

Deteksi dan Penghitungan *Speed bump* Berbasis YOLOv8 dan DeepSORT untuk Sistem Keselamatan Berkendara

Immawan Wicaksono^{*1}, Hilman Arif Rahmawan², Ali Rizal Chaidir³, Andrita Ceriana Eska⁴, Moh Asnoer Laagu⁵

^{1,2,3,4} Teknik Elektro, Universitas Jember, Jember

e-mail: ^{*1}immawanw@unej.ac.id, ²211910201091@mail.unej.ac.id, ³ali.rizal@unej.ac.id,

⁴andrita@unej.ac.id, ⁵asnoer@unej.ac.id

Abstrak

Keselamatan berkendara merupakan aspek penting dalam sistem transportasi, terutama pada jalan yang dilengkapi *speed bump* sebagai pembatas kecepatan. *Speed bump* yang tidak terlihat jelas dapat menimbulkan ketidaknyamanan, kerusakan kendaraan, dan potensi kecelakaan. Penelitian ini mengembangkan sistem deteksi dan penghitungan *speed bump* berbasis You Only Look Once version 8 (YOLOv8) yang dikombinasikan dengan DeepSORT dan garis virtual pada video rekaman kamera smartphone. Dataset yang digunakan terdiri atas 1.000 citra *speed bump* yang dianotasi secara manual menggunakan bounding box dan dibagi menjadi data latih, validasi, dan uji dengan rasio 70:20:10. Model dilatih pada Google Colab selama 100 epoch dengan ukuran citra 640 x 640 piksel dan batch size 32. Evaluasi dilakukan menggunakan precision, recall, mAP, akurasi deteksi, akurasi counting, dan frame per second (FPS). Hasil validasi menunjukkan precision 99,85%, recall 100%, mAP@0.5 sebesar 99,50%, dan mAP@0.5:0.95 sebesar 64,16%. Pada pengujian video lapangan, confidence threshold 0,5 memberikan rata-rata akurasi deteksi tertinggi sebesar 85,50%, sedangkan threshold 0,6 dan 0,7 masing-masing menghasilkan 63,64% dan 37,01%. Kecepatan pemrosesan sistem berada pada rentang 25-42 FPS. Hasil penelitian menunjukkan bahwa YOLOv8 mampu mendeteksi *speed bump* dengan baik pada kondisi pencahayaan memadai, namun akurasi penghitungan masih dipengaruhi oleh double counting akibat ketidakstabilan ID tracking.

Kata kunci—DeepSORT; deteksi objek; keselamatan jalan; *speed bump*; YOLOv8.

1. PENDAHULUAN

Speed bump merupakan perangkat pengendali kecepatan yang dipasang untuk mengurangi risiko kecelakaan pada area tertentu, seperti lingkungan permukiman, area parkir, kawasan pendidikan, dan ruas jalan dengan aktivitas pejalan kaki yang tinggi [1]–[3]. Meskipun berfungsi meningkatkan keselamatan, *speed bump* yang tidak terdeteksi oleh pengemudi dapat menyebabkan pengereman mendadak, ketidaknyamanan, kerusakan komponen kendaraan, atau risiko kecelakaan [2], [3]. Oleh karena itu, diperlukan sistem yang dapat mendeteksi keberadaan *speed bump* secara otomatis melalui citra atau video jalan. Perkembangan computer vision dan deep learning memungkinkan pendeteksian objek jalan secara otomatis pada berbagai skenario transportasi cerdas [6]–[8], [10]–[12], [20]. Algoritma YOLO dikenal sebagai pendekatan deteksi objek satu tahap yang mampu memprediksi kelas dan lokasi objek secara efisien [8], [9]. Perkembangan keluarga YOLO dan berbagai variannya menunjukkan peningkatan pada aspek akurasi, efisiensi, dan fleksibilitas arsitektur untuk kebutuhan deteksi objek secara real-time [8], [9], [11], [13], [18]. Pada penelitian ini, YOLOv8 dipilih karena memiliki keunggulan spesifik yang relevan dengan konteks deteksi *speed bump*, yaitu proses deteksi satu tahap, arsitektur yang efisien untuk pemrosesan citra berukuran 640 x 640 piksel, dukungan pelatihan kustom pada satu kelas objek, serta kemampuan menghasilkan bounding box dan confidence score yang dapat

