

**APLIKASI DETEKSI KELELAHAN MATA SAAT  
MENGUNAKAN KOMPUTER DENGAN  
HAAR CASCADE & EAR**

**TUGAS AKHIR**



diajukan oleh:

**Nama : Athabiq Tuah Wibisana**

**NIM : 19.01.4295**

**PROGRAM DIPLOMA  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2023**

**APLIKASI DETEKSI KELELAHAN MATA SAAT  
MENGUNAKAN KOMPUTER DENGAN  
HAAR CASCADE & EAR**

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat mencapai gelar Ahli Madya  
Komputer Program Diploma – Program Studi Teknik Informatika



diajukan oleh

**Nama : Athabiq Tuah Wibisana**

**NIM : 19.01.4295**

**PROGRAM DIPLOMA  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA**

**2023**

**HALAMAN PERSETUJUAN**  
**TUGAS AKHIR**  
**APLIKASI DETEKSI KELELAHAN MATA SAAT**  
**MENGGUNAKAN KOMPUTER DENGAN**  
**HAAR CASCADE & EAR**

yang dipersiapkan dan disusun oleh

**Athabiq Tuah Wibisana**

**19.01.4295**

Telah disetujui oleh Dosen Pembimbing Tugas Akhir  
pada tanggal 21 September 2022

Dosen Pembimbing



**Melwin Syafrizal, S.Kom., M.Eng**

**NIK. 190302105**

**HALAMAN PENGESAHAN**  
**TUGAS AKHIR**  
**APLIKASI DETEKSI KELELAHAN MATA SAAT**  
**MENGGUNAKAN KOMPUTER DENGAN**  
**HAAR CASCADE & EAR**

yang disusun dan diajukan oleh

**Athabiq Tuah Wibisana**

**19.01.4295**

Telah dipertahankan di depan Dewan Penguji  
pada tanggal 24 Februari 2023

**Susunan Dewan Penguji**

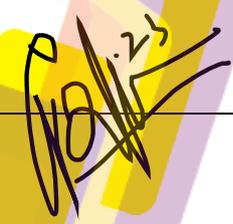
**Nama Penguji**

**Tanda Tangan**

**Hastari Utama, M.Cs**  
**NIK. 190302230**



**Wahid Miftahul Ashari, S.Kom., M.T**  
**NIK. 190302452**



Tugas Akhir ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Ahli Madya komputer  
Tanggal 24 Februari 2023

**DEKAN FAKULTAS ILMU KOMPUTER**



**Hanif Al Fatta, S.Kom., M.Kom.**  
**NIK. 190302096**

## HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertandatangan di bawah ini,

Nama mahasiswa : Athabiq Tuah Wibisana  
NIM : 19.01.4295

Menyatakan bahwa Tugas Akhir dengan judul berikut:

**Aplikasi Deteksi Kelelahan Mata Saat Menggunakan Komputer Dengan Haar Cascade & EAR**

Dosen Pembimbing : Melwin Syafrizal, S.Kom., M.Eng

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 24 Februari 2023

Yang Menyatakan,



Athabiq Tuah Wibisana

## HALAMAN PERSEMBAHAN

Dengan rasa syukur telah dibuatnya Tugas Akhir ini disusun untuk memenuhi persyaratan kelulusan pada Program Studi D3-Teknik informatika di Universitas Amikom Yogyakarta. Oleh karena itu saya mempersembahkannya kepada:

1. Saya persembahkan Tugas Akhir ini kepada orang tua dan keluarga besar saya yang telah membantu dan memberi motivasi untuk menyelesaikan Tugas Akhir ini.
2. Teman-teman saya baik itu teman kuliah dari Universitas Amikom Yogyakarta maupun universitas lain, teman kerja, dan teman daring yang merupakan seorang pengembang atau *Developer*.
3. Klub AMCC dari Universitas Amikom Yogyakarta terutama tim *Desktop Application*
4. Segenap staf pengajar, karyawan, dan seluruh mahasiswa Universitas Amikom Yogyakarta.

## KATA PENGANTAR

Dengan rasa syukur yang ditujukan kepada Tuhan Yang Maha Esa Allah SWT. Dengan diberinya kesempatan, rahmat, dan karunia untuk menyelesaikan Tugas Akhir yang berjudul, “APLIKASI DETEKSI KELELAHAN MATA SAAT MENGGUNAKAN KOMPUTER DENGAN HAAR CASCADE & EAR” yang dapat Penulis selesaikan. Penulis juga berharap bahwa Tugas Akhir bisa diberi peningkatan bagi yang membaca Tugas Akhir ini.

