

Algoritma *Scale Invariant Feature Transform* (SIFT) pada Deteksi Kendaraan Bermotor di Jalan Raya

YUSUP MIFTAHUDDIN, NUR FITRIANTI FAHRUDIN, MOCHAMAD FACHRY PRAYOGA

Program Studi Informatika, Institut Teknologi Nasional Bandung
Email: yusufm@itenas.ac.id

Received 30 Maret 2020 | *Revised* 30 April 2020 | *Accepted* 29 Mei 2020

ABSTRAK

Proses perhitungan jumlah kendaraan yang masih dilakukan secara manual dan membutuhkan banyak operator dalam pendataan. Berdasarkan hal itu, diperlukan sistem yang mampu mendeteksi dan mengklasifikasi kendaraan yang melintas di jalan raya secara otomatis. Dalam mengidentifikasi citra kendaraan, sistem menggunakan algoritma SIFT. Hasil fitur akan dibandingkan dengan metode K-Nearest Neighbor (KNN). Sistem dibangun untuk mendeteksi jenis kendaraan berat dengan mengukur tingkat akurasi keberhasilan berdasarkan nilai pencahayaan, jumlah objek, perubahan rotasi, serta pada kondisi siang dan malam hari. Dataset yang digunakan berjumlah 100 citra kendaraan berat. Kinerja sistem pada kondisi siang hari mendapat nilai presisi rata-rata 100%, nilai recall 54%, dan nilai akurasi 78%. Hasil pengukuran presisi dan recall, diperoleh nilai f-measure sebesar 67 %.

Kata kunci: *SIFT, kendaraan berat, K-Nearest Neighbour*

ABSTRACT

The process of collecting vehicles still done manually and requires a lot of human resources. Therefore, we need a system that can detect and classify vehicles passing on the highway automatically. SIFT is an algorithm for identification of an image. The features will be compared using the K-Nearest Neighbor (KNN) method. In this study, system will be designed to detect the type of heavy vehicle using the SIFT method to measure the accuracy of success based on the value of lighting, number of objects, changes in rotation, and day night conditions. Dataset used was 100 heavy vehicle images. The system performance during daytime conditions gets an average precision value of 100%, a recall value of 54%, and an accuracy value of 78%. From the results of precision and recall, the f-measure value is 67 %.

Keywords: *SIFT, heavy vehicles, K-Nearest Neighbour*

1. PENDAHULUAN

Pada era modern saat ini, kemacetan adalah hal yang sering dijumpai terutama di kota metropolitan. Upaya yang dilakukan untuk mengurai kemacetan adalah dengan memanfaatkan data volume kendaraan yang melintas di jalan raya kemudian diolah untuk memantau kepadatan lalu lintas. Pada umumnya, proses pendataan jumlah kendaraan dilakukan dengan menghitung satu persatu kendaraan (**Tarnoto, Lusiana, 2014**). Sebagai contoh, ketika terjadi libur panjang khususnya hari raya Idul Fitri, banyak petugas yang ditugaskan untuk mengatur serta menghitung setiap kendaraan yang lewat dan mengklasifikasikannya. Perhitungan manual ini membutuhkan banyak sumber daya manusia yang harus turun ke lapangan.

Berdasarkan Pedoman Kapasitas Jalan Indonesia (PKJI, 2014), kendaraan tergolong menjadi 4, diantaranya kendaraan ringan (kendaraan yang beroda 4), kendaraan berat (kendaraan yang beroda lebih dari 4), sepeda motor (kendaraan beroda 2 atau 3) dan kendaraan tak bermotor (kendaraan dengan roda yang digerakkan oleh orang atau binatang).

Penelitian sebelumnya tentang deteksi dan klasifikasi kendaraan di jalan raya dilakukan oleh Alvin Lazaro pada tahun 2017 menggunakan metode *Haar-like Feature* yang menghasilkan tingkat akurasi hingga 77,8% menggunakan citra rekaman video. Hal ini disebabkan adanya kesalahan pada klasifikasi yang disebabkan oleh faktor penskalaan objek kendaraan dimana kendaraan kecil terdeteksi menjadi kendaraan berat. Oleh karena itu, diperlukan pengembangan penelitian tersebut menggunakan metode lain untuk memperoleh tingkat akurasi yang lebih tinggi.

Al Kautsar dan Adi melakukan penelitian untuk mengembangkan sistem dengan menerapkan sensor *RFID* pada pendeteksian kendaraan dengan memiliki kelemahan pada detektor yang hanya dapat mendeteksi dengan keterbatasan jarak, dan biaya yang besar (**Al Kautsar, Adi, 2016**).

