

## DETEKSI KANTUK PENGEMUDI BUS TRANS METRO BANDUNG DENGAN PENDEKATAN RUMUS *EYE ASPECT RATIO*

Aufaryafi Baskara Kadi<sup>1</sup>, Rendy Munadi<sup>2</sup>, Nurwulan Fitriyanti<sup>3</sup>

<sup>1,2,3</sup>Fakultas Teknik Elektro, Universitas Telkom, Bandung

aufaryafibaskara@student.telkomuniversity.ac.id<sup>1</sup>, rendymunadi@telkomuniverstiy.ac.id<sup>2</sup>, nurwulanf@telkomuniversity.ac.id<sup>3</sup>

### Abstrak

Kecelakaan lalu lintas merupakan kejadian yang paling umum terjadi di dunia, terutama Indonesia. Kecelakaan lalu lintas banyak disebabkan oleh beberapa faktor, salah satunya yaitu rasa kantuk pengemudi. Rasa kantuk pengemudi sering timbul ketika pengemudi mulai kelelahan, maupun ketika perjalanan terasa membosankan seperti perjalanan jauh maupun ketika terlalu lama berada dalam kemacetan. Pada saat ini, belum banyak diaplikasikan sistem pendeteksi rasa kantuk pengemudi kendaraan. Pada perjalanan jauh, pengemudi bus harus selalu ditemani seorang ‘kernet’ atau pembantu pengemudi saat perjalanan, yang kurang efektif karena jika ‘kernet’ tersebut tertidur maka tidak ada yang memantau rasa kantuk pengemudi. *Eye Aspect Ratio (EAR)* bekerja dengan menghitung jarak *Euclidean* antar 6 titik *facial landmarks* pada masing-masing mata. Akurasi sistem akan diuji dengan mengakuisisi wajah pengemudi bus *TMB*. Keakuratan sistem akan didapatkan jika pengemudi terdeteksi mengantuk. Dari hasil pengujian, didapatkan nilai *threshold EAR* terbaik yaitu 0.23 dengan akurasi sistem 79%. Setelah dilakukan pengujian akurasi, dilanjutkan dengan mencoba deteksi secara *real-time*. Hasilnya, posisi wajah dan intensitas cahaya berpengaruh terhadap pendeteksian.

**Kata kunci:** deteksi kantuk, *eye aspect ratio*, *facial landmarks*, *real-time*, *Trans Metro Bandung (TMB)*.

### I. PENDAHULUAN

Rasa kantuk merupakan hal yang sering terjadi bagi setiap orang. Pada jalan raya, hal tersebut dapat disebabkan baik oleh rasa lelah pengemudi, maupun jalan yang terlalu monoton. Faktanya, rasa kantuk pengemudi merupakan salah satu penyumbang angka kecelakaan lalu lintas terbesar. Berdasarkan data dari Badan Administrasi Nasional Keselamatan Lalu Lintas (NHTSA) Amerika Serikat yang dikeluarkan pada tahun 2017, sedikitnya terjadi 232.000 kecelakaan yang terjadi, dengan 160.000 luka berat atau ringan dan 3.662 meninggal dunia [1]. Hal tersebut melatarbelakangi topik penelitian mengenai sistem pendeteksi kantuk yang dapat mendeteksi kantuk pengemudi bus, terutama pada bus *Trans Metro Bandung*. Penelitian ini bertujuan untuk mendeteksi kantuk pengemudi bus. Penelitian ini menggunakan bahasa pemrograman Python untuk melakukan proses pengambilan nilai akurasi serta proses deteksi yang akan menghasilkan nilai *threshold* hasil deteksi kantuk. Proses pengambilan nilai akurasi akan dilakukan melalui pendekatan rumus *Eye Aspect Ratio (EAR)*. Rumus ini digunakan sebagai rumus acuan dikarenakan kecepatan rumus ini dalam menentukan ukuran mata.

