

RANCANG BANGUN MONITORING DETEKSI BANJIR BERBASIS INTERNET OF THINGS MELALUI KENDALI APLIKASI INSTANT MESSENGER (STUDI KASUS: SUNGAI MAS)

Diko Alan Firmansyah¹, Billy Montolalu, S.Kom., M.Kom.² Hendy Briantoro, S.ST., M.T., Ph.D.³

^{1,2,3} Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom, Surabaya

¹dikoalanfirmansyah@student.telkomuniversity.ac.id, ²billymontolalu@.telkomuniversity.ac.id,

³hendybriantoro@telkomuniversity.ac.id

Abstrak

Pengendara transportasi mengalami kecelakaan yang diakibatkan mengantuk, dengan membangun deteksi kantuk pengendara. Deteksi kantuk pengendara dibangun menggunakan *Raspberry Pi 3B+* dengan metode *Facial Landmark* untuk mendeteksi wajah (mata dan mulut). Untuk menghitung mata tertutup menggunakan *Eye Aspect Ratio* (EAR) dan *Mouth Aspect Ratio* (MAR) untuk mulut menguap. Sebelum melakukan pengujian alat, dilakukan iterasi untuk mencari nilai threshold 0.1 hingga 0.3 mata tertutup dan threshold 0.6 hingga 1.0 mulut menguap dengan 1000+ gambar orang mengantuk. Nilai threshold yang didapat dari iterasi mata mendapatkan threshold 0.1 dengan tingkat akurasi 96.01% dan mulut mendapatkan threshold 1.0 dengan akurasi 100%. Pengujian dilakukan dalam kendaraan dengan threshold dan mendapatkan hasil dengan akurasi 100% dari 12 gambar.

Kata kunci : Deteksi kantuk, *Facial Landmark*, *Eye Aspect Ratio*, *Mouth Aspect Ratio*, threshold

Abstract

A transportation driver experienced an accident caused by drowsiness, leading to the development of a drowsiness detection system for drivers. This system was built using a Raspberry Pi 3B+ with the Facial Landmark method to detect facial features (eyes and mouth). To calculate eye closure, the Eye Aspect Ratio (EAR) was used, while the Mouth Aspect Ratio (MAR) was applied for yawning detection. Before testing the device, an iteration was conducted to find the appropriate threshold values, ranging from 0.1 to 0.3 for eye closure and from 0.6 to 1.0 for yawning, using over 1000 images of drowsy individuals. The iteration process resulted in an eye closure threshold of 0.1 with an accuracy rate of 96.01% and a yawning threshold of 1.0 with an accuracy rate of 100%. Testing was conducted in a vehicle using these thresholds, yielding a result of 100% accuracy from 12 images.

Keywords: Detection Drowsiness, *Facial Landmark*, *Eye Aspect Ratio*, *Mouth Aspect Ratio*, threshold

1. Pendahuluan

Transportasi memegang peranan penting dalam kehidupan masyarakat, khususnya transportasi darat roda empat yang mempermudah perpindahan antar tempat. Namun, transportasi ini juga berpotensi menimbulkan masalah seperti kecelakaan lalu lintas yang dapat berdampak serius pada berbagai aspek kehidupan[1]. Untuk mengatasi masalah ini, pemasangan aplikasi pendeteksi wajah mengantuk pada kendaraan roda empat dapat membantu meminimalisir kecelakaan. Mengenali emosi pengemudi dan mengambil tindakan yang tepat juga penting untuk mengurangi risiko kecelakaan akibat tekanan dan stres, sehingga pengemudi dapat menstabilkan emosinya sebelum melanjutkan perjalanan[2].