Wibowo, Muslim, Sarosa menerapkan deteksi tepi Canny pada proses segmentasi dan *Fuzzy C-Means* dalam menghitung kendaraan. Kontribusinya adalah pengambilan proses ekstraksi ciri dengan Canny (**Wibowo, Muslim, Sarosa, 2013**).

Adistyia, Muslim melakukan penelitian untuk mengklasifikasi jenis kendaraan dan menghitung banyaknya kendaraan melalui citra. Kontribusi penelitian ini adalah penggunaan metode *Backpropagation* dalam mengenali dan mengklasifikasi golongan kendaraan berdasarkan panjang dan lebar di titik sudut kendaraan, serta melakukan ekstraksi ciri menggunakan Sobel (**Adistyia, Muslim, 2016**).

Rosidin, Sugiantoro, Prayudi menerapkan SIFT dan histogram warna RGB dengan menghasilkan keakuratan yang lebih baik dalam menganalisa citra yang sudah diubah dari aslinya. (**Rosidin, Sugiantoro, Prayudi, 2018**).

Lionnie, Alaydrus melakukan percobaan menggunakan metode SIFT untuk mendeteksi pemalsuan pada gambar yang telah dimanipulasi dengan penyerangan jenis *copy-move* dan hasilnya dapat mendeteksi gambar yang telah di ubah arah dan ukurannya walaupun ada beberapa kecocokan yang salah (**Lionnie, Alaydrus, 2017**).

Pardede, Utami, Rochman mengkombinasikan algoritma *Principal Componen Analysis* dan SIFT untuk mengurangi dimensi pada citra (**Pardede, Utami, Rochman, 2017**).

Lazaro, Buliali, Amaliah melakukan penelitian untuk membuat sebuah program yang mampu melakukan deteksi jenis kendaraan serta menghitung kendaraan yang terdeteksi berdasarkan jenisnya. Kontribusi penelitian ini adalah melakukan klasifikasi jenis kendaraan di jalan raya karena penelitian sebelumnya hanya melakukan deteksi objek kendaraan.

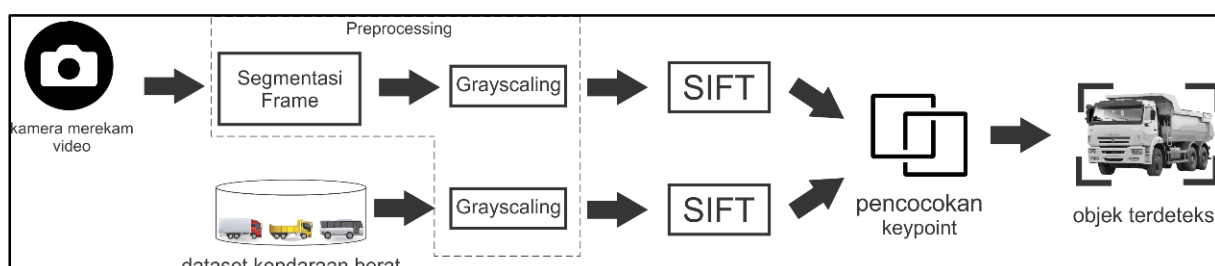
Berdasarkan pada penelitian-penelitian yang telah ada, penelitian ini menerapkan algoritma SIFT untuk mendeteksi kendaraan di jalan raya melalui kamera dengan jenis kendaraan yaitu kendaraan berat. Pengaruh pencahayaan objek yang diterima kamera pada saat pendeteksian akan diuji untuk mendapatkan hasil yang optimal. Algoritma SIFT dipilih karena bersifat lokal, khas, dan memiliki keunggulan dalam kuantitas, efisiensi, dan ekstensibilitas (Lowe, 2004).

Penelitian ini bertujuan untuk menghitung akurasi keberhasilan dari deteksi kendaraan berdasarkan jenis kendaraan berat dengan parameter pencahayaan pada kamera.

2. METODE PENELITIAN

2.1. Perancangan Sistem

Bagian ini memaparkan alur proses perancangan sistem pendeteksi objek kendaraan dengan menerapkan metode SIFT. Gambar 1 menunjukkan alur kerja sistem secara keseluruhan.



Gambar 1. Alur Kerja Sistem

Sistem yang dibangun adalah sistem deteksi kendaraan berdasarkan jenis kendaraan di jalan raya. Maka dari itu informasi pertama yang harus diketahui adalah citra kendaraan berdasarkan jenisnya yang dikumpulkan pada satu *dataset*. Sebelum dilakukan pencocokan, citra *query* didapatkan dari web kamera yang merekam video berupa *frame*. Algoritma SIFT digunakan untuk mengekstraksi fitur dari citra *query* dan citra latih pada *dataset* kendaraan yang sudah diklasifikasikan sebagai kendaraan berat kemudian dilakukan pencocokan *keypoint* antara kedua citra tersebut menggunakan *K-Nearest Neighbour*. Objek yang terdeteksi adalah kendaraan yang memiliki *keypoint* paling mirip kemudian objek akan ditandai dengan kotak berwarna hijau.

