

# Sistem Peringatan Dini Kantuk pada Pengemudi Malam Hari Menggunakan Metoda Facial Landmark Detection Berbasis Raspberry Pi 3 Modul B

I Komang Yoga Tri Pranata<sup>1</sup>, Cipta Ramadhani<sup>1</sup>, Giri Wahyu Wiriasto<sup>1</sup>

<sup>1</sup>Electrical Engineering, Mataram University. Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

## INFO ARTIKEL

### Article history

Received July 27, 2023  
Revised August 14, 2023  
Accepted August 15, 2023

### Keywords :

Sleepiness;  
Facial Landmark;  
Raspberry Pi;  
Driver;

## ABSTRACT

*Sleepiness is a condition where a person wants to rest so hence eye blinking increases. Humans generally need 8 hours of sleep over night to achieve optimal performance. The feeling of sleepy has a negative impact on the driver so that he becomes distracted or lack of concentration that can cause an accident that harms passenger seriously, even causes death. Therefore, in this paper, a study has been taken to minimize drowsiness while driving, in this study a tool will be made where the tool can detect or determine drowsiness while driving using Facial Landmark Detection which is implemented on the Raspberry Pi 3 Module B. The output of this tool in the form of an LED as a sign of drowsiness and a buzzer as a sign of a drowsiness alarm. The system created in this study shows that the drowsiness is able to be detected as expected. Based on the experiment, the data shows that the system detect sleepiness with a distance of 40cm by 80%, 50cm by 100%, and 60cm by 80% and the total result of 80%.*

Kantuk adalah kondisi seseorang ingin beristirahat sehingga kedipan mata mengalami peningkatan. Manusia pada umumnya membutuhkan tidur selama 8 jam setiap malamnya agar mencapai kinerja yang optimal. Rasa kantuk ini dapat menjadi dampak negatif bagi pengemudi sehingga menjadi tidak fokus, konsentrasi menjadi hilang dan merugikan penumpang sehingga bisa menyebabkan kematian. Oleh karena itu pada penelitian ini untuk meminimalisir rasa kantuk pada saat mengemudi, maka dalam penelitian ini akan dibuat sebuah alat dimana alat tersebut dapat mendeteksi atau mengetahui kondisi mengantuk pada saat mengemudi menggunakan Facial Landmark Detection yang diimplementasikan ke Raspberry Pi 3 Modul B. Output alat ini berupa LED sebagai tanda kantuk dan buzzer sebagai tanda alarm kantuk. Sistem yang dibuat pada penelitian ini menunjukkan bahwa dapat mendeteksi mata kantuk sesuai yang diharapkan. Berdasarkan hasil pengujian, sistem dapat mendeteksi mata kantuk dengan jarak 40cm sebesar 80%, 50cm sebesar 100%, dan 60cm sebesar 80% dan hasil keseluruhan sebesar 80%.

### Corresponding Author:

Cipta Ramadhani, Teknik Elektro Universitas Mataram, Jalan Majapahit 62, Mataram, 83115, Indonesia  
Email: [cipta.ramadhani@unram.ac.id](mailto:cipta.ramadhani@unram.ac.id)

## 1. PENDAHULUAN

Teknologi sudah makin berkembang pesat seperti yang kita ketahui bahwa di Indonesia rata-rata menggunakan kendaraan. Kendaraan ini ada bermacam macam tipenya, seperti sepeda motor, mobil, truk, dan masih banyak lagi. Teknologi ini diciptakan untuk memudahkan seseorang menuju suatu tempat dengan cepat tanpa perlu berjalan kaki.

Berdasarkan data dari Korlantas Polri yang di publikasikan oleh Kementerian Perhubungan menyatakan angka kecelakaan di Indonesia mencapai 103.645 kasus pada tahun 2021. Jumlah tersebut lebih tinggi dari

tahun 2020 yaitu sebesar 100.028 kasus kecelakaan. Kecelakaan menurut Komite Nasional Keselamatan Transportasi (KNKT) menyatakan bahwa 80 persen kecelakaan diakibatkan karena mengantuk dan letih [1].

Manusia pada umumnya membutuhkan tidur selama 8 jam setiap malamnya agar mencapai kinerja yang optimal. Rasa kantuk ini dapat menjadi dampak negatif bagi pengemudi sehingga menjadi tidak fokus, konsentrasi menjadi hilang dan merugikan penumpang sehingga bisa menyebabkan kematian. Mengantuk merupakan masalah yang sangat penting untuk diminimalisir. Oleh karena itu disarankan sebelum mulai kegiatan terlebih dahulu sarapan pagi agar otak tidak kekurangan oksigen dan menjadi optimal [2]. Rasa kantuk biasanya dialami pada malam hari di mana malam hari adalah waktu untuk beristirahat. Pada kondisi mengantuk, seseorang akan mengalami peningkatan kedipan mata. Selain itu, seseorang akan mengalami *microsleeps* atau tidur sekejap dengan durasi penutupan mata berkisar 5 detik atau lebih [3]. Faktor penyebab jumlah kecelakaan lalu lintas adalah faktor manusia, dimana memiliki persentase 69.7%. Salah satu faktor pengemudi mengantuk adalah mengemudi pada malam hari, kurang istirahat, jarak tempuh yang panjang dan tidak adanya teman yang menemani berbicara sehingga pengemudi menjadi bosan, jenuh dan akhirnya mengantuk [4].

