

DETEKSI KANTUK PENGENDARA RODA EMPAT MENGGUNAKAN HAAR CASCADE CLASSIFIER dan CONVOLUTIONAL NEURAL NETWORK

Cahaya Aji Saputra¹, Danang Erwanto², Putri Nur Rahayu³

Program Studi Teknik Elektro, Fakultas Teknik, Universitas Islam Kediri^{1,2,3}

E-mail: ajisaputra2697@gmail.com, danangerwanto@uniska-kediri.ac.id, putri_nur_rahayu@yahoo.co.id

Abstrak—Salah satu masalah kesehatan yang termasuk dalam penyakit tidak menular adalah kecelakaan lalu lintas. Kecelakaan lalu lintas memiliki dampak negatif seperti kerugian materi, cacat fisik, dan kematian sehingga dapat mempengaruhi derajat kesehatan masyarakat. Rasa kantuk saat berkendara merupakan salah satu kondisi yang tidak jarang diabaikan oleh para pengendara kendaraan bermotor dan merupakan salah satu hal yang menyebabkan terjadinya kecelakaan, terutama ketika berkendara dalam jarak yang cukup jauh. Kejadian mengantuk atau tertidur dalam waktu beberapa detik tersebut adalah *microsleep*. Pengemudi kendaraan bermotor sangat sensitif terhadap *microsleep* karena faktor kelelahan fisik selama mengemudi. Durasi *microsleep* sangat singkat yaitu diantara 3 detik hingga 5 detik, justru ada yang memiliki durasi sampai 10 detik. Penelitian ini mengembangkan pengolahan citra digital untuk mendeteksi kantuk pada pengendara mobil menggunakan metode pengenalan obyek Haar Cascade Classifier dan klasifikasi menggunakan Convolutional Neural Network. Masukan citra secara *real-time* pada sistem didapat dari kamera yang dipasang didepan pengemudi. Keluaran dari sistem terdapat suara alarm untuk peringatan bahwa pengendara sedang berada pada posisi mengantuk atau tertidur. Sistem dapat mendeteksi berbagai jenis-jenis mata dengan tingkat keberhasilan sebesar 100%. Akurasi rata-rata yang diperoleh untuk mendeteksi mata terbuka dan tertutup dengan jarak 30 – 50 Cm adalah 95,4%. Sedangkan akurasi rata-rata untuk mendeteksi kantuk adalah 93,9%. Rata-rata waktu komputasi sistem ini adalah 0.1069 detik yang akan mempercepat dalam pendeteksian kantuk.

Kata kunci : *Microsleep*, Pengolahan Citra Digital, Haar Cascade Classifier, Convolutional Neural Network

Abstract—One of the health problems included in non-communicable diseases is traffic accidents. Traffic accidents have negative impacts such as material loss, physical disability, and death so that they can affect the degree of public health. Drowsiness while driving is a condition that is often ignored by motorists and is one of the things that causes accidents, especially when driving long distances. The occurrence of drowsiness or falling asleep within a few seconds is *microsleep*. Motor vehicle drivers are very sensitive to *microsleep* due to physical fatigue during driving. *Microsleep* duration is very short, which is between 3 seconds to 5 seconds, some actually have a duration of up to 10 seconds. This study develops digital image processing to detect drowsiness in motorists using Haar Cascade Classifier object recognition and classification using Convolutional Neural Network. Real-time image input on the system is obtained from the camera installed in front of the driver. The output of the system is an alarm sound to warn that the driver is in a sleepy or asleep position. The system can detect various eye types with a 100% success rate. The average accuracy obtained for detecting open and closed eyes with a distance of 30-50 cm is 95.4%. While the average accuracy for detecting drowsiness was 93.9%. The average computation time of this system is 0.1069 seconds which will speed up the detection of sleepiness.

Keyword : *Microsleep*, Digital Image Processing, Haar Cascade Classifier, Convolutional Neural Network

I. PENDAHULUAN

Kecelakaan lalu lintas adalah salah satu masalah kesehatan yang termasuk kedalam penyakit tidak menular. Dampak negatif dari kecelakaan lalu lintas antara lain kerugian materi, cacat fisik, dan kematian sehingga dapat berpengaruh terhadap derajat kesehatan masyarakat [1]. Rasa kantuk saat berkendara merupakan salah satu kondisi yang tidak jarang diabaikan oleh para pengendara kendaraan bermotor dan merupakan salah satu hal yang menyebabkan terjadinya kecelakaan, apalagi ketika berkendara dalam jarak yang cukup jauh. Kejadian mengantuk atau tertidur dalam waktu beberapa detik tersebut adalah *microsleep*. Pengemudi kendaraan bermotor sangat sensitif terhadap *microsleep* karena faktor kelelahan fisik selama mengemudi. Durasi *microsleep* sangat singkat yaitu diantara 3 detik hingga 5 detik, justru ada yang memiliki durasi sampai 10 detik.