2.1.1 Proses Grayscale

Berikut contoh matriks yang diambil dari data latih dengan ukuran 3x3.

(0,0) R: 98 G: 155 B: 176	(0,1) R: 96 G: 153 B: 173	(0,2) R: 90 G: 148 B: 167
(1,0) R: 99 G: 155 B: 176	(1,1) R: 96 G: 153 B: 173	(1,2) R: 91 G: 148 B: 167
(2,0) R: 100 G: 155 B: 176	(2,1) R: 98 G: 153 B: 173	(2,2) R: 92 G: 148 B: 167

Matriks Citra RGB



(0,0) 145	(0,1) 142	(0,2) 137
(1,0) 145	(1,1) 142	(1,2) 137
(2,0) 145	(2,1) 141	(2,2) 138

Matriks Citra Grayscale

Proses perhitungan untuk titik koordinat 1,1:

$$\text{Grayscale} = 0.21 * 96 + 0.72 * 153 + 0.07 * 173$$

$$\text{Grayscale} = 20.16 + 110.16 + 12.11$$

$$\text{Grayscale} = 142.43 = 142$$

2.1.2. *Scale Invariant Feature Transform* (SIFT)

Metode SIFT menemukan dan menjelaskan fitur lokal citra. Algoritma SIFT menghasilkan *keypoint descriptor* yang dapat dibandingkan antara satu citra dengan citra lainnya sehingga dapat menentukan kemiripan antara kedua citra tersebut. SIFT memiliki empat tahapan utama yaitu deteksi *scale-space extrema*, penentuan *keypoint*, Penetapan orientasi pada setiap *keypoint*, dan menghitung vektor deskriptor. Urutan proses tersebut ditampilkan oleh Gambar 2.

Skala ruang citra merupakan fungsi $L(x, y, \sigma)$ dihasilkan dari perhitungan konvolusi kernel Gaussian Blur dapat dilihat pada Persamaan 1 Proses deteksi puncak skala ruang dilakukan menggunakan fungsi *Different of Gaussian* pada Persamaan 2 untuk mengidentifikasi titik-titik yang potensial.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2)$$

$$D(x, y, \sigma) = (L(x, y, k\sigma) - L(x, y, \sigma)) \quad (3)$$

keterangan:

$$G(x, y, \sigma) = \text{Gaussian Blur}$$

$$\pi = 3,14$$

$$I(x, y) = \text{Citra asli}$$

$$\sigma = 3$$

$$(L(x, y, \sigma) = \text{Gaussian Blur}, k = \sqrt{2})$$

$$e = 2,72 \text{ (nilai ketetapan)}$$

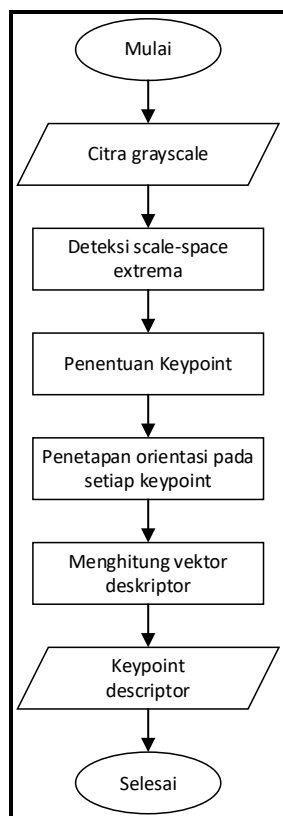
Proses penentuan *keypoint* dilakukan dengan mencari nilai *maxima* / *minima* DoG yang dideteksi dengan membandingkan sebuah piksel dengan tetangganya.

$$Z = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x} \quad (4)$$

$$D(Z) = D + \frac{1}{2} \frac{\partial D^{-1}}{\partial x} Z \quad (5)$$

Setelah ditemukan *keypoint*, selanjutnya proses penetapan orientasi pada setiap *keypoint* dengan cara mengumpulkan sudut gradien dan nilai disekitar *keypoint*.

$$M(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (6)$$



Gambar 2. Diagram Alir SIFT

Kemudian dicari orientasi yang unggul. Langkah terakhir adalah menghitung vektor deskriptor yang berujuan mencari invarian *keypoint* terhadap perubahan tingkat pencahayaan atau perubahan sudut pandang tiga dimensi sesuai Persamaan 7.