Paper ini melaporkan hasil penelitian yang dilakukan untuk untuk meminimalisir rasa kantuk pada saat mengemudi. Dalam penelitian ini dibuat sebuah alat dimana alat tersebut bisa mendeteksi atau mengetahui kondisi mengantuk pada saat mengemudi. Alat ini akan diterapkan di depan pengemudi sehingga alat dapat mendeteksi kantuk dengan cara mendeteksi mata dan alat ini dapat memantau kondisi dimana dilakukan perekaman wajah otomatis secara real time yang dapat diproses pada pengolahan citra digital. Berdasarkan latar belakang di atas, paper ini mengangkat judul “Sistem Peringatan Dini Kantuk Pada Pengemudi Malam Hari Menggunakan Metode Facial Landmark Detection Berbasis Raspberry Pi 3 Modul B”.

## 2.1 Kantuk



**Gambar 1 Kantuk**

Kantuk (*drowsiness*) adalah keadaan yang dimana seseorang ingin istirahat. Gambar 1 menunjukkan posisi pengemudi pada saat kantuk. Manusia pada umumnya membutuhkan tidur selama 8 jam setiap malamnya agar mencapai kinerja yang optimal. Kekurangan tidur dapat mengakibatkan kantuk sehingga terjadi peningkatan jumlah kedipan mata hingga terjadi adanya *microsleeps* dengan durasi penutupan mata berkisar 5 detik atau lebih [2].

## 2.2 Dlib

Dlib adalah sebuah library machine learning yang ditulis dalam Bahasa pemrograman C++ dan berfungsi untuk menganalisis bagian wajah dengan mengesktrak nilai gambar. Dengan mengekstrak nilai pada wajah manusia dlib akan menghasilkan 128 dimensional feature [5].

## 2.3 Facial Landmark

Gambar 2 [6] merupakan facial landmark yang ada pada wajah seseorang. Facial landmark digunakan untuk mendeteksi wajah berdasarkan titik-titik pada wajah manusia yang berfungsi sebagai titik jangkar pada

wajah. Facial landmark menggunakan dataset dari iBUG 300-W sehingga dLib dapat mendeteksi wajah dan titik tersebut ditandai dengan koordinat x dan y [7].

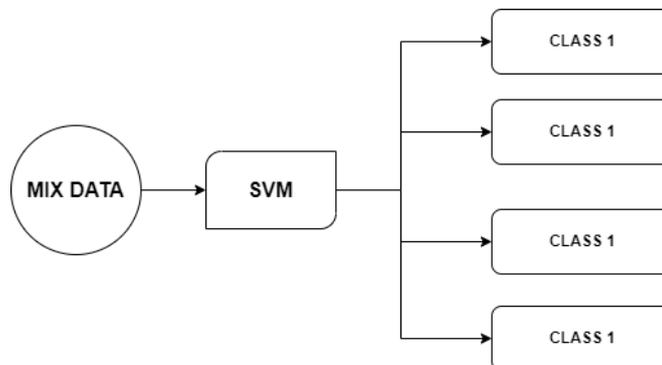


Gambar 2 Facial Landmark

#### 2.4 Histogram Of Orienteds Gradient

Histogram Of Oriented Gradients (HOG) adalah sebuah metode yang berguna untuk mencari karakteristik dari objek gambar dengan menggunakan objek manusia. Dalam proses awal metode HOG mengkonversi citra RGB (Red,Green,Blue) menjadi grayscale dan kemudian dilanjutkan dengan menghitung nilai gradien setiap piksel. Setelah nilai gradien maka proses selanjutnya yaitu menentukan jumlah bin orientasi yang akan digunakan dalam pembuatan histogram [8] [9].

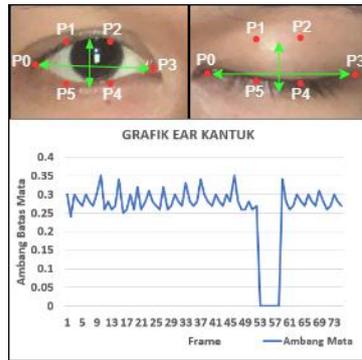
#### 2.5 Support Vector Machine



Gambar 3 Support Vector Machine

Algoritma Support Vector Machine (SVM) yang digunakan sebagai deteksi wajah dan mendeteksi titik tengara serta titik landmark yang terdeteksi. Setelah itu dikombinasikan ke dalam kategori seperti mata, hidung, mulut, alis, telinga, dagu, pipi, dan bibir. Proses klasifikasi ini digunakan secara baik dalam gambar yang tidak bergerak maupun penangkapan video secara langsung (real time) [10] [11] seperti yang terlihat pada gambar 3.

