

**PERANCANGAN APLIKASI WEB PENDETEKSI KANTUK
MENGUNAKAN ALGORITMA *YOLO***

SKRIPSI

Oleh:

**EVELYN GLENISZ
NIM. 2044031**



**JENJANG PENDIDIKAN STRATA-1
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME
MEDAN**

2024

LEMBAR PENGESAHAN

PERANCANGAN APLIKASI WEB PENDETEKSI KANTUK MENGUNAKAN ALGORITMA *YOLO*

SKRIPSI

Diajukan Untuk Melengkapi Persyaratan Guna

Mendapat Gelar Sarjana Strata Satu

Program Studi Teknik Informatika

Oleh :

EVELYN GLENISZ

NIM. 2044031

Disetujui Oleh:

Dosen Pembimbing I



Octara Pribadi, M.Kom

Dosen Pembimbing II



Robet, M.Kom

Medan, Juli 2024

Diketahui dan Disahkan Oleh :

Ketua STMIK TIME

Ketua Program Studi



Dr. Edi Wijaya, S.Kom, M.Kom, MM

Robet, M.Kom

Kepada

Yth. Ketua STMIK TIME

Dr. Edi Wijaya, S.Kom., M.Kom., M.M

Di Medan

Hal : **Surat Pernyataan Bebas Plagiat**

Saya yang bertanda tangan di bawah ini :

Nama : **Evelyn Glenisz**

NIM : **2044031**

Program Studi : **Teknik Informatika**

Angkatan : **2020**

Judul Skripsi : **Perancangan Aplikasi Web Pendeteksi Kantuk Menggunakan Algoritma YOLO**

Dengan ini menyatakan bahwa skripsi yang Saya tulis adalah karya ilmiah asli dari pikiran saya sendiri. Apabila kemudian hari terbukti bahwa skripsi tersebut merupakan Plagiat, maka saya bersedia gelar dan ijazah yang saya peroleh ditinjau kembali.

Demikian surat pernyataan ini Saya perbuat dengan pikiran yang sehat, tanpa tekanan dari siapapun.

Medan, **31 Juli 2024**

Yang membuat pernyataan,



Evelyn Glenisz

Diketahui oleh,



Hendri, M.Kom

Pembantu Ketua 1 STMIK TIME

ABSTRAK

Setiap orang bisa saja merasa kantuk jika tidak beristirahat dengan cukup dan hal tersebut dapat mengganggu aktivitas yang akan dilakukan. Tentunya itu akan membahayakan jika seseorang kantuk saat mengemudi mobil hingga sepeda motor. Kantuk (*drowsiness*) adalah kondisi dimana seseorang atau individu membutuhkan istirahat. Oleh karena itu, diperlukan suatu sistem yang dapat mendeteksi kantuk secara real-time. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah website yang dapat mendeteksi kantuk dengan menggunakan algoritma YOLO varian YOLOv8. YOLO adalah model *deep learning* yang terkenal dengan kecepatan dan akurasi dalam mendeteksi objek. Dataset yang digunakan berupa gambar kantuk dan tidak kantuk. Website ini dikembangkan dengan website yang diintegrasikan dengan model YOLOv8 yang telah dilatih, kemudian mendeteksi kantuk melalui kamera web secara real-time. Hasil pengujian menunjukkan akurasi yang tinggi, serta dapat memberikan peringatan kepada pengguna.

Keyword: Deteksi Kantuk, *You Only Look Once (YOLO)*, Real-time

ABSTRACT

Everyone can feel drowsy if they do not get enough rest, and this can interfere with their activities. This becomes particularly dangerous if someone feels drowsy while driving a car or riding a motorcycle. Drowsiness is a condition where a person or individual needs rest. Therefore, a system that can detect drowsiness in real-time is necessary. This research aims to design and implement a website that can detect drowsiness using the YOLO algorithm, specifically the YOLOv8 variant. YOLO is a deep learning model known for its speed and accuracy in object detection. The dataset used consists of images depicting drowsiness and non-drowsiness. The website is developed by integrating it with the trained YOLOv8 model, then detecting drowsiness through a webcam in real-time. Test results show high accuracy and the ability to provide warnings to users.

Keyword: *Drowsiness Detection, You Only Look Once (YOLO), Real-time*

KATA PENGANTAR

Puji dan ucapan syukur kepada kehadiran Tuhan Yang Maha Esa atas segala berkat rahmat dan hidayahnya sehingga penulis dapat menyelesaikan skripsi yang berjudul “PERANCANGAN APLIKASI WEB PENDETEKSI KANTUK MENGGUNAKAN ALGORITMA YOLO”.

Penelitian dari Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan program Strata-1 (S-1) pada Program Studi Sistem Informasi pada Sekolah Tinggi Manajemen Informatika dan Komputer (STMIKTIME). Penulis berharap agar hasil dari penelitian yang dilakukan ini dapat membawa banyak manfaat bagi berbagai pihak.

Dalam menyelesaikan tugas akhir ini, penulis mendapat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, melalui kesempatan ini penulis ingin menyampaikan rasa terimakasih kepada:

1. Bapak Octara Pribadi, M.Kom, selaku Pembimbing I yang telah membimbing penulis selama mengerjakan skripsi.
2. Bapak Robet, M.Kom, selaku Pembimbing II yang telah membimbing penulis selama mengerjakan skripsi ini sekaligus selaku Ketua Jurusan Program Studi Teknik Informatika STMIK TIME Medan.
3. Bapak Simon Kangga Lee, selaku Ketua Yayasan STMIK TIME Medan.
4. Bapak Dr. Edi Wijaya, S.Kom., M.M, selaku Ketua STMIK TIME Medan.
5. Bapak Hendri, M.Kom., selaku Pembantu Ketua I (Puket I) STMIK TIME Medan.
6. Seluruh Dosen STMIK TIME Medan yang telah mendidik dan membimbing penulis selama masa perkuliahan.
7. Orang tua penulis yang telah banyak memberikan dukungan kepada penulis.
8. Keluarga dan sanak saudara yang telah banyak memberikan dukungan kepada penulis.
9. Teman-teman yang telah banyak membantu penulis dan memberikan dorongan serta semangat dalam menyelesaikan skripsi ini.

10. Semua pihak yang telah membantu hingga selesainya penulisan skripsi ini.

Penulis menyadari bahwa dalam proses penyelesaian skripsi yang berjudul “Perancangan Aplikasi Web Pendeteksi Kantuk Menggunakan Algoritma YOLO” masih belum sempurna. Oleh karena itu, penulis mengharapkan saran, kritikan dan masukan yang dapat membangun demi perkembangan dan kemajuan penelitian selanjutnya. Penulis memohon maaf apabila terdapat kekurangan dalam skripsi ini. Semoga dalam penulisan skripsi ini dapat bermanfaat bagi pihak yang berkepentingan.

Medan, 31 Juli 2024

Penulis,



Evelyn Glenisz

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN.....	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan dan Manfaat Penelitian.....	3
1.5 Metodologi Penelitian	4
1.6 Analisis Sistem	6
1.7 Perancangan Sistem.....	6
1.8 Sistematika Penulisan.....	7
BAB II TINJAUAN PUSTAKA	9
2.1 Kantuk	9
2.2 Web.....	10
2.3 JavaScript	11
2.4 Python.....	11
2.5 Google Colab.....	12

2.6	<i>Deep Learning</i>	12
2.7	<i>Convolutional Neural Network (CNN)</i>	13
2.8	<i>You Only Look Once</i>	14
2.9	Arsitektur YOLOv8	17
2.10	<i>Confusion Matrix</i>	18
2.11	Penelitian Terdahulu	20
BAB III METODE PENELITIAN		25
3.1	Analisis	25
3.1.1	Analisis Proses	25
3.1.2	Analisis Metode Yang Digunakan	28
3.1.3	Proses Konversi ke Web	29
3.1.4	Proses Pengaktifan Kamera	29
3.2	Perancangan	30
3.2.1	Perancangan Tampilan	30
BAB IV HASIL DAN PEMBAHASAN		36
4.1	Hasil	36
4.1.1	Hasil Pengujian	36
4.1.2	Hasil Tampilan	41
4.2	Pembahasan	47
4.2.1	Kelebihan Sistem	47
4.2.2	Kekurangan Sistem	48
BAB V KESIMPULAN DAN SARAN		49
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA		51

DAFTAR TABEL

Tabel 1.1 Jadwal Penelitian	4
Tabel 2.1 Penelitian Terdahulu	20

DAFTAR GAMBAR

Gambar 2.1 Kondisi Kantuk.....	10
Gambar 2.2 Ilustrasi algoritma YOLO.....	15
Gambar 2.3 Arsitektur YOLOv8	17
Gambar 3.1 Tahapan Proses Penelitian	26
Gambar 3.2 Kode resize image	27
Gambar 3.3 Kode Pelatihan Menggunakan Parameter	28
Gambar 3.4 Kode Konversi ke Web.....	29
Gambar 3.5 Rancangan Tampilan Awal	31
Gambar 3.6 Rancangan Tampilan Keterangan.....	31
Gambar 3.7 Rancangan Tampilan Allow Access Camera	32
Gambar 3.8 Rancangan Tampilan Deteksi	33
Gambar 3.9 Rancangan Tampilan Warning.....	34
Gambar 3.10 Rancangan Tampilan Akses Kamera Diblokir	35
Gambar 4.1 Nilai Presisi dan Recall	36
Gambar 4.2 Nilai F1-Score	36
Gambar 4.3 Hasil Training	37
Gambar 4.4 Grafik F1 Confidence Curve	38
Gambar 4.5 Confusion Matrix Normalized.....	39
Gambar 4.6 Precision Confidence Curve	40
Gambar 4.7 Recall Confidence Curve.....	41
Gambar 4.8 Tampilan Awal	42
Gambar 4.9 Tampilan Keterangan.....	43
Gambar 4.10 Tampilan Allow Access Camera.....	44
Gambar 4.11 Tampilan Deteksi	44
Gambar 4.12 Tampilan saat pengguna terdeteksi kantuk.....	45
Gambar 4.13 Tampilan Warning	46
Gambar 4.14 Tampilan Akses Kamera Diblokir	47

DAFTAR LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup

Lampiran 2 Surat Keputusan Dosen Pembimbing Skripsi

Lampiran 3 Kartu Bimbingan Dosen I

Lampiran 4 Kartu Bimbingan Dosen II

BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap orang bisa saja merasa kantuk jika tidak beristirahat dengan cukup dan hal tersebut dapat mengganggu aktivitas yang akan dilakukan. Tentunya itu akan membahayakan jika seseorang kantuk saat mengemudi mobil hingga sepeda motor. Kondisi mengantuk seseorang antara lain dapat dilihat dari kondisi kelopak mata mulai berat, pandangan kabur serta kepala mulai tidak seimbang menahan beban sehingga mengharuskan untuk berbaring dan beristirahat [1].

Banyak pengemudi yang lebih memilih mengabaikan rasa kantuknya dan meneruskan perjalanannya hingga sampai ke tujuan dikarenakan hal yang mendesak, padahal hal ini dapat mengancam keselamatan pengemudi. Beberapa tahun terakhir ini, konsentrasi yang kurang saat sedang mengemudi menjadi salah satu penyebab terjadinya banyak kecelakaan [2]. Berdasarkan data Badan Pusat Statistik (2022), terjadi 109.215 kecelakaan pada tahun tersebut. Faktor utama penyebab meningkatnya kecelakaan lalu lintas adalah faktor manusia, dimana 69,7% kecelakaan disebabkan oleh perilaku manusia. Contoh faktor manusia adalah kelelahan saat berkendara. Lebih dari 25% penyebab kecelakaan adalah kelelahan, yang membuat pengemudi mengantuk saat mengemudi [3].

