

# Deteksi Kantuk Melalui Citra Wajah Menggunakan Metode *Gray Level Co-occurrence Matrix* (GLCM) dan Klasifikasi *Support Vector Machine* (SVM)

## Drowsiness Detection Through Face Image Using Gray Level Co-Occurrence Matrix (GLCM) Method and Support Vector Machine (SVM) Classifier

Noni Charimmah, Ervi Lanovia, Koredianto Usman, Ledy Novamizanti

S1 Teknik Telekomunikasi, Telkom University

Jl. Telekomunikasi Terusan Buahbatu, Bandung, Indonesia, 40288

nonicharimmah@student.telkomuniversity.ac.id\*, ervilanovia@student.telkomuniversity.ac.id,

korediantousman@telkomuniversity.ac.id, ledyaldn@telkomuniversity.ac.id

**Abstrak** – Tingginya angka kecelakaan di jalan raya menuntut perkembangan teknologi terkini agar dapat mencegah angka tersebut meningkat. Kecelakaan akibat pengemudi yang mengantuk merupakan penyumbang angka kecelakaan tertinggi. Salah satu pencegahan terhadap kecelakaan di jalan raya akibat mengantuk adalah dengan membuat suatu sistem deteksi kantuk melalui pengolahan citra. Sistem tersebut mengolah video yang di rekam untuk mengambil bagian mata dan mulut. Video diambil per-frame dan dilakukan face detection, eye detection, dan mouth detection. Proses tersebut dilakukan dengan menggunakan algoritma Viola-Jones. Setelah diperoleh citra mata dan mulut, dilakukan ekstraksi ciri menggunakan metode Gray Level Co-occurrence Matrix (GLCM). Keluaran dari proses ekstraksi yaitu ciri saat mata dan mulut terbuka atau tertutup. Selanjutnya, klasifikasi keadaan mata dan mulut menggunakan Support Vector Machine (SVM). Sistem akan menghasilkan peringatan ketika pengemudi terdeteksi mengantuk.

**Kata Kunci:** Kantuk, face detection, Viola-Jones, Gray Level Cooccurrence Matrix, Support Vector Machine

**Abstract** – The high number of accidents on the highway demands the latest technological developments in order to prevent that number from increasing. Accidents due to sleepy motorists are the highest contributor to accidents. One of the prevention of accidents on the highway due to drowsiness is to make a drowsiness detection system through image processing. The system processes the recorded video to take part of the eyes and mouth. Videos are taken per-frame and face detection, eye detection, and mouth detection are performed. The process is carried out using the Viola-Jones algorithm. After obtaining eye and mouth images, feature extraction is performed using the Gray Level Co-occurrence Matrix (GLCM) method. The output from the extraction process is characterized when the eyes and mouth are open or

closed. Furthermore, the classification of the state of the eyes and mouth using the Support Vector Machine (SVM). The system will produce a warning when the rider is detected sleepy.

**Keywords:** Drowsiness, face detection, Viola-Jones, Gray Level Cooccurrence Matrix, Support Vector Machine

## 1. Pendahuluan

Kondisi tubuh pengendara merupakan faktor penting dalam penentuan tingkat keselamatan pengendara. Dengan kondisi tubuh yang baik, seorang pengemudi akan mampu mengendarai kendaraan sehingga dapat sampai di tujuan dengan selamat. Pada tahun 2014 terjadi sebanyak 739 kasus kecelakaan lalu lintas di Indonesia yang disebabkan oleh kantuk pada pengemudi [1]. Jumlah kasus kecelakaan tersebut menuntut inovasi untuk mencegah meningkatnya jumlah kecelakaan di jalan raya akibat mengantuk. Pembuatan sistem deteksi kantuk adalah salah satu cara yang dapat dilakukan untuk mengurangi angka kecelakaan tersebut.

Sistem deteksi kantuk pada pengemudi merupakan teknologi yang telah cukup banyak diteliti. Sistem ini menitikberatkan pada penerapan pengolahan sinyal citra. Ada beberapa metode yang dapat digunakan. Metode-metode tersebut dapat dilakukan menggunakan parameter biologis maupun pendekatan pada kondisi kendaraan [2]. Metode yang dilakukan dengan parameter biologis antara lain *electroencephalogram* (EEG) dan *electrocardiogram* (ECG). Pendekatan pada kondisi kendaraan dapat dilihat dari kebiasaan pengemudi seperti berkedip, menguap, dsb. Salah satu teknik pengolahan citra yang dapat diterapkan pada pendekatan tersebut, yaitu *Gray Level Co-occurrence Matrix* (GLCM). GLCM adalah sebuah matriks yang jumlah kolom dan barisnya sama dengan tingkat keabuan (G) pada citra [3].

Beberapa penelitian terkait sistem deteksi kantuk pada pengemudi telah gencar dilakukan. Szidonia Lefkovits dkk. telah melakukan penelitian terkait dengan memanfaatkan filter gabor untuk mendeteksi dan mengidentifikasi keadaan mata pengemudi [4]. Sistem ini mampu mengidentifikasi kantuk secara akurat meskipun belum dapat diaplikasikan pada pengemudi berkacamata. Selain itu, Maninder Kahlon dan Subramaniam Ganesan telah membuat sistem deteksi kantuk dengan memanfaatkan *binary eyes image data* [5]. Sistem yang dirancang mengidentifikasi keadaan mata (terbuka atau tertutup) dengan memanfaatkan konsep transformasi *grayscale to binary image*. Namun, keakuratan deteksi yang dihasilkan dipengaruhi oleh efek cahaya dan posisi pengemudi sehingga nilai threshold yang dipakai belum adaptif. Selanjutnya, penelitian terkait sistem deteksi kantuk pada pengemudi juga telah dilakukan oleh Jun-Juh Yan dkk. dengan memanfaatkan *Percentage of Eyes Closure* (PERCLOS) dan pengolahan citra tipe *grayscale* [6]. Sistem yang dirancang dapat diaplikasikan pada pengemudi berkacamata namun sangat riskan jika diterapkan pada pengemudi yang berkulit gelap. Pada 2009, Brojeshwar Bowmick dan K. S. Chidanan Kumar telah melakukan penelitian menggunakan kamera *Infra Red* (IR) yang memanfaatkan metode SVM untuk mengklasifikasi keadaan mata [7]. Sistem yang dirancang juga menggunakan teknik GLCM dengan memanfaatkan fitur kontras sehingga keadaan mata mampu diindikasikan dengan baik.

