

APLIKASI PENGENALAN RAMBU LALU LINTAS MENGGUNAKAN METODE *SPEEDED UP ROBUST FEATURES (SURF)* DAN *SUPPORT VECTOR MACHINE* (SVM) BERBASIS *ANDROID*

TRAFFIC SIGN RECOGNITION APPLICATION USING SPEEDED UP ROBUST FEATURES (SURF) AND SUPPORT VECTOR MACHINE (SVM) BASED ON ANDROID

Nizar Akbar Arifin¹, Budhi Irawan, S.Si.,M.T², Casi Setianingsih, S.T.,M.T³

Prodi S1 Sistem Komputer, Fakultas Teknik, Universitas Telkom^{1,2,3}

nizarakbararifin@yahoo.com, budhiirawan@telkomuniversity.ac.id, setiacasie@telkomuniversity.ac.id

Abstrak

Rambu lalu lintas merupakan salah satu peraturan yang dapat berupa lambang maupun gambar yang berfungsi sebagai peringatan maupun petunjuk bagi tiap orang. Banyak wisatawan dalam negeri maupun luar negeri yang masih belum paham apa arti dari rambu lalu lintas yang berhubungan dengan pariwisata. Dengan memanfaatkan *Intelligent System and Machine Learning*, dirancang sebuah aplikasi mobile yang berfungsi dapat mengenali gambar rambu lalu lintas berbasis *android* menggunakan kamera *smartphone*. Dengan aplikasi ini, wisatawan dalam negeri maupun luar negeri dapat mengambil gambar rambu lalu lintas yang ada, lalu secara otomatis jika rambu lalu lintas itu terdeteksi, maka aplikasi akan menampilkan sebuah teks informasi berupa arti dari sebuah rambu lalu lintas tersebut dengan 2 bahasa, yaitu bahasa Indonesia dan bahasa Inggris. Dalam implementasi aplikasi ini menggunakan metode ekstraksi yaitu *Speeded Up Robust Features (SURF)* untuk mengenali pola suatu gambar. Dengan SURF ini dapat melakukan sebuah ekstraksi gambar yang diambil dan kemudian data yang di ekstraksi akan di klasifikasi berdasarkan metode *Support Vector Machine (SVM)* yang selanjutnya disimpan di *database*. Aplikasi ini bertujuan agar wisatawan dalam negeri maupun luar negeri paham arti rambu lalu lintas yang dipasang di objek wisata.

Kata Kunci : Rambu Lalu Lintas, *Android*, Metode *Speeded Up Robust Features (SURF)*, Metode *Support Vector Machine (SVM)*

Abstract

Traffic sign is one of the rules that may be emblems and pictures that serve as warnings or instructions for each person. Many tourists still do not understand what the meaning of traffic signs related to tourism. By utilizing the *Intelligent System and Machine Learning*, designed a mobile application that serves to recognize traffic signs picture using the camera android based smartphone. With this application, tourists can take pictures of traffic signs are there, then automatically if a traffic sign is detected, then the application will display a text information such as the meaning of a traffic sign with two languages; Indonesian and English. In the implementation of this application using the extraction method that is *Speeded Up Robust Features (SURF)* to identify the pattern of an image. With this SURF can perform an extraction picture taken and then the data that is extracted will be in the classification based on the method of *Support Vector Machine (SVM)* which are stored in the database. This application for many tourists understand the meaning of traffic signs installed in the attraction.

Keyword : Traffic Sign, *Android*, *Speeded Up Robust Features (SURF Method)*, *Support Vector Machine (SVM) Method*.

1. Pendahuluan

1.1 Latar Belakang

Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang dapat berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan [1]. Untuk mematuhi rambu lalu lintas tersebut, terlebih dahulu kita harus memahami dan mengetahui arti dari rambu-rambu lalu lintas [1].

Dengan banyaknya wisatawan dalam negeri maupun luar negeri yang belum mengerti arti rambu-rambu lalu lintas yang dipasang di objek wisata, maka dalam tugas akhir ini memanfaatkan sebuah teknologi intelligent system dan aplikasi mobile untuk menerjemahkan arti gambar rambu lalu lintas yang tersedia dalam 2 bahasa, yaitu bahasa Inggris dan bahasa Indonesia.

