



Perancangan Sistem Vision-Based Traffic Monitoring untuk Penghitungan dan Estimasi Kecepatan Kendaraan

Moch. Akbar Hanafi¹, Shofwa Aulia Mursyidah², Irgieta Villa Cantika³, Hasanuddin Al-Habib⁴, Yuni Rosita Dewi⁵

^{1, 2, 3, 4, 5} Program Studi Sains Data, Universitas Negeri Surabaya

¹ mochakar.23018@mhs.unesa.ac.id

² shofwa.23081@mhs.unesa.ac.id

³ irgieta.23107@mhs.unesa.ac.id

⁴ hasanuddinhabib@unesa.ac.id

⁵ yunidewi@unesa.ac.id

Corresponding author email: hasanuddinhabib@unesa.ac.id

Abstract: *Advancements in traffic engineering and the implementation of smart city concepts have driven the development of various automated methods for traffic monitoring, including the automatic estimation of vehicle speed. One such approach leverages computer vision techniques by utilizing surveillance cameras to capture and digitally analyze vehicle movements. In this study, the Mixture of Gaussians (MoG) method is combined with the ORB (Oriented FAST and Rotated BRIEF) algorithm to enable more precise measurement of distances between vehicle trajectories. Additionally, the system incorporates ORB feature matching across frames, allowing the Region of Interest (ROI) to adapt to minor shifts in camera position dynamically. The results indicate the method's accuracy in vehicle counting and speed estimation*

Keywords: *ORB, vehicle counting, speed detection, traffic surveillance, digital image processing*

Abstrak: Kemajuan dalam rekayasa lalu lintas serta implementasi konsep smart city telah mendorong munculnya berbagai metode otomatisasi dalam pemantauan lalu lintas, termasuk di antaranya pengukuran kecepatan kendaraan secara otomatis. Salah satu pendekatan yang digunakan berbasis visi komputer, dengan memanfaatkan kamera pengawas untuk merekam dan menganalisis pergerakan kendaraan secara digital. Dalam penelitian ini, pendekatan Mixture of Gaussians dikombinasikan dengan metode ORB (Oriented FAST and Rotated BRIEF) untuk memungkinkan pengukuran jarak antar titik lintas kendaraan dengan tingkat presisi yang lebih tinggi. Selain itu, sistem dilengkapi dengan fitur pencocokan ORB antar frame, sehingga Region of Interest (ROI) dapat menyesuaikan secara dinamis terhadap perubahan posisi kamera yang bersifat minor. Hasil penelitian menunjukkan tingkat akurasi metode dalam melakukan penghitungan kendaraan dan estimasi kecepatan.

Kata kunci: ORB, penghitungan kendaraan, estimasi kecepatan kendaraan, traffic surveillance, digital image processing

1. PENDAHULUAN

Rekayasa lalu lintas dan pengembangan konsep *smart city* telah mendorong lahirnya berbagai metode untuk otomatisasi pemantauan lalu lintas, salah satunya adalah mengukur kecepatan kendaraan secara otomatis. Salah satu pendekatan berbasis visi komputer, yang memanfaatkan kamera pengawas untuk merekam pergerakan kendaraan dan menganalisisnya secara digital. Sistem seperti ini banyak digunakan untuk meningkatkan keselamatan lalu lintas, menindak pelanggaran batas kecepatan, dan mendukung sistem transportasi cerdas.

Sebagian besar pendekatan terkini untuk deteksi dan pelacakan kendaraan menggunakan model pra-latih (*pre-trained*) berbasis *deep learning* seperti YOLO[1] [2], SSD [3], [4], atau Faster R-CNN [5], [6], yang memang terbukti sangat efektif dalam mendeteksi objek secara akurat di berbagai kondisi pencahayaan dan sudut pandang. Namun, penggunaan model pre-trained semacam ini memiliki beberapa kekurangan mendasar, antara lain kebutuhan daya komputasi tinggi, ketergantungan pada GPU, dan sering kali kurang fleksibel untuk digunakan di perangkat *edge* atau sistem *embedded* dengan sumber daya terbatas. Selain itu, tantangan dalam mengurangi *bitrate* [7] dalam kompresi video dan reduksi noise [8], [9], [10], [11], [12], [13] masih menjadi tantangan yang perlu diperhatikan.

