

Rancang Bangun Deteksi Dini Kantuk Berbasis Eye Aspect Ratio dan API menggunakan Python

Nelly Sofi¹, Tri Sulistyorini², M. Ryan Rifqi Firdaus³
Informatika, Universitas Gunadarma

Article History

Received : 2026-03-07

Revised : 2026-04-23

Accepted : 2026-05-29

Published : 2026-06-01

Corresponding author*:

tri_s2@staff.gunadarma.ac.id

Cite This Article: Sofi, N., Tri Sulistyorini, & M. Ryan Rifqi Firdaus. (2026). Rancang Bangun Deteksi Dini Kantuk Berbasis Eye Aspect Ratio dan API menggunakan Python. *Jurnal Teknik Dan Science*, 5(2), 20–28.

DOI:

<https://doi.org/10.56127/jts.v5i2.2752>

Abstract: Drowsiness during work can reduce productivity and increase the risk of errors, particularly in monotonous office jobs that rely heavily on screen-based activities. This study aims to design and develop a modular computer vision-based drowsiness detection system capable of detecting drowsiness conditions in real-time and providing automatic notifications to users. The system detects drowsiness conditions using the Eye Aspect Ratio (EAR) method through an internal laptop or PC camera. The implementation was carried out using the Python programming language with the OpenCV library for image processing and Dlib for facial landmark detection. Users are provided with a graphical user interface (GUI) application developed using Tkinter. The system automatically sends warning messages through the Telegram Bot API when the EAR value is detected below the threshold of 0.21 for more than 3 seconds. Notifications can be delivered automatically to both the user's and supervisor's Telegram accounts. The testing results indicate that the system is capable of operating in real-time with fast response times and stable notification delivery through Telegram. Warning messages are delivered concisely so as not to disrupt the user's workflow. Based on these findings, the developed drowsiness detection system has the potential to be utilized as a supporting tool to improve alertness and prevent productivity decline among office workers.

Keywords: Drowsiness Detection, Eye Aspect Ratio (EAR), Telegram Bot API, Modular System, Python, Office Productivity, Computer Vision.

PENDAHULUAN

Di era kerja modern, tuntutan waktu kerja yang panjang berdampak langsung pada pola tidur pekerja. Waktu kerja yang berlebihan terbukti berkorelasi dengan durasi tidur yang lebih singkat, yang pada akhirnya menurunkan kinerja kerja. Studi oleh Swanson et al. (2011) menunjukkan bahwa individu yang bekerja lebih dari 60 jam per minggu mengalami pengurangan durasi tidur baik pada hari kerja maupun hari libur. Kualitas tidur yang buruk ini memicu gangguan kognitif di tempat kerja, seperti kesulitan berkonsentrasi, kesulitan mengatur pekerjaan, serta peningkatan gejala kelelahan dan penurunan produktivitas. Dampak kantuk di lingkungan kerja cukup signifikan. Data menunjukkan bahwa 29% pekerja mengalami kantuk ekstrem atau bahkan tertidur saat bekerja dalam sebulan terakhir. Kondisi ini tidak hanya mengganggu hubungan interpersonal antar rekan kerja, tetapi juga meningkatkan risiko terjadinya kesalahan kerja. National Sleep Foundation juga mencatat bahwa lebih dari 60% pekerja mengalami gangguan tidur akibat beban kerja tinggi dan penggunaan perangkat digital yang berlebihan. Kantuk tidak hanya menurunkan efisiensi, tetapi juga mengancam keselamatan kerja, terutama pada pekerjaan yang menuntut ketelitian tinggi (Barnes et al., 2016). Mengatasi permasalahan tersebut,

