

DETEKSI DAN KLASIFIKASI RAMBU LALU LINTAS MENGGUNAKAN SUPPORT VECTOR MACHINE

TRAFFIC SIGN DETECTION AND CLASSIFICATION USING SUPPORT VECTOR MACHINE VECTOR

Agnes Dirgahayu Palit ¹, Mohamad Syahrul Mubarak ², Kurniawan Nur Ramadhani, S.T., M.T. ³

^{1,2,3} Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
¹ adirgapatit@gmail.com ² msyahrulmubarak@gmail.com ³ andiess2006@gmail.com

Abstrak

Traffic Sign Detection and Classification (Sistem Deteksi dan Klasifikasi Rambu Lalu Lintas) merupakan sistem untuk mendeteksi dan mengklasifikasikan rambu lalu lintas pada citra. Pada sistem ini, proses deteksi objek rambu lalu lintas, dibagi menjadi 2 bagian, yaitu dengan menyeleksi *channel* warna yang diinginkan (yang sesuai dengan warna rambu lalu lintas yang di cari), sebagai deteksi warna, dan menghitung derajat kebundaran dan segiempat objek sebagai deteksi bentuk objek. Sebagai ekstraksi ciri, *Histogram of Oriented Gradient* (HOG) digunakan untuk ekstraksi ciri bentuk, *colour moment* untuk ekstraksi warna. Sedangkan untuk klasifikasi menggunakan *Support Vector Machines* (SVM). Sistem yang telah dibangun pada penelitian ini menghasilkan performansi sebesar 93.5946% menggunakan *micro average f1-score*.

Kata Kunci: *Traffic sign detection and recognition, Support Vector Machine, Histogram of Oriented Gradient, colour moment*

Abstract

Traffic Sign Detection and Classification is system that build to detect and classification traffic sign in picture. This system, there are 2 part for detection object. First, object detection based on colour. It is channel colour selection (convenient with traffic sign colour that searching for). And second one, object detection based on shape, to measure roundness and squareness object. For Feature extraction, *Histogram of Oriented Gradient* (HOG) is used to shape extraction, *colour moment* is used to colour extraction dan *Support Vector Machine* (SVM) for traffic sign classification. System that has been builded, has generated 93.5946% for performace, with *micro average f1-score*.

Keyword: *Traffic sign detection and recognition, Support Vector Machine, Histogram of Oriented Gradient, colour moment.*

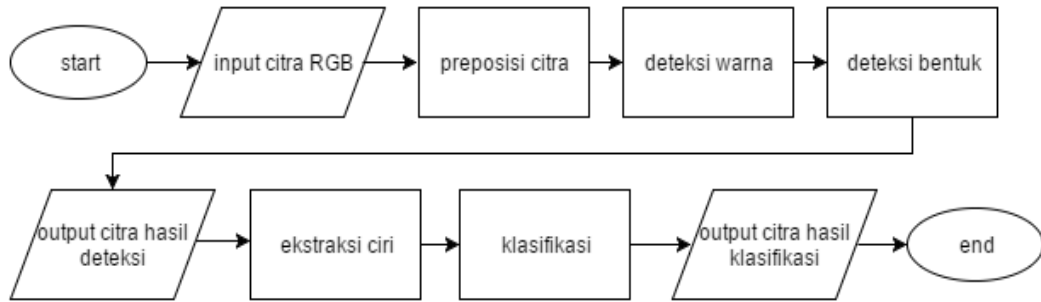
1. Pendahuluan

Kota-kota besar pasti tidak lepas dengan penggunaan rambu lalu lintas untuk meningkatkan keselamatan pengguna jalan. Rambu lalu lintas dirancang untuk membantu pengemudi mencapai tujuan mereka dengan aman, melalui penyediaan informasi rambu yang berguna [1]. Rambu lalu lintas dirancang dan ditempatkan di tempat yang mudah untuk dilihat [2], sehingga informasi visual seperti kewajiban mengemudi di jalur yang tepat, pembatasan kecepatan, dan lain-lain [3] dapat tersampaikan dengan baik pada pengendara mobil.

Meskipun demikian, hal yang tidak diinginkan bisa terjadi ketika informasi yang tersimpan dalam rambu lalu lintas tidak diterima dengan baik di jalan. Sebagai contoh, jika tanda-tanda jalan yang tidak terjawab atau salah memahami tanda-tanda, bahkan mungkin diabaikan. Kecelakaanpun itu bisa terjadi. Terlihat bahwa hal itu dapat menyebabkan konsekuensi yang fatal yang dapat merenggut nyawa manusia. Berdasarkan statistik terbaru menunjukkan bahwa lebih dari 98% dari kecelakaan mobil terjadi karena pengemudi terganggu [1].