Pada kesempatan ini, Penulis mengucapkan terima kasih kepada semua pihak yang telah membantu, memberikan semangat dan motivasi dalam pembuatan Tugas Akhir ini yang termasuk kepada:

1. Kepada keluarga saya yang selalu memberikan kasih sayang, doa, dukungan dan motivasi selama menjalani studi di Universitas Amikom Yogyakarta
2. Bapak Melwin Syafrizal, S.Kom., M.Eng selaku Dosen Pembimbing penulis
3. Bapak dan ibu selaku dosen prodi D3-Teknik Informatika
4. Dosen-dosen ilmu komputer Universitas Amikom Yogyakarta
5. Bapak Wahid Miftahul Ashari, S.Kom., M.T Dan Hastari Utama, M.Cs Sebagai Penguji pada Ujian Tugas Akhir

Semoga diberikan pahala dan kesehatan dalam menjalani aktivitasnya atas segala bentuk bantuan yang diberikan kepada penulis selama menjalani studinya di Universitas Amikom Yogyakarta. Penulis berharap Tugas Akhir ini dapat menambah pengetahuan bagi pembaca untuk memperluas pemikiran dalam membuat sebuah aplikasi terutama bagi kesehatan dan kepedulian terhadap mata. Penulis mengucapkan permohonan maaf jika selama proses penyusunan Tugas Akhir banyak melakukan kesalahan, baik secara lisan maupun tulisan.

Yogyakarta, 15 Januari 2023

Penulis

## DAFTAR ISI

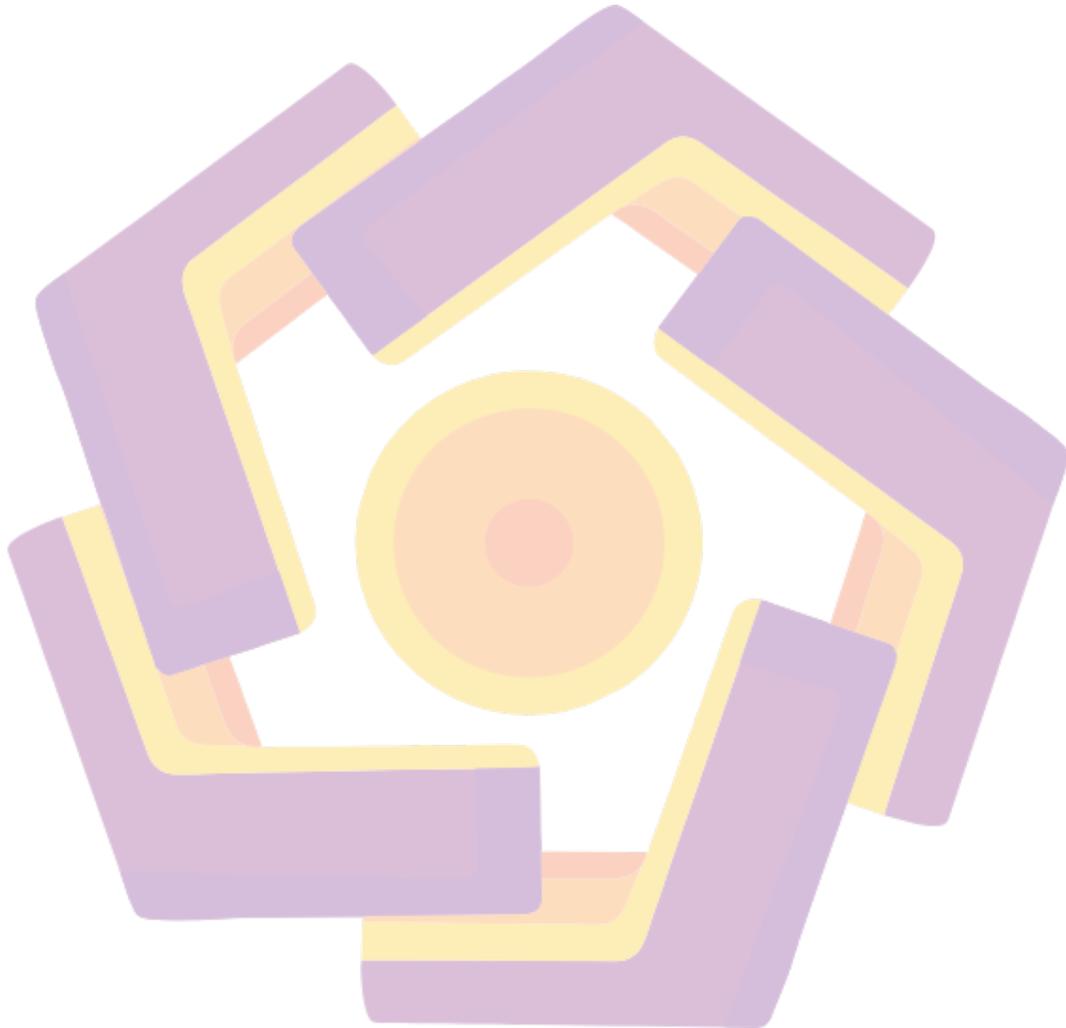
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiv
DAFTAR LAMBANG DAN SINGKATAN	xv
DAFTAR ISTILAH	xvi
INTISARI	xvii
ABSTRACT	xviii
<b>BAB I</b>	
<b>PENDAHULUAN</b>	<b>1</b>
1. 1 Latar Belakang	1
1. 2 Perumusan masalah	2
1. 3 Tujuan Penelitian	2
1. 4 Batasan Masalah	2
1. 5 Manfaat Penelitian	2
<b>BAB II</b>	
<b>TINJAUAN PUSTAKA</b>	<b>4</b>
2. 1 Penelitian Terdahulu	4
2. 2 Literature Review	15
2.2.1. Eye Fatigue	15
2.2.2. Computer vision syndrome	15