Sebagai alternatif, penelitian ini mengusulkan pendekatan murni berbasis pemrosesan citra klasik, tanpa melibatkan model *deep learning* atau sistem pra-latih. Dengan memanfaatkan teknik-teknik

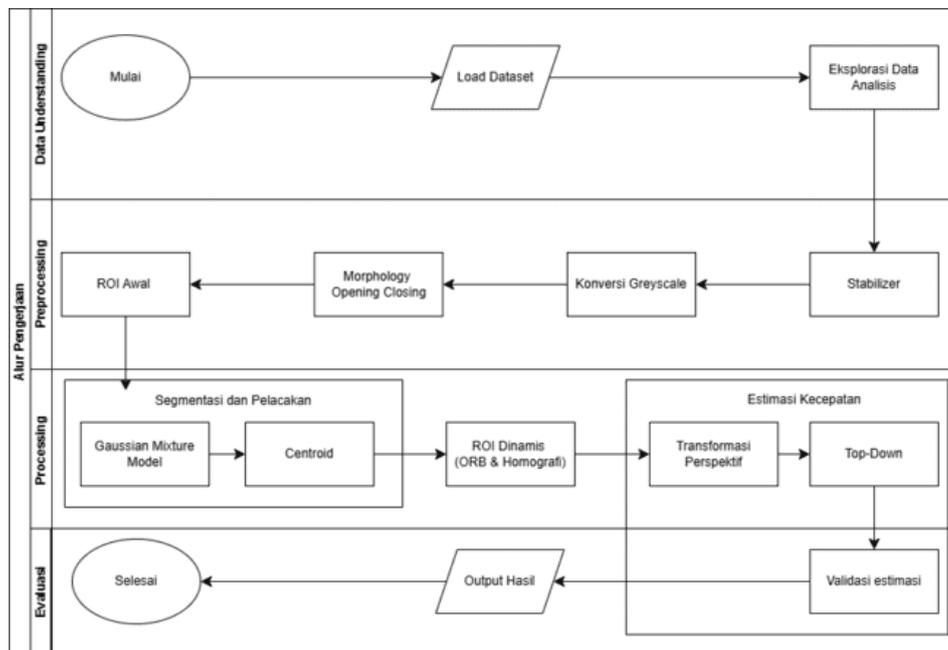
seperti *Background Substraction* [14], ORB (*Oriented FAST and Rotated BRIEF*) [15], transformasi perspektif [16] untuk mendapatkan tampilan *bird's-eye view*[17], serta algoritma pelacakan centroid untuk mengidentifikasi dan melacak kendaraan [18], sistem ini dirancang agar tetap efisien, ringan, dan dapat berjalan secara *real-time* bahkan di perangkat dengan kemampuan komputasi terbatas .

Metode yang diajukan mengandalkan proses kalibrasi secara manual melalui klik titik acuan pada citra video untuk menghasilkan transformasi homografi, yang memungkinkan konversi perspektif jalan dari tampilan miring ke tampilan atas. Dengan transformasi ini, sistem dapat melakukan pengukuran jarak antar titik lintas kendaraan secara lebih presisi. Selain itu, sistem juga dilengkapi dengan pencocokan fitur ORB dari *frame* ke *frame* agar posisi ROI (*Region Of Interest*) dapat beradaptasi terhadap perubahan kecil dalam orientasi atau posisi kamera .

Salah satu kelebihan utama dari pendekatan ini adalah kesederhanaan dan efisiensinya. Dibandingkan dengan pendekatan berbasis *deep learning*, sistem ini tidak membutuhkan dataset pelatihan, tidak memerlukan proses inferensi yang berat, dan dapat dengan mudah disesuaikan dengan kondisi jalan yang berbeda hanya dengan rekalisasi titik ROI secara manual. Hal ini membuatnya ideal untuk diterapkan di daerah atau sistem dengan sumber daya terbatas namun tetap membutuhkan solusi pemantauan kecepatan yang akurat.

Dengan pendekatan tersebut, penelitian ini bertujuan membangun sistem pengukuran kecepatan kendaraan berbasis video yang akurat, stabil, dan dapat berjalan secara mandiri menggunakan teknik pemrosesan citra klasik. Harapannya, sistem ini dapat menjadi solusi alternatif yang murah dan dapat diimplementasikan secara luas di berbagai lingkungan lalu lintas.