Dalam aplikasi mobile berbasis android ini, wisatawan dalam negeri maupun luar negeri ini dapat mengambil gambar rambu lalu lintas yang ada dengan membuka gallery atau membuka kamera smartphone yang berbasis android untuk diterjemahkan dalam bentuk bahasa Inggris maupun bahasa Indonesia. Untuk mengenali gambar rambu lalu lintas tersebut, dibutuhkan sebuah pola pengenalan gambar dengan sebuah metode ekstraksi ciri yaitu *Speeded Up Robust Features (SURF)* dengan melakukan pengolahan ekstraksi ciri pada sebuah gambar yang diambil dari kamera smartphone untuk mencari sebuah ciri khas pada suatu citra tersebut. Setelah itu menggunakan metode *Support Vector Machine (SVM)* sebagai klasifikasi dan *machine learning* di aplikasi ini untuk mencari citra yang *match* dengan di data *training*. Dengan adanya aplikasi mobile berbasis android ini dapat memudahkan wisatawan dalam negeri maupun luar negeri mengerti dan memahami rambu lalu lintas yang dipasang di bidang pariwisata.

1.2 Rumusan Masalah

Permasalahan yang diangkat dalam pembuatan tugas akhir ini adalah sebagai berikut :

1. Bagaimana cara membantu wisatawan untuk mengetahui arti rambu lalu lintas yang berhubungan dengan pariwisata ?
2. Bagaimana cara kerja aplikasi untuk mengenali rambu lalu lintas yang ada di objek wisata ?
3. Bagaimana cara mengenali pola gambar rambu lalu lintas dengan metode ekstraksi *Speeded Up Robust Features (SURF)* di *android* ?
4. Bagaimana performansi metode *Speeded Up Robust Features (SURF)* di *android* ?

1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Membuat aplikasi berbasis android untuk membantu wisatawan mengetahui arti rambu lalu lintas yang berhubungan dengan pariwisata.
2. Mengetahui cara kerja aplikasi pengenalan rambu lalu lintas yang berhubungan dengan pariwisata.
3. Mengetahui dan mengimplementasikan cara kerja aplikasi dengan metode ekstraksi ciri *Speeded Up Robust Features (SURF)* pada pola gambar rambu lalu lintas.
4. Melakukan pengujian performansi metode ekstraksi *Speeded Up Robust Features (SURF)*.

1.4 Batasan Masalah

Batasan permasalahan dari aplikasi pengenalan rambu lalu lintas ini adalah sebagai berikut :

1. Aplikasi pengenalan rambu lalu lintas ini dijalankan pada smartphone berbasis android minimal versi 4.
2. Jumlah data training pola rambu lalu lintas yang dipakai hanya rambu lalu lintas yang berhubungan dengan pariwisata sebanyak 25 pola rambu.
3. *Tools* yang digunakan yaitu Android Studio.
4. Gambar yang diolah hanya dalam bentuk format Portable Network Graphics (PNG).
5. Aplikasi ini tidak terhubung dengan internet (offline).
6. Proses ekstraksi ciri menggunakan *Speeded-Up Robust Features*.
7. Proses klasifikasi menggunakan *Support Vector Machine*.
8. Keluaran dari aplikasi pengenalan rambu lalu lintas ini berupa teks informasi yang tersedia dalam 2 bahasa, yaitu bahasa Indonesia dan bahasa Inggris.
9. Mengambil gambar rambu lalu lintas harus dalam kondisi di luar ruangan (outdoor).

10. Rambu lalu lintas yang diambil harus bersih, tidak ada coretan.
11. Rambu lalu lintas yang digunakan di aplikasi ini merupakan rambu lalu lintas yang dipasang di luar ruangan (outdoor).

2. Dasar Teori

2.1 Android

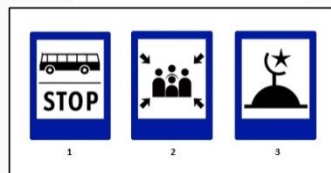
Ada beberapa jenis platform yang dapat digunakan untuk membuat aplikasi *mobile*, seperti *Windows Mobile*, *Symbian*, *iOS*, dan *Android*. Pada tugas akhir ini menggunakan platform *android* untuk mengimplementasikan aplikasi ini. Android adalah system operasi linux yang dimodifikasi untuk perangkat bergerak yang terdiri dari sistem operasi, *middleware*, dan aplikasi utama.

