

Implementasi Algoritma Speeded Up Robust Features (SURF) Pada Pengenalan Rambu – Rambu Lalu Lintas

Firma Firmansyah Adi^{#1}, Muhammad Ichwan^{*2}, Yusup Miftahuddin^{#3}

[#]Jurusan Teknik Informatika, Institut Teknologi Nasional
Jalan P.H.H. Mustofa No.23, Neglasari, Cibeunying Kaler, Kota Bandung

¹email.penulis1@domain.ekstensi

³yusupmiftahuddin@gmail.com

^{*}Jurusan Teknik Informatika, Institut Teknologi Nasional
Jalan P.H.H. Mustofa No.23, Neglasari, Cibeunying Kaler, Kota Bandung

²ichwan@itenas.ac.id

Abstract — One factor of increased violations on the highway is a violation of traffic signs, because the signs are not visible to the driver. In addition to this the conditions on road signs attached to the road have shortcomings such as twisting signs, imperfect beacons and non-standard beacons. So to be able to reduce the violation of traffic signs required a system that can recognize traffic. To be able to recognize traffic signs can be done visually and must be fast in recognizing. The image to be recognized can use the camera to retrieve information from signposts then the image is extracted with features of Speeded Up Robust Features (SURF) algorithm consisting of three stages: interest point detection, scale space, feature description and feature matching so that the system can recognize traffic signs. The research that has been done has resulted that the SURF algorithm in recognizing traffic signs is about 83,33% accurate to be the algorithm of introduction of traffic signs with the need to be fast and accurate. In addition, this algorithm is invariant to scale and invariant to rotation, so that the difference of slope and scale difference can still be recognized by using SURF algorithm.

Keywords— Traffic Sign, SURF, Kd-tree.

I. PENDAHULUAN

Pada saat mengemudikan kendaraan para pengemudi akan fokus terhadap keadaan jalan dan kearah kendaraan yang berada didepan, sehingga rambu – rambu yang terpasang disepanjang jalan tidak sengaja terhiraukan yang menyebabkan pelanggaran. Pelanggaran rambu – rambu lalu lintas merupakan salah satu faktor meningkatnya pelanggaran yang terjadi di jalan[1]. Salah satu solusi yang dapat mengurangi pelanggaran tersebut yaitu dengan membangun sistem yang dapat mengenali rambu – rambu lalu lintas untuk pengingat terhadap pengendara.

Rambu – rambu lalu lintas merupakan salah satu dari perlengkapan jalan, berupa lambang, huruf, angka,

kalimat dan/atau perpaduan diantaranya sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan[2]. Untuk mengurangi pelanggaran lalu lintas yang disebabkan rambu, maka diperlukan sistem yang dapat mengenali rambu – rambu lalu lintas. Rambu – rambu lalu lintas dapat dideteksi dan dikenali secara visual dengan menggunakan kamera sebagai media pengambilan informasi dari sebuah rambu – rambu lalu lintas. Tata letak pada rambu – rambu lalu lintas yang berbeda – beda dapat berpengaruh pada saat proses identifikasi, sehingga dalam melakukan identifikasi menggunakan metode Speeded Up Robust Features (SURF).

SURF atau Speeded Up Robust Features bertujuan untuk ekstraksi fitur dengan cara mendeteksi fitur lokal suatu citra dengan handal dan cepat. Algoritma ini sebagian terinspirasi oleh algoritma SIFT (Scale-Invariant Feature Transform) , terutama pada tahap scale space representation. Algoritma SURF menggunakan penggabungan algoritma citra integral (integral image) dan blob detection berdasarkan determinan dari matriks Hessian. Dalam implementasinya SURF memiliki empat tahapan yaitu Interest Point Detection Description dan Feature Matching[3].

II. LANDASAN TEORI

A. Tinjauan Pustaka

Terdapat penelitian – penelitian yang pernah dilakukan yang berhubungan dengan penelitian ini yaitu :

1. **Alexander A S Gunawan, Pascal Gerardus A, Wikaria Gazali** (2013), melakukan penelitian yang berjudul Pendeteksian Rambu Lalu Lintas Dengan Algoritma *Speeded Up Robust Features* (SURF), penelitian ini mendeskripsikan cara mendeteksi objek visual dengan kecepatan tinggi, dengan menggunakan algoritma *Speeded Up Robust Features* (SURF). Penelitian ini menghasilkan

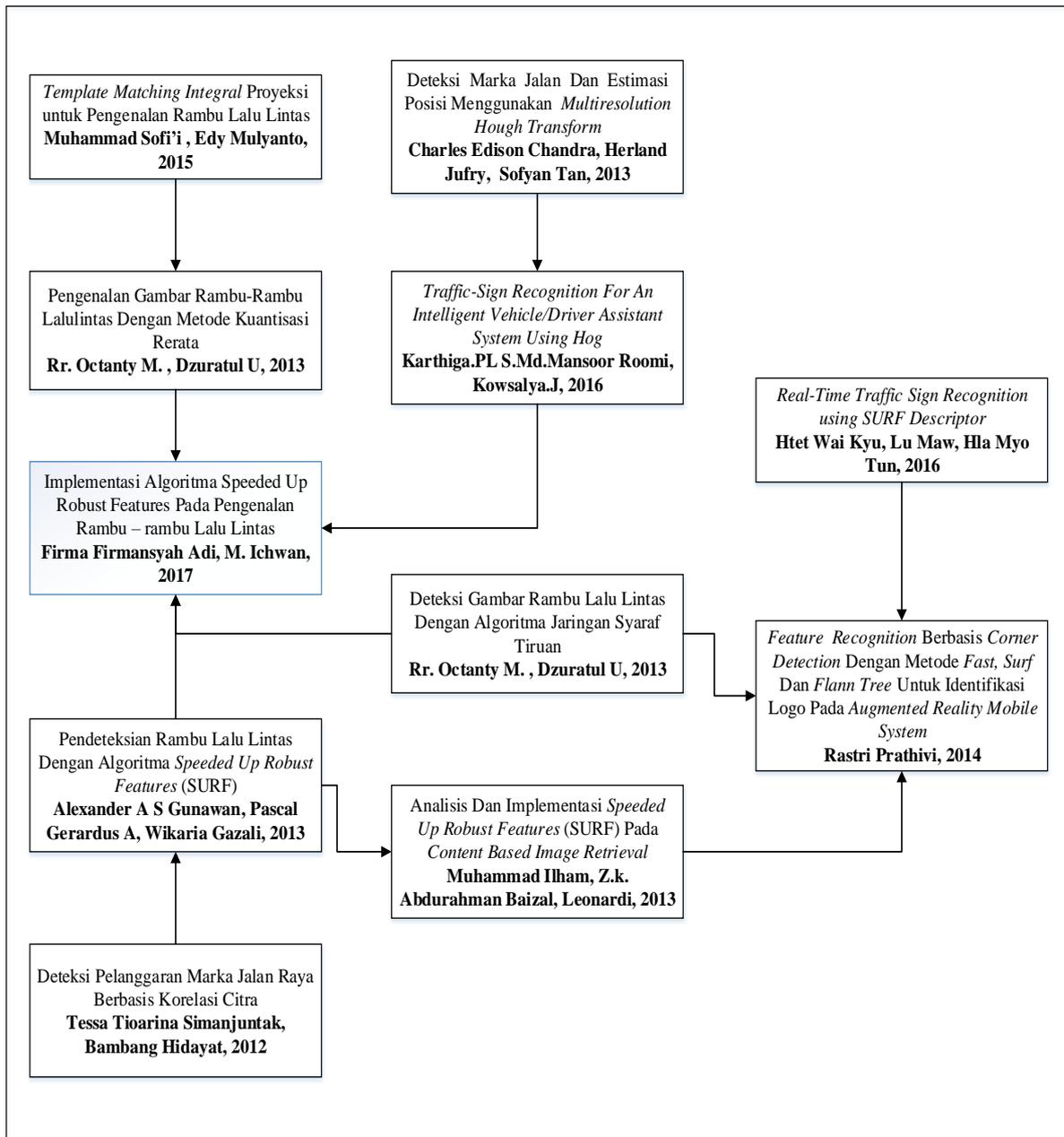
waktu pendeteksian rambu lalu lintas yang dapat dikenali dengan cukup cepat dengan waktu rata-rata pengenalan sebesar 101,2 milidetik. Selain itu algoritma SURF dapat mendeteksi dalam berbagai ukuran dan sudut yang ditolerir, dan bahkan memiliki kemampuan diskriminasi yang baik.