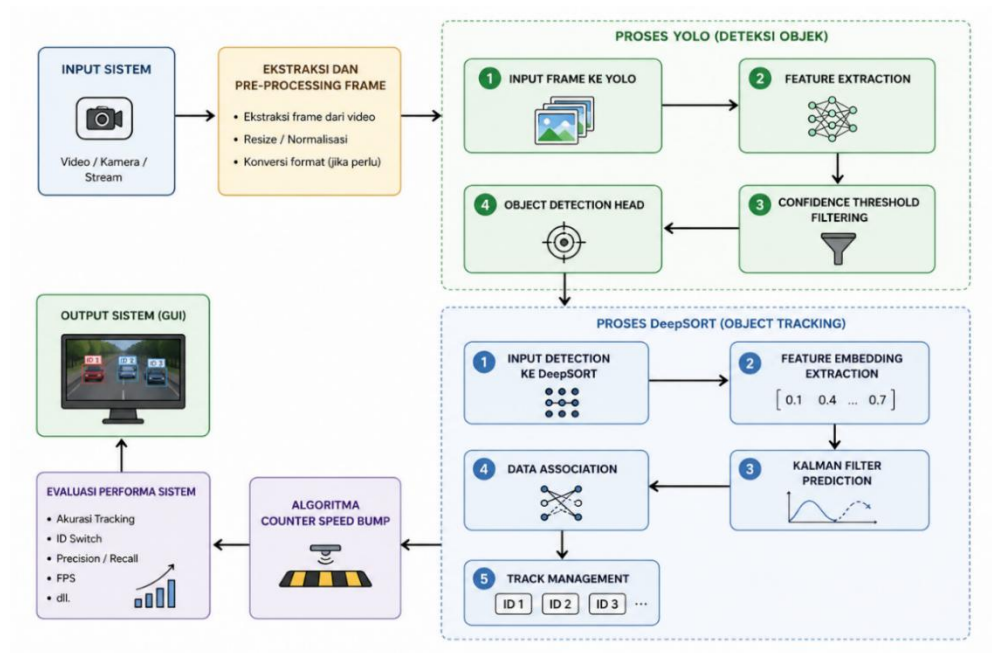
langsung digunakan pada proses tracking dan counting [8]–[11], [13]. Dalam konteks video, deteksi objek perlu dilengkapi dengan pelacakan agar identitas objek tetap konsisten antar-frame, terutama ketika hasil deteksi digunakan untuk menghitung objek yang melewati area atau garis tertentu [14]–[17]. DeepSORT dipilih karena mampu menggabungkan prediksi posisi berbasis Kalman filter dan pencocokan fitur visual, sehingga setiap objek yang terdeteksi dapat diberi ID unik selama muncul pada urutan frame. Meskipun *speed bump* merupakan objek statis, dalam video yang direkam dari kamera bergerak objek tersebut tampak mengalami perubahan posisi relatif dari frame ke frame. Oleh karena itu, mekanisme tracking tetap diperlukan agar *speed bump* yang sama tidak dihitung berulang. Penghitungan dilakukan menggunakan garis virtual, yaitu garis acuan yang ditempatkan pada area tertentu di frame video. Objek dihitung satu kali apabila titik tengah bounding box dengan ID unik melewati garis tersebut.

Beberapa penelitian sebelumnya telah membahas deteksi *speed bump* menggunakan deep neural network, kamera stereo, analisis blob, CNN, serta model deep learning ringan untuk kendaraan cerdas [2]–[5], [19]. Namun, masih terdapat keterbatasan teknis yang menjadi celah penelitian. Pendekatan berbasis kamera stereo atau ZED dapat memberikan informasi kedalaman, tetapi memerlukan perangkat khusus, kalibrasi kamera, dan biaya implementasi yang lebih tinggi dibandingkan kamera smartphone [2], [19]. Pendekatan analisis blob dan CNN relatif sederhana, tetapi performanya dapat menurun ketika warna atau tekstur *speed bump* memiliki kemiripan dengan permukaan jalan [5]. Selain itu, penggunaan kamera dashcam, CCTV, dan UAV memiliki karakteristik berbeda dengan kamera smartphone karena perbedaan posisi kamera, cakupan area, sudut pengambilan, dan kestabilan perekaman [12], [20]. Dashcam umumnya dipasang tetap pada kendaraan sehingga sudut pandang dan ketinggiannya lebih stabil, sedangkan CCTV bersifat statis dan hanya mencakup area jalan tertentu. Sebaliknya, kamera smartphone lebih fleksibel tetapi memiliki tantangan berupa variasi sudut pengambilan, getaran, perubahan posisi, dan perbedaan kualitas pencahayaan. Berdasarkan gap tersebut, penelitian ini menerapkan YOLOv8 untuk mendeteksi *speed bump* dari video rekaman kamera smartphone dan mengintegrasikannya dengan DeepSORT serta garis virtual untuk menghitung jumlah *speed bump* secara otomatis pada video offline dengan variasi pencahayaan pagi, siang, dan malam.