2.2 OpenCV

Library yang dipakai yaitu OpenCV. OpenCv adalah sebuah API (Application Programming Interface) [5]. OpenCV adalah library open source untuk computer vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi yang baik untuk gambar/video [5]. OpenCV (Open Source Computer Vision) adalah library untuk menangkap, memproses dan menampilkan gambar, baik 2D maupun 3D. OpenCV digunakan untuk pengenalan wajah, pengenalan gesture, deteksi gerakan dan lain-lain yang berkaitan dengan computer vision.

2.3 Rambu lalu lintas

Rambu lalu lintas merupakan objek yang akan di proses di pengolahan citra digital. Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang dapat berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan [2]. Sangat penting bagi para wisatawan untuk mematuhi rambu lalu lintas tersebut, terlebih dahulu kita harus memahami dan mengetahui arti dari rambu-rambu lalu lintas [2].



Gambar 2. 1 Contoh rambu lalu lintas

Banyak macam-macam pola rambu lalu lintas yang dipasang di jalan. Gambar 2.1 diatas merupakan contoh beberapa rambu lalu lintas di bidang pariwisata yang dipasang di jalan. Pada gambar 2.1 diatas nomor 1 merupakan pola rambu titik kumpul yang biasanya di pasang di tempat wisata. Untuk nomor 2 merupakan pola rambu peringatan angin kencang yang biasanya diletakan di pelabuhan atau jembatan penyebrangan antar pulau. Terakhir nomor 3 merupakan pola rambu pantai yang biasanya di pasang di pantai.

2.4 Citra digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometric), melakukan pemilihan citra ciri (feature image) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data [1]. Input dari pengolahan citra adalah citra, sedangkan output-nya adalah citra hasil pengolahan [1].

2.5 Metode ekstraksi SURF

SURF adalah local feature detector yang kuat, yang dapat digunakan pada computer vision seperti pengenalan objek atau rekonstruksi 3D. Algoritma SURF menggunakan penggabungan algoritma citra integral dan blob detection berdasarkan determinan dari matriks Hessian [3]. Salah satu kelebihan SURF adalah kecepatan pemrosesannya, hal ini disebabkan salah satunya oleh penggunaan gambar integral. Nilai dari gambar

integral ini berasal dari penjumlahan nilai grayscale dari gambar. SURF dirancang untuk mengekstrak kekhasan dan kesamaan fitur dari gambar. Algoritma SURF dibagi menjadi beberapa tahapan yaitu interest point detection dan feature description [3].