Saat ini teknologi identifikasi objek rambu lalu lintas secara otomatis, dapat menjadi salah satu alternatif dan menjadi cikal-bakal terobosan baru dalam keselamatan berkendara, karena meningkatkan keamanan lalu lintas adalah tujuan yang penting pada sistem transportasi pintar [4]. Salah satunya adalah dengan membangun Sistem Deteksi dan Klasifikasi Rambu Lalu Lintas, yang merupakan sistem yang dibangun untuk mendeteksi dan mengklasifikasikan kelompok-kelompok pixel citra rambu lalu lintas yang didapatkan, ke dalam kategori kelas rambu lalu lintas yang telah ditentukan. Dalam tulisan ini mendeteksi tanda-tanda lalu lintas yang ada di jalan raya berdasarkan warna dan bentuk yang diharapkan, menggunakan segmentasi warna yang akan mengekstraksi ciri bentuk dengan menggunakan HOG dan ekstraksi ciri warna dengan menggunakan *colour moment* dari rambu lalu lintas, dan diikuti dengan mengukur tingkat kebulatan dan segiempat rambu lalu lintas. Selain itu, sistem juga akan

mengklasifikasikan tanda-tanda lalu lintas berdasarkan tiga kategori rambu lalu lintas pada umumnya, dengan menggunakan Support Vector Machine (SVM). Ikhtisar alur program dapat dilihat pada Bagan 1.



Bagan 1 : ikhtisar alur program

2. Deteksi objek berdasarkan warna

Pada bagian deteksi, sistem mendapatkan RGB gambar gambar pinggir jalan, dan itu akan mengambil nilai merah, biru dan kuning untuk setiap pixel yang menangkap. RBG dan Y nilai untuk setiap pixel ditunjukkan pada Tabel 1.

Tabel 1 : Nilai RGB - CMY untuk setiap warna pada citra

warna	Nilai RGB – CMY					
	R	G	B	C	M	Y
<i>s</i>	0	0	0	0	0	0
<i>Red</i>	255	0	0	0	1	1
<i>Green</i>	0	255	0	1	0	1
<i>Blue</i>	0	0	255	1	1	0
<i>Cyan</i>	0	255	255	1	0	0
<i>Magenta</i>	255	0	255	0	1	0
<i>Yellow</i>	255	255	0	0	0	1

Teknik *filter* warna yang diterapkan untuk menentukan lokasi rambu lalu lintas di gambar sisi jalan asli. Biru, merah, kuning dilakukan pada gambar. Kemudian warna RGB dalam gambar ditransform ke-modus biner.

3. Deteksi objek berdasarkan bentuk

Setiap objek yang telah diperoleh dari proses deteksi warna, selanjutnya harus diperiksa dengan deteksi bentuk, dengan mengukur derajat kebulatan (untuk rambu *Mandatory* dan *Prohibition*) dan segiempat (untuk rambu *Danger*) dengan memperhitungkan area perimeter dari objek. Contoh input gambar RGB dapat dilihat pada Gambar 1.



Gambar 1 : Contoh Inputan Gambar RGB

Oleh karena bentuk dari objek yang di cari adalah berbeda, maka hal ini mempengaruhi cara menghitung nilai perimeter dari setiap objek. Untuk deteksi objek rambu *Mandatory* dan *Prohibition* cara mencari nilai perimeter dihitung berdasarkan selisih setiap pixel pada koordinat x dan y. Luas (area) yang digunakan adalah luas lingkaran. Perhitungannya dapat dilihat pada persamaan (1) dan (2).

$$Perimeter = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (1)$$

$$Area = \pi r^2 \quad (2)$$

$$Roundness = \frac{4 \times Area \times \pi}{(Perimeter)^2} \quad (3)$$

Khusus untuk rambu *Danger*, hitung derajat segiempat. Dengan menghitung luas segiempat (4), keliling (5), dan mencari derajat segiempat hasil kuadrat dari pembagian nilai minimal dan maksimal dari akar kuadrat dari luas dan keliling per empat (6) (7) (8).

$$Area = p \times l \quad (4)$$

$$Perimeter = 2(p + l) \quad (5)$$

$$Min = \min(\sqrt{Area}, \frac{Perimeter}{4}) \quad (6)$$

$$Min = \min(\sqrt{Area}, \frac{Perimeter}{4}) \quad (7)$$

$$Squareness = \left(\frac{Min}{Max}\right)^2 \quad (8)$$

Keterangan :

Perimeter	: keliling objek
Area	: luas
Roundness	: derajat kebundaran
Squareness	: derajat segiempat
π	: phi (3.14)
r	: radius
p	: panjang
l	: lebar

Masing-masing derajat kebundaran dan segiempat yang diperoleh diseleksi berdasarkan *threshold* yang ditentukan. *Threshold* yang ada berguna sebagai penyeleksi apakah objek yang ada ditentukan sebagai rambu lalu lintas atau tidak.