2. METODE PENELITIAN

2.1. Desain Sistem

Sistem yang dikembangkan terdiri atas empat tahap utama, yaitu akuisisi video, prapemrosesan frame, deteksi *speed bump* menggunakan YOLOv8, dan penghitungan objek menggunakan DeepSORT serta garis virtual. Integrasi deteksi objek dan pelacakan banyak digunakan pada sistem pemantauan lalu lintas karena mampu menghubungkan hasil deteksi antar-frame sebelum dilakukan proses counting [14]–[17]. Video masukan diperoleh dari rekaman kamera smartphone, kemudian dipecah menjadi frame. Setiap frame diubah ukurannya menjadi 640×640 piksel dan diproses oleh model YOLOv8 untuk menghasilkan bounding box, confidence score, dan label objek. Bounding box hasil deteksi kemudian diteruskan ke DeepSORT agar setiap *speed bump* memiliki ID yang konsisten. Counter bertambah satu apabila titik tengah bounding box dengan ID unik melewati garis virtual yang telah ditentukan. Secara umum, alur kerja sistem deteksi dan penghitungan *speed bump* yang dikembangkan dalam penelitian ini ditunjukkan pada Gambar 1.

Gambar 1. Blok diagram sistem deteksi dan penghitungan *speed bump*

2.2. Dataset dan Anotasi

Dataset penelitian terdiri atas 1.000 citra *speed bump* berformat JPG dengan resolusi rata-rata 816 x 460 piksel. Anotasi dilakukan secara manual menggunakan makesense.ai dengan format keluaran YOLO TXT. Contoh hasil anotasi bounding box pada citra *speed bump* ditunjukkan pada Gambar 2, sedangkan pembagian dataset penelitian ditampilkan pada Tabel 1.

Gambar 2. Contoh anotasi bounding box pada dataset *speed bump*

Tabel 1 menampilkan bahwa data dibagi menjadi 700 citra untuk pelatihan, 200 citra untuk validasi, dan 100 citra untuk pengujian. Pembagian ini digunakan untuk memastikan model memiliki data pelatihan yang cukup sekaligus menyediakan data validasi dan pengujian untuk mengevaluasi generalisasi model.

Tabel 1. Pembagian Dataset Penelitian

Subset	Proporsi	Jumlah citra
Pelatihan	70%	700
Validasi	20%	200
Pengujian	10%	100

2.3. *Pelatihan Model YOLOv8*

Pelatihan dilakukan menggunakan Google Colab dengan model YOLOv8m. Pemilihan YOLOv8 didasarkan pada kemampuannya menghasilkan keseimbangan antara akurasi dan efisiensi komputasi pada berbagai tugas deteksi objek [6], [8]–[11], [13]. Parameter pelatihan yang digunakan adalah epoch 100, ukuran citra 640×640 piksel, batch size 32, dan perangkat GPU Tesla T4. Model terbaik disimpan sebagai best.pt dan digunakan pada tahap validasi serta pengujian video lapangan.

2.4. *Tracking dan Counting Berbasis DeepSORT*

DeepSORT digunakan untuk melacak objek *speed bump* yang telah terdeteksi oleh YOLOv8. Pendekatan serupa telah banyak digunakan pada sistem deteksi, tracking, dan counting objek berbasis video karena mampu memberikan ID unik pada objek yang terdeteksi antar-frame [14]–[17]. Mekanisme tracking memadukan prediksi posisi menggunakan Kalman filter dan asosiasi data berdasarkan kesesuaian deteksi antar-frame. Penghitungan objek dilakukan dengan garis virtual. Apabila titik tengah bounding box *speed bump* melewati garis tersebut, sistem menambahkan nilai counter. ID objek yang telah dihitung disimpan agar tidak dihitung berulang. Namun, apabila tracking tidak stabil dan ID objek berubah, potensi double counting dapat terjadi.