Dalam penelitian sebelumnya membahas tentang “Sistem Deteksi Kantuk Pada Pengendara Roda Empat Menggunakan *Eye Blink Detection*”. Pada penelitian tersebut pendeteksian dengan ditandai dengan melihat kondisi mata apakah tertutup atau terbuka serta durasi kedipan mata [1]. Dan juga pada penelitian sebelumnya yang lain membahas tentang “Deteksi Kantuk Pada Pengemudi Mobil Menggunakan *Eye Aspect Ratio* Dengan Metode *Facial Landmark*”. Pada penelitian tersebut menggunakan metode *Haarcascade* untuk mendeteksi apakah ada wajah atau tidak, kemudian mendeteksi mata pada wajah, proses selanjutnya mendeteksi bukaan mata menggunakan *eye aspect ratio* [3]. Dari penelitian sebelumnya, penggunaan metode *Haarcascade* serta *Eye Blink Detection* cenderung lebih mendeteksi pada suatu kelas objek tertentu seperti berfokus hanya pada mata. Sedangkan pada penelitian ini menggunakan *YOLO* karena secara kecepatan, akurasi dan fleksibilitas dalam deteksi objek yang kompleks atau real-time. *You only look once (YOLO)* adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi skor paling tinggi akan dianggap sebagai sebuah pendeteksian [4].

Berdasarkan uraian latar belakang di atas, maka akan disusun skripsi dengan judul **“Perancangan Aplikasi Web Pendeteksi Kantuk Menggunakan Algoritma *YOLO*”** dengan pertimbangan bahwa algoritma *YOLO* karena metode tersebut memiliki kecepatan dan akurasi yang baik dalam melakukan deteksi objek [5].

1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan, maka permasalahan yang dihadapi:

1. Bagaimana menghitung akurasi, presisi, recall dan F1-Score untuk mengukur kebagusan model?
2. Bagaimana merancang sebuah aplikasi dengan mengimplementasikan algoritma *YOLO*, sehingga dapat mendeteksi secara real-time?
3. Bagaimana memilih dataset yang dapat digunakan dalam pelatihan dan pengujian terhadap *YOLO*?

1.3 Batasan Masalah

Berikut ini adalah batasan masalah pada penelitian adalah sebagai berikut:

1. *Dataset* yang digunakan diperoleh dari Roboflow https://universe.roboflow.com/custom-yolo/my_data_sleep
2. Parameter yang digunakan adalah epochs, *batch size*, *optimizer*, *ke* dan *learning rate*.

1.4 Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang sebuah aplikasi untuk mendeteksi kantuk yang membantu pengguna dalam pencegahan terjadinya kecelakaan.

2. Merancang sebuah aplikasi dengan mengimplementasi algoritma *YOLO* untuk mengambil gambar secara real-time.

Manfaat dari penelitian ini adalah:

1. Pengguna dapat mendeteksi adanya kantuk.
2. Pengguna terutama pengemudi dapat mencegah terjadinya kecelakaan.

1.5 Metodologi Penelitian

Dalam penelitian ini, metodologi penelitian yang digunakan dalam penelitian:

1. Jadwal dan Tempat Penelitian

Penelitian berisikan jadwal penelitian dari awal hingga penelitian selesai.

Tabel 1.1 Jadwal Penelitian

No	Waktu Keterangan	2023								2024											
		November				Desember				Januari				Febuari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Identifikasi Masalah	■	■																		
2	Pengumpulan Data			■	■																
3	Analisa Sistem					■	■	■	■												
4	Perancangan Sistem								■	■	■										
5	Pembangunan Sistem									■	■	■	■	■	■	■	■				
6	Uji Coba Sistem																	■	■		
7	Penulisan Laporan Skripsi																			■	■

2. Bahan dan Alat

Untuk menjalankan aplikasi, bahan dan alat yang digunakan adalah *hardware* dan *software* dalam proses penelitian.

a. Perangkat keras (*hardware*)

Dalam merancang dan menguji sistem, *hardware* yang digunakan adalah:

1. *Processor* 11th Gen Intel(R) Core(TM) i5-11400H @ 2.70GHz
2.69 GHz
2. RAM 16 GB
3. Perangkat *Mouse*
4. Perangkat *keyboard*
5. Kamera *smartphone* atau komputer yang memiliki kamera

b. Perangkat lunak (*software*) yang digunakan adalah:

Dalam merancang dan menguji sistem, *software* yang digunakan adalah:

1. *Visual Studio Code*
2. *TensorFlow*
3. *Chrome*
4. *Google Colab*
5. *Figma*

3. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Metode Deskriptif

Pengumpulan data yang berhubungan dengan kantung.

b. Metode Studi Pustaka

Pengumpulan data dengan mencari sumber pustaka melalui jurnal, media cetak, perpustakaan maupun melalui sumber referensi internet.

1.6 Analisis Sistem

Analisis sistem pada penelitian ini mencakup:

1. Proses *Training*, pelatihan dan pengujian model kantuk menggunakan *Google Colab* dengan menggunakan algoritma *YOLO* dan kemudian di *export* menggunakan *TensorFlow*. Selama *training* model terdapat 2 *classes* yaitu *drowsy* dan *non drowsy*.
2. Proses validasi dan pengujian untuk memastikan bahwa deteksi kantuk berfungsi dengan baik dan pengujian memastikan tingkat akurasi yang cukup tinggi.
3. Pengembangan aplikasi web menggunakan *Visual Studio Code* dengan mengimplementasikan model menggunakan *library TensorFlow.js* untuk mengoperasikan model deteksi kantuk melalui perangkat komputer atau *smartphone* yang memiliki kamera

1.7 Perancangan Sistem

Perancangan sistem pada penelitian ini terbagi menjadi 3 tahapan yaitu:

1. Perancangan *prototype* tampilan dengan menggunakan software Figma.
2. Pelatihan model pada *machine learning* menggunakan *Google Colab* dan *TensorFlow*.

3. Proses pengembangan aplikasi web menggunakan bahasa pemrograman HTML, CSS dan JavaScript untuk menggunakan model yang dihasilkan sehingga dapat mendeteksi kantuk.

1.8 Sistematika Penulisan

Untuk mempermudah memahami permasalahan yang dibahas, maka skripsi ini akan disusun dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

Dalam Bab I berisikan latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, analisis sistem, perancangan sistem serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam Bab II membahas tinjauan pustaka yang mendukung pembahasan, dan referensi teori yang diperoleh dari berbagai sumber yang berhubungan dengan pembuatan aplikasi.

BAB III METODE PENELITIAN

Dalam Bab III membahas algoritma *YOLO* dalam deteksi kantuk dan perancangan yang akan dibangun.

BAB IV HASIL DAN PEMBAHASAN

Dalam Bab IV berisikan tahapan dalam pengembangan aplikasi, sistem, hasil *testing* dan implementasi serta kelebihan dan juga kekurangan dari aplikasi.

BAB V KESIMPULAN DAN SARAN

Dalam Bab V berisikan kesimpulan dan saran yang diuraikan dari hasil pengujian sistem yang dilakukan.

BAB II

TINJAUAN PUSTAKA

2.1 Kantuk

Kantuk (*Drowsiness*) adalah kondisi Dimana seseorang atau individu membutuh tidur. Kondisi ini sering di alami pada malam hari ataupun siang hari dan merupakan hal yang wajar. Rasa kantuk pada umumnya disebabkan karena kurangnya waktu tidur yang cukup [3]. Kondisi mengantuk terdapat beberapa macam sehingga dapat dikategorikan seseorang sedang mengantuk atau tidak. Kondisi mengantuk seseorang antara lain dapat dilihat dari kondisi kelopak mata mulai berat, pandangan kabur serta kepala mulai tidak seimbang menahan beban sehingga mengharuskan untuk berbaring dan istirahat [1].

Banyak sekali penyebab yang dapat ditimbulkan dari kelelahan. Pada lingkungan pendidikan, mahasiswa kurang dapat fokus pada pembelajaran yang sedang berlangsung dikarenakan kelelahan. Pada lingkungan pekerjaan, misalnya tambang, pekerja dapat mengalami kecelakaan kerja dikarenakan kelelahan yang menyebabkan pekerja tersebut kurang berhati-hati. Pada lalu lintas, menurut data Kepolisian di Indonesia, 61% kecelakaan disebabkan oleh *human error*. [15]



Gambar 2.1 Kondisi Kantuk

2.2 Web

Website atau biasanya disingkat dengan *web* merupakan sebutan untuk sekumpulan halaman *web* di dalam jaringan internet [6]. Website atau disingkat dengan *web*, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. Website adalah suatu sistem informasi berupa teks, gambar, suara dan lain-lain, yang disimpan pada web server internet untuk ditampilkan dalam format hypertext [7].

Teknologi *web* atau dikenal dengan *world wide web (www)* dibangun menggunakan bahasa pemrograman HTML (*Hypertext Markup Language*), PHP (*Hypertext Preprocessor*), CSS (*Cascading Style Sheets*), JavaScript dan lain-lain. Dokumen-dokumen dalam website dapat dilihat oleh pengguna menggunakan aplikasi *web browser* seperti *mozilla firefox*, *opera*, *microsoft edge*, *chrome* dan aplikasi *web browser* lainnya [6].

2.3 JavaScript

JavaScript merupakan bahasa pemrograman tingkat tinggi yang diperkenalkan pertama kali tahun 1995 untuk lingkungan pengembangan aplikasi berbasis web dinamis di sisi *client*, yang memungkinkan pengembang untuk mengembangkan website dengan tampilan menarik. JavaScript mengalami peningkatan kemampuan yang pesat, antara lain berupa: dukungan terhadap pemrograman berorientasi objek, pemrograman fungsional, structural, prosedural, *event-driven*, prototyping, serta kemudahan pengembangan aplikasi di sisi server.

JavaScript framework terdiri atas dua jenis, yaitu: FrontEnd JavaScript framework dan Back-End JavaScript framework. Front-End JavaScript framework menyediakan dukungan pengembangan tampilan dan tatap muka website yang interaktif dan menarik untuk memudahkan pengguna di dalam menggunakan layanan berbasis web sekaligus meningkatkan user experience, sedangkan Back-End JavaScript framework berkaitan dengan internal sistem, dalam bentuk dukungan terhadap pemrosesan data, parsing data, suplai data, penanganan request, routing, dan validasi data [11].

2.4 Python

Python adalah *scripting language* yang berorientasi objek. Bahasa pemrograman ini dapat digunakan untuk pengembangan perangkat lunak dan bisa dijalankan melalui berbagai sistem operasi. Saat ini, Python juga merupakan bahasa yang populer bagi bidang *data science* dan analisis. Hal ini dikarenakan oleh dukungan bahasa Python terhadap *library-library* yang didalamnya menyediakan

fungsi analisis data dan fungsi *machine learning*, *data preprocessing tools*, serta visualisasi data [12].

Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan *Guido Monty Python's Flying Circus*. Van Rossum mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak intuitif dan memiliki pustaka yang luas [13].

