tahap Inkremental satu dilakukan pembuatan logika program beserta model algoritma yang akan digunakan hingga melatih model algoritma tersebut. Model algoritma yang digunakan pada penelitian ini adalah YOLOv3. YOLO merupakan pendekatan sistem pendeteksian objek yang ditargetkan untuk pemrosesan secara real time. YOLO meringkaskan pendeteksian objek sebagai regresi tunggal, piksel gambar diarahkan langsung ke dalam bounding box spesial yang terpisah berdasarkan data yang telah disiapkan [7]. Proses dari inkremental satu terdiri dari proses analisis kebutuhan fungsionalitas dan non-fungsionalitas, desain logika model, implementasi berupa code, dan pengujian terhadap model aplikasi yang telah dikerjakan.

- Inkremental Dua

Pada tahap inkremental dua, penelitian akan fokus terhadap pembuatan GUI. Dalam pembuatan GUI ini memakai platform yang disebut dengan PyQt5. PyQt5 merupakan suatu framework yang dapat digunakan sebagai pengenalan suatu objek kendaraan, dan juga dapat digunakan sebagai pembuatan suatu user interface.

## III. HASIL DAN PEMBAHASAN

### A. Inkremental Satu

- Analisis

Dalam tahap ini dibutuhkan analisis kebutuhan fungsional dan non-fungsional yang dapat menunjang selama penelitian berlangsung. Kebutuhan fungsional tersebut antara lain :

- *Handphone* yang digunakan dapat dijadikan himpunan data latih
- *Pc* yang digunakan dapat mengoperasikan pembuatan himpunan data, pembelajaran model, hingga pendeteksian kendaraan
- Sistem dapat mendeteksi objek kendaraan yang melintas
- Sistem dapat melacak objek yang terdeteksi
- Sistem dapat menghitung dan menampilkan jumlah objek kendaraan yang melakukan pelanggaran

Adapun kebutuhan non-fungsionalitas seperti yang ditunjukkan pada Tabel 1 dan Tabel 2 berikut

Tabel 1 Daftar perangkat keras

Perangkat Keras	Nama
Laptop	Asus Vivobook S14
Handphone	Xiaomi Mi Note 10

Tabel 2 Daftar perangkat lunak

Perangkat Lunak	Versi
Spyder	4.2.1
Miniconda	4.10.3
Google Colab	n/a

- **Desain**

Pada penelitian ini menggunakan yolov3 yang menjadi *load weight* dengan menggunakan *darknet-53* sebagai susunan layer yang dikonfigurasi seperti pada Tabel 3. Lalu dibuat skenario latihan data dengan tujuan untuk mencari model dengan iterasi terbaik sehingga dapat dipakai dalam penelitian ini.

Tabel 3 Konfigurasi Yolov3

Jenis Konfigurasi	Keterangan
batch	64
subdivisions	64
Max_batches	8000
width	416
height	416
classes	3
filters	24

Jumlah *batch* menentukan jumlah gambar yang akan di proses sebelum *network weight* melalui pembaharuan. Lalu jumlah *subdivision* memiliki tujuan untuk memproses sisipan *batch size* sekaligus pada GPU. *Max\_batches* adalah batas iterasi dalam melakukan pelatihan data. *Height* dan *Weight* merupakan dimensi gambar data masukkan yang akan dilatih. *Classes* adalah jumlah kelas yang ingin dideteksi. Pelatihan data pada pendeteksian ini menggunakan hasil rekaman *video* yang berlokasi pada halte bus walikota jakarta timur, dan halte bus utan kayu rawamangun dengan himpunan data yang berjumlah 470 data gambar, dan terbagi atas tiga kelas yang berbeda yaitu motor, mobil, dan bus.

- **Implementasi**

Pada tahap ini, total data gambar yang telah direkam oleh *handphone* dalam format .jpg akan membentuk suatu himpunan data yang perlu untuk diberi anotasi sehingga dapat terjadinya proses latihan data dengan yolo. Proses pemberian anotasi ini memiliki tujuan untuk mengenali objek yang akan dideteksi. Setelah semua gambar tersebut sudah dikenali. Selanjutnya dilakukan proses unduh *darknet-53*, cara kerja darknet ini yaitu dengan melakukan proses latihan kembali dengan menggunakan model yang telah ada dan sudah disediakan oleh *darknet*. Selanjutnya masukkan berkas konfigurasi model yolo dalam format .cfg dengan konfigurasi yang sudah ditentukan pada Tabel 3. Setelah itu barulah proses latihan data dapat dieksekusi. Setelah data berhasil dilatih maka akan terbentuk suatu model yang akan disimpan untuk dilakukan proses deteksi sehingga objek kendaraan yang melintasi jalur bus dapat terdeteksi dengan garis yang menjadi parameter deteksi kendaraan.