2.5. *Evaluasi Kinerja*

Kinerja model dievaluasi menggunakan precision, recall, mAP@0.5, mAP@0.5:0.95, akurasi deteksi, akurasi counting, dan FPS. Precision dan recall digunakan untuk menilai ketepatan dan kelengkapan deteksi objek, sedangkan mAP digunakan untuk mengevaluasi kualitas prediksi bounding box pada ambang IoU tertentu [6], [8]–[11], [13]. Akurasi deteksi dihitung berdasarkan jumlah *speed bump* yang berhasil dideteksi dibandingkan jumlah ground truth. Akurasi counting dihitung berdasarkan kedekatan jumlah hasil counter sistem terhadap jumlah ground truth. FPS digunakan untuk menilai kecepatan pemrosesan sistem pada video uji. Pengujian video dilakukan pada dua lokasi, yaitu rute Jalan Gunung Agung–Jalan Rotawu, Sumbersari, Jember dengan tujuh *speed bump* dan kawasan perumahan dengan sebelas *speed bump*. Pengujian dilakukan pada kondisi pagi, siang, dan malam, serta menggunakan confidence threshold 0,5; 0,6; dan 0,7.

3. HASIL DAN PEMBAHASAN

3.1 *Tahapan Penerapan YOLOv8 dan DeepSORT*

Tahapan penerapan YOLOv8 dan DeepSORT pada penelitian ini dilakukan secara berurutan mulai dari pemrosesan video masukan hingga penghitungan jumlah *speed bump*. Video hasil rekaman kamera smartphone terlebih dahulu diekstraksi menjadi frame. Setiap frame kemudian melalui tahap prapemrosesan berupa penyesuaian ukuran citra menjadi 640×640 piksel agar sesuai dengan ukuran input model YOLOv8. Setelah itu, frame diproses oleh model YOLOv8 untuk mendeteksi objek *speed bump* dan menghasilkan keluaran berupa bounding box, label kelas, serta nilai confidence score. Hasil deteksi dari YOLOv8 selanjutnya diteruskan ke algoritma DeepSORT untuk proses tracking. DeepSORT memberikan ID unik pada setiap *speed bump* yang terdeteksi dan mempertahankan ID tersebut antar-frame berdasarkan informasi posisi dan fitur visual objek. Meskipun *speed bump* merupakan objek statis, kamera yang bergerak menyebabkan objek tampak berpindah posisi relatif pada video, sehingga proses tracking tetap diperlukan agar objek yang sama tidak dihitung berulang. Setelah ID objek diperoleh, sistem menerapkan garis virtual sebagai acuan counting. Counter akan bertambah satu ketika titik tengah bounding box dengan ID unik melewati garis virtual yang telah ditentukan. Tahapan ini menghasilkan keluaran akhir berupa video dengan bounding box, ID tracking, confidence score, nilai FPS, dan jumlah *speed bump* yang terhitung.

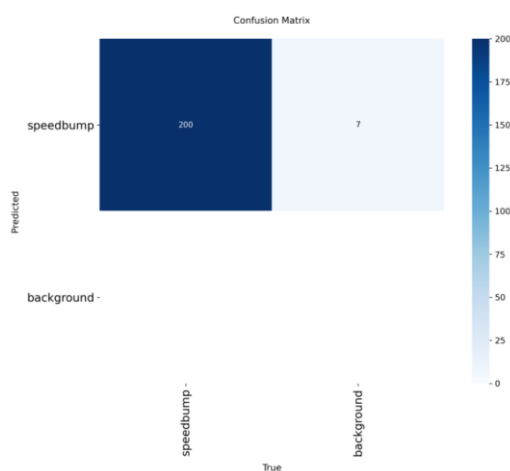
3.2 Hasil Validasi Model

Hasil validasi ditampilkan oleh Tabel 2 dan Gambar 3 yang menunjukkan bahwa model memiliki performa deteksi yang tinggi. Nilai precision sebesar 99,85% menunjukkan bahwa hampir seluruh prediksi *speed bump* merupakan prediksi benar. Nilai recall 100% menunjukkan bahwa seluruh objek *speed bump* pada data validasi berhasil dikenali. Nilai mAP@0.5 sebesar 99,50% menunjukkan kualitas deteksi sangat baik pada ambang IoU standar, sedangkan mAP@0.5:0.95 sebesar 64,16% menunjukkan bahwa presisi lokasi bounding box masih dapat ditingkatkan pada ambang IoU yang lebih ketat.

Tabel 2. Hasil Validasi Model YOLOv8

Metrik	Nilai
Precision	99,85%
Recall	100%
mAP@0.5	99,50%
mAP@0.5:0.95	64,16%
mAP pelatihan @0.5	0,988
mAP pelatihan @0.5:0.95	0,637

Pada Gambar 3 dapat dilihat bahwa model berhasil mendeteksi semua speedbump yang ada tanpa terlewat satu pun. Kesalahan model hanya 7 kasus dari total 207 objek yang mengindikasikan model tidak pernah gagal mengenali speedbump yang sebenarnya ada di dalam frame.