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (7)$$

2.1.2.1 Scale Space Extrema Detection

Contoh perhitungan pada titik koordinat $x = 1, y = 1$ dengan skala $\sigma=3, \pi = 3.14$ dan $e = 2.72$ adalah:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad G(x, y, \sigma) = \frac{1}{2 * 3.14 * 1.6^2} 2.72^{-(1^2+1^2)/2 * 1.6^2}$$

$$G(x, y, \sigma) = 0.0421$$

Setelah didapatkan nilai Gaussian Blur sebesar 0.0421, kemudian dicari nilai scale space yang mempunyai nilai *grayscale* sebesar 142 dengan Persamaan 2.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$L(x, y, \sigma) = 0.0421 * 142$$

$$L(x, y, \sigma) = 5.97$$

Berikut hasil perhitungan Gaussian Blur untuk setiap koordinat

(0,0) 9.02	(0,1) 7.42	(0,2) 4.13
(1,0) 7.26	(1,1) 5.97	(1,2) 3.30
(2,0) 3.90	(2,1) 3.21	(2,2) 1.80

Selanjutnya untuk dapat menghitung nilai *Different of Gaussian* (DoG), diperlukan perhitungan nilai *Gaussian blur* dan skala ruang pada koordinat yang sama yaitu (1,1) dengan skala $k\sigma$ dimana $k = \sqrt{2}$ dengan Persamaan berikut

$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-(x^2 + y^2)/2(k\sigma)^2}$$

$$G(x, y, k\sigma) = \frac{1}{2 * 3.14 * (\sqrt{2} * 1.6)^2} 2.72^{-(1^2 + 1^2)/2 * (\sqrt{2} * 1.6)^2} \quad G(x, y, k\sigma) = 0.0256$$

2.1.2.2 Keypoint Localization

Masukan dari tahap ini adalah nilai DoG yang telah didapatkan pada perhitungan sebelumnya yaitu sebesar 2.34 pada posisi koordinat (1.1). Untuk menentukan posisi ekstremum dari *keypoint* dapat menggunakan Persamaan 4 dengan contoh perhitungan sebagai berikut:

$$\begin{aligned} Z &= -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x} & Z &= -(2.34)^{-1} + (2.34) \\ Z &= -\left(\frac{\partial D}{\partial x} * x\right)^{-1} \frac{\partial D}{\partial x} & Z &= -\frac{1}{2.34} (2.34) \\ Z &= -(D * x) * x)^{-1} + (D * x) & Z &= -\frac{2.34}{2.34} \\ Z &= -(2.34 * 1) * 1)^{-1} + (2.34 * 1) & Z &= -1 \end{aligned}$$

Dengan demikian didapatkan hasil posisi ekstremum terletak pada nilai -1. Kemudian dihitung nilai *keypoint* pada posisi ekstremum tersebut dengan menggunakan Persamaan 5 dimana nilai D adalah DoG sebesar 2.34 dan Z adalah nilai ekstremum sebesar -1. Berikut adalah contoh perhitungan *keypoint*.

$$D(Z) = D + \frac{1}{2} \frac{\partial D^{-1}}{\partial x} Z \quad D(Z) = 2.34 + \frac{1}{2} (2.34^{-1} * 2) - 1 \quad D(Z) = 1.91$$

2.1.2.3 Penetapan Orientasi

Penggunaan *Gaussian smooth* L dengan skala terdekat dengan skala *keypoint*. Pada sampel citra L(x,y) hasil Gaussian Blur, magnitude M(x,y) dihitung dengan menggunakan rumus Persamaan 6 dimana $x = 1$, $y = 1$ dan perhitungannya adalah

$$M(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$M(x, y) = \sqrt{(L(1+1, 1) - L(1-1, 1))^2 + (L(1, 1+1) - L(1, 1-1))^2}$$

$$M(x, y) = \sqrt{(3.21 - 7.42)^2 + (3.3 - 7.26)^2}$$

$$M(x, y) = \sqrt{33.4057} = 5.78$$

2.1.2.4 Keypoint Descriptor

Untuk mencari *keypoint descriptor*, skala ruang pada citra $L(x,y)$ hasil Gaussian Blur dihitung menggunakan Persamaan 7 dimana $x = 1$ dan $y = 1$ dan perhitungannya adalah sebagai berikut.