4. HOG

Histogram of oriented gradients (HOG) adalah keterangan fitur yang digunakan untuk mendeteksi benda-benda dalam *computer vision* dan pengolahan citra. Teknik HOG menghitung kejadian orientasi gradien dalam porsi lokal dari gambar pada *window detection* (wilayah tujuan – ROI (Region of Interest)) [2].

Sebelum dilakukan pengenalan rambu-lalu lintas, citra harus dipotong dan diubah ukurannya untuk 64 x 64 pixel agar sesuai dengan gambar di data uji standar. Kandidat yang dideteksi sebagai rambu diekstraksi dengan HOG untuk mendapatkan ekstraksi fitur dan *colour moment* untuk ekstraksi warna.

Dataset secara keseluruhan di atur ukurannya secara seragam, misalnya 64 x 64 piksel, kemudian diekstrak fitur menggunakan HOG. Dengan metode ini, akan diperoleh ciri bentuk yang khas dari rambu lalu lintas dan tepi yang kuat, maka gradien ythe efisien dapat menangkap fitur ini. Di sisi lain, HOG dapat membantu untuk mencapai skala invarian.

Pertama, hitung orientasi gradien dengan mendiskritkan dalam sudut biner. Perhitungan gradien dengan filter 2 dimensi, yaitu secara vertikal (y) dan horisontal (x). Lalu hitung masing-masing turunannya dan hitung magnitude (besar gradien) seperti pada persamaan di bawah ini.

$$\text{sumbu } x : \frac{df}{dx} = \frac{f(x+h) - f(x-h)}{2h} \quad (9)$$

$$\text{sumbu } y : \frac{df}{dy} = \frac{f(x+h) - f(x-h)}{2h} \quad (10)$$

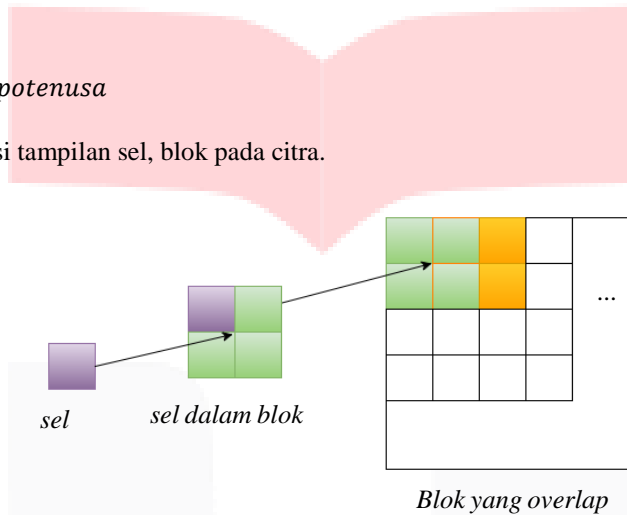
$$R = \sqrt{x^2 + y^2} \quad (11)$$

$$\theta = \tan^{-1}\left(\frac{x}{y}\right) \quad (12)$$

dengan:

- R* : magnitude
- θ* : orientasi gradien
- df* : turunan *f*
- dx* : turunan *x*
- dy* : turunan *y*
- x* : sumbu *x*
- y* : sumbu *y*
- R* : panjang hipotenusa

Berikut adalah ilustrasi tampilan sel, blok pada citra.



Gambar 2 : Sel yang menyusun blok

Dari Gambar 2 di atas terlihat bahwa citra akan di bagi kedalam sejumlah blok tertentu, misalnya 7x7 blok persegi dan bila diperlukan diimplementasi sejumlah overlap pada citra.

Kemudian, bagi orientasi gradien ke dalam sejumlah bin. Dan lakukan konkatinasi histogram, sehingga dihasilkan jumlah dimensi fitur citra sebagai berikut, misalkan dibagi jumlah blok = 7 x 7 = 49, dalam 1 blok = 4 sel (2 x 2 sel). Dan memiliki 9 bins. Maka jumlah dimensi fiturnya untuk 1 citra adalah 49 x 4 x 9 = 1764.