2. METODE PENELITIAN



Gambar 1. Diagram Alur Penelitian

2.1 Background Substraction

Deteksi objek pada sistem ini dilakukan melalui pemisahan latar belakang menggunakan pendekatan *Background Subtractor MOG2 (Mixture of Gaussians 2)*. Metode ini membentuk model statistik untuk setiap piksel sebagai kombinasi dari beberapa distribusi Gaussian [19],

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \cdot \mathcal{N}(X_t; \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

Pada persamaan 1, $\omega_{i,t}$ adalah bobot, $\mu_{i,t}$ rata-rata, dan $\Sigma_{i,t}$ kovarians dari Gaussian ke- i . Model ini menangkap variasi latar belakang, dan piksel yang menyimpang dari model dominan dianggap sebagai objek bergerak dan memperbaruinya secara adaptif berdasarkan perubahan visual dalam urutan video menggunakan rumus berikut [7].

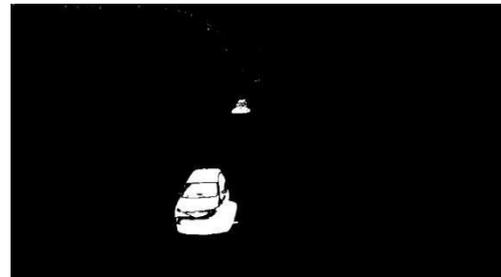
$$\begin{aligned} \mu_{i,t} &= (1 - \rho)\mu_{i,t-1} + \rho X_t \\ \sigma_{i,t}^2 &= (1 - \rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^2 \\ \omega_{i,t} &= (1 - \alpha)\omega_{i,t-1} + \alpha M_{i,t} \quad (2) \end{aligned}$$

Pada persamaan 2, α adalah *learning rate* dan M_i bernilai 1 jika Gaussian ke- i cocok dengan piksel X_t dan 0 jika tidak.

Piksel yang tidak sesuai dengan model dominan dianggap sebagai bagian dari objek bergerak, menghasilkan mask biner sebagai representasi area kendaraan.



Gambar 2a.



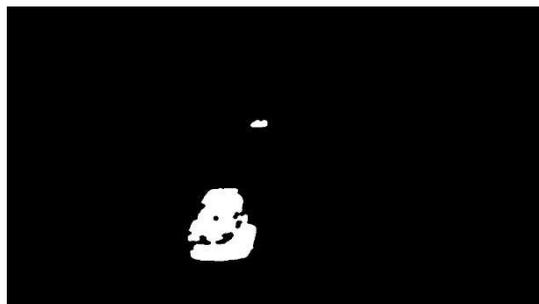
Gambar 2b.

Gambar 2. Perbandingan Hasil *Background Substraction*: (2a) Frame Asli (2b) Frame Hasil MOG2

Gambar 2 (a) merupakan *frame* asli dari video dan (b) menunjukkan hasil segmentasi *foreground* dalam bentuk mask biner yang diperoleh dari proses pemisahan latar belakang menggunakan algoritma MOG2. Dalam frame ini, piksel berwarna putih merepresentasikan area yang diklasifikasikan sebagai objek bergerak (*foreground*), sedangkan piksel berwarna hitam menunjukkan latar belakang.

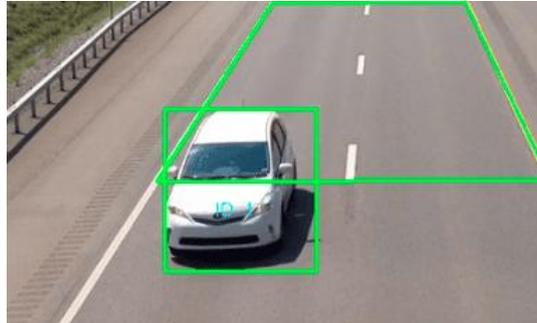
2.2 Filtering dan Contour Detection

Hasil *background subtraction* masih mengandung *noise* berupa area kecil yang tidak merepresentasikan objek. Untuk memperbaikinya, digunakan operasi morfologi, yaitu teknik pemrosesan citra biner yang memanipulasi struktur objek berdasarkan bentuk geometrisnya [20]. Dalam sistem ini, diterapkan *opening* untuk menghilangkan *noise* kecil dan *closing* untuk menutup celah pada objek [21]. Setelah itu, dilakukan deteksi kontur untuk menemukan area objek, dan hanya kontur dengan luas lebih dari 800 piksel yang dianggap valid dan diberikan *bounding box* untuk pelacakan.



