

BAB III METODOLOGI PENELITIAN

3.1. Metode Pengumpulan Data

Metode pengumpulan data adalah cara atau teknik yang dilakukan dalam memperoleh data pendukung pada penulisan laporan penelitian. Pengumpulan data dilakukan dengan pengumpulan gambar rambu dari sumber data dan studi literatur.

3.1.1. Pengumpulan Gambar Rambu Lalu Lintas

Tahapan ini dilakukan untuk mendapatkan gambar-gambar dan informasi dari Dinas Perhubungan Darat sebagai pelaksana peraturan menteri mengenai rambu lalu lintas yang berlaku di Indonesia. Gambar dan informasi yang diperoleh akan digunakan dalam pengembangan program deteksi.

3.1.2. Studi Literatur

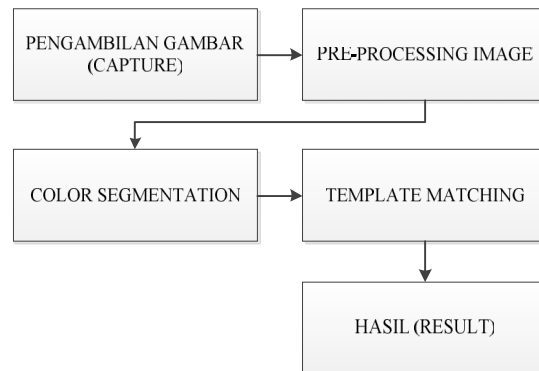
Penelitian dilakukan dengan membaca buku-buku, literatur-literatur dan jurnal-jurnal yang diperoleh dari perpustakaan, akademi, dan situs-situs pendukung yang memiliki hubungan langsung dengan obyek penelitian yang dipilih. Tujuan dari studi pustaka adalah menemukan teori-teori pendukung dalam pengembangan sistem untuk dijadikan referensi dalam penelitian.

3.2. Arsitektur Deteksi Rambu Lalu Lintas

Penelitian tentang deteksi rambu lalu lintas mencoba menggunakan arsitektur sistem deteksi pada gambar bergerak. Arsitektur deteksi rambu lalu lintas dijelaskan melalui gambaran umum proses deteksi rambu lalu lintas.

3.2.1. Gambaran Umum Deteksi Rambu Lalu Lintas

Gambar 3.1 berikut adalah gambaran umum mengenai proses deteksi rambu lalu lintas dari tahap awal sampai memperoleh hasil.



Gambar 3.1 Alur Deteksi Rambu Lalu Lintas

Berdasarkan gambaran diatas, penelitian mengenai deteksi rambu lalu lintas dengan teknik *color mapping* dan *template matching* ini diawali dengan proses pengambilan gambar, kemudian tahapan *pre-processing image*, tahapan *color segmentation*, dan tahap *template matching* untuk memperoleh hasil deteksi.

3.2.2. Pengambilan Gambar (*Capturing*)

Untuk dapat melakukan analisa gambar dengan menggunakan komputer, terlebih dahulu dibutuhkan gambar atau citra dalam bentuk digital. Citra rambu lalu lintas yang digunakan dalam penelitian ini diambil dari Dinas Perhubungan yang merujuk pada Peraturan Menteri No. 13 Tentang Rambu Lalu Lintas. Citra berdasarkan rujukan kemudian dijadikan acuan dalam penelitian ini dan akan digunakan sebagai citra acuan atau citra template pada proses pengujian.

Selain citra rambu lalu lintas sebagai citra template, citra atau gambar lain yang digunakan pada proses pengujian adalah gambar bergerak (video). Video yang diambil menggunakan kamera digunakan pada proses pengujian sebagai citra masukan. Perekaman video ini diambil dengan menggunakan kamera digital dan berupa rekaman perjalanan di jalan raya berlalu lintas pada saat berkendara dengan kendaraan roda empat dengan beberapa kriteria yaitu kondisi cuaca cerah dan video rekaman diambil pada siang hari, dimana kamera yang digunakan dapat merekam gambar video dengan

cukup baik. Hasil dari pengambilan video ini selanjutnya akan digunakan untuk melakukan percobaan deteksi rambu lalu lintas yang menjadi tujuan penelitian.