Pada penelitian ini penulis merancang suatu sistem deteksi kantuk melalui citra wajah. Sistem ini memanfaatkan data video dalam ruang warna RGB. Dari video tersebut akan diambil frame-frame sehingga akan didapatkan citra mata dan mulut yang akan diekstraksi cirinya menggunakan metode GLCM dan *Support Vector Machine* (SVM). Metode GLCM berbasis pada pengekstraksian ciri statistik orde dua yang dapat mengenali tekstur citra lebih baik dibandingkan dengan pengekstraksian ciri statistik orde satu [8]. Klasifikasi SVM digunakan karena memiliki performansi yang baik dengan hasil klasifikasi yang kuat walaupun tidak disertai dengan tahapan *pre-processing* dan mampu mengurangi waktu pemrosesan citra [9].

## 2. Metode Penelitian

Penelitian dilakukan menggunakan hasil perancangan sistem yang telah dibuat. Sistem yang dirancang merupakan hasil dari studi literatur yang telah dilakukan secara mendalam mengenai pengolahan citra, *face detection*, GLCM, dan SVM

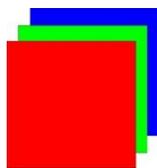
## 2.1. Pengolahan Citra

Citra adalah sebuah representasi dalam penggambaran suatu objek yang disimpan dalam bentuk visual. Pada perkembangannya, citra dibedakan menjadi citra digital dan analog. Contoh citra analog, yaitu gambar yang terdapat pada televisi, hasil CT scan, dan XRay. Sedangkan contoh citra digital adalah gambar yang biasa disimpan di memori telepon, laptop, dan lain sebagainya. Citra terdiri dari beberapa jenis ruang warna, yaitu RGB, *grayscale*, HSV, YCbCr, dan Biner. Namun, pada penelitian ini jenis citra yang digunakan adalah RGB dan *grayscale*.

Pengolahan sinyal citra telah berkembang seiring berjalannya waktu. Pengolahan ini bertujuan agar mendapatkan citra yang diinginkan melalui *object tracking*, *gender recognition*, *face recognition*, dan masih banyak lagi. Pengolahan ini tentunya bergantung pada tujuan pengolah. Pada pengolahan sinyal citra, umumnya citra akan perlu melewati proses-proses tertentu. Proses yang dilakukan tergantung pada tujuan pengolahan citra tersebut. Proses tersebut dapat berupa pengubahan ruang warna citra (misal: RGB menjadi *grayscale*), *resizing* (pengubahan ukuran citra), dan sebagainya. Proses lainnya dapat juga berupa ekstraksi, pengklasifikasian, dan lain sebagainya.

## 2.2. Citra Red, Green, Blue (RGB)

Citra *Red, Green, Blue* atau RGB adalah salah satu jenis ruang warna yang ada pada citra. Ruang warna ini terdiri dari tiga lapisan, yaitu *red* (R), *green* (G), dan *blue* (B). Hal ini dapat dilihat pada Gambar 1. Pada ruang warna ini, ketiga lapisan warna memiliki nilai masing-masing yang menunjukkan intensitas warna. Ruang warna ini terdiri dari delapan bit dengan jumlah warna maksimum yang dapat dipakai, yaitu 256 warna. Dengan menggunakan pengkodean delapan bit tersebut, maka intensitas warna pada tipe ruang warna ini akan memiliki rentang angka nol (0) hingga 255. Pencampuran intensitas warna tertentu dari *red*, *green*, dan *blue* akan menghasilkan warna baru dengan intensitas yang berbeda-beda.



Gambar 1. Lapisan pada citra RGB.

## 2.3. Citra Grayscale

Berbeda dengan citra RGB, yang terdiri dari tiga lapisan, yaitu *red* (R), *green* (G), dan *blue* (B), citra *grayscale* hanya terdiri dari satu lapisan intensitas warna, yaitu abu-abu. Hal ini dikarenakan pada citra *grayscale* memiliki nilai yang sama pada lapisan *red*, *green*, dan *blue*. Citra *grayscale* juga terdiri dari delapan (8) bit dengan nilai intensitas berada pada rentang nol (0) hingga 255. Pada jenis ruang warna ini, intensitas terendah (0) dari citranya akan didefinisikan dengan warna hitam dan yang tertinggi akan didefinisikan dengan warna putih (255).

Pada simulasi yang digunakan, sintaks yang diperlukan untuk konversi dari ruang warna RGB ke *grayscale* adalah `rgb2gray`. Sebagai contoh, perintah `I = rgb2gray(RGB)` yang akan mengubah citra RGB menjadi *grayscale* `I` [5]. Formulasi untuk melakukan konversi dari ruang warna RGB ke *grayscale* adalah sebagai berikut:

$$I = 0,2989R + 0,5870G + 0,1141B \quad (1)$$

dengan `I` menyatakan citra grayscale hasil konversi, `R` menyatakan nilai komponen merah (red), `G` menyatakan nilai komponen hijau (green), dan `B` menyatakan nilai komponen biru (blue).

## 2.4. Face, Eyes, dan Mouth Detection

*Face detection* merupakan sebuah proses untuk mendeteksi bagian wajah pada citra. Proses ini dilakukan untuk mendapatkan bagian wajah. Dari tahap ini, bagian wajah dan non wajah akan didapatkan. Bagian wajah akan digunakan untuk tahap selanjutnya. Bagian selain wajah akan

diabaikan ataupun dibuang. *Face detection* dapat dilakukan dengan berbagai metode. Pada penelitian ini, proses *face detection* akan dilakukan dengan memanfaatkan algoritma *Viola-Jones* [10].