2.2.3. Graphical User Interface	16
2.2.4. Threading	16
2. 3 Landasan Teori	17
2.2.1. EAR	17
2.2.2. Euclidean distance	18
2.2.3. Haar Cascade	18
2.2.4. Opencv	19
2.2.5. System Tray	19
2.2.6. Face Landmark	20
<b>BAB III</b>	
<b>METODOLOGI PENELITIAN</b>	22
3.1 Pengumpulan kebutuhan	22
3.1.1. Metode pengembangan sistem	22
3.2.1. Analisis	22
3.2.2. Desain	23
3.2.3. Implementasi	28
3.2.4. Pengujian	29
3.2.5. Pemeliharaan	32
<b>BAB IV</b>	
<b>HASIL DAN PEMBAHASAN</b>	33
4.1 Implementasi	33
4.1.1. Penguinstalan Library	33
a. Dlib	34
b. Numpy dan Scipy	35
c. Imutils	36
d. Pathlib	37
e. Pygame	37
f. Opencv	39
g. Tkinter	40
h. PIL	40
i. Pystray	41

j. Win10toast	42
k. Time dan Date time	42
l. Threading	43
m. Deep Face	44
n. Request dan JSON	45
4.1.2. Implementasi Library pada program	46
a. Tampilan kode pada welcome.py	46
b. Tampilan kode pada main.py	48
4.2 Pengujian	57
4.2.1. Tampilan GUI welcome.py	58
4.2.2. Tampilan GUI program main.py	60
<b>BAB V</b>	
<b>KESIMPULAN DAN SARAN</b>	70
5.1 Kesimpulan	70
5.2 Saran	70
<b>DAFTAR PUSTAKA</b>	71
<b>LAMPIRAN</b>	73
1. Source Code welcome.py	73
2. Source code main.py	77
3. Tampilan Flowchart seluruh aplikasi.	114

## DAFTAR TABEL

Tabel 2.1 Tabel penelitian terdahulu.	4
Tabel 2.2 Tabel perbandingan penelitian.	9
Tabel 3.1 Tabel kondisi mata yang dihitung dengan EAR.	24
Tabel 3.2 Tabel pengujian Blackbox main.py.	30
Tabel 3.3 Tabel pengujian Blackbox welcome.py.	32
Tabel 4.2 Tabel pengujian.	58