3.2.3. *Image Pre-processing*

Tahap *image pre-processing* dilakukan untuk meningkatkan kualitas citra masukan. Kualitas dari citra masukan biasanya memiliki beberapa kekurangan untuk diproses lebih lanjut, *noise* maupun intensitas yang rendah bisa menjadi masalah dalam proses selanjutnya. Tujuan dari proses ini adalah untuk mengintensifkan ataupun mengurangi detail yang tidak digunakan pada citra masukan sehingga memungkinkan evaluasi citra dapat dilakukan lebih cepat. Pada tahap ini dilakukan perbaikan tingkat kecerahan dari gambar, menyesuaikan intensitas warna, mengurangi *noise*, dan *geometric transformation* untuk menyesuaikan bentuk gambar tangkapan.

3.2.4. Segmentasi Warna

Segmentasi warna merupakan proses pemisahan segmen dalam suatu gambar berdasarkan warna yang terkandung pada gambar tersebut. Untuk melakukan segmentasi pada gambar, nilai-nilai warna pada gambar tersebut direpresentasikan ke dalam bentuk angka-angka yang mewakili warna pada setiap pikselnya, hal ini bertujuan untuk memudahkan dalam menentukan nilai indeks warna. Pada proses segmentasi akan digunakan ruang warna HSV untuk memberikan nilai warna yang lebih akurat.

3.2.5. *Template Matching*

Metode *template matching* biasanya digunakan untuk menemukan bagian dari gambar yang cocok dengan gambar contoh. Dengan mengetahui intensitas piksel dari gambar contoh, kita dapat membandingkannya dengan intensitas piksel dari gambar uji. Perbandingan intensitas ini dihitung dengan metode *Cross Correlation*.

3.3. Metode Perancangan Perangkat Lunak

Metode yang digunakan untuk perancangan perangkat lunak deteksi rambu lalu lintas adalah metode *prototype*. Metode *prototype* dipilih karena cocok untuk mengembangkan sebuah perangkat lunak yang akan dikembangkan kembali. Tahapan dalam penerapan metode *prototype*, yaitu : *communication*, *quick plan*, *modelling (quick design)*, *construction of prototype*, *deployment (delivery and feedback)*.

3.3.1. *Communication* (Komunikasi)

Aktivitas atau kegiatan yang dilakukan dalam tahapan komunikasi antara lain yaitu melakukan pengumpulan data yang diperlukan untuk merancang aplikasi yang memiliki data tepercaya dan tepat tentang rambu lalu lintas di Indonesia. Selain itu juga mengumpulkan data dan kebutuhan perangkat lunak yang diperlukan untuk kemudian dianalisis untuk proses selanjutnya.

3.3.2. *Quick Plan* (Perencanaan Cepat)

Data dan kebutuhan perangkat lunak dianalisis dan menghasilkan sebuah user requirement. Adapun spesifikasi kebutuhan perangkat lunak yang dibutuhkan sebagai sarana pengembangan aplikasi adalah sebagai berikut :

1. Sistem Operasi yang digunakan dalam pembuatan aplikasi ini adalah Windows 10 64-bit
2. Perangkat lunak pendukung pengembangan aplikasi antara lain :
 - a. Code::Blocks, sebagai Integrated Development Environment (IDE) pengembangan aplikasi, dengan library OpenCV.
 - b. MinGW, sebagai kompilator bahasa pemrograman C++.
 - c. Microsoft Visio, untuk mendesain flowchart.
 - d. OpenCV, sebagai *library computer Vision*.

Untuk spesifikasi hardware yang digunakan pada penelitian ini adalah sebagai berikut :

1. Spesifikasi komputer yang digunakan adalah :
 - a. Processor Intel(R) Core(TM) i5-6200U CPU @ 2.30Ghz
 - b. RAM 4 GB
 - c. VGA NVIDIA GeForce 940MX 2GB
 - d. Monitor 15,6"
 - e. Harddisk 1 TB
 - f. Keyboard dan Optical Mouse
2. Kamera yang digunakan adalah Xiaomi Yi Action Cam dengan kualitas video rekaman sampai dengan 1080p pada 60fps.

Sedangkan aplikasi yang dibutuhkan adalah aplikasi yang mampu memberikan informasi dan notifikasi ketika mendeteksi suatu rambu. Informasi yang diberikan berupa nama rambu dan gambar rambu sehingga pengguna dapat mengetahui rambu yang berlaku di jalan tersebut.

