

SISTEM PEMANTAUAN KEPADATAN KENDARAAN OTOMATIS BERBASIS DETEKSI GAMBAR MENGGUNAKAN ALGORITMA YOLOV8

AUTOMATIC VEHICLE DENSITY MONITORING SYSTEM BASED ON IMAGE DETECTION USING THE YOLOV8 ALGORITHM

Gredy Christian Hendrawan Putra^{1*}, Andreas Nugroho Sihananto, S.Kom., M.Kom.¹

*E-mail: gredy24chris@gmail.com

¹Program Studi Informatika, Fakultas Ilmu Komputer,
Universitas Pembangunan Nasional “Veteran” Jawa Timur

Abstrak

Kemacetan lalu lintas perkotaan semakin kompleks, sehingga membutuhkan sistem manajemen lalu lintas yang lebih cerdas dan adaptif. Proposal ini menyajikan simulasi sistem monitoring lalu lintas yang berbasis *Intelligent Traffic Control System* (ITCS) dengan memanfaatkan deteksi volume kendaraan melalui kamera CCTV guna mengoptimalkan pemantauan lampu lalu lintas secara waktu nyata (*realtime*). Data lalu lintas dikumpulkan, disimpan dalam basis data terpusat, dan diintegrasikan dengan pemantauan langsung dan penghitungan kendaraan secara *realtime*. Tujuan utama ITCS adalah untuk meningkatkan efisiensi lalu lintas, mengurangi waktu tunggu, dan menyediakan solusi manajemen lalu lintas berbasis data. Dengan terciptanya sebuah usulan sistem simulasi ini, diharapkan kedepannya dapat diimplementasikan pada lampu lalu lintas yang sesungguhnya.

Kata kunci: *Intelligent Traffic Control System, YOLOv8, Internet of Things, Manajemen lalu lintas pintar, Sistem monitoring*

Abstract

Urban traffic congestion is becoming increasingly complex, requiring a more intelligent and adaptive traffic management system. This proposal presents a simulation of a traffic monitoring system based on an Intelligent Traffic Control System concept (ITCS) that utilizes vehicle volume detection via CCTV cameras to optimize traffic light control in real time. Traffic data is collected, stored in a centralized database, and integrated with live monitoring and real-time vehicle counting. The main objectives of ITCS are to improve traffic efficiency, reduce waiting times, and provide data-driven traffic management solutions. With the creation of this proposed simulation system, it is hoped that it can be implemented in actual traffic lights in the future.

Keywords: *Intelligent Traffic Control System, YOLOv8, Internet of Things, Smart traffic control, Monitoring system*

1. PENDAHULUAN

Kemacetan lalu lintas merupakan masalah krusial di kota-kota besar Indonesia [1], yang menyebabkan kerugian ekonomi signifikan [2] mencapai Rp 100 triliun per tahun di Jakarta—serta meningkatkan polusi udara yang berdampak buruk pada kesehatan masyarakat [3]. Salah

satu tantangan utama dalam mengatasi masalah ini adalah kurangnya data lalu lintas yang akurat dan real-time untuk pengambilan keputusan yang efektif [4].

Untuk menjawab tantangan tersebut, Sistem Pemantauan Lalu Lintas Cerdas yang berbasis *Intelligent Traffic Monitoring System* (ITCS) menawarkan solusi dengan memanfaatkan kecerdasan buatan dan kamera CCTV untuk mendeteksi dan mengklasifikasikan kendaraan secara otomatis [5][6]. Teknologi ini terbukti mampu mencapai akurasi deteksi yang tinggi [7], sehingga dapat menyediakan data fundamental yang valid untuk analisis dan manajemen lalu lintas modern.

Penelitian ini bertujuan mengembangkan sebuah prototipe sistem pemantauan kepadatan kendaraan dengan memanfaatkan teknologi visi komputer, dengan mengintegrasikan kamera CCTV lalu lintas dan algoritma deteksi objek YOLOv8 [9]. Sistem ini dirancang untuk menghitung jumlah dan jenis kendaraan di ruas jalan secara *real time* [10]. Diharapkan, sistem ini dapat menghasilkan data lalu lintas yang akurat dan terstruktur, yang dapat dimanfaatkan oleh pemangku kebijakan sebagai dasar strategis untuk perencanaan rekayasa lalu lintas dan mendukung pengembangan infrastruktur *smart city* di Indonesia.