berbagai metode deteksi kantuk berbasis teknologi telah dikembangkan. Salah satu metode yang banyak digunakan adalah Eye Aspect Ratio (EAR), yang mengukur perubahan rasio aspek mata sebagai indikator kondisi mengantuk (Soukupova & Cech, 2016). Penelitian Suryowinoto et al. (2022) menetapkan bahwa kedipan dengan durasi lebih dari 3 detik dapat dikategorikan sebagai kondisi mengantuk, khususnya pada konteks pengemudi kendaraan. Namun, sebagian besar penelitian serupa masih terbatas pada bidang transportasi atau lingkungan laboratorium, sehingga belum sepenuhnya disesuaikan dengan kebutuhan dan karakteristik lingkungan kerja perkantoran. Akurasi deteksi kantuk juga dipengaruhi oleh nilai ambang batas EAR yang digunakan. Penelitian Dewi et al. (2022) menunjukkan bahwa nilai threshold EAR sebesar 0,2 memberikan akurasi dan nilai AUC terbaik dalam mendeteksi kedipan, terutama pada pengguna dengan karakteristik mata kecil atau pengguna kacamata. Hal ini mengindikasikan bahwa penyesuaian nilai ambang sesuai karakteristik pengguna sangat penting agar sistem dapat bekerja lebih akurat dan adaptif.

Integrasi sistem deteksi kantuk ke platform yang mudah diakses seperti API masih jarang diaplikasikan. Padahal, integrasi ini diperlukan agar sistem dapat diterapkan secara fleksibel di lingkungan kantor.

Oleh karena itu, penelitian ini merancang sistem deteksi kantuk modular berbasis EAR yang terintegrasi dengan Telegram Bot API, diharapkan sistem ini mampu mendeteksi kantuk secara real-time dan mengirimkan notifikasi otomatis untuk meningkatkan kewaspadaan serta produktivitas pekerja kantoran.

METODOLOGI PENELITIAN

Penelitian ini menggunakan metode Research and Development dengan pendekatan eksperimental untuk merancang, mengimplementasikan dan menguji sistem deteksi kantuk berbasis visi computer, dengan tahapan sebagai berikut :

1. Analisis Kebutuhan : melakukan studi literatur, menentukan threshold EAR 0,21, kebutuhan hardware dan softwarena serta integrasi Telegram API.
2. Desain Sistem : rancang arsitektur modular seperti modul deteksi wajah, modul perhitungan EAR dan modul notifikasi.
3. Implementasi : pemrograman menggunakan Phyton, OpenCV, Dlib, Tkinter, integrasi ke Telegram Bot API
4. Pengujian : yang dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan spesifikasi yang telah ditentukan. Pengujian dilakukan dengan menguji fungsionalitas API dan output yang dihasilkan terhadap input yang diberikan, serta memverifikasi logika perhitungan EAR dan penentuan status kantuk dalam modul deteksi.

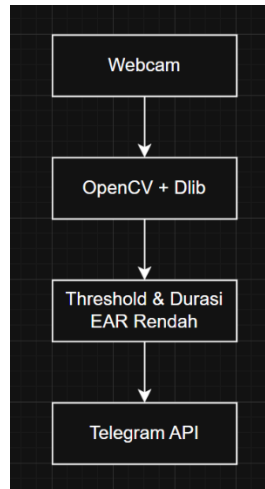
HASIL DAN PEMBAHASAN

Penelitian ini merancang dan membangun sebuah sistem deteksi kantuk berbasis real-time ini dirancang untuk mengidentifikasi kondisi kantuk pada pengguna, khususnya pekerja kantoran, dengan memanfaatkan analisis citra wajah melalui kamera laptop. Deteksi dilakukan dengan memantau pergerakan mata, yang kemudian dihitung dalam bentuk rasio yang dikenal sebagai *Eye Aspect Ratio* (EAR). Ketika nilai EAR berada di bawah ambang batas tertentu dalam durasi waktu tertentu, sistem akan mengindikasikan bahwa pengguna sedang mengantuk.