Gambar 3. Hasil Operasi Morfologi dari Background Subtraction

2.3 Deteksi Object Berbasis Centroid



Gambar 4. Hasil Pelacakan Kendaraan Dengan Informasi ID

Gambar 4 memperlihatkan hasil deteksi dan pelacakan kendaraan. Setelah objek terdeteksi, sistem mengekstraksi titik pusat (*centroid*) dari setiap area tersegmentasi. Pelacakan antar *frame* dilakukan dengan mengasosiasikan *centroid* terdekat menggunakan perhitungan jarak *Euclidean* [22]:

$$D(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

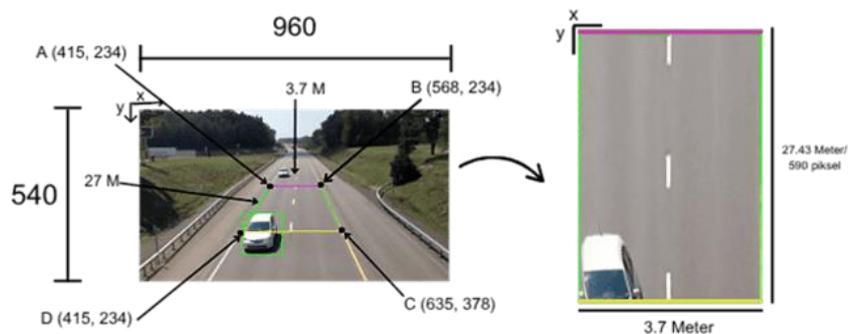
Di mana $D(p, q)$ adalah jarak antara *centroid* objek pada *frame* sebelumnya dan *centroid* pada *frame* saat ini. *Centroid* yang saling berdekatan dianggap berasal dari objek yang sama, dan diberi ID unik untuk pelacakan berkelanjutan.

2.4 Metode ORB (Oriented FAST and ROTATED BRIEF)

Untuk menjaga kestabilan dan akurasi posisi *Region of Interest* (ROI), sistem menerapkan pendekatan pencocokan fitur menggunakan ORB (*Oriented FAST and Rotated BRIEF*). ORB merupakan metode ekstraksi dan deskripsi fitur visual yang menggabungkan deteksi sudut FAST dengan deskripsi BRIEF yang telah dimodifikasi agar tahan terhadap rotasi dan skala [15].

Dalam setiap *frame*, titik-titik fitur yang terdeteksi pada ROI awal dibandingkan dengan fitur pada frame-frame berikutnya menggunakan pengukuran jarak *Hamming*. Transformasi spasial yang dihitung dari pasangan fitur yang cocok digunakan untuk memperbaiki posisi ROI secara dinamis. Dengan demikian, sistem mampu mempertahankan ROI yang konsisten meskipun terjadi sedikit perubahan posisi kamera atau kondisi visual.

2.5 Transformasi Perspektif



Gambar 5. Visualisasi ROI (Region of Interest) dan Transformasi Perspektif



Gambar 5 menunjukkan proses transformasi perspektif dari kamera ke tampilan *bird-eye view* untuk keperluan pengukuran spasial dan pelacakan kendaraan. *Region of Interest* (ROI) ditentukan dari empat titik koordinat pada citra asli, yaitu A(415,234), B(568,234), C(635,378), dan D(415,378), yang membentuk area jalan yang menjadi fokus analisis.

ROI ini kemudian dipetakan ke bidang datar menggunakan transformasi perspektif, menghasilkan tampilan *top-down* (*bird-eye*) [17] dengan ukuran sebenarnya, yaitu 3.7 meter untuk lebar jalur dan 27 meter untuk panjang area pantauan. Skala konversi piksel ke meter juga diperoleh, yaitu 143.7 piksel untuk 27 meter, yang digunakan untuk mengestimasi kecepatan kendaraan dari perubahan posisi antar frame. Transformasi ini penting untuk memastikan bahwa pergerakan objek dapat dianalisis dalam satuan metrik secara konsisten.