2.5 Google Colab

Google Colab adalah sebuah IDE untuk pemrograman Python Dimana pemrosesan akan dilakukan oleh server Google yang memiliki perangkat keras dengan performa yang tinggi. Dari sisi perangkat lunak, Google Colab telah menyediakan hampir Sebagian besar pustaka (*library*) yang dibutuhkan. Dari sisi perangkat keras, Google Colab menyediakan layanan berupa media penyimpanan yang terintegrasi dengan Google Drive, prosesor yang berupa CPU, GPU, dan TPU, serta RAM. Dengan jaminan kemampuan servernya yang stabil hampir keseluruhan pemrosesan tidak menemukan kendala dengan Google Colab selama koneksi jaringan internet lancar [10].

2.6 Deep Learning

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lebih banyak lapisan maka

model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *machine learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi

Prinsip ini terus berkembang hingga *deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer Vision*, *Speech Recognition* dan *Natural Language Processing*. *Deep Learning* merupakan salah satu teknik dalam *machine learning* yang memiliki arsitektur yang lebih mendalam disbanding dengan teknik *machine learning* lainnya dalam menyelesaikan masalah prediksi maupun klasifikasi. Arsitektur umum *deep learning* adalah *Deep Neural Network* (DNN), *Deep Belief Network* (DBN), *Deep Convolutional Network* (DCN) [8].

2.7 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah salah satu jenis Neural Network (NN) yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*. Secara garis besar, CNN tidak jauh beda dengan Neural Network biasanya. Yang membedakan antara CNN dengan NN ialah arsitektur dari CNN dibagi menjadi dua bagian besar, yaitu *Feature Extraction Layer* dan *Fully Connected Layer*

a. *Feature Extraction Layer*

Proses yang terjadi pada bagian ini adalah melakukan “*encoding*” dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan

image tersebut (*Feature Extraction*). *Feature Extraction Layer* terdiri dari dua bagian yaitu *Convolutional Layer* dan *Pooling*.

b. *Fully Connected Layer*

Activation map yang dihasilkan dari *feature extraction layer* masih berbentuk multidimensional array, sehingga harus melakukan *reshape activation map* menjadi sebuah vektor agar bisa digunakan sebagai input dari *fully connected map*. *Layer* ini memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*. *Layer* ini adalah *layer* yang biasanya digunakan dalam penerapan *multi layer perceptron* dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap neuron pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversible, sedangkan *fully connected layer* hanya dapat diimplementasikan di akhir jaringan [9].

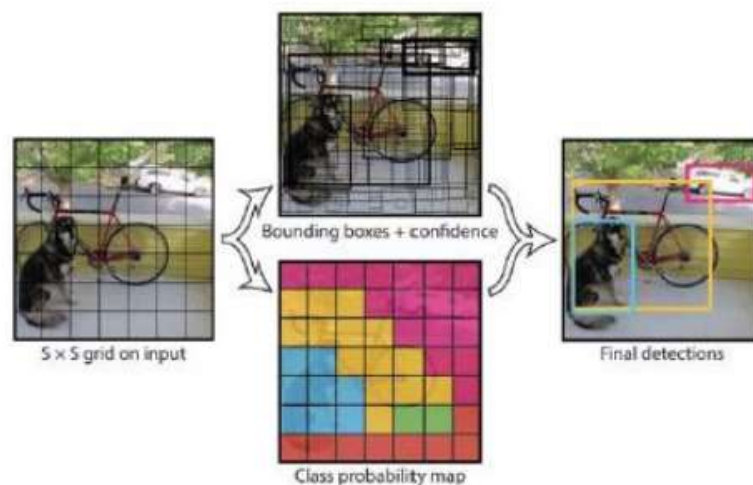
2.8 *You Only Look Once*

You Only Look Once (YOLO) adalah pengembangan dari metode *Convolutional Neural Network (CNN)* menjadi sebuah algoritma baru dan berfokus pada deteksi objek dengan menggunakan ulang atau kembali hasil pengklasifikasi untuk menjalankan fitur deteksi. Arsitektur terpadu *YOLO* sangat cepat dan mampu memproses gambar secara real-time dengan 45 fps (*frame per seconds*) [15]. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra

di beberapa lokasi dan skala. Daerah dengan citra yang diberi skor paling tinggi akan dianggap sebagai sebuah pendeteksian [4].

YOLO membagi citra masukan ke dalam suatu *grid* berukuran $S \times S$. Ukuran dari *grid cell* tersebut tergantung pada input *size* yang digunakan pada suatu arsitektur. Setiap sel bertugas untuk memprediksi objek yang ada di dalam sel tersebut dengan *bounding box* berserta *confidence* yang merupakan nilai probabilitas keberadaan suatu objek pada *bounding box* tersebut.

Kemudian, setelah *bounding box* dipetakan berdasarkan nilai *confidence* yang dihasilkan, *YOLO* akan memprediksi kelas dari objek yang terdapat pada *bounding box* tersebut beserta kemungkinannya, sehingga terbentuklah *class probability map*. Dari sekian banyak *bounding box* yang dihasilkan, untuk mendapatkan *bounding box* beserta kelas objek dengan probabilitas yang tinggi, maka dari seluruh hasil prediksi tersebut, hanya yang melampaui *threshold* saja yang akan digunakan. Jika terdapat duplikasi *bounding box*, maka *Non-max Suppresion (NMS)* berperan untuk menghilangkan duplikat tersebut. Ilustrasi algoritma *YOLO* digambarkan pada Gambar 2.2 [5].



Gambar 2.2 Ilustrasi algoritma *YOLO* [5]

Pada penelitian ini menggunakan YOLOv8 yang memperkenalkan pengoptimalan tambahan dan modul baru untuk meningkatkan akurasi dan efisiensi. Arsitektur YOLOv8 adalah evolusi dari YOLOv7 yang memperkenalkan beberapa perbaikan dan kelebihan baru. Salah satu kelebihan utama YOLOv8 adalah peningkatan performa dan akurasi dalam deteksi objek. Baik dalam performa, akurasi, dan kapasitas deteksi objek. Berikut adalah beberapa keunggulan metode YOLOv8:

1. Performa tinggi

YOLOv8 memiliki performa yang lebih baik dibandingkan dengan versi sebelumnya, dengan deteksi objek yang lebih cepat dan akurat.

2. Model yang lebih kecil

Model YOLOv8 memiliki ukuran yang lebih kecil dibandingkan versi sebelumnya, sehingga lebih mudah untuk digunakan dan diimplementasikan.

3. Ekstraksi fitur yang lebih baik

YOLOv8 menggunakan metode ekstraksi fitur yang lebih baik dibandingkan versi sebelumnya, sehingga menghasilkan deteksi objek yang lebih akurat.

4. Kemampuan multi-skala

YOLOv8 memiliki kemampuan multi-skala, sehingga dapat menangani objek dengan ukuran berbeda pada gambar yang sama.

5. Kemampuan deteksi banyak objek

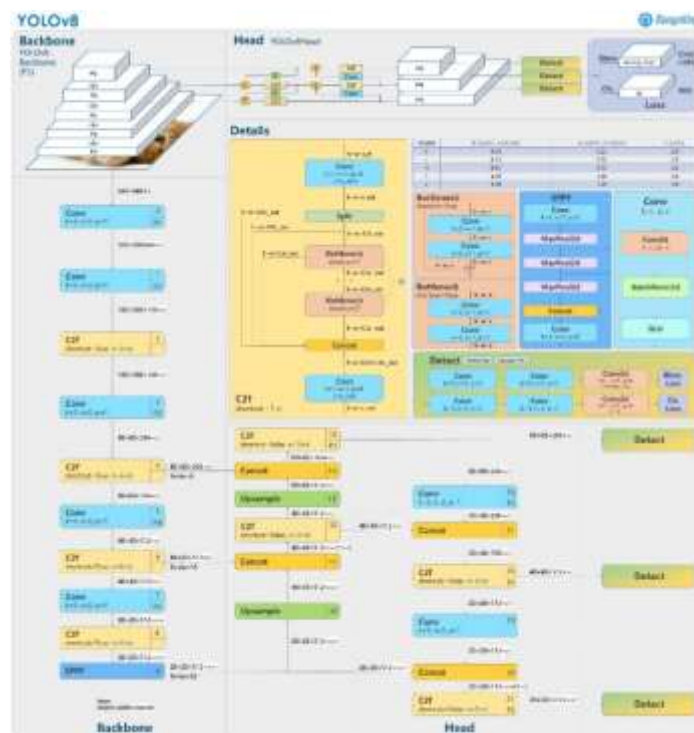
YOLOv8 dapat melakukan deteksi banyak objek pada gambar yang sama, mempermudah proses deteksi objek.

6. Kemampuan deteksi objek pada gambar berukuran besar

YOLOv8 dapat melakukan deteksi objek pada gambar berukuran besar dengan akurasi yang baik, mempermudah proses deteksi objek [16] .

2.9 Arsitektur YOLOv8

Arsitektur YOLOv8 terdiri dari jaringan tulang punggung (*backbone network*), leher (*neck*), dan kepala (*head*). Jaringan tulang punggung menggunakan *Feature Pyramid Network* (FPN) untuk mengekstraksi fitur dari gambar input, sedangkan leher menggunakan serangkaian *Cross-Layer Connection* (CLC) untuk menyempurnakan fitur ini. Kepala mengambil fitur yang disempurnakan dan memprediksi kotak pembatas, skor keas objek, dan akurasi untuk setiap objek dalam citra. Memiliki 105 *layer*. Gambar arsitektur digambarkan pada Gambar 2.3.



Gambar 2.3 Arsitektur YOLOv8

Dimana model ini memiliki *Anchor Free Detection*. Ini berarti memprediksi langsung pusat objek bukan *offset* dari kotak jangkar (*anchor box*). Anchor box adalah bagian yang terkenal rumit dari model YOLO sebelumnya, karena mereka mungkin bukan distribusi kumpulan data khusus [16].

2.10 *Confusion Matrix*

Confusion Matrix adalah sebuah tabel yang menunjukkan kinerja dari sebuah model klasifikasi yang memiliki data jawaban benar (*supervise*). Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, F-score dan masih banyak lagi variabel yang dapat dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan. Isi dari tabel *confusion matrix* ada 4, yaitu

1. *True Positive (TP)*, kondisi dimana model mengklasifikasikan data sebagai ya (*TRUE*) dan jawaban aktualnya adalah ya (*TRUE*)
2. *True Negative (TN)*, kondisi dimana model mengklasifikasikan data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah tidak (*FALSE*)
3. *False Positive (FP)*, kondisi dimana model mengklasifikasikan data sebagai ya (*TRUE*) dan jawaban aktualnya adalah tidak (*FALSE*)
4. *False Negative (FN)*, kondisi dimana model mengklasifikasikan data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah ya (*TRUE*)

Persamaan yang akan digunakan berdasarkan data dari *confusion matrix* adalah akurasi, presisi, *recall* dan *f-score*. Akurasi merupakan pengukuran seberapa

benar sebuah sistem dapat mengklasifikasi dari keseluruhan. Akurasi dapat dihitung menggunakan persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Presisi merupakan perbandingan jumlah data yang dikategorikan positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan dan yang terklasifikasi secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Presisi dapat dihitung menggunakan persamaan (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall merupakan pengukuran untuk data yang klasifikasi positif yang benar oleh sistem. *Recall* dapat dihitung menggunakan persamaan (3)

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F-Score bertujuan untuk menghitung kombinasi dari presisi dan *recall*. *F-Score* akan menggunakan *harmonic mean* dari presisi dan *recall*. *F-Score* dapat dihitung menggunakan persamaan (4) [14].

$$F - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4)$$

2.11 Penelitian Terdahulu

Berdasarkan observasi yang dilakukan, penulis menemukan beberapa penelitian sebelumnya untuk dijadikan referensi, yaitu:

Tabel 2.1 Penelitian Terdahulu

No.	Penulis	Judul	Keterangan
1	S. Maslikah, R. Alfita dan A. F. Ibadillah [1]	Sistem Deteksi Kantuk Pada Pengendara Roda Empat Menggunakan <i>Eye Blink Detection</i>	Peneliti mengangkat topik ini dengan menggunakan metode <i>Haarcascade</i> serta <i>Eye Blink Detection</i> yang cenderung lebih mendeteksi pada suatu kelas objek tertentu seperti berfokus hanya pada mata. Dimana seseorang teridentifikasi mengantuk ditandai dengan melihat kondisi mata apakah tertutup atau terbuka serta durasi kedipan mata.
2	N. Ramadhani L. Q., E. Suhartono, S.	Deteksi Kantuk Pada Pengemudi Berdasarkan Penginderaan Wajah	Peneliti mengangkat topik ini untuk membangun sistem otomasi deteksi mengantuk

No.	Penulis	Judul	Keterangan
	Aulia dan S. Hadiyoso [2]	Menggunakan PCA dan SVM	pada pengemudi sebagai salah satu tindakan preventif dalam mencegah kecelakaan akibat mengantuk saat berkendara. Sistem yang dibuat pada penelitian ini mampu mengenali ekspresi wajah pengemudi dan mengklasifikasikannya ke dalam 2 kelas yaitu normal dan kantuk dengan menggunakan metode PCA dengan parameter <i>eigenface</i> sebagai ekstraksi ciri dan klasifikasi SVM menggunakan <i>straight One Against One</i> (OAO) dan <i>One Against All</i> (OAA).
3	C. K. U. Ngikku, A. Rabi' dan Subairi [3]	Deteksi Kantuk Pada Pengemudi Mobil Menggunakan <i>Eye Aspect Ratio</i> Dengan	Pada penelitian ini menggunakan metode <i>Haarcascade</i> untuk mendeteksi apakah ada wajah atau tidak, kemudian

No.	Penulis	Judul	Keterangan
		Metode <i>Facial Landmark</i>	mendeteksi mata pada wajah, proses selanjutnya mendeteksi bukaan mata menggunakan <i>eye aspect ratio</i> . Metode <i>eye aspect ratio</i> (EAR) ini sering digunakan untuk menghitung kedipan mata seseorang setiap menitnya. Perhitungan <i>eye aspect ratio</i> (EAR) ini dihitung berdasarkan koordinat mata kiri dan kanan yang terdapat pada facial landmarks.
4	S. N. Himawan, R. Sohiburroyan dan N. B. Nugraha [8]	Deteksi Kantuk Pengemudi Menggunakan <i>Deep Learning</i>	Pada penelitian ini dikembangkan model deep learning untuk mendeteksi kantuk pada pengemudi. Deep learning tidak hanya digunakan untuk mengenali gambar lagi tetapi digunakan untuk mengatasi masalah lainnya, salah satunya

No.	Penulis	Judul	Keterangan
			adalah klasifikasi data temporal. Dalam masalah pengenalan aktivitas manusia, aktivitas manusia bisa dikenali dari data sensor seperti barometer dan akselerometer yang datanya termasuk ke dalam data temporal. Model akan mendeteksi gambar kelopak mata dan kondisi menguap pada pengemudi.
5	H. A. Fauzan and A. Kurniawan [15]	Aplikasi <i>Warning Alert</i> Pendeteksi Kelelahan Ekspresi Wajah Pada Pengemudi Secara Real-Time Menggunakan Metode <i>You Only Look Once</i> Berbasis Website	Peneliti mengangkat topik ini untuk mengurangi dampak negative yang disebabkan oleh kelelahan. Peneliti menggunakan teknologi Darknet untuk membuat model deteksi kelelahan berdasarkan citra ekspresi wajah dengan tingkat keakuratan wajah yang tinggi. Dimana ketika

No.	Penulis	Judul	Keterangan
			model mendeteksi adanya kelelahan pada wajah, maka muncul peringatan dan sirine. Penelitian ini menggunakan YOLOv4 karena pada hasil perbandingan deteksi objek YOLOv4 dengan versi sebelumnya (YOLOv3), Bochkovskiy dkk meningkatkan nilai AP (Average Precision) dan fps dari YOLOv3 sebesar 10% dan 12%

BAB III

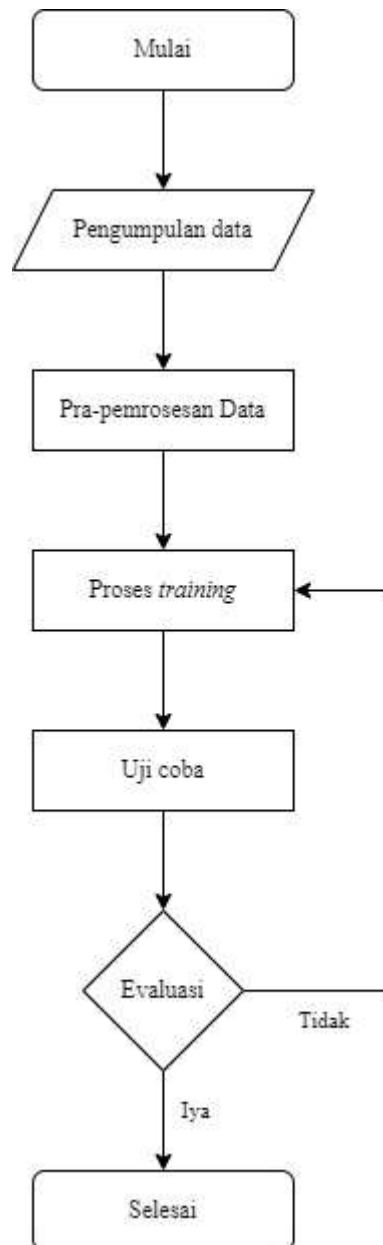
METODE PENELITIAN

3.1 Analisis

Pada tahapan analisis, proses analisis dari *website* untuk mendeteksi kantung dengan menggunakan algoritma YOLO (*You Only Look Once*). Algoritma yang digunakan untuk penelitian adalah algoritma YOLO (*You Only Look Once*) dengan versi YOLOv8. YOLOv8 mengonversi video ke frame citra yang diproses secara terpisah oleh model YOLOv8 kemudian melakukan deteksi objek pada setiap frame, setiap frame citra ini kemudian diproses secara terpisah. YOLOv8 akan mengevaluasi setiap frame dan memberikan prediksi objek dalam frame tersebut, seperti *bounding box* (kotak pembatas) dan label objek dan terakhir menggabungkan hasil deteksi untuk menghasilkan video baru dengan objek yang telah terdeteksi dilengkapi dengan *bounding box* dan label.

3.1.1 Analisis Proses

Penelitian akan dilakukan dengan berbagai tahap yaitu pengumpulan data, pra-pemrosesan data, proses *training*, uji coba dan evaluasi. Tahapan proses YOLO dapat dilihat pada gambar 3.1.



Gambar 3.1 Tahapan Proses Penelitian

Berdasarkan Gambar 3.1, tahapan-tahapan proses pada penelitian ini diuraikan sebagai berikut:

1. Pengumpulan Data

Pada tahap pengumpulan data, data yang dibutuhkan untuk melatih jaringan diambil berupa citra gambar kantuk (*drowsy*) dan tidak kantuk (*non drowsy*). Dataset yang digunakan untuk *training* pada sistem agar dapat digunakan untuk

mendeteksi kantuk diperoleh dari Roboflow yang dengan 2 *classes* yaitu *drowsy* dan *non drowsy*. Jumlah gambar terdapat 3.991 citra gambar.

2. Pra-pemrosesan Data

Pada tahap pra-pemrosesan data, citra pada pengumpulan data memiliki ukuran yang beragam sehingga pada tahap ini dilakukan *resize* dengan memperpanjang atau menyempitkan citra mencapai ukuran yang sama yaitu 256x256 piksel agar mempermudah proses *training* yang akan dilakukan.

resize to 256

```
[ ] from PIL import Image
    import os, sys

[ ] def resize(path, dirs):
    for item in dirs:
        if os.path.isfile(path+item):
            im = Image.open(test_path+item)
            f, e = os.path.splitext(test_path+item)
            imResize = im.resize((256,256))
            imResize.save(f+'.jpg', 'JPEG', quality=100)

[ ] test_path = "/content/my_data_sleep-15/test/images/"
    test_dirs = os.listdir( test_path )
    resize(test_path, test_dirs)

[ ] test_path = "/content/my_data_sleep-15/train/images/"
    test_dirs = os.listdir( test_path )
    resize(test_path, test_dirs)

[ ] test_path = "/content/my_data_sleep-15/valid/images/"
    test_dirs = os.listdir( test_path )
    resize(test_path, test_dirs)
```

Gambar 3.2 Kode *resize image*

3. Proses *Training*

Pada proses *training*, dataset split terdiri dari *train set* 77%, *valid set* 13% dan *test set* 10%. Parameter yang digunakan pada penelitian adalah epochs sebesar 50, 100, dan 200, *batch size* berkisar 10 hingga 64, *optimizer* menggunakan Adam, *learning rate* berkisar 0,001 hingga 0,01 dan *patience* sebesar 20.

```
from ultralytics import YOLO
model = YOLO('yolov8n.yaml')
model.train(data=dataset, location='/data.yaml', batch=10, epochs=100, imgsz=256, optimizer='Adam', lr=0.001, patience=20)
```

Gambar 3.3 Kode Pelatihan Menggunakan Parameter

4. Uji Coba

Setelah proses *training* selesai maka akan dilakukan proses uji coba dimana model deteksi yang telah dilatih akan diuji dengan dataset yang belum pernah dilatih sebelumnya. Proses uji coba ini untuk memastikan model telah dilatih mampu mendeteksi objek dengan akurasi tinggi dan konsisten.

5. Evaluasi

Pada tahap evaluasi, menganalisa kinerja model dengan menggunakan *confusion matrix* dengan minimal akurasi, *precision*, *recall* dan *f1-score* sebesar 85% dalam mengevaluasi untuk deteksi kantuk. Jika tidak mencapai angka tersebut maka akan kembali ke tahap proses *training* dengan pengujian nilai parameter yang berbeda.

3.1.2 Analisis Metode Yang Digunakan

Algoritma yang digunakan untuk penelitian adalah algoritma YOLO (*You Only Look Once*) dengan versi YOLOv8. YOLOv8 mengonversi video ke frame citra yang diproses secara terpisah oleh model YOLOv8 kemudian melakukan deteksi objek pada setiap frame, setiap frame citra ini kemudian diproses secara terpisah. YOLOv8 akan mengevaluasi setiap frame dan memberikan prediksi objek 25 dalam frame tersebut, seperti *bounding box* (kotak pembatas) dan label objek dan terakhir menggabungkan hasil deteksi untuk menghasilkan video baru dengan objek yang telah terdeteksi dilengkapi dengan *bounding box* dan label.

Pada tahap penelitian menggunakan proses pelatihan *Transfer Learning* dengan dataset yang telah diberi label serta *pretrained weight*. Sebelum proses pelatihan, akan di-*compile* sebuah *framework* pelatihan yang bernama “Darknet” dan ditentukan beberapa nilai hyperparameter. Proses pelatihan akan dilakukan melalui Google Colab yang menyediakan GPU yang memadai serta dari sisi perangkat lunak, Google Colab telah menyediakan hampir sebagian besar pustaka (*library*) yang dibutuhkan.