Penelitian sebelumnya “Prototype Deteksi Level Kantuk Berdasarkan Eye Aspect Ratio Menggunakan Metode Facial Landmark Berbasis Orange Pi” penulis Ridwan Muhammad, Am.Andri Nofiar, Pribadi Antoni, Sahfitri Lestari. Penelitian ini menggunakan *facial landmark* sebagai pendeteksi wajah dan *eye aspect ratio* merupakan parameter untuk mengukur mata pengguna dalam kondisi terbuka atau tertutup. Hasil dari penelitian ini berupa sudut kamera lebih besar dari 60°, jarak kamera lebih dari 25 cm dan kurang dari 163 cm. Berhasil membuat deteksi level kantuk dengan menggunakan pengukuran nilai *eye aspect ratio* yaitu ambang batas berupa 0,25[3].

Facial Landmark merupakan metode lokalisasi tempat tertentu pada wajah seseorang yang menunjukkan fitur-fitur tertentu. Sudut mata, ujung hidung, sudut lubang hidung, sudut mulut, titik akhir busur alis, telinga, lobus telinga, dan dagu adalah landmark wajah yang paling sering ditemukan. *Facial Landmark* akan mengidentifikasi 68 titik pada wajah untuk menentukan bentuk biologis dari wajah. Dengan mengamati pergerakan titik landmark pada wajah saat seseorang mengekspresikan diri, dapat menganalisis emosi wajah. Penempatan titik landmark pada wajah seseorang serta jarak dan sudut antara titik-titik tersebut dapat digunakan untuk mengenali wajah seseorang,

seperti area sekitar mata bermanfaat karena setiap individu memiliki lokasi titik tertentu. Pergerakan titik landmark pada wajah dapat dipantau dalam frame video[4][5].

OpenCV adalah library pengolahan citra dalam pemrograman Python yang dapat digunakan untuk menampilkan gambar atau video, mengubah warna RGB ke grayscale, dan melakukan banyak hal lainnya. Library Dlib menganalisis bagian wajah dengan mengekstrak nilai gambar, dan menghasilkan vektor fitur 128 dimensi untuk pengolahan gambar wajah menggunakan metode *facial landmark*[6].

Kamera digunakan untuk memantau dan menganalisis kondisi wajah pengemudi secara *real time*. Dengan teknologi pemrosesan citra, kamera dapat mendeteksi tanda-tanda kantuk seperti mata tertutup. Data yang dihasilkan berupa gambar dan video pengemudi yang kemudian diproses dengan *raspberry pi*[7].

2. Dasar Teori

2.1 Facial Landmark

Facial Landmarks Detection merupakan contoh output yang berperan dalam memprediksi struktur geometri yang terkait dengan data citra wajah. *Facial Landmarks* merujuk pada sekumpulan titik penting yang terdapat dalam citra wajah manusia. Facial Landmark digunakan untuk memberikan 68 titik koordinat pada wajah pengemudi. Titik koordinat bertujuan untuk menentukan perubahan pada bagian wajah seperti mata, mulut, dan alis [8].

2.2 Eye Aspect Ratio (EAR)

Metode EAR ini umumnya digunakan untuk menghitung frekuensi mata tertutup. Pengukuran EAR ini dilakukan dengan memperhitungkan mata kiri dan mata kanan dengan *landmarks* wajah. Dalam rumus EAR, pembilang menghitung jarak antara landmark mata secara vertikal, sementara penyebut menghitung jarak antara landmark mata secara horizontal. Penyebut diberi bobot yang sesuai karena hanya terdapat satu set titik horizontal tetapi dua set titik vertikal [9].

2.3 Mouth Aspect Ratio (MAR)

MAR adalah perbandingan antara jarak vertikal mulut dan jarak horizontal mulut. Dalam estimasi MAR, tiga jarak vertikal dan satu jarak horizontal diperhitungkan. Ketika pengemudi menguap, jarak vertikal mulut akan meningkat sedangkan jarak horizontal mulut akan sedikit berkurang. Untuk deteksi yang akurat, nilai MAR untuk mulut menguap diasumsikan lebih besar dari 0,43. Sementara itu, nilai-nilai MAR lainnya yang kurang dari ambang batas tersebut akan diabaikan [10].