2.6 Estimasi Kecepatan Berdasarkan Jarak Top-Down

Kecepatan kendaraan dihitung berdasarkan waktu tempuh antara dua garis virtual (*entry and exit*) pada tampilan *top-down*. Jarak antara garis-garis tersebut dikalibrasi dalam satuan meter berdasarkan referensi visual seperti panjang marka jalan. Saat *centroid* kendaraan melintasi garis masuk dan kemudian garis keluar, waktu pada masing-masing peristiwa dicatat berdasarkan indeks *frame* dan dikonversi ke detik sesuai *frame rate*.

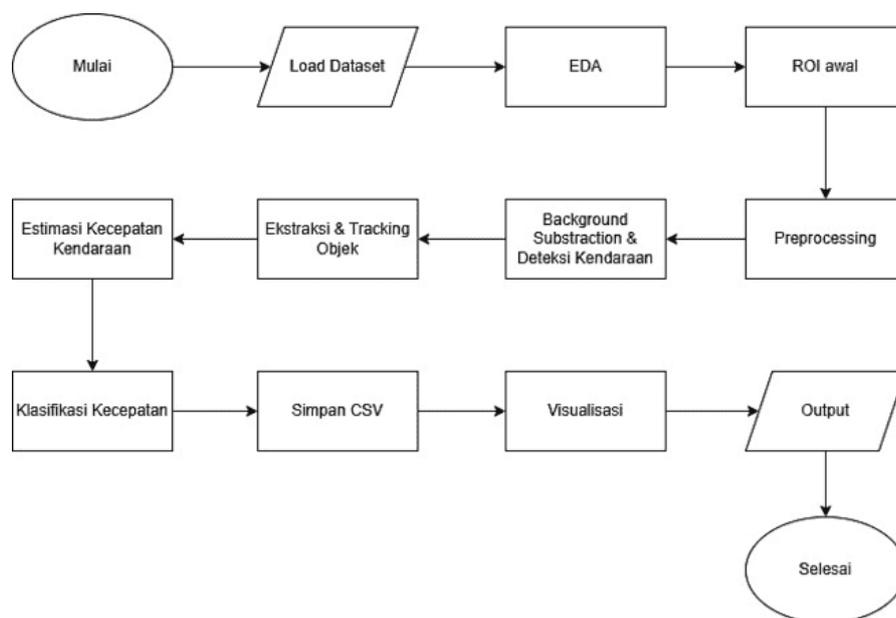
Kecepatan rata-rata v dari kendaraan dihitung menggunakan rumus dasar:

$$v = d/t$$

Dimana d adalah jarak antar ROI dalam meter, dan t adalah selisih waktu tempuh dalam detik. Proses ini menghasilkan estimasi kecepatan yang bersifat per-kendaraan dan dapat dimanfaatkan untuk analisis lalu lintas maupun sistem peringatan pelanggaran kecepatan.

3. HASIL DAN PEMBAHASAN

Penelitian dilakukan melalui langkah-langkah load dataset sampai dengan output analisis ditampilkan pada Gambar 6.