- **Pengujian**

Pada tahap ini sistem dapat mengenali objek kendaraan pada jalur bus dengan baik. Namun terdapat kondisi yang menyebabkan terjadinya kesalahan saat deteksi seperti yang ditunjukkan pada Gambar 2



Gambar 2 kesalahan pengenalan objek

Berdasarkan perkiraan dari hasil uji coba yang telah dilakukan. Hal ini dapat disebabkan karena terdapat dua objek dari kelas yang berbeda pada *frame* sebelumnya yang saling berdekatan hal ini juga dapat diperkuat berdasarkan pada teori *Kalman Filter* yang memberikan perkiraan beberapa variabel dari waktu ke waktu [3]. Dan dapat dipicu oleh jumlah data yang masih minim yaitu 470 total data gambar dengan tiga kelas yang berbeda yang diambil dari dua lokasi tempat yang dijadikan sebagai data latih. Pada Gambar 3 merupakan hasil uji coba model dengan iterasi terbaik

Bobot file.cfg	mAp	Ap mobil	Ap motor	Ap bus	IoU
1000 Iterasi	85,61%	87,86%	95,93%	73,05%	61,88
2000 Iterasi	98,46%	99,90%	99,01%	96,45%	75,99
3000 Iterasi	99,89%	99,96%	99,79%	99,92%	79,95
4000 Iterasi	99,98%	100%	99,96%	99,98%	82,16
5000 Iterasi	99,87%	99,98%	99,63%	100%	82,57
6000 Iterasi	99,96%	99,91%	99,98%	99,99%	82,83
7000 Iterasi	99,97%	99,98%	99,94%	100%	84,29
8000 Iterasi	100%	100%	100%	100%	86,04

Gambar 3 Skor hasil uji coba

Berdasarkan Gambar 3 yang telah diujikan data latih yang paling baik untuk digunakan adalah dengan menggunakan Iterasi 6000, karena *average precision* yang dihasilkan pada setiap kelas kendaraan tidak mencapai 100% yang membuat data latih tersebut tidak menjadi *overfit* sehingga variasi *input* data uji dapat terbaca dengan baik.

## B. Inkremental Dua

### • Analisis

Dalam tahap ini dibutuhkan analisis kebutuhan fungsional dan non-fungsional pembuatan GUI aplikasi yang berfungsi sebagai media pemutaran berkas video deteksi. Kebutuhan fungsional tersebut antara lain :

- Sistem dapat menjalankan aplikasi dengan baik
- Sistem dapat mengeksekusi perintah *input video*
- Sistem dapat mengeksekusi perintah arah pada tombol *slider*

- Sistem dapat mengeksekusi perintah *pause & play video* yang dijalankan

Adapun kebutuhan non-fungsional yang dibutuhkan dari keperluan GUI ini. Pada penelitian ini dibutuhkan laptop dengan spesifikasi Seperti yang ditunjukkan pada Tabel 1. Pada Tabel 4 merupakan daftar perangkat lunak dalam pengembangan GUI aplikasi.

Tabel 4 Daftar perangkat lunak\_2

Perangkat Lunak	Versi
Balsamic	3
Spyder	4.2.1
Miniconda	4.10.3

### • Desain

Tahap ini merupakan pembuatan tampilan dari aplikasi ke dalam bentuk *mock up* yang ditunjukkan seperti pada Gambar 4



Gambar 4 Mockup GUI

### • Implementasi

Proses ini merupakan realisasi hasil analisis dan desain *mock up* yang sudah dilakukan dengan menggunakan *PyQt5* dengan kode pemrograman Python. Gambar 5 merupakan tampilan aplikasi saat menjalankan berkas video hasil deteksi.