2. **Muhammad Ilham, Z.k. Abdurahman Baizal, Leonardi** (2013), melakukan penelitian yang berjudul Analisis Dan Implementasi *Speeded Up Robust Features* (SURF) Pada *Content Based Image Retrieval*. Tujuan dari penelitian ini adalah membuat sistem *Content Based Image Retrieval* dengan mengimplementasikan metode SURF dan untuk mengukur keakuratan sistem CBIR dalam memperoleh citra. Hasil dari penelitian ini adalah tingkat akurasi sistem *Speeded Up Robust Features* (SURF) tidak bergantung dengan objek gambar yang diberikan. Sistem akan menangkap image berdasarkan keseluruhan ciri *image* baik itu berupa bentuk, warna dan tekstur yang ada pada ciri suatu gambar
3. **Rastri Prathivi** (2014), melakukan penelitian yang berjudul *Feature Recognition Berbasis Corner Detection Dengan Metode Fast, Surf Dan Flann Tree Untuk Identifikasi Logo Pada Augmented Reality Mobile System*. pada penelitian ini dirancang dan dibangun aplikasi *augmented reality* pada perangkat *mobile* yang bertujuan untuk mengidentifikasi logo dengan menggunakan 3 metode yaitu FAST, SURF dan FLANN TREE. Setelah melakukan implementasi dan pengujian maka dihasilkan akurasi rata-rata 73 % untuk kondisi normal, 69% untuk kondisyscale invariant, 65% untuk kondisi rotation 90⁰ dan 66% untuk kondisi rotation 180⁰. Kontribusi terhadap penelitian yang dilakukan adalah nilai akurasi dari metode SURF diatas 60% yang terdiri dari beberapa sudut pengambilan, sehingga nilai akurasi SURF cukup tinggi meskipun dilakukan dalam berbagai sudut.
4. **Tri Harsono, Achmad Basuki, Nana Ramadijanti** (2016), melakukan penelitian yang

berjudul Pengenalan Gambar Rambu-Rambu Lalulintas Dengan Metode Kuantisasi Rerata yang membahas tentang pengenalan rambu – rambu lalu lintas dengan memanfaatkan ciri bentuk dengan menggunakan teknik kuantisasi rata – rata. Proses pengenalan dilakukan dengan menggunakan *template matching* antara vektor gambar rambu-rambu yang dimasukan (*vector query*) dengan semua tanda rambu-rambu yang ada dalam *database* (*vector template*). Hasil percobaan yang cukup baik pada saat ukuran sampling (*segment*) 4x4, didapatkan presentase *error* rata - rata sebesar 9.19% dengan performansi sebesar 90.81.

5. **Charles Edison Chandra, Herland Jufry, Sofyan Tan** (2013), melakukan penelitian yang berjudul Deteksi Marka Jalan Dan Estimasi Posisi Menggunakan *Multiresolution Hough Transform* yang menjelaskan mengenai membuat suatu algoritma untuk mendeteksi marka jalan serta mengestimasi posisi dan sudut dari kamera terhadap marka jalan yang ditemukan. Implementasinya menggunakan sebuah kamera untuk mengambil *sample* citra yang kemudian diproses menggunakan *Multiresolution Hough Transform* sebagai salah satu metode utama dalam pendeteksian marka jalan. Hasil *Computional Cost* yang didapatkan pada algoritma ini cukup lambat yaitu dengan waktu rata-rata yang dibutuhkan sebesar 4.329301 detik. Namun, algoritma ini memiliki kelebihan tersendiri yaitu tidak hanya bisa mendeteksi marka jalan tapi juga dapat mengestimasi posisi dan sudut kamera terhadap marka jalan dengan baik, dengan *error* posisi rata-rata sebesar 2.0520 cm dan *error* sudut rata-rata sebesar 1.35550.

Dari beberapa penelitian yang telah dilakukan memiliki kontribusi terhadap penelitian yang dilakukan, maka dapat diuraikan kedalam pemetaan tinjauan pustaka seperti pada gambar 1.