#### I. KAJIAN TEORI

##### A. Computer Vision

*Computer Vision (CV)* merupakan salah satu ilmu komputasi yang bekerja berdasarkan masukan gambar ataupun video. Pada dasarnya, CV bertindak seperti *Artificial Intelligence (AI)*. CV dapat mengklasifikasi berbagai jenis objek pada gambar, seperti foto kepadatan lalu lintas dengan menandai jenis kendaraan tersebut. Selain itu, CV juga dapat mendeteksi objek dengan menemukan lokasi objek pada gambar kemudian mengklasifikasi objek tersebut, seperti mendeteksi kepadatan lalu lintas dari CCTV dengan mendeteksi kendaraan, lalu mengklasifikasikan sesuai jenis

kendaraan, dan mengambil kesimpulan apakah keadaan lalu lintas tersebut padat atau lancar [2]. *OpenCV* merupakan perpustakaan dari CV berbasis *open-source*. *OpenCV* tersedia baik dalam bahasa pemrograman C, Python, maupun C++. *OpenCV* dapat mengolah citra secara *real-time* yang memanipulasi citra untuk melakukan berbagai aktivitas seperti menampilkan gambar dari kamera maupun membuat AI untuk pengenalan objek dunia nyata [3].

##### B. Facial Landmarks

*Facial Landmarks* adalah metode yang digunakan untuk mendeteksi wajah, pengenalan wajah, serta kontur kepala. Metode ini menggambarkan *facial landmarks* dengan menggunakan *Ensemble of Regression Trees* dari himpunan bagian yang memberi nilai indeks ke perkiraan bentuk [4]. *Facial landmarks* dapat mendeteksi ketika mulut terbuka maupun tertutup, serta gerakan mata ke kiri, atas, kanan, bawah, juga ketika mata terbuka dan tertutup [5]. *Facial Landmarks* membuat 68 titik pada wajah lengkap mulai dari dagu, pipi, hidung, mata, alis, serta mulut manusia. 68 pola tersebut dapat berubah mengikuti arah gerakan kepala, seperti ketika kepala lurus ke depan, melihat kiri, kanan, atas, dan bawah, maupun memutar [6], [7].

##### C. Eye Aspect Ratio

*Eye Aspect Ratio (EAR)* merupakan nilai skalar yang mengambil bagian dari *facial landmarks* yaitu bagian mata kiri dan kanan dari wajah, dengan mengambil titik ke-37 hingga 48 pada *facial landmarks*. Nilai EAR dapat dihitung dengan menghitung Jarak *Euclidean* antar 6 titik mata. Jika mata semakin rapat, maka nilai EAR mendekati atau mencapai 0, sedangkan jika mata terbuka lebar, maka EAR bernilai konstan sekitar 0.25 hingga 0.35 [8]. EAR dapat dihitung dengan menggunakan rumus:

$$EAR = \frac{\|P_2 - P_6\| + \|P_3 - P_5\|}{2 \|P_1 - P_4\|} \quad (1)$$

Titik P2, P3, P5, dan P6 digunakan untuk mengukur ketinggian mata, sedangkan titik P1 dan P4 untuk mengukur kelebaran mata. Nilai EAR dihitung berdasarkan satu mata. Untuk mendapatkan nilai EAR kedua mata, maka EAR dari mata kiri ditambah dengan EAR mata kanan dibagi dengan 2, serta menggunakan rumus berikut.

$$EAR_{avg} = \frac{EAR_{kiri} + EAR_{kanan}}{2} \quad (2)$$

Nilai *threshold* EAR merupakan nilai EAR saat mata sudah tidak dikategorikan sebagai terbuka. Jika nilai EAR di atas nilai *threshold* yang ditentukan, maka mata dikategorikan sebagai terbuka. Sedangkan jika nilai EAR di bawah nilai *threshold*, maka mata dikategorikan sebagai tertutup [9], [10]. Pada penelitian sebelumnya, didapat bahwa nilai EAR saat kedua mata terbuka berkisar antara 0.2 hingga 0.3. Saat ditentukan dengan nilai *threshold* 0.2, didapat nilai *F1-Score* (nilai rata-rata harmonik dari presisi dan *recall*) sebesar 0.82 [11].

