

Implementasi *Haar Cascade* Untuk Deteksi Kendaraan Bermotor Pada Pemantauan Lalu Lintas Kota Malang

Rian Setya Budi^{1*}, Deddy Rudhistiar¹, Nurlaily Vandyansyah¹

¹Program Studi S1 Teknik Informatika, Institut Teknologi Nasional Malang, Indonesia

*Email: riansbudi19@gmail.com

Info Artikel	Abstrak
Kata Kunci : <i>haar cascade</i>, deteksi kendaraan, evaluasi kinerja, CCTV lalu lintas, kota malang Keywords : <i>haar cascade</i>, <i>vehicle detection</i>, <i>performance evaluation</i>, <i>traffic CCTV</i>, <i>malang city</i> Tanggal Artikel Dikirim : 27 November 2024 Direvisi : 6 Januari 2025 Diterima : 10 Februari 2025	Pemantauan lalu lintas di kota besar seperti Kota Malang semakin penting untuk mendukung ketertiban dan analisis jalan raya. Meskipun infrastruktur CCTV telah tersedia, teknologi deteksi kendaraan secara otomatis masih terbatas. Penelitian ini mengimplementasikan metode <i>Haar Cascade</i> untuk deteksi kendaraan bermotor pada rekaman CCTV guna meningkatkan akurasi dan efektivitas pemantauan lalu lintas. Evaluasi kinerja sistem dilakukan menggunakan metrik akurasi, presisi, <i>recall</i>, <i>F1 score</i>, serta pengujian <i>blackbox</i> untuk mengukur keandalan fungsi sistem. Hasil menunjukkan performa deteksi terbaik pada pagi hari, dengan akurasi 94.84%, presisi 84.21%, <i>recall</i> 100%, dan <i>F1 score</i> 91.43% untuk mobil, serta akurasi 93.81%, presisi 92.45%, <i>recall</i> 94.23%, dan <i>F1 score</i> 93.33% untuk motor. Namun, performa menurun pada malam hari, dengan akurasi 88.89% dan <i>F1 score</i> 71.43% untuk mobil, serta akurasi 46.15% dan <i>F1 score</i> 63.64% untuk motor. Pengujian <i>blackbox</i> menunjukkan bahwa setiap fungsi sistem berjalan sesuai dengan yang diharapkan tanpa ditemukan <i>error</i> dalam pengoperasian fitur utama. Penelitian ini menunjukkan bahwa metode <i>Haar Cascade</i> efektif untuk deteksi kendaraan di siang hari, namun membutuhkan peningkatan untuk kondisi pencahayaan rendah. Abstract <i>Traffic monitoring in major cities like Malang has become increasingly important to support order and road analysis. Although CCTV infrastructure is available, the technology for automatic vehicle detection remains limited. This study implements the Haar Cascade method for motor vehicle detection on CCTV recordings to improve the accuracy and effectiveness of traffic monitoring. The system's performance evaluation was conducted using metrics such as accuracy, precision, recall, F1 score, and blackbox testing to assess system reliability. The results show the best detection performance in the morning, with an accuracy of 94.84%, precision of 84.21%, recall of 100%, and an F1 score of 91.43% for cars, and an accuracy of 93.81%, precision of 92.45%, recall of 94.23%, and an F1 score of 93.33% for motorcycles. However, performance decreased at night, with an accuracy of 88.89% and an F1 score of 71.43% for cars, and an accuracy of 46.15% and an F1 score of 63.64% for motorcycles. Blackbox testing shows that all system functions operate as expected without any errors in the main features. This study demonstrates that the Haar Cascade method is effective for vehicle detection during the day but requires improvement for low-light conditions.</i>

1. PENDAHULUAN

Perkembangan teknologi informasi dan pengawasan berbasis video di kota-kota besar semakin pesat, salah satunya adalah penerapan *Closed-Circuit Television* (CCTV) untuk pemantauan lalu lintas. Di Kota Malang, penggunaan CCTV telah menjadi bagian penting dalam mendukung ketertiban dan keamanan di jalan raya. Meskipun infrastruktur CCTV telah memadai, pemanfaatan teknologi ini untuk deteksi kendaraan bermotor secara otomatis masih terbatas [1]. Sistem pemantauan lalu lintas yang mampu mendeteksi dan menghitung kendaraan secara akurat sangat penting untuk mendukung berbagai aplikasi, seperti manajemen lalu lintas, analisis kepadatan jalan, dan perencanaan infrastruktur transportasi. Dengan meningkatnya volume kendaraan di kota-kota besar, sistem deteksi otomatis menjadi solusi yang diperlukan untuk meningkatkan efisiensi dan efektivitas pemantauan lalu lintas [2].

Metode *Haar Cascade Classifier*, yang telah dikenal luas sebagai teknik deteksi objek berbasis visi komputer, menawarkan potensi untuk diterapkan dalam deteksi kendaraan bermotor. Metode ini memanfaatkan algoritma berbasis fitur untuk mendeteksi objek dengan kecepatan dan akurasi yang baik [3]. Dibandingkan dengan metode seperti YOLO (*You Only Look Once*) atau CNN (*Convolutional Neural Network*), *Haar Cascade* memiliki keunggulan dalam hal efisiensi komputasi, terutama untuk perangkat dengan sumber daya terbatas. Selain itu, *Haar Cascade* lebih sederhana untuk diimplementasikan dan memerlukan *dataset* pelatihan yang lebih kecil dibandingkan YOLO atau CNN, yang sering membutuhkan komputasi tinggi dan *dataset* besar untuk mencapai performa optimal. Namun, *Haar Cascade* memiliki keterbatasan pada kondisi pencahayaan yang rendah, di mana metode berbasis Deep Learning, seperti YOLO atau CNN, biasanya menunjukkan performa yang lebih baik berkat kemampuan mereka dalam menangkap fitur yang lebih kompleks [4]. Penelitian ini bertujuan untuk mengimplementasikan *Haar Cascade* dalam mendeteksi kendaraan bermotor melalui rekaman CCTV di Kota Malang. Untuk memaksimalkan utilitasnya, sistem ini diintegrasikan ke dalam aplikasi berbasis web menggunakan *framework* Flask. Dengan solusi ini, deteksi kendaraan dapat dilakukan secara otomatis dan memberikan data *real-time* mengenai *volume* serta jenis kendaraan bermotor yang melintas di jalan raya [5]. Evaluasi kinerja dilakukan untuk memastikan keandalan sistem dalam berbagai kondisi, seperti siang hari, hujan, dan malam. Analisis dilakukan dengan menggunakan metrik seperti akurasi, presisi, dan *recall*, untuk menilai efektivitas sistem deteksi kendaraan [6].