Upaya untuk menghindari dan mengurangi angka kecelakaan yang akan terjadi pada masa akan datang, diperlukan sebuah sistem yang bisa mendeteksi kantuk pada pengemudi yang harus mengemudikan kendaraannya ketika malam hari. Mata merupakan salah satu indra manusia yang digunakan untuk pendeteksian reaksi manusia oleh perangkat cerdas. Analisis ukuran permukaan iris dapat digunakan untuk menentukan keadaan mata dalam kondisi terbuka ataupun tertutup [2]; Beberapa penelitian pendeteksi kantuk menggunakan pengolahan citra digital pernah dilakukan [3], [4] dan [5].

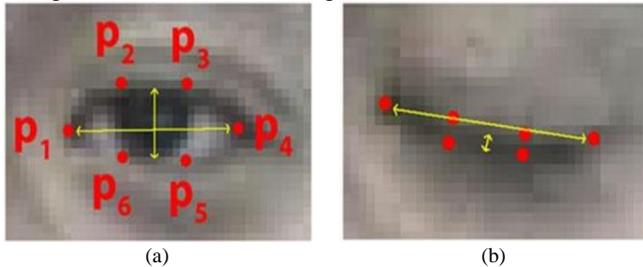
Bertolak dari permasalahan diatas, maka penulis mengembangkan suatu sistem berbasis pengolahan citra digital untuk mendeteksi kantuk pada pengendara roda empat berdasarkan durasi kedipan mata secara *real-time* menggunakan Haar Cascade Classifier dan Convolutional Neural Network (CNN). Pada penelitian ini, tahapan yang dilakukan yaitu Region of Interesting (ROI) mata dengan menerapkan Haar Cascade Classifier dan mendeteksi kondisi mata terbuka dan tertutup menggunakan Convolutional Neural Network, kemudian langkah terakhir melakukan deteksi kantuk dengan cara menghitung lamanya mata terpejam dengan mengacu pada teori *microsleep*. Dengan adanya sistem pendeteksi kantuk sehingga diharapkan dapat menurunkan angka kecelakaan yang sering terjadi karena faktor mengantuk pengemudi.

II. TINJAUAN PUSTAKA

A. Kantuk

Kantuk (*drowsiness*) ialah keadaan dimana seseorang ingin tidur. Namun kondisi kantuk pada waktu-waktu yang tidak tepat dapat mengakibatkan hal yang fatal. Mengantuk

dapat disebabkan oleh beberapa faktor antara lain: kelelahan bekerja, kurangnya tidur yang cukup, dan lain sebagainya. Kondisi mengantuk terdapat beberapa macam sehingga dapat dikategorikan seseorang sedang mengantuk atau tidak. Kondisi mengantuk seseorang antara lain dapat dilihat dari kondisi kelopak mata mulai berat, pandangan kabur serta kepala mulai tidak seimbang menahan beban sehingga mengharuskan untuk berbaring dan istirahat.



Gbr. 1 a. Visualisasi landmark mata pada kondisi mata terbuka; b. Visualisasi landmark mata pada kondisi mata tertutup [6]

Mengantuk pada umumnya disebabkan karena bekerja sepanjang waktu atau pergantian *shift* kerja, konsumsi obat, kondisi tubuh yang kurang sehat, menurunnya kualitas dan kuantitas tidur. Selain itu kantuk juga disebabkan oleh gejala prediabetes, kurangnya olahraga, perbedaan psikologi, gaya hidup serta gejala anemia. aktivitas yang bisa dilakukan untuk menghindari terjadinya kantuk antara lain dengan cara minum air atau kopi, berbicara pada diri sendiri, mendengarkan musik keras, meningkatkan kecepatan, mengobrol dengan penumpang atau membuka kaca jendela.

B. Parameter Kantuk

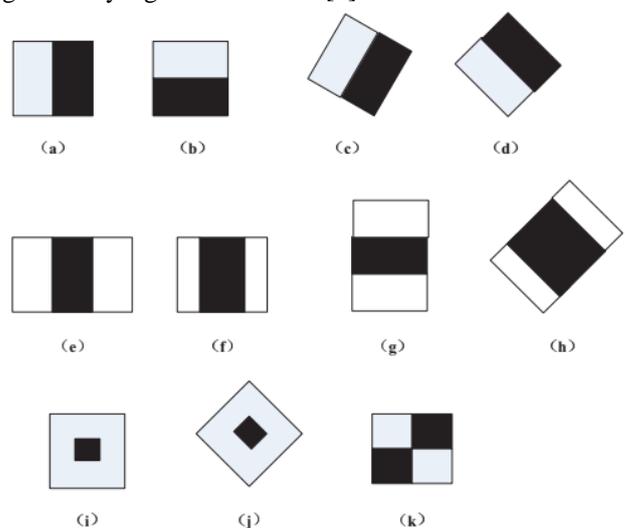
Teori *Microsleep* merupakan suatu kejadian hilangnya kesadaran atau pengawasan seseorang karena merasakan kelelahan ataupun mengantuk. Peristiwa *microsleep* biasanya akan berjalan sekitar 1 detik hingga 2 menit, akan tetapi durasi tersebut dapat bertambah lebih lama jika pengemudi benar-benar memasuki fase tidur. *Microsleep* sering terjadi ketika seseorang memiliki aktifitas pekerjaan yang monoton seperti halnya berkendara atau menatap layar pada waktu yang cukup lama.[3]