2. METODOLOGI

Penelitian ini bertujuan untuk mengembangkan sebuah prototipe sistem pemantauan kepadatan lalu lintas secara otomatis menggunakan pendekatan computer vision. Metodologi penelitian disusun secara sistematis yang mencakup beberapa tahapan utama: akuisisi dan persiapan data, perancangan dan pemodelan sistem, serta pengujian dan evaluasi model [11]. Seluruh alur kerja penelitian ini dirancang agar dapat direplikasi untuk penelitian selanjutnya dan divalidasi hasilnya. Rincian setiap tahapan akan dijelaskan pada sub-bab berikut.

2.1 Akuisisi dan Persiapan Data

Tahap ini merupakan fondasi dari pemodelan sistem, di mana kualitas data sangat menentukan performa akhir model. Prosedur pada tahap ini dibagi menjadi tiga langkah utama: pengumpulan, pelabelan, serta pemisahan dan pra-pemrosesan data.

2.1.1 Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini berupa kumpulan gambar (*dataset*) yang diambil dari siaran langsung (*live streaming*) CCTV lalu lintas di Kota Surabaya. Pengambilan data dilakukan melalui aplikasi publik SITS (*Surabaya Intelligent Transport System*) pada tanggal 7 Januari 2025 sekitar pukul 17:50 WIB, di mana pada waktu tersebut kondisi lalu lintas cenderung padat. Total gambar yang berhasil dikumpulkan adalah 120 *frame* yang merepresentasikan berbagai skenario kepadatan, mulai dari kondisi sepi hingga padat.



Gambar 1. Pratinjau *dataset* dengan tingkat kepadatan kendaraan tinggi



Gambar 2. Pratinjau *dataset* dengan tingkat kepadatan kendaraan rendah

2.1.2 Pelabelan Data (*Data Labelling*)

Setelah data gambar terkumpul, langkah selanjutnya adalah pelabelan. Proses ini bertujuan untuk menandai setiap objek kendaraan di dalam gambar agar dapat dikenali oleh model. Pelabelan dilakukan menggunakan *online tool* bernama Roboflow. Setiap kendaraan ditandai dengan kotak pembatas (*bounding box*) dan diklasifikasikan ke dalam tiga kelas objek yang telah ditentukan, yaitu: ‘bus’, ‘car’ (mobil), dan ‘motorcycle’ (sepeda motor). Proses ini memastikan model dapat belajar membedakan jenis-jenis kendaraan yang ada di jalan raya.

2.1.3 Pemisahan dan Pra-Pemrosesan Data

Dataset yang telah dilabeli kemudian dibagi menjadi tiga bagian dengan proporsi sebagai berikut:

- Data Latih (*Train Set*): 70% dari total *dataset*, digunakan untuk melatih model.
- Data Validasi (*Validation Set*): 18% dari total *dataset*, digunakan untuk mengevaluasi performa model selama proses pelatihan dan melakukan penyesuaian *hyperparameter*.
- Data Uji (*Test Set*): 12% dari total dataset, digunakan untuk menguji performa akhir model pada data yang belum pernah dilihat sebelumnya.

Seluruh gambar dalam dataset kemudian melalui tahap pra-pemrosesan (*preprocessing*), di mana ukurannya diubah secara seragam menjadi 640x640 piksel. Selain itu, diterapkan pula teknik *Auto-Orient* untuk memastikan semua gambar memiliki orientasi yang benar.

2.2 Perancangan dan Pemodelan Sistem

Tahap ini merupakan inti dari penelitian, di mana model deteksi kendaraan dikembangkan menggunakan algoritma *You Only Look Once* versi 8 (YOLOv8). Algoritma ini dipilih karena kemampuannya dalam melakukan deteksi objek secara cepat dan akurat, sehingga sangat cocok untuk aplikasi pemantauan lalu lintas secara *real-time* [9].