Gambar 1. Tinjauan Pustaka

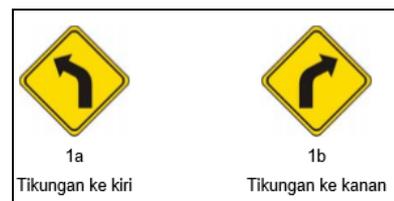
B. Rambu – rambu Lalu Lintas

Rambu – rambu lalu lintas adalah salah satu dari perlengkapan jalan, berupa lambang, huruf, angka, kalimat dan/atau perpaduan diantaranya sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan yang diatur dalam peraturan perundang – undangan[4]. Rambu rambu lalu lintas terdiri dari empat kategori yaitu :

1. Rambu Peringatan

Rambu peringatan digunakan untuk memberi peringatan kemungkinan ada bahaya atau tempat berbahaya

di depan pengguna. Warna dasar rambu peringatan berwarna kuning dengan lambang atau tulisan berwarna hitam seperti digambarkan pada gambar 2.



Gambar 2. Contoh rambu peringatan

2. Rambu Larangan

Rambu larangan menunjukkan perbuatan yang dilarang dilakukan oleh pemakai jalan. Warna dasar rambu larangan berwarna putih dan lambang atau tulisan berwarna hitam atau merah seperti yang digambarkan pada gambar 3.



Gambar 3. Contoh rambu larangan

3. Rambu Perintah

Rambu perintah menyatakan perintah yang wajib dilakukan oleh pemakai jalan. Rambu perintah berbentuk bundar berwarna biru dan lambang atau tulisan berwarna putih serta merah untuk garis serong sebagai batas akhir perintah seperti yang digambarkan pada gambar 4.



Gambar 4. Contoh rambu perintah

4. Rambu Petunjuk

Rambu petunjuk merupakan rambu pendahulu petunjuk jurusan, rambu petunjuk jurusan dan rambu penegas jurusan yang menyatakan petunjuk arah untuk mencapai tujuan antara lain kota, daerah/wilayah serta rambu yang menyatakan nama jalan dinyatakan dengan warna dasar hijau dengan lambang dan/atau tulisan warna putih seperti yang digambarkan pada gambar 5.



Gambar 5. Contoh rambu petunjuk

Untuk memenuhi kebutuhan adalah keseragaman bentuk dan ukuran rambu, desain rambu yang memenuhi standar dan mudah dipahami, lokasi rambu berhubungan dengan pengemudi sehingga pengemudi yang berjalan dengan kecepatan normal dapat memiliki waktu yang cukup untuk merespon, serta pemeliharaan rambu diperlukan agar rambu tetap terpelihara dengan baik.

C. Speeded Up Robust Features

SURF bertujuan untuk mendeteksi fitur lokal suatu citra dengan handal dan cepat. Algoritma ini sebagian terinspirasi oleh algoritma SIFT (*Scale-Invariant Feature Transform*). Algoritma SURF menggunakan penggabungan algoritma citra integral (*integral image*) dan *blob detection* berdasarkan determinan dari matriks Hessian. Seperti halnya SIFT algoritma ini juga bersifat *invariant scale* sehingga rotasi dan skala tidak terlalu berpengaruh[3]. Dalam implementasinya, algoritma SURF dibagi menjadi beberapa tahapan sebagai berikut:

1. Integral Image

Tahap awal dari algoritma SURF ini adalah mempersiapkan citra masukan. Citra masukan adalah citra dengan format *grayscale*. Pada tahap ini, citra dari hasil kamera direpresentasikan menjadi citra integral yang kemudian akan menghasilkan representasi citra[3].

2. Interest Point Detection

Deteksi titik perhatian (*interest point*) digunakan untuk memilih titik yang mengandung banyak informasi dan sekaligus stabil terhadap gangguan lokal atau global dalam citra digital[3]. Dalam algoritma SURF, dipilih detektor titik perhatian yang mempunyai sifat invarian terhadap skala, yaitu *blob detection*. Blob merupakan area pada citra digital yang memiliki sifat yang konstan atau bervariasi dalam kisaran tertentu. Untuk melakukan komputasi *blob detection* ini, digunakan determinan dari matriks Hessian (DoH) dari citra pada rumus 1 berikut ini.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

Rumus 1. Hessian Matriks

3. Feature Description

Fitur didefinisikan sebagai bagian yang mengandung banyak informasi suatu citra, dan fitur ini digunakan sebagai titik awal untuk algoritma deteksi objek^[3]. Tujuan dari proses deteksi fitur ini adalah untuk mendapatkan deskripsi dari fitur-fitur dalam citra yang diamati. Langkah pertama adalah melihat orientasi yang dominan pada titik perhatian yang terdapat dalam citra, kemudian membangun suatu area yang akan diambil nilainya dan mencari fitur korespondensi pada citra pembandingan

4. Feature Matching

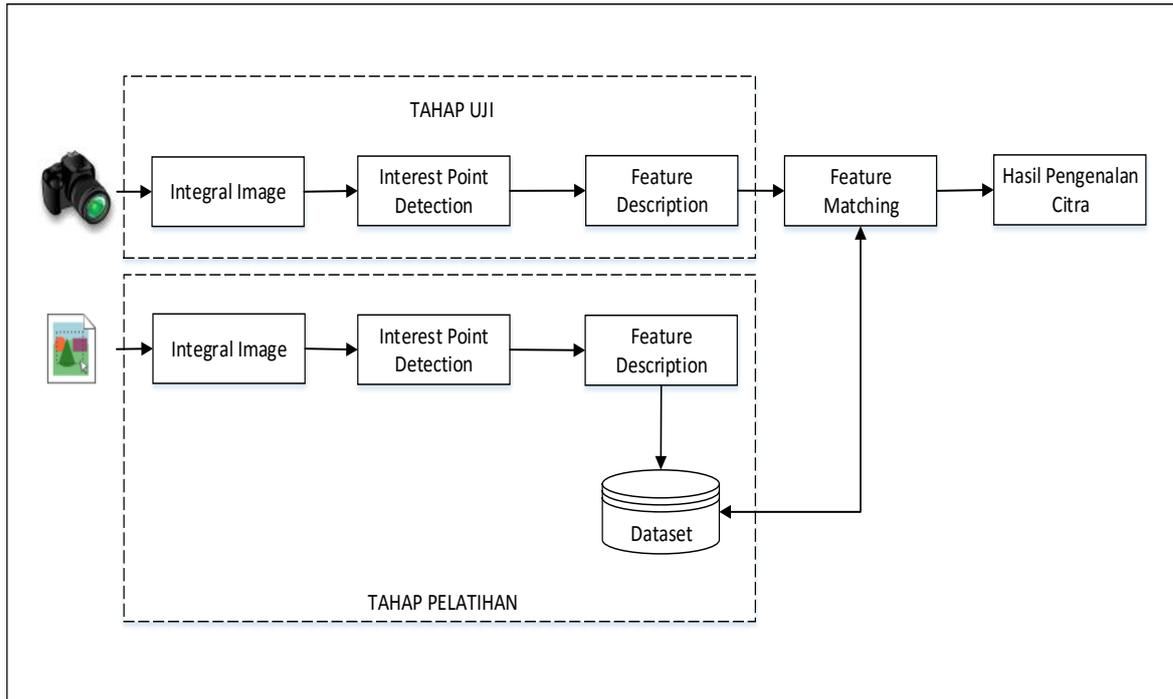
Dalam tahap ini yaitu membandingkan fitur hasil perhitungan proses sebelumnya tetapi hanya bila terdapat

perbedaan kontras, yang dideteksi melalui tanda dari trace matriks Hessian. Dengan cara ini, biaya komputasi dari algoritma SURF bisa dikatakan sangat minim [3]