Gambar 3. Confusion matrix hasil pelatihan model YOLOv8

3.3 Pengaruh Confidence Threshold terhadap Deteksi

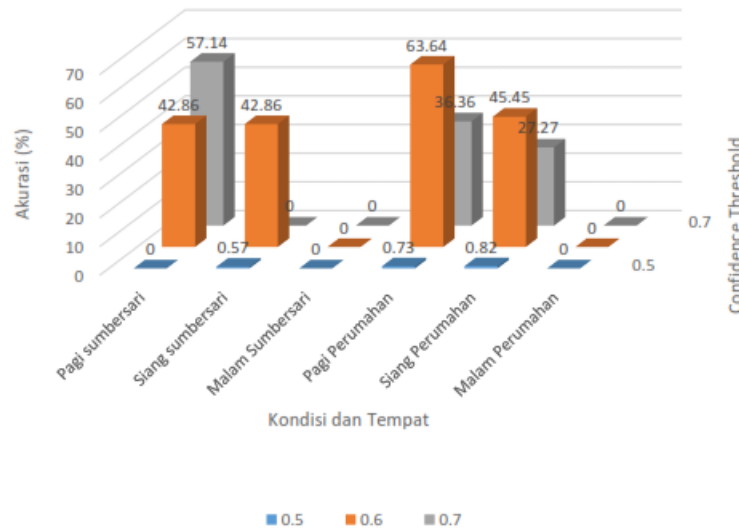
Pengujian pada video lapangan pada Tabel 3 dan Gambar 4 menunjukkan bahwa confidence threshold berpengaruh langsung terhadap kemampuan sistem mendeteksi *speed bump*.

Tabel 3. Pengaruh Confidence Threshold terhadap Akurasi Deteksi *Speed bump*

Threshold	Jumlah skenario	Rata-rata akurasi deteksi	Kecenderungan hasil
0,5	6	85,50%	Paling baik untuk deteksi
0,6	6	63,64%	Menurun karena deteksi ber-confidence sedang tereliminasi
0,7	6	37,01%	Terlalu ketat untuk video lapangan

Threshold 0,5 menghasilkan rata-rata akurasi deteksi tertinggi sebesar 85,50%. Ketika threshold dinaikkan menjadi 0,6, rata-rata akurasi deteksi menurun menjadi 63,64%. Pada threshold 0,7, akurasi deteksi turun lebih jauh menjadi 37,01%. Penurunan ini menunjukkan bahwa beberapa *speed bump* memiliki confidence score di bawah ambang tinggi, terutama pada

kondisi visual yang kurang ideal seperti pencahayaan rendah, permukaan jalan yang kurang kontras, atau sudut pandang kamera yang berubah.



Gambar 4. Pengaruh confidence threshold terhadap akurasi counting speed bump

3.4 Hasil Pengujian Deteksi dan Counting

Hasil pada Tabel 4 memperlihatkan bahwa deteksi objek dan counting memiliki karakteristik berbeda. Pada beberapa skenario, deteksi speed bump dapat mencapai 100%, tetapi hasil counter menjadi jauh lebih besar daripada ground truth.

Tabel 4. Hasil Pengujian Deteksi dan Counting Speed bump pada Video Lapangan

Kondisi	Th.	GT	Terdeteksi	Akurasi deteksi	Hasil counter	Akurasi counting
Pagi Sumbersari	0,5	7	6	85,71%	18	0%
Siang Sumbersari	0,5	7	7	100%	10	57,14%
Malam Sumbersari	0,5	7	7	100%	42	0%
Pagi Perumahan	0,5	11	8	72,73%	8	72,73%
Siang Perumahan	0,5	11	6	54,55%	9	81,82%
Malam Perumahan	0,5	11	11	100%	39	0%
Pagi Sumbersari	0,6	7	5	71,43%	11	42,86%
Siang Sumbersari	0,6	7	3	42,86%	3	42,86%
Malam Sumbersari	0,6	7	6	85,71%	31	0%
Pagi Perumahan	0,6	11	6	54,55%	7	63,64%
Siang Perumahan	0,6	11	3	27,27%	5	45,45%
Malam Perumahan	0,6	11	11	100%	22	0%
Pagi Sumbersari	0,7	7	2	28,57%	4	57,14%
Siang Sumbersari	0,7	7	0	0%	0	0%
Malam Sumbersari	0,7	7	4	57,14%	17	0%
Pagi Perumahan	0,7	11	4	36,36%	4	36,36%
Siang Perumahan	0,7	11	2	18,18%	3	27,27%
Malam Perumahan	0,7	11	9	81,82%	27	0%

Dari Tabel 4 terlihat pada skenario malam Sumbersari dengan threshold 0,5, seluruh speed bump terdeteksi, tetapi counter membaca 42 objek dari ground truth 7 objek. Hal ini menunjukkan adanya double counting yang disebabkan oleh perubahan ID saat tracking. Dengan demikian, keberhasilan deteksi YOLOv8 belum selalu menjamin keberhasilan counting apabila kestabilan tracking belum optimal seperti pada Gambar 5.