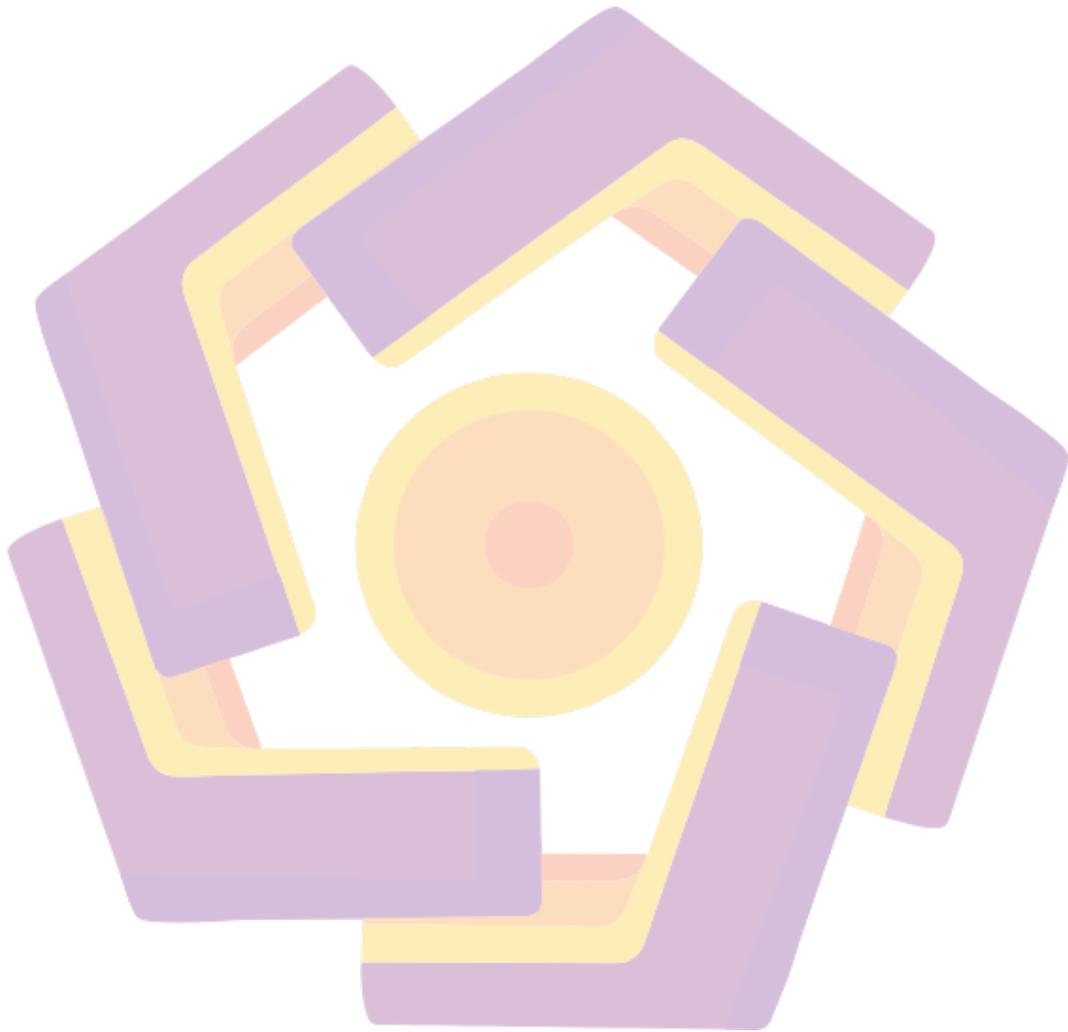


## DAFTAR GAMBAR

Gambar 2.1. Tampilan EAR saat di implementasi dengan Opencv.	17
Gambar 2.2. Teknik Haar Feature yang digunakan pada Haar Cascade.	19
Gambar 2.3. Opencv dengan Haar Cascade untuk mendeteksi wajah.	20
Gambar 2.4. Tampilan <i>System Tray</i> pada sebuah aplikasi.	21
Gambar 2.5. titik dan point pada <i>Facial Landmark</i> .	21
Gambar 2.6. <i>Facial Landmark</i> dengan dlib dan opencv..	22
Gambar 3.1. Aliran metode waterfall pada program	23
Gambar 3.2. Tampilan seluruh layar monitor dengan GUI berukuran 600 x 334.	25
Gambar 3.3. Tampilan seluruh layar monitor dengan GUI berukuran 950 x 640.	25
Gambar 3.4. Flowchart pendeteksi kelelahan mata pada program.	26
Gambar 3.5. Flowchart dari program GUI main.py.	27
Gambar 3.7. Flowchart fitur penggunaan cuaca.	28
Gambar 3.8. Tampilan diagram UML pada aplikasi.	29
Gambar 4.1. Cara instalasi menggunakan pip jika python terdaftar di Path..	34
Gambar 4.2. Cara instalasi menggunakan pip jika python tidak terdaftar di Path.	34
Gambar 4.3. Pengujian jika library dapat digunakan pada IDE VScode.	34
Gambar 4.4. Pengujian jika library tidak dapat digunakan pada IDE VScode.	35
Gambar 4.5. Cara menggunakan version untuk melakukan tes sebuah <i>Library</i> .	35
Gambar 4.6. Melakukan tes dlib jika berhasil di install dengan output versinya.	36
Gambar 4.9. Melakukan tes Numpy dan scipy dengan output versinya.	36
Gambar 4.10 Contoh kode untuk menampilkan mean dan nilai <i>Minimum</i>	37
Gambar 4.11. Melakukan tes imutils dengan output versinya.	37
Gambar 4.12. Contoh tes kode untuk mengubah gambar.	37
Gambar 4.13. <i>Output</i> dari kode diatas dengan opencv.	38
Gambar 4.14. Melakukan tes pathlib dengan menggunakan modul path.	38
Gambar 4.15. Melakukan tes pygame dengan memberikan output versinya.	39
Gambar 4.16. Contoh tes kode dengan pygame untuk membuat sebuah game.	39
Gambar 4.17. <i>Output</i> pada kode diatas.	39
Gambar 4.18. Melakukan tes opencv dengan memberi output versinya.	40
Gambar 4.19. Contoh tes kode dengan Opencv untuk menampilkan gambar.	40
Gambar 4.20. <i>Output</i> pada kode opencv diatas.	40
Gambar 4.21. kode untuk melakukan tes pada tkinter.	41
Gambar 4.22. Output tampilan kode.	41
Gambar 4.23. Melakukan tes PIL dengan menampilkan versinya.	41
Gambar 4.24. Contoh tes kode untuk mengubah ukuran gambar dengan PIL.	42
Gambar 4.25. Contoh kode untuk melakukan tes pystray.	42
Gambar 4.26. Melakukan tes pystray pada aplikasi.	43
Gambar 4.27. Contoh tes kode dengan Win10toast.	43
Gambar 4.28. Tampilan dari <i>Output</i> kode win10toast diatas	43
Gambar 4.29. Melakukan tes date time dan time untuk menampilkan waktu.	44
Gambar 4.30. Contoh kode program tanpa threading.	44
Gambar 4.31. <i>Output</i> kode tanpa threading dengan waktunya.	44
Gambar 4.32. Contoh kode program dengan threading.	45
Gambar 4.33. Output kode threading dengan waktunya.	45
Gambar 4.34. Contoh tes kode untuk menampilkan data wajah.	46
Gambar 4.35. Tampilan deepface yang menampilkan status wajah.	46