III. METODE PENELITIAN

A. Gambaran Sistem

Sistem pengenalan rambu – rambu lalu lintas digambarkan kedalam blok diagram untuk mengetahui gambaran umum sistem yang dijelaskan pada gambar 5..



Gambar 6. Blok Diagram Sistem

Berdasarkan pada gambar 6 yang merupakan sistem yang dirancang, terdapat terdapat 3 tahapan utama yaitu :

1. Proses awal, merubah format RGB menjadi Grayscale dengan menghitung nilai representasi citra integral (integral image).
2. Mendeteksi titik-titik fitur, mencari fitur-fitur blob-like pada citra dengan menggunakan matriks Hessian
 - a. Membentuk piramid citra: melakukan konvolusi tapis kotak dengan ukuran yang semakin besar dengan citra untuk membentuk piramid citra
 - b. Mencari ekstrema dari determinan matriks Hessian: menghitung nilai dari determinan matriks Hessian kemudian mencari nilai ekstremanya (nilai maksima atau minima dibandingkan dengan nilai-nilai tetangganya)
 - c. Lokalisasi calon fitur: mencari lokasi calon fitur pada setiap ruang skala (scale space) dengan

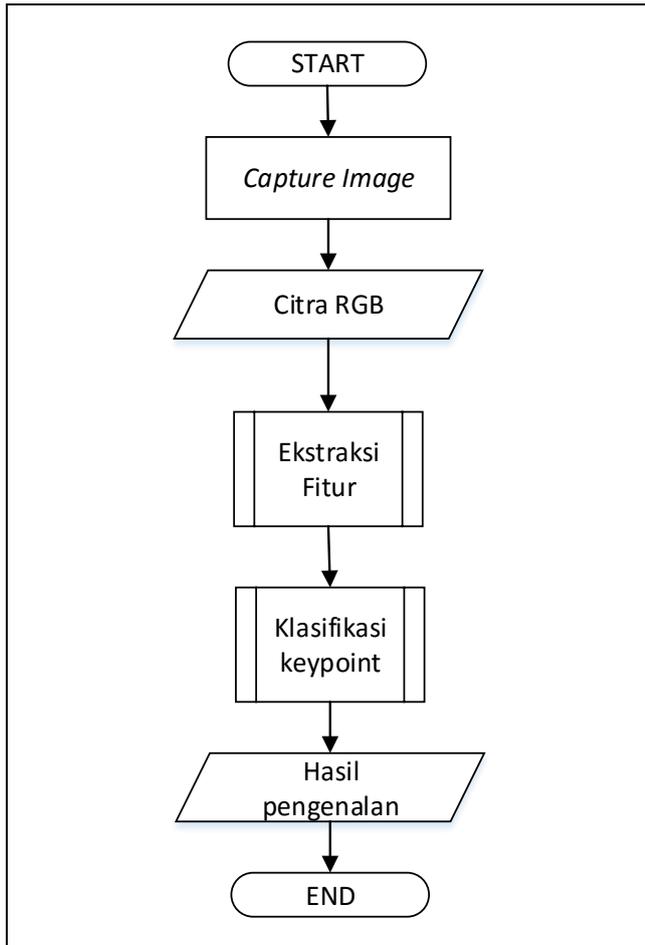
3. Pendeskripsian fitur, mendeskripsikan titik-titik fitur yang sudah di deteksi
4. Pencocokan fitur, yaitu mencari nilai kemiripan dari setiap titik titik yang menarik antara citra uji dan citra latih dengan menggunakan FLANN, yaitu dengan mencari nilai tetangga dari keypoint yang memiliki nilai kemiripan dengan menghasilkan nilai NNDR(*Nearest Neighbour Distance Ratio*) sehingga dari nilai kemiripan tersebut sistem dapat mengenali rambu – rambu lalu lintas

B. Flow Chart Sistem

Alur Kerja Sistem bertujuan untuk menampilkan urutan atau langkah – langkah dari sebuah sistem. Alur

kerja yang dijelaskan merupakan sistem yang bekerja pada sistem sehingga dapat mengenali rambu – rambu lalu lintas yang digambarkan secara keseluruhan pada Gambar 7

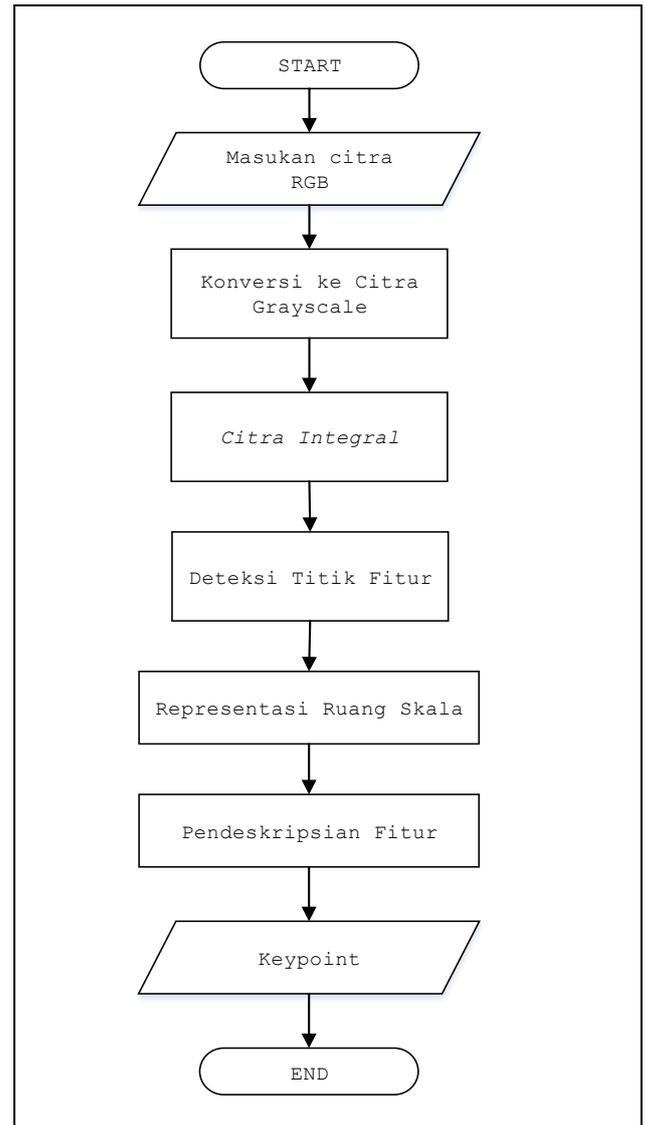
Sistem dimulai dengan menampilkan menu utama kemudian menghubungkan kamera dengan sistem, jika kamera terhubung maka menampilkan tampilan kamera dan mengaktifkan tombol *capture*. Setelah meng*capture image* langkah selanjutnya adalah melakukan ekstraksi fitur pada citra tersebut yang menghasilkan *keypoint* yang berguna pada tahap selanjutnya untuk melakukan klasifikasi citra.