Pengemudi yang mengalami *microsleep* tidak sadar jika dirinya akan memasuki kondisi tidur bahkan hingga tertidur. Keadaan *microsleep* juga dapat terjadi dengan mata terbuka dengan pandangan kosong. *Microsleep* juga ditandai dengan kondisi mengedipkan mata dengan frekuensi yang sering dan gerakan kepala seperti mengangguk serta tidak mampu mengingat suatu hal yang terjadi pada beberapa menit sebelumnya. Setelah tertidur, pengemudi yang mengalami *microsleep* sering terbangun dengan perasaan lebih segar pada waktu yang singkat.

C. Haar Cascade Classifier

Haar Like Feature atau yang disebut sebagai *Haar Cascade Classifier* merupakan *rectangular* (persegi) *feature*, yang memberikan isyarat secara eksplisit terhadap sebuah citra atau *image*. *Haar cascade classifier* merupakan suatu metode pengenalan obyek yang dikenalkan oleh Paul Viola

dan Michael Jhon. Oleh sebab itu metode *haar cascade classifier* dinamakan juga dengan metode Viola & Jhon. Gagasan dari *Haar like feature* yaitu melakukan pengenalan obyek berdasarkan parameter sederhana dari fitur tetapi bukan merupakan nilai piksel dari citra obyek tersebut. Kelebihan dari teknik ini adalah waktu komputasi yang sangat singkat sehingga mempercepat proses pengenalan, karena pada metode *Haar like feature* hanya tergantung dari jumlah piksel didalam persegi bukan tergantung pada setiap nilai piksel pada sebuah citra [7]. Teknik pengenalan menggunakan *Haar like feature* merupakan teknik yang menerapkan statistik model (*classifier*). Tahapan dalam melakukan pengenalan objek pada citra menggunakan *Haar Like Feature* adalah mengintegrasikan empat kunci utama yaitu *Haar Like Feature*, *Integral Image*, *Adaboost learning* dan *Cascade Classifier* [8]. Gambar 2 adalah penerapan *Haar Like Feature* secara umum yang dapat digunakan pada semua skala peningkatan pengklasifikasi dan dapat dengan cepat dihitung dari versi integral citra yang akan dideteksi [9].



Gbr. 2 Penerapan *Haar Like Feature*: a – d. Pada fitur tepi; e – h. Pada fitur garis; i dan j. Pada fitur *center-surround*; k. Pada fitur garis diagonal khusus [10]

Haar Like Feature merupakan fitur yang berbasis pada *Wavelet Haar*. *Wavelet Haar* merupakan gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah). Pada citra dua dimensi, interval rendah adalah gelap sedangkan interval tinggi adalah terang. Selanjutnya dilakukan kombinasi-kombinasi kotak yang dipakai dalam proses pengenalan objek secara visual agar memperoleh hasil optimal. *Setiap Haar-like feature* dibangun oleh gabungan kotak - kotak hitam dan putih.

Haar Like Feature ditetapkan dengan kaidah mengurangi daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai selisihnya yang didapatkan melebihi nilai ambang atau *threshold*, maka dapat disimpulkan bahwa *Haar Like Feature* tersebut ada. Nilai dari *Haar Like Feature* merupakan selisih antara jumlah nilai-nilai piksel derajat keabuan didalam daerah kotak hitam dan daerah kotak putih sebagaimana ditunjukkan oleh gambar 1.

$$f(x) = \text{SumBlack rectangle} - \text{SumWhite rectangle} \quad (1)$$

Integral *image*/citra integral atau yang disebut juga dengan istilah *summed area table* dipakai untuk menetapkan ada atau tidaknya ratusan *Haar Like Feature* dalam sebuah citra dengan cara menghitung jumlah nilai dengan cepat dan efisien. Citra integral didapatkan dari persamaan 2 berikut [11].

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

Dari persamaan 2 $ii(x, y)$ merupakan citra integral pada lokasi piksel (x, y) sedangkan $i(x', y')$ merupakan citra aslinya. Detektor *Haar Cascade Classifier* menggunakan *Adaboost (adaptive boosting) learning* atau yang disebut *rejection cascade* [9]. *Adaboost learning* adalah suatu algoritma *machine learning* yang digunakan untuk membuat suatu klasifikasi secara spesifik pada fitur yang dianggap penting dengan melakukan pembelajaran terhadap beberapa *classifier* yang telah dihasilkan. Persamaan 3 merupakan persamaan untuk melakukan klasifikasi menggunakan *adaboost learning* [12].