Setelah dilakukan konkatinasi dari hasil perhitungan sebelumnya, akan diterapkan normalisasi pada setiap blok.

$$L_1 - \text{norm} : v = \frac{v}{\sqrt{\|v\|^2 + \epsilon}} \quad (13)$$

$$L_2 - \text{norm} : v = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \quad (14)$$

dengan :

- v* : fitur blok
- $\epsilon = 0.1$ (konstanta)
- L* - norm : normalisasi

5. Colour Moment

Metode color moment digunakan untuk mendapatkan ekstraksi ciri warna dari citra inputan. Yang dilakukan dalam color moment adalah mencari *mean* (rata-rata), standar deviasi dan skewness dari layar R,G, dan B.

Menghitung mean bertujuan untuk mendapatkan informasi nilai rata-rata warna dari setiap channel citra (15). Standar devias dihitung untuk mencari nilai sebaran distribusi warna pada pixel citra (16). Dan *skewness* untuk menghitung derajat asimetri dari sebaran warna pada citra (17) [3].

$$mean_i = \sum_{j=1}^n \frac{p_{ij}}{n} \quad (15)$$

$$\sigma_i = \sqrt{\left(\frac{\sum_{j=1}^n (p_{ij} - mean_i)^2}{n}\right)} \quad (16)$$

$$s = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^n (p_{ij} - mean_i)^3\right)} \quad (17)$$

Dimana n adalah jumlah pixel dan citra inputan, p_{ij} adalah pixel ke-j untuk layar ke-i pada citra, dan σ_i adalah standar deviasi ke i.

6. SVM

Support Vector Machines (SVM) merupakan klasifikasi pola dan teknik regresi berdasarkan dasar matematika dari teori statistika [4]. Klasifikasi dilakukan dengan pengenalan pola yang dibantu dengan pembangunan batas-batas keputusan yang optimal dengan memisahkan data ke dalam beberapa kategori yang ditentukan dengan melakukan pendekatan fungsi real (dikenal juga dengan fungsi regresi) dengan juga membangun fungsi dengan interpolasi terbaik yang di berikan oleh dataset (kumpulan data) [5]

Pada SVM terdapat beberapa kernel. Kernel yang ada antara lain linier, quadratic, radial, polinomial dan sigmoid.

Tabel 2 : Daftar definisi fungsi kernel [12]

No.	Kernel	Definisi fungsi	
1	Linear	$K(x,y) = x \cdot y$	(18)
2	Radial (Gaussian)	$K(x,y) = \exp\left(-\frac{\ x-y\ ^2}{2\sigma^2}\right)$	(19)
3	Polinomial	$K(x,y) = (x \cdot y)^d$	(20)
4	Sigmoid	$K(x,y) = \tanh(x \cdot y + c)^d$	(21)

Namun secara umum kernel linier memiliki beberapa sifat yang sangat baik, dan ide utama dari SVM adalah untuk membangun suatu *hyperline* sebagai pengambil keputusan dari sedemikian rupa sehingga margin pemisahan antara contoh positif dan negatif di maksimalkan. Lebih tepatnya, SVM merupakan implementasi pemikiran metode minimisasi risiko struktural [6].

SVM merupakan salah satu metode kernel yang paling sukses dengan memberikan label pada data latih :

$$\{(x_i, y_i)\}_{n_i=1} \quad (22)$$

dengan :

$$x_i \in R_d, R : \text{objek}$$

$$y_i \in \{-1, +1\};$$

$$i=\{1,2,3,\dots\}$$

Pada metode SVM *supervised*, total citra yang dikelompokkan untuk membangun kernel. Kemudian, modifikasi dari kernel dasar selesai. Sejumlah SVM yang dilatih secara terpisah menggunakan algoritma *bootstrap* dan di agregasi dengan sarana teknik kombinasi yang sesuai. Kernel yang dimaksud disini adalah fungsi encoding yang memiliki kesamaan dengan sampel yang tidak memiliki label.

Inti dari metode ini adalah menemukan *hyperplane* yang merupakan garis yang dapat memisahkan dua kelompok data dengan label tertentu (misalnya $y=-1$ dan $y=+1$), dengan lebar margin paling maksimal.