2.4 Raspberry Pi 3 Model B+

Raspberry Pi adalah *microprosesor* yang efisien dengan prosesor ARM Quad-core 1,4 GHz 64 Bit. Ini berbasis Linux dan mendukung berbagai sistem operasi seperti Raspbian, Ubuntu Mate, Ubuntu Core, dan lainnya. Raspberry Pi dapat diprogram menggunakan Python, BASIC, C, C++, Java, Perl, dan Ruby. *Raspberry Pi 3B+* berperan sebagai pembuatan program deteksi kantuk. Program yang dibuat dengan *library* yang ada dalam *raspberry pi* berupa *dlib*, *python*, dan *opencv*. *Raspberry Pi* dapat bekerja secara *real time* untuk deteksi kantuk dengan sumber daya rendah [11].

2.5 Kamera USB

Kamera USB adalah tipe kamera yang mudah digunakan dan seringkali kompatibel dengan berbagai platform komputasi. Kamera USB dapat dengan mudah di hubungkan ke perangkat seperti komputer, Raspberry Pi menggunakan port USB. Kamera digunakan untuk memproses data citra wajah dalam deteksi kantuk. Kamera dapat digunakan secara *real time* untuk memantau wajah pengemudi dalam mengidentifikasi ekspresi wajah, pergerakan mata, dan gerakan kepala [12].

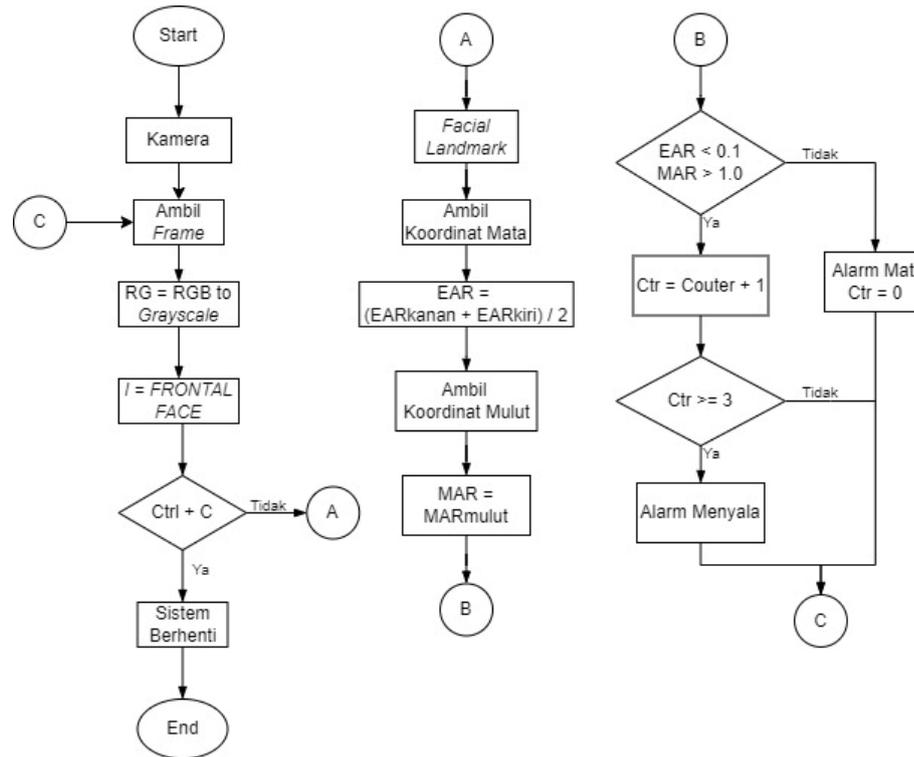
2.6 Drowsiness

Drowsiness merupakan kondisi dimana pengemudi mengantuk. Pengemudi mengantuk pada siang hari karena kurangnya waktu tidur, meskipun terlihat sederhana kantuk dapat memicu timbulnya sebuah masalah. Kantuk menjadi masalah serius dalam konteks keselamatan, terutama di jalan raya atau dalam pekerjaan yang memerlukan perhatian penuh. Pengemudi yang mengantuk berisiko lebih tinggi mengalami kecelakaan karena penurunan refleks dan perhatian. Setelah kantuk terdeteksi, sistem ini biasanya memberikan peringatan atau saran kepada individu untuk beristirahat atau menepi terlebih dahulu [13].

