

## Sistem Peringatan Tersemat untuk Pengemudi Mengantuk

### Embedded Alert System for Drowsy Drivers

Erika Lety Istikhomah Puspita Sari<sup>1</sup>, I Ketut Agung Enriko<sup>2</sup>

<sup>1</sup>Indonesia Telecommunication & Digital Research Institute, <sup>2</sup>Institut Teknologi Telkom Purwokerto  
Jl. Gegerkalong Hilir No.47, Bandung<sup>1</sup>, Jl. D.I Panjaitan No.128, Purwokerto, Jawa Tengah<sup>2</sup>

---

#### Informasi Makalah

Diajukan : 10 Juni 2023

Diterima : 13 Juni 2023

Diterbitkan : 16 Juni 2023

---

#### Kata Kunci:

*Driver drowsiness Detection, Machine Learning, Facial Feature, Deep Learning, Embedded*

---

#### Keyword:

Driver drowsiness Detection, Machine Learning, Facial Feature, Deep Learning, Embedded

---

#### INTISARI

Pendeteksian *Driver Drowsiness* (DDD) merupakan teknologi keselamatan kendaraan penting yang dirancang untuk mencegah kecelakaan akibat kantuk pengemudi. Dalam penelitian ini, pendekatan baru diajukan menggunakan model jaringan saraf konvolusi (CNN) ringan yang terdiri dari 44.853 parameter. Berkat ke ringannya, model ini bekerja secara efisien bahkan pada perangkat dengan sumber daya terbatas. Hasil percobaan menunjukkan kinerja kompetitif model ini dibandingkan dengan model yang ada dengan ukuran input dan jumlah parameter yang lebih besar. Dalam hal akurasi, metode ini mencapai akurasi sebesar 92,06% pada dataset *Curtin Emotion Wheels* (CEW) yang mengesankan. Bahkan dalam kondisi pencahayaan yang sulit, performa model ini tetap luar biasa jika digabungkan dengan kamera termal. Secara khusus, model ini mencapai akurasi yang luar biasa sebesar 95,10% pada jarak kamera 0,3-meter dari wajah pengemudi. Selain itu, metode ini memiliki karakteristik kecepatan yang sangat baik, sehingga cocok digunakan pada perangkat tertanam. Kecepatan rata-rata perangkat Raspberry Pi 4 diperkirakan mencapai 5 frames per detik (FPS). Hal ini menunjukkan kepraktisan dan kelayakan penerapan pendekatan ini dalam skenario waktu nyata, yang semakin meningkatkan keselamatan pengemudi.

---

#### ABSTRACT

Driver Drowsiness Detection (DDD) is an important vehicle safety technology designed to prevent accidents caused by driver drowsiness. In this study, a novel approach is proposed using a lightweight Convolutional Neural Network (CNN) model consisting of 44,853 parameters. Due to its lightweight nature, this model operates efficiently even on devices with limited resources. Experimental results demonstrate the competitive performance of this model compared to existing models with larger input sizes and parameter counts. In terms of accuracy, this method achieves an impressive accuracy of 92.06% on the Curtin Emotion Wheels (CEW) dataset. Even in challenging lighting conditions, the performance of this model remains exceptional when combined with a thermal camera. Specifically, the model achieves an outstanding accuracy of 95.10% at a distance of 0.3 meters from the driver's face. Additionally, this method exhibits excellent speed characteristics, making it suitable for embedded devices. The average speed on a Raspberry Pi 4 device is estimated to be 5 frames per second (FPS). This demonstrates the practicality and feasibility of implementing this approach in real-time scenarios, further enhancing driver safety.

---

#### Korespondensi Penulis:

Erika Lety Istikhomah Puspita Sari  
Indonesia Telecommunication and Digital Research Institute (ITDRI)  
Jl. Gegerkalong Hilir No.47, Bandung  
Email : erikalety459@gmail.com

## 1. PENDAHULUAN

Saat ini, rasa kantuk pengemudi menjadi salah satu penyebab utama kecelakaan lalu lintas. Sangat wajar bagi pengemudi yang berkendara jarak jauh tertidur di belakang kemudi. Mengemudi terus-menerus bisa melelahkan dan membosankan. Pengemudi mungkin merasa mengantuk bahkan tertidur karena kurangnya aktivitas. Namun, banyak pendukung kendaraan otonom berpendapat bahwa industri pertama yang sepenuhnya diubah oleh mobil/truk otonom (bahkan sebelum kendaraan komersial) adalah truk jarak jauh. Sistem DDD ini diharapkan dapat mengurangi kecelakaan lalu lintas. Metode untuk mendeteksi kantuk diklasifikasikan menjadi metode subjektif dan objektif.