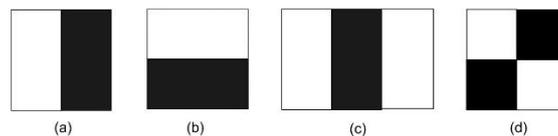
Sama halnya dengan *face detection*, *eye* dan *mouth detection* pada citra merupakan sebuah proses untuk mendeteksi bagian mata dan mulut. Proses ini diperlukan agar fokus pengamatan dapat dilakukan dengan lebih baik. Dari proses ini, akan didapati bagian mata dan non mata serta mulut dan non mulut sehingga pada penelitian akan dapat digunakan untuk proses selanjutnya. *Eye detection* dilakukan dengan cara mendeteksi bagian mata dan membuang bagian selain mata sedangkan *mouth detection* dilakukan dengan mendeteksi bagian mulut dan membuang bagian selain mulut. Proses ini dapat dilakukan dengan memanfaatkan berbagai cara. Cara-cara tersebut diantaranya, memanfaatkan *edge detection*, menggunakan EAR, memanfaatkan metode ekstraksi ciri, dan sebagainya. Dengan algoritma yang sama dengan *face detection*, pada penelitian ini, *eyes* dan *mouth detection* akan dilakukan dengan memanfaatkan algoritma *Viola Jones*.

## 2.5. Algoritma *Viola-Jones*

Algoritma *Viola-Jones* telah diperkenalkan pada 2001 oleh Paul Viola dan Michael Jones. Algoritma ini merupakan salah satu yang paling baik untuk melakukan *face objects* dan *upper body detection*. *Face objects detection* tersebut dapat berupa wajah, hidung, mulut, mata, dan pupil. Pada simulasi yang digunakan, algoritma ini menggunakan metode `vision.CascadeObjectDetector`. Algoritma *Viola-Jones* terdiri dari tiga tahapan untuk deteksi bagian wajah, yaitu *Haar-like Features*, *integral image*, *AdaBoost*, dan *Cascade Classifiers*.

### 2.5.1. *Haar-like Features*

*Haar-like features* digunakan untuk mendeteksi fitur dari suatu citra. Ada beberapa fitur pada Haar. Fitur tersebut terdiri dari susunan dua persegi panjang, tiga persegi panjang, dan empat persegi panjang yang dibagi ke dalam empat fitur. Keempat fitur tersebut terlihat seperti pada Gambar 2 [11]. Pada fitur Haar, terdapat dua piksel, yaitu piksel hitam dan putih yang dapat digambarkan pula dalam bentuk matriks. Piksel-piksel tersebut menggambarkan daerah gelap dan daerah terang pada citra dua dimensi. Daerah gelap dan terang tersebut didefinisikan dengan nilai interval tinggi (+1) dan rendah (-1).



Gambar 2. Haar features. (a) Matriks yang bernilai  $[-1 \quad 1]$ , (b) Matriks yang bernilai  $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ , (c) Matriks yang bernilai  $[-1 \quad 1 \quad -1]$ , (d) Matriks yang bernilai  $\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$ .

Haar pada dasarnya merupakan produk skalar dari citra dan Haar template. Maka, untuk mendapatkan nilai fitur pada fitur Haar, diperlukan perhitungan pada piksel-piksel yang ada. Perhitungan tersebut dilakukan dengan mengurangkan nilai piksel di daerah terang dengan nilai piksel di daerah gelap.

### 2.5.2. *Integral Image*

Pada sebuah *integral image*, nilai pada piksel ( $x$ ) didapatkan dari penjumlahan dari piksel yang ada di atas dan di sebelah kiri dari piksel ( $x$ ). Penjumlahan pada persegi panjang dari fitur Haar akan dapat dilakukan menggunakan empat susun referensi. Dari jumlah tersebut, maka penjumlahan untuk perbedaan pada dua buah persegi panjang akan dapat dilakukan dengan memanfaatkan delapan referensi. Karena fitur dua persegi panjang menyatakan penjumlahan persegi panjang yang saling berdampingan, maka dapat dilakukan penjumlahan pada enam susun

referensi, delapan pada fitur tiga persegi panjang, dan sembilan pada fitur empat persegi panjang. Formulasi untuk integral image adalah sebagai berikut [10]:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{2}$$

dengan  $ii(x,y)$  menyatakan integral image dan  $i(x',y')$  menyatakan citra asli.

**2.5.3. AdaBoost**

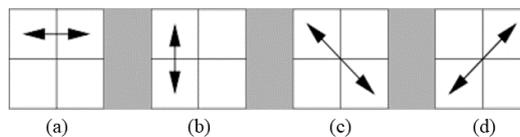
*AdaBoost* adalah sebuah metode *machine learning* untuk mendeteksi wajah. Sebelumnya, pada fitur Haar, jika terdapat kombinasi pada kelima fitur yang ada, fitur-fitur lain yang akan terbentuk akan mencapai 160000+ fitur. *AdaBoost* melakukan eliminasi terhadap fitur yang tidak relevan yang dibentuk pada fitur Haar. sehingga dapat berguna untuk mendeteksi wajah. *AdaBoost* akan mengidentifikasi semua 160000+ fitur dan setelah itu akan memberikan bobot ke semua fitur tersebut. Selanjutnya, fitur-fitur yang terpilih akan dianggap baik untuk dimasukkan pada proses evaluasi jika fitur-fitur tersebut menghasilkan performansi yang lebih baik dibanding random guessing (mendeteksi lebih dari setengah kasus). Fitur-fitur tersebut disebut sebagai *weak classifier*. Pada kombinasi linier *weak classifiers*, *AdaBoost* akan menghasilkan sebuah *strong classifier*.

**2.5.4. Cascade Classifiers**

*Cascade classifiers* digunakan untuk mengkombinasikan banyak *features* secara efisien. *Cascade classifier* akan melakukan klasifikasi secara bertingkat. Pada setiap tingkatnya, klasifikasi dilakukan menggunakan beberapa *strong classifier*. Masukan pada tiap tingkatan yang ada pada *cascade classifiers* disebut dengan *sub window*. Klasifikasi pada setiap tingkatnya akan menyatakan apakah *sub window* yang diberikan mengandung wajah atau tidak. Jika tidak, maka *sub window* tersebut akan dibuang sehingga tidak akan memasuki tingkatan selanjutnya.