3. Perancangan Sistem

3.1 Desain Sistem

Perancangan kerja sistem pada penelitian Rancang Bangun Deteksi Kantuk Pengemudi Menggunakan Metode Facial Landmark untuk mendeteksi mata tertutup dan mulut menguap. Pada gambar 3.1 dapat dilihat sebagai berikut:



Gambar 3. 1 Flowchart Sistem

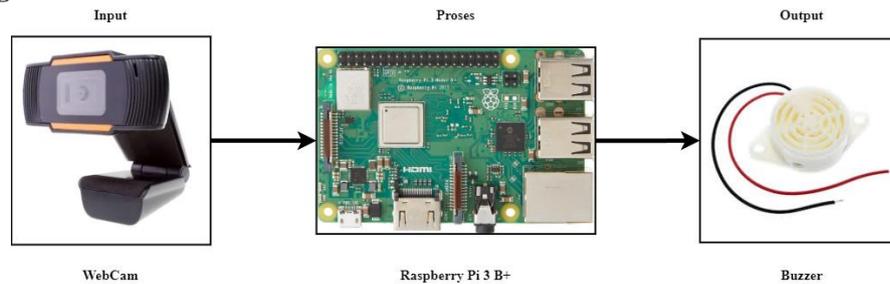
Sistem dimulai dengan menyalnya kamera secara *real-time*. Kamera mengambil gambar *frame* yang akan dijadikan gambar *grayscale*. Gambar *grayscale* diproses dengan *Frontal Face* untuk memperkecil komputasi untuk focus ke wajah pengendara. Setelah sistem diolah akan digunakan *facial landmark* untuk memberikan titik-titik koordinat pada wajah. Titik koordinat yang digunakan untuk mendeteksi mata dan mulut pengendara. Nilai koordinat pada mata akan diolah menggunakan rumus *eye aspect ratio* untuk mentukan mata tertutup dan nilai koordinat mulut diolah menggunakan rumus *mouth aspect ratio* untuk menentukan mulut menguap.

Apabila nilai EAR kurang dari *threshold* 0.1 pada sistem memiliki nilai *counter* lebih besar atau sama dengan 3, maka *alarm buzzer* akan berbunyi selama nilai EAR kurang dari *threshold* 0.1. Ketika nilai EAR lebih besar dari *threshold* 0.1, maka *alarm buzzer* akan mati dan *counter* akan direset. Jika nilai MAR lebih dari *threshold* 1.0 pada sistem memiliki nilai *counter* lebih besar atau sama dengan 3, maka *alarm buzzer* akan berbunyi selama nilai *threshold* MAR lebih dari 1.0. Ketika *threshold* MAR kurang dari 1.0, maka *alarm buzzer* akan mati dan *counter* akan direset. Sistem akan berhenti berjalan dengan memasukkan keyword pada terminal berupa karakter “Ctrl+C” dari pengemudi. Untuk spesifikasi alat-alat yang digunakan di penelitian ini akan ditunjukkan didalam table 3.1 dibawah ini.

Tabel 3. 1 Spesifikasi Alat

No	Perangkat	Spesifikasi
1	Raspberry Pi 3 Modul B+	Broadcom BCM2837B0, quad-core Cortex-A53, 1.4 GHz 1 GB LPDDR2 Slot kartu microSD, 5V DC mikro-USB Ethernet 10/100 Mbps, Wi-Fi 802.11b/g/n, Bluetooth 4.2, 4 port USB 2.0 1 port HDMI (1080p), 40 pin GPIO 1 port kamera, 1 port tampilan
2	Kamera	1280 x 720, 110° wide angle Adjustable Focus Lens, USB connectivity Automatic Brightness Adjustment AWB : Automatic White Balance Frame Rate: 30 fps Lens: Manual Focus Lens
3	Buzzer	tegangan 5V atau 3.3V
4	Power Bank	Battery 10Ah