Dalam deteksi objektif, tidak ada umpan balik yang diberikan kepada pengemudi dan deteksi didasarkan pada karakteristik fisik pengemudi, sedangkan deteksi subjektif terkait dengan aspek fisik pengemudi. Jenis deteksi lensa dibagi lagi menjadi jenis kontak dan non-kontak. Sistem yang diusulkan didasarkan pada metode non-kontak karena lebih murah dibandingkan dengan metode kontak. Sistem pendeteksi kantuk tak lepas dari teknologi sensor yang digunakan untuk mendeteksi fitur wajah pengemudi. Tiga teknologi sensor yang umum digunakan adalah kamera RGB, sensor kedalaman, dan perangkat seluler [1].

Salah satu teknik deteksi tidur adalah kerangka deteksi *landmark* wajah berbasis database. Penelitian terkait analisis wajah telah menarik perhatian dalam beberapa tahun terakhir dan diharapkan dapat digunakan di berbagai bidang seperti perekaman video, pemantauan, pernikahan, perawatan kesehatan, dan pemantauan kebiasaan sehari-hari pengguna.

Salah satu teknik deteksi kantuk yang telah dikembangkan adalah kerangka deteksi *landmark* wajah berbasis database. Penelitian tentang analisis wajah telah menarik perhatian dalam beberapa tahun terakhir dan memiliki potensi aplikasi yang luas dalam berbagai bidang seperti perekaman video, pemantauan, perawatan kesehatan, dan pemantauan kebiasaan sehari-hari pengguna.

Tujuan dari fitur wajah adalah untuk mengenali aktivitas manusia dalam lingkungan tertentu. Pendekatan berbasis perilaku mencoba memprediksi kantuk berdasarkan perilaku seperti menguap, berkedip per menit, gerakan kepala, ekspresi wajah, dan rasio bentuk mata. Salah satu pendekatan yang paling populer dalam kategori ini adalah visi komputer. Dalam visi komputer, pemrosesan gambar wajah pengemudi digunakan untuk memprediksi kantuk pengemudi. Algoritme pembelajaran mendalam dalam visi komputer terbukti sangat efektif karena memiliki tingkat akurasi yang tinggi dalam tugas klasifikasi gambar [2].

Berdasarkan pemahaman dan analisis terhadap metode-metode yang ada, penelitian ini mengusulkan metode baru yang akan mengatasi beberapa kekurangan yang ada dalam deteksi kantuk pengemudi. Rincian tentang metode yang diusulkan dan alasan pemilihannya akan dijelaskan lebih lanjut di bagian selanjutnya naskah ilmiah ini.

## 2. METODE PENELITIAN

Dalam berbagai pendekatan, langkah-langkah diambil untuk deteksi *landmark* wajah, pelacakan objek, dan metode deteksi kegugupan pengemudi dapat digunakan dalam dua cara, yaitu pendekatan kontak atau pendekatan tanpa kontak. Sifat teknik ini sangat bergantung pada domain aplikasi. *Landmark* wajah merupakan fase yang penting namun menantang dalam deteksi kegugupan. Ini telah diterapkan untuk memecahkan masalah seperti penyesuaian wajah, estimasi posisi kepala, pertukaran wajah, deteksi kedipan, dll. Biasanya digunakan untuk menandai dan menggambarkan area penting wajah, alis, hidung, dan mulut [3].