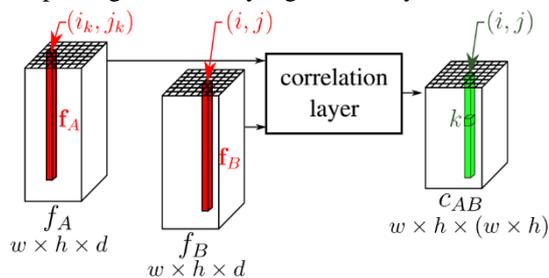
$$h_j x = \begin{cases} 1, & \text{if } p_j f_j x < p_j \theta_j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Cascade classifier adalah suatu *classifier* yang digunakan untuk mengenali obyek yang telah dilakukan *learning* sebelumnya .

D. Convolutinal Neural Network

Convolutional Neural Network (CNN) adalah salah satu pengembangan dari *Artificial Neural Network (ANN)*/Jaringan Syaraf Tiruan (JST) yang merupakan inspirasi dari jaringan syaraf manusia dan biasa digunakan pada data citra untuk mendeteksi dan mengenali suatu objek pada sebuah citra [8]. CNN termasuk pada jenis *Deep Neural Network* karena dalam algoritma tersebut memiliki tingkat jaringan dan banyak diterapkan pada aplikasi pengolahan data citra. CNN memiliki dua metode; yaitu klasifikasi menggunakan *feedforward* dan tahap pembelajarannya menggunakan *backpropagation*. Cara kerja dari CNN yaitu adanya kesamaan dengan *Multi Layer Perceptron (MLP)*. akantetapi dalam melakukan klasifikasi menggunakan CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti pada klasifikasi menggunakan MLP dimana setiap neuron-nya hanya memiliki ukuran satu dimensi saja.

CNN adalah pengembangan lebih lanjut dari metode MLP karena CNN menggunakan metode yang mirip dengan metode MLP tetapi dengan dimensi yang lebih banyak.



Gbr. 3 Korelasi peta perhitungan dengan fitur CNN [13]

Pada algoritma CNN ini, masukkan dari *layer* sebelumnya tidak menggunakan *array* 1 dimensi tetapi menggunakan *array* 2 dimensi. Jika dianalogikan dengan fitur dari wajah manusia, *layer* pertama merupakan refleksi dari goresan-goresan dengan arah berbeda, sedangkan pada *layer* kedua merupakan fitur seperti bentuk mata, mulut, dan hidung mulai terlihat, hal ini terjadi karena dilakukan penggabungan/*pooling* dari *layer* pertama yang masih berupa goresan-goresan, pada *layer* ketiga akan terbentuk kombinasi fitur-fitur mata, mulut, dan hidung yang selanjutnya akan dapat menyimpulkan wajah orang tertentu.

Tahapan pertama dalam algoritma CNN yaitu tahap konvolusi. Pada tahap ini dikerjakan dengan mengaplikasikan sebuah kernel yang digunakan tergantung dari jumlah fitur yang akan dihasilkan. Selanjutnya pada tahap berikutnya yaitu melakukan fungsi aktivasi. Fungsi aktivasi ini pada umumnya memakai *Rectifier Linear Unit (ReLU)*, Kemudian setelah keluar dari fungsi aktivasi langkah selanjutnya adalah melakukan proses *polling*. Proses ini di ulang beberapa kali sehingga diperoleh peta fitur yang cukup untuk dilanjutkan ke tahap *Fully Connected neural network*. Sehingga dari situlah dapat dihasilkan *output class*.

III. PERANCANGAN SISTEM

Pada penelitian ini, kamera diletakkan di depan pengemudi dengan posisi sejajar terhadap bahu atau dada pengemudi. Jika posisi kamera terlalu keatas maka akan mengganggu pengemudi yang sedang mengemudi. Pada pengujian ini, posisi pengemudi dikondisikan seperti sedang mengemudi.



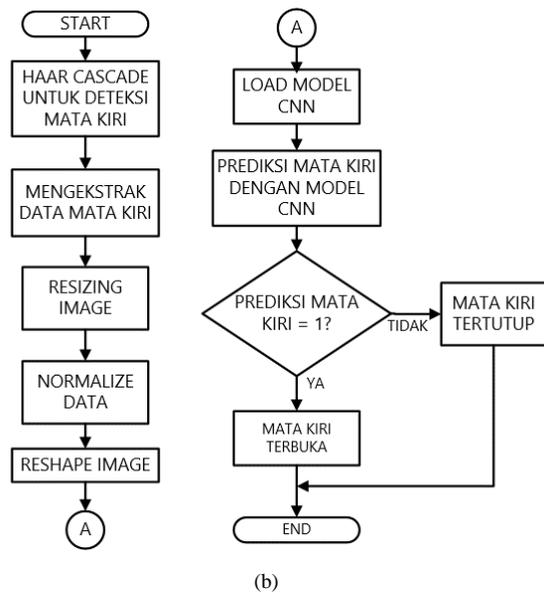
Gbr. 4 Peletakan kamera dalam mobil

Deteksi keberadaan mata menggunakan *Haar Cascade Classifier*. Implementasi *Haar Cascade Classifier* menggunakan *Open Source Computer Vision Library* atau yang diberi nama *OpenCV* [14] dengan lisensi permisif BSD yang lebih bebas daripada lisensi GPL [6]. deteksi kondisi mata kanan dan mata kiri terbuka atau tertutup menggunakan CNN. Gambar mata terbuka dan tertutup digunakan sebagai *training model* pada proses CNN, serta untuk mendeteksi kantuk berdasarkan lamanya durasi kedipan mata. *Output* yang bertuliskan mata terbuka dan tertutup akan muncul pada sistem sesuai keadaan mata pengendara serta peringatan berupa *alarm* akan berbunyi jika pengendara telah mengantuk atau memejamkan matanya lebih dari 3 detik.