3.3.3. *Modelling (Permodelan)*

Tahapan ini merupakan tahap permodelan, permodelan yang dilakukan adalah cara kerja sistem, rancangan aplikasi yang diperoleh berdasarkan analisis kebutuhan aplikasi pada tahapan sebelumnya.

3.3.3.1. Tahapan Kerja Aplikasi Deteksi Rambu Lalu Lintas

Tahapan kerja pada aplikasi deteksi rambu lalu lintas adalah sebagai berikut :

1. Tahap Pembacaan Video dan *Template Image*

Video yang digunakan adalah rekaman perjalanan menggunakan kendaraan roda empat di jalan protokol Bandar Lampung, berikut adalah data video yang digunakan pada penelitian ini,

Tabel 3.1 Video rekaman

Nama Jalan	Jalan R.A. Kartini		
Kecepatan Kendaraan	20 km/jam	30km/jam	40km/jam

Video yang akan digunakan dalam penelitian ini bertipe mp4 memiliki ukuran 1280 x 720 piksel dengan 29,970 frame per detik.

Template image atau gambar *template* yang digunakan adalah gambar pada daun rambu lalu lintas yang akan dideteksi. Pada penelitian ini akan digunakan gambar daun rambu yang banyak ditemukan pada jalan protokol yang menjadi lokasi perekaman video. Ukuran yang akan digunakan adalah 50 x 50 piksel dengan tipe gambar adalah JPEG (*Joint Photographic Expert Group*).

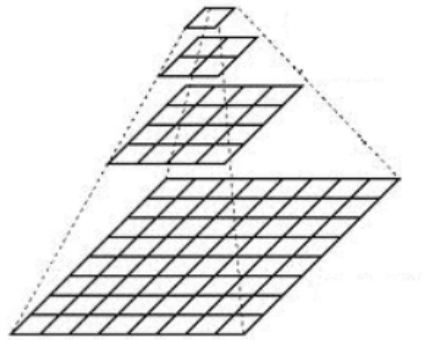
2. Tahap *pre-processing image*

Tahap ini dilakukan sebelum melakukan analisis gambar atau tahap-tahap lainnya dalam *computer vision*, tujuannya untuk dapat mengurangi *noise* yang ada, mengatur tingkat kecerahan gambar dan menonjolkan bagian tertentu dari gambar untuk mempermudah proses analisis.

a. *Subsampling*

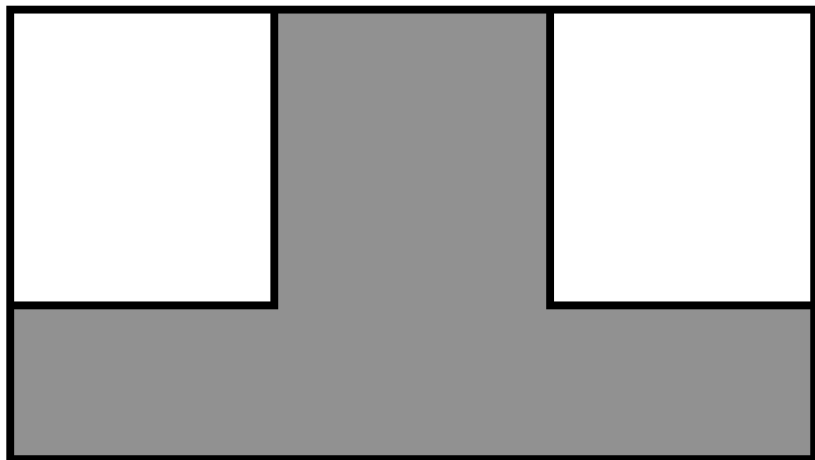
Subsampling digunakan untuk mempercepat proses kalkulasi atau analisis dengan mengurangi jumlah informasi (jumlah piksel atau jumlah kanal warna) yang digunakan. Pada penelitian ini akan digunakan salah satu teknik *image pyramid* yaitu *gaussian pyramid*. *Gaussian pyramid* akan digunakan untuk mengurangi jumlah piksel pada gambar dengan menghilangkan semua baris dan kolom yang memiliki indeks genap dan membentuk gambar baru pada *gaussian pyramid* yang bisa kita sebut sebagai G_{i+1} dari gambar asal G_i . Dimana setiap gambar hasil *gaussian pyramid* akan memiliki ukuran hanya seperempat dari gambar aslinya. Pada gambar 3.3 digambarkan ilustrasi dari proses *gaussian Pyramid*, dimana penampang paling bawah

adalah gambar asli (G_i) dan penampang diatasnya adalah dari *gaussian pyramid*, dan hasil terasi selanjutnya.