**2.6. Gray Level Co-occurrence Matrix (GLCM)**

GLCM merupakan salah satu metode identifikasi ciri yang pertama kali diperkenalkan oleh Haralick pada 1973. Metode ini mengekstraksi citra untuk mengidentifikasi tekstur sehingga dapat digunakan untuk membedakan satu citra dengan yang lainnya. GLCM akan membentuk sebuah matriks *co-occurrence* terlebih dahulu lalu akan mengekstraksi ukuran statistik dari matriks yang telah dibentuk. Metode GLCM bekerja dengan membandingkan hubungan spasial antar piksel. Perbandingan ini dilakukan dengan cara mencari derajat keabuan setiap dua piksel yang terpisah sejauh  $d$  dan berada pada sudut  $\theta$  yang tetap. Besar sudut tersebut adalah  $\theta = 0^\circ, 45^\circ, 90^\circ, \text{ dan } 135^\circ$ . Gambar 3 menunjukkan arah perhitungan menggunakan GLCM [13].



Gambar 3. Arah perhitungan menggunakan GLCM. (a) Arah perhitungan untuk  $\theta = 0^\circ$ . (b) Arah perhitungan untuk  $\theta = 90^\circ$ . (c) Arah perhitungan untuk  $\theta = 135^\circ$ . (d) Arah perhitungan untuk  $\theta = 45^\circ$ .

Dasar dari GLCM adalah pada matriks *co-occurrence*. Matriks ini berdimensi  $N_g$  yang menyatakan tingkat keabuan suatu citra. Unsur  $[i,j]$  pada matriks didapatkan dengan menghitung jumlah piksel dengan nilai  $i$  berbatasan dengan sebuah piksel dengan nilai  $j$  dan kemudian membagi matriks dengan total jumlah yang dibuat oleh perbandingan seperti itu. Setiap masukan lalu dianggap sebagai probabilitas bahwa piksel dengan nilai  $i$  akan berbatasan dengan piksel pada nilai  $j$  [13]. Matriks *co-occurrence* dapat dinyatakan sebagai berikut:

$$G = \begin{bmatrix} p(1,1) & p(1,2) & \dots & p(1, N_g) \\ p(2,1) & p(2,2) & \dots & p(2, N_g) \\ \vdots & \vdots & \ddots & \vdots \\ p(N_g,1) & p(N_g,2) & \dots & p(N_g, N_g) \end{bmatrix} \quad (3)$$

Fitur dari metode ekstraksi ciri GLCM yang digunakan, yaitu [13-16]:

1. *Homogeneity* atau *Inverse Different Moment*

*Homogeneity*, yaitu mengukur kedekatan distribusi elemen dalam GLCM ke GLCM diagonal. Formulasi *homogeneity* dapat dinyatakan sebagai berikut:

$$IDM = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j) \quad (4)$$

dengan  $p(i,j)$  menyatakan elemen matriks *co-occurrence* baris  $i$  dan kolom  $j$ .

2. *Correlation*

Fitur ciri *correlation* mengukur korelasi piksel satu dengan tetangganya yang berada dalam satu citra. Formulasi *correlation* dapat dinyatakan sebagai berikut:

$$COR = \sum_i \sum_j \frac{(i,j)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (5)$$

dengan  $\mu_x$  menyatakan nilai rata-rata elemen pada kolom matriks  $p(i,j)$ ,  $\mu_y$  menyatakan nilai rata-rata elemen pada baris matriks  $p(i,j)$ ,  $\sigma_x$  menyatakan standar deviasi elemen pada kolom matriks  $p(i,j)$ , dan  $\sigma_y$  menyatakan standar deviasi elemen pada baris matriks  $p(i,j)$ .

3. *Energy* atau *Angular Second Moment*

Energi bernilai satu (1) untuk citra konstan. Energi juga disebut sebagai *uniformity*, *uniformity of energy*, dan *angular second moment*. Formulasi *energy* dapat dinyatakan sebagai berikut:

$$ASM = \sum_i \sum_j p(i,j)^2 \quad (6)$$

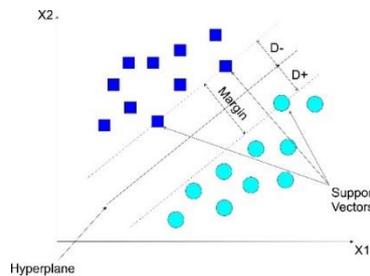
4. *Contrast*

Pengekstraksian fitur *contrast* adalah dengan mengukur intensitas antara sebuah piksel dengan tetangganya yang ada dalam satu citra. Formulasi *contrast* dapat dinyatakan sebagai berikut:

$$CON = \sum_i \sum_j |i - j|^2 p(i,j) \quad (7)$$

2.7. *Support Vector Machine (SVM)*

Metode klasifikasi SVM pada dasarnya dilakukan dengan mencari hyperplane antara kelas-kelas sehingga dapat membedakan kelas satu dengan yang lainnya. Skema SVM dapat dilihat pada Gambar 4.



Gambar 4. Skema *Support Vector Machine (SVM)*.

*Hyperplane* adalah batas keputusan yang berfungsi untuk pendukung klasifikasi pada data. *Hyperplane* didapatkan dari batas pemisah yang paling baik untuk sebuah SVM dan mengandung *margin* terbesar antara dua kelas. *Margin* atau *weight vector* yang dimaksud adalah jarak antara *hyperplane* dan data terdekat yang disebut juga dengan *pattern* atau *support vector*. *Support vector* merupakan batas dari lempengan paralel [17]. Mencari nilai *hyperplane* pada kasus linear

dan non linear berbeda. Pada kasus *hyperplane* yang linear, formulasi yang dapat digunakan adalah sebagai berikut:

$$g(\vec{x}) = \vec{w}x + w_0 \tag{8}$$

dengan  $\vec{w}$  adalah *weight vector* atau *margin*,  $x$  adalah koordinat dari *support vectors*, dan  $w_0$  adalah konstanta.