Pada penelitian ini, durasi yang digunakan sebagai acuan sistem deteksi kantuk diambil dari teori *microsleep* yang menyatakan bahwa durasi mata tertutup orang mengantuk adalah lebih dari 3 detik, hingga ada yang sampai dengan 10 detik. Pada tampilan monitor terdapat juga durasi atau waktu yang digunakan untuk menghitung saat pengendara menutup mata, dan setelah menutup mata apabila mata terbuka lagi maka durasi atau waktu akan kembali ke angka 0 detik.

A. Deteksi mata tertutup/terbuka

Dalam deteksi mata terbuka atau tertutup citra wajah yang terdeteksi digunakan sebagai masukan, selanjutnya dilakukan proses ROI mata kanan dan kiri dengan menggunakan *file Haar Cascade Classifier*, selanjutnya dilakukan mengubah ukuran piksel citra (*resizing image*), normalisasi data, mengubah matriks data citra menjadi vektor (*reshape image*), load model CNN yang sebelumnya telah dilakukan *training*, Kemudian membuat prediksi serta pelabelan yang menandakan mata kanan dan kiri apakah dalam kondisi terbuka atau tertutup.

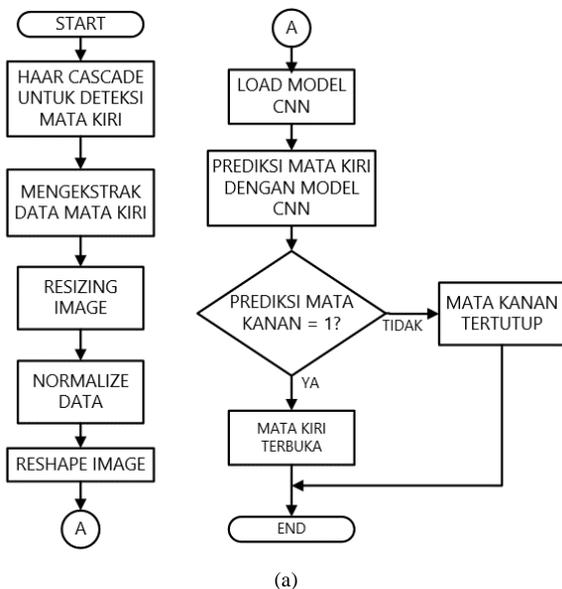


(b)

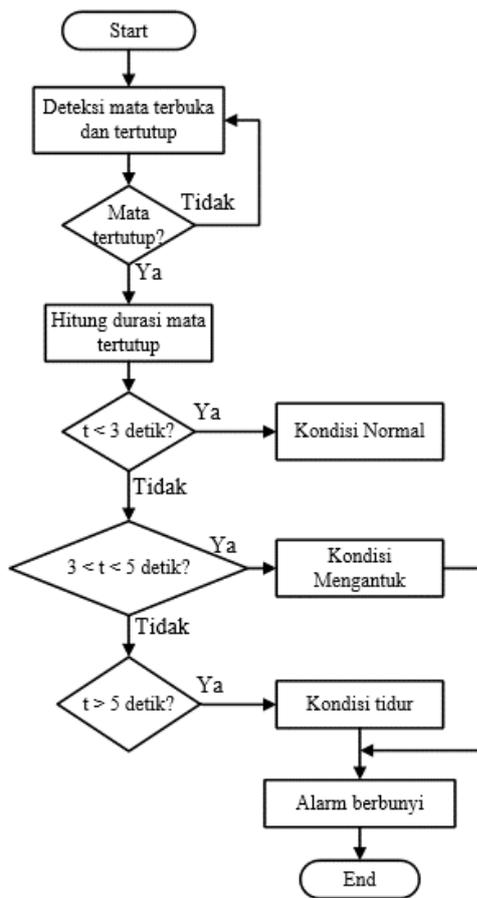
Gbr. 5 Flowchart deteksi kondisi mata terbuka/tertutup pada a. Mata kanan; b. Mata kiri

Gambar 5 adalah *flowchart* deteksi dengan kondisi mata kanan dan kiri dalam keadaan terbuka atau tertutup menggunakan algoritma CNN. Untuk prediksi kondisi mata dibandingkan dengan model CNN yang telah di-*training* sebelumnya.