Gambar 7. Flowchart Sistem

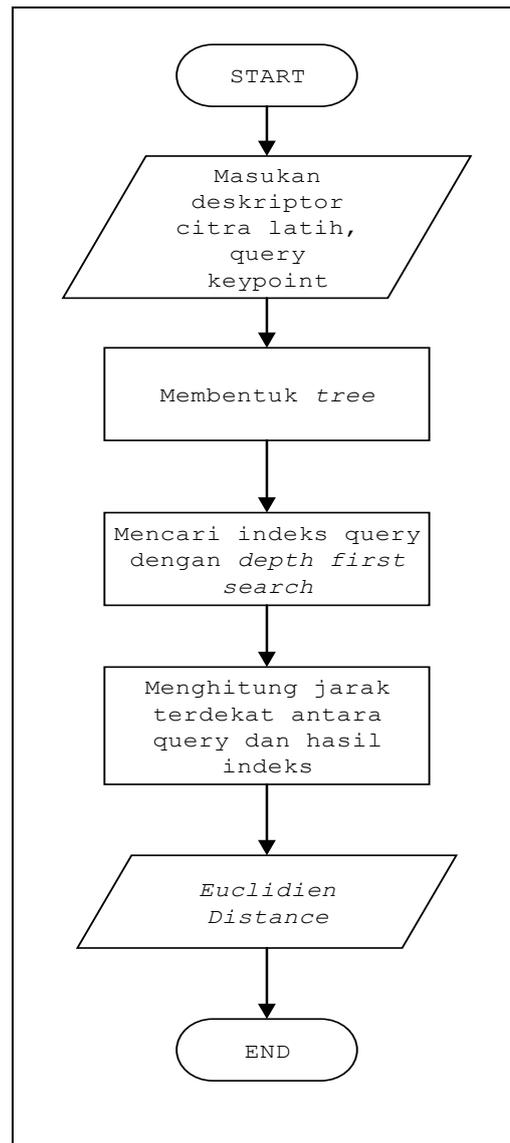
Pada alur kerja yang digambarkan pada gambar 7 terdapat 2 sub proses yaitu ekstraksi fitur dimana menggunakan algoritma surf dan klasifikasi keypoint yang akan diuraikan pada Gambar 8 dan Gambar 9.

Pada ekstraksi fitur menggunakan algoritma SURF yang bertujuan untuk menghasilkan nilai – nilai keypoint yang dapat digunakan untuk pengklasifikasian pada tahap selanjutnya.



Gambar 8. Flowchart Ekstraksi Fitur

Tahap awal ekstraksi fitur adalah dengan mengkonversi citra menjadi citra *grayscale* yang tujuannya adalah untuk menghitung citra integral yang bertujuan untuk mempercepat proses komputasi. Kemudian langkah selanjutnya adalah dengan mengimplementasikan algoritma SURF yang terdiri dari deteksi titik fitur, representasi ruang skala dan pendeskripsian fitur.



Gambar 9. Flowchart Klasifikasi

Klasifikasi *keypoint* hasil ekstraksi fitur berguna untuk mengenali citra uji rambu – rambu lalu lintas, tahap ini menggunakan metode *kd-tree* dimana metode ini menggunakan *binary tree* yang kemudian mencari nilai terdekat dengan menggunakan *euclidian distance*.

IV. HASIL DAN PEMBAHASAN

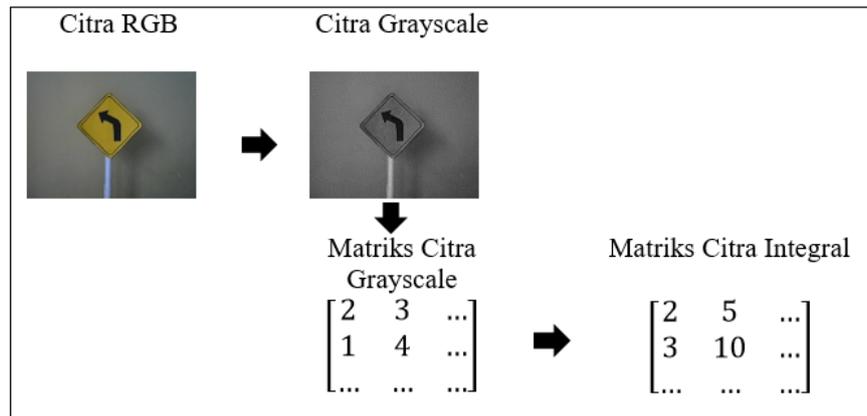
A. Implementasi

Implementasi pada sistem pengenalan rambu – rambu lalu lintas menggunakan bahasa pemrograman java dan openCV. Pada tahap implementasi terdiri dari implementasi ekstraksi fitur, implementasi klasifikasi dan implementasi system

1) Implementasi Ekstraksi Fitur

Pada tahap implementasi ekstraksi ciri dengan menggunakan algoritma SURF yang bertujuan untuk mencari point – point yang menarik dengan menggunakan openCV. Proses ekstraksi pada algoritma SURF ini terdiri dari 3 tahapan yaitu, citra integral, deteksi titik fitur dan pendeskripsian fitur

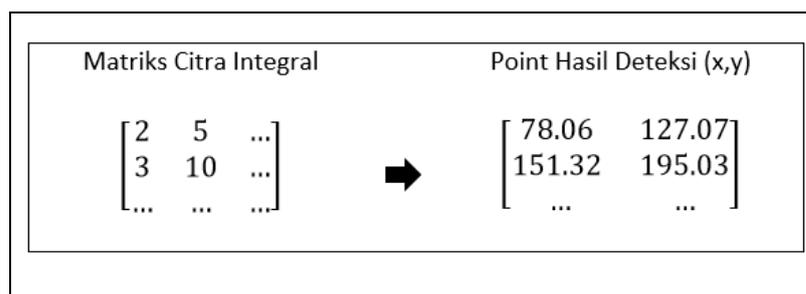
a. Implementasi citra integral yaitu dengan menggunakan citra grayscale yang kemudian dihitung integral citra nya dengan menambahkan pixel kiri dan pixel atas dari suatu citra. Proses pada tahapan ini digambarkan pada gambar 10.



