

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Faktor yang mempengaruhi performansi kerja salah satunya adalah kelelahan kerja. Kelelahan kerja dapat berupa kelelahan fisik dan mental. Cedera akibat kelelahan kerja memiliki risiko yang tinggi dan dapat berujung pada kematian jika dibiarkan [1]. Faktor yang menyebabkan kelelahan kerja dapat terdiri dari banyak hal [2]. Kelelahan kerja memiliki pola dasar yaitu kelelahan secara fisiologis dan kelelahan psikologis [3].

Kelelahan memiliki banyak definisi. Salah satu definisi adalah kurangnya energi secara fisik atau mental, yang tidak dapat dideteksi dengan baik. Kelelahan juga dapat didefinisikan sebagai kantuk [4]. Tingkat kelelahan pekerja di suatu perusahaan perlu diperhatikan agar dapat meminimasi hal negatif yang ditimbulkan karena kelelahan kerja [1]. Solusi umum dari masalah diatas adalah adanya sebuah pengingat di saat pekerja mulai mengantuk.

Banyak penelitian tentang sistem pengenalan kantuk yang memanfaatkan kecerdasan buatan, diantaranya yaitu pengenalan wajah yang merupakan salah satu topik penelitian paling banyak digunakan karena potensi aplikasinya dalam kontrol akses, pengawasan otomatis, keamanan informasi, komunikasi multimedia, antarmuka manusia – mesin, dan lain-lain. Dibandingkan dengan teknologi otentikasi *biometric* lainnya, pengenalan wajah memiliki keunggulan karena tidak membutuhkan kerja sama banyak orang dan terjaga keamanannya. Metode yang banyak digunakan didasarkan pada estimasi gerak di wilayah wajah yang dideteksi dari *file haar cascade* [5].

Sistem deteksi kantuk akan disajikan dalam sebuah aplikasi desktop yang dibangun dengan Tkinter *library*. Python digunakan untuk membangun sistem deteksi kantuk karena Python merupakan bahasa pemrograman yang paling banyak digunakan serta mudah dimengerti, selain itu Python memiliki *library* lengkap didalamnya yang mendukung dalam bidang kecerdasan buatan khususnya mengenai *computer vision*. Dengan adanya sistem deteksi kantuk ini diharapkan dapat memberikan peringatan awal terkait kantuk bagi para pekerja yang bekerja di

depan monitor komputer, sehingga dapat mencegah hal-hal kekeliruan dalam bekerja dan tetap menjaga kesehatan fisik saat bekerja.

Berdasarkan uraian latar belakang tersebut, maka dirancang dan dibangun sebuah aplikasi desktop pendeteksi kantuk dengan kecerdasan buatan yang menggunakan Python. Penelitian ini berjudul “Drowsinesslab: Aplikasi Desktop Deteksi Kantuk Berbasis *Facial Landmark* Menggunakan Dlib, OpenCV dan Tkinter” sebagai objek dalam menyelesaikan tugas akhir.

## 1.2 *State of The Art*

*State of the art* adalah bentuk keaslian karya ilmiah yang dibuat agar tidak ada tindakan plagiat sebagai bentuk pembajakan terhadap karya orang lain. Dalam hal ini, *state of the art* menjelaskan perbandingan terhadap riset yang telah dilakukan sebelumnya dan menjadi acuan pembuatan tugas akhir. Perbandingan tersebut yaitu:

Tabel 1.1 Referensi

Judul	Peneliti	Tahun	Deskripsi
<i>Real-Time Eye Blink Detection using Facial Landmarks</i>	Tereza Soukupova and Jan Čech.	2017	Algoritma <i>real time</i> untuk mendeteksi kedipan mata dalam urutan video dari kamera standar diusulkan. Detektor <i>landmark</i> terbaru, yang dilatih pada set data menunjukkan ketahanan yang sangat baik terhadap orientasi wajah dengan kamera, pencahayaan yang bervariasi, dan ekspresi wajah. Artikel ini menunjukkan bahwa <i>landmark</i> terdeteksi cukup tepat untuk memperkirakan tingkat pembukaan mata secara andal. Algoritma yang diusulkan memperkirakan posisi <i>landmark</i> , mengekstrak kuantitas skalar tunggal <i>Eye Aspect Ratio</i> (EAR) yang mencirikan lubang mata di setiap <i>frame</i> . Klasifikasi SVM mendeteksi kedipan mata sebagai pola nilai EAR dalam <i>window temporal</i> singkat.
<i>Drowsiness Detection of a Driver using Conventional Computer Vision Application</i>	Hitendra Garg	2020	Artikel ini mengusulkan Sistem Deteksi Kantuk <i>Real-Time</i> (RT-DDS) yang dapat diterapkan pada kendaraan bermotor dengan bantuan aplikasi <i>Computer Vision</i> konvensional. Sistem ini menggunakan berbagai aplikasi <i>Computer Vision</i> dengan menggunakan kecepatan kedip, penutup mata, menguap secara efektif dan cepat mengidentifikasi rasa kantuk pengemudi selama mengemudikan kendaraan.