Beberapa studi telah menunjukkan tingkat keberhasilan yang signifikan dalam pengembangan sistem deteksi kendaraan bermotor. Hardiansyah (2020) mengembangkan sistem untuk mendeteksi dan menghitung kendaraan secara *real-time* menggunakan OpenCV dan Python. Sistem ini berhasil mengklasifikasikan kendaraan ke dalam kategori kendaraan ringan, berat, dan sepeda motor dengan menggunakan teknik pengurangan latar belakang, penyaringan gambar, dan segmentasi. Namun, penelitian ini tidak secara khusus mengevaluasi metode *Haar Cascade* untuk deteksi kendaraan [7]. Mohamed Elhelw (2018) memanfaatkan fitur *Haar-like* dan *Compressive Tracking* untuk sistem deteksi dan pelacakan kendaraan secara *real-time*. Penelitian ini menunjukkan akurasi tinggi hingga 93% dengan waktu pemrosesan rata-rata 0,2017 detik per *frame*, namun kinerjanya sangat bergantung pada kualitas pengklasifikasi *AdaBoost cascade* dan model berbasis penampilan adaptif [8]. Penelitian Ramadhani et al. (2017) fokus pada penggunaan metode *Haar Cascade Classifier* untuk deteksi kendaraan pada sistem pengawasan lalu lintas di Jakarta. Dengan mengimplementasikan algoritma *AdaBoost* dan fitur *Haar-like*, penelitian ini mencapai tingkat akurasi tinggi dengan rata-rata *recall* 0,986 dan presisi 0,978 setelah diuji pada beberapa video lalu lintas [3]. Namun, sebagian besar penelitian tersebut belum secara mendalam mengevaluasi performa metode pada kondisi pencahayaan rendah, seperti malam hari atau dalam kondisi cuaca buruk. Keterbatasan ini menjadi tantangan utama, karena metode berbasis fitur seperti *Haar Cascade* cenderung mengalami penurunan akurasi dalam mendeteksi objek di lingkungan dengan pencahayaan yang tidak memadai. Penelitian ini berupaya mengisi gap tersebut dengan mengevaluasi performa metode *Haar Cascade* pada berbagai kondisi lingkungan, termasuk pencahayaan rendah, untuk memberikan gambaran yang lebih komprehensif tentang kemampuan dan keterbatasan metode ini.

Meskipun penelitian-penelitian ini menunjukkan potensi besar dalam implementasi sistem deteksi kendaraan, ada celah yang dapat diisi dengan menyesuaikan teknologi untuk kondisi lokal di Kota Malang. Faktor seperti kondisi pencahayaan, sudut pandang kamera, dan jenis kendaraan yang lebih beragam membutuhkan penelitian lebih lanjut untuk mengoptimalkan metode *Haar Cascade* dalam mendeteksi kendaraan bermotor pada rekaman CCTV lokal [2], [7]. Penelitian ini bertujuan untuk memberikan kontribusi baru dengan mengimplementasikan sistem berbasis *Haar Cascade* untuk mendeteksi kendaraan di Kota Malang. Sistem ini diharapkan mampu menangani berbagai kondisi lalu lintas dan memberikan data akurat yang dapat mendukung manajemen lalu lintas secara efektif.

2. METODE PENELITIAN

2.1 Haar Cascade Classifier

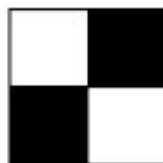
Haar Cascade Classifier adalah metode berbasis *machine learning* yang digunakan untuk mendeteksi objek dalam gambar atau video. Metode ini dikembangkan oleh Paul Viola dan Michael Jones pada tahun 2001, dan telah diterapkan secara luas dalam berbagai aplikasi pengenalan objek, seperti wajah, mata, dan kendaraan. Metode ini bekerja dengan memanfaatkan fitur *Haar*, yang merupakan fitur persegi panjang untuk mendeteksi tepi, garis, atau area intensitas yang berbeda pada gambar. Metode ini mempunyai 4 tahapan dalam mendeteksi sebuah objek, seperti *Haar-like Features*, *integral images*, *Adaboost learning*, dan klasifikasi *Cascade* [9].