Dalam beberapa tahun terakhir, penelitian terkait Analisis Wajah menjadi perhatian dan diharapkan dapat diterapkan dalam berbagai bidang, seperti pengambilan video, pemantauan, kehidupan bantuan, perawatan kesehatan, dan pengamatan kebiasaan sehari-hari pengguna [4]. Fitur Wajah bertujuan untuk mengidentifikasi aktivitas yang dilakukan oleh orang dalam suatu lingkungan. Sistem Deteksi Kantuk tidak dapat dipisahkan dari teknologi sensor yang digunakan untuk mengenali fitur wajah pengemudi. Tiga teknologi sensor yang umum digunakan adalah kamera RGB, sensor kedalaman, dan perangkat *wearable* [5].

Salah satu teknik yang digunakan dalam deteksi kegugupan adalah kerangka kerja sistem identifikasi *landmark* wajah berbasis data [1]. Pengenalan tingkat kantuk manusia menggunakan kamera konvensional agak terbatas karena tidak dapat mendeteksi informasi internal tubuh manusia dan berbagai peristiwa yang sering tidak terlihat dengan baik. Penggunaan teknologi sensor yang menghasilkan peta kedalaman kondisi pengemudi akan bermanfaat dalam menyediakan lebih banyak detail untuk kebutuhan deteksi tingkat kantuk [2].

Secara umum, kegugupan dapat didefinisikan sebagai keadaan di mana kepala condong ke atas, condong ke salah satu sisi, condong ke belakang, atau tetap tidak bergerak selama beberapa detik. Selain itu, tingkat kantuk dapat diidentifikasi melalui gerakan mata. Sebelum terjadi kegugupan, mungkin tidak ada gerakan pupil, frekuensi kedipan yang berkurang, atau keadaan menguap [1].

Dalam penelitian ini, pendekatan yang menggunakan identifikasi visual dan menggunakan sensor untuk mendeteksi pengemudi yang teridentifikasi menggunakan sensor. *Eye Aspect Ratio* adalah salah satu teknik

yang tersedia yang digunakan dalam deteksi kegugupan untuk memperkirakan kondisi pengemudi. Dalam pendekatan sensor, dua sensor digunakan, yaitu sensor denyut nadi dan sensor gelombang otak. Sensor denyut nadi yang digunakan adalah *nama sensor* yang banyak digunakan dan umum digunakan dalam perangkat mikrokontroler. Sensor denyut nadi digunakan untuk mendeteksi perilaku denyut nadi saat pengemudi mengantuk [6]. Dalam pendekatan visual, penelitian ini menggunakan kamera yang mampu mengidentifikasi beberapa parameter seperti postur tubuh, ekspresi wajah, rambut, dan kondisi mata. Untuk mengidentifikasi kegugupan secara visual, penelitian ini menggunakan beberapa perangkat lunak seperti *python*, *mediapipe*, *jupyter*, *openCV*, dan *scikitLearn* [7]. Untuk *pipeline* dalam penelitian ini, dimulai dengan membaca aliran video dan memilih berapa banyak frame dalam video di mana frame tersebut akan diproses dan diidentifikasi untuk menentukan kondisi pengemudi dalam keadaan kantuk atau tidak [8].

Proses klasifikasi melibatkan beberapa teknik. Analisis Komponen Utama (PCA) umum digunakan untuk menemukan jumlah data yang lebih kecil dan tidak berkorelasi untuk mengurangi dimensi fitur [1]. Teknik klasifikasi lain yang diurutkan berdasarkan perkembangan historis adalah K-Nearest Neighbor (KNN) classifier [1] dan SVM [1, 2, 7, 10, 13], Hyper-sphere Multi-class SVM [6], Naive-Bayes [9, 10], Random Forest [10], *Multilayer Perceptron* (MLP) [7, 10], *Convolutional Neural Networks* (CNN) [4, 8, 13], LSTM [12, 14, 15, 16, 17], ResNet [20], Transformer [21, 22, 23]. Kompleksitas komputasi dari algoritma-algoritma di atas dalam memprediksi satu sampel berdasarkan yang tertinggi hingga yang terendah adalah  $O(kN^2d)$  untuk KNN,  $O(Knd^2)$  untuk Lapisan Konvolusi dalam CNN,  $O(sd^2)$  untuk lapisan rekurent dalam LSTM,  $O(s^2d)$  untuk lapisan self-attention dalam Transformer,  $O(d(n+o))$  untuk MLP/ANN,  $O(Td)$  untuk *Random Forest*,  $O(dc)$  untuk Naïve Bayes, dan  $O(d)$  untuk SVM. Di mana  $N$  adalah jumlah sampel dalam dataset,  $k$  adalah jumlah tetangga,  $K$  adalah jumlah kernel,  $d$  adalah jumlah fitur input,  $s$  adalah jumlah urutan,  $T$  adalah jumlah pohon,  $n$  adalah jumlah neuron,  $o$  adalah jumlah neuron keluaran, dan  $c$  adalah jumlah kelas. KNN tidak digunakan dalam eksperimen ini karena memiliki kompleksitas komputasi tertinggi dalam membuat prediksi. Selain itu, dengan adanya  $N$ , waktu eksekusi prediksi sangat dipengaruhi oleh ukuran dataset. Meskipun kompleksitasnya dapat dikurangi menjadi  $O(\log(d)+N)$ , waktu eksekusi tetap dipengaruhi oleh ukuran dataset [24]. SVM, *Naïve Bayes*, dan *Random Forest* memiliki kompleksitas yang hampir sama dengan model pembelajaran mesin yang ringan. Karena kami melakukan klasifikasi menggunakan fitur titik kunci yang diperoleh dari Fitur Wajah 3D, lapisan konvolusi tidak diperlukan lagi.