D. Pengukuran Sistem

Pada pengukuran sistem, digunakan tiga metode atau cara untuk mengukur atau mengevaluasi performa dari sistem yaitu akurasi, presisi, serta *recall*.

Presisi merupakan perbandingan dari data *True Positive* (TP) dengan jumlah data yang saat diprediksi positif. Semakin kecil nilai dari nilai *False Positive* (FP), maka nilai presisi semakin besar. Data *True Positive* (TP) diartikan sebagai data yang ada di kelas positif terdeteksi positif, sedangkan data *False Positive* (FP) sebagai data yang ada di kelas negatif terdeteksi positif. Pada sistem ini, TP diartikan sebagai data terdeteksi mengantuk pada dataset mengantuk, sedangkan FP data tidak mengantuk namun terdeteksi mengantuk.

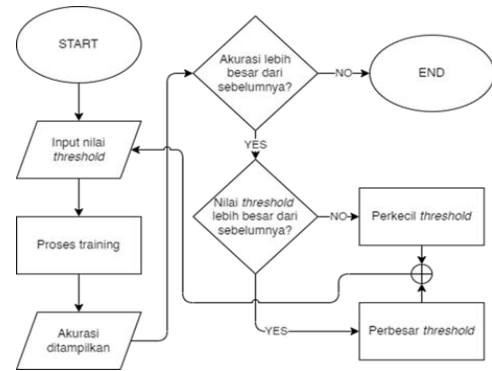
*Recall* merupakan perbandingan data TP terhadap jumlah data yang sebenarnya positif. Semakin kecil nilai *False Negative* (FN), semakin besar pula nilai *recall*. Berbeda dengan FP, data *False Negative* (FN) sebagai data yang ada di kelas positif terdeteksi negatif. Pada sistem ini, FN merupakan data mengantuk namun tidak terdeteksi mengantuk.

Akurasi merupakan pengukuran sistem yang mengambil baik data TP ditambah dengan data TN dan dibagi dengan jumlah seluruh data. Data *True Negative* (TN) merupakan data yang ada di kelas negative terdeteksi negatif. Ini menunjukkan kalau data pada dataset tidak mengantuk juga tidak terdeteksi mengantuk [12].

III. METODE

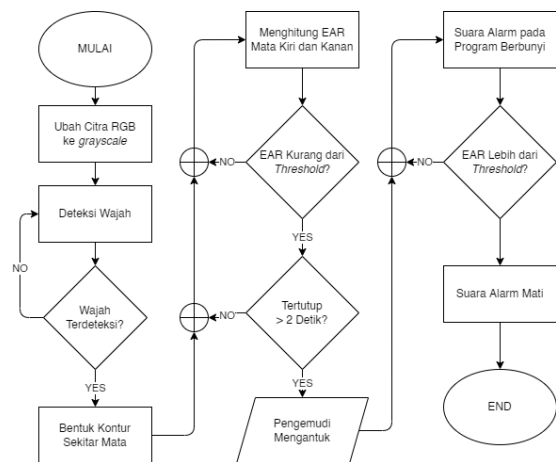
A. Desain Sistem

Terdapat dua tahap yang ada pada sistem ini, yaitu tahap *training* dan tahap deteksi. Diagram alir *training* sistem dapat dilihat pada gambar 1.



GAMBAR 1  
DIAGRAM ALIR TRAINING

Proses *training* data pertama kali, maka dilakukan pemasukan nilai *threshold* terlebih dahulu dengan nilai 0.25. Setelah itu, dilakukan *training* data untuk mengukur akurasi gambar. Setelah selesai *training*, maka nilai akurasi akan ditampilkan. Karena *training* pertama kali, maka dilakukan 2 *training* dengan menaikkan dan menurunkan *threshold*. Jika sudah dilakukan, lalu dicek apakah jika dinaikkan atau diturunkan nilai *threshold* nya, nilai akurasi lebih baik. Jika dinaikkan lebih baik, maka perbesar *threshold*, jika diturunkan lebih baik, perkecil *threshold*. Proses dilakukan hingga nilai akurasi *training* sebelumnya lebih baik daripada *training* yang dijalankan.