$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \qquad \theta(x, y) = \tan^{-1} \frac{3.3 - 7.26}{3.21 - 7.42}$$

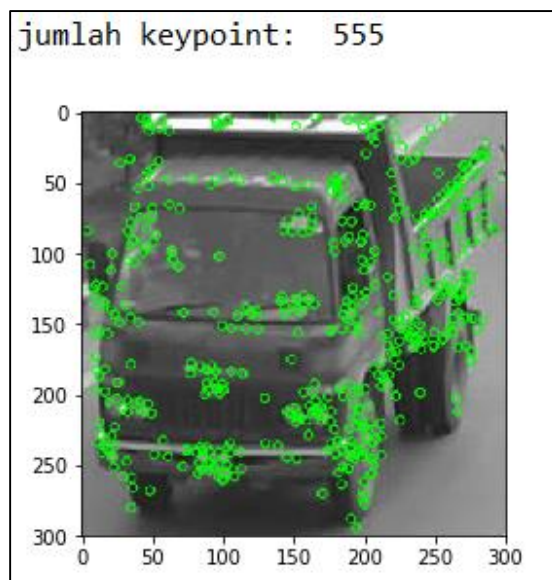
$$\theta(x, y) = \tan^{-1} \frac{L(1, 1+1) - L(1, 1-1)}{L(1+1, 1) - L(1-1, 1)} \qquad \theta(x, y) = \tan^{-1} 0.9406$$

$$\theta(x, y) = 0.7547 \text{ atau } 43.25^\circ$$

3. HASIL DAN PEMBAHASAN

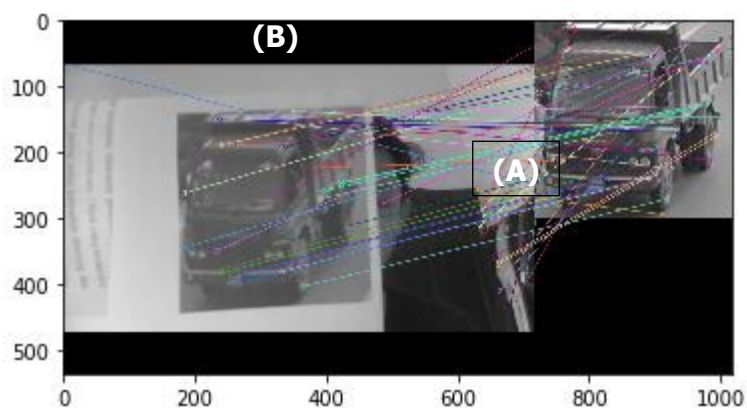
3.1. Pengujian Sistem

Algoritma SIFT akan mengekstraksi fitur dari citra yang ditangkap oleh kamera (*query*) dengan keluaran berupa *keypoint*. *Keypoint* tersebut kemudian akan dicocokkan dengan *keypoint* dari data latih. Gambar 3 menunjukkan *keypoint* yang dihasilkan dari citra kendaraan besar.



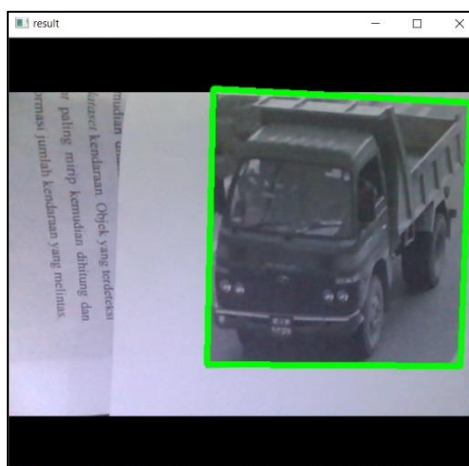
Gambar 1. Keypoint Citra

Jumlah *keypoint* yang dihasilkan dari suatu citra sangat banyak dan beragam sehingga pada pengujian ini akan dibatasi sebesar 50 *keypoint* yang akan dicocokkan menggunakan metode KNN.



Gambar 2. Pencocokan Keypoint

Gambar 4 terdapat garis-garis yang menghubungkan antara citra dari data latih (A) dan citra *query* (B) menunjukkan kecocokan antara *keypoint* kedua citra tersebut. Kemudian objek kendaraan citra *query* akan diberi tanda berupa kotak berwarna hijau seperti pada Gambar 5.