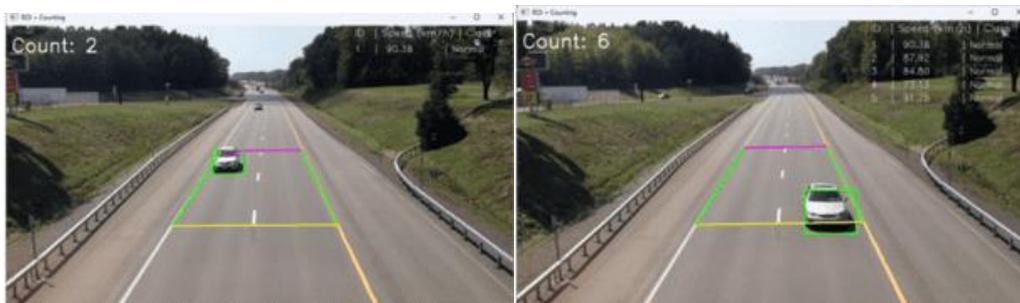


Gambar 6. Diagram Alir Pengerjaan Proyek

3.1. Hasil Deteksi dan Pelacakan Kendaraan

Sistem yang dikembangkan berhasil mendeteksi dan melacak kendaraan secara akurat menggunakan metode *background subtraction* (MOG2) yang dilengkapi dengan pemrosesan morfologi dan centroid tracking. ROI (*Region of Interest*) disesuaikan secara dinamis menggunakan transformasi homografi berbasis pencocokan fitur ORB agar tetap presisi terhadap perubahan perspektif selama video berlangsung. Pada pengujian dengan video uji, terdeteksi enam kendaraan yang melaju dari arah jauh menuju kamera, yang berhasil dilacak dan dihitung kecepatannya dengan tepat.

Visualisasi hasil deteksi pada salah satu *frame* dapat dilihat pada Gambar 7, di mana bounding box hijau menunjukkan kendaraan yang terdeteksi, sedangkan dua garis berwarna ungu dan kuning menunjukkan posisi garis *entry* dan *exit* secara dinamis berdasarkan ROI.



Gambar 7a.

Gambar 7b.

Gambar 7. Visualisasi Hasil Deteksi, Pelacakan, dan Counting Kendaraan Pada Beberapa Frame

3.2. Estimasi Kecepatan Kendaraan

Kecepatan kendaraan dihitung berdasarkan waktu yang dibutuhkan objek untuk melewati area pengamatan antara garis *entry* dan *exit*. Panjang lintasan dalam dunia nyata diasumsikan sejauh 27,33 meter, dan diperoleh dari kalibrasi terhadap ROI melalui transformasi perspektif ke bird's-eye view.

Untuk memperoleh estimasi jarak nyata antara ROI *Entry* dan ROI *Exit*, digunakan acuan visual berupa marka jalan putih putus-putus yang terlihat jelas dalam video. Berdasarkan dokumen *Manual on Uniform Traffic Control Devices (MUTCD)* dan referensi dari *South Carolina Department of Transportation (SCDOT)*, diketahui bahwa satu segmen garis marka memiliki panjang 10 kaki ($\pm 3,05$ meter), dan jarak antar garis (*gap*) adalah 30 kaki (9 meter) [23]. Dengan mengamati tiga garis dan dua celah di antaranya dalam area ROI, diperoleh total jarak sekitar 90 kaki atau 27.43 meter yang digunakan sebagai nilai jarak sebenarnya dalam perhitungan kecepatan.

Tabel 1. Estimasi kecepatan kendaraan berdasarkan waktu lintasan dan jarak hasil kalibrasi.

ID Kendaraan	Δ Waktu (s)	Kecepatan (km/jam)
1	0.73	90.38
2	0.77	87.92
3	0.80	84.80



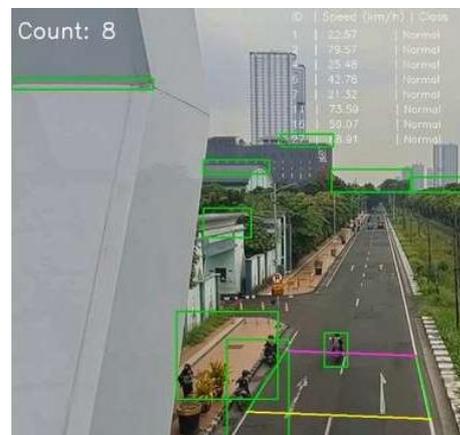
4	0.97	73.13
5	0.73	91.75
8	0.77	88.92

Berdasarkan data tersebut, seluruh kendaraan diklasifikasikan dalam kategori normal, karena berada di bawah ambang batas kecepatan yang ditetapkan (100 km/jam).

3.3. Hasil Uji Validasi Estimasi Kecepatan Kendaraan



Gambar 8a.



Gambar 8b.

Gambar 8. Hasil uji Validasi Kecepatan Estimasi

Tabel berikut merangkum estimasi kecepatan kendaraan berdasarkan waktu lintasan:

Tabel 2. Grafik Perbedaan Data Kecepataan Aktual Dengan Data Kecepatan Estimasi.