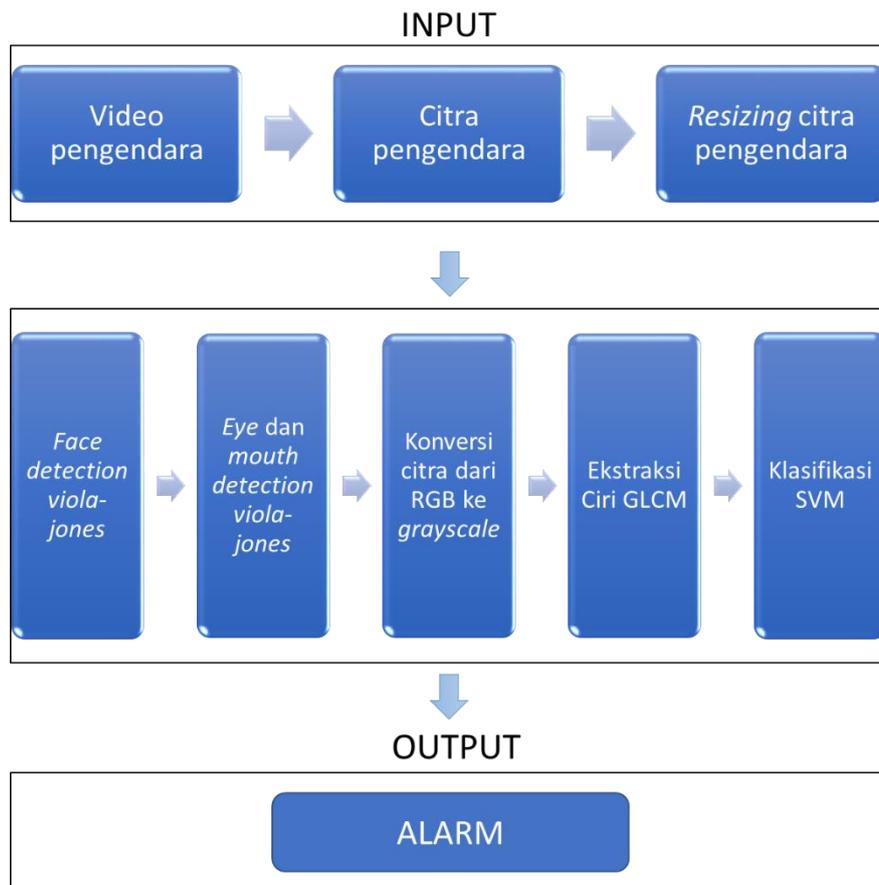
Pada SVM terdapat dua kelas data, yaitu kelas (-1) dan kelas (+1). Kelas (-1) menunjukkan kelas yang condong dekat dengan sisi negatif pada diagram. Pada Gambar 4, kelas (-1) dinyatakan dengan data berbentuk kotak persegi dan kelas (+1) dinyatakan dengan data berbentuk bulat. Sedangkan  $D^-$  dan  $D^+$  menunjukkan jarak antara *hyperplane* dengan batas pemisah pada *support vectors*. Untuk mendapatkan nilai *weight vector* atau *margin* yang digunakan pada (8), formulasi dapat dinyatakan sebagai berikut:

$$\vec{w} = (x_1, x_2)_{+1} - (x_1, y_2)_{-1} \tag{9}$$

dengan  $(x_1, x_2)_{+1}$  merupakan koordinat *support vectors* pada kelas (+1) dan  $(x_1, y_2)_{-1}$  menyatakan koordinat *support vectors* pada kelas (-1).

**2.8. Perancangan Sistem**

Sistem deteksi yang dirancang penulis terdiri dari tiga bagian utama, yaitu input, proses, dan output. Masukan berupa citra pengendara yang telah diambil dari *frame* video dan selanjutnya akan diproses dengan keluaran berupa alarm. Proses ekstraksi menggunakan metode GLCM. Sedangkan metode klasifikasi menggunakan SVM. Gambar 5 merupakan rancangan blok diagram sistem deteksi kantuk.



Gambar 5. Blok diagram sistem deteksi kantuk.

Input diambil menggunakan kamera. Masukan berupa video pengendara dengan format .mp4 dan resolusi  $1920 \times 1080$  piksel. Data masukan akan diambil per frame untuk diproses. Sebelum masuk ke bagian proses, akan dilakukan *resizing* pada frame-frame yang masuk. Selanjutnya, citra akan diolah menggunakan software. Proses pengolahan citra pada sistem ini terdiri dari lima tahapan, yaitu *face detection*, *eyes*, dan *mouth detection*, konversi ruang warna dari RGB ke *grayscale*, ekstraksi ciri, serta klasifikasi keadaan mata dan mulut. Pada akhir sistem, keluaran yang dihasilkan berupa alarm yang menyala ketika pengendara mobil terdeteksi mengantuk dan alarm akan tetap mati jika sistem tidak mendeteksi apapun.

### 2.8.2. Pengambilan Video Pengendara

Proses pengambilan video pengendara dilakukan menggunakan kamera pada *handphone*. Resolusi yang digunakan mencapai  $1920 \times 1080$  piksel dengan tipe RGB. Kamera diletakkan pada jarak 30 cm di bagian depan atas pengendara agar mampu menghasilkan video dengan persentase wajah sebesar minimal 90% dari total *frame* yang dihasilkan. Dari video yang didapatkan, *frame* akan diambil kemudian dijadikan sebagai input dari sistem dan selanjutnya akan diproses sebagai citra pengendara. Citra pengendara yang dihasilkan dapat seperti pada Gambar 6.



Gambar 6. Citra pengendara yang dihasilkan.

### 2.8.3. Pre-processing

Tahapan *pre-processing* citra pengendara meliputi citra pengendara yang digunakan sebagai *input* adalah *frame* dari video yang telah diperoleh dengan ruang warna RGB dan resolusi  $1920 \times 1080$  piksel. Setelah itu, *resizing* citra pengendara dilakukan. Citra *input* (citra pengendara) diubah ukurannya sesuai dengan data latih yang akan digunakan. Dari hasil *pre-processing*, citra yang diperoleh adalah hasil *resize* dari citra asli pengendara.