3.1.3 Proses Konversi ke Web

Proses ini dilakukan setelah model *machine learning* pada tahapan proses dengan hasil yang memuaskan sehingga dapat *export* ke *web* dengan TensorFlow Js. TensorFlow Js digunakan untuk mengubah model tersebut ke format yang ringan dan dijalankan di lingkungan *web*. Setelah konversi selesai, unduh model tersebut kemudian jalankan pada JavaScript. Kode konversi ke web dapat dilihat pada Gambar 3.4.

```
# Export the model
model.export(format="tfjs", verbose=True)
```

Gambar 3.4 Kode Konversi ke Web

3.1.4 Proses Pengaktifan Kamera

Pengaktifan kamera dibutuhkan untuk melakukan pedeteksian kantung. Pada proses pengaktifan kamera, JavaScript menyediakan API untuk mengakses dan mengontrol perangkat media seperti kamera melalui browser. Untuk mengaktifkan kamera sebelumnya akan memunculkan izin pengaksesan perangkat

kamera dengan menggunakan `getUserMedia()` untuk meminta akses kamera melalui browser. Jika akses diberikan maka *stream* video dari kamera akan ditampilkan dalam sebuah elemen video HTML.

3.2 Perancangan

Pada tahapan perancangan, akan dirancang kerangka tampilan yang akan dibangun.

3.2.1 Perancangan Tampilan

Perancangan tampilan pada *website* deteksi kantuk digambarkan menggunakan *software figma*. Berikut ini beberapa rancangan tampilan aplikasi yang akan dibangun.

1. Rancangan Tampilan Awal

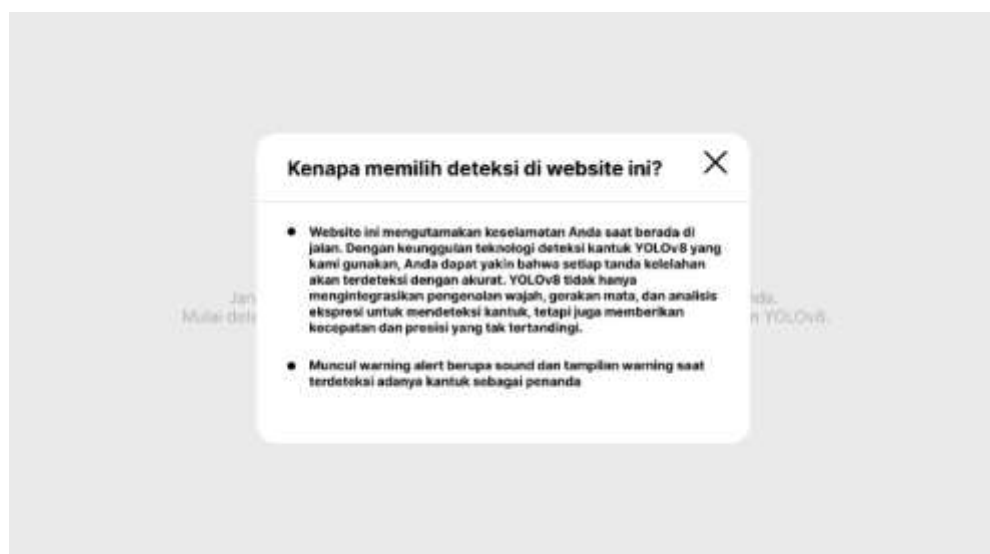
Tampilan awal dari aplikasi dengan menekan tombol ‘Mulai’ maka akan memunculkan tampilan *allow access camera* untuk melakukan proses deteksi. Kemudian, untuk tombol ‘Kenapa mendeteksi kantuk disini?’ jika ditekan maka akan muncul tampilan keterangan yang menjelaskan alasan memilih deteksi di *website* tersebut. Rancangan tampilan awal dapat dilihat pada Gambar 3.5.



Gambar 3.5 Rancangan Tampilan Awal

2. Rancangan Tampilan Keterangan

Tampilan keterangan akan menampilkan alasan *user* menggunakan website untuk mendeteksi kantuk serta keterangan mengenai cara kerja website jika terdeteksi kantuk. Tombol 'x' yang terletak di sudut kanan atas jika ditekan maka akan menutup tampilan keterangan kemudian kembali ke tampilan awal. Rancangan tampilan keterangan dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan Tampilan Keterangan

3. Rancangan Tampilan *Allow Access Camera*

Tampilan *allow access camera* akan muncul setelah pengguna menekan tombol ‘Mulai’ pada tampilan awal. Tombol ‘Allow’ untuk mengizinkan akses kamera untuk melakukan pendeteksian, sedangkan tombol ‘Block’ berarti tidak mengizinkan akses kamera. Rancangan Tampilan *Allow Access Camera* dapat dilihat pada Gambar 3.7.

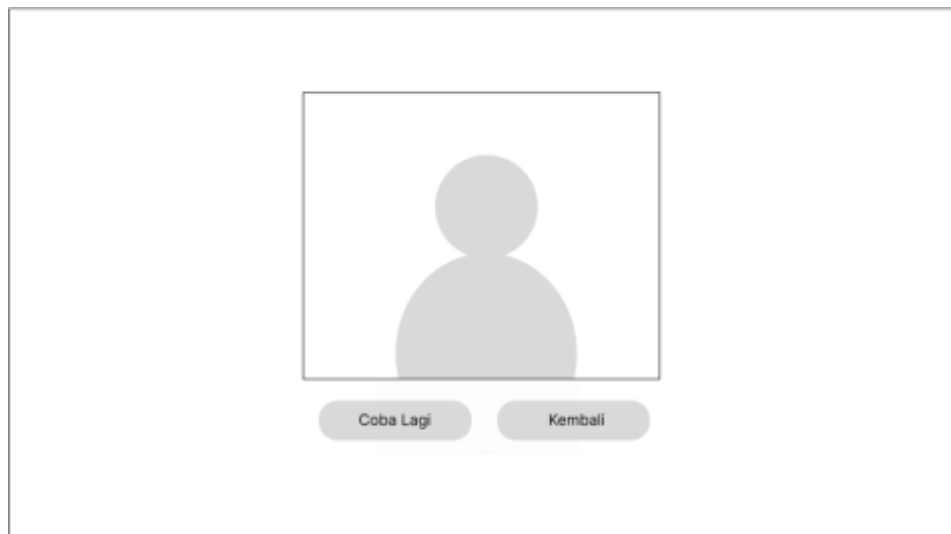


Gambar 3.7 Rancangan Tampilan *Allow Access Camera*

4. Rancangan Tampilan Deteksi

Tampilan deteksi akan muncul setelah *user* menekan tombol ‘*Allow*’ pada tampilan *allow access camera* yang muncul setelah pengguna menekan tombol ‘Mulai’ pada tampilan awal, kemudian setelah pengguna menekan ‘Allow’ pada *allow access camera* maka akan memunculkan tampilan deteksi. Tombol ‘Coba Lagi’ dapat ditekan setelah pengguna terdeteksi kantuk dan memunculkan tampilan *warning*, setelah pengguna menekan

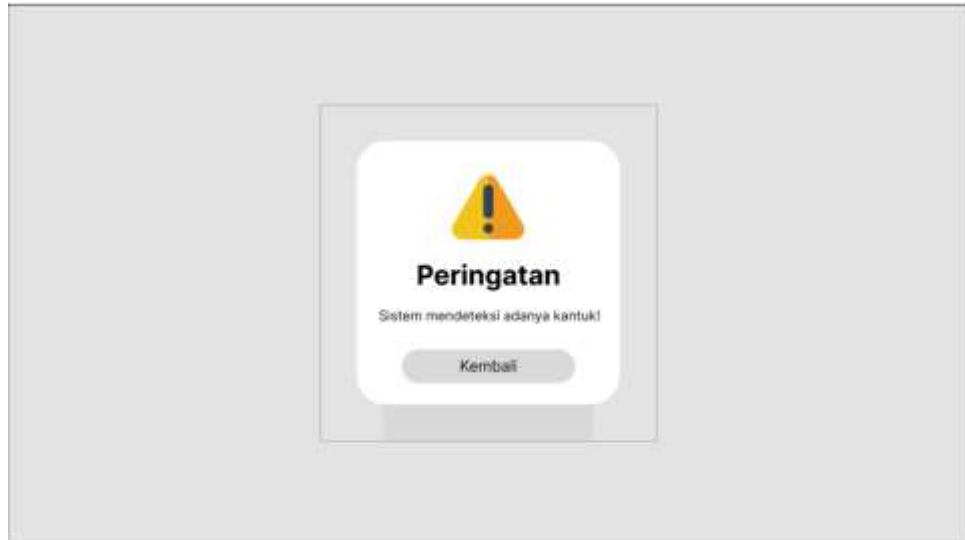
tombol ‘Kembali’ pada tampilan *warning* dan pengguna ingin melakukan deteksi lagi maka harus menekan tombol ‘Coba Lagi’ untuk memulai proses deteksi, sedangkan tombol ‘Kembali’ pada halaman deteksi pada saat ditekan maka akan menutup tampilan deteksi dan kembali ke halaman awal. Rancangan tampilan deteksi dapat dilihat pada Gambar 3.8.



Gambar 3.8 Rancangan Tampilan Deteksi

5. Rancangan Tampilan *Warning*

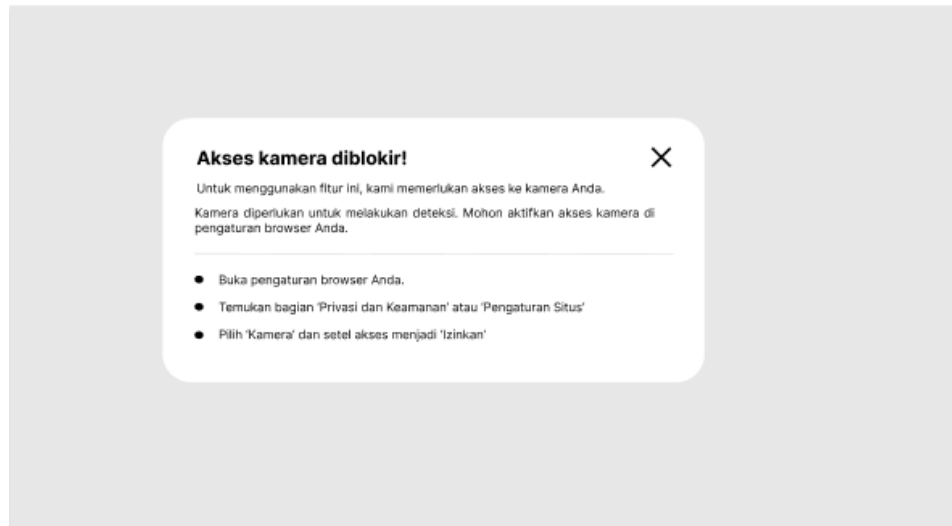
Tampilan *Warning* akan muncul setelah *user* terdeteksi adanya kantuk yang melebihi presentase kantuk 65% dalam waktu 3 detik. Tampilan *warning alert* akan ditampilkan beserta *sound alert* yang akan berhenti berbunyi setelah *user* menekan tombol ‘Kembali’. Rancangan tampilan *warning* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Rancangan Tampilan *Warning*

6. Rancangan Tampilan Akses Kamera Diblokir

Tampilan akses kamera diblokir akan muncul ketika pengguna menekan 'Block' atau 'Blokir' saat *allow access camera*. Tampilan berisikan penjelasan mengenai kegunaan kamera pada website serta cara menyetel agar dapat mengubah menjadi izinkan menggunakan kamera untuk melakukan pendeteksian. Rancangan tampilan akses kamera diblokir dapat dilihat pada Gambar 3.10.