Desain Sistem

Perancangan sistem dilakukan secara modular untuk memisahkan setiap fungsi utama ke dalam komponen yang terstruktur

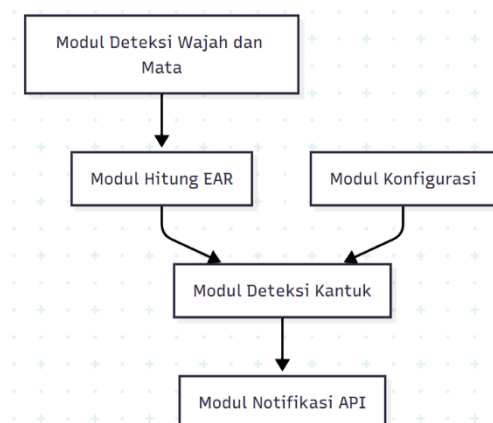
- Arsitektur Sistem



Gambar 1. Arsitektur Sistem

Arsitektur sistem pada Gambar 1 menunjukkan alur kerja sistem deteksi kantuk berbasis pengolahan citra real-time. Sistem menangkap citra wajah melalui kamera yang diposisikan ± 40 cm dari pengguna. Citra kemudian diproses menggunakan Dlib untuk mendeteksi wajah dan mengekstraksi titik landmark mata. Selanjutnya, sistem menghitung Eye Aspect Ratio (EAR) berdasarkan jarak vertikal dan horizontal landmark mata. Jika nilai EAR berada di bawah ambang batas 0,21 selama lebih dari 3 detik, sistem menyimpulkan pengguna dalam kondisi mengantuk. Saat kondisi kantuk terdeteksi, sistem mengirim notifikasi peringatan melalui Telegram Bot API agar pengguna segera beristirahat. Seluruh proses ditampilkan pada GUI yang menampilkan video real-time, status deteksi, dan nilai EAR untuk memudahkan pemantauan. Integrasi komponen yang terstruktur, sistem ini mampu mendeteksi kantuk secara responsif dan real-time untuk mendukung produktivitas kerja.

- Desain Modul



Gambar 2. Desain Modul

1. Deteksi Wajah dan Mata

Menggunakan **Dlib** untuk mendeteksi posisi wajah dan 68 titik landmark. Titik yang digunakan untuk perhitungan EAR adalah 6 titik pada area mata.

2. Perhitungan EAR

Modul ini menerima input berupa koordinat landmark mata dari modul sebelumnya dan menghitung **Eye Aspect Ratio (EAR)** dengan rumus:

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

Gambar 3. Rumus EAR

3. Logika Deteksi Kantuk

Jika nilai EAR < **0.2** selama lebih dari **3 detik**, pengguna dianggap mengantuk.

- **Threshold EAR** yaitu 0.21 sebagai batas bawah.
- **Durasi minimum** yaitu 3 detik untuk memastikan kondisi kantuk bukan disebabkan oleh kedipan biasa.

5. Pengiriman Notifikasi

Ketika kondisi kantuk terdeteksi, modul ini akan mengirimkan peringatan kepada pengguna melalui **Telegram Bot API**. Fungsi utama dari modul ini:

- Mengkonstruksi pesan peringatan.
- Mengirim pesan ke akun Telegram yang sudah dikonfigurasi.
- Mendukung ekspansi ke platform lain (misalnya WhatsApp, Email, dll).

Implementasi

Implementasi sistem dilakukan sesuai rancangan yang telah dibuat. Pengembangan menggunakan bahasa Python dengan menerapkan konsep pemrograman berorientasi objek (OOP) agar sistem lebih modular dan mudah dikembangkan.

- **Implementasi Deteksi Wajah dan Mata**

Pada bagian ini dideklarasikan tiga variabel utama. `cap` digunakan untuk mengaktifkan kamera melalui `cv2.VideoCapture()` agar sistem dapat menangkap video real-time. `detector` memanggil fungsi `get_frontal_face_detector()` dari `dlib` untuk mendeteksi wajah pada setiap frame. `predictor` memuat model `shape_predictor_68_face_landmarks.dat` guna menentukan 68 titik landmark wajah, seperti mata dan mulut, yang dipakai untuk analisis deteksi kantuk.