3.2 Perancangan Sistem



Gambar 3.2 Perancangan Sistem

Penjelasan dari perancangan alat yang akan digunakan dalam deteksi kantuk. Kamera sebagai *input* data yang menangkap wajah untuk dijadikan citra gambar dalam *frame*. Kemudian gambar *frame* akan ditampilkan di *raspberry pi* dan akan diproses dengan sistem yang sudah dibuat. Sistem yang terdapat dalam *raspberry pi* berupa EAR untuk menghitung ukuran mata tertutup dengan *threshold* dan MAR untuk menghitung ukuran mulut terbuka dengan *threshold*. Kamera akan terus *looping* gambar *frame* untuk mendapatkan nilai mata tertutup dan mulut terbuka dari *threshold* yang sudah ditentukan. *Output* berupa *alarm buzzer* yang akan berbunyi jika terdapat *frame* mata tertutup dengan nilai dibawah *threshold* dan mulut menguap dengan nilai diatas *threshold*.

3.3 Menentukan Threshold

Untuk menentukan nilai *threshold* menggunakan dataset orang mengantuk dengan jumlah 1000+ gambar. Iterasi dilakukan dengan menggunakan nilai *threshold* mulai dari 0.1 hingga 0.3 untuk mata dan 0.6 hingga 1.0 untuk mulut. Cara ini untuk mendapatkan nilai *threshold* terbaik dari mata dan mulut untuk mendeteksi kantuk penengdara.

3.4 Menghitung EAR

Penjelasan rumus *eye aspect ratio* (EAR) untuk menghitung rasio antara jarak vertikal dan horizontal mata, sebagai berikut:

$$EAR = \frac{|p2-p6|+|p3-p5|}{2|p1-p4|} \quad 3.1$$

- P1 = Titik sudut luar kiri mata
- P2 = Titik tepi atas mata bagian luar
- P3 = Titik tepi atas mata bagian dalam
- P4 = Titik sudut luar kanan mata
- P5 = Titik bawah mata bagian dalam
- P6 = Titik tepi bawah mata bagian luar

Untuk menghitung *eye aspect ratio* (EAR), pertama hitung dua jarak vertikal pada mata antara titik p2 dan p6, serta antara p3 dan p5. Kemudian, jumlahkan kedua jarak vertikal. Selanjutnya bagi hasil penjumlahan tersebut dengan dua kali jarak horizontal antara titik p1 dan p4. EAR ini memberikan rasio antara tinggi dan lebar mata, di mana nilai yang lebih tinggi menunjukkan mata yang lebih terbuka, dan nilai yang lebih rendah menunjukkan mata yang lebih tertutup.

3.5 Menghitung MAR

Rumus *mouth aspect ratio* (MAR) untuk mengukur keadaan mulut terutama untuk mendeteksi mulut menguap, sebagai berikut:

$$MAR = \frac{|p2-p8|+|p3-p7|+|p4-p6|}{3|p1-p5|} \quad 3.2$$

- P1 = Titik sudut kiri mulut bagian luar
- P2 = Titik tepi atas mulut
- P3 = Titik tepi atas mulut, di dalam p2
- P4 = Titik tepi atas mulut, di dalam p3
- P5 = Titik sudut kanan mulut bagian luar
- P6 = Titik tepi bawah mulut, di dalam p7
- P7 = Titik tepi bawah mulut, di dalam p8
- P8 = Titik tepi bawah mulut bagian luar

Mouth Aspect Ratio (MAR) dihitung dengan mengukur tiga jarak vertikal di mulut: antara p2-p8, p3-p7, dan p4-p6, yang masing-masing mewakili tinggi mulut di bagian luar, tengah, dan dalam. Jumlahkan ketiga jarak ini, lalu bagi dengan dua kali jarak horizontal antara p1 dan p5 (lebar mulut). MAR menunjukkan rasio antara tinggi dan lebar mulut, yang digunakan untuk menentukan seberapa terbuka atau tertutupnya mulut

4. Hasil Pengujian dan Analisis

4.1 Hasil Alat

Selama tahap awal percobaan, tujuan utama adalah pengembangan prototipe alat yang selaras dengan skema perancangan alat pada bab 3. Pembuatan prototipe alat mengikuti spesifikasi yang telah ditetapkan. Hasil dari prototipe tersebut dibuktikan pada gambar 4.1.