2.6 Eye Aspect Ratio

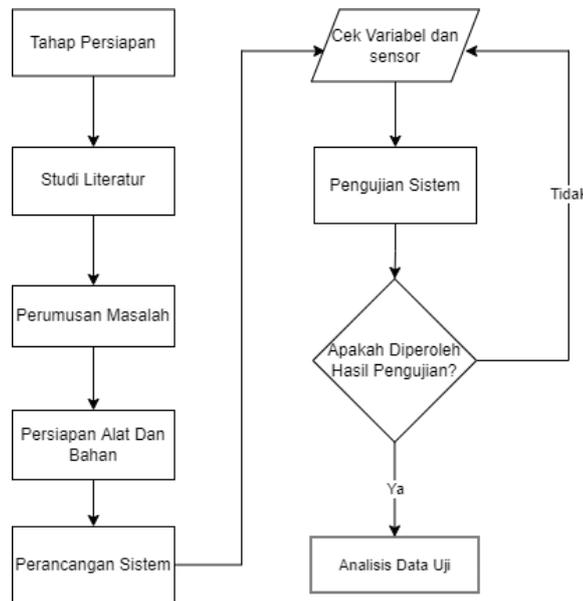


Gambar 4 Eye Aspect Ratio

Terdapat 6 landmark yang mendekati dari paket dlib dibutuhkan supaya terlokalisasi pada tiap mata kanan atau mata kiri. Terdapat enam titik landmark untuk mata yakni P0, P1, P2, P3, P4, P5,. Pertama, menghitung vertical jarak Euclidean untuk pasangan titik (P1, P5) & (P2, P4) [12] [13]. Berikutnya menghitung jarak horizontal Euclides untuk titik (P0, P3). Seperti yang terlihat pada gambar 4, spike yang muncul menunjukkan bahwa nilai EAR akan turun jika kondisi pengemudi dalam keadaan kantuk.

$$EAR = \frac{(|P1-P5|+|P2-P4|)}{2 \times |P0-P3|} \tag{1}$$

2. METODE



Gambar 5. Diagram Blok Penelitian

Secara umum desain penelitian tugas akhir ini memiliki diagram blok penelitian. Gambar 5 terdapat tahap awal atau tahap persiapan, studi literatur, perumusan masalah, persiapan alat dan bahan, perancangan sistem, cek variabel dan sensor, pengujian sistem, dan dilakukan analisis data uji.

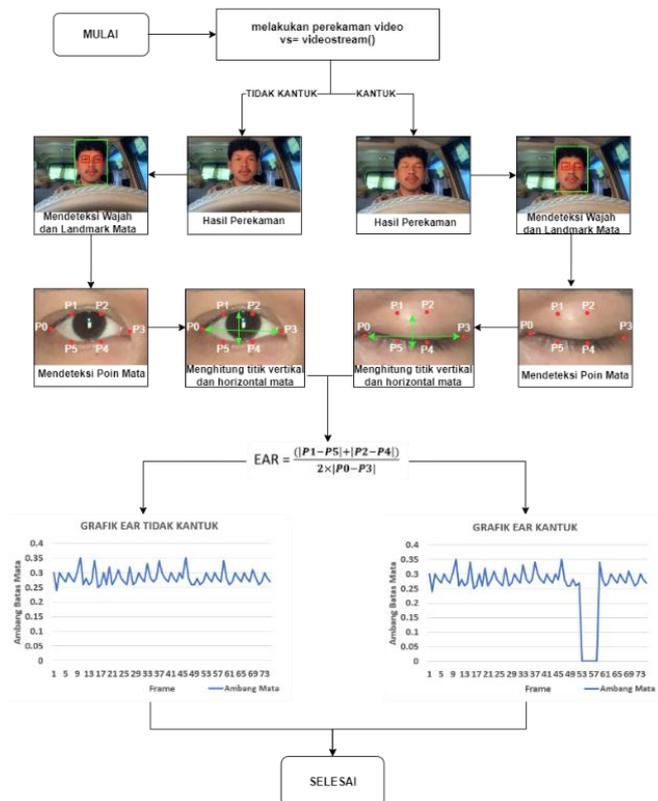
Proses perancangan aplikasi dimulai dengan memasukkan library deteksi wajah dan prediksi landmark pada wajah, selanjutnya menginput EYE\_AR\_THRESHOLD, EYE\_AR\_FRAME, dan Counter. Kemudian masuk dalam tahap pengambilan gambar. Dimana gambar yang semula RGB diubah menjadi grayscale kemudian dilakukan proses ekstraksi fitur HOG dan mencari titik landmark wajah dan mata pada pengemudi, setelah itu masuk ke sub proses def eye\_aspect\_ratio(eye) dimana ini adalah fungsi def dengan Sistem Peringatan Dini Kantuk Pada Pengemudi Malam Hari Menggunakan Metode Facial Landmark Detection Berbasis Raspberry Pi 3 Modul B (Pranata, et. al.)

parameter `eye_aspect_ratio(eye)` yang dijadikan sebagai perhitungan jarak vertical dan horizontal pada mata. Kemudian masuk ke proses menghitung rata-rata ambang batas mata dari kedua mata dan hasilnya jika `ear` lebih dari `EYE_AR_THRESHOLD` maka LED hijau akan menyala dan jika kurang maka counter bertambah nilai +1 dan jika counter lebih besar sama dengan `EYE_AR_FRAME` maka tulisan peringatan akan muncul kemudian buzzer dan LED merah akan menyala.

Proses perancangan alat pendeteksi kantuk terdapat Modul Pi Camera Night Vision yang disambungkan dengan Raspberry Pi 3 Modul B. Kemudian pin pada Raspberry Pi 3 Modul B dihubungkan ke output yang berupa buzzer dan LED. Dimana pada pin 12 (GPIO23) Dihubungkan ke resistor 560ohm kemudian di sambungkan ke LED merah dan buzzer. Keluaran dari LED merah dan buzzer di hubungkan ke ground. Kemudian pada pin 16 (GPIO23) dihubungkan ke resistor 560ohm setelah itu disambungkan ke LED hijau. Keluaran dari LED hijau di hubungkan ke ground.