Gambar 1. *Edge Features* [10]



Gambar 2. *Line Features* [10]



Gambar 3. *Four-rectangle Features* [10]



Gambar 4. Pendeteksian Kendaraan dengan *Haar-like Features*

Fitur persegi panjang yang memberikan indikasi tertentu pada gambar disebut *Haar-like* Fitur. Contoh variasi *Haar-like* Fitur yang umum ditunjukkan pada Gambar 1, 2, dan 3. Bergantung pada jumlah piksel di dalam fitur persegi panjang dan bukan nilai setiap piksel dalam gambar, *Haar-like* Fitur memberikan kecepatan tinggi komputasi [8]. Dengan menggunakan citra integral, nilai *Haar-like* Fitur dihitung untuk menentukan nilai deteksi objek. Seperti terlihat pada Gambar 4, gambar integral dapat menghitung nilai dengan akurat dan cepat dengan menghasilkan presentasi gambar baru dengan memanfaatkan nilai wilayah yang sebelumnya telah dipindai oleh *Haar-like* Fitur tertentu. Nilai gambar integral ditentukan dengan menjumlahkan nilai indeks sebelumnya, mulai dari kiri atas dan terus ke kanan bawah. Selain itu, gambar integral dapat dihitung menggunakan Persamaan 1 [9], seperti yang ditunjukkan pada gambar masukan nilai integral dan hasil yang dihitung dari Persamaan 1 dibawah ini.

$$s(x, y) = i(x, y) + s(x, y - 1) + s(x - 1, y1) + s(x - 1, y - 1) \quad (1)$$

Keterangan:

$s(x, y)$ = Integral Image pada koordinat (x, y) .

$i(x, y)$ = Intensitas Piksel pada koordinat (x, y) dalam citra *grayscale*.

	1	2	3	4
1	3	7	7	3
2	1	3	3	1
3	5	9	9	5
4	3	6	6	3

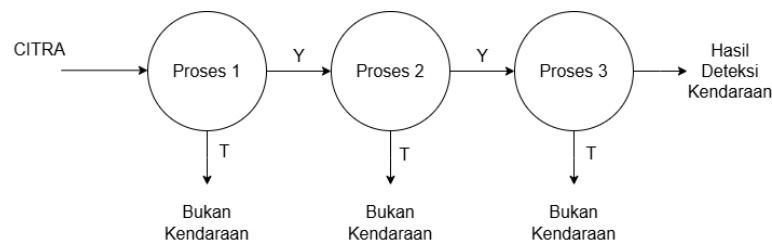
a

	1	2	3	4
1	3	10	17	20
2	4	14	24	28
3	9	28	47	56
4	12	37	62	74

b

Gambar 5. (a) Citra Inputan dan (b) Hasil Inputan Citra Integral [3]

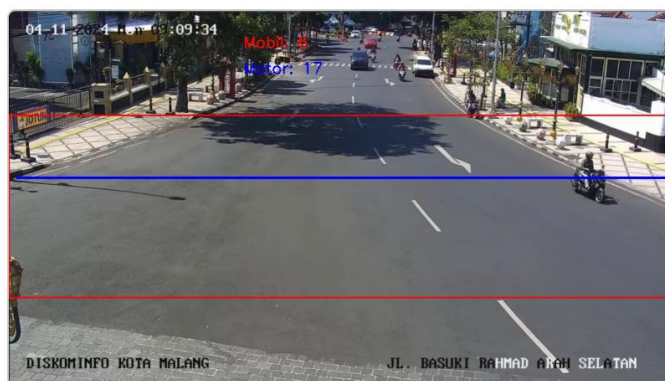
Nilai yang telah dihitung menggunakan *integral image* kemudian akan dibandingkan dengan nilai ambang dari fitur spesifik yang disediakan oleh *AdaBoost*. Ini harus diselesaikan untuk menemukan fitur potensial karena tidak semua fitur relevan untuk digunakan dalam deteksi objek tertentu [9].



Gambar 6. Alur Klasifikasi Haar [8]

Klasifikasi kuat yang dibuat oleh *AdaBoost* dapat mendeteksi objek secara bertahap pada sebuah *cascade*. Setiap sub-jendela dipindai untuk kriteria spesifik pada setiap langkah; lebih lanjut, sub-jendela yang mengandung objek positif digunakan sebagai umpan untuk penyaringan tingkat berikutnya dengan kriteria yang lebih spesifik hingga memperoleh sub-jendela yang diprediksi sebagai mobil. Di sisi lain, *sub-jendela* yang tidak mengandung objek positif ditandai sebagai latar belakang dan dipisahkan oleh *sub-jendela* yang mengandung objek positif [3].

2.2 Region of Interest



Gambar 7. Region Of Interest

Region of Interest adalah area tertentu dari gambar yang dianggap penting untuk mengumpulkan data untuk penelitian ini [11]. Karena proses pengumpulan data difokuskan pada wilayah gambar tertentu untuk deteksi objek, pelacakan objek, dan penghitungan objek, seperti yang ditunjukkan pada Gambar 7, *Haar Cascade Classifier* tidak perlu memindai seluruh wilayah gambar [3].

2.3 Dataset

Dataset yang digunakan dalam penelitian ini berasal dari rekaman video CCTV lalu lintas di Kota Malang, yang diperoleh melalui situs resmi CCTV publik (<https://cctvmalang.24hour.id>). Video diambil dari lokasi strategis di Jl. Jenderal Basuki Rahmat, Kota Malang, yang merupakan area dengan lalu lintas padat. *Dataset* terdiri dari tiga video dengan total durasi 6 menit, masing-masing berdurasi dua menit. Video mencakup tiga kondisi utama: pagi hari dengan pencahayaan optimal, malam hari dengan pencahayaan rendah, dan hujan yang mempengaruhi visibilitas objek. Teknik validasi data dilakukan dengan membagi video ke dalam skenario pengujian berdasarkan kondisi lingkungan untuk mengevaluasi performa sistem pada setiap kondisi tersebut. Setiap video dianalisis secara manual untuk menghitung jumlah kendaraan aktual sebagai acuan (*ground truth*), sehingga hasil deteksi dapat dibandingkan menggunakan metrik akurasi, presisi, *recall*, dan *F1 score*. Pendekatan ini memastikan bahwa evaluasi sistem mencerminkan kemampuan deteksi pada situasi dunia nyata dengan variasi kondisi pencahayaan dan cuaca.