Gambar 5. Contoh hasil deteksi dan counting *speed bump* pada video lapangan

3.5 Performa Kecepatan Sistem

Kecepatan pemrosesan sistem pada pengujian video berada pada rentang 25–42 FPS dengan rata-rata keseluruhan sekitar 31 FPS yang terlihat pada Tabel 5. Nilai ini menunjukkan bahwa sistem telah mampu melakukan pemrosesan video dengan kecepatan mendekati real-time. Namun, performa masih bersifat fluktuatif karena dipengaruhi oleh proses GUI, pembacaan video, rendering frame, tracking DeepSORT, counting, serta pencatatan FPS. Perbedaan antara kecepatan inferensi model pada tahap validasi dan kecepatan sistem video lapangan terjadi karena pada pengujian lapangan sistem tidak hanya menjalankan deteksi YOLOv8, tetapi juga melakukan pelacakan, penghitungan, visualisasi, dan penyimpanan data performa secara bersamaan.

Tabel 5. Performa Kecepatan Pemrosesan Sistem Berbasis FPS

Statistik FPS	Nilai
Minimum rata-rata FPS	25
Maksimum rata-rata FPS	42
Rata-rata seluruh skenario	31
Rentang FPS minimum	19-26
Rentang FPS maksimum	47-105

Berdasarkan Tabel 5, rata-rata FPS sistem berada pada nilai 31 FPS, dengan minimum rata-rata 25 FPS dan maksimum rata-rata 42 FPS. Nilai ini menunjukkan bahwa sistem telah mampu memproses video pada kisaran kecepatan yang layak untuk pemrosesan mendekati real-time. Namun, rentang FPS minimum 19–26 dan rentang FPS maksimum 47–105 menunjukkan bahwa performa masih fluktuatif antar-skenario. Fluktuasi tersebut dipengaruhi oleh kompleksitas frame, kondisi pencahayaan, jumlah deteksi yang diproses, proses tracking DeepSORT, rendering tampilan GUI, serta pencatatan data performa. Dengan demikian, meskipun rata-rata FPS sudah cukup baik, kestabilan kecepatan pemrosesan masih perlu ditingkatkan agar sistem lebih andal ketika diterapkan pada perangkat dengan sumber daya komputasi terbatas

3.6 Pembahasan

Secara umum, YOLOv8 mampu mengenali pola visual *speed bump* dengan baik. Hal ini ditunjukkan oleh nilai precision dan recall yang tinggi pada tahap validasi. Nilai recall 100% mengindikasikan bahwa model sangat sensitif terhadap keberadaan *speed bump*, sedangkan precision 99,85% menunjukkan bahwa sebagian besar prediksi yang dihasilkan merupakan deteksi yang benar. Nilai mAP@0.5 yang tinggi juga menunjukkan bahwa model mampu menghasilkan bounding box yang cukup baik pada ambang IoU standar. Namun, nilai

mAP@0.5:0.95 yang lebih rendah menunjukkan bahwa presisi lokasi bounding box masih dapat ditingkatkan pada ambang IoU yang lebih ketat. Pada pengujian video lapangan, confidence threshold berpengaruh terhadap jumlah objek yang berhasil dideteksi. Threshold 0,5 menghasilkan akurasi deteksi tertinggi karena prediksi dengan confidence sedang tetap diproses oleh sistem. Sebaliknya, threshold 0,6 dan 0,7 cenderung mengeliminasi beberapa deteksi yang sebenarnya benar, terutama pada kondisi visual yang kurang ideal seperti pencahayaan rendah, kontras jalan yang lemah, dan perubahan sudut pandang kamera.

DeepSORT berperan penting dalam proses counting karena sistem perlu mempertahankan ID objek antar-frame sebelum objek dihitung ketika melewati garis virtual. Pada kasus *speed bump*, objek sebenarnya bersifat statis, tetapi tampak bergerak relatif terhadap kamera karena video direkam dari kendaraan yang bergerak. Kondisi ini menyebabkan posisi, ukuran, dan tampilan *speed bump* berubah dari frame ke frame. Apabila DeepSORT mampu mempertahankan ID yang sama, satu *speed bump* dapat dihitung satu kali. Namun, apabila terjadi perubahan ID akibat ketidakstabilan bounding box, pencahayaan rendah, atau perubahan perspektif, objek yang sama dapat dianggap sebagai objek baru sehingga menyebabkan double counting.