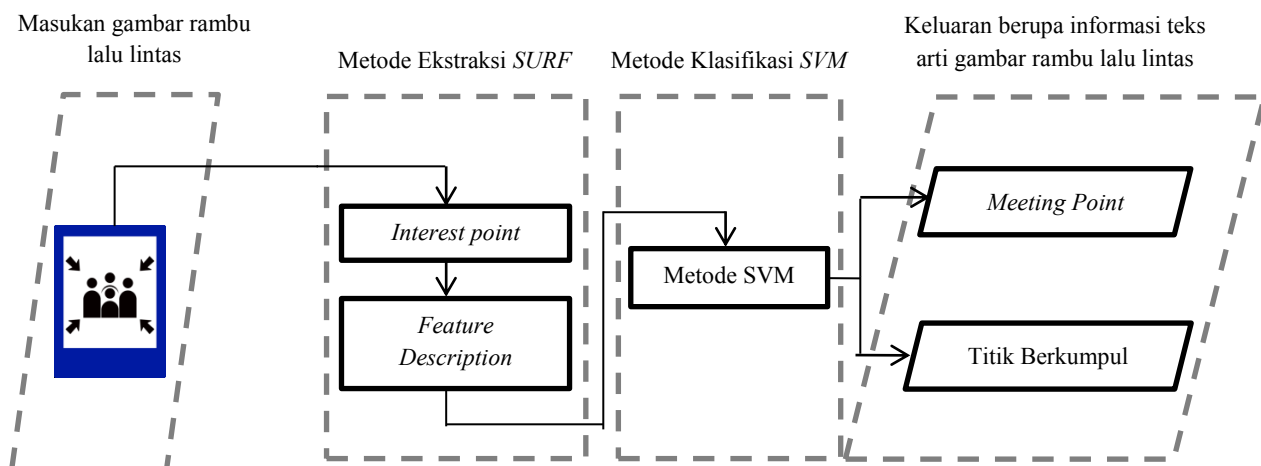
2.6 Metode klasifikasi SVM

Pada tahap klasifikasi, SVM adalah metode klasifikasi berdasarkan garis tepi terbesar [4]. Dalam machine learning, SVM merupakan metode pembelajaran berbasis ruang vector yang bertujuan untuk menemukan batas keputusan yang memiliki lebar maksimum dari dua titik antara kedua kategori pada data pelatihan [4]. Model SVM adalah representasi dari contoh sebagai titik dalam ruang, dipetakan sehingga contoh kategori terpisah dibagi oleh celah yang jelas dan selebar mungkin.

3. Perancangan sistem

3.1 Gambaran umum sistem

Secara umum, aplikasi ini mempunyai tahapan-tahapan untuk menjalankannya dengan benar. Tahapan-tahapan akan dibuat sesuai gambar dibawah ini .



Untuk gambaran umum sistem, seperti gambar 3.1 diatas. Pada awal sistem, kita masukan input gambar rambu lalu lintas dengan format (.PNG) atau (.JPEG). Selanjutnya masuk ke proses ekstraksi ciri gambar dengan metode *Speeded-Up Robust Features (SURF)* yang terdiri 2 proses, yaitu *interest point* dan *feature description* untuk melakukan pengenalan pada gambar rambu. Selanjutnya, hasil dari ekstraksi ciri pada gambar diklasifikasikan dengan *machine learning* metode *Support Vector Machine (SVM)*. Output yang dihasilkan adalah sebuah teks informasi dari arti gambar rambu lalu lintas yang di input di awal.

3.2 Spesifikasi kebutuhan perangkat

Berdasarkan perancangan, dibutuhkan perangkat lunak pendukung untuk implementasi ke dalam Aplikasi Pengenalan Rambu Lalu Lintas, antara lain sebagai berikut :

- Android Studio 2.2.3
- NetBeans IDE 8.1
- Eclipse Mars.2
- Android Software Development Kit (SDK)

e. *Java Development Kit (JDK)*

Sedangkan untuk perangkat keras yang digunakan untuk pengujian dan perancangan adalah sebagai berikut :

1) *Notebook* dengan spesifikasi sebagai berikut:

- a) Sistem Operasi Windows 8 64bit
- b) Intel® Core™ i-5-3337U CPU @ 1.80GHz
- c) RAM 8 GB

2) *Smartphone* dengan spesifikasi sebagai berikut :

- a) Sistem operasi versi *android* 6.0 (Marshmallow)
- b) CPU Quad-core 2.5 GHz Krait 400
- c) Ram 2 GB
- d) Kamera belakang 16 MP

4. Pengujian dan Analisis

4.1 Pengaruh resolusi dan citra rambu untuk data latih

Untuk melakukan sebuah data latih memerlukan banyak parameter untuk dijadikan acuan agar bisa di klasifikasi. Di pengujian ini, dibutuhkan beberapa parameter, yaitu mendapatkan keypoint citra terkecil, ukuran data latih, tingkat akurasi deteksi rambu, dan waktu komputasi SURF.

Tabel 4. 1 Data Ukuran Resolusi Citra Asli

Parameter	Keypoint rambu terkecil	Ukuran Data Training (KB)	Tingkat Akurasi (%)	Waktu Komputasi SURF (Detik)
Resolusi (Pixel)				
60 x 48	1	395	60	0.76
120 x 96	18	6230	92	5.53
240 x 192	71	24422	96	15.28
500 x 400	255	87570	96	52.83

Berdasarkan hasil table 4.1 diatas, dapat disimpulkan data latih yang dipakai, yaitu resolusi 500 x 400 pixel. Resolusi ini dipilih, karena memiliki rambu keypoint terkecil yang besar diantaranya. Keypoint rambu terkecil yang paling besar ini sangat penting dan berpengaruh untuk dijadikan acuan untuk masuk ke metode klasifikasi SVM. Walaupun ukuran data nya yang sangat besar dan komputasi yang lama, tetapi mempunyai tingkat akurasi yang tinggi untuk dijadikan data latih.