GAMBAR 2  
DIAGRAM ALIR DETEKSI

Langkah pertama pada proses deteksi yaitu citra diubah dari warna RGB menjadi *grayscale* untuk mendeteksi wajah secara langsung menggunakan *webcam*. Setelah diubah, deteksi wajah dilakukan menggunakan "*shape\_predictor*" dari Dlib menggunakan 68 *facial landmarks*. Jika wajah terdeteksi, maka yang diambil dari *facial landmarks* hanyalah bagian mata kiri dan mata kanan serta membentuk kontur berupa garis pada mata kiri dan kanan. Setelah itu, didapatkan nilai EAR dengan menghitung jarak Euclidean antar masing-masing enam titik mata untuk mata kiri dan kanan, dan hasilnya ditambah dan dibagi dua. Jika nilai EAR kurang dari *threshold* yang ditetapkan dan mata tertutup lebih dari 2 detik, maka alarm akan berbunyi yang menandakan bahwa pengemudi

mengantuk. Lalu jika EAR sudah lebih dari *threshold*, maka alarm akan nonaktif.

B. Pengambilan Dataset

Dalam penelitian ini, dataset didapatkan dengan cara melakukan pengambilan citra berupa wajah dari pengemudi bus Trans Metro Bandung menggunakan *webcam*. Pengambilan citra wajah pengemudi dilakukan di dalam bus TMB dengan menempatkan *webcam* pada *dashboard* pengemudi bus. Semua citra wajah pengemudi bus dalam format (.jpg) pada penelitian ini untuk tahap pengambilan akurasi, presisi, dan *recall*, serta menggunakan data *real-time* dari kamera untuk percobaan secara *real-time*. Total data citra pada penelitian ini adalah 248 wajah pengemudi yang didapatkan dari merekam wajah pengemudi bus TMB. Dari 248 audio data tersebut, 200 data diperoleh saat pengemudi membuka mata atau tidak mengantuk. Lalu, 48 di antaranya diperoleh saat pengemudi mengantuk. Data tersebut akan digunakan untuk pengambilan akurasi, presisi, serta *recall* dari nilai *threshold* EAR yang diukur.

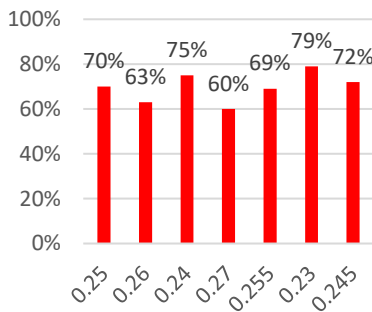
IV. HASIL DAN PEMBAHASAN

A. Perencanaan Analisis Sistem

*Training* sistem ini menggunakan bahasa pemrograman Python. Sistem menggunakan OpenCV untuk mendapatkan gambar pengemudi dari *dataset* yang ada dan menggunakan *facial landmarks* untuk mendeteksi wajah pengemudi dari kelas dataset yang sudah ada. Hasil yang akan dibandingkan yaitu akurasi, presisi, serta *recall* dari nilai *threshold* EAR. Nilai *threshold* yang akan digunakan pada sistem ini yaitu 0.25, 0.26, 0.24, 0.27, 0.255, 0.23, dan 0.245. Hasil akurasi, presisi, serta *recall* dari nilai *threshold* EAR tersebut akan menentukan nilai *threshold* terbaik untuk sistem ini. Nilai *threshold* terbaik tersebut akan digunakan saat proses deteksi secara *real-time*.

B. Pengujian Akurasi *Training* Sistem

Berikut merupakan grafik hasil perbandingan akurasi sistem.