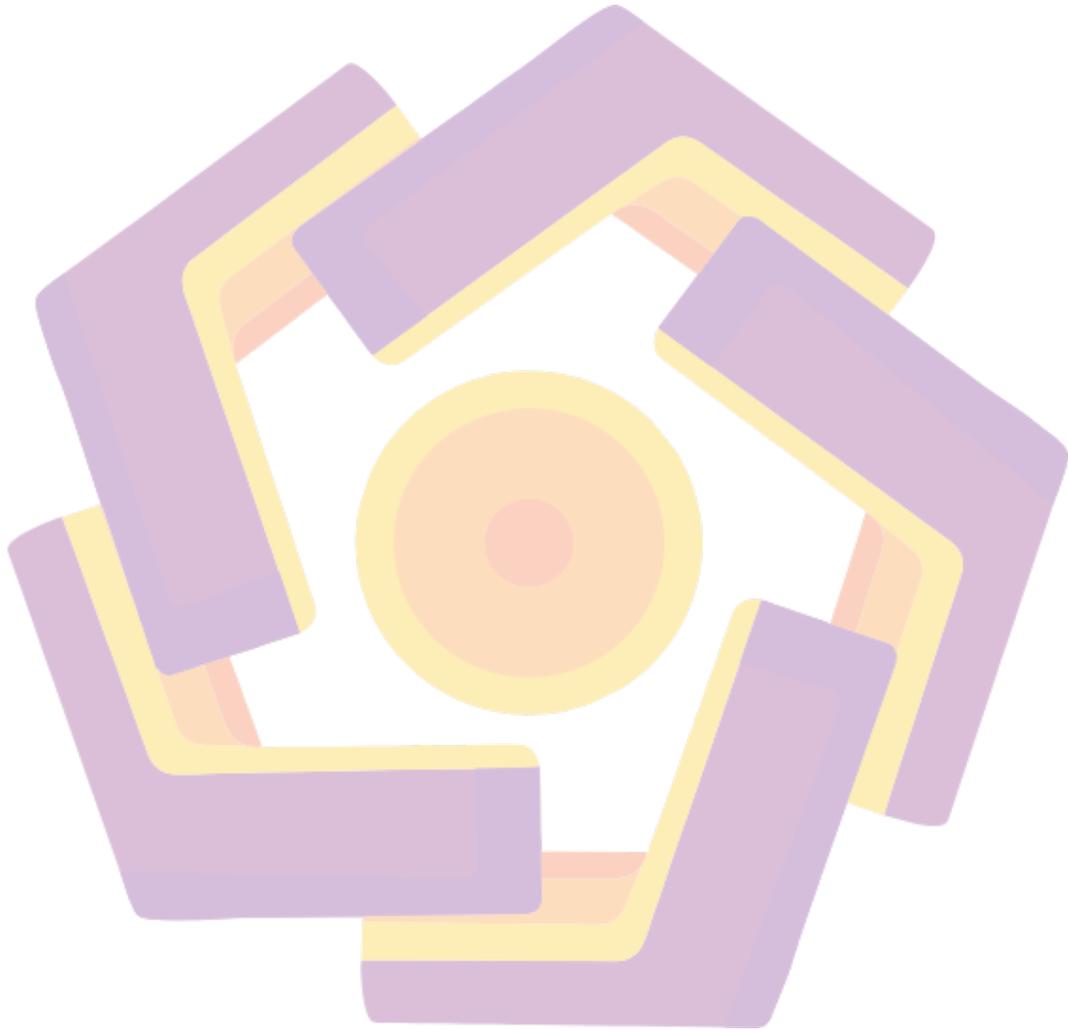
Gambar 4.36. Tampilan tes kode penggunaan request dan JSON.	47
Gambar 4.37. fitur <i>VideoCapture</i> opencv2 yang digunakan pada <i>welcome.py</i> .	47
Gambar 4.38. Proses menampilkan video dengan canvas dan opencv.	48
Gambar 4.39. Kode agar video berjalan terus.	48
Gambar 4.40. Kode dimana pengecekan kamera dilakukan.	48
Gambar 4.41. Kode dimana ketika kamera utama ditemukan.	49
Gambar 4.42. Tampilan <i>Library</i> yang digunakan <i>main.py</i> .	49
Gambar 4.43. Tampilan peran <i>scipy</i> untuk menghitung jarak euclidean.	50
Gambar 4.44. Tampilan peran <i>face_utils</i> untuk mengambil indeks.	50
Gambar 4.45. Tampilan peran <i>numpy</i> sebagai operator perhitungan array.	50
Gambar 4.46. Tampilan peran canvas sebagai peran desain pada GUI.	51
Gambar 4.47. PIL sebagai pemuatan gambar dari <i>array</i> .	51
Gambar 4.48. Datetime untuk menampilkan tanggal dan waktu.	51
Gambar 4.49. Pathlib sebagai pemersingkat jalur file.	52
Gambar 4.50. Peran opencv2 sebagai <i>real-time capture</i> .	52
Gambar 4.51. Peran opencv2 sebagai pendeteksi wajah dengan Haar Cascade.	52
Gambar 4.52. Winsound sebagai pemutar suara pada tombol.	53
Gambar 4.53. Deepface sebagai fitur pendeteksi status wajah pengguna.	53
Gambar 4.54. Requests dan JSON digunakan sebagai fitur ramalan cuaca..	53
Gambar 4.55. Pemanggilan data cuaca dari hasil JSON Openweather.org.	54
Gambar 4.56. <i>Library</i> random sebagai pemutar GIF secara acak.	54
Gambar 4.57. <i>Library</i> ctypes untuk menghitung waktu <i>uptime</i> .	54
Gambar 4.58. <i>Library</i> webbrowser digunakan sebagai <i>Hyperlink</i> .	55
Gambar 4.59. Dlib sebagai alat untuk mendeteksi wajah.	55
Gambar 4.60. Proses dimana dlib melakukan pendeteksian wajah.	55
Gambar 4.61. Dlib digunakan untuk melakukan ekstraksi index ke <i>Array</i> .	55
Gambar 4.62. Tampilan hasil array untuk perhitungan EAR.	56
Gambar 4.63. Win10toast dan pygame sebagai notifikasi ketika pengguna lelah.	56
Gambar 4.64. Threading sebagai <i>multi-process</i> pada program.	57
Gambar 4.65. Pystray sebagai fitur <i>system tray</i> pada program.	57
Gambar 4.66. <i>Library</i> os yang digunakan untuk menggunakan nama komputer.	57
Gambar 4.67. Tampilan pencarian kamera.	59
Gambar 4.68. Tampilan pencarian kamera sukses	60
Gambar 4.69. Tampilan selamat datang.	60
Gambar 4.70. Tampilan pilihan untuk pengguna.	61
Gambar 4.71. Tampilan utama program.	62
Gambar 4.72. Tampilan kinerja ramalan cuaca.	62
Gambar 4.73. Tampilan error jika salah memasukan kota.	63
Gambar 4.74. Tampilan fitur deep face pada program ketika wajah ditemukan.	64
Gambar 4.75. Tampilan fitur deepface ketika wajah pengguna tidak ditemukan.	64
Gambar 4.76. Tampilan fitur <i>camera preview</i> .	65
Gambar 4.77. Tampilan jika tombol dimatikan.	65
Gambar 4.78. Tampilan notifikasi ketika mendeteksi mata pengguna lelah.	66
Gambar 4.79. Tampilan GIF ketika mata pengguna lelah.	66
Gambar 4.80. Tampilan aplikasi pada mode <i>system tray</i>	67
Gambar 4.82. Tampilan perubahan latar pada waktu pagi hari.	67
Gambar 4.83. Tampilan perubahan latar pada waktu siang hari.	68
Gambar 4.84. Tampilan perubahan latar pada waktu sore hari.	68