4.2 Banyaknya jumlah data latih yang digunakan

Pada pengujian ini, yaitu melakukan pengujian untuk mendapatkan jumlah data yang akan di latih sebanyak berapa agar mendapatkan tingkat akurasi yang tinggi. Jumlah data tiap rambu yang digunakan, yaitu 3 data, 6 data, dan 9 data. Sehingga data latih yang kita uji sebanyak 75 data, 150 data, dan 225 data.

Pengujian ini berdasarkan hasil pengujian diatas sebelumnya, maka data yang dipakai di pengujian ini, yaitu data dengan citra asli yang berukuran 500 x 400. Parameter ini diambil karena pada pengujian sebelumnya, pada citra dengan warna asli dengan ukuran 500 x 400 pixel memiliki akurasi yang tinggi.

Rumus yang dipakai tingkat akurasi = $\frac{\text{jumlah rambu benar}}{25} \times 100 \%$

Tabel 4. 2 Tingkat Akurasi Deteksi Citra Terhadap Jumlah Data Latih

Parameter	Keypoint rambu terkecil	Ukuran Data Training (KB)	Tingkat Akurasi (%)	Waktu Komputasi SURF (Detik)
Banyaknya data latih				
75	255	10503	4	15.16
150	255	39695	4	27.38
225	255	87570	96	52.83

Pada hasil tabel 4.2 diatas, dapat kita simpulkan bahwa keypoint rambu terkecil yang di dapatkan sama. Yang menjadi berbeda, yaitu ukuran data latih, tingkat akurasi deteksi rambu, dan waktu komputasi SURF. Berdasarkan hasil diatas, maka jumlah data yang layak dipakai untuk menjadi data latih yaitu sebanyak 225 data. Jumlah 225 data ini dipilih karena sudah sangat tinggi tingkat akurasinya dibandingkan jumlah data lainnya, walaupun ukuran data latih akan semakin besar dan waktu komputasinya SURF semakin lama.

4.3 Ketepatan deteksi rambu lalu lintas terhadap sudut pengambilan

Pada pengujian ini di implementasikan di *smartphone android*. Di pengujian ini, posisi saat pengambilan citra rambu dengan menggunakan *smartphone* sangat diperhatikan untuk pengujian. Pengujian ini bertujuan untuk mengetahui tingkat akurasi data uji yang diambil melalui kamera *smartphone* dengan berbagai posisi dan sudut yang ditentukan saat mengambil citra rambu lalu lintas tersebut. Parameter yang dipakai yaitu posisi saat pengambilan dengan mengukur sudut saat mengambil citra rambu lalu lintas. Posisi dan sudut buat pengujian dipilih berdasarkan berbagai sudut pengambilan, yaitu posisi dari bawah rambu, tepat rambu, kanan rambu, dan kiri rambu. Dimulai dari dengan sudut 15,30, dan 45 derajat. Untuk tingkat akurasi, yaitu mengukur tingkat akurasi keberhasilan deteksi citra rambu dengan berbagai sudut pengambilan yang diambil. Rumus yang dipakai tingkat akurasi = $\frac{\text{jumlah rambu benar}}{25} \times 100 \%$

Tabel 4. 3 Tingkat Akurasi Deteksi Citra Terhadap Sudut Pengambilan

	Tingkat Keberhasilan Deteksi Terhadap Sudut Pengambilan									
	B 15°	B 30°	B 45°	T 0°	Ka 15°	Ka 30°	Ka 45°	Ki 15°	Ki 30°	Ki 45°
Total Deteksi Benar	24	22	17	24	23	20	14	23	20	13
Tingkat Akurasi (%)	96	88	68	96	92	80	56	92	80	52