### 2.8.4. Processing Citra Pengendara

Citra pengendara yang telah dilakukan *resizing* akan masuk ke tahapan *processing*. Bagian inilah yang akan menentukan tingkat keberhasilan deteksi yang dilakukan. Bagian ini juga akan menentukan hasil dari deteksi, apakah alarm akan berbunyi atau tidak. Gambar 7 merupakan blok diagram sistem secara umum.

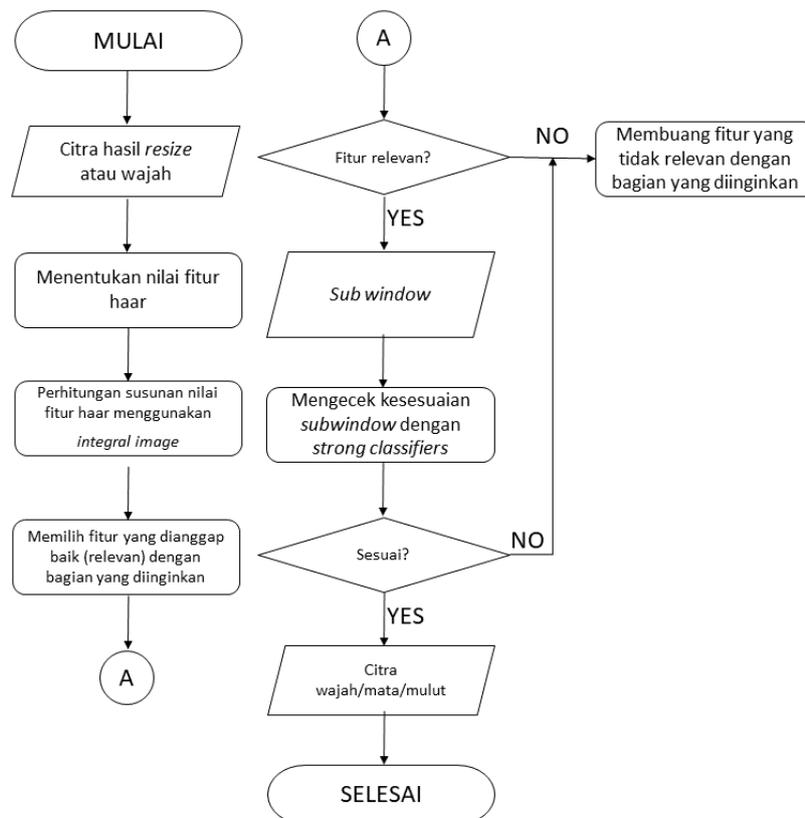


Gambar 7. Blok diagram sistem secara umum.

### 2.8.5. Face, Eye, dan Mouth Detection Menggunakan Viola-Jones

Pada alur kerjanya, untuk mendapatkan bagian wajah, mata, dan mulut menggunakan *Viola-Jones* terdiri dari tahapan yang sama. Hal ini dapat dilihat pada alur bagian *face*, *eye*, dan *mouth detection* menggunakan *Viola-Jones* seperti pada Gambar 8. Perbedaan antara ketiga deteksi ini ada pada *input* yang digunakan dan *output* yang dihasilkan. Pada *face detection*, tentunya input yang digunakan, yaitu citra pengendara hasil akuisisi awal. Dari proses tersebut, *output* yang akan

dihasilkan berupa citra wajah. Sedangkan *eye* dan *mouth detection* menggunakan *output* dari *face detection* tersebut sebagai input yang setelah diproses akan didapatkan *output* berupa citra mata dan mulut.

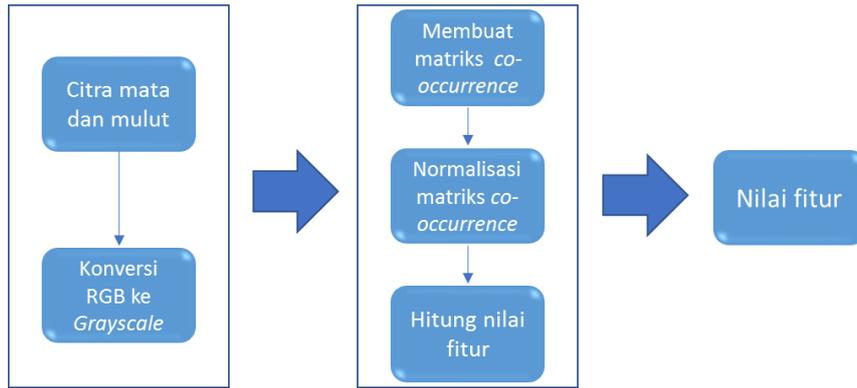


Gambar 8. Alur bagian *face*, *eye*, dan *mouth detection* menggunakan *Viola-Jones*.

Berdasarkan Gambar 8, bahwa untuk *face detection*, citra input adalah citra yang telah dilakukan *resizing* pada tahap *pre-processing*. Sedangkan untuk *eye* dan *mouth detection*, citra input adalah citra hasil dari *face detection*. Sistem menentukan nilai fitur haar pada citra input dengan menghitung nilai-nilai piksel yang ada. Selanjutnya, dilakukan perhitungan susunan nilai fitur haar dengan *integral image*. Pada tahap ini, dilakukan penjumlahan pada persegi panjang dari fitur Haar. Penjumlahan tersebut dapat dilakukan dengan menggunakan empat susun referensi. Selanjutnya, sistem memilih fitur yang dianggap relevan, yaitu yang memiliki bagian paling sesuai dengan bagian yang diinginkan. Dalam hal ini, wajah, mata, dan mulut. Jika fitur dianggap relevan, maka fitur masuk ke tahapan selanjutnya dan akan disebut sebagai *sub window*. Fitur yang dianggap tidak relevan akan dibuang. Kemudian, sistem mengecek kesesuaian *sub window* dengan beberapa classifier yang merupakan *strong classifiers*. Apabila *sub window* tidak memenuhi kesesuaian di salah satu tingkat, maka *sub window* akan dibuang. Jika sesuai, maka akan masuk ke proses selanjutnya. Output dari proses *face detection* adalah citra wajah sedangkan untuk *eye* dan *mouth detection* adalah citra mata dan mulut.