- **Implementasi Logika Deteksi Kantuk**

Deteksi mata menggunakan metode Eye Aspect Ratio (EAR) untuk mendeteksi kondisi mata. Perhitungan EAR diambil dari rasio jarak vertikal dan horizontal antara enam titik landmark mata yang dihasilkan model shape predictor. Secara matematis, jarak vertikal diukur dari pasangan titik (p2,p6) dan (p3,p5), sedangkan jarak horizontal dari pasangan titik (p1,p4). Nilai EAR kemudian digunakan untuk menentukan apakah mata terbuka atau tertutup.

- **Implementasi Deteksi Kantuk dan Penentuan Waktu kirim ke API**

Kantuk dideteksi berdasarkan ambang batas Eye Aspect Ratio (EAR). Apabila nilai EAR berada di bawah ambang batas, sistem menganggap mata tertutup dan mencatat waktu awalnya pada `sleep_start_time`. Durasi mata tertutup dihitung dari selisih waktu saat ini dengan `sleep_start_time` dan disimpan pada `elapsed`. Bila `elapsed` melebihi batas waktu yang ditentukan, sistem menganggap pengguna mengantuk dan mengirim peringatan melalui Telegram API sekali saja.

`telegram_sent` merupakan status yang diubah menjadi True untuk mencegah pengiriman berulang. Semua variable `sleep_start_time`, `elapsed`, dan `telegram_sent` direset apabila

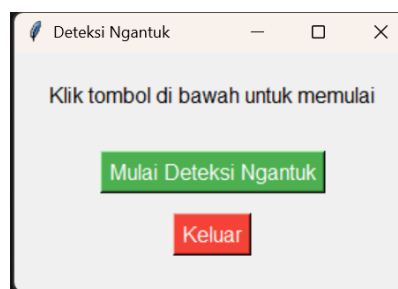
nilai EAR kembali di atas ambang batas. Berdasarkan mekanisme ini, deteksi berjalan real-time dan adaptif sesuai kondisi mata pengguna.

- **Integrasi Telegram Antarmuka Pemrograman Aplikasi**

Pengiriman notifikasi otomatis melalui Telegram API secara real-time dilengkapi dengan fungsi `send_telegram_message(message)`. Fungsi ini menggunakan URL API Telegram dengan token bot dan chat ID sebagai penerima. Pengiriman pesan melalui metode POST menggunakan pustaka `requests`, berisi `chat_id` dan teks peringatan. Apabila berhasil, sistem akan menampilkan “Pesan Telegram berhasil dikirim”. Apabila gagal, akan tampil pesan error beserta penyebabnya. Berdasarkan integrasi ini, hasil deteksi kantuk dapat dipantau jarak jauh lewat Telegram, sehingga pengawasan di lingkungan kerja menjadi lebih efisien dan real-time.

- **Antarmuka Pengguna**

Antarmuka untuk sistem ini berupa grafis berbasis Tkinter agar pengguna dapat mengoperasikannya tanpa menyentuh kode. Fungsi `jalankan_aplikasi` membuat jendela utama berjudul “Deteksi Ngantuk” ukuran 300×180 piksel, pada jendela ini memiliki dua tombol. Tombol “Mulai Deteksi Ngantuk” berwarna hijau untuk menjalankan fungsi deteksi di thread terpisah, sehingga tampilan tetap responsif. Tombol “Keluar” berwarna merah untuk menutup aplikasi dengan aman. Fungsi `mainloop` membuat jendela tetap berjalan dan merespons setiap aksi pengguna. Antarmuka ini membuat pengguna lebih mudah menggunakan sistem ini.



Gambar 4. Tampilan GUI

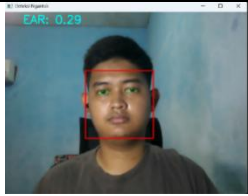



PENGUJIAN

Pengujian dilakukan untuk memastikan sistem berjalan sesuai harapan. Pengujian dilakukan pada beberapa aspek : fungsionalitas sistem, pengujian validasi EAR dan pengujian integrasi API.