Gambar 10. Integral Image

b. *Interest Point Detection* menggunakan fungsi `featureDetector.detect(param1, param2)`; dimana parameter 1 merupakan matriks citra yang akan di deteksi berupa nilai masukan pada fungsi ini dan

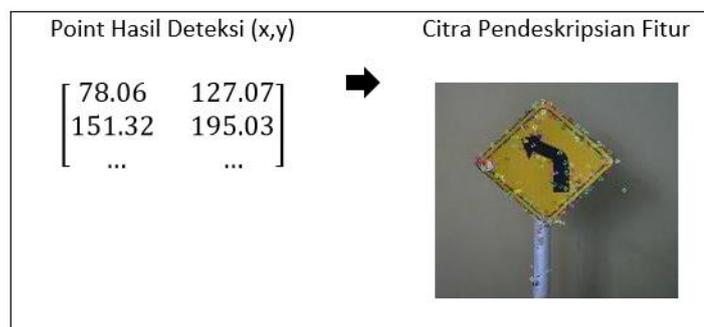
parameter 2 merupakan variabel untuk menyimpan list hasil deteksi berupa output yang dihasilkan yang digambarkan pada gambar 11.



Gambar 11. Interest Point Detection

Feature descriptor menggunakan fungsi `descriptorExtractor.compute(param1,param2,param3)`; dimana parameter 1 (nilai masukan) adalah matriks citra yang akan dideteksi, parameter 2((nilai masukan)) adalah nilai keypoint hasil dari tahapan *interest point*

detection dan parameter 3 adalah output pada fungsi ini berupa list dari keypoint. Setelah point yang menarik diketahui maka dilakukan *draw* pada citra uji berdasarkan point – point yang telah dihasilkan. Tahapan ini digambarkan pada gambar 12

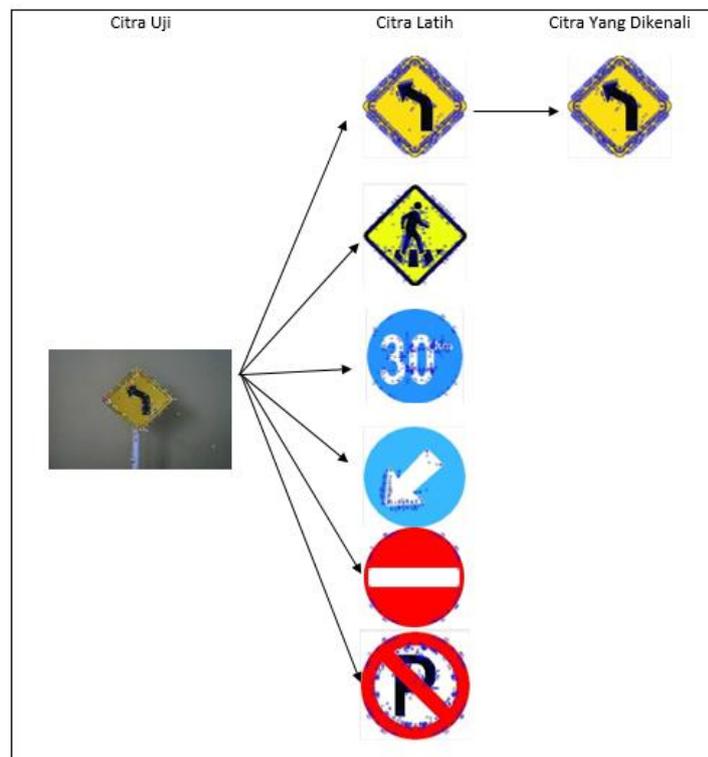


Gambar 12. Feature Description

c. *Feature Matching*, pada tahap ini dilakukan matching pada citra yang diuji dan citra latih. Karena pada sistem ini bersifat pengenalan maka citra latih yang diperlukan adalah data latih yang lebih dari satu, maka untuk kebutuhan tersebut pada proses *feature matching* ditambahkan tahapan klasifikasi pada keypoint hasil deteksi untuk dapat mengenali citra tersebut. Tahap ini merupakan proses pengenalan dengan menggunakan k-dimensional tree dengan mencari nilai tetangga terdekat, dalam sistem ini yang terdapat dalam algoritma SURF pada tahap feature matching dengan menggunakan fungsi sebagai berikut :

`descriptorMatcher.knnMatch(objectDescriptors, sceneDescriptors, matches, 2)`; terdapat tiga parameter yang memiliki tujuan sebagai berikut :

1. Parameter 1 adalah nilai dari deskriptor hasil dari feature deskriptor pada tahap ekstraksi fitur berupa nilai keypoint dari data latih.
2. Parameter 2 adalah nilai dari deskriptor hasil dari feature deskriptor pada tahap ekstraksi fitur berupa nilai keypoint dari data uji.
3. Parameter 3 adalah variabel untuk menyimpan hasil dari proses matching deskriptor 1 dan deskriptor 2 yang menghasilkan index dari keypoint data latih dan index dari keypoint data uji yang memiliki kemiripan, serta menghasilkan nilai NNDR (*Nearest Neighbour Distance Ratio*).
4. Parameter 4 adalah nilai *k* berupa nilai jumlah tetangga terdekat yang akan disimpan.