Kendaraan	Kecepatan Asli	Kecepatan Estimasi	Selisih Kecepatan
Motor 1	20	22.57	2.57
Motor 2	70	68.9	1.1

MAE (*Mean Absolute Error*)

$$MAE = \frac{|22.57 - 20| + |68.9 - 70|}{2} = \frac{2.57 + 1.1}{2} = 1.835$$

MAPE (*Mean Absolute Percentage Error*)

$$MAPE = \frac{1}{2} \left(\frac{2.57}{20} + \frac{1.1}{70} \right) \times 100 = \frac{1}{2} (12.85 + 1.57) = 7.21\%$$



Akurasi

$$Akurasi = 100\% - MAPE = 92.79\%$$

Hasil uji validasi estimasi kecepatan menunjukkan bahwa penerapan transformasi perspektif ke *bird eye's view* secara signifikan berhasil menampilkan estimasi kecepatan yang mendekati kecepatan sebenarnya. Transformasi ini membantu merepresentasi posisi dan perpindahan kendaraan secara proporsional terhadap dimensi ruang nyata, sehingga perhitungan estimasi kecepatan berdasarkan perubahan posisi antar frame menjadi lebih sesuai dengan kondisi riil. Temuan ini mendukung efektivitas pendekatan *bird eye's view* dalam sistem pemantauan lalu lintas berbasis visi komputer.

4. KESIMPULAN

Penelitian ini mengusulkan pendekatan berbasis visi komputer dengan menggabungkan algoritma *Mixture of Gaussians* (MoG) dan ORB (*Oriented FAST and Rotated BRIEF*) untuk deteksi dan pelacakan objek kendaraan. MoG digunakan untuk memisahkan objek bergerak dari latar belakang (*background subtraction*), sedangkan ORB berfungsi untuk mendeteksi sudut (*corner detection*) dan melacak fitur objek antar *frame* video. Kombinasi kedua metode ini memungkinkan sistem mendeteksi pergerakan objek secara stabil dan efisien, serta mempertahankan akurasi dalam kondisi lalu lintas yang dinamis. Deteksi yang akurat ini menjadi dasar untuk menghitung jumlah kendaraan dan mengestimasi kecepatan kendaraan menggunakan transformasi perspektif ke sudut pandang atas (*bird's eye view*), sehingga ukuran dalam piksel dapat dikonversi ke satuan meter. Hasil validasi menunjukkan bahwa sistem mampu memberikan estimasi kecepatan yang mendekati nilai sebenarnya. Untuk objek Motor 1, akurasi estimasi kecepatan mencapai 88,6%, sementara untuk Motor 2, akurasinya lebih tinggi, yaitu 98,4%. Nilai MAE yang diperoleh sebesar 1.835 km/jam, dan MAPE sebesar 7.21%, menghasilkan tingkat akurasi rata-rata estimasi kecepatan sebesar 92.79%. Dengan demikian, metode ini terbukti efektif dan menjanjikan untuk diimplementasikan dalam sistem pemantauan lalu lintas *real-time*.

REFERENSI

- [1] A. S. Kusuma, A. I. Pradana, and B. W. Pamekas, “Pengembangan Sistem Perhitungan Jumlah Kendaraan Berdasarkan Jenis Kendaraan Menggunakan Algoritma YOLO Secara Realtime,” *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 7, no. 2, pp. 166–179, 2024.
- [2] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, “Object Tracking Menggunakan Algoritma You Only Look Once (YOLO) v8 untuk Menghitung Kendaraan,” *Komputa: Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, pp. 91–99, 2023.
- [3] A. R. Jannah, “SISTEM PENGHITUNG PENUMPANG KENDARAAN UMUM MENGGUNAKAN KAMERA BERBASIS SINGLE SHOT DETECTOR (SSD),” Universitas Islam Sultan Agung Semarang, 2024.
- [4] I. Dienurrahman, “Rancang bangun dan analisis sistem penghitung kendaraan berbasis deep learning dengan arsitektur single shot multibox detector (SSD)= Development and analysis of deep learning based vehicle counter with single shot multibox detector architecture (SSD),” 2018.
- [5] D. Haryanto and others, “Sistem Penghitungan Kendaraan Otomatis Menggunakan Metode Object Detection Algoritma CNN,” *Buffer Informatika*, vol. 10, no. 2, pp. 11–18, 2024.
- [6] M. Othmani, “A vehicle detection and tracking method for traffic video based on faster R-CNN,” *Multimed Tools Appl*, vol. 81, no. 20, pp. 28347–28365, 2022.