- **Pengujian Fungsionalitas Sistem**

Pengujian fungsional dilakukan untuk menilai kemampuan sistem mendeteksi wajah dan mata dalam berbagai kondisi. Subjek diuji pada jarak ± 40 cm (jarak ideal antara wajah dan kamera) dari kamera dengan dua kondisi mata berbeda. Tabel 1 merupakan hasil ujicoba :

Tabel 1. Hasil Ujicoba Fungsionalitas Sistem

Jenis Mata	Kesimpulan
 <p>Mata Normal Tidak Mengantuk</p>	Sistem Mampu Mendeteksi Wajah
 <p>Mata Normal Mengantuk</p>	Sistem Mampu Mendeteksi Wajah
 <p>Mata Sipit Tidak Mengantuk</p>	Sistem Mampu Mendeteksi Wajah
 <p>Mata Sipit Mengantuk</p>	Sistem Mampu Mendeteksi Wajah

Sistem berhasil mendeteksi wajah dan menghitung EAR dengan akurat. Tingkat keberhasilan mencapai 90% dari 40 percobaan, seperti terlihat pada tabel di berikut :

Tabel 2. Hasil Percobaan 40x Running

Jenis Mata	Kondisi	Percobaan	Berhasil	Gagal
Normal	Normal	10	9	1
Normal	Mengantuk	10	9	1
Sipit	Normal	10	8	2

Sipit	Mengantuk	10	10	0
Jumlah	2	40	36	4

Pada tabel di atas merupakan percobaan 40x running dengan dua kondisi (mata normal dan mengantuk) per dua jenis mata (mata normal dan sipit).

$$AVG = \frac{\text{Jumlah Uji Coba Terdeteksi}}{\text{Jumlah Uji Coba}} \times 100 \%$$

$$= \frac{36}{40} \times 100 \% = 0,9 = 90\%$$

Kegagalan deteksi sebagian besar terjadi pada kondisi pencahayaan redup dan pergerakan wajah yang cepat. Hal ini konsisten dengan temuan penelitian sebelumnya bahwa kualitas pencahayaan merupakan faktor krusial dalam sistem *computer vision* berbasis kamera RGB.

Berdasarkan hal tersebut disimpulkan bahwa sistem memiliki keandalan yang cukup tinggi pada jarak ideal (± 40 cm) dengan pencahayaan memadai, sebagaimana ditunjukkan pada ilustrasi hasil pengujian di table 2.

• **Pengujian Validasi EAR**

Validasi EAR dilakukan untuk memastikan bahwa sistem mampu membedakan kondisi mata terbuka dan tertutup dengan akurat. Pengujian dilakukan pada lima kondisi, yaitu mata terbuka lebar, terbuka normal, setengah tertutup, tertutup, serta kondisi tanpa wajah hasil pengujian ditunjukkan pada tabel berikut :

Tabel 3. Kondisi Mata Normal

Kondisi Mata	EAR yang diharapkan	EAR yang dihasilkan
Terbuka Lebar	>0.28	0.31
Terbuka Normal	0.25 – 0.28	0.27
Setengah tertutup	0.20 – 0.24	0.21
Tertutup	<0.20	0.15
Tanpa Wajah	-	-

Tabel 4. Kondisi Mata Sipit

Kondisi Mata	EAR yang diharapkan	EAR yang dihasilkan
Terbuka Lebar	>0.25	0.28
Terbukan Normal	0.23 – 0.25	0.24
Setengah Tertutup	0.21 – 0.23	0.22
Tertutup	<0.20	0.16
Tanpa Wajah	-	-

Hasil pengujian menunjukkan bahwa nilai EAR yang dihasilkan konsisten dengan literatur fisiologi mata manusia. Pada kondisi mata terbuka, nilai EAR berada di atas 0.25, sedangkan pada kondisi tertutup nilainya menurun signifikan hingga <0.20. Hal ini berlaku baik untuk mata normal maupun mata sipit, sebagaimana ditunjukkan pada Tabel 3 dan Tabel 4, sehingga dapat disimpulkan bahwa algoritme EAR yang diterapkan memiliki reliabilitas tinggi sebagai indikator deteksi kantung.