Jika x adalah vektor pada ruang vektor, maka fungsi *hyperplane* dapat ditulis sebagai berikut :

$$|\vec{w} \cdot \vec{x}_i + b| \quad (23)$$

Dengan w adalah *weight* sebagai koefisien vektor dan b adalah bias. Jika asumsi hyperplane sudah membagi sempurna kedua kelas, maka dapat dirumuskan pattern (x_i) yang memenuhi formula sebagai berikut :

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \quad (24)$$

$$\vec{w} \cdot \vec{x}_i + b \leq 1 \quad (25)$$

Jarak antara vektor training x_i dan hyperplane disebut margin yang didefinisikan sebagai :

$$\frac{|\vec{w}_i \cdot \vec{x}_i + b|}{\|\vec{w}\|} \quad (26)$$

Dengan memaksimalkan nilai jarak antara *hyperplane* dan titik anggota pada masing-masing kelas dapat ditemukan margin terbesar, yaitu dengan $(1/\|\vec{w}\|)$. Hal ini dikaitkan dengan *Quadratic Programming* (QP) problem yang berfungsi mencari titik minimal persamaan (27) dengan memperhatikan constrain persamaan (23).

$$\min_w \tau(w) = \frac{1}{2\|w\|^2} \quad (27)$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \quad (28)$$

Masalah ini dapat dipecahkan dengan Lagrange Multiplier sebagai salah satu teknik komputasi.

$$L(\vec{w}, b, \alpha) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0) \quad (29)$$

$$i = 1, 2, 3, \dots, l$$

α_i (*alpha/Lagrange Multiplier*) berfungsi menemukan minimum/maksimum lokal yang bergantung pada batasan constrain pada persamaan (23). Nilai α_i itu sendiri bernilai lebih besar atau sama dengan nol (31). Hal yang perlu diperhatikan adalah mengoptimalkan nilai pada persamaan (29), dengan $L = 0$ sebagai titik optimal gradien. Hal tersebut dapat dilakukan dengan memaksimalkan nilai L menurut α (30), dan meminimalkan nilai L menurut w .

Maximize :

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \quad (30)$$

dengan :

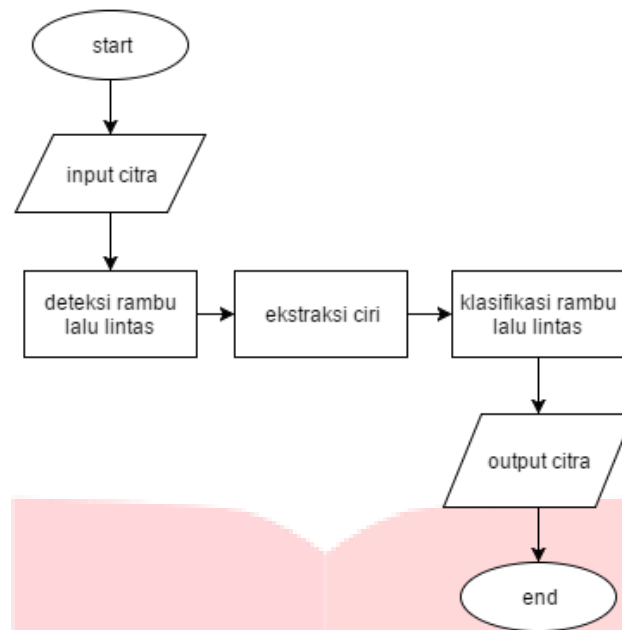
$$\alpha_i \geq 0 \quad (31)$$

$$\sum_{i=1}^N (\alpha_i y_i) = 0 \quad (32)$$

Setelah menemukan nilai α_i , cari data yang bersesuaian dengan $\alpha_i \geq 0$. Data yang bersesuaian tersebut yang kita sebut dengan support vector [7].

7. Rancangan Sistem

Sistem Deteksi dan Klasifikasi Rambu Lalu lintas (*Traffic Sign Detection and Classification*) yang dibangun pada Tugas Akhir ini adalah sebuah sistem yang memiliki tujuan untuk membantu manusia untuk mendeteksi dan mengklasifikasikan bentuk dan makna dari rambu lalu lintas. Diagram yang mendeskripsikan langkah-langka yang akan dilakukan oleh sistem secara umum, dapat dilihat pada Bagan 2.



Bagan 2 : Alur kerja program secara umum

Setelah citra berhasil diekstraksi dan dikelompokkan ke dalam data set yang berbeda, dataset yang dijadikan sebagai data latih akan diproses atau dipelajari dalam sistem untuk memperoleh kelas-kelas dari sekumpulan data latih yang diidentifikasi.

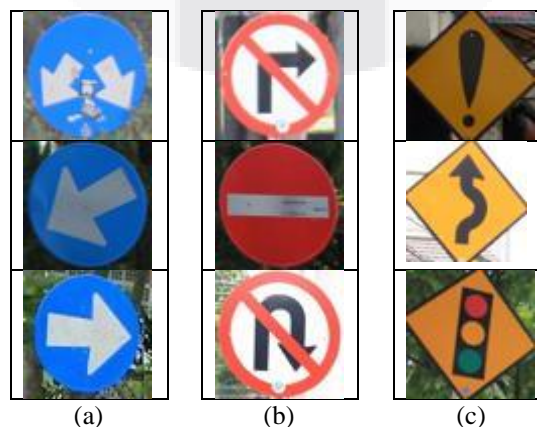
Secara umum, terdapat beberapa langkah yang perlu dilakukan oleh sistem untuk mendeteksi dan mengklasifikasikan bentuk atau pola pada rambu lalu lintas, diantaranya adalah preposisi citra, deteksi warna, deteksi bentuk, ekstraksi ciri, dan klasifikasi.