Video input diekstraksi menjadi beberapa baris frame/gambar untuk analisis pada tahap preprocessing. Untuk setiap prediksi, masukan yang digunakan adalah 3 frame secara berurutan dalam jendela waktu kecil. Metode yang telah kami kembangkan menggabungkan teknik tradisional perhitungan jarak euclidean dari data fitur wajah yang diperoleh melalui proses Deteksi *Landmark* Wajah dengan informasi tentang arah pandangan wajah. Sehingga proses pengambilan keputusan untuk menentukan kondisi mata tertutup/terbuka menjadi lebih fleksibel dengan besarnya perbedaan ambang batas dalam lebar mata yang berubah seiring perbedaan pandangan pengemudi.

#### A. *Preprocessing*

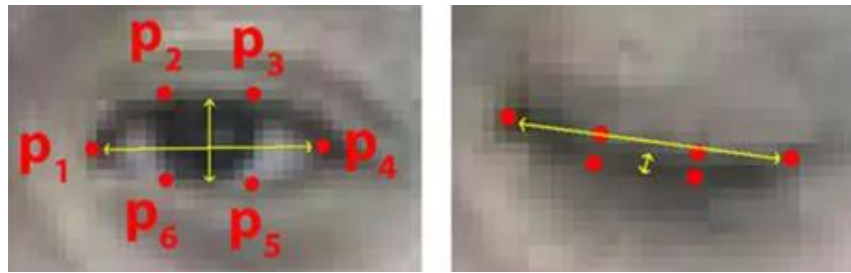
Proses deteksi kondisi mata dilakukan dalam jendela waktu 1,5 detik, kira-kira 3 frame berdasarkan kemampuan komputasi Raspberry Pi 4. Rentang waktu ini adalah estimasi durasi pengemudi yang dinyatakan menutup mata di tengah waktu dan tidak secara keliru dikenali sebagai kedipan.

#### B. *Face Detection*

Kami menggunakan model dari *BlazeFace*. *BlazeFace* adalah detektor wajah yang ringan dan memiliki performa baik yang dirancang khusus untuk inferensi GPU *mobile*. Ia berjalan dengan kecepatan 200-1000+ FPS pada perangkat flagship. Kinerja *super-realtime* ini memungkinkannya diterapkan pada pipa *augmented reality* apa pun yang membutuhkan wilayah minat wajah yang akurat sebagai masukan untuk model tugas tertentu, seperti estimasi titik kunci wajah 2D/3D atau geometri, fitur atau ekspresi wajah, dan segmentasi wajah.