• **Pengujian Integrasi API**

Integrasi dengan API Telegram diuji untuk memverifikasi kemampuan sistem dalam memberikan notifikasi real-time. Beberapa skenario diuji, termasuk kondisi normal (EAR < 0.2 selama > 3 detik), koneksi internet terputus, token bot tidak valid, dan chat ID yang salah.

Tabel 5. Pengujian Integrasi API

Skenario	Input	Output
Kirim notifikasi normal	EAR < 0.2 selama > 3 detik	Pesan peringatan dikirim ke Telegram
Bot offline	Koneksi internet dimatikan	Gagal mengirim pesan, sistem tetap berjalan
Token salah	BOT_TOKEN diubah ke token tidak valid	Pesan Berhasil Terkirim namun tidak muncul pesan pada Bot.
Chat ID salah	CHAT_ID tidak sesuai	Pesan Berhasil Terkirim namun tidak muncul pesan pada bot.

Hasil pengujian menunjukkan bahwa sistem berhasil mengirimkan notifikasi pada kondisi normal, sementara pada kondisi gangguan sistem tetap berjalan tanpa mengalami *crash*. Pesan gagal terkirim apabila token bot atau chat ID tidak sesuai, namun proses deteksi tetap berfungsi dengan baik. Hal ini membuktikan bahwa sistem memiliki **toleransi kesalahan (fault tolerance)** yang memadai, sehingga dapat digunakan pada kondisi operasional dunia nyata yang tidak selalu stabil.

KESIMPULAN

Penelitian ini telah berhasil merancang dan membangun sistem deteksi kantuk berbasis pendekatan Eye Aspect Ratio (EAR) yang diimplementasikan secara modular serta dikemas dalam bentuk Application Programming Interface (API) berbasis Python. Sistem ini dikembangkan sesuai dengan tahapan perancangan yang telah ditetapkan. Berdasarkan hasil uji coba, sistem mampu mendeteksi kondisi kantuk secara real-time dengan menurunnya nilai EAR ketika mata dalam keadaan tertutup. Pengujian juga menunjukkan bahwa sistem dapat merespons perubahan kondisi mata dengan baik dan stabil, serta dapat dijalankan secara efisien tanpa membebani performa perangkat secara signifikan. Selain itu, hasil pengujian dengan data realtime menunjukkan bahwa sistem memiliki tingkat akurasi sebesar 90%, sehingga layak digunakan sebagai prototipe awal dalam pengembangan sistem deteksi kantuk untuk mendukung keselamatan kerja.

DAFTAR PUSTAKA

Albadawi, Y., & Alredhaei, A. (2023). Real-Time Machine Learning-Based Driver Drowsiness Detection Using Visual Features.

Dewanti, N. P., Jingga, N. A., & Wahyudiono, Y. D. A. (2022). The Relationship between Work Shifts and Work Environment with Nurse Fatigue in the Emergency Department. *Indonesian Journal of Occupational Safety and Health*, 11(2), 178–186. <https://doi.org/10.20473/ijosh.v11i2.2022.178-186>

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10, 1755–1758.

- Soukupova, T., & Cech, J. (2016). Real-Time Eye Blink Detection using Facial Landmarks. Center for Machine Perception, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University in Prague, 1–8.
- Swanson, L. M., Arnedt, J. T., Rosekind, M. R., Belenky, G., Balkin, T. J., & Drake, C. (2011). Sleep disorders and work performance: Findings from the 2008 National Sleep Foundation Sleep in America poll. *Journal of Sleep Research*, 20(3), 487–494. <https://doi.org/10.1111/j.1365-2869.2010.00890.x>
- Szeliski, R. (2011). Introduction. 1–25. https://doi.org/10.1007/978-1-84882-935-0_1
- Yuen, S. C. K., Zakaria, N. H. B., Su, G. E., Hassan, R., Kasim, S., & Sutikno, T. (2024). Real-time smart driver sleepiness detection by eye aspect