Gambar 5 Tampilan GUI

- **Pengujian**  
Proses ini dilakukan untuk mengetahui fungsi pada tampilan GUI yang telah diimplementasi dapat berjalan dengan baik sesuai dengan kebutuhan fungsionalitas dengan menggunakan skenario pengujian *black box*. Skenario pengujian *black box* bertujuan untuk memastikan setiap fungsi pada tombol dapat berjalan dengan baik. Pada Tabel 5 merupakan hasil dari skenario pengujian *black box*

Tabel 5 Pengujian black box

Nama Tombol	Fungsi	Ekspektasi	Hasil
Button open video	Membuka file dalam format .mp4, .avi	Aplikasi dapat memproses file yang akan dipilih untuk dideteksi	Berhasil
Button play video	Memutar file video	Aplikasi dapat menjalankan file video dengan baik	Berhasil
Button slider	State video dapat dimaju mundur sesuai keinginan user	Video yang dijalankan dalam aplikasi dapat bergerak maju mundur sesuai dengan keinginan user	Berhasil

#### IV. SIMPULAN

Berikut merupakan kesimpulan yang didapatkan selama penelitian berlangsung antara lain:

- Pada penelitian ini sistem dapat melakukan deteksi kendaraan berupa motor, mobil, dan bus dengan menggunakan algoritma YOLOv3
- Metode pelacakan *Kalman Filter* baik digunakan untuk melacak kendaraan dengan catatan nilai IoU yang tinggi dan jumlah himpunan data yang banyak. Sehingga tidak terjadi kesalahan dalam pengenalan objek.
- Jumlah iterasi tidak sembarangan di tingkatkan, namun tetap disesuaikan dengan jumlah perhitungan *Average Precision* pada setiap kelas kendaraan. Dalam penelitian ini, model dengan 6000 iterasi memiliki hasil yang lebih baik jika dibandingkan dengan model yang dilatih sebanyak 8000 iterasi. Hal ini dikarenakan nilai *average precision* pada setiap kelas *overfit* sehingga deteksi hanya dapat dilakukan ketika data uji memiliki *angle*, jenis kendaraan, dan kondisi yang sama persis dengan yang ada pada data uji. Kondisi inilah yang disebut sebagai *overfit*. Adapun pertimbangan lain terkait nilai IoU yang mempengaruhi hasil deteksi.

Selain itu pada penelitian terdapat saran maupun kritik yang membangun untuk penelitian selanjutnya sebagai berikut :

- Dibutuhkan himpunan data kendaraan Indonesia yang diambil dari sudut, dan dalam kondisi yang bervariasi, sehingga meningkatkan akurasi dalam mendeteksi

kendaraan, dan mengurangi adanya kesalahan saat melakukan deteksi

- Dilakukan perincian dalam melakukan deteksi, seperti dalam mendeteksi mobil, maka kelas mobil tersebut harus terbagi dalam beberapa jenis seperti sedan, suv, truck, dan lain-lain
- Dilakukan penambahan *screenshot* secara otomatis, dan pengenalan plat nomor kendaraan yang melintasi garis pembatas saat melanggar jalur bus
- Diharapkan dapat diimplementasikan pada perangkat keras *cctv*

#### DAFTAR RUJUKAN

- [1] I. Supriyatna, "Sejumlah Tindakan Konyol Para Penerobos Jalur Busway," *Kompas.com*, 2017. <https://megapolitan.kompas.com/read/2017/12/13/14583851/sejumlah-tindakan-konyol-para-penerobos-jalur-busway?page=all> (accessed Feb. 18, 2021).
- [2] F. P. S. Arinto Tri Prabowo, "E-Tilang motor, pelanggaran terbanyak terobos jalur Transjakarta," *Viva.co.id*, 2020. <https://www.viva.co.id/berita/metro/126779-6-e-tilang-motor-pelanggaran-terbanyak-terobos-jalur-transjakarta> (accessed Aug. 03, 2020).
- [3] Y. Kim and H. Bang, "Introduction to Kalman Filter and Its Applications," *Introd. Implementations Kalman Filter*, pp. 1–16, 2019, doi: 10.5772/intechopen.80600.
- [4] R. S. Pressman, *Software Engineering: A Practitioner's Approach. 7th Edition. New York : McGraw-Hill*. 2010.
- [5] D. Mirandolle, I. Van De Weerd, and S. Brinkkemper, "Incremental method engineering for process improvement - A case study," *IFIP Adv. Inf. Commun. Technol.*, vol. 351 AICT, pp. 4–18, 2011, doi: 10.1007/978-3-642-19997-4\_3.
- [6] A. Fauzan, "Konsep Pemodelan Perangkat Lunak," *ngeneke.com*, 2019. <https://www.ngeneke.com/konsep-pemodelan-perangkat-lunak/> (accessed Mar. 30, 2021).
- [7] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *J. Phys. Conf. Ser.*, vol. 1004, no. 1, 2018, doi: 10.1088/1742-6596/1004/1/012029.