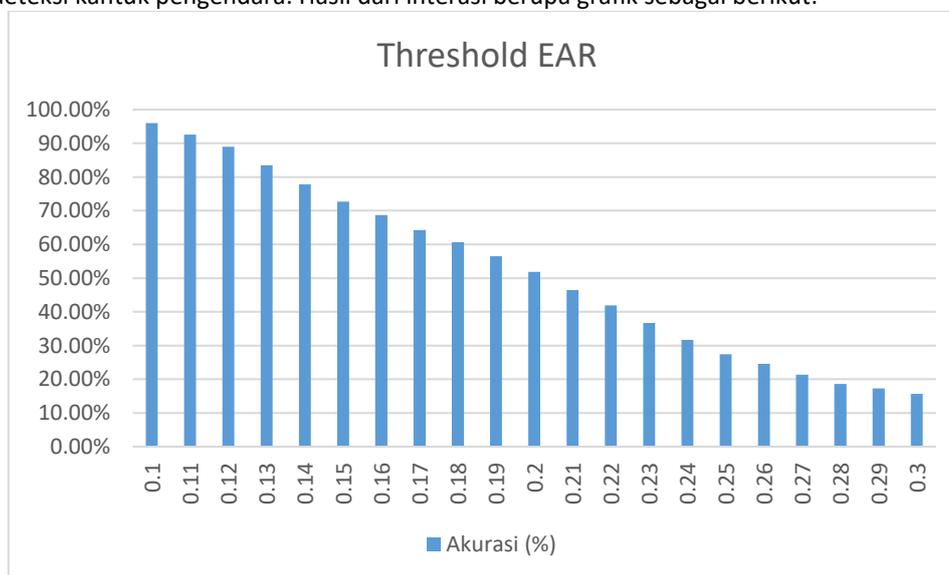


Gambar 4. 1 Prototipe Alat

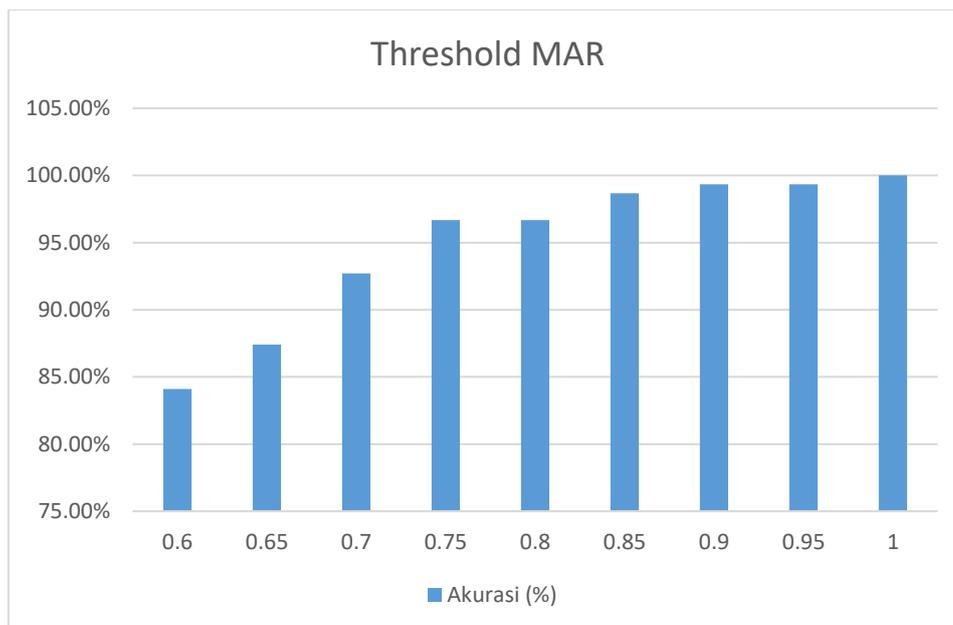
Berdasarkan gambar 4.1 maka dapat dilihat bahwa prototipe yang digunakan sesuai dengan skema alat yang telah dijelaskan pada bab 3. Terdapat sebuah raspberry pi yang telah terhubung dengan lcd untuk layer os raspberry pi, gpio untuk penghubung dengan buzzer, webcam sebagai pemantau wajah pengendara, dan power bank tenaga raspberry pi. Kamera terletak pada setir mobil menghadap wajah pengendara, supaya kamera dapat menangkap wajah meliputi mata dan mulut untuk deteksi kantuk.