Proses pemodelan dilakukan dengan memanfaatkan *tools jupyter notebook* dari Google Colaboratory yang telah disediakan oleh Roboflow, yang menyederhanakan proses pengaturan teknis. Model dilatih selama 50 *epochs* dengan parameter *learning rate* dan *batch size* yang telah dioptimalkan oleh *platform*. Alur kerja algoritma deteksi kendaraan menggunakan YOLOv8 dapat dijelaskan melalui pseudokode berikut:

Vehicle Detection Model Algorithm using YOLOv8 explained in Pseudocode

```
BEGIN
# Load YOLOv8 Model from Roboflow
```

```
MODEL ← Load_YOLOv8_Model("Roboflow_Model_Link")

# Load Video or Camera Stream
VIDEO_SOURCE ← Open_Video_Stream("camera/video.mp4")

WHILE VIDEO_SOURCE has frames DO
    FRAME ← Read_Next_Frame(VIDEO_SOURCE) # Read video frame

    # Perform Object Detection
    DETECTIONS ← MODEL.Predict(FRAME)

    FOR EACH DETECTION IN DETECTIONS DO
        IF DETECTION.Class == "Vehicle" THEN
            DRAW_BoundingBox(FRAME, DETECTION.Position)
            COUNT_VEHICLES += 1
        ENDIF
    END FOR

    # Display the Processed Frame
    Show_Frame(FRAME)

    # Save Detection Results
    Save_To_Database(COUNT_VEHICLES, Timestamp)
END WHILE

Close_Video_Stream(VIDEO_SOURCE)
END
```

2.3 Pengujian dan Evaluasi Model

Setelah model selesai dilatih, dilakukan pengujian untuk mengukur kinerjanya secara kuantitatif. Pengujian dilakukan menggunakan video sampel berdurasi 6 menit yang juga diambil dari aplikasi SITS CCTV Surabaya. Metrik evaluasi yang digunakan untuk menilai performa model adalah sebagai berikut:

- Precision:** Mengukur tingkat akurasi prediksi positif [13]. Metrik ini menjawab pertanyaan: "Dari semua objek yang dideteksi sebagai kendaraan, berapa persen yang benar-benar kendaraan?".

$$Precision = \frac{TP}{TP + FP}$$

- Recall:** Mengukur kemampuan model dalam menemukan semua objek kendaraan yang relevan di dalam gambar [13]. Metrik ini menjawab pertanyaan: "Dari semua kendaraan yang ada di gambar, berapa persen yang berhasil terdeteksi?".

$$Precision = \frac{TP}{TP + FN}$$

- F1-Score:** Rata-rata harmonik dari *Precision* dan *Recall*, memberikan skor tunggal yang menyeimbangkan kedua metrik tersebut [13].

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

- Mean Average Precision (mAP):** Metrik standar dalam deteksi objek yang mengukur akurasi model secara keseluruhan dengan mempertimbangkan kurva *Precision-Recall* pada berbagai tingkat ambang batas *Intersection over Union (IoU)* [14].

Di mana TP (*True Positive*) adalah jumlah deteksi yang benar, FP (*False Positive*) adalah jumlah deteksi yang salah, dan FN (*False Negative*) adalah jumlah objek yang tidak terdeteksi. Selain

metrik di atas, waktu inferensi (*inference time*) juga diukur untuk memastikan model dapat bekerja secara efisien dalam skenario *real-time*.

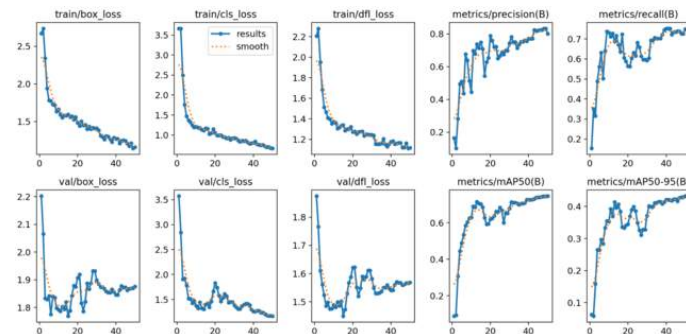
3. HASIL DAN PEMBAHASAN

3.1 Pembahasan Hasil Percobaan

Hasil evaluasi dari dataset validasi berdasarkan simulasi menunjukkan kinerja yang baik, meskipun terdapat hasil yang buruk pada salah satu metrik. Dengan set data validasi yang terdiri dari 12 gambar, model berhasil mendeteksi 462 kendaraan dari tiga kelas objek. Laporan perhitungan validasi lengkap dapat dilihat pada Tabel 1. Visualisasi hasil pelatihan model dapat dilihat pada Gambar 4.