Gambar 3.2 Ilustrasi dari *Gaussian Pyramid*

Medote lain yang digunakan adalah dengan menggunakan sebagian dari keseluruhan dari video masukan. Berdasarkan informasi bahwa rambu lalu lintas dipasang pada bagian kiri dan kanan jalan dan asumsi bagian bawah dari video merupakan badan jalan dan trotoar, dari video akan digunakan hanya 2/3 bagian kanan dan 2/3 bagian kiri dengan ilustrasi seperti pada gambar 3.3, bagian tidak berwarna adalah area yang akan digunakan pada proses deteksi.



Gambar 3.3 Ilustrasi bagian video yang digunakan

b. *Histogram Equalization*

Histogram merupakan distribusi dari nilai dari suatu gambar yang dapat direpresentasikan dalam bentuk satu atau beberapa dimensi. *Histogram Equalization* merupakan salah satu metode untuk menyeimbangkan kecerahan atau kontras dari suatu gambar (Jain, 1989) sehingga didapatkan gambar dengan tingkat kontras yang seimbang dengan meningkatkan kontras pada area dengan kontras rendah, dan mengurangi kontras pada area berkontras tinggi. Pada penelitian ini, *histogram equalization* digunakan untuk menyeimbangkan tingkat kecerahan pada gambar masukan.

c. *Denoising*

Suatu gambar yang dihasilkan dari perangkat visual dipastikan memiliki *noise* yang disebabkan oleh faktor lingkungan, perangkat untuk mengambil gambar, dan lain sebagainya.

Salah satu teknik dalam mengurangi *noise* pada gambar adalah *smoothing* atau *blurring*. Ada beberapa teknik *smoothing* atau *blurring* yaitu *normalized box filter*, *gaussian filter*, *median filter*, *bilateral filter*. Pada penelitian ini digunakan teknik *gaussian filter*, dimana perhitungannya menggunakan *gaussian kernel* pada *input array* untuk menentukan keluaran berupa *array* pada area tersebut.

3. Tahap segmentasi warna

Untuk dapat melakukan proses segmentasi warna, perlu ditentukan warna yang digunakan untuk melakukan proses segmentasi, ini dapat dilakukan dengan menggunakan warna yang ada pada rambu lalu lintas yang akan dideteksi. Selain mendapatkan warna dari rambu lalu lintas, jumlah warna pada video uji juga perlu dilakukan

proses pengurangan jumlah warna, untuk dapat membantu proses segmentasi. Untuk melakukan proses pengurangan warna dapat dilakukan beberapa metode, salah satu metodenya adalah dengan pemetaan warna (*color mapping*), dimana warna dengan jarak tertentu dari titik pusat akan dipetakan menjadi warna yang sama dengan titik pusat. Teknik yang digunakan pada pemetaan warna ini adalah *color quantization*.

4. Tahap *Template Matching* untuk Mendeteksi Rambu

Template Matching digunakan untuk menemukan rambu lalu lintas yang menjadi gambar *template* pada gambar input bergerak (video). Teknik ini biasanya digunakan untuk mencari objek yang memiliki sedikit perbedaan ataupun objek serupa pada gambar input dari gambar *template*. Algoritma dalam proses *template matching* yaitu mencari nilai kesamaan atau perbedaan antara gambar input dan gambar *template* dengan menggeser gambar *template* pada gambar input pada setiap posisi yang memungkinkan.

Untuk menemukan nilai kesamaan ataupun perbedaan dari gambar input dan gambar *template* digunakan perhitungan jarak *euclidean* dimana dilakukan perhitungan perbandingan jarak antara titik-titik perpotongan dari kedua dimensi gambar atau dikenal dengan *cross correlation*, dengan rumusnya adalah :

$$D_{CrossCorrelation}(i,j) = \sum_{(m,n)} f(i+m,j+n) \cdot t(m,n)$$

Dimana $f(i,j)$ adalah gambar input dan $t(m,n)$ adalah gambar *template*. Untuk dapat mendapatkan nilai kesamaan atau perbedaan yang lebih akurat, dapat digunakan normalisasi untuk mendapatkan nilai antara 0.0 sampai dengan 1.0, dimana nilai 1.0 dikatakan sebagai perfect match atau kemiripan sempurna. Normalisasi dari