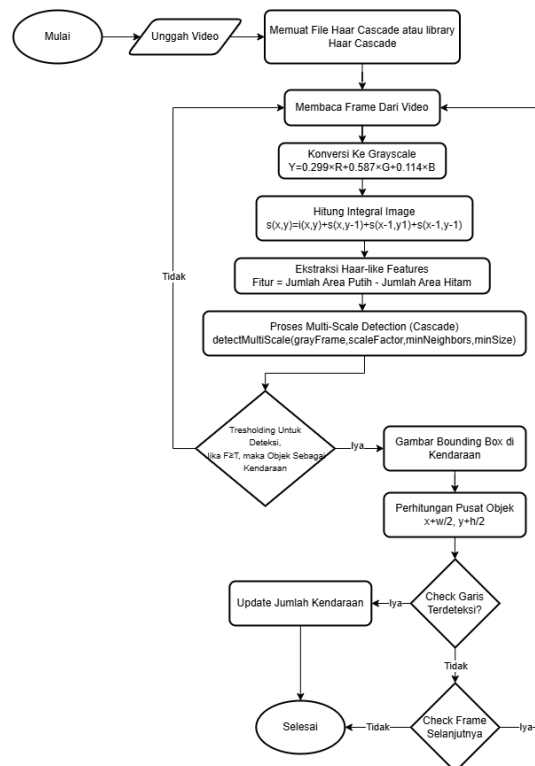
Setiap *frame* video diproses dengan mengonversinya ke skala abu-abu untuk menyederhanakan pemrosesan dan mengurangi *noise* [12]. Pemilihan *dataset* ini bertujuan untuk mencerminkan berbagai kondisi lingkungan yang umum terjadi di Kota Malang.

2.3 Implementasi Sistem

Aplikasi deteksi kendaraan dikembangkan menggunakan *framework Flask*, yang memungkinkan integrasi dengan antarmuka web. Implementasi mencakup langkah-langkah berikut:

1. *Preprocessing Data*: Video diubah menjadi *frame-frame* individual, di mana setiap *frame* dikonversi ke format skala abu-abu dan disesuaikan ukurannya [13].
2. Deteksi Objek: Fitur *Haar Cascade* digunakan untuk mendeteksi kendaraan roda dua dan roda empat.
3. Penghitungan Kendaraan: Sistem menghitung jumlah kendaraan yang melewati area pada *Region of Interest* dalam video, menggunakan *bounding box* sebagai penanda objek terdeteksi.
4. Visualisasi Hasil: *Output* ditampilkan melalui halaman web, di mana pengguna dapat melihat hasil deteksi kendaraan secara *real-time* atau berdasarkan video yang diunggah.

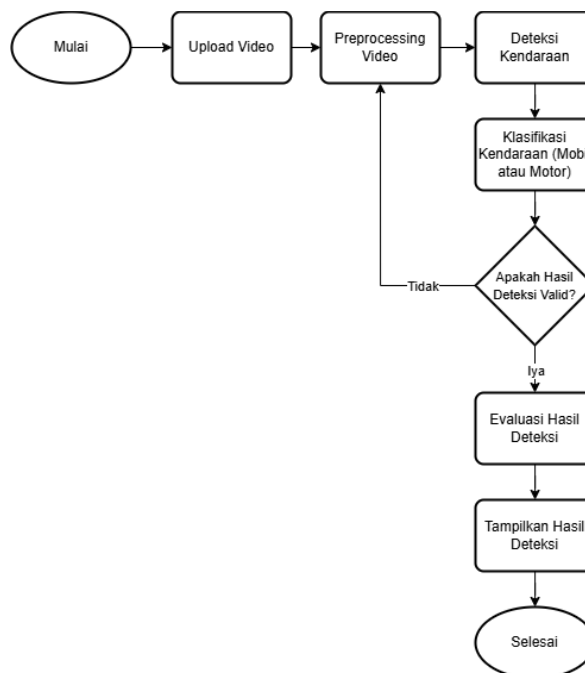
2.3.1 Flowchart Metode Haar Cascade



Gambar 8. Flowchart Metode Haar Cascade

Pada Gambar 8 menunjukkan alur deteksi objek menggunakan *Haar Cascade Classifier*. Proses dimulai dengan unggah video sebagai input, diikuti inisialisasi *cascade classifier* dan konversi *frame* ke skala abu-abu. Sistem menghitung *integral image*, ekstraksi fitur *Haar-like*, dan melakukan *multi-scale detection* untuk mendeteksi kendaraan dalam berbagai ukuran. Jika deteksi memenuhi ambang batas, *bounding box* digambar di sekitar kendaraan, dan posisi pusat objek dihitung [14]. Kendaraan yang melewati garis deteksi akan dihitung, dan jumlahnya diperbarui hingga semua frame selesai diproses, menandai akhir analisis.