#### C. *Face Landmark Identification*

Bekerja pada gambar penuh atau bingkai video, dapat berfungsi sebagai langkah pertama dari hampir semua aplikasi computer vision terkait wajah, seperti titik kunci wajah 2D/3D, estimasi geometri kontur atau permukaan, klasifikasi fitur atau ekspresi wajah, dan segmentasi wilayah wajah. Tugas berikutnya dalam pipa computer vision dapat ditentukan dalam hal pemotongan wajah yang tepat. Dikombinasikan dengan estimasi beberapa titik kunci wajah yang diberikan oleh *BlazeFace*, pemotongan ini juga dapat diputar sehingga wajah di dalamnya berada di tengah, dinormalisasi skala, dan memiliki sudut *roll* yang mendekati nol. Ini menghilangkan kebutuhan invariansi translasi dan rotasi yang signifikan dari model tugas khusus, memungkinkan alokasi sumber daya komputasi yang lebih baik.

Gambar 1. Tampilan *Face Landmark*

#### D. *EAR Calculation*

Untuk menghitung tingkat kegugupan pengemudi, penutupan mata harus didefinisikan. *Eye aspect ratio* (EAR) adalah rasio tinggi terhadap lebar mata, dan ini digunakan untuk membedakan apakah mata terbuka atau tertutup. Ketika EAR lebih besar dari nilai ambang batas 0,2, mata dianggap tertutup [10]. Penentuan lebar mata ditentukan berdasarkan 6 titik yang diambil dari fitur *landmark* mata. Kemudian, estimasi rasio lebar akan dilakukan melalui nilai jarak *euclidean*.

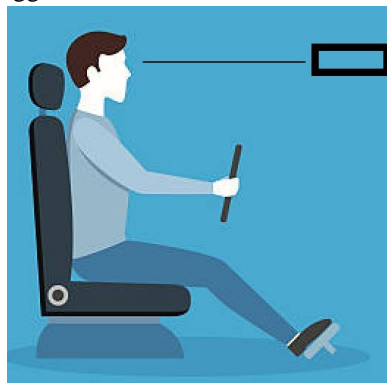
$$d(x,y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \dots (1)$$

#### E. *Eye Classifier*

Kami menggunakan empat pengklasifikasi untuk memprediksi tindakan. Pengklasifikasi tersebut diuji untuk melihat yang paling optimal dalam perhitungan dan akurasi. Pengklasifikasi yang kami gunakan dalam eksperimen ini adalah *SVM*, *Decision Tree*, *Random Forest*, dan *Naïve Bayes*. Perbedaan dalam jumlah lingkungan pengemudi digunakan sebagai bahan uji untuk menentukan model mana yang cocok untuk metode kami. Kami melakukan eksperimen terhadap pengaruh perbedaan penggunaan kacamata atau tidak. Enam fitur yang diperoleh dari langkah sebelumnya digunakan sebagai input untuk masing-masing pengklasifikasi.

#### F. *Simulation*

Proses simulasi dilakukan dalam kondisi nyata dengan kondisi saat siang dan malam. Dengan berbagai perbedaan intensitas cahaya dan penggunaan kacamata. Posisi kamera ditempatkan pada tingkat mata.



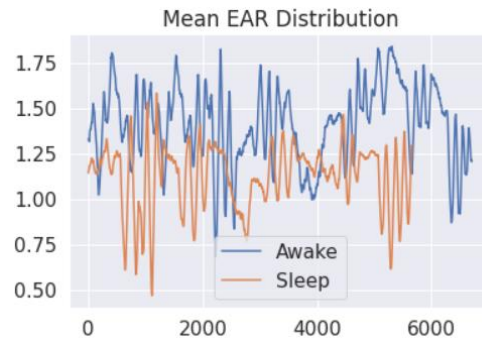
Gambar 2. Tampilan Mengemudi

#### G. *Metric Evaluation*

Kinerja dihitung melalui empat metrik: akurasi, presisi, recall, dan skor F1. Semakin tinggi nilai metrik tersebut, semakin baik kinerja modelnya. Selain itu, kecepatan pelatihan dan inferensi juga diperhatikan untuk mengetahui seberapa efisien model-model tersebut.