8. Pengujian dan Hasil Analisis

Dataset yang digunakan diambil dari Diambil dari seputaran kab./kota Bandung (lembang, Dago, Dago Atas, Cihampelas, Ciumbeluit – Punclut, jl.Diponegoro-kawasan Gedung Sate, Sukajadi, Setiabudi, kawasan Buah batu, Batununggal, kawasan jl.Asia Afrika), kawasan Telkom, dan Google street.

Citra yang dideteksi dan dikenali terdiri dari rambu tanda wajib (*Mandatory* : biru, bulat, putih), rambu tanda larangan (*Prohibitory* : merah, hitam, putih), dan rambu tanda bahaya (*Danger* : kuning, hitam). Dengan spesifikasi data sebagai berikut :

Ukuran data latih	: 64 x 64 px
Ukuran data tes	: 3088 x 2056 px
Warna	: RGB
Format citra	: *.jpeg



Gambar 2 : Contoh kategori data rambu-rambu lalu lintas di Indonesia

(a) Rambu Mandatory, (b)Rambu Prohibition, (c)Rambu Danger

Kondisi citra rambu lalu lintas pada *dataset*, antara lain memiliki kondisi kebisingan pada citra yang sedikit (dataset tidak cacat: plat rambu tidak terlipat, warna rambu masih jelas, tertutup objek lain dan tidak tercoret), kondisi pengambilan dataset langit berawan, cerah-berawan, kondisi objek rambu lalu lintas tepat berhadapan dengan kamera, dengan jarak objek terhadap kamera $\pm 3-4$ meter.

Seperti yang dapat dilihat pada tabel 1, Dataset yang ada dibagi menjadi beberapa kelas berdasarkan jenis dan bentuk yang dikenali secara umum, antara lain sebagai berikut :

- a. Kelas 1 : rambu tanda wajib / *Mandatory*
- b. Kelas 2 : rambu tanda larangan / *Prohibitory*
- c. Kelas 3 : rambu tanda bahaya / *Danger*
- d. Kelas 4 : bukan rambu / *Non sign*

Pembagian dataset yang digunakan pada pengujian, dapat dilihat pada Tabel 3.

Tabel 3 : Tabel pembagian data pengujian

Jenis Citra		Jumlah Objek Rambu	
Data Uji (68)	Citra yang mengandung Rambu (62)	Mandatory	26
		Prohibition	27
		Danger	19
	Citra yang tidak mengandung Rambu (6)		
Total Objek Rambu		72	

8.1 Analisa hasil pengujian

Berikut hasil pengujian untuk semua skenario.

8.1.1 Skenario 1-1

Skenario 1-1 bertujuan untuk mencari konfigurasi terbaik dari ukuran blok pada citra, konfigurasi yang diujikan adalah 4 x 4 blok dan 8 x 8 blok. Parameter lain yang berkombinasi dengan skenario 1-1 dan di atur adalah antara lain Besar overlap = 50%, jumlah sel = 4 x 4 sel, jumlah bin = 9, parameter *colour moment* yang dihitung *mean* dan standar deviasi dan fungsi kernel SVM adalah kernel linier.

Tabel 4 : hasil skenario 1-1

Jumlah blok	F1-score (%)	Akurasi (%)
4 x 4 blok	65,3846	82,6923
8 x 8 blok	91,0256	95,5128

Berdasarkan Tabel 4, pembagian jumlah blok mempengaruhi performansi akurasi sistem. Konfigurasi terbaik untuk ukuran 8 x 8 blok adalah dengan performansi f1-score 92,3077 %. konfigurasi tersebut akan digunakan pada pengujian skenario selanjutnya.

8.1.2 Skenario 1-2

Skenario 1-2 bertujuan untuk mencari konfigurasi terbaik besar overlap yang dilakukan, konfigurasi yang diujikan adalah 0% dan 50% overlap. Parameter lain yang berkombinasi dengan skenario 1-2 dan di atur adalah antara lain jumlah blok = 8 x 8 blok, jumlah sel = 4 x 4 sel, jumlah bin = 9, parameter *colour moment* yang dihitung *mean* dan standar deviasi dan fungsi kernel SVM adalah kernel linier.