Dengan demikian, hasil penelitian menunjukkan bahwa performa deteksi YOLOv8 sudah baik, tetapi performa counting masih sangat dipengaruhi oleh kestabilan tracking. Hal ini terlihat pada beberapa skenario ketika deteksi mencapai 100%, tetapi hasil counter jauh lebih besar daripada jumlah ground truth. Kondisi tersebut menunjukkan bahwa keberhasilan deteksi objek belum selalu menjamin keberhasilan penghitungan apabila mekanisme tracking dan aturan garis virtual belum optimal.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan YOLOv8 untuk deteksi *speed bump* dan DeepSORT dengan garis virtual untuk penghitungan objek pada video rekaman kamera smartphone. Model validasi memperoleh precision 99,85%, recall 100%, mAP@0.5 sebesar 99,50%, dan mAP@0.5:0.95 sebesar 64,16%. Pada pengujian video lapangan, threshold 0,5 memberikan rata-rata akurasi deteksi terbaik sebesar 85,50%, sedangkan threshold 0,6 dan 0,7 masing-masing menghasilkan 63,64% dan 37,01%. Kecepatan sistem berada pada rentang 25–42 FPS, sehingga sistem telah menunjukkan kemampuan pemrosesan yang mendekati real-time, meskipun kestabilan performa masih perlu ditingkatkan pada skenario video lapangan. Keterbatasan utama terletak pada akurasi counting akibat double counting dan ketidakstabilan ID tracking, terutama pada kondisi pencahayaan rendah atau tampilan *speed bump* yang kurang kontras.

5. SARAN

Penelitian selanjutnya disarankan untuk memperluas dataset dengan variasi jenis *speed bump*, kondisi jalan, sudut kamera, cuaca, kecepatan kendaraan, dan pencahayaan yang lebih beragam. Selain itu, performa sistem pada kondisi malam hari perlu ditingkatkan melalui augmentasi data atau image enhancement. Mekanisme tracking dan counting juga perlu dioptimalkan untuk mengurangi double counting, misalnya dengan memperkuat aturan garis virtual menggunakan region of interest (ROI), arah gerak objek, dan filter temporal agar satu objek hanya dihitung ketika memenuhi kriteria lintasan tertentu. Pengaturan parameter DeepSORT atau penggunaan algoritma pelacakan lain juga dapat dipertimbangkan untuk meningkatkan kestabilan ID objek. Dari sisi komputasi, sistem dapat dioptimasi menggunakan model yang lebih ringan, TensorRT, OpenVINO, atau teknik frame skipping agar lebih stabil untuk penerapan berbasis edge computing.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada LPPM yang telah memberikan dukungan pendanaan terhadap penelitian ini. Penulis juga mengucapkan terima kasih kepada Program Studi

Teknik Elektro, Fakultas Teknik, Universitas Jember atas dukungan fasilitas dan bimbingan selama pelaksanaan penelitian ini.