Gambar 3. Hasil Pencocokan

Pengujian dilakukan dengan citra yang memiliki 1, 2, 4, 6, dan 8 objek kendaraan berat yang berpengaruh terhadap ukuran objek yang akan dideteksi. Setiap pengujian dilakukan sebanyak 5 kali menggunakan parameter nilai pencahayaan (lux) dengan total pengujian 25 kali pada setiap parameter. Hasil dari pengujian ditunjukkan pada tabel berikut:

Tabel 1. Hasil Pengujian pada Pencahayaan 800 Lux

No	Pengujian	Pengujian pertama	Pengujian kedua	Pengujian ketiga	Pengujian keempat	Pengujian kelima
1	1 Objek	1 dari 1 Terdeteksi	1 dari 1 Terdeteksi	1 dari 1 Terdeteksi	1 dari 1 Terdeteksi	1 dari 1 Terdeteksi
2	2 Objek	2 dari 2 Terdeteksi	2 dari 2 Terdeteksi	2 dari 2 Terdeteksi	2 dari 2 Terdeteksi	2 dari 2 Terdeteksi
3	4 Objek	4 dari 4 Terdeteksi	3 dari 4 Terdeteksi	4 dari 4 Terdeteksi	4 dari 4 Terdeteksi	4 dari 4 Terdeteksi
4	6 Objek	6 dari 6 Terdeteksi	4 dari 6 Terdeteksi	4 dari 6 Terdeteksi	3 dari 6 Terdeteksi	4 dari 6 Terdeteksi
5	8 Objek	5 dari 8 Terdeteksi	6 dari 8 Terdeteksi	7 dari 8 Terdeteksi	5 dari 8 Terdeteksi	6 dari 8 Terdeteksi

Tabel 1 menunjukkan hasil pengujian pendeteksian objek kendaraan berat dengan nilai pencahayaan 800lux. Nilai pencahayaan ini didapat dari aplikasi *smartphone* yang menggunakan sensor cahaya. Untuk mengukur akurasi dari pengujian diatas, jumlah nilai seluruh objek yang terdeteksi dibagi dengan total percobaan. Seperti pada Persamaan 8

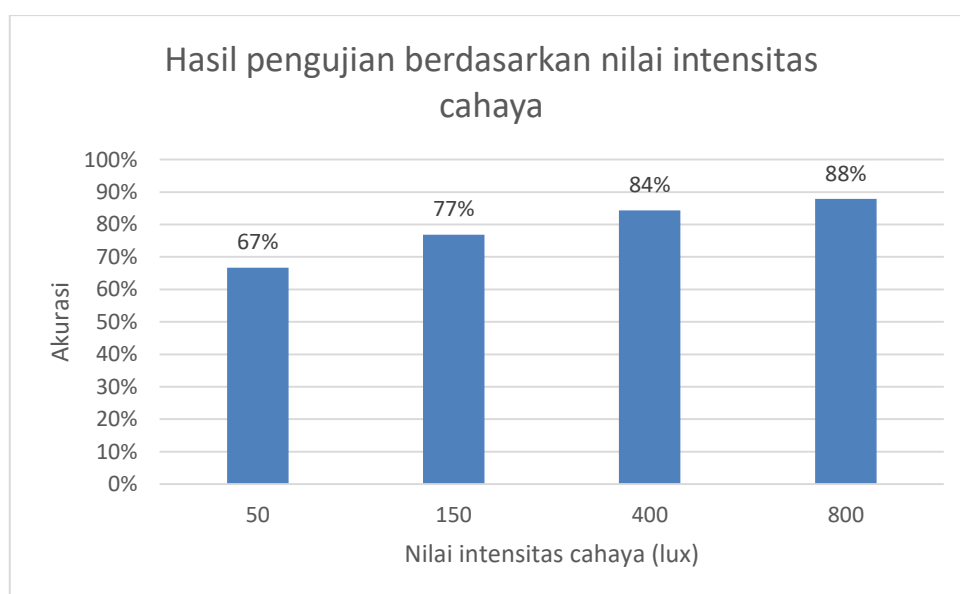
$$Akurasi = \frac{Jumlah\ nilai\ seluruh\ objek\ yang\ terdeteksi}{total\ percobaan} \quad (8)$$

Setiap hasil yang terdeteksi sesuai dengan jumlah objek sebenarnya diberi nilai 1. Sebagai contoh, pada pengujian ke 1, jumlah objek yang terdeteksi adalah 2 dari 2 yang diharapkan sehingga memiliki nilai 1. Pada pengujian ke 4 dari 6 objek, jumlah objek yang terdeteksi adalah 3 dari 6 objek, sehingga memiliki nilai 0,5. Sedangkan objek bukan kendaraan berat yang tidak terdeteksi memiliki nilai 1 karena membuktikan bahwa sistem hanya mendeteksi jenis kendaraan berat.