Judul	Peneliti	Tahun	Deskripsi
Deteksi Kantuk Menggunakan Kombinasi <i>Haar Cascade</i> dan <i>Convolutional Neural Network</i>	Rahma Tiara Puteri, Fitri Utaminingrum	2020	Penelitian ini mengembangkan sistem pendeteksi kantuk menggunakan <i>Haar Cascade</i> dan <i>Convolutional Neural Network</i> . Input pada sistem didapat dari <i>Webcam Logitech C310</i> . Keluaran dari sistem yaitu tulisan peringatan mengantuk pada monitor saat pengemudi dalam keadaan mengantuk serta terdapat suara alarm untuk peringatan. Rata-rata akurasi sistem untuk pendeteksian wajah menggunakan <i>Haar Cascade</i> yaitu 100%, rata-rata akurasi untuk pendeteksian mata terbuka dan tertutup pada jarak 30-50 cm yaitu 97.23% dan rata-rata akurasi untuk pendeteksian kantuk sebesar 97.23%. Sistem ini memiliki rata-rata waktu komputasi sebesar 0.2075 s yang akan memudahkan untuk mendeteksi kantuk secara cepat.
<i>A Score based Approach to Detect Drowsiness using OpenCV</i>	Dr. Kranthi Kumar, G. Niharika, D. Yashaswin, M. Durga Sravani, T. Arjun, V. Sai Varun	2020	Sistem deteksi kantuk menggunakan metodologi Python dengan <i>library</i> seperti OpenCV, PyGame, <i>file Haar Cascade</i> untuk mendeteksi kantuk seseorang dimana masukan diambil dari umpan video langsung dan diberikan sebagai masukan untuk model CNN. Salah satu metode untuk mendeteksi rasa kantuk adalah dengan menghitung waktu saat orang tersebut menutup mata dengan menghitung EAR dari setiap <i>frame</i> . Di setiap <i>frame</i> , area mata ditempatkan dan dilatih menggunakan pengklasifikasi <i>Haar cascade</i> .
<i>Using Tkinter of Python to Create Graphical User Interface (GUI) for Scripts in LNLS</i>	D. B. Beniz, A. M. Espindola	2016	Python banyak digunakan untuk membuat skrip mencakup kebutuhan yang berbeda dalam skenario komputasi. Di LNLS ( <i>Brazilian Synchrotron Light Laboratory</i> ) telah berhasil mengembangkan skrip Python untuk mengontrol operasi <i>beamlines</i> , termasuk kasus pembuatan GUI ( <i>Graphical User Interface</i> ) menggunakan Tkinter untuk salah satu <i>beamlines</i> LNLS, DXAS ( <i>Dispersive X-ray Absorption Spectroscopy</i> )

Penelitian mengenai deteksi kantuk dengan kecerdasan buatan telah banyak dilakukan dan banyak publikasi riset yang mengangkat topik ini. Sebagaimana yang tertera dalam Tabel 1.1 dimana terdapat beberapa referensi terkait deteksi kantuk menggunakan kecerdasan buatan. *Detector landmark* telah banyak diaplikasikan dalam penelitian mengenai pengenalan wajah dalam kecerdasan buatan [6]. Artikel [7] mengusulkan algoritma yang memperkirakan posisi landmark, mengekstrak kuantitas skalar tunggal *Eye Aspect Ratio* (EAR) yang mencirikan lubang mata di setiap *frame*, berdasarkan penelitian tersebut nilai EAR yang terdeteksi sebagai kantuk adalah kurang dari 0.3. Berdasarkan artikel [8] rata-rata akurasi sistem untuk

pendeteksian wajah menggunakan *Haar Cascade* yaitu 100%, rata-rata akurasi untuk pendeteksian mata terbuka dan tertutup pada jarak 30-50 cm yaitu 97.23% dan rata-rata akurasi untuk pendeteksian kantuk sebesar 97.23%. Sistem ini memiliki rata-rata waktu komputasi sebesar 0.2075 s yang akan memudahkan untuk mendeteksi kantuk secara cepat, durasi minimal untuk mendeteksi seseorang mengantuk adalah 3 detik selama seseorang menutup mata berdasarkan teori *microsleep*. Arikel [5] mengusung sistem deteksi kantuk menggunakan metodologi Python dengan *library* seperti OpenCV, PyGame, *file Haar Cascade* digunakan untuk mendeteksi kantuk seseorang dimana masukan diambil dari umpan video langsung dan diberikan sebagai masukan untuk model CNN. Python memiliki *library* khusus untuk membangun GUI agar tampilan aplikasi menjadi lebih menarik, seperti pada artikel [9] yang telah membangun sebuah GUI menggunakan *library* Tkinter dari Python.