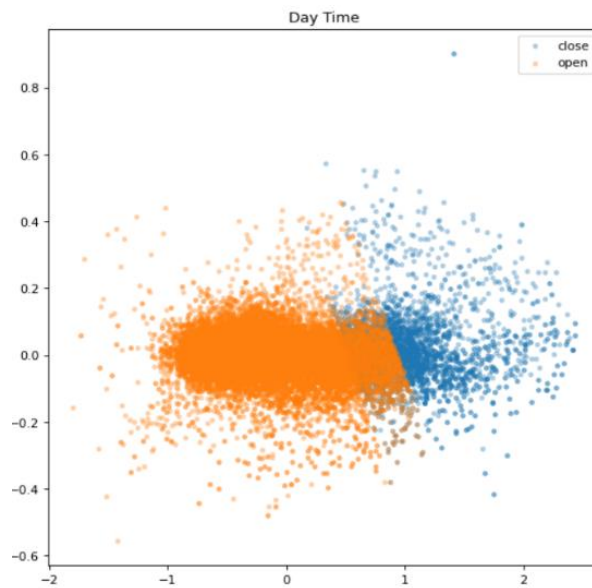
### 3. HASIL DAN PEMBAHASAN

Dua eksperimen dilakukan untuk menyelidiki model yang dikembangkan untuk mengimplementasikan sistem Deteksi Kegugupan. Pertama, mereka dievaluasi untuk mengklasifikasikan kondisi mata menggunakan tiga model yang berbeda dalam kondisi siang hari. Kedua, mereka dievaluasi untuk mengklasifikasikan kondisi mata dengan menggunakan model yang sama dalam kondisi malam hari. Akhirnya, kinerja dan waktu pemrosesan mereka dievaluasi dengan cermat.



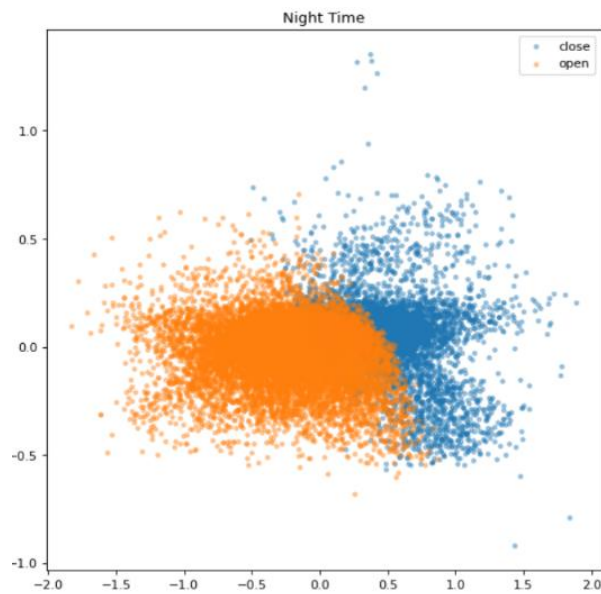
Gambar 3. Tampilan *Mean EAR Distribution*

A. Klasifikasi pada kondisi waktu siang hari



Gambar 4. 2D PCA Diagram Scatter Fitur untuk Kondisi Siang Hari

B. Klasifikasi pada kondisi waktu malam hari



Gambar 5. Diagram Scatter Fitur untuk Kondisi Malam Hari

Tabel 1. Hasil Perbandingan Percobaan dalam Kondisi Siang Hari dan Malam Hari

Model	Distance	Precision	Recall	F1 Score	Accuracy
Siang Hari	0.2 m	95.50%	94.30%	95.20%	95.10%
	0.5 m	94.50%	94.30%	95.05%	90.05%
Malam Hari	0.2 m	93.96%	90.30%	94.15%	92.06%
	0.5 m	97.53%	88.27%	89.86%	87.10%

Pada tabel 1, dapat melihat perbandingan kinerja model yang diusulkan pada kondisi siang dan malam hari. Dapat dilihat bahwa akurasi model tersebut cukup baik baik pada siang maupun malam hari dengan akurasi di atas 90%. Pada siang hari, akurasi terbaik mencapai 95,19% dan 87,10% pada malam hari. Semakin dekat jarak pengemudi, akurasi cenderung semakin baik, tentunya karena gambar akan lebih jelas. Pada kondisi malam, jarak antara 0,5 m dan 0,2 m memiliki kesenjangan yang cukup jauh dibandingkan dengan jarak kamera lainnya.