Dari tabel 4.3 diatas, dapat kita lihat hasil dari berbagai posisi sudut tengah menghasilkan tingkat akurasi yang tinggi, walaupun masih ada yang tidak terdeteksi. Berdasarkan diatas, tingkat akurasi ketika citra diambil dari bawah, yaitu sudut pengambilan 15 derajat mempunyai tingkat akurasi 96%, sudut pengambilan 30 derajat mempunyai tingkat akurasi 88%, dan sudut pengambilan 45 derajat mempunyai tingkat akurasi 68%. Untuk sudut pengambilan dari sisi kanan objek, dapat dilihat sudut pengambilan 15 derajat mempunyai tingkat akurasi 92%, sudut pengambilan 30 derajat mempunyai tingkat akurasi 80% , dan sudut pengambilan 45 derajat mempunyai tingkat akurasi 56%. Sedangkan untuk sudut pengambilan dari sisi kiri objek, dapat dilihat sudut pengambilan 15 derajat mempunyai tingkat akurasi 92%, sudut pengambilan 30 derajat mempunyai tingkat akurasi 80% , dan sudut pengambilan 45 derajat mempunyai tingkat akurasi 52%.

Dapat kita simpulkan, sudut yang memiliki tingkat akurasi yang tinggi, yaitu posisi pengambilan citra rambu lalu lintas tepat di tengah 0 derajat dengan 96%. Hal ini disebabkan posisi saat mengambil citra tepat berada di tengah objek, sehingga memungkinkan dapat mendeteksi dengan benar. Adapun sudut pengambilan yang memiliki tingkat akurasi yang rendah yaitu dari sisi kiri objek citra dengan 45 derajat sebesar 52%. Hal ini disebabkan juga karena pengambilan terlalu miring terhadap objek sehingga sulit untuk mendeteksi citra rambu lalu lintas. Sehingga aplikasi ini berjalan dengan maksimal ketika mengambil citra rambu pada tepat di 0 derajat.

4.4 Pengaruh cahaya terhadap deteksi rambu lalu lintas

Proses pengujian selanjutnya adalah pengujian terhadap pengaruhnya cahaya saat mendeteksi rambu lalu lintas dengan menggunakan kamera *smartphone*. Pada pengujian ini, pengaruh cahaya yang digunakan dibagi menjadi 2 kondisi, yaitu banyak cahaya saat pagi sampai siang hari dan kurang cahaya saat sore hari. Maksud dari banyak cahaya, yaitu pengaruh cahaya terhadap citra rambu sangat terang. Satuan untuk mengukur intensitas cahaya disini yaitu menggunakan LUX. LUX adalah satuan metric ukuran cahaya pada suatu permukaan. Disini parameter banyak cahaya bisa dimulai untuk pengambilan citra rambu lalu lintas dari jam 9 pagi sampai jam 2 siang dengan 4105 LUX sampai 10896 LUX. Untuk kurang cahaya, yaitu pengaruh cahaya sangat kurang untuk pengambilan citra rambu lalu lintas. Parameter kurang cahaya ini bisa dimulai untuk pengambilan citra rambu lalu lintas dari jam 4 sore sampai jam 6 sore dengan 191 LUX sampai dengan 4104 LUX.

Posisi pengujian ini diambil dari posisi sudut tengah 0 derajat karena sesuai pengujian sebelumnya, posisi dari sudut tengah 0 derajat ini memiliki tingkat akurasi yang tinggi. Pengujian ini dilakukan bertujuan untuk mengetahui bagaimana pengaruh cahaya terhadap deteksi citra rambu lalu lintas. Pengaruh cahaya sangat penting untuk mendeteksi citra rambu, karena cahaya disini menentukan keypoint yang di deteksi.

Rumus yang dipakai tingkat akurasi = $\frac{\text{jumlah rambu benar}}{25} \times 100 \%$.

Tabel 4. 4 Tingkat Akurasi Deteksi Citra Terhadap Cahaya

	Tingkat Kecerahan Cahaya	
	4106 - 10896	191 - 4105
Total Deteksi Benar	24	22
Tingkat Akurasi (%)	96	88

Dari hasil pengujian diatas, dapat disimpulkan bahwa keberhasilan deteksi rambu lalu lintas mempunyai tingkat akurasi yang tinggi ketika diambil saat 4105 LUX sampai dengan 10896 LUX . Pengaruh banyak cahaya ini sangat penting karena untuk mendapatkan keypoint rambu terkecil yang besar. Sehingga ketika gelap atau kurang cahaya, keypoint tiap citra rambu terlalu sedikit dan susah diklasifikasi di data latih. Sehingga aplikasi ini berjalan dengan baik dan maksimal ketika citra rambu lalu lintas berada pada 4105 LUX sampai dengan 10896 LUX.