DAFTAR PUSTAKA

- [1] Kementerian Perhubungan Republik Indonesia, 2021, Peraturan Menteri Perhubungan Nomor 14 Tahun 2021 tentang Alat Pengendali dan Pengaman Pengguna Jalan, Kementerian Perhubungan Republik Indonesia, Jakarta.
- [2] Peralta-López, J.-E., Morales-Viscaya, J.-A., Lázaro-Mata, D., Villaseñor-Aguilar, M.-J., Prado-Olivarez, J., Pérez-Pinal, F.-J., Padilla-Medina, J.-A., Martínez-Nolasco, J.-J., and Barranco-Gutiérrez, A.-I., 2023, *Speed bump* and Pothole Detection Using Deep Neural Network with Images Captured through ZED Camera, *Applied Sciences*, Vol. 13, No. 14, Art. no. 8349, doi: <https://doi.org/10.3390/app13148349>.
- [3] Wang, R., Luo, X., Ye, Q., Jiang, Y., and Liu, W., 2024, Research on Visual Perception of *Speed bumps* for Intelligent Connected Vehicles Based on Lightweight FPNNet, *Sensors*, Vol. 24, No. 7, Art. no. 2130, doi: <https://doi.org/10.3390/s24072130>.
- [4] Xiang, X., Chen, X., Tang, X., Luo, Q., and Ding, Y., 2024, Research on Detection of Multiple Types of *Speed bump* Defects Based on CRSCCG-YOLOv5s, *IEEE Access*, Vol. 12, pp. 116786–116800.
- [5] Nugraha, E. I., and Pranoto, Y. M., 2021, Deteksi Polisi Tidur pada Jalan Menggunakan Analisis Blob dan Konvolusional Neural Network, *Journal of Intelligent Systems and Computation*, pp. 16–22.
- [6] Taufiqurrahman, T., Hadi, A. P., and Siregar, R. E., 2024, Evaluasi Performa YOLOv8 dalam Deteksi Objek di Depan Kendaraan dengan Variasi Kondisi Lingkungan, *Jurnal Minfo Polgan*, Vol. 13, No. 2, pp. 1755–1773.
- [7] Santoso, A. D., Nugroho, J. B., and Setyaningtyas, E. P., 2024, Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV dan YOLOv4, *Jurnal AI dan SPK*, Vol. 2, No. 2, pp. 127–131.
- [8] Megantara, N. A., and Utami, E., 2025, Object Detection Using YOLOv8: A Systematic Review, *Sistemasi: Jurnal Sistem Informasi*, Vol. 14, No. 3.
- [9] Jocher, G., Chaurasia, A., and Qiu, J., 2023, Ultralytics YOLO, Software documentation, URL: <https://docs.ultralytics.com/>.
- [10] Zhao, R., Tang, S. H., Supeni, E. E. B., Rahim, S. A., and Fan, L., 2024, Z-YOLOv8s-Based Approach for Road Object Recognition in Complex Traffic Scenarios, *Alexandria Engineering Journal*, Vol. 106, pp. 298–311, doi: <https://doi.org/10.1016/j.aej.2024.07.011>.
- [11] Khalili, B., and Smyth, A. W., 2024, SOD-YOLOv8: Enhancing YOLOv8 for Small Object Detection in Traffic Scenes, *arXiv preprint arXiv:2408.04786*, URL: <https://arxiv.org/abs/2408.04786>.
- [12] Zhu, Y., Wang, Y., An, Y., Yang, H., and Pan, Y., 2024, Real-Time Vehicle Detection and Urban Traffic Behavior Analysis Based on UAV Traffic Videos on Mobile Devices, *arXiv preprint arXiv:2402.16246*, URL: <https://arxiv.org/abs/2402.16246>.

- [13] Schneidereit, T., Gohrenz, S., and Breuß, M., 2025, Object Detection Characteristics in a Learning Factory Environment Using YOLOv8, arXiv preprint arXiv:2503.10356, URL: <https://arxiv.org/abs/2503.10356>.
- [14] Cahyani, P. A., Mardiana, M., Wintoro, P. B., and Muhammad, M. A., 2024, Vehicle Counting System Using YOLOv5 and DeepSORT Algorithms, Jurnal Teknik Informatika dan Sistem Informasi, Vol. 10, No. 1, pp. 86–99, doi: <https://doi.org/10.28932/jutisi.v10i1.7519>.
- [15] Kumar, S., Singh, S. K., Varshney, S., Singh, S., Kumar, P., Kim, B.-G., and Ra, I.-H., 2023, 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, Art. no. 16869, doi: <https://doi.org/10.3390/su152416869>.
- [16] Yang, F., Zhang, X., and Liu, B., 2022, Video Object Tracking Based on YOLOv7 and DeepSORT, arXiv preprint arXiv:2207.12202, URL: <https://arxiv.org/abs/2207.12202>.
- [17] Durve, M., Orsini, S., Tiribocchi, A., Montessori, A., Tucny, J.-M., Lauricella, M., Camposeo, A., Pisignano, D., and Succi, S., 2023, Benchmarking YOLOv5 and YOLOv7 Models with DeepSORT for Droplet Tracking Applications, arXiv preprint arXiv:2301.08189, URL: <https://arxiv.org/abs/2301.08189>.
- [18] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M., 2022, YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors, arXiv preprint arXiv:2207.02696, URL: <https://arxiv.org/abs/2207.02696>.
- [19] Liang, C., Fu, J., Luo, L., and Yu, M., 2025, SBP-YOLO: A Lightweight Real-Time Model for Detecting *Speed bumps* and Potholes, arXiv preprint arXiv:2508.01339, URL: <https://arxiv.org/abs/2508.01339>.
- [20] Antariksa, G., Chakraborty, R., Somvanshi, S., Das, S., Jalayer, M., Patel, D. R., and Mills, D., 2025, Comparative Analysis of Advanced AI-Based Object Detection Models for Pavement Marking Quality Assessment during Daytime, arXiv preprint arXiv:2503.11008, URL: <https://arxiv.org/abs/2503.11008>.