Gambar 13. Klasifikasi

Pada tahap ini yang digambarkan pada gambar 13, citra uji dibandingkan dengan semua citra latih dimana nilai yang dibandingkan adalah nilai keypoint antara data uji dan data latih sehingga akan menghasilkan nilai tetangga terdekat yang disebut

dengan NNDR (*Nearest Neighbour Distance Ratio*). Sehingga nilai terdekat tersebut dijadikan dasar untuk menentukan kemiripan dari citra yang diuji.

pengenalan dari setiap rambu dapat dihitung dengan menggunakan persamaan :

$$Accuracy = \frac{\sum_{successfull}}{\sum_{recognizable}} 100\%$$

a) Pengujian Skala 640x480

Pengujian dilakukan dengan menggunakan citra masukan dengan ukuran 640x480 yang dilakukan rotasi sebanyak 5 kali yaitu 90°, 75°, 60°, 45° dan 30°.

TABEL I
PENGUJIAN SISTEM SKALA 640x480

No	Citra Uji	Kemiringan	Jumlah Keypoint yang mirip	Waktu (ms)	Hasil
1		0°	136	659	✓
2		75°	46	1415	✓
3		60°	19	1949	✓
4		45°	25	772	✓
5		30°	15	822	✓
6		0°	73	762	✓
7		75°	48	1178	✓
8		60°	42	1316	✓
9		45°	57	1319	✓
10		30°	55	1635	✓
11		0°	18	451	✓
12		75°	20	914	✓
13		60°	7	721	✓
14		45°	19	763	✓
15		30°	18	1334	✓
16		0°	62	1092	✓
17		75°	39	1353	✓
18		60°	28	1809	✓
19		45°	32	619	✓
20		30°	25	708	✓
21		0°	16	1421	X
22		75°	27	1981	X
23		60°	7	2734	X
24		45°	9	2291	X
25		30°	9	641	X
26		0°	76	754	✓
27		75°	51	2500	✓
28		60°	43	1024	✓
29		45°	29	954	✓
30		30°	38	1959	✓
Jumlah rambu yang dikenali				25	
Rata – rata waktu pengenalan(ms)				1261,66	
Rata – rata keypoint yang mirip				36,3	

Pada pengujian yang ditampilkan pada Tabel I dengan menggunakan skala 640x480 didapatkan nilai akurasi dari sistem dengan cara perhitungan sebagai berikut :

$$Accuracy = \frac{\sum_{rambu\ yang\ dikenali}}{\sum_{rambu\ yang\ diuji}} 100\%$$

$$Accuracy = \frac{25}{30} 100\%$$

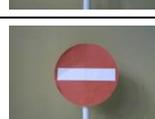
$$Accuracy = 83,33\%$$

Berdasarkan pada pengujian nilai akurasi pada sistem pengenalan rambu – rambu lalu lintas sebesar 83,33% pada skala 640x480 yang dijelaskan pada Tabel I dengan rata – rata keypoint yang cocok adalah sejumlah 40,84 keypoint dengan jumlah keypoint yang cocok terkecil sebesar 7 keypoint dan jumlah keypoint terbesar adalah 136 keypoint. Rata – rata waktu pengenalan selama 1261,66 ms atau 1,261 detik dengan waktu pengenalan tercepat adalah 451 ms atau 0,451 detik dan waktu pengenalan terlama adalah 2500 ms atau 2,5 detik.

a) Pengujian Skala 800x600

Pengujian dilakukan dengan menggunakan citra masukan dengan ukuran 640x480 yang dilakukan rotasi sebanyak 5 kali yaitu 90°, 75°, 60°, 45° dan 30°.

TABEL III
PENGUJIAN SISTEM SKALA 800x600

No	Citra Uji	Kemiringan	Jumlah Keypoint yang mirip	Waktu (ms)	Hasil
1		0°	119	938	✓
2		75°	34	1657	✓
3		60°	27	1819	✓
4		45°	23	2276	✓
5		30°	27	762	✓
6		0°	71	694	✓
7		75°	52	2099	✓
8		60°	48	1309	✓
9		45°	46	541	✓
10		30°	61	832	✓
11		0°	14	557	✓
12		75°	15	383	✓
13		60°	10	356	✓
14		45°	12	1032	✓
15		30°	18	375	✓
16		0°	68	544	✓
17		75°	30	427	✓
18		60°	32	1238	✓
19		45°	30	476	✓
20		30°	39	1134	✓
21		0°	15	444	X
22		75°	23	1033	X
23		60°	7	1821	X
24		45°	8	1228	X
25		30°	9	542	X
26		0°	64	850	✓
27		75°	53	580	✓
28		60°	58	599	✓
29		45°	40	628	✓

No	Citra Uji	Kemiringan	Jumlah Keypoint yang mirip	Waktu (ms)	Hasil
30		30°	49	1599	√
Jumlah rambu yang dikenali				25	
Rata – rata waktu pengenalan(ms)				959,1	
Rata – rata keypoint yang mirip				36,73	

Pada pengujian yang ditampilkan pada Tabel II dengan menggunakan skala 800x600 didapatkan nilai akurasi dari sistem dengan cara perhitungan sebagai berikut :

$$\text{Accuracy} = \frac{\sum \text{rambu yang dikenali}}{\sum \text{rambu yang diuji}} 100\%$$

$$\text{Accuracy} = \frac{25}{30} 100\%$$

$$\text{Accuracy} = 83,33\%$$

Berdasarkan pada pengujian nilai akurasi pada sistem pengenalan rambu – rambu lalu lintas sebesar 83,33% pada skala 800x600 yang dijelaskan pada Tabel II dengan rata – rata keypoint yang cocok adalah sejumlah 36,73 keypoint dengan jumlah keypoint yang cocok terkecil sebesar 7 keypoint dan jumlah keypoint terbesar adalah 136 keypoint. Rata – rata waktu pengenalan selama 959,1 ms atau 0,959 detik dengan waktu pengenalan tercepat adalah 451 ms atau 0,451 detik dan waktu pengenalan terlama adalah 2500 ms atau 2,5 detik.

b) Pengujian Skala 1024x720

Pengujian dilakukan dengan menggunakan citra masukan dengan ukuran 640x480 yang dilakukan rotasi sebanyak 5 kali yaitu 90°, 75°, 60°, 45° dan 30°.