4.5 Pengaruh jarak pengambilan citra terhadap ketepatan deteksi citra rambu lalu lintas

Pengujian selanjutnya adalah pengujian pengaruh jarak terhadap deteksi citra rambu lalu lintas. Pada pengujian ini kondisi jarak dibagi menjadi 3 kondisi, yaitu jarak dekat, jarak normal, dan jarak jauh. Pengukuran jarak ini berdasarkan kotak hijau yang ada di kamera *smartphone*. Resolusi dari kotak hijau ini berukuran 500 x 400 pixel sesuai dengan *referensi* undang-undang. Dibuat kotak hijau di kamera ini juga sangat penting untuk deteksi citra rambu lalu lintas ini. Sehingga background diluar kotak hijau akan dihilangkan dan hanya tersisa citra yang ada di kotak hijau dengan ukuran 500 x 400 pixel .

Maksud jarak pendek, yaitu citra rambu lalu lintas terlalu masuk ke dalam kotak hijau atau citra rambu dapat diambil dengan jarak kurang lebih 142 Cm. Untuk jarak normal, yaitu citra rambu lalu lintas ini berada pas dalam kotak hijau atau citra rambu dapat diambil dengan jarak kurang lebih 164 Cm. Terakhir, jarak jauh yaitu citra rambu lalu lintas diambil dengan background belakangnya atau citra rambu dapat diambil dengan jarak kurang lebih 198 Cm. Tujuan pengujian ini untuk mengetahui tingkat akurasi deteksi rambu lalu lintas saat pengambilan citra dipengaruhi oleh jarak.

Rumus yang dipakai untuk tingkat akurasi = $\frac{\text{jumlah rambu benar}}{25} \times 100 \%$.

Tabel 4. 5 Keberhasilan Deteksi Rambu Terhadap Jarak Pengambilan

	Jarak Pengambilan Citra (CM)		
	±142	±164	±198
Total Deteksi Benar	20	24	18
Tingkat Akurasi (%)	80	96	72

Dari tabel 4.5 diatas, keberhasilan untuk deteksi rambu lalu lintas saat jarak kurang lebih 142 cm mempunyai tingkat akurasi sebesar 80%. Untuk pengambilan dengan jarak kurang lebih 164 cm, mempunyai tingkat akurasi sebesar 96%. Dan saat pengambilan dengan jarak kurang lebih 198 cm, mempunyai tingkat akurasi sebesar 72%.

Berdasarkan tabel 4.7 diatas, dapat disimpulkan bahwa aplikasi ini berjalan baik dan maksimla saat pengambilan citra rambu lalu lintas dengan jarak kurang lebih 164cm karena mempunyai tingkat akurasi sangat tinggi, yaitu sebesar 96%. Hal ini dikarenakan semakin kita tepat mengambil citra rambu lalu lintas di kotak hijau, maka semakin bagus dan akurat keypoint yang di dapat sehingga mendapatkan akurasi yang tinggi juga.

4.6 Pengaruh Perfomansi

Pengujian ini dilakukan untuk membandingkan perfomansi aplikasi saat di gunakan di jenis *smartphone* yang berbeda-beda. Parameter yang digunakan yaitu versi *android smartphone*, processor *smartphone*, RAM *smartphone*, dan waktu saat komputasi di *smartphone* tersebut.

Tabel 4. 3 Hasil Pengujian Lama Waktu Respon Sistem

No	Nama <i>Smartphone</i>	Versi Android	Processor (GHz)	RAM (GB)	Waktu Mendeteksi SURF (Detik)
1	<i>Smartphone 1</i>	Kitkat (4.4)	1.2 Quad-Core	2	11.72
2	<i>Smartphone 2</i>	Lolipop (5.1)	1.3 Quad-Core	2	8.75
3	<i>Smartphone 3</i>	Marshmallow (6.1)	2.5 Quad-Core	2	5.36

Berdasarkan tabel 4.3 diatas, maka dapat dilihat aplikasi berjalan baik dan cepat ketika menggunakan perangkat yang mempunya processor 2.5 GHz Quad-Core, memiliki versi android Marshmallow (6.1) dengan waktu deteksi citra selama 5.36 detik. Semakin besar processor yang digunakan dan memiliki versi android yang terbaru, maka semakin cepat pula waktu yang dibutuhkan untuk deteksi rambu lalu lintas ini.