4.2 Hasil Threshold

Hasil yang didapatkan dari threshold menggunakan dataset gambar dengan interaksi mata 0.1 hingga 0.3 dan mulut 0.6 hingga 1.0. Melakukan interaksi untuk mendapatkan nilai threshold terbaik yang akan digunakan dalam deteksi kantuk pengendara. Hasil dari interaksi berupa grafik sebagai berikut:



Gambar 4. 1 Grafik Threshold EAR



Gambar 4.2 Grafik Threshold MAR

Hasil grafik EAR pada gambar 4.2 dengan threshold 0.1 mendapat nilai akurasi tertinggi dengan 96.01% dan threshold 0.3 mendapat nilai akurasi terkecil dengan 15.72%. Pada grafik MAR pada gambar 4.3 mendapat nilai akurasi tertinggi sebanyak 100% dengan threshold 1.0 dan threshold 0.6 mendapat nilai akurasi terkecil dengan 84.16%.

Dengan hasil grafik dari gambar 4.2 dan 4.3 dapat diterapkan pada deteksi kantuk pengendara. Threshold akan ditetapkan sebagai abang batas pada EAR dengan threshold 0.1 untuk mengetahui mata tertutup. Pada abang batas MAR dengan threshold 1.0 untuk mulut menguap.

4.3 Hasil Pengujian Sistem

Hasil percobaan alat dengan threshold yang sudah ditentukan sebelumnya untuk mendeteksi kantuk pengendara. Hasil percobaan alat dapat dilihat pada gambar 4.4.



Gambar 4. 4 Hasil Pengujian

Rumus menghitung akurasi dari alat:

$$\text{Akurasi} = \frac{\text{jumlah benar}}{\text{total percobaan}} \times 100\%, \text{ Sehingga} \quad 4.1$$

$$\text{Akurasi} = \frac{12}{12} \times 100\% = 100\% \quad 4.2$$

Pada gambar 4.4 terlihat hasil pengujian sistem yang dilakukan dalam kendaraan dengan beberapa *frame*. EAR threshold 0.1 dapat mendeteksi mata tertutup dengan hasil 6 *frame* gambar dan MAR threshold 1.0 berhasil mendeteksi mulut menguap dengan hasil 6 *frame* gambar. Dari 12 *frame* gambar mendapat nilai akurasi 100% deteksi kantuk pengendara dengan benar.

5. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, sebuah sistem deteksi kantuk dengan menggunakan Raspberry Pi dapat dibuat dengan tujuan mengurangi pengendara mengantuk. Sistem dibuat menggunakan metode *Facial Landmark* dibantu dengan *library dlib*, dan *opencv*. Pada saat program dijalankan metode *facial landmark* berjalan dengan baik pada *raspberru pi*. Sistem saat mendeteksi pengendara mengantuk terdapat delay. Melakukan iterasi threshold dengan 1000+ gambar untuk menentukan nilai threshold EAR dan MAR terbaik sebelum melakukan pengujian. Threshold 0.1 mendapat nilai akurasi 96.01% untuk mendeteksi mata tertutup, threshold 1.0 mendapat nilai akurasi 100% untuk mendeteksi mulut menguap. Hasil pengujian dengan threshold yang sudah didapatkan dengan 12 gambar mendapat nilai akurasi 100% dalam mendeteksi kantuk pengendara.