2.3.2 Flowchart Sistem

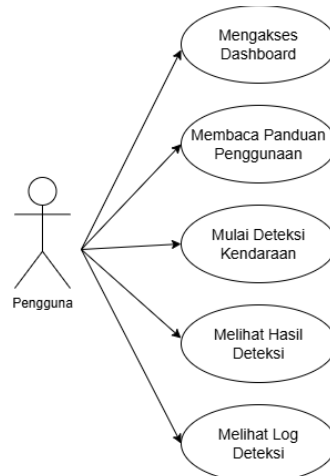


Gambar 9. Flowchart Sistem

Pada gambar 9 menunjukkan *flowchart* alur algoritma deteksi kendaraan dengan metode *Haar Cascade*. Proses dimulai dengan unggah video, kemudian video tersebut kemudian melalui tahap *preprocessing*, di mana video dipecah menjadi *frame* individu, setiap *frame* dikonversi menjadi skala abu-abu (*grayscale*). Selain itu, proses *filtering* diterapkan untuk mengurangi *noise* yang dapat mengganggu deteksi. Setelah *preprocessing*, sistem melakukan deteksi kendaraan menggunakan metode *Haar Cascade*. Pada tahap ini, *bounding box* dihasilkan untuk menandai objek kendaraan yang terdeteksi dalam *frame*. Kendaraan yang telah terdeteksi kemudian diklasifikasikan berdasarkan jenisnya, yaitu mobil atau motor. Sistem selanjutnya mengevaluasi validitas hasil deteksi. Jika hasil deteksi tidak valid, sistem kembali ke tahap *preprocessing* untuk memperbaiki *frame*.

Jika hasil deteksi valid, evaluasi dilakukan dengan membandingkan hasil deteksi dengan data *ground truth*. Sistem menghitung metrik performa, termasuk *True Positive* (TP), *False Positive* (FP), dan *False Negative* (FN). Berdasarkan data ini, metrik seperti akurasi, presisi, *recall*, dan *F1 score* dihitung untuk mengukur kinerja sistem secara keseluruhan. Hasil deteksi dan evaluasi ditampilkan dalam bentuk laporan yang mencakup jumlah kendaraan berdasarkan jenis serta grafik atau tabel evaluasi performa. Proses berakhir setelah seluruh langkah selesai dijalankan, dan sistem kembali menunggu input video berikutnya atau berhenti jika tidak ada operasi lebih lanjut.

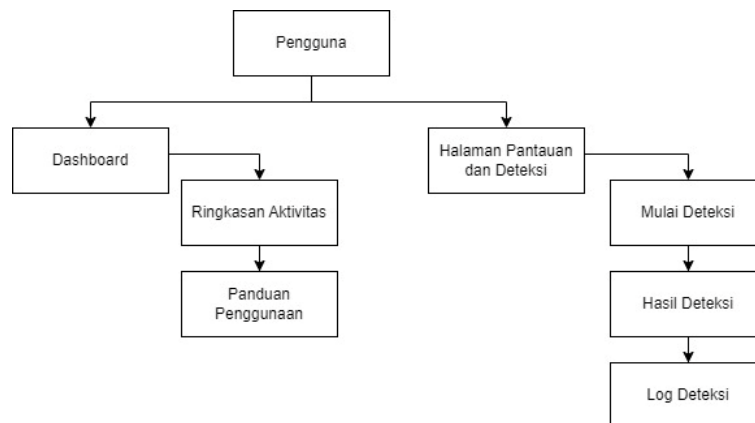
2.4 Use Case Diagram



Gambar 10. Use Case Diagram

Pada Gambar 10 *Use case diagram* ini menunjukkan lima fitur utama yang dapat diakses Pengguna Umum tanpa *login*: mengakses *dashboard*, membaca panduan, memulai deteksi, melihat hasil deteksi, dan log deteksi. Fitur ini memudahkan publik dalam memantau lalu lintas secara efektif dan transparan.

2.4 Struktur Menu



Gambar 11. Struktur Menu

Pada Gambar 11 menunjukkan struktur menu sistem yang meliputi *dashboard* untuk informasi sistem deteksi, menu Pantau dan Deteksi untuk mengunggah video lalu lintas, serta hasil deteksi yang menampilkan perhitungan dan laporan analisis lalu lintas.

2.5 Evaluasi Sistem

Evaluasi sistem dilakukan untuk mengukur kinerja metode *Haar Cascade* menggunakan empat metrik utama [6]:

1. Akurasi (*Accuracy*): Mengukur persentase deteksi yang benar dibandingkan total data.

$$\text{Akurasi} = \frac{\text{Jumlah Deteksi Benar}}{\text{Jumlah Total Kendaraan}} \times 100\% \quad (2)$$

2. Presisi (*Precision*): Mengukur proporsi deteksi positif yang benar.

$$\text{Presisi} = \frac{\text{Deteksi Benar}}{\text{Deteksi Benar} + \text{Deteksi Salah}} \quad (3)$$

3. *Recall (Sensitivity)*: Mengukur kemampuan sistem untuk mendeteksi semua objek yang benar.

$$Recall = \frac{Deteksi\ Benar}{Deteksi\ Benar + Deteksi\ yang\ Terlewat} \quad (4)$$

4. *F1 Score*: Menggabungkan presisi dan recall untuk memberikan nilai kinerja yang seimbang.

$$F1 = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (5)$$

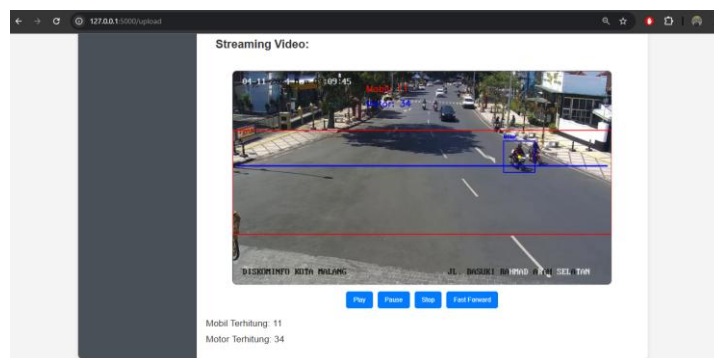
Setiap metrik dihitung berdasarkan hasil deteksi pada enam video uji dengan kondisi lingkungan yang berbeda. Proses evaluasi melibatkan validasi manual untuk memastikan keakuratan hasil [6].