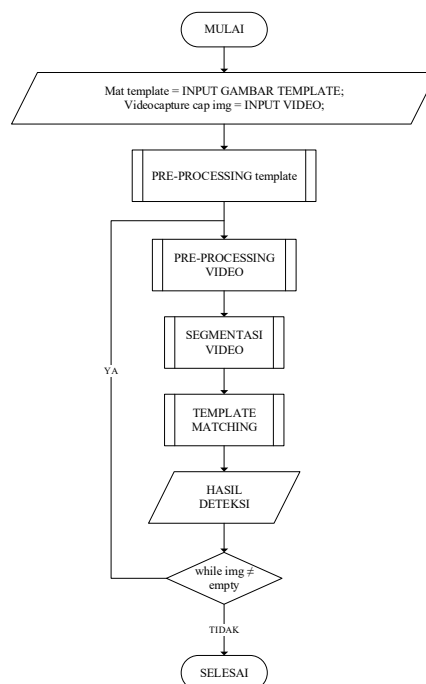
perhitungan *cross correlation* dapat dijabarkan dengan rumus sebagai berikut :

$$D_{NormCrossCorr}(i, j) = \frac{\sum_{(m,n)} f(i + m, j + n) \cdot t(m, n)}{\sqrt{\sum_{(m,n)} f(i + m, j + n)^2 \sum_{(m,n)} t(m, n)^2}}$$

Hasil dari perhitungan *normalised cross correlation* merupakan deretan nilai yang dapat dikatakan sebagai deretan derajat kecocokan antara dua gambar yang dibandingkan. Pada teknik ini, nilai tertinggi pada deret nilai merupakan titik kemiripan yang paling ideal.

3.3.3.2. Flowchart Aplikasi

Usulan aplikasi ini akan menerima input berupa video dan gambar *template* yang digunakan dalam proses deteksi, dan menampilkan hasil deteksi pada video. Gambar 3.4 adalah *flowchart* program utama yang diusulkan.

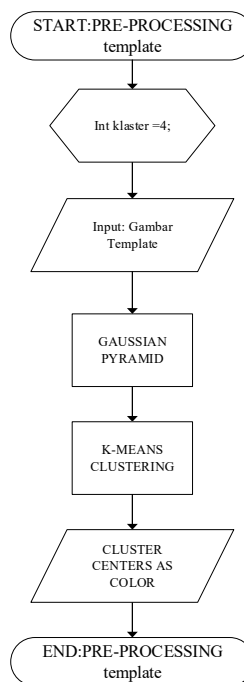


Gambar 3.4 *Flowchart* Program Deteksi Rambu Lalu Lintas

Aplikasi ini akan menerima dua input yang kemudian masuk ke subproses mulai dari *pre-processing*, segmentasi dan *template matching*, yang kemudian memberikan hasil deteksi dari rambu lalu lintas. Proses akan berulang dan perhitungan akan dilakukan pada setiap frame video satu persatu sampai video selesai.

1. Subproses *Pre-Processing* Gambar *Template*

Subproses *pre-processing* mengolah input yang ada sebelum melanjutkan ke subproses lainnya. Rancangan *flowchart* untuk subproses ini pada gambar 3.5 untuk gambar *template* dan gambar 3.6 untuk gambar pada video.



Gambar 3.5 *Flowchart Pre-Processing* Gambar *Template*

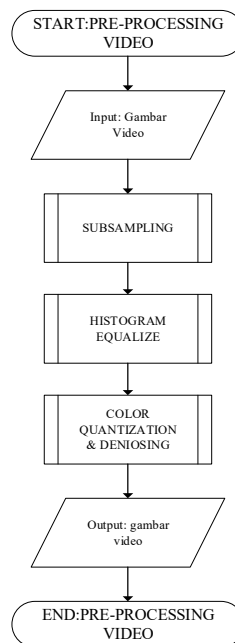
Subproses *preprocessing* pada gambar *template* dimulai dengan menentukan jumlah warna yang akan digunakan kemudian pada proses segmentasi. Jumlah warna yang ditargetkan digunakan sebagai jumlah klaster pada algoritma klasterisasi *k-means*, jumlah

klaster 4 digunakan dengan pertimbangan bahwa warna yang dimiliki rambu lalu lintas tidak melebihi jumlah tersebut.