- [7] K. A. M. Salih, I. A. Ali, and R. J. Mstafa, “Impact of Video Motion Content on HEVC Coding Efficiency,” *Computers*, vol. 13, no. 8, p. 204, 2024.
- [8] B. Setiyono, D. R. Sulistyaningrum, H. Alhabib, and others, “Improvement of sub region matching illumination transfer in hybrid shadow removal method for moving vehicle video,” *International Journal of Engineering & Technology*, vol. 7, no. 4, pp. 2515–2520, 2018.
- [9] D. Y. Sheth *et al.*, “Unsupervised deep video denoising,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1759–1768.
- [10] A. Pascale, E. Macedo, C. Guarnaccia, and M. C. Coelho, “Smart mobility procedure for road traffic noise dynamic estimation by video analysis,” *Applied Acoustics*, vol. 208, p. 109381, 2023.
- [11] A. Ayoub, W. El-Shafai, M. Aouf, F. E. Abd El-Samie, E. K. I. Hamad, and S. El-Rabaie, “Enhanced DehazeFormer: an image and video haze removal method based on SwinTransformer for superior quality,” *Multimed Tools Appl*, pp. 1–33, 2025.
- [12] G. Vaksman, M. Elad, and P. Milanfar, “Patch craft: Video denoising by deep modeling and patch matching,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2157–2166.
- [13] G. Ramesh, J. Logeshwaran, J. Gowri, and A. Mathew, “THE MANAGEMENT AND REDUCTION OF DIGITAL NOISE IN VIDEO IMAGE PROCESSING BY USING TRANSMISSION BASED NOISE ELIMINATION SCHEME.,” *ICTACT Journal on Image & Video Processing*, vol. 13, no. 1, 2022.
- [14] M. Neuhausen, J. Teizer, and M. König, “Construction worker detection and tracking in bird’s-eye view camera images,” in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2018, pp. 1–8.
- [15] C. Hong, M. Zhong, Z. Jia, C. You, and Z. Wang, “A stereo vision SLAM with moving vehicles tracking in outdoor environment,” *Mach Vis Appl*, vol. 35, no. 1, p. 5, 2024.
- [16] M. Z. Payudan, “TA: Sistem Deteksi Pelanggaran Jarak Social Distancing dengan Transformasi Bird’s Eye View Menggunakan Yolo-V3,” Universitas Dinamika, 2022.
- [17] M. Neuhausen, J. Teizer, and M. König, “Construction worker detection and tracking in bird’s-eye view camera images,” in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2018, pp. 1–8.
- [18] L. Rahmawati and K. Adi, “Rancang bangun penghitung dan pengidentifikasi kendaraan menggunakan Multiple Object Tracking,” *Youngster Physics Journal*, vol. 6, no. 1, pp. 70–75, 2017.
- [19] A. F. Syalsabilla and A. F. Rozy, “Klasifikasi Tingkat Sengketa Pemilu 2024 di Indonesia Menggunakan Metode Gaussian Mixture Model,” *Prosiding PITNAS Widyaiswara*, vol. 1, pp. 404–416, 2024.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 2018. [Online]. Available: <https://books.google.co.id/books?id=0F05vgAACAAJ>
- [21] R. Szeliski, *Computer Vision: Algorithms and Applications*. in Texts in Computer Science. Springer International Publishing, 2022. [Online]. Available: <https://books.google.co.id/books?id=QptXEAAQBAJ>
- [22] A. Salim, “Estimasi kecepatan kendaraan melalui video pengawas lalu lintas menggunakan parallel line model,” Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah~, 2020.



SENADA

Seminar Nasional Sains Data 2025 (SENADA 2025)
UPN “Veteran” Jawa Timur

E-ISSN 2808-5841
P-ISSN 2808-7283

- [23] U. States. N. A. C. on Uniform Traffic Control Devices, *Manual on uniform traffic control devices for streets and highways*. Department of Transportation, Federal Highway Administration, 1978.