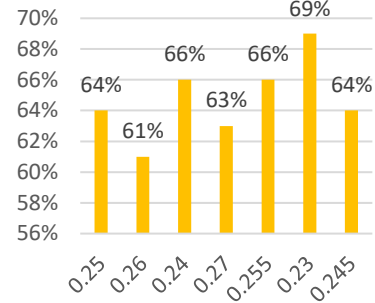


GAMBAR 3 GRAFIK HASIL AKURASI TRAINING

Gambar 3 menunjukkan bahwa nilai *threshold* yang berbeda-beda akan menghasilkan akurasi yang cukup berbeda dan beragam. Nilai *threshold* EAR dengan akurasi tertinggi didapat saat nilai *threshold* diatur pada nilai 0.23. Akurasi tertinggi tersebut mencapai 79%. Sedangkan, akurasi terendah

didapat saat nilai *threshold* 0.27 dengan tingkat akurasi sebesar 60%.

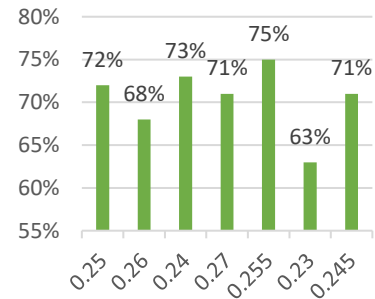
Grafik hasil perbandingan presisi dari *training* sistem ditunjukkan dengan gambar 4,



GAMBAR 4 GRAFIK HASIL PRESISI TRAINING

Presisi terbaik diraih pada saat nilai *threshold* EAR diuji pada 0.23. Nilai presisi terbaik tersebut sebesar 69%. Presisi terendah didapat saat nilai *threshold* diatur ke 0.26, dengan nilai presisi sebesar 61%.

Grafik hasil perbandingan *recall* dari *training* sistem, ditunjukkan dengan gambar 5



GAMBAR 5 GRAFIK 1 HASIL RECALL TRAINING

Recall terbaik sebesar 75% didapat saat pengujian nilai *threshold* 0.255. Hasil *recall* terendah justru didapat pada nilai *threshold* 0.23, yang memiliki nilai akurasi tertinggi. Hasil *recall* terendah tersebut sebesar 63%.

C. Pengujian Deteksi Sistem

Gambar hasil deteksi secara *real-time* dengan menggunakan nilai *threshold* EAR yang sudah ditetapkan pada proses *training* sistem, yaitu 0.23. ditunjukkan dengan gambar 6,



(A)

(B)



(C)

GAMBAR 6

(A) DETEKSI MATA TERBUKA, (B) DETEKSI MATA TERTUTUP, DAN (C) DETEKSI OBJEK TERTIDUR

Gambar 6A menunjukkan mata terdeteksi terbuka lebar. Nilai EAR terdeteksi sebesar 0.342 pada mata. Nilai tersebut lebih besar dari nilai *threshold* EAR yang ditetapkan yaitu 0.23, sehingga objek tidak terdeteksi mengantuk. Pada gambar 6B, mata terdeteksi tertutup. Nilai EAR sebesar 0.122 lebih kecil daripada nilai *threshold* EAR membuat sistem mendeteksi objek mengantuk dan ditampilkan pada sistem bahwa pengemudi mengantuk. Pada gambar 3C, sistem mengindikasikan bahwa objek tertidur. Ini dikarenakan nilai EAR jauh lebih kecil daripada *threshold* yaitu 0.093, serta nilai EAR mata objek sudah lebih dari 4 detik di bawah dari nilai *threshold*.



(A)

(B)

GAMBAR 7 (A)

DETEKSI WAJAH MENUNDUK DAN (B) DETEKSI MINIM CAHAYA

Pada gambar 7A, objek sebenarnya sedang menunduk, namun sistem mendeteksi sebagai objek sedang mengantuk. Akibatnya, sistem menjadi salah mendeteksi atau *False Positive*. Sedangkan pada gambar 7B, sistem tidak mendeteksi wajah pada kondisi minim cahaya. Ini diakibatkan oleh kurangnya kemampuan kamera dalam mendeteksi saat minim cahaya.