**2.8.6. Ekstraksi Ciri Bagian Mata dan Mulut**

Ekstraksi ciri bagian mata dan mulut pada penelitian ini menggunakan metode GLCM. Input yang digunakan berupa citra mata dan mulut yang telah diperoleh menggunakan *Viola-Jones*. Pada sistem ini, input akan diubah ruang warnanya menjadi tipe *grayscale* untuk mempermudah dalam proses ekstraksi ciri menggunakan metode GLCM. Tahapan ekstraksi ciri bagian mata dan mulut dapat dilihat pada Gambar 9.

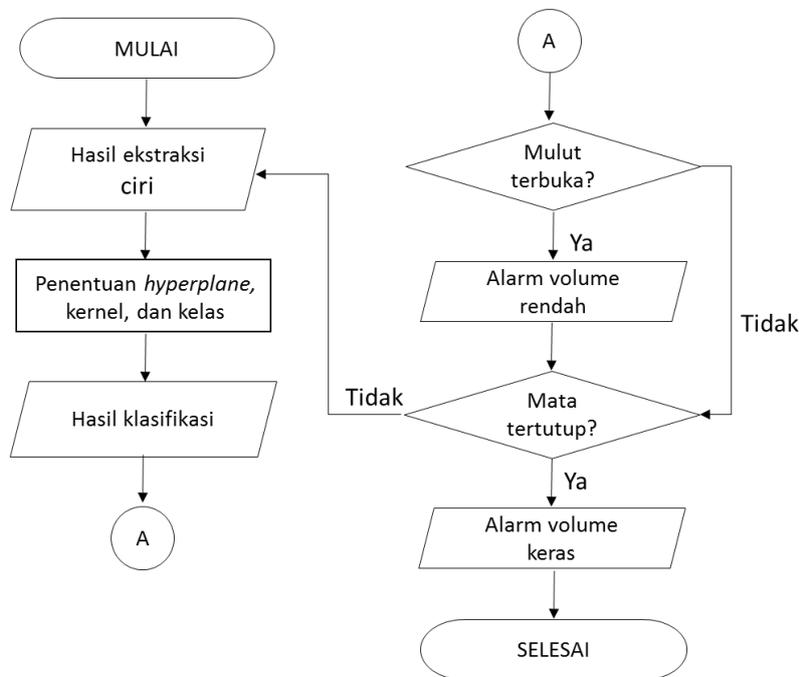


Gambar 9. Blok diagram proses ekstraksi ciri mata dan mulut.

Input dari tahapan ini adalah citra mata dan mulut yang telah dihasilkan dari proses *eye* dan *mouth detection*. Ruang warna citra input akan dikonversi ke ruang warna *grayscale* untuk memudahkan dalam penentuan nilai keabuan. Selanjutnya, akan dibuat matriks *co-occurrence* dengan membandingkan nilai derajat keabuan piksel satu dengan piksel tetangganya. Setelah mendapatkan matriks *co-occurrence*, harus dilakukan normalisasi yang berfungsi untuk menyederhanakan nilai matriks yang dihasilkan. Nilai fitur yang akan dihitung adalah *homogeneity*, *correlation*, *energy*, dan *contrast*.

**2.8.7. Klasifikasi Keadaan Mata dan Mulut**

Setelah mendapatkan tekstur mata akan dilakukan klasifikasi yang berfungsi untuk memisahkan kondisi mata ke dalam dua kelas. Kedua kelas tersebut adalah kondisi mata terbuka dan kondisi mata tertutup. Selanjutnya, kondisi mulut juga akan diklasifikasikan menjadi mulut terbuka dan tertutup. Klasifikasi ini menggunakan SVM. Alur sistem klasifikasi menggunakan SVM digambarkan pada Gambar 10.



Gambar 10. Alur sistem klasifikasi menggunakan SVM.

*Output* yang dihasilkan terdiri dari dua jenis alarm, yaitu dengan suara keras dan dengan suara ringan. Apabila terdeteksi sebanyak lebih dari enam *frame* mengandung informasi mulut terbuka (menguap), maka sistem akan mengeluarkan alarm dengan suara ringan untuk memperingati. Hal tersebut bertujuan untuk mengingatkan pengendara untuk berhati-hati karena tubuhnya mulai kelelahan. Apabila terdeteksi sebanyak lebih dari enam *frame* mengandung informasi mata tertutup, maka akan disimpulkan bahwa pengendara ada pada kondisi mengantuk. Pada kondisi tersebut, alarm akan berbunyi dengan suara yang keras. Alarm tersebut berfungsi untuk mengingatkan pengendara bahwa pengendara harus meningkatkan kewaspadaan karena kondisi tubuh yang sudah tidak baik untuk memegang kendali pada kendaraan. Kondisi terbuka atau tertutupnya mata akan dilihat dari perbedaan tekstur mata yang sebelumnya telah diekstraksi. Selain dari kedua kondisi yang telah disebutkan, alarm akan mati.

### 3. Hasil dan Pembahasan

Parameter pengujian yang diamati pada penelitian ini adalah *accuracy* dan lamanya komputasi yang dilakukan oleh sistem yang dirancang. Parameter-parameter tersebut adalah sebagai berikut:

#### 1. *Accuracy*

Parameter *accuracy* akan menguji data dan membandingkan jumlah data yang teridentifikasi dengan benar dengan total data yang diuji oleh sistem pada hasil pengujian. Formulasi keakuratan pada sistem adalah sebagai berikut:

$$Accuracy = \frac{Total\ data\ benar}{Jumlah\ total\ data} \times 100\% \quad (10)$$

Dalam mengukur keakuratan pada hasil sistem, *error* juga perlu diperhitungkan. *Error* tersebut diukur dengan membandingkan total data yang teridentifikasi dengan salah dengan total data yang diujikan pada sistem. Formulasi dari *error* (kesalahan sistem) adalah sebagai berikut:

$$Error = \frac{Total\ data\ salah}{Jumlah\ total\ data} \times 100\% \quad (11)$$

#### 2. Waktu Komputasi

Parameter waktu komputasi diukur untuk menentukan lamanya waktu yang dibutuhkan sistem yang dirancang untuk mengeksekusi perintah. Semakin singkat waktu yang dibutuhkan sebuah sistem untuk melakukan komputasi, maka semakin baik sistem tersebut berjalan. Namun sebaliknya, semakin banyak waktu yang dibutuhkan untuk menyelesaikan sistem, maka sistem tersebut belum optimal.