3. HASIL DAN PEMBAHASAN

3.1 Pengujian dan Hasil Sistem

Sistem deteksi kendaraan menggunakan metode *Haar Cascade Classifier* diuji dengan input berupa video yang diperoleh dari CCTV di Kota Malang. Video diproses *frame per frame* untuk mendeteksi kendaraan bermotor roda dua dan roda empat. Proses deteksi melibatkan konversi citra ke skala abu-abu, ekstraksi fitur *Haar-like*, dan klasifikasi berdasarkan model yang telah dilatih sebelumnya.

Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi kendaraan dengan *bounding box*, seperti yang ditunjukkan pada Gambar 12. Deteksi berhasil dilakukan pada berbagai kondisi, seperti siang hari dengan cahaya optimal dan malam hari dengan pencahayaan terbatas.



Gambar 12. Hasil Deteksi Kendaraan pada Video Siang Hari

3.1.1 Data Uji

Pada penelitian ini, pengujian sistem deteksi kendaraan dilakukan menggunakan tiga video yang diambil dari rekaman CCTV di lokasi Jl. Basuki Rahmat, Kayutangan, Kota Malang. Ketiga video dipilih untuk merepresentasikan berbagai kondisi, yaitu pagi hari (cerah), saat hujan, dan malam hari. Hal ini bertujuan untuk mengevaluasi performa sistem dalam kondisi pencahayaan dan cuaca yang berbeda.

Data uji mencakup dua jenis kendaraan, yaitu roda dua (motor) dan roda empat (mobil). Perhitungan manual dilakukan untuk mendapatkan nilai aktual sebagai *ground truth* [15]. Data hasil perhitungan manual dirangkum dalam Tabel 1 dibawah ini:

Tabel 1. Rekapitulasi Data Perhitungan Manual

No	Kondisi Lingkungan	Jumlah Mobil	Jumlah Motor
1	Pagi	16	104
2	Hujan	15	45
3	Malam	9	21

3.2 Evaluasi Sistem

Evaluasi kinerja sistem dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *F1 Score*, yang dihitung berdasarkan hasil deteksi dari tiga video uji. Data uji melibatkan video dengan durasi rata-rata 2 menit, berisi lalu lintas padat hingga sedang.

Tabel 2. Hasil Evaluasi Sistem

No	Kondisi Lingkungan	Jenis Kendaraan	TP	FP	FN	Akurasi	Presisi	Recall	F1 Score
1	Pagi	Mobil	16	3	0	94.84%	84.21%	100%	91.43%
		Motor	98	8	6	93.81%	92.45%	94.23%	93.33%
2	Siang	Mobil	12	15	3	89.13%	70.59%	80%	74.81%
		Motor	44	0	1	97.83%	100%	97.78%	98.88%
3	Malam	Mobil	5	0	4	88.89%	100%	55.56%	71.43%
		Motor	21	24	0	46.15%	46.67%	100%	63.64%
4	Rata-rata		32.67	8.33	2.33	85.11%	82.32%	87.93%	82.25%

Catatan:

TP : True Positive atau Kendaraan Terdeteksi

FP : False Positive atau Kendaraan Tidak Terdeteksi

FN : False Negative atau Kendaraan Terdeteksi Terlewat

Hasil menunjukkan sistem mampu mencapai rata-rata akurasi 85.11% dengan *F1 Score* 82.25%, menunjukkan kinerja yang cukup baik dalam mendeteksi kendaraan.

3.3 Hasil Analisis

Hasil evaluasi menunjukkan bahwa performa sistem deteksi kendaraan menggunakan metode *Haar Cascade Classifier* bervariasi berdasarkan kondisi pencahayaan. Akurasi tertinggi dicapai pada pagi hari dengan nilai 94.84% untuk mobil dan 93.81% untuk motor. Namun, akurasi menurun secara signifikan pada malam hari, terutama untuk motor, dengan nilai hanya mencapai 46.15%. Penurunan ini disebabkan oleh tingginya tingkat *false positive* akibat pencahayaan yang tidak memadai, yang menyebabkan fitur Haar sulit mengenali pola kendaraan secara konsisten.

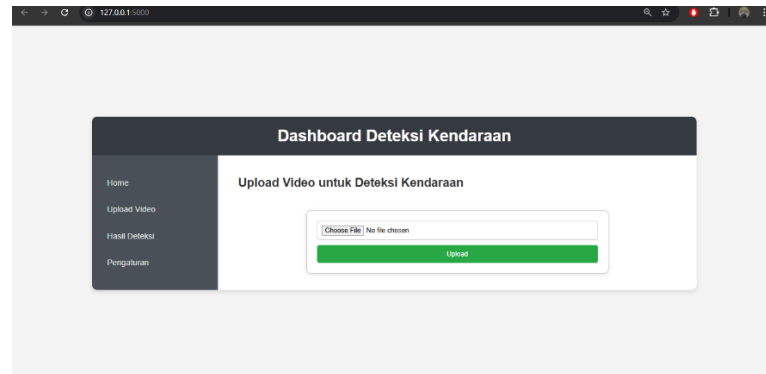
Presisi sistem menunjukkan tingkat keakuratan deteksi yang baik pada pagi hari, yaitu 84.21% untuk mobil dan 92.45% untuk motor. Sebaliknya, presisi motor pada malam hari turun drastis menjadi 46.67%, mengindikasikan banyaknya deteksi palsu. Faktor utama penyebabnya adalah *noise* pada data akibat bayangan, pantulan cahaya, dan objek tidak relevan yang terdeteksi sebagai kendaraan.