Tabel 1. Hasil validasi menggunakan Model Kustom

12 gambar, 0 latar belakang, 0 rusak: 100% 12/12 [00:00<?, ?it/s]			Kotak (P, R, mAP50, mAP50-95): 100% 1/1 [00:01<00:00, 1,02 detik/itu]			
Kelas	Gambar	Contoh	P	R	mAP50	mAP50-95
semua	12	462	0,801	0,748	0,746	0.431
bis	5	10	0,938	0.9	0,91	0.63
mobil	11	152	0,839	0,651	0,723	0.427
sepeda motor	10	300	0,627	0,693	0,606	0,235



Gambar 3. Hasil pelatihan model

3.1.1 Precision (P)

Grafik presisi menunjukkan peningkatan yang cukup stabil selama proses pelatihan model. Seiring bertambahnya jumlah periode, presisi meningkat secara konsisten hingga mencapai 80,1%. Peningkatan ini mencerminkan kemampuan model untuk mengurangi jumlah positif palsu, sehingga lebih akurat dalam mengidentifikasi kendaraan dengan tepat tanpa mendeteksi objek yang salah.

3.1.2 Recall (R)

Grafik *recall* menunjukkan tren yang relatif stabil dengan sedikit fluktuasi selama proses pelatihan. Model meningkatkan kinerjanya dalam mendeteksi objek yang relevan seiring bertambahnya jumlah *epoch*. Di akhir pelatihan, nilai *recall* mencapai 74,8%, yang menunjukkan bahwa model mampu mengenali sebagian besar kendaraan dalam citra validasi.

3.1.3 mAP@0.5

Grafik menunjukkan pola yang cukup stabil dengan peningkatan bertahap hingga akhir pelatihan. Nilai akhir mAP@0.5 sebesar 74,6% menunjukkan bahwa model memiliki akurasi yang cukup

baik dalam mendeteksi kendaraan dengan toleransi kesalahan yang kecil pada posisi kotak pembatas.

3.1.4 mAP@0.5:0.95

Grafik mAP@0.5:0.95 secara eksplisit menunjukkan kendala yang lebih besar dalam generalisasi model untuk objek dengan ukuran dan kondisi yang berbeda [12]. Tidak seperti mAP@0.5 yang cukup tinggi, metrik ini memiliki nilai akhir yang lebih rendah, yaitu 43,1%. Hal ini menunjukkan bahwa model kesulitan mempertahankan kinerja deteksi yang tinggi ketika ambang batas IoU semakin ketat (hingga 0,95). Grafik ini kemungkinan besar memiliki tren yang lebih datar atau peningkatan yang lebih lambat dibandingkan grafik lainnya, yang menunjukkan bahwa model masih memiliki keterbatasan dalam mendeteksi objek dengan posisi kotak pembatas yang lebih presisi dengan berbagai ukuran.

3.1.5 Training Box, Cls, dan DFL Loss

Grafik *training box loss*, *cls loss*, dan *DFL loss* umumnya menunjukkan tren menurun selama proses pelatihan, yang menandakan bahwa model semakin mampu memahami dataset seiring bertambahnya *epoch*. *Box loss* menurun signifikan pada awal pelatihan lalu stabil, menunjukkan peningkatan akurasi model dalam menentukan koordinat *bounding box*. *Cls loss* juga turun secara bertahap, yang menandakan kemampuan model semakin baik dalam membedakan kelas kendaraan (mobil, sepeda motor, dan bus). Sementara itu, *DFL loss* yang berhubungan dengan distribusi prediksi koordinat *bounding box* juga konsisten menurun, menunjukkan bahwa model semakin yakin terhadap posisi dan ukuran kotak deteksi yang dihasilkan.

3.1.6 Validation Box, Cls, dan DFL Loss

Grafik *validation box loss*, *cls loss*, dan *DFL loss* menunjukkan tren yang serupa dengan *training loss*, meskipun terdapat sedikit perbedaan. *Validation box loss* cenderung menurun, tetapi sesekali mengalami fluktuasi kecil yang kemungkinan disebabkan oleh variasi pada dataset validasi. *Validation cls loss* juga mengalami penurunan, menandakan peningkatan kemampuan model dalam membedakan kelas kendaraan. Namun, apabila grafik ini menunjukkan peningkatan pada tahap akhir pelatihan, hal tersebut dapat menjadi indikasi *overfitting*, yaitu kondisi ketika model sangat baik pada data latih tetapi kurang mampu melakukan generalisasi pada data validasi. Sementara itu, *validation DFL loss* menunjukkan penurunan yang relatif stabil, mendukung peningkatan akurasi *bounding box* pada dataset validasi.