B. Deteksi Kantuk



(a)



Gbr. 6 Flowchart deteksi kantuk

Cara kerja sistem adalah sebagai berikut:

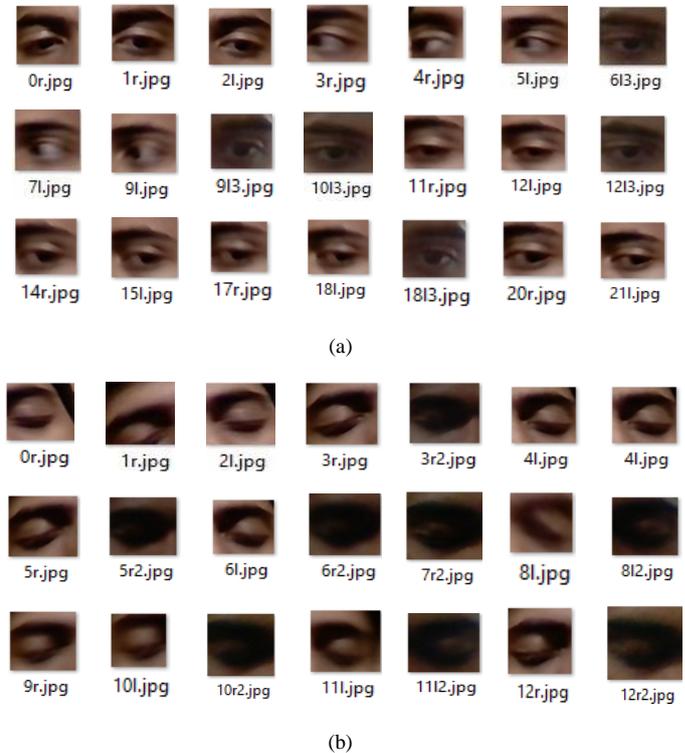
1. *Start*, program dan webcam dijalankan.
2. Deteksi mata menggunakan *Haar Cascade Classifier*,
3. Kemudian akan dideteksi kondisi mata dari wajah pengemudi apakah dalam keadaan terbuka atau tertutup. deteksi kondisi mata ini menggunakan CNN
4. Deteksi kantuk, akan dicek apakah mata dalam kondisi tertutup 100% atau dalam kondisi terbuka.
5. Jika kondisi mata tertutup, maka program akan memeriksa durasi mata tertutup:
 - a. Jika durasi mata tertutup < 3 detik, maka kondisi *user* normal.
 - b. Jika durasi mata tertutup antara 3 – 5 detik, maka kondisi *user* mengantuk.
 - c. Jika durasi mata tertutup > 5 detik, maka kondisi *user* tertidur.

IV. HASIL

Hasil pengujian sistem deteksi kantuk pada pengendara roda empat diambil dari durasi ketika mata menutup secara *real-time* dengan menerapkan *Haar Cascade Classifier* dan CNN meliputi : ROI citra mata menggunakan algoritma *Haar Cascade Classifier*, pengujian kondisi mata terbuka maupun tertutup menggunakan CNN, pengujian deteksi kantuk berdasarkan durasi lamanya mata tertutup, perhitungan waktu komputansi sistem.

A. Pengambilan citra mata

Setelah kamera mengambil citra kemudian system melakukan ROI pada obyek mata kanan dan kiri menggunakan *Haar Cascade Classifier* baik dalam kondisi terbuka maupun tertutup. Gambar 7a. merupakan citra mata kanan dan kiri dalam kondisi terbuka dalam berbagai sudut pengambilan citra. Sedangkan gambar 7b. merupakan citra mata kanan dan kiri dalam kondisi tertutup dalam berbagai sudut pengambilan citra.



Gbr. 7 Merupakan citra mata kanan maupun mata kiri dalam kondisi terbuka dan tertutup