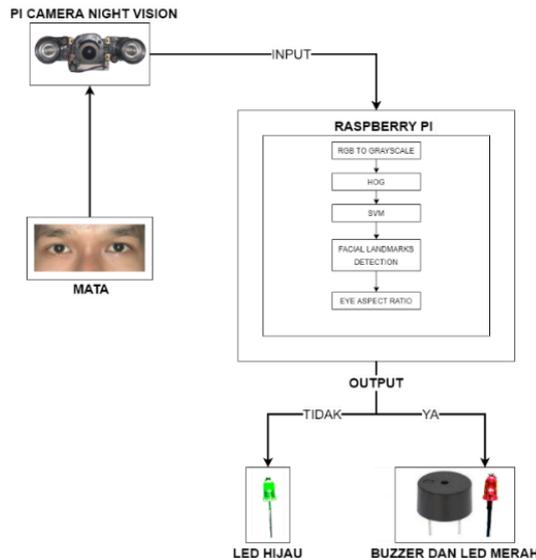
Terdapat cara kerja dari algoritma Histogram Of Oriented Gradients (HOG) dan Support Vector Machine (SVM) yang berguna untuk mendeteksi objek atau gambar. Tahap pertama masukkan gambar setelah itu gambar yang berupa citra RGB (Red, Green, Blue) di ubah menjadi gambar citra grayscale. Maka akan didapatkan nilai gradien dari setiap piksel tersebut. Setelah mendapatkan hasil gradien dari setiap piksel, dilanjutkan ke tahap pemerataan histogram dimana tahap ini dilakukan penyebaran nilai tingkat keabuan citra dibuat menjadi rata. Kemudian dilanjutkan ke tahap ekstraksi fitur HOG dimana setiap gambar disamakan diubah ukurannya menjadi 100 x 100 piksel untuk mengurangi biaya komputasi. Kemudian HOG di ekstraksi untuk setiap gambar menggunakan ukuran sel 8 x 8 piksel dan ukuran blok 2 x 2. Proses tersebut menjadi akhir dalam algoritma HOG, kemudian menentukan landmark mata pada wajah menggunakan facial landmark dengan dataset dari IBUG 300 -w.

Selanjutnya dataset tersebut dilatih menggunakan SVM, kemudian menghitung label dan data dengan hyperplane dengan cara memisahkan dua buah kelas, kelas wajah dan bukan wajah menggunakan kernel linear SVM. Selanjutnya ketahap melatih model SVM dengan cara pelatihan dan pengujian diulang agar memastikan kekokohan metode SVM. Kemudian didapatkan hasil klasifikasi yang selanjutnya dapat digunakan untuk mendeteksi kantuk pada pengemudi melalui mata pada wajah manusia.



Gambar 6. Proses Eye Aspect Ratio

Terdapat bagaimana proses dari Eye Aspect Ratio dimana terdapat 68 titik landmark pada wajah seperti yang terlihat pada gambar 6. Landmark mata pada wajah terdapat 12 titik diantaranya 6 titik mata kiri dan 6 titik mata kanan yang masing – masing 2 titik horizontal dan 4 vertikal. Dalam proses pertama dilakukan perekaman video setelah itu terdapat 2 kategori dimana dalam posisi kantuk atau tidak kantuk. Selanjutnya, hasil perekaman tersebut mendeteksi wajah dan landmark mata pada manusia. Setelah itu landmark tersebut mendeteksi jumlah point yang berada di mata dan kemudian menghitung jarak vertical dan horizontal dengan rumus yang dimasukkan ke dalam program pendeteksi kantuk. Kemudian terdapat gambaran hasil grafik mata kantuk dan tidak dimana dengan melihat grafik, ketika ambang batas mata turun maka dikategorikan mengantuk.

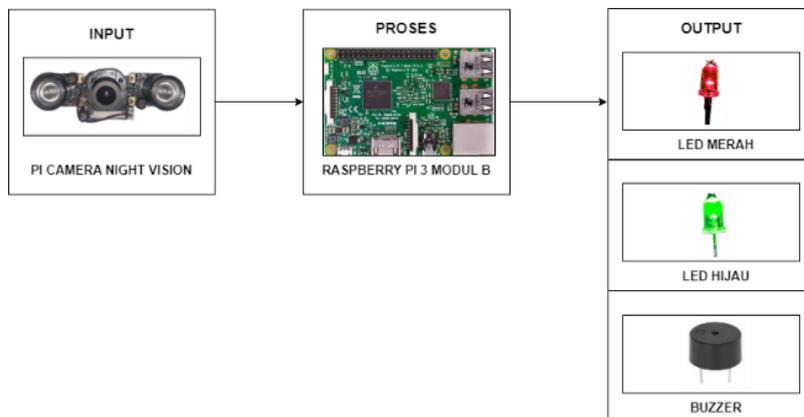


Gambar 7. Deskripsi Sistem

Seperti yang terlihat pada gambar 7, terdapat *Pi Camera Night Vision* yang berguna untuk mengambil video pada pengemudi kemudian diolah oleh program dari raspberry. Video tersebut diubah dari gambar RGB menjadi grayscale setelah itu mendeteksi wajah menggunakan Pustaka dari *Histogram Of Oriented Gradients (HOG) + Linear SVM (Support Vector Machine)* dan prediksi kondisi mata menggunakan *Facial Landmark* dengan dataset dari *iBUG 300 –W* kemudian menghitung nilai *Eye Aspect Ratio* pada kedua mata dan mendapatkan output berupa LED merah, LED hijau, buzzer.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi dan perangkat keras