3.1.7 F1-Score

Berdasarkan hasil evaluasi model, diperoleh nilai *Precision* sebesar 80.1% (0.801) dan *Recall* sebesar 74.8% (0.748). Dengan menggunakan rumus *F1-Score*, diperoleh perhitungan sebagai berikut:

$$F1 = \frac{2 \times (0.801 \times 0.748)}{0.801 + 0.748}$$

$$F1 = \frac{2 \times 0.599}{1.549}$$

$$F1 = \frac{1.198}{1.549}$$

$$F1 = 0.774 = 77.4\%$$

Dari hasil tersebut, nilai *F1-Score* sebesar 77.4% menunjukkan keseimbangan yang cukup baik antara *Precision* dan *Recall*. Hal ini menandakan bahwa model cukup andal dalam mendeteksi kendaraan tanpa menghasilkan terlalu banyak *false positive* maupun *false negative*.

3.1.8 Inference Time

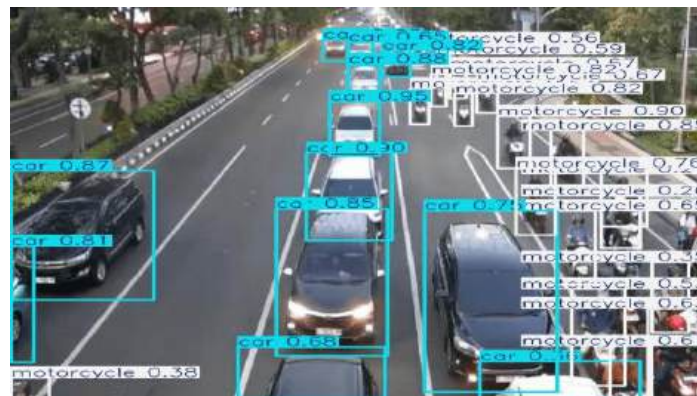
Model YOLOv8 menunjukkan efisiensi yang tinggi dalam pemrosesan gambar. Rata-rata waktu *preprocess* tercatat sebesar 0.4 ms, *inference time* sebesar 17.1 ms, serta *postprocess* sebesar 45.8 ms per gambar. Hasil ini menegaskan bahwa model mampu melakukan deteksi kendaraan dengan kecepatan tinggi, sehingga sesuai untuk kebutuhan aplikasi *realtime* seperti pada sistem ITCS.

3.2 Hasil Akhir

Berdasarkan pengamatan melalui video hasil deteksi kendaraan, diketahui bahwa *confidence score* dari setiap kendaraan rata-rata berada di atas 0,70. Hasil video deteksi kendaraan dapat diakses melalui tautan

<https://drive.google.com/drive/folders/1uGxtKEeCLv6YCCsgxUMzLA73nh7Onvx1?usp=sharing> Pemodelan YOLOv8 untuk deteksi kendaraan menggunakan *Python Notebook* dapat diakses melalui tautan

<https://colab.research.google.com/drive/1gpcKd2TkWvVywivE8V8e8g7H5aliudTP?usp=sharing>. Pratinjau hasil deteksi gambar menggunakan model YOLOv8 ditampilkan pada Gambar 5.



Gambar 4. Preview hasil deteksi kendaraan dengan *Bounding Box* dan *Confidence Score*

4. KESIMPULAN DAN SARAN

Model YOLOv8 menunjukkan performa yang cukup baik dalam mendeteksi kendaraan pada *dataset* validasi yang terdiri dari 12 gambar dengan total 462 *instances* dari tiga kelas objek. Hasil evaluasi menunjukkan bahwa model mencapai *precision* sebesar 80,1%, *recall* sebesar 74,8%, dan *mAP@0.5* sebesar 74,6%, yang menandakan tingkat deteksi yang cukup akurat. Namun, performa pada *mAP@0.5:0.95* hanya mencapai 43,1%, yang menunjukkan adanya kesulitan model dalam mendeteksi objek dengan ukuran berbeda dan kondisi yang lebih kompleks [12]. Nilai *F1-Score* sebesar 77,4% juga diperoleh, yang menandakan adanya keseimbangan yang baik antara *precision* dan *recall*.