Tabel 5 : hasil skenario 1-2

Besar overlap	F1-score (%)	Akurasi (%)
0%	93,5897	96,7949
50%	91,0256	95,5128

Berdasarkan Tabel 5, keberadaan overlap akan mempengaruhi performansi akurasi sistem. Konfigurasi terbaik adalah dengan malakukan 0% overlap, dengan performansi f1-score 93,5897 %. Konfigurasi tersebut akan digunakan pada pengujian skenario selanjutnya.

8.1.3 Skenario 1-3

Skenario 1-3 bertujuan untuk mengetahui pengaruh dari konfigurasi dari besar ukuran sel dalam blok yang telah dikonfigurasi sebelumnya. Parameter lain yang berkombinasi dengan skenario 1-3 dan di atur adalah antara lain jumlah blok = 8 x 8 blok, besar overlap = 0 %, jumlah bin = 9, parameter *colour moment* yang dihitung *mean* dan standar deviasi dan fungsi kernel SVM adalah kernel linier.

Tabel 6 : hasil skenario 1-3

Jumlah sel	F1-score (%)	Akurasi (%)
4 x 4 sel	91,0256	95,5128
2 x 2 sel	93,5897	96.7949

Berdasarkan Tabel 6, pembagian jumlah sel mempengaruhi performansi akurasi sistem. Konfigurasi parameter terbaik jumlah sel adalah 2 x 2 sel dengan performansi f1-score 93,5897. Konfigurasi tersebut akan digunakan pada pengujian skenario selanjutnya.

8.1.4 Skenario 1-4

Skenario 1-4 bertujuan untuk mencari konfigurasi terbaik pembagian jumlah bin HOG. Konfigurasi yang di ujikan adalah 2 bin, 3 bin, 6 bin dan 9 bin. Ukuran 2 bin memiliki selisih 90 derajat (1-90, 91-180). Ukuran 3 bin memiliki selisih 60 derajat (1-60, 61-120, 121-180). Ukuran 6 bin memiliki selisih 30 derajat (1-30, 31-60, 61-90, dst), dan 9 bin memiliki selisih 20 derajat (1-20, 21-40, 41-60, dst). Parameter lain yang berkombinasi dengan skenario 1-3 dan di atur adalah antara lain jumlah blok = 4x4 blok, besar overlap = 0 %, jumlah sel = 2 x 2 sel, parameter *colour moment* yang dihitung *mean* dan standar deviasi dan fungsi kernel SVM adalah kernel linier.

Tabel 7 : hasil skenario 1-4

Jumlah bin	F1-score (%)	Akurasi (%)
2 bin	47,4359	73,718
3 bin	47,4359	73,718
6 bin	93,5897	96.7949
9 bin	93,5897	96.7949

Berdasarkan Tabel 7 pembagian jumlah bin mempengaruhi performansi sistem, dengan konfigurasi terbaik adalah dengan membagi jumlah bin sebanyak 6 atau 9, dengan performansi akurasi 93,5897 %. Konfigurasi tersebut akan digunakan pada pengujian skenario selanjutnya.

8.1.5 Skenario 2-1

Skenario 2-1 bertujuan untuk mencari konfigurasi terbaik dengan mencari nilai mean, standar deviasi, skewness dan kombinasi dari parameter-parameter tersebut pada *colour moment*. Parameter lain yang akan dikombinasikan dengan pengujian skenario 2-1 adalah antara lain jumlah blok = 8 x 8 blok, besar overlap = 0 %, jumlah sel = 2 x 2 sel, jumlah bin = 9, fungsi kernel adalah linear.

Tabel 8 : hasil skenario 2-1

Parameter yang dihitung	F1-score (%)	Akurasi (%)
<i>Mean</i>	93.5897	96.7949
Standar Deviasi	85.8974	92,949
<i>Skewness</i>	85.8974	92,949
<i>Mean</i> dan standar deviasi	93.5897	96.7949
<i>Mean</i> dan <i>skewness</i>	93.5897	96.7949

Standar deviasi dan <i>skewness</i>	93.5897	96.7949
<i>Mean</i> , standar deviasi dan <i>skewness</i>	93.5897	96.7949

Berdasarkan Tabel 8 perhitungan yang digunakan pada metode *colour moment* akan mempengaruhi performansi akurasi sistem konfigurasi terbaik adalah cukup menghitung nilai mean per-plane citra, atau dengan kombinasi masing-masing perhitungan, dengan hasil akurasi adalah 93.5897%. Konfigurasi tersebut akan digunakan pada pengujian skenario selanjutnya.