Gambar 3.10 Rancangan Tampilan Akses Kamera Diblokir

BAB IV

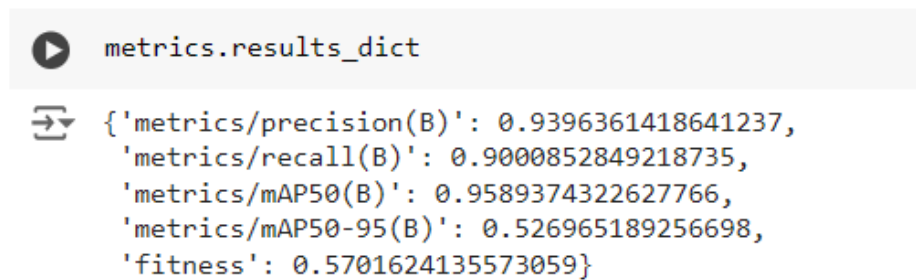
HASIL DAN PEMBAHASAN

4.1 Hasil

Pada sub bab ini, akan membahas mengenai hasil *training* dan hasil tampilan dari rancangan mengenai website deteksi kantuk menggunakan metode Algoritma YOLOv8.

4.1.1 Hasil Pengujian

Pada hasil pengujian, akan ditampilkan *confusion matrix* berupa *precision*, *recall* dan *f1-score*. Pada Gambar 4.1 dan Gambar 4.2 menunjukkan nilai dari *training*.



```
metrics.results_dict
{'metrics/precision(B)': 0.9396361418641237,
 'metrics/recall(B)': 0.9000852849218735,
 'metrics/mAP50(B)': 0.9589374322627766,
 'metrics/mAP50-95(B)': 0.526965189256698,
 'fitness': 0.5701624135573059}
```

Gambar 4.1 Nilai Presisi dan Recall

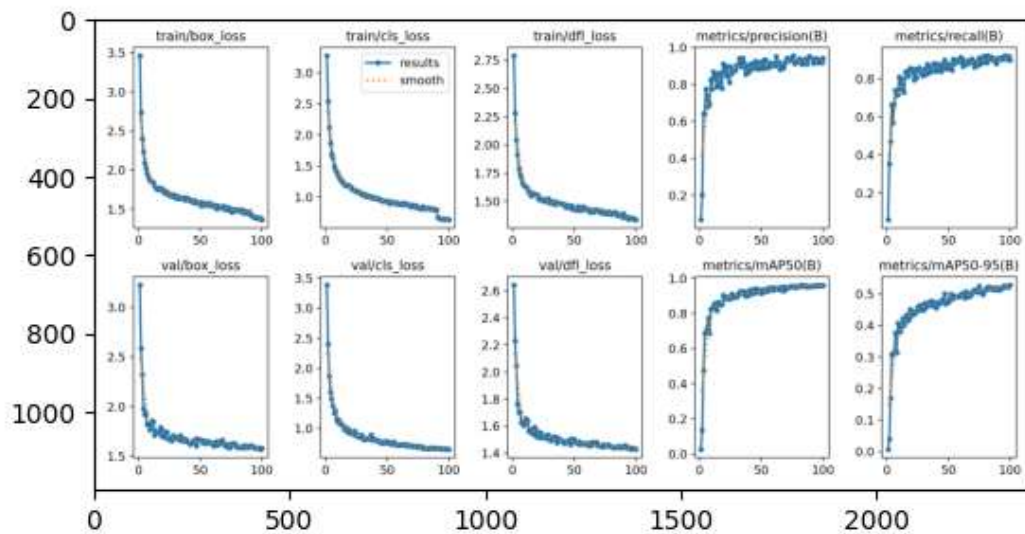


F1 Score

$$F1 = \frac{Precision * Recall}{Precision + Recall}$$

```
f1 = (metrics.results_dict['metrics/precision(B)] * metrics.results_dict['metrics/recall(B)]) / (metrics.results_dict['metrics/precision(B)] + metrics.results_dict['metrics/recall(B)'])
f1
```

Gambar 4.2 Nilai F1-Score



Gambar 4.3 Hasil *Training*

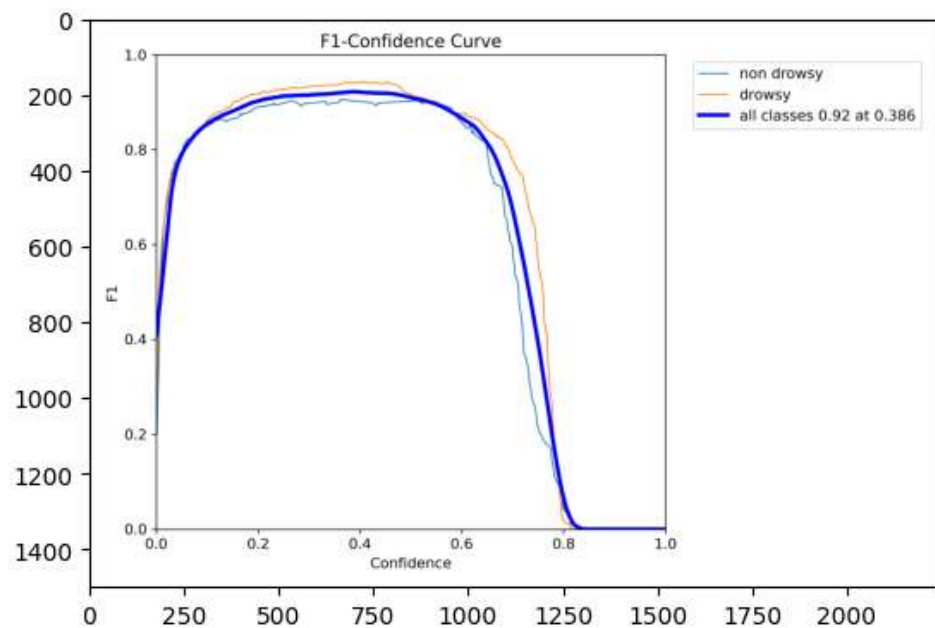
Pada Gambar 4.3 merupakan hasil dari kinerja model. Beragam grafik dan metrik yang menggambarkan kinerja model YOLO selama proses *training* dan validasi. Pada grafik pertama hingga ketiga pada baris atas menunjukkan penurunan "train/box_loss", "train/cls_loss", dan "train/df_l_loss" yang menunjukkan bahwa selama pelatihan, model secara konsisten mengurangi kesalahan prediksi *bounding box*, klasifikasi, dan distribusi fitur. Penurunan yang stabil ini mengindikasikan bahwa model sedang belajar dengan baik dari data pelatihan.

Grafik di baris kedua menunjukkan metrik yang sama tetapi untuk data validasi ("val/box_loss", "val/cls_loss", dan "val/df_l_loss"). Penurunan pada *loss* validasi yang serupa dengan data pelatihan mengindikasikan bahwa model juga berkinerja baik pada data yang tidak dilatih sebelumnya, menunjukkan generalisasi yang baik.

Grafik terakhir di baris atas menunjukkan "metrics/precision(B)" dan "metrics/recall(B)", di mana *precision* dan *recall* mendekati 1, menandakan bahwa

model sangat efektif dalam mendeteksi kantuk dengan tingkat kesalahan yang sangat rendah. Precision yang tinggi menunjukkan bahwa sebagian besar deteksi yang dilakukan oleh model adalah benar, sementara *recall* yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar contoh yang relevan.

Grafik terakhir di baris bawah menunjukkan "metrics/mAP50(B)" dan "metrics/mAP50-95(B)", yang menunjukkan *mean Average Precision* (mAP) pada berbagai *threshold*. Nilai mAP yang tinggi mengindikasikan bahwa model memiliki kinerja yang sangat baik dalam mendeteksi dan mengklasifikasikan objek pada berbagai ukuran dan skala.

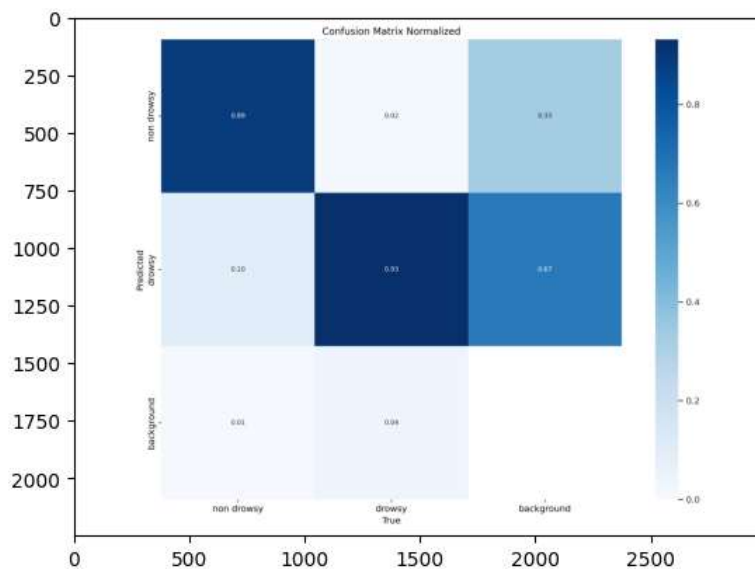


Gambar 4.4 Grafik *F1 Confidence Curve*

F1 confidence curve adalah alat visual yang digunakan untuk mengevaluasi kinerja model klasifikasi dalam konteks ketersediaan dan ketepatan prediksi yang

dibuat oleh model. Kurva ini menggambarkan hubungan antara nilai *F1-score* (yang merupakan harmonik dari presisi dan recall) dengan tingkat keyakinan atau kepercayaan dari prediksi model.

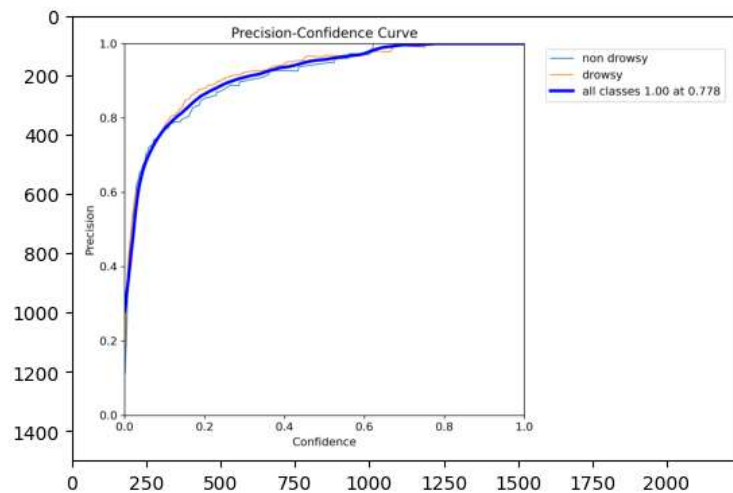
Nilai maksimum F1 untuk semua kelas adalah 0.92 pada tingkat *confidence* 0.386, yang ditunjukkan oleh teks di bagian atas grafik. Grafik *F1 Confidence Score* dapat dilihat pada Gambar 4.4.