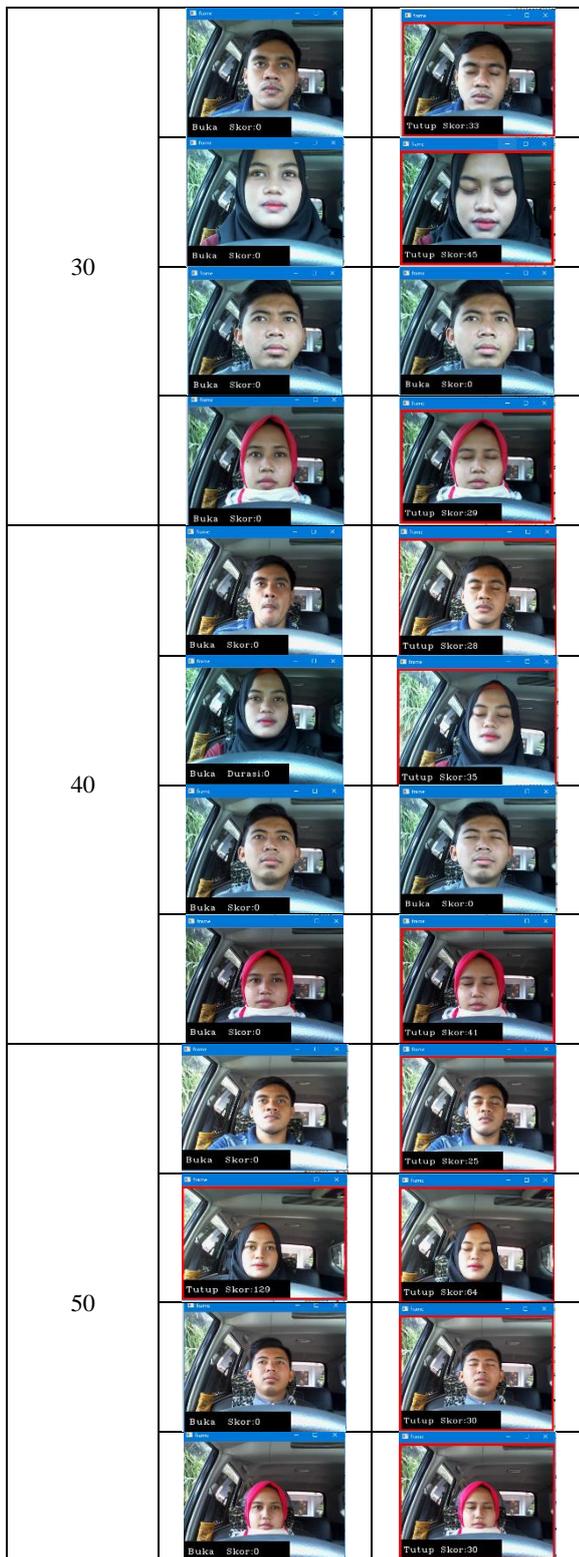
Gambar 7 merupakan citra tampilan mata ketika dalam kondisi terbuka dan tertutup. Dari gambar tersebut, sistem mampu mengenali mata baik ketika terbuka maupun tertutup. Setelah area mata dikenali, selanjutnya fokus pada area mata saja dalam mendeteksi mata yang dikenali apakah dalam kondisi terbuka atau tertutup.

B. Pengujian deteksi mata terbuka dan tertutup

Pengujian ini untuk mendeteksi mata terbuka dan tertutup. Pada pengujian deteksi mata terbuka dan tertutup ini penulis membuat jarak 30 Cm, 40 Cm, dan 50 Cm terhadap kamera ke muka pengemudi. Pengujian disesuaikan seperti kondisi sedang mengemudi di dalam mobil. Tabel 1 merupakan hasil pengujian deteksi mata terbuka dan tertutup menggunakan klasifikasi CNN.

TABEL 1
HASIL PENGUJIAN DETEKSI MATA TERBUKA DAN TERTUTUP

Jarak (Cm)	Mata terbuka	Mata tertutup
------------	--------------	---------------



Pada tabel 1 menunjukkan program dapat mengenali kondisi mata terbuka dan tertutup pada jarak 30 Cm, 40 Cm dan 50 Cm. ketika mengenali kondisi mata terbuka, maka pada tampilan ditandai dengan notifikasi “Buka”. Ketika mengenali kondisi mata tertutup, maka pada tampilan ditandai dengan notifikasi “Tutup” dan mulai menghitung waktu.

C. Pengujian deteksi kantuk

Pada pengujian deteksi kantuk dilaksanakan menggunakan sampel sebanyak 11. Kamera dipasang dengan jarak 30 Cm, 40 Cm dan 50 Cm terhadap muka pengemudi dan dikondisikan seperti kondisi sedang mengemudi di dalam mobil.

Durasi yang digunakan sebagai acuan untuk mengenali kondisi kantuk diambil dari teori *microsleep* yang menyatakan bahwa durasi pengemudi mengantuk sekitar 3 hingga 5 detik, selain itu ada yang durasinya sampai dengan 10 detik. Jika durasi mata tertutup lebih dari 3 detik maka alarm akan berbunyi (*On*). Jika setelah kondisi tertutup, kemudian mata terbuka lagi maka perhitungan durasi kembali ke nol, Tabel 2 merupakan hasil uji coba sistem terhadap deteksi kantuk pada pengendara roda empat.

TABEL 2
HASIL PENGUJIAN DETEKSI KANTUK

No	Jarak (Cm)	Alarm	Keterangan
1	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
2	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
3	30	<i>On</i>	Berhasil
	40	<i>Off</i>	Gagal
	50	<i>On</i>	Berhasil
4	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
5	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
6	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
7	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
8	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
9	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
10	30	<i>Off</i>	Gagal
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil
11	30	<i>On</i>	Berhasil
	40	<i>On</i>	Berhasil
	50	<i>On</i>	Berhasil

Pada pengujian terhadap deteksi kantuk, didapatkan hasil bahwa tingkat keberhasilan sistem untuk mengetahui pengemudi mengantuk atau tidak adalah 93,9%.