Tabel 2. Nilai Pengujian

No	Pengujian	Pengujian pertama	Pengujian kedua	Pengujian ketiga	Pengujian keempat	Pengujian kelima
1	1 Objek	1,0	1,0	1,0	1,0	1,0
2	2 Objek	1,0	1,0	1,0	1,0	1,0
3	4 Objek	1,0	0,8	1,0	1,0	1,0
4	6 Objek	1,0	0,7	0,7	0,5	0,8
5	8 Objek	0,6	0,8	0,9	0,6	0,8

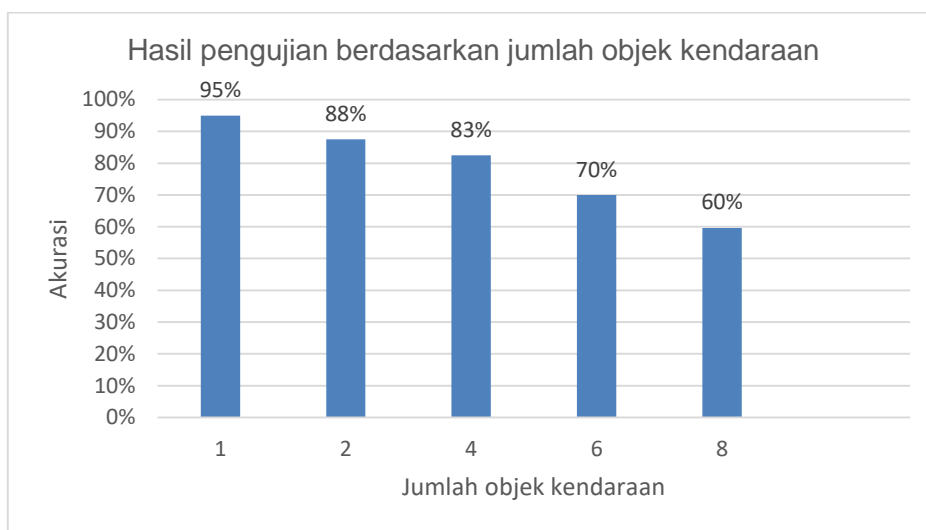
Dari hasil pengujian pada Tabel 2, jumlah nilai dari seluruh objek yang terdeteksi adalah 11,97 kemudian dibagi dengan total 25 percobaan sehingga menghasilkan tingkat akurasi sebesar 0,879 atau 88%. Pengujian selanjutnya adalah dengan mengubah nilai lux sebesar 50, 150, dan 400 yang hasilnya ditunjukkan pada Tabel 3.



Gambar 6. Grafik Pengujian Berdasarkan Pencahayaan

Dari Gambar 6 dapat dilihat bahwa nilai pencahayaan yang lebih tinggi dapat menghasilkan tingkat akurasi yang lebih tinggi. Pada nilai pencahayaan yang rendah, terdapat *noise* pada

hasil kamera yang mempengaruhi pencocokan sehingga objek yang buram tidak dapat terdeteksi.



Gambar 7. Hasil Pengujian Berdasarkan Jumlah Objek Yang Dideteksi

Gambar 7 menunjukkan banyaknya objek juga mempengaruhi pendeteksian. Semakin banyak objek, semakin kecil akurasi yang dihasilkan. Banyaknya objek menyebabkan ukuran objek tersebut menjadi lebih kecil atau jarak yang lebih jauh sehingga dibutuhkan kamera dengan resolusi yang tinggi agar objek terlihat jelas. Dari hasil pengujian terhadap rotasi objek, terbukti bahwa algoritma SIFT ini tahan terhadap perubahan rotasi yang ditandai dengan terdeteksinya objek dengan kecocokan keypoint lebih dari 50 (nilai threshold) pada perubahan rotasi antara 0°-180° searah dan berlawanan jarum jam.

Untuk menguji kinerja sistem dilakukan pengujian deteksi kendaraan di jalan raya pada kondisi siang dan malam hari dan menghasilkan nilai *recall* 54%, *precision* 100%, *accuracy* 78%, dan *f-measure* 67%. Untuk lebih jelasnya dapat dilihat pada Tabel 3.

Tabel 3. Pengujian Pada Kondisi Siang Hari

No	Pengujian	TP	TN	FP	FN	Recall	Precision	Accuracy	F1
1	Gambar 1	5	10	0	3	63%	100%	83%	77%
2	Gambar 2	2	13	0	1	67%	100%	94%	80%
3	Gambar 3	1	8	0	5	17%	100%	64%	29%
4	Gambar 4	3	9	0	5	38%	100%	71%	55%
5	Gambar 5	1	0	0	0	100%	100%	100%	100%
6	Gambar 6	1	2	0	2	33%	100%	60%	50%
7	Gambar 7	2	21	0	1	67%	100%	96%	80%
8	Gambar 8	4	2	0	4	50%	100%	60%	67%
Rata-rata						54%	100%	78%	67%