Kemudian dilanjutkan dengan melakukan subsampling menggunakan *gaussian pyramid* untuk mengurangi jumlah piksel yang akan digunakan kemudian proses *template matching*. Selanjutnya dengan algoritma klasterisasi *k-means*, warna yang ada pada gambar rambu akan diklasterisasi menjadi 4 warna, dan warna tersebut akan disimpan untuk digunakan pada proses lanjutan.

2. Subproses *Pre-Processing* pada Gambar Video

Subproses ini menggunakan gambar yang berasal dari rangkaian sejumlah gambar yang ada pada video.

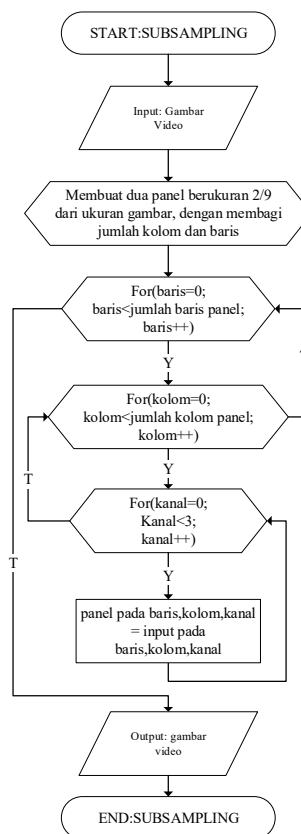


Gambar 3.6 *Flowchart Pre-Processing* pada Gambar Video

Proses *pre-processing* video memiliki beberapa subproses sendiri untuk mendapatkan hasil yang lebih maksimal dalam proses deteksi. Subproses yang akan dilakukan yaitu *subsampling*, *histogram equalize*, dan *color quantization* serta *denoising*.

a. *Subsampling*

Proses ini dilakukan untuk dapat mengurangi komputasi yang tidak diperlukan dalam proses *template matching*. Langkah awal yang akan digunakan adalah membuat dua buah variabel baru untuk menampung gambar, dengan ukuran masing-masing hanya $\frac{2}{9}$ dari ukuran gambar video. Ukuran ini diambil berdasarkan informasi bahwa rambu lalu lintas diletakkan pada bagian kanan dan kiri jalan dan daun rambu terletak lebih tinggi dari badan jalan dan trotoar.



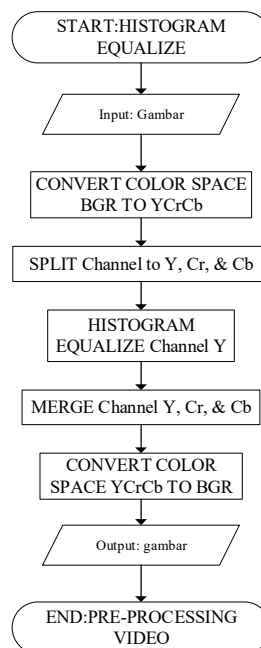
Gambar 3.7 *Flowchart* Proses *Subsampling* pada Video

Variabel yang digunakan untuk menampung potongan gambar yang akan digunakan, kemudian diisi dengan informasi yang ada pada gambar input. Proses ini dilakukan dengan melakukan iterasi pada baris, kolom dan kanal warna pada variabel tersebut

dan mengambil informasi acara satu persatu pada setiap baris dan kolom berdasarkan kanal warna sampai dengan baris dan kolom pada variabel tersebut terisi seluruhnya.

b. *Histogram Equalize*

Subproses *histogram equalize* digunakan pada gambar video input untuk menormalkan intensitas pada kanal warna. Kanal warna yang akan seimbangkan adalah kanal warna yang memiliki pengaruh pada intensitas cahaya atau kecerahan dan kontras pada gambar. Gambar 3.8 menggambarkan alur dari proses *histogram equalize*.