5. Kesimpulan dan Saran

Dari hasil pengujian dan analisi pada bab sebelumnya, maka dapat disimpulkan beberap hal sebagai berikut :

1. Data latih yang digunakan untuk aplikas ini yaitu data citra asli yang mempunyai resolusi 500 x400 pixel dan jumlah data yang dilatih sebanyak 225 data. Data ini digunakan karena dengan 225 data citra asli yang digunakan mempunyai tingkat akurasi deteksi citra rambu lalu lintas benar yang tinggi yaitu 96%. Walaupun ukuran data latih ini besar dan waktu prosesnya lama.
2. Ketepatan deteksi citra rambu lalu lintas ini mempunya tingkat akurasi benar yang tinggi sebesar 96%, ketika mengambil citra rambu ini tepat di kotak hijau yang ditampilkan di layar *smartphone* dan diambil saat rentang tingkat kecerahan cahaya pada 4106 LUX sampai dengan 10896 LUX..
3. Pada metode SURF, jumlah keypoint terkecil yang besar sangat mempengaruhi tingkat akurasi untuk mengenal suatu citra rambu. Semakin besar resolusi, maka semakin tinggi pula jumlah keypoint yang di

deteksi dan semakin besar juga akurasi yang di dapat, walaupun untuk ukuran data latih sangat besar dan waktu komputasi nya lama.

4. Agar data bisa di latih dan di proses oleh SVM, maka dibutuhkan jumlah keypoint terkecil sebagai acuan untuk memangkas jumlah keypoint citra yang besar. Hal ini disebabkan karena pada metode klasifikasi SVM hanya mau memproses data yang seragam, sehingga jumlah keypoint citra yang besar akan dipangkas menjadi jumlah keypoint yang terkecil.
5. Perancangan metode ekstraksi ciri dengan metode *Speeded-Up Robust Features* (SURF) dan metode klasifikasi *Support Vector Machine* (SVM) dapat diterapkan pada perangkat *smartphone* android dalam bentuk aplikasi. Berdasarkan pengujian beta, 60% responden memberikan penilaian baik terhadap fungsi dan kegunaan aplikasi ini.

Ada beberapa hal yang dapat menjadi bahan masukan atau dikembangkan terkait pengujian diatas, antara lain :

1. Mengembangkan aplikasi ini dengan *platform* selain android, seperti iOS.
2. Pada tahap klasifikasi. Dapat mengembangkan klasifikasi selain SVM, seperti KNN.

Daftar Pustaka

- [1] Sutoyo T, Muyanto Edy, Suhartono Vincent, Nurhayati Oky Dwi, Wijanarto.,2009.*Teori Pengolahan Citra Digital*.Yogyakarta : ANDI.
- [2] Trisula Sign. "Pengertian Rambu Lalu Lintas". Diakses 30 September 2016. <http://multirambu-ts.blogspot.co.id/2014/07/pengertian-rambu-lalu-lintas.html>.
- [3] Gunarso Muhammad. 2004. *Deteksi Objek Pada Gambar Menggunakan Algoritma Speeded-Up Robust Features*. Artikel Jurnal. Universitas Muhammadiyah Jember, Jember.
- [4] Farsiah, Laina., Fuadi Abidin, Taufik., dan Munadi, Khairul. 2013. *Klasifikasi Gambar Berwarna Menggunakan K-Nearest Neighbor dan Support Vector Machine*. Artikel. Univeristas Syiah Kuala, Banda Aceh.
- [5] Arsy, lazuardi., Nurhayati, Oky Dwi., Martono, Kurniawan Teguh. 2016. *Aplikasi Pengolahan Citra Digital Meat Detection Dengan Metode Segmentasi K-Mean Clustering Berbasis Android*. Jurnal Teknologi dan Sistem Komputer. Universitas Diponegoro. Semarang.