8.1.6 Skenario 3-1

Skenario 3-1 merupakan kumulatif performansi sistem keseluruhan deteksi dan klasifikasi rambu lalu lintas, menggunakan parameter-parameter terbaik yang telah diujikan pada skenario tahap sebelumnya. Parameter yang diujikan pada tahap ini adalah fungsi kernel pada SVM. Parameter lain yang berkombinasi dengan skenario 3-1 dan di atur adalah antara lain jumlah blok = 8 x 8 blok, besar overlap = 0 %, jumlah sel = 2 x 2 sel, jumlah bin = 9, serta parameter *colour moment* yang dihitung *mean* dan standar deviasi.

Tabel 9 : hasil skenario 3-1

Fungsi kernel	F1-score (%)	Akurasi (%)
Linear	93,5897	96.7949
Radial	67,9487	83.974
Polinomial	66,6667	83.333
Sigmoid	24,3590	62.179

Berdasarkan Tabel 9 jenis fungsi kernel yang digunakan akan mempengaruhi performansi akurasi sistem. Konfigurasi terbaik untuk parameter fungsi kernel adalah kernel linear, dengan akurasi sistem keseluruhan yang dihasilkan adalah 93,5897 %.

9. Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan, maka kesimpulan dari sistem klasifikasi rambu-rambu lalu lintas ini diantaranya adalah,

1. Konfigurasi parameter yang terbaik pada sistem deteksi dan klasifikasi rambu lalu lintas ini adalah dengan mengatur nilai parameter sebagai berikut, untuk parameter HOG, jumlah blok per citra adalah 8 x 8 blok, besar overlap subblok adalah 0%, ukuran sel adalah 2 x 2 sel, pembagian jumlah bin bisa dipilih dari 6 atau 9. Untuk parameter yang dihitung pada *colour moment* dapat menghitung nilai *mean* per plane citra, atau kombinasi dari *mean*, standar deviasi dan *skewness*. Sedangkan parameter fungsi kernel pada SVM digunakan fungsi kernel linear.
2. Dengan menggunakan hasil konfigurasi parameter terbaik di atas, performansi sistem secara keseluruhan menggunakan f1-score adalah 93,5897%. Performansi ini setara dengan akurasi sebesar 96.7949 %.

Adapun saran yang dapat diimplementasikan untuk perbaikan dan penyempurnaan sistem klasifikasi rambu-rambu lalu lintas ini antara lain,

1. Sistem dapat dikembangkan untuk bisa mengenal semua rambu lalu lintas dengan lebih detail, dengan jumlah data rambu lalu lintas pada setiap item yang lebih banyak.

Daftar Pustaka

- [1] R. T. K. Z. I. r. L. V. G. Victor Adrian Prisacariu, "Integrating Object detection with 3D Tracking Towards a Better Driver Assistance System," p. 1, 2010.
- [2] B. T. Navneet Dalal, "Histogram of Oriented Gradients for Human Detection".
- [3] I. R. M. A. USA, Biometric : Concepts, Methodologies, Tools, and Applications, Hersey PA, 2017.
- [4] H. Fleyeh, "Traffic and road Sign Recognition," *Digital Vetenskapliga Arkivet*, p. 81, 2008.
- [5] M. Shi, Road and Traffic Signs Recognition using Support Vector Machines, Sweden: Department of Computer Engineering, Dalarna University , 2006.

- [6] R. o. C. f. T. S. Recognition, “Lanlan Liu, Shuangdong Zhu,” *2006 6th International Conference on ITS Telecommunications* , p. 1, 2006.
- [7] I. K. E. P. M. Cahyo Permata, “Deteksi Mobil Menggunakan Histogram of Oriented Gradient,” no. Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS), pp. 2-3.
- [8] P. Milano, “Lecturer material : Visualisation Technique in data Mining,” dalam *Tecniche di Apprendimento Automatico per Applicazioni di Data Mining*, Milan, Politecnoco Milano, 2001, p. 4.
- [9] M. H. A. H. S. S. Safat B. Wali, “Comparative Survey on Traffic Sign Detection and Recognition,” <http://pe.org.pl/articles/2015/12/8.pdf>, p. 1, 2015.
- [10] Y. L. F. S. Huaping Liu, “Traffic Sign Using Group Sparce Coding,” *Elsevier*, p. 1, 2014.
- [11] J. Feng, *Traffic Sign Detection and Recognition System for Intelligent Vehicles*, Canada: University of Ottawa, 2014, p. 69.
- [12] Irawati, “Support Vector Machine,” dalam *OPTIMISASI PARAMETER SUPPORT VECTOR MACHINE (SVM) MENGGUNAKAN ALGORITME GENETIKA*, Bogor, Institut Pertanian Bogor, 2010, p. 3.