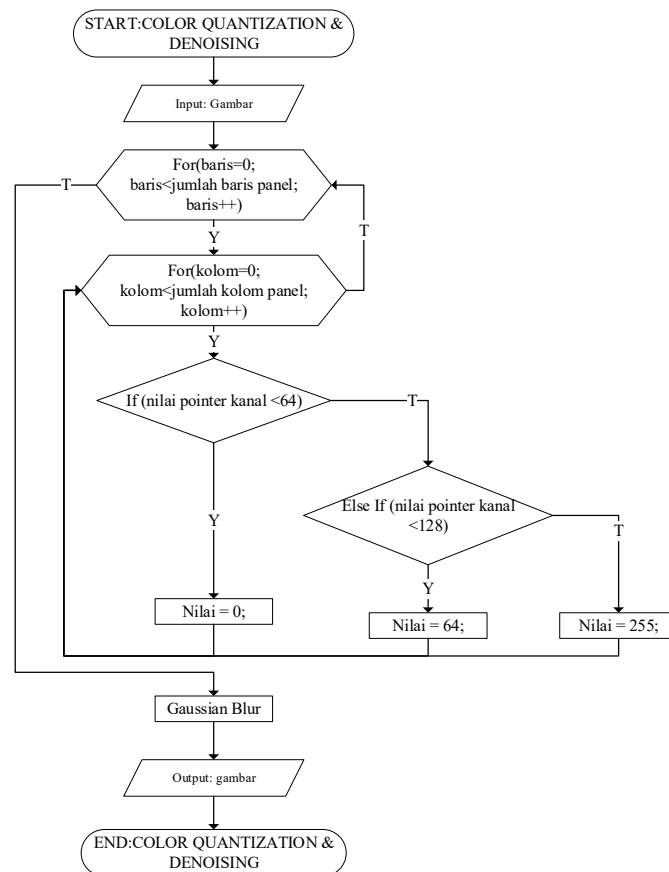
Gambar 3.8 *Flowchart* Proses *Histogram Equalize*

Proses ini akan memetakan ruang warna BGR ke ruang warna YCrCb yang memiliki kanal Y sebagai kanal yang mempengaruhi tingkat kecerahan dan kontras gambar. Setelah dipetakan ke ruang warna YCrCb, tiga kanal warna yang ada dipisahkan (*split*) agar proses *histogram equalize* dapat dilakukan pada kanal Y. Setelah melakukan *histogram equalize*,

proses dikembalikan lagi dengan menyatukan kanal (*merge*) warna dan memetakan kembali kanal warna ke dalam ruang warna BGR.

c. *Color Quantization dan Denoising*

Proses *color quantization* bertujuan untuk mengurangi jumlah warna yang digunakan pada gambar. Proses ini dilakukan dengan melakukan pemetaan intensitas warna setiap piksel sehingga warna yang didapatkan dapat diturunkan ke dimensi yang diinginkan.



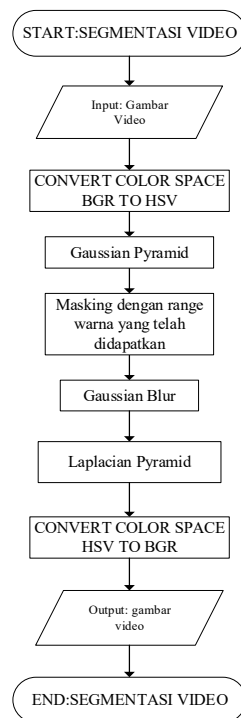
Gambar 3.9 *Flowchart* Proses *Color Quantization* dan *Denoising*

Gambar 3.9 menjabarkan proses yang akan memotong jumlah warna dengan mencari nilai terdekat dari warna yang ada.

Selanjutnya dilakukan proses *denoising* dengan teknik *gaussian blur*.

3. Proses Segmentasi

Setelah proses *pre-processing image*, langkah selanjutnya adalah melakukan segmentasi pada gambar input, tujuannya adalah untuk mendapatkan hasil yang lebih maksimal dengan mengurangi informasi-informasi yang tidak digunakan pada tahapan selanjutnya. Segmentasi ini akan menggunakan informasi kanal warna yang dibandingkan dengan warna yang didapat pada proses *pre-processing* gambar template, dan menghilangkan informasi-informasi yang tidak digunakan pada proses *template matching*.