Gambar 4.5 *Confusion Matrix Normalized*

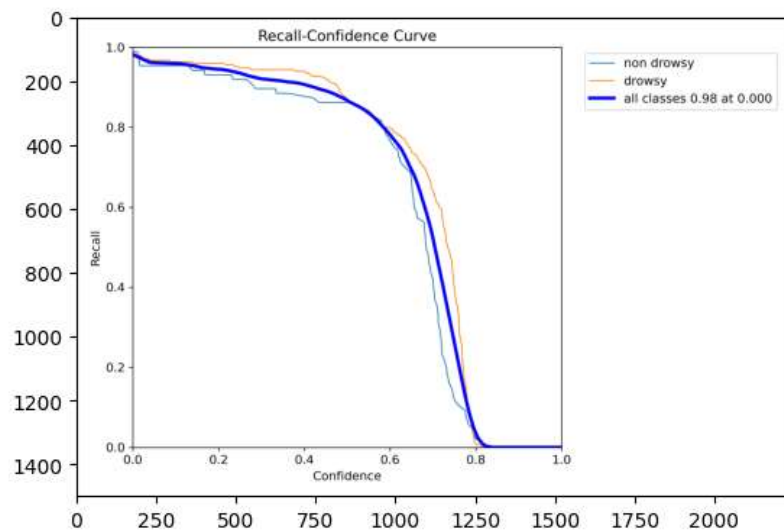
Confusion matrix normalized adalah confusion matrix yang menggambarkan proporsi atau persentase dari setiap kelas yang diklasifikasikan dengan benar atau salah oleh model. Ini memberikan pemahaman yang lebih baik tentang distribusi kesalahan klasifikasi relatif terhadap jumlah data yang ada. Hasil yang diperoleh dengan prediksi *non drowsy* (tidak kantuk) dihasilkan dengan nilai rata-rata akurasi sebesar 89% dan prediksi *drowsy* (kantuk) dihasilkan dengan nilai

rata-rata akurasi sebesar 93%. Confusion matrix normalized dapat dilihat pada Gambar 4.5.



Gambar 4.6 Precision Confidence Curve

Precision confidence curve adalah grafik yang menggambarkan hubungan antara *precision* (ketepatan) dari prediksi model dan tingkat kepercayaan (*confidence*) dari prediksi tersebut. Nilai maksimum *precision* untuk semua kelas adalah 1.00 pada tingkat *confidence* 0.778, yang ditunjukkan oleh teks di bagian atas grafik. Berdasarkan dari grafik yang dihasilkan dapat dikatakan bahwa semakin cepat nilai kelas *drowsy* dan kelas *non drowsy* mencapai puncak tertingginya maka sistem yang dibuat akan semakin baik dalam memprediksi suatu objek. Nilai *Confidence* tersebut menunjukkan seberapa baik sistem tersebut dalam memprediksi suatu objek. Precision confidence curve dapat dilihat pada Gambar 4.6.



Gambar 4.7 Recall Confidence Curve

Recall confidence curve adalah grafik yang menggambarkan hubungan antara *recall* (sensitivitas) dari prediksi model dan tingkat kepercayaan (*confidence*) dari prediksi tersebut. Nilai maksimum *recall* untuk semua kelas adalah 0.98 dan tingkat *confidence* 0.000 Hal ini karena nilai Recall merupakan suatu matrik yang digunakan untuk mengukur suatu hal yang positif., yang ditunjukkan oleh teks di bagian atas grafik. *Recall confidence curve* dapat dilihat pada Gambar 4.7.

4.1.2 Hasil Tampilan

Pada hasil tampilan, akan disajikan hasil yang telah dibangun berupa website deteksi kantuk dengan menggunakan algoritma YOLOv8.

1. Tampilan Awal

Saat pertama kali membuka website maka akan memunculkan tampilan awal seperti terlihat pada Gambar 4.8.



Gambar 4.8 Tampilan Awal

Pengguna dapat menekan tombol ‘Mulai’ untuk memunculkan tampilan *allow access camera* sebelum dialihkan ke tampilan deteksi untuk memulai pendeteksian. Selain itu jika pengguna menekan tombol ‘Kenapa mendeteksi kantung disini?’ untuk mendapatkan keterangan mengenai website.

2. Tampilan Keterangan

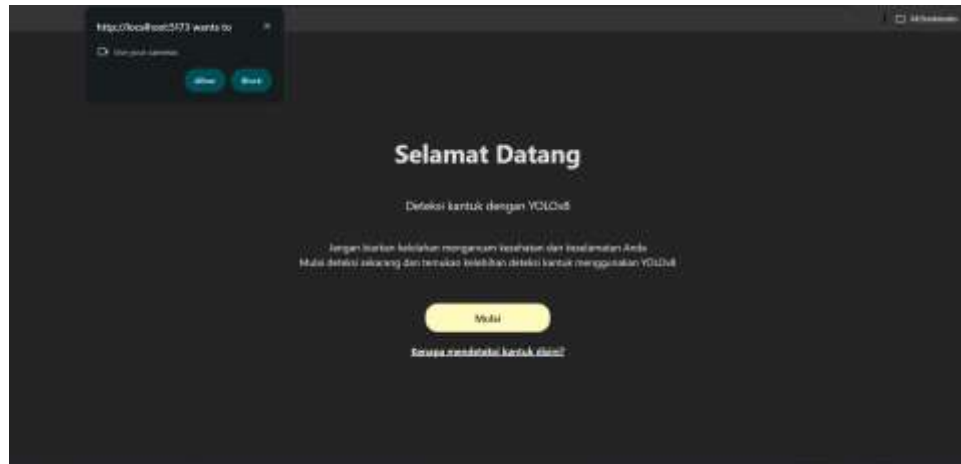
Tampilan keterangan adalah tampilan yang menjelaskan mengenai website yang dibangun. Tampilan ini akan ditampilkan setelah pengguna menekan tombol ‘Kenapa mendeteksi kantung disini?’ pada tampilan awal. Pada tombol ‘x’ di sudut atas sebelah kanan setelah ditekan maka akan menutup tampilan keterangan dan kembali ke halaman awal. Tampilan keterangan dapat dilihat pada Gambar 4.9.



Gambar 4.9 Tampilan Keterangan

3. Tampilan *Allow Access Camera*

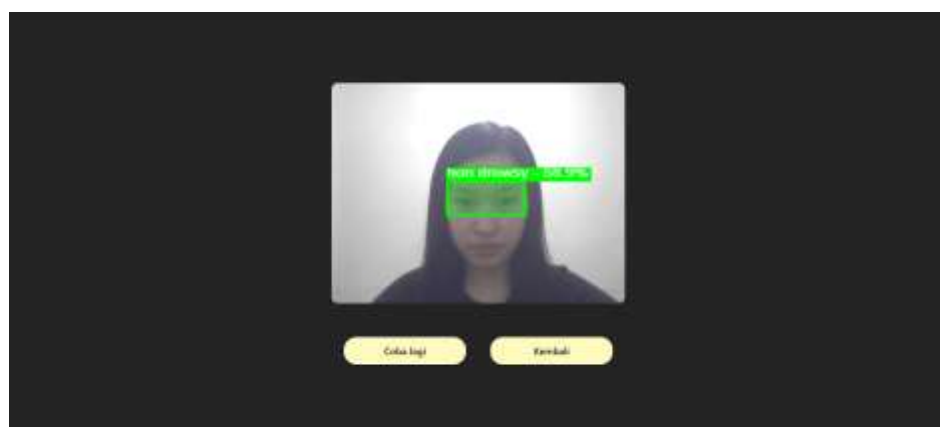
Tampilan *Allow Access Camera* adalah tampilan untuk meminta pengizinan akses kamera untuk melakukan pendeteksian. Tampilan *allow access camera* akan muncul setelah user menekan tombol ‘Mulai’ pada tampilan awal. *Allow Access Camera* untuk mendapatkan izin penggunaan kamera yang akan digunakan dalam deteksi. Tombol ‘Allow’ berarti pengguna mengizinkan akses penggunaan kamera sehingga dapat dialihkan ke tampilan deteksi untuk memulai pendeteksian. Sedangkan tombol ‘Block’ berarti pengguna tidak mengizinkan akses penggunaan kamera dimana akan ditampilkan tampilan akses kamera diblokir. Tampilan *Allow Access Camera* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Tampilan *Allow Access Camera*

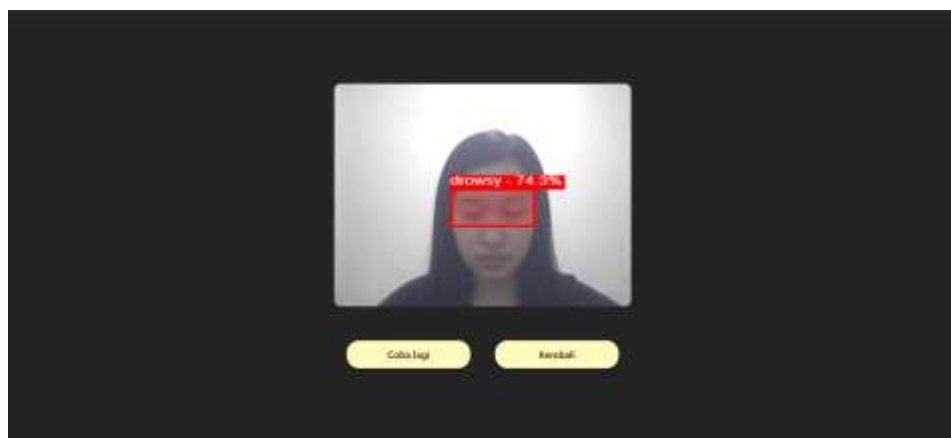
4. Tampilan Deteksi

Tampilan deteksi adalah tampilan untuk melakukan pendeteksian kantuk. Pada Gambar 4.11 berupa tampilan deteksi dimana akan dimulai proses pendeteksian kantuk apabila saat pendeteksian berlangsung tidak adanya tanda-tanda kantuk (*non drowsy*) maka tampilan akan terlihat seperti gambar diatas. Tombol 'Kembali' setelah ditekan maka akan dialihkan ke tampilan awal. Tampilan deteksi dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tampilan Deteksi

Pada Gambar 4.12 berupa tampilan deteksi dimana pengguna terdeteksi kantuk (*drowsy*) namun kurang dari 3 detik sehingga tidak langsung memunculkan tampilan *warning* dan hanya memunculkan kotak merah tanpa memunculkan tampilan peringatan dan *sound alert*. Tombol ‘Coba Lagi’ bisa ditekan setelah pengguna terdeteksi kantuk yang melebihi 65% selama 3 detik dan ingin mencoba kembali deteksi kantuk.



Gambar 4.12 Tampilan saat pengguna terdeteksi kantuk

5. Tampilan *Warning*

Tampilan *warning* adalah tampilan yang menunjukkan adanya peringatan yang terjadi. Tampilan ini muncul jika pengguna terdeteksi adanya kantuk yang melebihi 65% dalam waktu 3 detik maka akan memunculkan tampilan peringatan serta *sound alert* hingga pengguna menekan tombol kembali untuk mematikan *sound alert* serta tampilan peringatan. Tombol ‘Kembali’ setelah ditekan maka akan menutup tampilan *warning* serta *sound alert* akan di stop. Tampilan peringatan dapat dilihat pada Gambar 4.13



Gambar 4.13 Tampilan *Warning*

6. Tampilan Akses Kamera Diblokir

Jika pengguna tidak mengizinkan akses kamera maka akan memunculkan tampilan akses kamera diblokir. Pada tampilan ini menjelaskan tujuan dari permintaan izin akses kamera serta ditampilkan cara agar dapat memberikan izin akses kamera untuk memulai proses deteksi. Tombol 'x' yang ada di sudut kanan atas setelah ditekan maka akan menutup tampilan akses kamera diblokir dan kembali ke halaman awal. Jika pengguna tetap tidak memberikan izin maka setiap pengguna menekan 'Mulai' pada tampilan awal maka akan muncul tampilan akses kamera diblokir. Tampilan akses kamera diblokir dapat dilihat pada Gambar 4.14.