### C. Processing Time

Pada tabel 3, dapat melihat perbandingan kecepatan inferensi untuk setiap sub-proses dari model *pipeline* yang kami bangun. Dapat dilihat bahwa proses paling berat terdapat pada *Face Landmark Detection* dengan kecepatan 0,05 detik pada perangkat dengan GPU dan 0,10 detik pada perangkat Raspberry Pi 4. Perbedaan kecepatan antara GPU desktop dan Raspberry Pi 4 sekitar 2 kali lebih cepat dibandingkan dengan GPU desktop. FPS yang diperkirakan dapat diperoleh oleh Raspberry Pi 4 adalah 5 FPS, ini menunjukkan bahwa metode yang digunakan tepat untuk dijalankan pada perangkat *embedded*.

Tabel 2. Processing Time Proposed Method

Device	Model	Mean Speed (Seconds)	Total FPS
Desktop GPU	Face Landmark Detection	0.05	15
	Eye Classification	0.000088	
	Face Landmark Detection	0.10	
Raspberry Pi 4	Eye Classification	0.01	5

## 4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan Model DDD (*Driver Drowsiness Detection*) yang menggunakan metode klasifikasi untuk mendeteksi kondisi mengantuk pada pengemudi. Hasil eksperimen menunjukkan bahwa model yang diusulkan mampu mencapai keseimbangan antara akurasi dan kecepatan. Meskipun memiliki ukuran input dan jumlah parameter yang lebih kecil dibandingkan model yang tersedia, metode ini masih mampu bersaing dengan kinerja yang baik.

Selain itu, metode ini terbukti efektif dalam berbagai situasi, termasuk dalam kondisi pencahayaan rendah dengan bantuan kamera inframerah. Pada jarak kamera 0,2 meter dari wajah pengemudi, metode ini mencapai akurasi terbaik sebesar 95,10%.

Dari segi kecepatan, metode ini juga cocok untuk diimplementasikan pada perangkat *embedded*, seperti Raspberry Pi 4, dengan kecepatan rata-rata sekitar 5 frame per detik (FPS).

Namun, perlu dicatat bahwa dalam naskah ini tidak dijelaskan secara rinci mengenai sistem peringatan atau "*alert system*" yang digunakan setelah terdeteksi kondisi mengantuk pada pengemudi. Informasi lebih lanjut mengenai sistem peringatan yang diimplementasikan dan bagaimana metode ini dapat berkontribusi dalam meningkatkan keselamatan pengemudi perlu ditambahkan dalam naskah ilmiah ini.