TABEL IIIII
PENGUJIAN SISTEM SKALA 1024X720

No	Citra Uji	Kemiringan	Jumlah Keypoint yang mirip	Waktu (ms)	Hasil
1		0°	15	1188	√
2		75°	84	1227	√
3		60°	87	1544	√
4		45°	106	1443	√
5		30°	119	2237	√
6		0°	67	1236	√
7		75°	132	1426	√
8		60°	44	1222	√
9		45°	54	1433	√
10		30°	53	2203	√

No	Citra Uji	Kemiringan	Jumlah Keypoint yang mirip	Waktu (ms)	Hasil
11		0°	15	1376	√
12		75°	15	1547	√
13		60°	65	1331	√
14		45°	45	1211	√
15		30°	57	1112	√
16		0°	53	1476	√
17		75°	21	1604	√
18		60°	27	1401	√
19		45°	19	1208	√
20		30°	29	1565	√
21		0°	2	2212	X
22		75°	5	1231	X
23		60°	6	1233	X
24		45°	23	2211	X
25		30°	26	2313	X
26		0°	50	1487	√
27		75°	41	1654	√
28		60°	29	2046	√
29		45°	58	1433	√
30		30°	42	1121	√
Jumlah rambu yang dikenali				25	
Rata – rata waktu pengenalan(ms)				1531,033	
Rata – rata keypoint yang mirip				46,3	

Pada pengujian dengan menggunakan skala 1024x720 didapatkan nilai akurasi dari sistem dengan cara perhitungan sebagai berikut :

$$\text{Accuracy} = \frac{\sum \text{rambu yang dikenali}}{\sum \text{rambu yang diuji}} 100\%$$

$$\text{Accuracy} = \frac{25}{30} 100\%$$

$$\text{Accuracy} = 83,33\%$$

Berdasarkan pada pengujian nilai akurasi pada sistem pengenalan rambu – rambu lalu lintas sebesar 83,33% pada skala 1024x720 yang dijelaskan pada Tabel III dengan rata – rata keypoint yang cocok adalah sejumlah 46,3 keypoint dengan jumlah keypoint yang cocok terkecil sebesar 2 keypoint dan jumlah keypoint terbesar adalah 136 keypoint. Rata – rata waktu pengenalan selama 1531,033 ms atau 1,531 detik dengan waktu pengenalan tercepat adalah 1112 ms atau 1,112 detik dan waktu pengenalan terlama adalah 2313 ms atau 2,313 detik.

c) Hasil Perhitungan Pengujian

Hasil perhitungan pengujian didapatkan dari pengujian dengan menggunakan, sehingga untuk hasil perhitungan akurasi sistem secara keseluruhan dapat dihitung dengan mengambil nilai rata – rata dari nilai akurasi kamera 5MP dan kamera VGA.

$$\text{Accuracy} = \frac{83,33 + 83,33 + 83,33}{3}$$

$$\text{Accuracy} = 83,33\%$$

Berdasarkan dari pengujian skala optimal adalah skala 800x600 yang memerlukan waktu tercepat antara ketiga skala, skala makin besar maka nilai jumlah *keypoint* yang cocok semakin besar. Namun nilai *keypoint* dipengaruhi juga oleh cahaya pada citra seperti dari perbandingan citra dibawah ini :

TABEL IVV
PERBANDINGAN CITRA

640x480	800x600	1024x720
		

Terlihat seperti pada Tabel IV bahwa pada pengujian skala 800x600 citra yang dihasilkan adalah citra yang memiliki cahaya yang cukup jika dibandingkan dengan skala 640x480 dan skala 1024x720, sehingga dapat disimpulkan cahaya pada saat pengambilan gambar berpengaruh terhadap waktu dan jumlah *keypoint* yang cocok.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dengan mengimplementasikan metode SURF untuk melakukan pengenalan rambu – rambu lalu lintas maka dapat disimpulkan bahwa algoritma SURF cukup baik untuk mengenali rambu – rambu lalu lintas dengan nilai akurasi sistem pengenalan rambu – rambu lalu lintas sebesar 83,33 % pada hasil pengujian yang dilakukan dengan menggunakan skala 640x480, 800 x600 dan 1024 x720 dengan rotasi 90°, 75 °, 60 °, 45 ° dan 30 ° yang menghasilkan rata – rata waktu pengenalan adalah 1250,6 ms atau 1,2506 detik dengan jumlah *keypoint* rata – rata adalah 39,77 dimana cahaya pada saat pengambilan citra

mempengaruhi waktu dan jumlah dari *keypoint* yang cocok. Sehingga dengan waktu pengenalan sekitar 1,25 detik memiliki kemampuan yang cepat untuk mendeteksi rambu – rambu lalu lintas dengan memiliki kemampuan untuk mendeteksi rambu pada sudut – sudut tertentu.

DAFTAR PUSTAKA

- [1] Sadono, Soni. Budaya Tertib Berlalu Lintas “Kajian Fenomenologis Atas Masyarakat Pengendara Sepeda Motor Di Kota Bandung”.2016.
- [2] Kementrian Perhubungan. *Peraturan Menteri Perhubungan Republik Indonesia Nomor Pm 13 Tahun 2014*. Kementrian Perhubungan Republik Indonesia. 2014
- [3] Lowe, D. G. 1999 *Object Recognition From Local Scale-Invariant Features*. Proceeding of the International Conference on Computer Vision, Corfu Sept
- [4] Zhou, Huiyu., Wu, Jiahua. *Digital Image Processing Part 1*, Ventus Publishing London. 2010
- [5] Affandi Faisal. 2012. Rambu – rambu Lalu Lintas Di Indonesia. [Online] (<https://www.scribd.com/doc/7554094/rambu-lalu-lintas-di-Indonesia>)
- [6] Simanjuntak, T Tessa., Hidayat, Bambang., Atmaja Dwi, Ratri. *Deteksi Pelanggaran Marka Jalan Raya Berbasis Korelasi Citra Teknik Telekomunikasi*, Fakultas Teknik Elektro, Universitas Telkom.2012.
- [7] Sofi'i, Muhammad., Mulyanto, Edy. *Template Matching Integral Proyeksi untuk Pengenalan Rambu Lalu Lintas*. Jurusan Teknik Informatika, Universitas Dian Nuswantoro Semarang. 2015
- [8] Yannis, G. and Antoniou, C. State-of-the-art on advanced driver assistance systems. Workshop on “The role of Advanced Driver Assistance Systems on traffic safety and efficiency”, 4 – 18, National University of Athens. 2000
- [9] Kyu, Htet Wai., Maw, Lu., Tun, Hla Myo. *Real-Time Traffic Sign Recognition using SURF Descriptor*. International Journal Of Scientific & Technology Research. 2016
- [10] Laganiere, Robert. *OpenCV 2 Computer Vision Application Programming Cookbook*, Packt Publishing, Birmingham. 2011
- [11] M. Shneier. *Road Sign Detection And Recognition*, IEEE Computer Society International Conference on Computer Vision and Pattern Recognition. 2005
- [12] Petter, M. Stefan. *Real-time System*. Department of Communications, Information Technology and the Art, Australian Research Council. 2008