Hasil dari perancangan deteksi kantuk semakin meningkat seiring banyaknya penelitian terkait hal tersebut. Akurasi yang tinggi akan membuat sistem yang dirancang lebih baik lagi dan semakin kecil kesalahan yang dilakukan oleh sistem. Hal tersebut akan mengakibatkan sistem dapat dipercaya hasilnya untuk melakukan eksekusi yang diinginkan. Selain itu, waktu komputasi yang singkat juga akan membuat sistem lebih efisien dalam melakukan pendeteksian. Dengan begitu, sistem yang dirancang akan dapat bermanfaat untuk diterapkan di kehidupan masyarakat.

### 4. Kesimpulan

Penelitian ini mengusulkan sistem deteksi akan diterapkan pada pengendara mobil. Sistem deteksi diharapkan akan mampu diterapkan pada pengendara berkacamata maupun tidak berkacamata. Namun, pada sistem yang dirancang diperlukan adanya pengembangan karena deteksi sangat bergantung pada intensitas cahaya yang terkandung pada citra yang digunakan. Dengan dirancangnya sistem deteksi ini diharapkan mampu mengurangi tingkat kecelakaan akibat pengendara yang mengantuk.

### Referensi

- [1] Adminkorlantas, "Mabes Polri : Angka Kecelakaan Pemudik 2015 Turun," *NTMC Polri*, 2015. [Online]. Available: <http://ntmcpolri.info/home/mabes-polri-angka-kecelakaan-pemudik-2015-turun/>. [Accessed: 23-Feb-2019].

- [2] I. Gupta, N. Garg, A. Aggarwal, N. Nepalia, and B. Verma, "Real-Time Driver's Drowsiness Monitoring Based on Dynamically Varying Threshold," *2018 11th Int. Conf. Contemp. Comput. IC3 2018*, pp. 1–6, 2018.
- [3] P. Mohanaiah, P. Sathyanarayana, and L. Gurukumar, "Image Texture Feature Extract Approach," *Int. J. Sci. Res. Publ.*, vol. 3, no. 5, pp. 1–5, 2013.
- [4] S. Lefkovits, L. Lefkovits, and S. Emerich, "Detecting the eye and its openness with Gabor filters," *2017 5th Int. Symp. Digit. Forensic Secur. ISDFS 2017*, 2017.
- [5] M. Kahlon and S. Ganesan, "Driver Drowsiness Detection System Based on Binary Eyes Image Data," *IEEE Int. Conf. Electro Inf. Technol.*, vol. 2018–May, pp. 209–215, 2018.
- [6] J. J. Yan, H. H. Kuo, Y. F. Lin, and T. L. Liao, "Real-time driver drowsiness detection system based on PERCLOS and grayscale image processing," *Proc. - 2016 IEEE Int. Symp. Comput. Consum. Control. IS3C 2016*, pp. 243–246, 2016.
- [7] B. Bhowmick and K. S. C. Kumar, "Detection and classification of eye state in ir camera for driver drowsiness identification," *ICSIPA09 - 2009 IEEE Int. Conf. Signal Image Process. Appl. Conf. Proc.*, pp. 340–345, 2009.
- [8] C. N. Rao, S. S. Sastry, K. Mallika, H. S. Tiong, and K. B. Mahalakshmi, "Co-Occurrence Matrix and Its Statistical Features as an Approach for Identification Of Phase Transitions Of Mesogens," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 2, no. 9, pp. 4531–4538, 2013.
- [9] M. M. Sani, K. A. Ishak, and S. A. Samad, "Evaluation of Face Recognition System Using Support Vector Machine," *SCORED - 2009 IEEE Student Conf. Proc.*, pp. 2009–2011, 2009.
- [10] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," 2001.
- [11] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," *Image Process. Line*, vol. 4, pp. 128–148, 2014.
- [12] C.P. Riesmala, A. Rizal, L. Novamizanti, *Pengenalan Motif Batik Dengan Analisis Struktur dan Warna Pada Citra Digital*, Skripsi Sarjana pada IT Telkom Bandung, 2012.
- [13] Michael V. Boland, "Haralick texture features," *murphylab*, 1999. [Online]. Available: [http://murphylab.web.cmu.edu/publications/boland/boland\\_node26.html](http://murphylab.web.cmu.edu/publications/boland/boland_node26.html). [Accessed: 15-Mar-2019].
- [14] I. The Mathworks, "Properties of gray-level co-occurrence matrix," *MathWorks*. [Online]. Available: <https://www.mathworks.com/help/images/ref/graycoprops.html>. [Accessed: 15-Mar-2019].
- [15] IPGS Pradnyana, L. Novamizanti, H Fauzi , Perancangan Sistem Pendeteksi Genangan Air Potensi Perkembangbiakan Nyamuk Melalui Foto Citra Udara Dengan Metode Gray Level Co-occurrence Matrix (GLCM), *eProceedings of Engineering 2 (2)*, 2015.
- [16] P.D. Wananda, L. Novamizanti, R.D Atmaja, Sistem Deteksi Cacat Kayu dengan Metode Deteksi Tepi SUSAN dan Ekstraksi Ciri Statistik, *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi & Elektronika*, Vol 6, No 1, 2018.
- [17] I. The Mathworks, "Support Vector Machines for Binary Classification," *MathWorks*. [Online]. Available: <https://www.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html>. [Accessed: 15-Mar-2019].