D. Pengujian waktu komputansi

Pengujian terhadap waktu komputansi yang digunakan untuk mengetahui seberapa cepat waktu komputasi dari sistem. Tabel 3 adalah hasil pengujian waktu komputasi.

TABEL 3
WAKTU KOMPUTASI

Sampel pengujian ke	Waktu Komputansi		
	Pada Jarak 30 Cm	Pada Jarak 40 Cm	Pada Jarak 50 Cm
1	0,1064 Detik	0,1068 Detik	0,1005 Detik
2	0,1081 Detik	0,1060 Detik	0,0973 Detik
3	0,1061 Detik	0,1239 Detik	0,0966 Detik
4	0,1122 Detik	0,1139 Detik	0,1101 Detik
5	0,1177 Detik	0,0996 Detik	0,1028 Detik
6	0,0990 Detik	0,1211 Detik	0,0963 Detik
7	0,1031 Detik	0,0979 Detik	0,1050 Detik
8	0,1055 Detik	0,0981 Detik	0,1021 Detik
9	0,1133 Detik	0,1119 Detik	0,1110 Detik
10	0,1018 Detik	0,0923 Detik	0,1733 Detik
11	0,1024 Detik	0,0996 Detik	0,08703 Detik
Rata-rata	0,106873 Detik	0,106464 Detik	0,107457 Detik

Pada Tabel 3 adalah data waktu komputansi dari hasil dari pengujian terhadap 11 sampel pengujian. Pengujian dilakukan pada jarak 30 Cm, 40 Cm dan 50 Cm. Dari pengujian dengan jarak 30 Cm diperoleh besar rata-rata waktu komputasi adalah 0,106873 detik. Pengujian dengan jarak 40 Cm diperoleh besar rata-rata waktu komputasi adalah 0,106464 detik. Sedangkan pengujian dengan jarak 50 Cm diperoleh besar rata-rata waktu komputasi adalah 0,107457 detik. Sehingga rata-rata total waktu komputasi yang didapatkan pada jarak 30 – 50 Cm yaitu sebesar 0.106931 detik.

V. KESIMPULAN

Dari hasil penelitian deteksi kantuk pengendara roda empat menggunakan *Haar Cascade Classifier* dan CNN, sistem dapat mengenali mata menggunakan *Haar Cascade Classifier* dengan tingkat keberhasilan sebesar 100%. Akurasi rata-rata yang diperoleh untuk mendeteksi mata terbuka dan tertutup dengan jarak 30 – 50 Cm adalah 95,4%. Sedangkan akurasi rata-rata untuk mendeteksi kantuk adalah 93,9%. Rata-rata waktu komputasi sistem ini adalah 0.1069 detik yang akan mempercepat dalam pendeteksian kantuk.

REFERENSI

- [1] A. Hidayati and L. Y. Hendrati, "Analisis Risiko Kecelakaan Lalu Lintas Berdasar Pengetahuan, Penggunaan Jalur, dan Kecepatan Berkendara," *J. Berk. Epidemiol.*, vol. 4, no. 2, pp. 275–287, 2016.
- [2] M. Daniluk, M. Rezaei, R. Nicolescu, and R. Klette, "Eye Status Based on Eyelid Detection: A Driver Assistance System," in *Lecture Notes in Computer Science*, 2014, vol. 8671, pp. 171–178.
- [3] M. Amirullah, H. Kusuma, and T. Tasripan, "Sistem Peringatan Dini Menggunakan Deteksi Kemiringan Kepala pada Pengemudi Kendaraan Bermotor yang Mengantuk," *J. Tek. ITS*, vol. 7, no. 2, pp. F281–F286, 2019.
- [4] I. Imanuddin, F. Alhadi, R. Oktafian, and A. Ihsan, "Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode Viola Jones," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 321–329, 2019.
- [5] B. Hartiansyah, "DETEKSI DAN IDENTIFIKASI KONDISI

- KANTUK PENGENDARA KENDARAAN BERMOTOR MENGGUNAKAN EYE DETECTION ANALYSIS," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 3, no. 1, pp. 59–64, 2019.
- [6] A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka Opencv Dan Dlib Python," *SAINSTECH*, vol. 28, no. 2, 2018.
- [7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.
- [8] M. Syarif and W. Wijanarto, "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login Sistem," *Techno. com*, vol. 14, no. 4, pp. 242–249, 2015.
- [9] L. Cuimei, Q. Zhiliang, J. Nan, and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," in *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 2017, pp. 483–487.
- [10] Y. Li, X. Xu, N. Mu, and L. Chen, "Eye-gaze tracking system by haar cascade classifier," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, 2016, pp. 564–567.
- [11] C. Zhang and Z. Zhang, *A Survey of Recent Advances in Face Detection*. 2010.
- [12] L. Arreola, "Stabilization and Object Tracking based on Computer Vision: Application in a Quad-rotor," 2018.
- [13] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6148–6157.
- [14] A. Priadana and M. Habibi, "Face detection using haar cascades to filter selfie face image on instagram," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIIT)*, 2019, pp. 6–9.