Sistem tidak dapat mendeteksi kendaraan pada kondisi malam hari disebabkan kurangnya jumlah pencahayaan dan cahaya yang dikeluarkan oleh kendaraan menyebabkan kamera tidak dapat menangkap objek kendaraan dengan jelas. Hal ini berbanding lurus dengan pengujian sistem terhadap nilai pencahayaan yang telah dilakukan pada tahap sebelumnya

4. KESIMPULAN

Pada penelitian ini telah dipaparkan bagaimana penerapan algoritma Scale Invariant Feature Transform (SIFT) dalam mendeteksi kendaraan berat. Tahapan utama dari sistem pendeteksian ini adalah melakukan ekstraksi fitur pada citra uji dan kumpulan citra pada dataset menggunakan SIFT untuk mendapatkan keypoint dari setiap citra. Keypoint dari citra uji akan dicocokkan dengan keypoint dari citra pada dataset menggunakan K-Nearest Neighbors (K-NN) untuk mendapatkan tingkat kemiripan yang tinggi. Hasil pengujian menunjukkan bahwa nilai pencahayaan berpengaruh pada pendeteksian objek dengan pencahayaan optimal pada 800 lux atau pada kondisi siang hari. Semakin banyak jumlah objek yang akan dideteksi semakin kecil pula ukuran objek tersebut dan menyebabkan objek kurang terlihat jelas oleh kamera. Sedangkan posisi objek yang dirotasi searah sumbu x tetap terdeteksi sehingga tidak berpengaruh terhadap pendeteksian karena keypoint descriptor yang dihasilkan algoritma SIFT tahan terhadap perubahan posisi benda. Berdasarkan hasil pengujian kinerja sistem pada kondisi siang hari, nilai accuracy yang didapatkan sebesar 78%, precision 100%, recall 54% dan f-measure 67%. Sedangkan pada kondisi malam hari, sistem tidak dapat mendeteksi objek dikarenakan kurangnya nilai pencahayaan. Hasil pengujian sistem tersebut menunjukkan bahwa sistem berhasil mendeteksi hanya kendaraan berat, namun tidak dapat mendeteksi kendaraan berat yang terhalang oleh kendaraan lain.

DAFTAR RUJUKAN

- Adisty, R., & Muslim, M. A. (2016). Deteksi dan Klasifikasi Kendaraan menggunakan Algoritma Backpropagation dan Sobel. *Journal of Mechanical Engineering and Mechatronics*, 1(2), 65–73.
- Al Caruban, R., Sugiantoro, B., & Prayudi, Y. (2018). Analisis Pendeteksi Kecocokan Objek Pada Citra Digital Dengan Metode Algoritma Sift Dan Histogram Color Rgb. *Cyber Security dan Forensik Digital*, 1(1), 20-27.
- Al Kautsar, H. V., & Adi, K. (2016). Implementasi Object Tracking Untuk Mendeteksi Dan Menghitung Jumlah Kendaraan Secara Otomatis Menggunakan Metode Kalman Filter Dan Gaussian Mixture Model. *Youngster Physics Journal*, 5(1), 13-20.
- Husada, M. G., Utami, D. B., & Zar, I. (2019). Karakteristik Metode Sift dalam Aplikasi Sistem Pengenalan Motif Batik. *MIND Journal*, 4(2), 122-131.
- Kadir, A., & Susanto, A. (2013). Teori dan aplikasi pengolahan citra. *Yogyakarta: Andi*.
- Kementerian Pekerjaan Umum. (2014). *Pedoman Kapasitas Jalan Perkotaan*.
- Lionnie, R., & Alaydrus, M. (2017). Sistem Pendeteksi Gambar Termanipulasi Menggunakan Metode SIFT. *Techné: Jurnal Ilmiah Elektroteknika*, 16(02), 133-140.

- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- Susanto, A. (2019). Penerapan Operasi Morfologi Matematika Citra Digital Untuk Ekstraksi Area Plat Nomor Kendaraan Bermotor. *Pseudocode*, 6(1), 49-57.
- Tarnoto, T., & Lussiana, L. (2014). Rancang Bangun Penghitung Kendaraan Secara Otomatis Berbasis Client Server. *Jurnal Ilmiah Ilmu Komputer Program Studi Sistem Komputer*.
- Wibowo, D. W., Muslim, M. A., & Sarosa, M. (2014). Perhitungan Jumlah dan Jenis Kendaraan Menggunakan Metode Fuzzy C-means dan Segmentasi Deteksi Tepi Canny. *Jurnal EECCIS*, 7(2), 103-110.