## II. KESIMPULAN

Perancangan sistem deteksi kantuk pengemudi bus Trans Metro Bandung secara *real-time* dengan pendekatan rumus *Eye Aspect Ratio* berhasil dilakukan dengan nilai *threshold* EAR paling baik didapatkan pada nilai 0.23 dengan nilai akurasi sebesar 79%. Sedangkan, akurasi terendah didapatkan saat nilai *threshold* 0.27 dengan akurasi 60%. Presisi paling baik didapatkan pada nilai *threshold* 0.23 dengan persentase 69%. Sedangkan, presisi paling rendah didapatkan pada nilai *threshold* 0.26 dengan nilai presisi 61%. *Recall* terbaik didapat saat nilai *threshold* 0.255 dengan nilai 75%. Namun, *recall* terendah justru didapat saat nilai *threshold* 0.23 dengan persentase 63%, yang memiliki akurasi tertinggi.

Intensitas cahaya mempengaruhi deteksi secara *real-time* karena pada saat kondisi gelap, sistem kesulitan mendeteksi mata. Posisi wajah saat deteksi *real-time* juga berpengaruh,

karena saat menunduk terkadang terdeteksi mengantuk walau tidak sedang mengantuk.

Pengembangan untuk penelitian selanjutnya perlu menambahkan dataset pengemudi terutama saat mengantuk untuk menambah akurasi, presisi, serta *recall* dari nilai *threshold* pada sistem dan penambahan kamera *infrared* untuk pendeteksian yang lebih baik pada malam hari. Penelitian ini juga dapat diimplementasikan pada kendaraan umum secara masal supaya mendapatkan nilai yang lebih akurat.

## REFERENSI

- [1] National Highway Traffic Safety Administration, "Traffic Safety Facts", 2017.
- [2] B. Sikka, *Elements of Deep Learning for Computer Vision*, First. New Delhi: BPB Publications, 2021.
- [3] J. Howse and J. Minichino, *Learning OpenCV 4 Computer Vision with Python 3*, Third. London, UK: Packt Publishing, 2020.
- [4] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Sep. 2014, pp. 1867–1874. doi: 10.1109/CVPR.2014.241.
- [5] G. Prihartanto, "Implementasi Sistem Pendeteksi Kantuk pada Pengemudi menggunakan Metode Dlib", *Repository.telkomuniversity.ac.id*, 2019.
- [6] Z. H. Feng, J. Kittler, M. Awais, P. Huber, and X. J. Wu, "Face Detection, Bounding Box Aggregation and Pose Estimation for Robust Facial Landmark Localisation in the Wild," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Aug. 2017, vol. 2017-July, pp. 2106–2111. doi: 10.1109/CVPRW.2017.262.
- [7] A. Saeed, A. Al-Hamadi, and A. Ghoneim, "Head Pose Estimation on Top of Haar-Like Face Detection: A Study Using the Kinect Sensor," *Sensors (Switzerland)*, vol. 15, no. 9, pp. 20945–20966, Aug. 2015, doi: 10.3390/s150920945.
- [8] C. Dewi, R. C. Chen, X. Jiang, and H. Yu, "Adjusting Eye Aspect Ratio for Strong Eye Blink Detection Based on Facial Landmarks," *PeerJ Comput Sci*, vol. 8, Apr. 2022, doi: 10.7717/peerj-cs.943.
- [9] A. Kuwahara, K. Nishikawa, R. Hirakawa, H. Kawano, and Y. Nakatoh, "Eye fatigue estimation using blink detection based on Eye Aspect Ratio Mapping (EARM)," *Cognitive Robotics*, vol. 2, pp. 50–59, 2022, doi: 10.1016/j.cogr.2022.01.003.
- [10] M. Çakar, K. YILDIZ, and Ö. DEMİR, "Thumbnail Selection with Convolutional Neural Network Based on Emotion Detection," *International Journal of Advances in Engineering and Pure Sciences*, Dec. 2021, doi: 10.7240/jeps.900561.
- [11] B. Fernando, A. Sridhar, S. Talebi, J. Waczak, and D. J. Lary, "Unsupervised Blink Detection Using Eye Aspect Ratio Values," Mar. 2022, doi: 10.20944/preprints202203.0200.v1.
- [12] H. Dalianis, *Clinical text mining: Secondary use of electronic patient records*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-78503-5.