Gambar 4.85. Tampilan perubahan latar pada waktu memasuki malam hari.	69
Gambar 4.86. Tampilan perubahan latar pada waktu tengah malam.	69
Gambar 4.87 Tampilan ketika semua fitur di aktifkan.	70



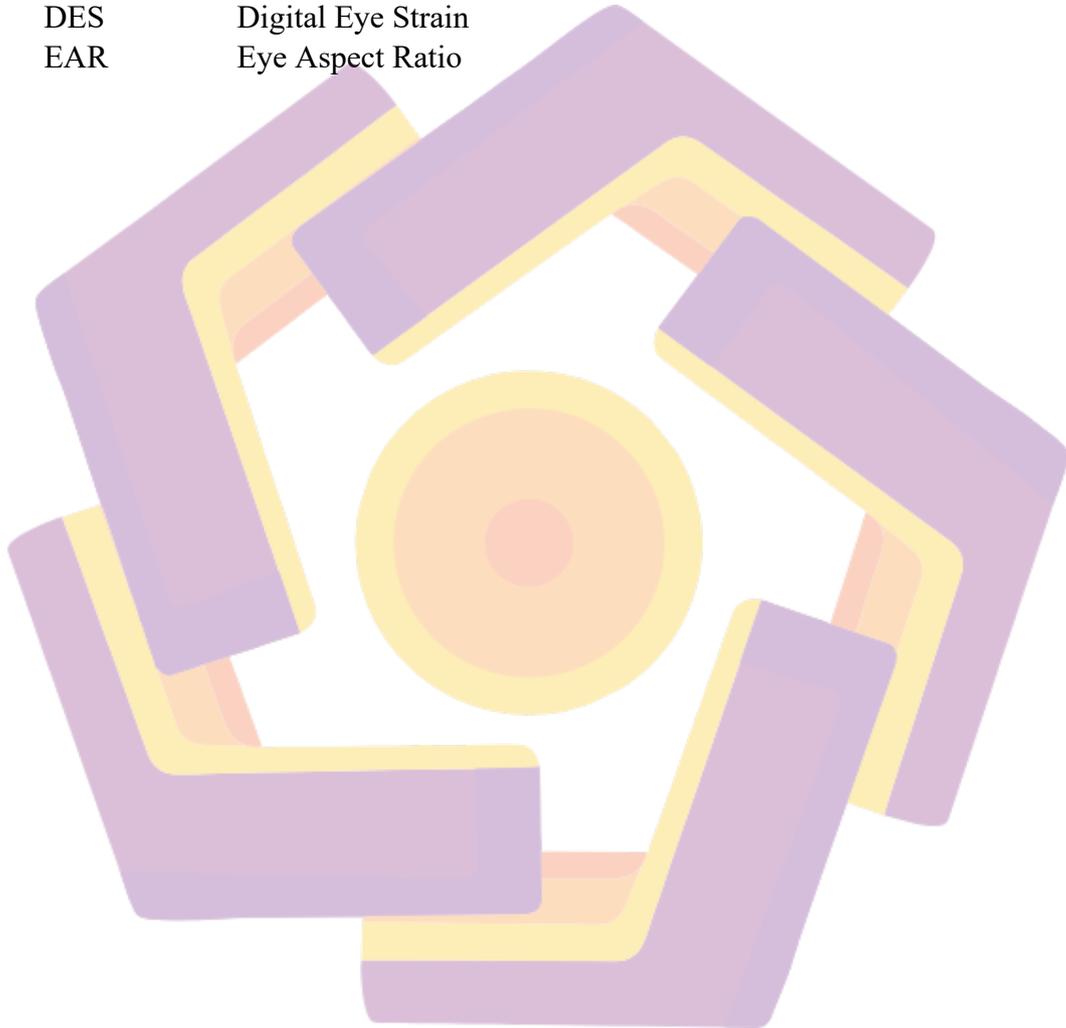
## DAFTAR LAMPIRAN

Lampiran 1. <i>Source code</i> welcome.py	57
Lampiran 2. <i>Source code</i> main.py	61
Lampiran 3. Tampilan Flowchart seluruh aplikasi.	98



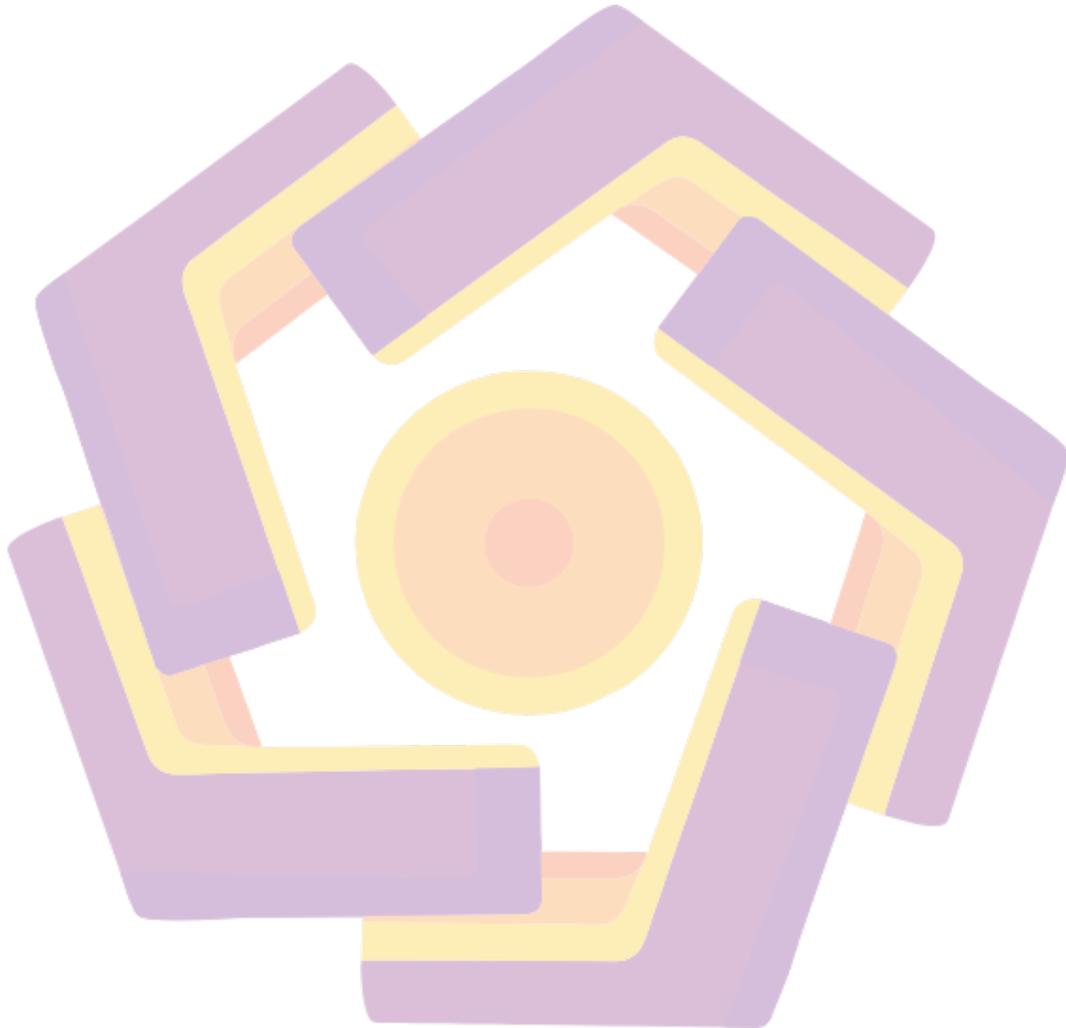
## DAFTAR LAMBANG DAN SINGKATAN

PIL	Python Imaging Library
EFD	Eye Fatigue Detection
JSON	JavaScript Object Notation
VDT	Visual Display Terminals
GUI	Graphical User Interface
UML	Unified Modeling Language
CVS	Computer Vision Syndrome
DES	Digital Eye Strain
EAR	Eye Aspect Ratio



## DAFTAR ISTILAH

Image processing	Proses pengolahan gambar .
Object detection	Proses menemukan dan mengekstrak objek.
Quality of life	Kualitas dan kenyamanan.
Computer Vision	Komputer dapat memahami dan menafsirkan gambar.
Euclidean	Perhitungan jarak antara dua titik.
Function	kumpulan perintah yang digunakan menjalankan tugas.