Gambar 10 Blok Diagram Perangkat Keras

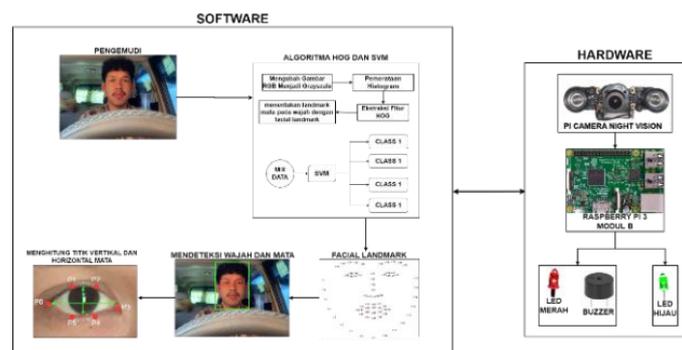
Implementasi perangkat keras merupakan suatu proses pemasangan atau perakitan alat yang digunakan dalam membangun sistem pendeteksi kantuk. Dimana input yaitu Pi Camera Night Vision yang berguna sebagai sensor untuk pengambilan data mata pada wajah manusia, untuk proses yaitu Raspberry Pi 3 modul B yang berguna untuk mengolah data citra pada pi camera night vision kemudian memproyeksikan hasil ke monitor apakah pengemudi tersebut kantuk atau tidak kantuk dan output berupa LED merah, LED hijau, dan Buzzer, dimana hasil dari proses yang dikirim ke output ketika dinyatakan kantuk maka LED merah dan Buzzer akan hidup dan jika tidak kantuk maka LED hijau akan menyala. Terdapat pembuatan perangkat keras, dimana pembuatannya menggunakan triplek untuk menjadi kotak pada raspberry pi, kipas, dan pi camera night vision. Kemudian untuk pemasangan LED merah, LED hijau, dan Buzzer menggunakan project board.

### 3.2 Implementasi Perangkat Lunak



Gambar 11. Implementasi Perangkat Lunak

Gambar 11 merupakan hasil dari tampilan sistem pendeteksi kantuk dimana isi dari tampilan tersebut mendeteksi wajah dan mata pengemudi dan terdapat peringatan pada layar program ketika pengemudi mengalami kantuk. Selain itu output LED merah dan buzzer akan aktif juga. Setelah itu tahap selanjutnya adalah pengujian system. Hasil dari perancangan sistem alat pendeteksi kantuk keseluruhan dimana terdapat kotak komponen yang didalamnya terdapat Raspberry pi, kipas pendingin, dan pi camera night vision lalu pada breadboard terdapat LED dan buzzer.



Gambar 12. Tampilan desain arsitektur sistem

Gambar 12 menampilkan desain arsitektur sistem yang dibuat. Terdapat tampilan bagian software dan hardware dalam mendeteksi kantuk. Dimana bagian software melakukan proses penangkapan gambar pada wajah manusia kemudian dilakukan normalisasi menggunakan kombinasi hog + svm sebagai metode untuk mendeteksi objek. Kemudian tahap facial landmark dengan dataset dari ibug 300 -w berguna untuk menentukan titik koordinat pada wajah manusia dan setelah itu menghitung nilai eye aspect ratio dari kedua mata. Untuk bagian hardware terdapat pi camera night vision berguna untuk mendeteksi objek, kemudian raspberry berguna untuk mengolah data citra dan hasil dari pendeteksian jika kantuk maka led merah dan buzzer akan menyala dan jika tidak kantuk maka led hijau akan menyala.

### 3.3 Pengujian Sistem

#### 1. Pengujian Deteksi Jarak dan Sudut Kemiringan Wajah Pagi Hari

**Tabel 1.** Pengujian Deteksi Jarak dan Sudut Kemiringan Wajah Pagi Hari

Jarak	Percobaan	Sudut	Wajah	Mata	Kantuk	Hasil Program	Jeda Kantuk	Respon	Output
40 Cm	1	0°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.07 Detik		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.19 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.03 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	1.97 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	7.73 Detik		Aktif
	6	75°	Terdeteksi	Terdeteksi	Kantuk	Tidak	-		Tidak
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
50 Cm	1	0°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	1.91 Detik		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	1.96 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.02 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.19 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	5.10 Detik		Aktif
	6	75°	Terdeteksi	Terdeteksi	Kantuk	Tidak	-		Tidak
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
60 Cm	1	0°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.02 Detik		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.07 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.10 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.22 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	5.19 Detik		Aktif
	6	75°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	9.96 Detik		Aktif
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak

Tabel 1 menunjukkan hasil pengujian deteksi jarak dan sudut kemiringan wajah pagi hari. Didapatkan kesimpulan bahwa pada jarak 40 cm dengan sudut 0°- 60° berhasil mendeteksi wajah, mata dan kantuk dan hasil output berupa LED merah dan buzzer aktif. Tetapi pada sudut 75° mengalami pendeteksian yang lambat karena pendeteksian mata yang kurang akurat. Untuk sudut 75° hanya mendeteksi bagian wajah dan mata, untuk kantuk tidak terdeteksi oleh program dikarenakan sulit mendeteksi mata dan sudut 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala. Pada jarak 50 cm dengan sudut 0°- 60° berhasil mendeteksi wajah, mata dan kantuk dan hasil output berupa LED merah dan buzzer aktif. Tetapi pada sudut 60° mengalami pendeteksian yang lambat dikarenakan pendeteksian mata kantuk kurang akurat. Untuk sudut 75° hanya mendeteksi bagian wajah dan mata, untuk kantuk tidak terdeteksi oleh program dikarenakan sulit mendeteksi mata dan sudut 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala.

Pada jarak 60 cm dengan sudut 0°- 60° berhasil mendeteksi wajah, mata dan kantuk dan hasil output berupa LED merah dan buzzer aktif. Tetapi pada sudut 60° mengalami pendeteksian yang lambat dikarenakan pendeteksian mata kantuk kurang akurat. Untuk sudut 75° mendeteksi bagian wajah, mata, dan kantuk, tetapi terdeteksi tetapi mengalami pendeteksian yang lambat dikarenakan pendeteksian mata kantuk kurang akurat dan sudut 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala.

## 2. Pengujian Deteksi Jarak dan Sudut Kemiringan Wajah malam Hari

**Tabel 2.** Pengujian Deteksi Jarak dan Sudut Kemiringan Wajah malam Hari

Jarak	Percobaan	Sudut	Wajah	Mata	Kantuk	Hasil Program	Jeda Kantuk	Respon	Output
40 Cm	1	0°	Terdeteksi	Terdeteksi	Tidak	Terdeteksi	-		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.62 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	6.76 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	9.21 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	11.31 Detik		Aktif
	6	75°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
50 Cm	1	0°	Terdeteksi	Terdeteksi	Tidak	Terdeteksi	-		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.90 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.60 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.81 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	12.01 Detik		Aktif
	6	75°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
60 Cm	1	0°	Terdeteksi	Terdeteksi	Tidak	Terdeteksi	-		Aktif
	2	15°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	3.42 Detik		Aktif
	3	30°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	3.36 Detik		Aktif
	4	45°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	2.40 Detik		Aktif
	5	60°	Terdeteksi	Terdeteksi	Kantuk	Terdeteksi	7.04 Detik		Aktif
	6	75°	Tidak	Tidak	Kantuk	Tidak	-		Tidak
	7	90°	Tidak	Tidak	Kantuk	Tidak	-		Tidak

Tabel 2 menunjukkan hasil pengujian deteksi jarak dan sudut kemiringan wajah malam hari. Didapatkan kesimpulan bahwa pada jarak 40 cm dengan sudut 0° mengalami pendeteksian yang salah dikarenakan pada saat tidak kantuk atau mata terbuka alat pendeteksi mendeteksi kantuk. Hal ini dikarenakan alat sulit mendeteksi ambang batas mata, video streaming berwarna hitam putih, bola mata dan kulit mata berwarna sama pada pendeteksian malam hari. Pada sudut 15° berhasil mendeteksi wajah, mata, kantuk, FPS normal dan hasil output berupa LED merah dan buzzer aktif. Untuk sudut 30° - 60° mengalami pendeteksian yang lambat karena pendeteksian mata yang kurang akurat dan faktor terlalu dekat dengan kamera. Untuk sudut 75° dan 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala.

Pada jarak 50 cm dengan sudut 0° mengalami pendeteksian yang salah dikarenakan pada saat tidak kantuk atau mata terbuka alat pendeteksi mendeteksi kantuk. Hal ini dikarenakan alat sulit mendeteksi ambang batas mata, video streaming berwarna hitam putih, dan bola mata kulit mata berwarna sama pada pendeteksian malam hari. Pada sudut 15°-45° berhasil mendeteksi wajah, mata, kantuk, FPS normal dan hasil output berupa LED merah dan buzzer aktif. Untuk sudut 60° mengalami pendeteksian yang lambat karena pendeteksian mata yang kurang akurat sehingga kamera sulit mendeteksi. Untuk sudut 75° dan 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala.

Pada jarak 60 cm dengan sudut 0° mengalami pendeteksian yang salah dikarenakan pada saat tidak kantuk atau mata terbuka alat pendeteksi mendeteksi kantuk. Hal ini dikarenakan alat sulit mendeteksi

ambang batas mata, video streaming berwarna hitam putih, bola mata dan kulit mata berwarna sama pada pendeteksian malam hari. Pada sudut 15°-45° berhasil mendeteksi wajah, mata, kantuk, FPS normal dan hasil output berupa LED merah dan buzzer aktif. Untuk sudut 60° mengalami pendeteksian yang lambat karena pendeteksian mata yang kurang akurat sehingga kamera sulit mendeteksi. Untuk sudut 75° dan 90° tidak mendeteksi bagian wajah, mata dan kantuk tersebut sehingga LED hijau tetap menyala.