Recall sistem lebih tinggi untuk motor dibandingkan mobil dalam berbagai kondisi. Pada malam hari, *recall* motor mencapai 100%, tetapi mobil hanya sebesar 55.56%, yang menunjukkan bahwa sistem cenderung mendeteksi semua objek, meskipun tidak selalu akurat. Hal ini terjadi karena sistem memprioritaskan sensitivitas deteksi, tetapi kurang optimal dalam menyaring objek yang sebenarnya kendaraan.

F1 Score, yang mengukur keseimbangan antara presisi dan *recall*, tertinggi dicapai pada pagi hari untuk motor (93.33%). Namun, pada malam hari, *F1 Score* motor turun menjadi 63.64% karena tingginya ketidakseimbangan antara deteksi benar dan *false positive*. Secara keseluruhan, hasil ini mengindikasikan bahwa pencahayaan sangat memengaruhi performa deteksi kendaraan. Kurangnya *preprocessing* seperti penyesuaian kontras atau penghapusan *noise* menjadi salah satu penyebab utama penurunan performa sistem pada kondisi pencahayaan rendah.

3.4 Tampilan Web Sistem

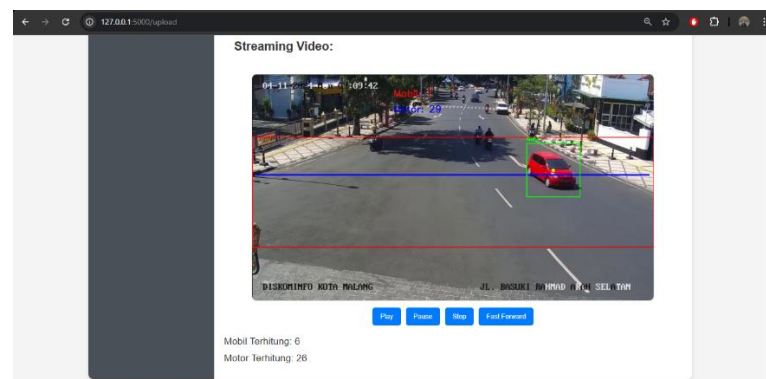
3.4.1 Tampilan *Dashboard*



Gambar 13. *Tampilan Dashboard*

Pada gambar 13 menampilkan halaman *dashboard* yang terdapat beberapa informasi tentang sistem *form* yang digunakan sebagai memasukkan video yang akan digunakan untuk deteksi kendaraan.

3.4.2 Tampilan Halaman *Upload*



Gambar 14. *Tampilan Halaman Upload*

Pada Gambar 14, ditunjukkan tampilan unggahan video pada sistem deteksi kendaraan. Pengguna dapat melihat proses deteksi dan perhitungan kendaraan yang sedang berjalan. Selain itu, terdapat beberapa tombol untuk mengontrol pemutaran video.

3.5 Pengujian *Blackbox*

Pengujian *blackbox* dilakukan untuk memastikan bahwa fungsi-fungsi utama dalam sistem berjalan sesuai dengan spesifikasi yang telah dirancang. Tabel di bawah ini merangkum hasil pengujian *blackbox* pada setiap fitur sistem.

Tabel 3. Hasil Pengujian *Blackbox*

No	Fitur yang Diuji	Deskripsi Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Upload Video	Pengguna mengunggah file video untuk proses deteksi	Video berhasil diunggah dan siap diproses	Berhasil	Lulus
2	Deteksi Kendaraan	Sistem mendeteksi kendaraan pada video	<i>Bounding box</i> ditampilkan pada kendaraan yang terdeteksi	Berhasil	Lulus
3	Perhitungan Kendaraan	Sistem menghitung jumlah kendaraan yang melewati garis	Jumlah kendaraan sesuai dengan deteksi <i>bounding box</i>	Berhasil	Lulus

No	Fitur yang Diuji	Deskripsi Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
4	Tampilkan Hasil Deteksi	Hasil deteksi berupa laporan jumlah kendaraan ditampilkan	Laporan hasil deteksi terlihat jelas di halaman	Berhasil	Lulus
5	Streaming Real-time	Sistem menampilkan deteksi secara <i>real-time</i>	Video <i>real-time</i> dengan <i>bounding box</i> kendaraan terlihat	Berhasil	Lulus

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan metode *Haar Cascade Classifier* untuk mendeteksi kendaraan bermotor pada video lalu lintas yang diambil dari CCTV Kota Malang. Hasil pengujian menunjukkan bahwa sistem deteksi kendaraan memiliki performa yang cukup baik, terutama pada kondisi pencahayaan optimal seperti pagi hari. Pada kondisi ini, akurasi mencapai 94.84% untuk mobil dan 93.81% untuk motor. Namun, performa sistem menurun secara signifikan pada malam hari, terutama untuk deteksi motor, yang dipengaruhi oleh tingginya jumlah *false positive*.

Analisis lebih lanjut mengungkapkan bahwa kondisi lingkungan sangat memengaruhi kinerja sistem. Pada saat hujan, *noise* akibat cipratan air atau kabut menyebabkan penurunan akurasi. Sementara itu, pada malam hari, pencahayaan yang kurang memadai mengakibatkan presisi deteksi kendaraan menurun drastis [16]. Pengujian *blackbox* menunjukkan bahwa fitur-fitur utama seperti unggah video, proses deteksi kendaraan, dan penampilan hasil deteksi berjalan sesuai dengan ekspektasi tanpa ada error.

Dari sisi efektivitas, metode *Haar Cascade* terbukti efisien dan mampu mendeteksi kendaraan dengan baik pada pencahayaan yang cukup, memberikan tingkat *recall* yang tinggi, terutama untuk kendaraan roda dua. Hal ini menunjukkan bahwa metode ini cocok untuk diterapkan dalam kondisi lalu lintas siang hari dengan pencahayaan optimal, menjadikannya solusi yang cepat dan hemat sumber daya untuk pemantauan lalu lintas.