## INTISARI

Kelelahan merupakan hal umum yang terjadi pada kehidupan sehari-hari seperti saat bekerja, berkendara, dan berolahraga. Saat lelah kita terkadang tanpa sadar sering memaksa tubuh kita untuk bekerja terutama pada mata kita. Di masa sekarang ini yang dimana komputer lebih digunakan dalam melakukan pekerjaan dan tentu saja mata merupakan bagian tubuh yang lebih dominan digunakan saat melakukan aktivitas di depan komputer seperti bekerja, bermain game, berkomunikasi, dan hal lainnya. Dan disitulah dimulainya orang sering tanpa sadar memaksa matanya dalam melakukan aktivitasnya.

Terdapat sebuah library *Machine Learning* yang umum digunakan, yaitu *Opencv2*. *Opencv* digunakan untuk melakukan *real-time capture* dengan menggunakan algoritme seperti Haar Cascade untuk mendeteksi sebuah objek dengan menggunakan *Computer vision*. *Computer vision* melakukan metode *Acquiring*, *Processing*, dan *Analyzing* untuk memahami atau mempelajari suatu objek. Metode ini dapat digunakan untuk mendeteksi permukaan wajah seseorang dengan *Dlib*. Setelah *Dlib* melakukan prediksi pada wajah seseorang, selanjutnya dilakukanlah ekstraksi indeks kedua mata dengan *imutils* yang merupakan sebuah *library* untuk *Image Processing*. Jika ditemukan indeks kedua mata, maka dilakukan pendeteksian pada mata dengan algoritme EAR atau disebut juga dengan *Eye Aspect Ratio* yang dimana algoritme tersebut digunakan untuk mendeteksi kedipan mata. Ada *Threshold* dalam algoritme EAR saat mendeteksi kedipan mata. Jika mata melewati *Threshold* tersebut, maka mata sedang dalam keadaan berkedip. Jika tekanan tersebut diganti dengan posisi mata sebelum tertutup total, maka dapat digunakan untuk mendeteksi kelelahan pada mata.

Dengan menggunakan algoritme dan library tersebut, dapat dibuat aplikasi pendeteksi kelelahan mata bagi pengguna komputer yang dimana aplikasi tersebut bekerja untuk memonitori kondisi mata pengguna komputer lelah atau tidak. Jika mata pengguna lelah, maka pengguna diberi pemberitahuan. Penelitian ini dapat dimanfaatkan bagi orang yang tertarik dengan pengaplikasian *Computer vision* dan membuat aplikasi tersebut dapat digunakan bagi orang-orang yang tidak terlalu mendalami ilmu komputer. Penelitian ini juga bermanfaat bagi orang yang tertarik dalam kepedulian kesehatan manusia terutama dalam menggunakan komputer yang dimana sekarang ini orang-orang lebih sering menggunakan komputer dalam kehidupan kesehariannya.

**Kata kunci:** Algoritme, Perpustakaan, Mata, *Computer vision*, aplikasi

## **ABSTRACT**

*Fatigue is common in everyday life, such as when working, driving, and exercising. When we are tired, we unconsciously force our bodies to work, especially on our eyes. In the present era, where computers are more used in doing work, the eyes are the part of the body that is more dominantly used when doing activities in front of the computer, such as working, playing games, communicating, and other things. And that's where people often unknowingly force their eyes to do their activities.*

*There is a commonly used Machine Learning library, namely Opencv2. Opencv is used to do real-time capture using algorithms such as Haar Cascade to detect an object using Computer vision. Computer vision performs Acquiring, Processing, and Analyzing methods to understand or study a thing. This method can detect the surface of a person's face with Dlib. After Dlib makes predictions on a person's face, the index extraction of both eyes is performed with cuteils, a library for Image Processing. If the indices of both eyes are found, then the eye is detected with the EAR algorithm, also known as the Eye Aspect Ratio, where the algorithm detects eye blinks. There is a Threshold in the EAR algorithm when it sees an eye blink. If the eye passes the Threshold, then the eye is blinking. If the position of the eye replaces the pressure before it is completely closed, it can be used to detect eye fatigue.*

*Using these algorithms and libraries, an eye fatigue detection application can be made for computer users where the application works to monitor the condition of computer users' eyes if they are tired. If the user's eyes are tired, then the user is notified. This research can be used for people interested in computer vision and make the application usable for people not too deep into computer science. This research is also helpful for people interested in human health care, especially in using computers, where people nowadays use computers more often in their daily lives.*

**Keyword:** *Algorithm, Library, Eyes, Computer vision, Application*