### 3. Pengujian Sistem Alat Pendeteksi Kantuk

**Tabel 3.** Data Diri Pengemudi

Pengemudi	Nama	Umur
1.	I Komang Yoga Tri Pranata	22
2.	I Ketut Sabdayasa	61
3.	I Wayan Joni Indra Arnatha	33

**Tabel 4.** Hasil Pengujian Sistem Alat Pendeteksi Kantuk

Pengemudi	peercobaan	Waktu Terdeteksi	FPS	Jarak Alat	Hasil Program	Output	Keterangan
1.	1.	3:37	15.1	40 cm	Terdeteksi	Aktif	Kantuk
	2.	7:04	19.5	40 cm	Terdeteksi	Aktif	Tidak Kantuk
	3.	9:51	22.8	40 cm	Terdeteksi	Aktif	Kantuk
	4.	13:14	21.8	40 cm	Terdeteksi	Aktif	Kantuk
	5.	18:52	15.2	40 cm	Terdeteksi	Aktif	Kantuk
2.	1.	3:48	20.2	50 cm	Terdeteksi	Aktif	Kantuk
	2.	7:34	20.0	50 cm	Terdeteksi	Aktif	Kantuk
	3.	9:04	21.2	50 cm	Terdeteksi	Aktif	Kantuk
	4.	15:35	20.4	50 cm	Terdeteksi	Aktif	Kantuk
	5.	18:59	19.0	50 cm	Terdeteksi	Aktif	Kantuk
3.	1.	2:33	21.3	60 cm	Terdeteksi	Aktif	Tidak Kantuk
	2.	4:28	21.8	60 cm	Terdeteksi	Aktif	Kantuk
	3.	6:20	21.6	60 cm	Terdeteksi	Aktif	Tidak Kantuk
	4.	9:31	20.6	60 cm	Terdeteksi	Aktif	Kantuk
	5.	14:59	20.6	60 cm	Terdeteksi	Aktif	Kantuk

Tabel 3 dan 4 secara berturut-turut merupakan data diri obyek(orang) yang diuji dan hasil pengujian secara langsung dilapangan. Proses pengujian ini dilakukan dijalan tembolak lingkaran selatan, NTB. Dimana pengemudi 1 melakukan pengujian pada tanggal 22 Oktober 2022 jam 23.18 WITA, pengemudi 2 melakukan pengujian pada tanggal 23 Oktober 2022 jam 21.51 WITA, dan pengemudi 3 melakukan pengujian pada tanggal 24 Oktober 2022 jam 23.18 WITA. Hasil pengujian pendeteksi kantuk yang dilakukan menggunakan kendaraan roda empat secara real-time.

Dari hasil pengujian tersebut didapatkan hasil bahwa Pada subjek 1 dengan jarak 40 cm dan sampel sebanyak 5 mendapatkan jumlah keterangan kantuk sebanyak 4 kantuk dan 1 tidak kantuk, hal ini terjadi karena pada perekaman wajah terlihat pengemudi tidak mengalami kantuk sedangkan alat pendeteksi mengatakan bahwa pengemudi tersebut mengantuk. Hal ini disebabkan terjadinya delay pada program dimana FPS rendah hanya sebesar 15-23 FPS dan saat pendeteksian pada mata, mata tertutup dan setelah itu mata tidak terdeteksi sehingga dikatakan kantuk oleh program. Hasil akurasi didapatkan sebesar :

$$\begin{aligned} \text{Akurasi mata terdeteksi kantuk pada jarak 40 cm} &= \frac{\text{jumlah keterangan kantuk}}{\text{jumlah sampel}} \times 100\% \\ &= \frac{4}{5} \times 100\% = 80\% \end{aligned}$$

Pada subjek 2 dengan jarak 50 cm dan sampel sebanyak 5 mendapatkan jumlah keterangan kantuk sebanyak 5 kantuk dan 0 tidak kantuk. Hal ini terjadi karena jarak berpengaruh terhadap pendeteksian kantuk dan posisi kepala dapat mempengaruhi pendeteksian kantuk. Hasil akurasi didapatkan sebesar:

$$\begin{aligned} \text{Akurasi mata terdeteksi kantuk pada jarak 50 cm} &= \frac{\text{jumlah keterangan kantuk}}{\text{jumlah sampel}} \times 100\% \\ &= \frac{5}{5} \times 100\% = 100\% \end{aligned}$$

Pada subjek 3 dengan jarak 50 cm dan sampel sebanyak 5 mendapatkan jumlah keterangan kantuk sebanyak 3 kantuk dan 2 tidak kantuk Hal ini disebabkan terjadinya delay pada program dimana FPS rendah hanya sebesar 20-22 FPS, untuk jarak juga berpengaruh terhadap pendeteksian kantuk dan lebar mata juga berpengaruh pada pendeteksian kantuk.

$$\begin{aligned} \text{Akurasi mata terdeteksi kantuk pada jarak 60 cm} &= \frac{\text{jumlah keterangan kantuk}}{\text{jumlah sampel}} \times 100\% \\ &= \frac{3}{5} \times 100\% = 60\% \end{aligned}$$