**DAFTAR PUSTAKA**

- [1] K. Ashwini, R. Amutha, and S. Aswin Raj, "Skeletal Data based Activity Recognition System," *Proc. 2020 IEEE Int. Conf. Commun. Signal Process. ICCSP 2020*, pp. 444–447, 2020, doi: 10.1109/ICCSP48568.2020.9182132.
- [2] C. Changhong and G. Zongliang, "Action recognition from a different view," *China Commun.*, vol. 10, no. 12, pp. 139–148, 2013, doi: 10.1109/CC.2013.6723886.
- [3] N. Dawar and N. Kehtarnavaz, "Continuous detection and recognition of actions of interest among actions of non-interest using a depth camera," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 4227–4231, 2018, doi: 10.1109/ICIP.2017.8297079.
- [4] E. P. Ijjina and C. K. Mohan, "Human action recognition based on recognition of linear patterns in action bank features using convolutional neural networks," *Proc. - 2014 13th Int. Conf. Mach. Learn. Appl. ICMLA 2014*, pp. 178–182, 2014, doi: 10.1109/ICMLA.2014.33.
- [5] O. C. Ann and L. B. Theng, "Human activity recognition: A review," *Proc. - 4th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2014*, no. March, pp. 389–393, 2014, doi: 10.1109/ICCSCE.2014.7072750.
- [6] J. Liu, J. Yang, Y. Zhang, and X. He, "Action recognition by multiple features and hyper-sphere multi-class SVM," *Proc. - Int. Conf. Pattern Recognit.*, pp. 3744–3747, 2010, doi: 10.1109/ICPR.2010.912.
- [7] M. Mobark, S. Chuprat, and T. Mantoro, "Improving the accuracy of complex activities recognition using accelerometer-embedded mobile phone classifiers," *Proc. 2nd Int. Conf. Informatics Comput. ICIC 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/IAC.2017.8280606.
- [8] S. Park and D. Kim, "Study on 3D action recognition based on deep neural network," *ICEIC 2019 - Int. Conf. Electron. Information, Commun.*, pp. 1–3, 2019, doi: 10.23919/ELINFOCOM.2019.8706490.
- [9] T. Perumal, Y. L. Chui, M. A. Bin Ahmadon, and S. Yamaguchi, "IoT based activity recognition among smart home residents," *2017 IEEE 6th Glob. Conf. Consum. Electron. GCCE 2017*, vol. 2017-Janua, no. Gcce, pp. 1–2, 2017, doi: 10.1109/GCCE.2017.8229478.
- [10] P. M. D. Alex, A. Ravikumar, J. Selvaraj, and A. Sahayadhas, "Research on human activity identification based on image processing and artificial intelligence," *Int. J. Eng. Technol.*, vol. 7, no. 3.27 Special Issue 27, pp. 174–178, 2018, doi: 10.14419/ijet.v7i3.27.17754.
- [11] H. Wang and L. Wang, "Cross-Agent Action Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2908–2919, 2018, doi: 10.1109/TCSVT.2017.2746092.
- [12] P. Mishra, S. Dey, S. S. Ghosh, D. B. Seal, and S. Goswami, "Human Activity Recognition using Deep Neural Network," *2019 Int. Conf. Data Sci. Eng. ICDSE 2019*, pp. 77–83, 2019, doi: 10.1109/ICDSE47409.2019.8971476.
- [13] M. M. Hossain Shuvo, N. Ahmed, K. Nouduri, and K. Palaniappan, "A hybrid approach for human activity recognition with support vector machine and 1d convolutional neural network," *Proc. - Appl. Imag. Pattern Recognit. Work.*, vol. 2020-Octob, pp. 6–10, 2020, doi: 10.1109/AIPR50011.2020.9425332.
- [14] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM Networks for Mobile Human Activity Recognition," no. January 2016, 2016, doi: 10.2991/icaite-16.2016.13.
- [15] S. Yu, L. Qin, and Q. Yin, "A C-LSTM Neural Network for Human Activity Recognition Using Wearables," 2018 Int. Symp. Sens. Instrum. IoT Era, ISSI 2018, no. 6140017010001, pp. 1–6, 2018, doi: 10.1109/ISSI.2018.8538129.
- [16] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "Skeleton-based human activity recognition using ConvLSTM and guided feature learning," *Soft Comput.*, vol. 26, no. 2, pp. 877–890, 2022, doi: 10.1007/s00500-021-06238-7.
- [17] W. Zhu et al., "Co-Occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," 30th AAAI Conf. Artif. Intell. AAAI 2016, no. September 2017, pp. 3697–3703, 2016.
- [18] E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante, "A Human Activity Recognition System Using Skeleton Data from RGBD Sensors," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/4351435.
- [19] S. Gaglio, G. Lo Re, and M. Morana, "Human Activity Recognition Process Using 3-D Posture Data," *IEEE Trans. Human-Machine Syst.*, vol. 45, no. 5, pp. 586–597, 2015, doi: 10.1109/THMS.2014.2377111.
- [20] H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Learning and recognizing human action from skeleton movement with deep residual neural networks," pp. 25 (6 .)-25 (6 .), 2017, doi: 10.1049/cp.2017.0154.
- [21] B. Jiang, J. Yu, L. Zhou, K. Wu, and Y. Yang, "Two-Pathway Transformer Network for Video Action Recognition," pp. 1089–1093, 2021, doi: 10.1109/icip42928.2021.9506453.
- [22] J. He and S. Gao, "TBSN: Sparse-Transformer Based Siamese Network for Few-Shot Action Recognition," 2021 2nd Inf. Commun. Technol. Conf. ICTC 2021, pp. 47–53, 2021, doi: 10.1109/ICTC51749.2021.9441568.
- [23] H. Seong, J. Hyun, and E. Kim, "Video multitask transformer network," *Proc. - 2019 Int. Conf. Comput. Vis. Work. ICCVW 2019*, no. 1, pp. 1553–1561, 2019, doi: 10.1109/ICCVW.2019.00194.