Dari sisi efisiensi, model mampu melakukan inferensi dengan rata-rata waktu 17,1 ms per gambar, sehingga layak digunakan pada aplikasi *realtime* seperti sistem *Intelligent Traffic Control System* (ITCS). Selain itu, hasil deteksi juga menunjukkan bahwa *bounding box* memiliki skor keyakinan rata-rata di atas 0,70, yang semakin memperkuat reliabilitas model dalam skenario praktis.

5. DAFTAR RUJUKAN

- [1] F. R. Harahap, “Dampak urbanisasi bagi perkembangan kota di Indonesia,” *Jurnal Society*, vol. I, no. 1, hlm. 35–45, 2013.
- [2] D. O. M. Kawulur, A. T. Naukoko, dan M. T. B. Maramis, “Analisis dampak kemacetan terhadap ekonomi pengguna jalan, depan tugu taman kota Manado,” *Jurnal Berkala Ilmiah Efisiensi*, vol. 20, no. 1, hlm. 83–93, 2020.
- [3] W. Jatmiko, “Analisis Dampak Pemasangan ATCS Terhadap Emisi Gas Buang (CO₂) di Jl. Jend. Sudirman Kota Tangerang,” *Jurnal Pembangunan Wilayah & Kota*, vol. 9, no. 2, hlm. 134–143, 2013. [Online]. Available: <https://doi.org/10.14710/pwk.v9i2.6518>
- [4] X. Zhang, "Artificial Intelligence in Intelligent Traffic Signal Control," dalam *Proceedings of the 3rd International Conference on Software Engineering and Machine Learning*, 2025, hlm. 113–120. [Online]. Available: <https://doi.org/10.54254/2755-2721/118/2025.20846>
- [5] Alfith, “Pengembangan perancangan smart traffic light berbasis LDR sensor dan timer delay system,” *Jurnal Teknik Elektro ITP*, vol. 8, no. 1, hlm. 35–39, 2019. [Online]. Available: <https://doi.org/10.21063/jte.2019.3133807>
- [6] F. W. W. Dany, “Kecerdasan buatan kurangi 20 persen kepadatan lalu lintas Jakarta,” *Harian Kompas*, 4 Juli 2023. [Online]. Available: <https://www.kompas.id/baca/metro/2023/07/04/kecerdasan-buatan-kurangi-20-persen-kepadatan-lalu-lintas-jakarta> [Accessed 1 Agustus 2025]
- [7] P. R. Boppuru, P. K. Kukatlappalli, C. R. Chowdary, dan J. Hussain, “AI-based YOLO V4 intelligent traffic light control system,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 16, no. 4, hlm. 53–61, 2022. [Online]. Available: <https://doi.org/10.14313/jamris/4-2022/33>
- [8] V. Mandal, A. R. Mussah, P. Jin, dan Y. Adu-Gyamfi, “Artificial Intelligence Enabled Traffic Monitoring System,” *arXiv preprint arXiv:2010.01217*, 2020.
- [9] A. Farid, F. Hussain, K. Khan, M. Shahzad, U. Khan, dan Z. Mahmood, “A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments,” *Applied Sciences*, vol. 13, no. 5, hlm. 3059, 2023. [Online]. Available: <https://doi.org/10.3390/app13053059>
- [10] J. Azimjonov dan A. Özmen, “A real-time vehicle detection and a novel vehicle tracking systems for estimating and monitoring traffic flow on highways,” *Advanced Engineering Informatics*, vol. 50, hlm. 101429, 2021. [Online]. Available: <https://doi.org/10.1016/j.aei.2021.101429>
- [11] Anna, Nurmallasari, dan Y. Rohayani, “Penerapan metode waterfall dalam perancangan sistem informasi akuntansi pengiriman barang,” *Jurnal Sistem Informasi Akuntansi*, vol. 1, no. 1, hlm. 1–9, 2021. [Online]. Available: <https://doi.org/10.31294/justian.v1i1.279>
- [12] S. Kumar, S. K. Singh, S. Varshney, S. Singh, P. Kumar, B.-G. Kim, dan I.-H. Ra, “Fusion of deep sort and yolov5 for effective vehicle detection and tracking scheme in real-time

traffic management sustainable system,” *Sustainability*, vol. 15, no. 24, hlm. 16869, 2023.
[Online]. Available: <https://doi.org/10.3390/su152416869>

- [13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer, 2014, pp. 740–755. doi: https://doi.org/10.1007/978-3-319-10602-1_48