Namun, metode ini memiliki keterbatasan dalam menghadapi kondisi dengan pencahayaan rendah atau lingkungan yang kompleks, di mana tingkat *false positive* cenderung meningkat. Untuk mengatasi hal ini, pengembangan lebih lanjut dapat mencakup pengintegrasian metode *Haar Cascade* dengan teknik berbasis *Deep Learning*, seperti *Convolutional Neural Networks* (CNN), untuk meningkatkan akurasi deteksi pada berbagai kondisi. Selain itu, eksplorasi *dataset* dari lokasi berbeda dengan karakteristik lingkungan yang beragam dapat membantu meningkatkan generalisasi sistem. Implementasi algoritma *preprocessing*, seperti *Histogram Equalization*, juga dapat diterapkan untuk meningkatkan kualitas citra sebelum proses deteksi. Perbaikan-perbaikan ini diharapkan dapat meningkatkan keandalan sistem dalam pemantauan lalu lintas tidak hanya di Kota Malang tetapi juga di lokasi lain dengan tantangan lingkungan yang serupa.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada Institut Teknologi Nasional Malang atas fasilitas dan sumber daya yang disediakan.

DAFTAR PUSTAKA

- [1] M. Harahap, J. Elfrida, P. Agusman, M. Rafael, R. Abram, and K. Andrianto, "Sistem Cerdas Pemantauan Arus Lalu Lintas Dengan YOLO (You Only Look Once v3)," *Semin. Nas. APTIKOM*, pp. 367–376, 2019.
- [2] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," *Eurasip J. Image Video Process.*, vol. 2018, no. 1, pp. 1–7, 2018, doi: 10.1186/s13640-018-0350-2.
- [3] M. I. Ramadhani, A. E. Minarno, and E. B. Cahyono, "Vehicle Classification using Haar Cascade Classifier Method in Traffic Surveillance System," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 3, no. 1, pp. 57–64, 2017, doi: 10.22219/kinetik.v3i1.546.
- [4] M. A. Kamil and Y. M. Djaksana, "JIIC: JURNAL INTELEK INSAN CENDIKIA PERBANDINGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) DAN ALGORITMA YOU ONLY LOOK ONCE (YOLO) UNTUK DETEKSI WAJAH COMPARISON OF CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM AND YOU ONLY LOOK ONCE (YOLO) ALG," no. November, 2024.
- [5] A. Morchid *et al.*, "IoT-enabled fire detection for sustainable agriculture: A real-time system using flask and embedded technologies," *Results Eng.*, vol. 23, no. July, p. 102705, 2024, doi: 10.1016/j.rineng.2024.102705.
- [6] R. Romindo, O. P. Barus, and J. J. Pangaribuan, "Implementasi Metode Naive Bayes Classifier Terhadap Klasifikasi Topik Kemacetan Lalu Lintas Indonesia Melalui Tweet," *J. Media Inform. Budidarma*, vol. 8, no. 2, p. 1087, 2024, doi: 10.30865/mib.v8i2.7470.

- [7] P. Studi and T. Informatika, "Deteksi Perjalanan Kendaraan Untuk Mengukur Kepadatan Lalu Lintas Menggunakan Menggunakan OpenCV Python," *J. Ilmu Komput.*, vol. 3, no. 2, pp. 1–3, 2020.
- [8] S. Elkerdawy, M. Elhelw, and R. Sayed, "Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking," *ResearchGate*, vol. 252, no. November, pp. 381–390, 2013, doi: 10.1007/978-3-319-03413-3_27.
- [9] P. Viola and M. Jones, "Robust Real-time Object Detection Paul," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004, doi: <https://doi.org/10.1023/B:VISI.0000013087.49260.fb>.
- [10] Alexander Mordvintsev and Abid K, "OpenCV-Python Tutorials Documentation," 2017. <https://readthedocs.org/projects/opencv24-python-tutorials/downloads/pdf/stable/> (accessed Nov. 25, 2024).
- [11] J. Zhang, X. Zou, L. D. Kuang, J. Wang, R. S. Sherratt, and X. Yu, "CCTSDb 2021: A More Comprehensive Traffic Sign Detection Benchmark," *Human-centric Comput. Inf. Sci.*, vol. 12, 2022, doi: 10.22967/HGIS.2022.12.023.
- [12] A. Mashudi, F. Rofii, and M. Mukhsim, "Sistem Kamera Cerdas Untuk Deteksi Pelanggaran Marka Jalan," *JASEE J. Appl. Sci. Electr. Eng.*, vol. 1, no. 01, pp. 15–25, 2020, doi: 10.31328/jasee.v1i01.4.
- [13] W. S. Pambudi and B. M. Simorangkir, "Facetracker menggunakan metode Haar Like Feature dan PID pada model simulasi," *J. Teknol. dan Inform.*, vol. 2, no. 2, pp. 142–154, 2012.
- [14] Q. Wang, Q. Zhang, X. Liang, Y. Wang, C. Zhou, and V. I. Mikulovich, "Traffic lights detection and recognition method based on the improved yolov4 algorithm," *Sensors*, vol. 22, no. 1, pp. 1–20, 2022, doi: 10.3390/s22010200.
- [15] D. Indra, H. Herman, and F. S. Budi, "Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis Computer Vision," *Komputika J. Sist. Komput.*, vol. 12, no. 1, pp. 53–62, 2023, doi: 10.34010/komputika.v12i1.9082.
- [16] M. Hasanah *et al.*, "Automatic Car Detection Using Haar Cascade Classifier and Convolutional Neural Network for Traffic Density Estimation," *Indones. J. Artif. Intell. Data Min.*, vol. 4, no. 1, pp. 11–18, 2021.