Berdasarkan Tabel 1.1 di atas sudah ada beberapa penelitian terkait topik deteksi kantuk. Oleh karena itu, pada penelitian tugas akhir ini akan dilakukan penelitian yang berjudul “Drowsinesslab: Aplikasi Desktop Deteksi Kantuk Berbasis *Facial Landmark* Menggunakan Dlib, OpenCV dan Tkinter” dengan memuat teori dan sistem yang sudah ada sebelumnya kemudian dimuat dalam sebuah *software* yang dapat digunakan secara gratis untuk PC dengan OS Windows sehingga dapat digunakan di lingkungan kerja yang membutuhkan pengawasan secara mandiri di meja kerja masing-masing.

Metode deteksi yang digunakan berupa gabungan dari beberapa metode deteksi wajah diantaranya *computer vision*, *haar cascade classifier*, *dlib's facial landmark* dan *Eye Aspect Ratio* (EAR). Nilai referensi EAR yang digunakan adalah 0.3, oleh karena itu jika terdapat nilai EAR kurang dari 0.3 didefinisikan sebagai kantuk. Nilai fps yang dimiliki *webcam* adalah 15 fps sehingga nilai ambang batas *frame* yang digunakan adalah 50 untuk mendapatkan durasi minimal sesuai teori *microsleep* yaitu 3 detik. Input dari aplikasi berupa *real time video* sedangkan output sistem deteksi kantuk ini adalah tulisan peringatan dan bunyi alarm ketika seseorang terdeteksi mengantuk. Aplikasi didesain menggunakan bahasa pemrograman Python yang sudah terinstal beberapa *library* diantaranya OpenCV,

Dlib, Pygame dan Tkinter. Penelitian ini dilakukan untuk *project* tugas akhir yang memiliki *novelty* yang memadai.

### 1.3 Rumusan Masalah

Rumusan masalah dalam penelitian sebagai berikut:

1. Bagaimana rancang bangun aplikasi Drowsinesslab berbasis *facial landmark* menggunakan OpenCV, Dlib dan Tkinter?
2. Bagaimana kinerja aplikasi Drowsinesslab tersebut?

### 1.4 Tujuan

Tujuan yang ingin dicapai dari penelitian ini yaitu:

1. Merancang dan membangun aplikasi Drowsinesslab berbasis *facial landmark* menggunakan OpenCV, Dlib dan Tkinter.
2. Mengetahui nilai akurasi menggunakan *machine learning* dan konsumsi daya, CPU dan memori dari aplikasi Drowsinesslab.

### 1.5 Manfaat

Manfaat yang ingin dicapai dari penelitian ini adalah:

1. Manfaat bagi Bidang Akademis  
Memperkaya khazanah salah satu bidang ilmu pengetahuan yaitu Bahasa Pemrograman, khususnya Python yang dapat dipakai untuk membangun sistem kecerdasan buatan dan *image processing*.
2. Manfaat Praktis  
Dengan menggunakan aplikasi deteksi kantuk ini memberikan peringatan dini kepada seseorang ketika sedang mengantuk agar terhindar dari kekeliruan dan masalah kesehatan saat bekerja.

### 1.6 Batasan Masalah

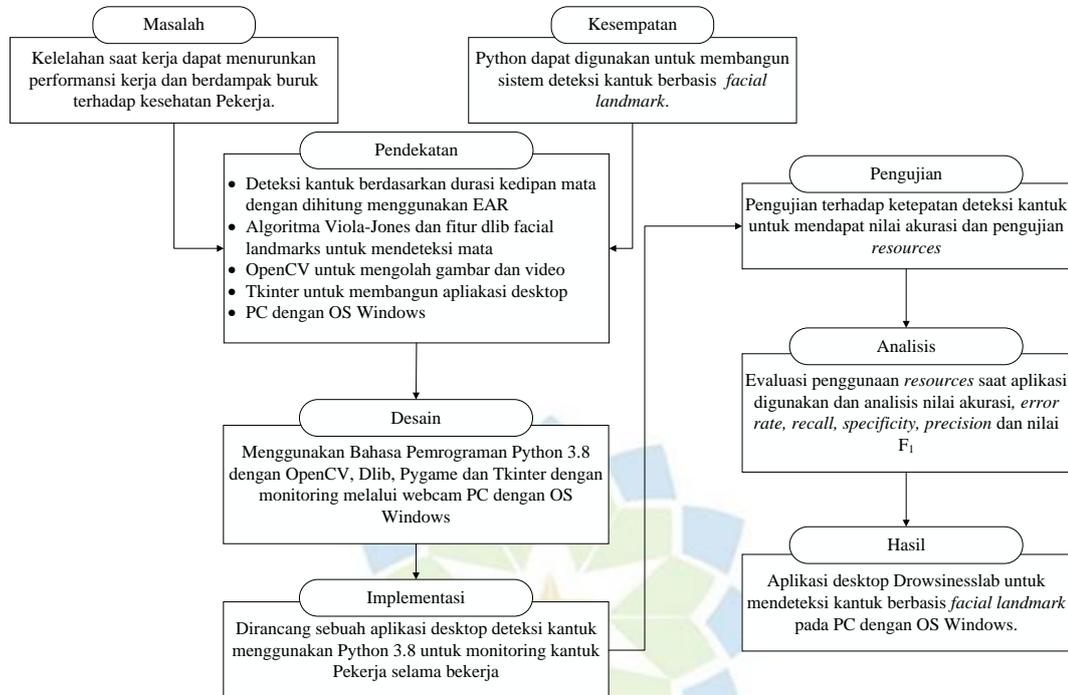
Diperlukan Batasan masalah dalam pembuatan sistem deteksi kantuk sehingga dapat diperoleh hasil yang sesuai dengan tujuan pembuatan. Adapun pembatasan masalah pada penelitian ini adalah sebagai berikut:

1. Jarak kamera dengan manusia adalah  $< 1$  meter.
2. Aplikasi dapat bekerja dengan maksimal pada PC yang telah dilengkapi *HD web camera*.
3. Aplikasi dapat digunakan pada PC dengan OS Windows.
4. Tidak ada penghalang mata ketika kamera bekerja mendeteksi wajah kecuali kacamata bening atau transparan dan kontak lensa atau *softlens*.
5. Intensitas cahaya yang cukup mendukung kinerja maksimal dari sistem yaitu lebih dari 300 Lumen.
6. Pendeteksi kantuk digunakan untuk mendeteksi satu orang pekerja pada sebuah PC.
7. Evaluasi kinerja menggunakan *machine learning*, parameter yang diukur berupa akurasi, *error rate*, *recall*, *specificity*, *precision* dan F1.
8. Evaluasi *resource* PC menggunakan *Task Manager* dan *Open Hardware Monitor*, parameter yang diukur berupa penggunaan CPU, *Memory* dan *Power*.
9. Metode deteksi wajah menggunakan *computer vision*, *haar cascade classifier* untuk lokalisasi wajah, *dlib's facial landmark* untuk lokalisasi mata, dan *Eye Aspect Ratio* (EAR) mengukur keterbukaan mata.
10. Nilai *threshold* EAR diatur sebesar 0.3 sebagai nilai ambang batas mata tertutup yang mampu memberikan hasil deteksi sebagai kantuk.
11. Nilai *threshold frame* diatur sebesar 50 sebagai nilai ambang batas jumlah *frame* dan nilai fps yang dimiliki webcam 720p adalah 15 fps sehingga durasi minimum seseorang terdeteksi kantuk adalah 3 detik.
12. Desain aplikasi menggunakan Python, OpenCV untuk mengolah gambar dan video, Tkinter membuat GUI, dan Pygame sebagai pustaka suara alarm

## 1.7 Kerangka Berfikir

Kerangka berfikir dibuat berdasarkan pemikiran yang dimuat dalam bagan sistematis mengenai informasi hasil perumusan masalah agar mempercepat

pemahaman terkait alur logis penelitian dan penelitian dapat dilakukan secara terstruktur. Gambar 1.1 menjelaskan kerangka berfikir sebagai berikut.



Gambar 1.1 Kerangka berfikir

## 1.8 Sistematika Penulisan

Penulisan proposal penelitian ini memiliki sistematika penulisan berikut penjabarannya:

### BAB I PENDAHULUAN

Pada bab ini meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, *state of the art*, kerangka berfikir, dan sistematika penulisan.

### BAB II TEORI DASAR

Pada bab ini membahas mengenai dasar teori dan pandangan umum tentang kantuk, *dlib's facial landmark detector*, *haar cascade classifier*, *Eye Aspect Ratio (EAR)*, *computer vision*, Python, OpenCV, pygame, dan tkinter.

### BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metodologi dan jadwal penelitian.

### BAB IV PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi penjelasan mengenai perancangan aplikasi Drowsinesslab yaitu perancangan sistem dan desain aplikasi. Selain itu dijelaskan bagaimana implementasi aplikasi Drowsinesslab.

#### BAB V PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan pengujian terhadap aplikasi Drowsinesslab yang terdiri dari pengujian kinerja dan pengukuran *resources*. Dijelaskan juga analisis dari hasil pengujian.

#### BAB VI PENUTUP

Pada bab ini berisi kesimpulan dan saran.