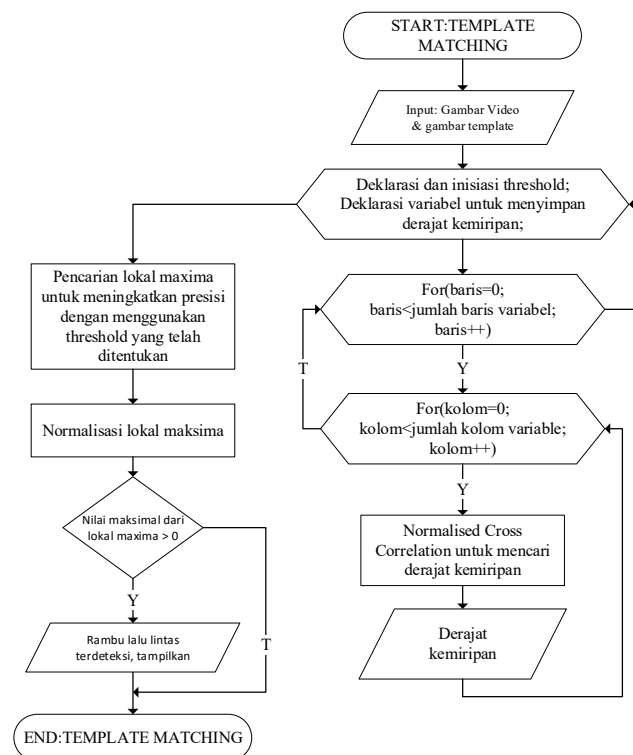
Gambar 3.10 *Flowchart* Proses Segmentasi Video

Proses akan dilakukan pada ruang warna HSV, sehingga gambar akan dipetakan terlebih dahulu dari ruang warna BGR ke HSV. Selanjutnya akan dilakukan *downsampling* dengan teknik *gaussian pyramid* sebelum dilakukan segmentasi dengan warna yang telah

didapatkan. Setelah proses segmentasi, akan dilakukan proses *denoising* dengan *gaussian blur* dan dikembalikan ke ukuran asal dengan teknik *laplacian pyramid*. Teknik *laplacian pyramid* akan memperbesar ukuran gambar menjadi 2 kali lebih besar dan kemudian baris dan kolom baru akan diisi dengan nilai yang dikalkulasi dengan pendekatan kebalikan dari gaussian.

4. Proses *Template Matching*

Proses selanjutnya yang akan dilakukan adalah *template matching*, pada proses ini akan menggunakan algoritma *template matching* dimana setiap posisi atau lokasi pada gambar input akan dihitung derajat kemiripan dengan gambar *template*. Nilai derajat kemiripan akan dibandingkan dengan nilai ambang batas (*threshold*) untuk menentukan apakah rambu lalu lintas telah terdeteksi atau belum terdeteksi.



Gambar 3.11 *Flowchart* Proses *Template Matching*

3.3.4. *Construction of Prototype*

Tahapan ini merupakan kontruksi atau implementasi terhadap aplikasi yang akan dibuat sesuai dengan desain yang telah dilakukan sebelumnya. Kebutuhan dan solusi dari permasalahan yang ada diubah dari desain ke dalam sebuah aplikasi.

3.3.5. *Deployment*

Tahapan ini adalah tahapan penggunaan aplikasi untuk di uji coba untuk diketahui apakah aplikasi ini telah mencapai tujuan dan kebutuhan dari penelitian. Pengujian yang dilakukan menggunakan beberapa kondisi sebagai berikut:

3.3.5.1. Pengujian dengan Kondisi Kecepatan Kendaraan

Proses pengujian aplikasi deteksi rambu lalu lintas akan dilakukan pada beberapa rekaman perjalanan dengan kecepatan kendaraan yang berbeda. Proses pengujian berdasarkan kecepatan akan dilakukan pada hasil rekaman pada tabel 3.1, dimana kecepatan yang akan diujikan adalah 20km/jam, 30km/jam, dan 40km/jam pada jalan protokol yaitu jalan R.A.Kartini. Pengujian ini dilakukan untuk mengetahui pengaruh kecepatan kendaraan dengan hasil aplikasi deteksi rambu lalu lintas.

3.3.5.2. Pengujian dengan Kondisi Pencahayaan

Proses pengujian aplikasi deteksi rambu lalu lintas selanjutnya berdasarkan kondisi pencahayaan yang berbeda. Uji berdasarkan kondisi pencahayaan yang berbeda dilakukan untuk mengetahui pengaruh kondisi pencahayaan pada aplikasi deteksi rambu lalu lintas. Pengujian dilakukan pada video rekaman yang sama dengan kondisi pencahayaan yang berbeda, yaitu dengan menggunakan rekaman yang diredupkan tingkat kecerahannya

Selain dari pengujian diatas, dilakukan juga pengujian dengan menggunakan rambu peringatan sebagai nilai pembanding pada hasil uji.