Hasil keseluruhan pendeteksian kantuk dengan menggunakan 3 subjek dengan masing-masing 5 sampel kantuk didapatkan akurasi sebesar :

$$\begin{aligned} \text{Akurasi mata terdeteksi keseluruhan} &= \frac{\text{jumlah keterangan kantuk}}{\text{jumlah sampel}} \times 100\% \\ &= \frac{12}{15} \times 100\% = 80\% \end{aligned}$$

## 6. KESIMPULAN

Dalam merancang sistem alat pendeteksi kantuk, untuk tahap mendeteksi area mata pada wajah menggunakan library dlib facial landmark dari iBUG 300-3 dan untuk mendeteksi objek atau wajah menggunakan kombinasi dari metode Histogram Of Oriented Gradients (HOG) dan Support Vector Machine (SVM). Setelah itu dilanjutkan ke tahap mencari nilai Eye Aspect Ratio (EAR) pada mata, dimana dilakukan perhitungan lebar dan tinggi mata pada mata kanan dan kiri sebagai acuan apakah pengemudi tersebut mengalami kantuk atau tidak dengan cara jika mata tertutup 0,3 detik dengan frame sebesar 5 atau lebih maka mata tersebut dikatakan mengantuk. Setelah melakukan perancangan sistem alat pendeteksi kantuk, selanjutnya dilakukan pengujian alat kepada pengemudi secara real-time pada malam hari yang berguna untuk mengetahui kinerja alat yang telah dibuat. Dilakukan pengujian menggunakan 3 subjek masing – masing 5 sampel kantuk dengan jarak 40, 50, 60cm. Pada subjek 1 dengan jarak 40 cm didapatkan akurasi sebesar 80%, subjek 2 dengan jarak 50 cm didapatkan akurasi sebesar 100%, subjek 3 dengan jarak 60 cm didapatkan akurasi sebesar 60 %, subjek keseluruhan didapatkan akurasi sebesar 80% dan error sebesar 20%. Sehingga dari subjek tersebut dapat disimpulkan hasil dari pengujian sistem alat pendeteksi kantuk mendapatkan hasil sesuai yang diharapkan

## DAFTAR PUSTAKA

- [1] TribunLombok.com, 2021. 2021, Tingkat Kecelakaan Lalu Lintas di Mataram Naik, 64 Orang Korban Meninggal Dunia. <https://lombok.tribunnews.com/2021/12/30/2021-tingkat-kecelakaan-lalu-lintas-di-mataram-naik-64-orang-korban-meninggal-dunia>
- [2] D. C. J. M. Pardede, A.M. Rumagit, S. Tangkawarouw, & G. Kaunang. (2022). Deteksi Pengendara Mengantuk Menggunakan Metode Eye Tracking Berbasis Raspberry Pi. <http://repo.unsrat.ac.id/3581/1/JurnalDeanClain-16021103052-JTEK.pdf>
- [3] R.T. Puteri, & F. Utamingrum, (2020). Deteksi Kantuk Menggunakan Kombinasi Haar Cascade dan Convolutional Neural Network. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 4(3), 816–821.
- [4] M. Rabbani, D. Wahiddin, & R. Rahmat, (2021). Haarcascade Classifier Dan Eye Aspect Ratio Untuk Mengidentifikasi Mata Kantuk Pada Pengemudi Mobil. *Conference On Innovation And Application Of Science And Technology (Ciastech)*, 0, 437-444. Retrieved from <http://publishing-widyagama.ac.id/ejournal-v2/index.php/ciastech/article/view/3339>
- [5] R. P. H. Sejati, & R. Mardhiyyah, (2021). Deteksi Wajah Berbasis Facial Landmark Menggunakan OpenCV Dan Dlib. *Jurnal Teknologi Informasi*, 5(2), 144–148. <https://doi.org/10.36294/jurti.v5i2.2220>

- [6] A. Rosebrock. (2017). Facial landmarks with dlib, OpenCV, and Python. <https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>
- [7] N. Wahyudiana and S. Budi, “Perbandingan Performa Pre-Trained Classifier dLib dan HAAR Cascade ( OpenCV ) Untuk Mendeteksi Wajah,” vol. 1, no. November, pp. 374–385, 2019.
- [8] D. Navneet dan B. Triggs. (2005). “Histograms of Oriented Gradients for Human Detections”. Proc. IEEE Conf.on Computer Vision & Pattern Recognition. 886- 893.
- [9] K. Mittal, (2020). A Gentle Introduction Into The Histogram Of Oriented Gradients. <https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa>
- [10] A. Jindal, & R. Priya, (2019). Landmark Points Detection in Case of Human Facial Tracking and Detection. International Journal of Engineering and Advanced Technology, 9(2), 3769–3776. <https://doi.org/10.35940/ijeat.b3367.129219>
- [11] W. Widayani, & H. Harliana, (2021). Analisis Support Vector Machine Untuk Pemberian Rekomendasi Penundaan Biaya Kuliah Mahasiswa. Jurnal Sains Dan Informatika, 7(1), 20–27. <https://doi.org/10.34128/jsi.v7i1.268>
- [12] D. Pandey. (2021). Eye Aspect Ratio(EAR) and Drowsiness detector using dlib. <https://medium.com/analytics-vidhya/eye-aspect-ratio-ear-and-drowsiness-detector-using-dlib-a0b2c292d706>
- [13] 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, [Online]. Available: <https://ejournal.itn.ac.id/index.php/jati/article/download/541/507>