Daftar Pustaka:

- [1] I. G. A. G. , S. G. I. K. , & T. P. U. K. A. Suryadarmawan, “Tingkat Kecelakaan dan Lokasi Daerah Rawan Kecelakaan pada Jalan Nasional di Kabupaten Karangasem,” 2022.
- [2] N. Cintisa, E. Suhartono, and S. Aulia, “Pengenalan Ekspresi Pada Raut Wajah Pada Keselamatan Berkendara Menggunakan Principal Component Analysis (PCA) dan Linier Discriminant Analysis (LDA) Facial Emotion Recognition For Safety Driving Using Principal Component Analysis and Linier Discriminant Analysis,” 2019.
- [3] M. Ridwan et al., “PROTOTYPE DETEKSI LEVEL KANTUK BERDASARKAN EYE ASPECT RATIO MENGGUNAKAN METODE FACIAL LANDMARK BERBASIS ORANGE Pi,” 2024. [Online]. Available: <http://ojsamik.amikmitragama.ac.id>
- [4] J. P. Teknologi, D. Pembelajaran, F. Dumi, H. Fikriyah, A. Nugroho, and A. F. Suni, “REKAYASA Eye Fatigue Detection in Vehicle Drivers Based on Facial Landmarks Features,” 2021, doi: 10.15294/rekayasa.v19i1.28699.
- [5] I. Putu et al., “Detection of Student Drowsiness Using Ensemble Regression Trees in Online Learning During a COVID-19 Pandemic Deteksi Kantuk Peserta Didik Menggunakan Ensemble Regression Trees Pada Pembelajaran Daring Dimasa Pandemi COVID-19,” *Jurnal Informatika dan Teknologi Informasi*, vol. 19, no. 2, pp. 229–244, 2022, doi: 10.31515/telematika.v19i2.7044.
- [6] A. Dicky Purwanto and F. Utamingrum, “Deteksi Pergerakan Mata dan Kedipan Untuk Memilih Empat Menu Display Menggunakan Probabilitas Berdasarkan Facial Landmark,” 2019. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] A. Asvin Mahersatillah Suradi, S. Alam, M. Furqan Rasyid, I. Djafar, U. Dipa Makassar, and J. K. Perintis Kemerdekaan, “Sistem Deteksi Kantuk Pengemudi Mobil Berdasarkan Analisis Rasio Mata Menggunakan Computer Vision,” 2023.
- [8] A. H. A. P. Perdana, “Implementasi Sistem Deteksi Mata Kantuk Berdasarkan Facial Landmarks Detection Menggunakan Metode Regression Tres,” 2019.
- [9] C. K. U Nggiku and A. Rabi, “Deteksi Kantuk Pada Pengemudi Mobil Menggunakan Eye Aspect Ratio Dengan Metode Facial Landmark,” *Seminar Nasional Fortei Regional*, vol. 7, 2022, [Online]. Available: <https://binaryupdates.com/>
- [10] L. Thulasimani, P. P, and P. S. P, “Real Time Driver Drowsiness Detection Using Opencv And Facial Landmarks,” 2021.
- [11] Y. K. Dubey and S. Damke, “Baby monitoring system using image processing and IoT,” *Int J Eng Adv Technol*, vol. 8, no. 6, pp. 4961–4964, Aug. 2019, doi: 10.35940/ijeat.F9254.088619.
- [12] E. Aviolita, A. Soetedjo, and K. SomawirataI, “Pendeteksi Penyakit Tanaman Buah Naga Menggunakan Kamera Berbasis Raspberry Pi,” 2019.
- [13] M. A. Maulana, “Deteksi Kantuk Pada Pengendara Roda Empat Melalui Citra Wajah Menggunakan Metode Facial Landmark,” 2022.