Gambar 4.14 Tampilan Akses Kamera Diblokir

4.2 Pembahasan

Pada sub bab ini, perancangan website deteksi kantuk dengan menerapkan algoritma YOLOv8 yang menggunakan bahasa pemrograman HTML, CSS, JavaScript sedangkan dalam *train model* menggunakan bahasa pemrograman python.

4.2.1 Kelebihan Sistem

Adapun kelebihan dari aplikasi deteksi kantuk yang dibangun antara lain sebagai berikut:

1. Aplikasi menggunakan algoritma *YOLOv8* dalam mendeteksi kantuk yang dikenal dapat mampu mendeteksi objek secara real-time dengan kecepatan yang tinggi.
2. Dapat memunculkan *warning* berupa *sound alert* ketika pengguna terdeteksi kantuk.

4.2.2 Kekurangan Sistem

Adapun kekurangan dari aplikasi deteksi yang dibangun antara lain sebagai berikut:

1. Pada pendeteksian adanya ketergantungan terhadap kualitas kamera yang menurunkan akurasi deteksi.
2. Kondisi pencahayaan dapat memengaruhi akurasi prediksi seperti terlalu gelap ataupun terlalu terang.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan terhadap deteksi kantuk, penulis dapat menarik beberapa kesimpulan yaitu:

1. Dalam mendeteksi objek, *matrix* yang digunakan adalah *precision*, *recall* dan *F1-Score*. *Precision* mengukur seberapa tepat model dalam memprediksi objek sebagai positif, *recall* mengukur seberapa baik model dalam mendeteksi semua objek yang ada, dan *F1-Score* adalah rata-rata harmonis dari *precision* dan *recall*.
2. Penerapan dari algoritma *YOLO* terbukti dapat mendeteksi secara real-time karena algoritma *YOLO* adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time.
3. Pemilihan dataset yang sesuai yaitu kantuk, sehingga dapat melakukan pelatihan dan pengujian model untuk melakukan pendeteksian.

5.2 Saran

Adapun beberapa saran dari penelitian dari deteksi kantuk menggunakan algoritma *YOLO* yaitu:

1. Pada penelitian selanjutnya dapat dikembangkan dengan menggunakan dataset yang lebih besar dengan varian yang lebih banyak agar dapat meningkatkan efektivitas prediksi.

2. Dapat dikembangkan dengan menambahkan fitur *history* untuk pencatatan informasi berupa waktu pengguna terdeteksi kantuk.
3. Dapat menambahkan (fitur yang dipertimbangkan seperti menguap dan juga posisi kepala)

DAFTAR PUSTAKA

- [1] S. Maslikah, R. Alfita dan A. F. Ibadillah, “Sistem Deteksi Kantuk Pada Pengendara Roda Empat Menggunakan Eye Blink Detection,” *Jurnal FORTECH*, vol. 1, no. 1, pp. 123-128, 2020.
- [2] N. Ramadhani L. Q., E. Suhartono, S. Aulia dan S. Hadiyoso, “Deteksi Kantuk Pada Pengemudi Berdasarkan Penginderaan Wajah Menggunakan PCA dan SVM,” *Jurnal Rekayasa Elektrika*, vol. 17, no. 2, pp. 129-136, 2021.
- [3] C. K. U. Ngikku, A. Rabi' dan Subairi, “Deteksi Kantuk Pada Pengemudi Mobil Menggunakan Eye Aspect Ratio Dengan Metode Facial Landmark,” *Seminar Nasional ForteiRegional 7*, vol. 5, no. 1, pp. 72-78, 2022.
- [4] C. Geraldly dan C. Lubis, “PENDETEKSI DAN PENGENALAN JENIS MOBIL MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE DAN CONVOLUTIONAL NEURAL NETWORK,” *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 8, no. 2, pp. 197-199, 2020.
- [5] A. Aprilino dan I. H. A. Amin, “IMPLEMENTASI ALGORITMA YOLO DAN TESSERACT OCR PADA SISTEM DETEKSI PLAT MOBIL OTOMATIS,” *Jurnal TEKNOINFO*, vol. 16, no. 1, pp. 54-59, 2022.
- [6] M. Orisa, A. Faisol dan M. I. Ashari, “PERANCANGAN WEBSITE COMPANY PROFILE MENGGUNAKAN DESIGN SCIENCE RESEARCH METHODOLOGY (DSRM),” *JINTEKS (Jurnal Informatika Teknologi dan Sains)*, vol. 5, no. 1, pp. 160-164, 2023.
- [7] Z. H. Nasution, A. Lubis dan E. Hariyanto, “RANCANG BANGUN SISTEM E-ARSIP BERBASIS WEB MENGGUNAKAN METODE DESIGN THINKING,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 6, no. 1, pp. 60-65, 2023.

- [8] S. N. Himawan, R. Sohiburroyan dan N. B. Nugraha, “Deteksi Kantuk Pengemudi Menggunakan Deep Learning,” *Seminar Nasional Industri dan Teknologi (SNIT)*, 2022.
- [9] A. R. Wasril, M. S. Ghozali dan M. B. Mustafa, “PEMBUATAN PENDETEKSI OBYEK DENGAN METODE YOU ONLY LOOK ONCE (YOLO) UNTUK AUTOMATED TELLER MACHINE (ATM),” *Majalah Ilmiah UNIKOM*, vol. 17, no. 1, pp. 69-76, 2019.
- [10] R. G. Guntara, “Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajaah Menggunakan Algoritma Deep Learning YOLOv7,” *Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55-60, 2023.
- [11] P. A. E. Pratama, “Pengujian Performansi Lima Back-End JavaScript Framework Menggunakan Metode GET dan POST,” *JURNAL RESTI*, vol. 4, no. 6, pp. 1216-1225, 2020.
- [12] D. A. Manali dan G. Gunadi, “IMPLEMENTASI METODE DATA MINING K-MEANS CLUSTERING TERHADAP DATA PEMBAYARAN TRANSAKSI MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON PADA CV DIGITAL DIMENSI,” *INFOTECH: JOURNAL OF TECHNOLOGY INFORMATION*, vol. 8, no. 1, pp. 45-54, 2022.
- [13] M. Romzi dan B. Kurniawan, “PEMBELAJARAN PEMROGRAMAN PYTHON DENGAN PENDEKATAN LOGIKA ALGORITMA,” *JTIM: Jurnal Teknik Informatika Mahakarya*, vol. 3, no. 2, pp. 37-40, 2020.
- [14] K. A. Shianto, K. Gunadi dan E. Setyati, “Deteksi Jenis Model Menggunakan Metode YOLO dan Faster R-CNN,” *JURNAL INFRA*, vol. 7, no. 1, 2019.
- [15] H. A. Fauzan and A. Kurniawan, "Aplikasi Warning Alert Pendeteksi Kelelahan Ekspresi Wajah Pada Pengemudi Secara Real-Time Menggunakan Metode You Only Look Once Berbasis Website," *Jurnal Manajemen Informatika*, vol. 14, no. 1, 2022.
- [16] Yanto, F. Aziz dan Irmawati, “YOLOV8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 3, pp. 1437-1444, 2023.

RIWAYAT HIDUP

Nama : Evelyn Glenisz
Tempat/Tanggal Lahir : Medan, 13 September 2002
Jenis Kelamin : Perempuan
Agama : Buddha
Alamat : Jl m.basir no 36 lingkungan 32
Email : evelynnnglenisz@gmail.com

Pendidikan Formal :
2008-2014 : SD DR. WAHIDIN SUDIROHUSODO
2014-2017 : SMP DR.WAHIDIN SUDIROHUSODO
2017-2020 : SMA DR.WAHIDIN SUDIROHUSODO
2020-Sekarang : STMIK TIME Medan

Medan, Juli 2024



Evelyn Glenisz

SURAT KEPUTUSAN
NO. 070/SK/P1/STMIKTIME/XII/2023

Tentang
**PENGANGKATAN DOSEN PEMBIMBING SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
STMIK TIME**

1. **Membaca** : a. Hasil pemeriksaan berkas dan kelengkapan mahasiswa.
b. Presentasi Proposal Judul Skripsi mahasiswa tertanggal 09 Oktober - 18 November 2023.
2. **Mengingat** : a. Kurikulum Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time Program Studi Teknik Informatika.
b. Penyesuaian teknis pelaksanaan Skripsi Program Studi Teknik Informatika.
3. **Menimbang** : Bahwa skripsi mahasiswa merupakan wujud aplikasi kristalisasi sikap kemampuan dan ketrampilan mahasiswa/i dalam upaya menerapkan ilmu pengetahuan yang telah dipelajari dibangku kuliah, serta skripsi merupakan salah satu mata kuliah yang memiliki bobot 6 SKS, maka dipandang perlu untuk menetapkan Dosen Pembimbing I dan Dosen Pembimbing II Skripsi.

4. **Menetapkan** :
Pertama :

Mengangkat Saudara :

Pembimbing I	Octara Pribadi, M.Kom
Pembimbing II	Robet, M.Kom

Sebagai Dosen Pembimbing Skripsi atas :

2044031	Evelyn Glenisz	Teknik Informatika
---------	----------------	--------------------

Dengan Judul Skripsi :

Perancangan Aplikasi Web Pendeteksi Kantuk Menggunakan Algoritma Yolo

- Mahasiswa Program Strata Satu (S1) Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK)Time.
- Kedua : Apabila terdapat kekeliruan di kemudian hari tentang penetapan Dosen Pembimbing Skripsi ini akan diperbaiki sebagaimana mestinya
- Ketiga : Surat Keputusan ini berlaku sejak tanggal ditetapkan 07 Desember 2023.

Ditetapkan di
Medan, 07 Desember 2023



Hendri.M.Kom
Pembantu Ketua I STMIK TIME



STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : EVELYN GLENISZ
NIM : 2044031
JUDUL SKRIPSI : PERANCANGAN APLIKASI WEB PENDETEKSI KANTUK MENGGUNAKAN ALGORITMA YOLO
NAMA DOSEN : OCTARA PRIBADI, M.KOM
DOSEN PEMBIMBING : ☒ I (SATU) ☐ II (DUA)

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
19 Januari 2024	REVISI BAB I	
8 Maret 2024	BAB I ACC DAN BAB II TAMBAH PENELITIAN TERDAHULU	
24 April 2024	BAB II ACC DAN BAB III REVISI ANALISIS PROSES	
30 April 2024	BAB III ACC	
23 Juli 2024	REVISI BAB IV	
24 Juli 2024	BAB IV DAN BAB V ACC	

Ketentuan :

1. Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
2. Formulir Bimbingan wajib diisi dan ditunjukkan ke dosen pembimbing tiap bimbingan
3. Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
4. Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan, 30 April 2024

Dosen Pembimbing,

Mahasiswa,



OCTARA PRIBADI, M.KOM

EVELYN GLENISZ



STMIK TIME

SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER - TIME

FORMULIR BIMBINGAN SKRIPSI

NAMA : EVELYN GLENISZ
NIM : 2044031
JUDUL SKRIPSI : PERANCANGAN APLIKASI WEB PENDETEKSI KANTUK MENGGUNAKAN ALGORITMA YOLO
NAMA DOSEN : ROBET, M.KOM
DOSEN PEMBIMBING : ☐ I (SATU) ☒ II (DUA)

TANGGAL BIMBINGAN	CATATAN BIMBINGAN	PARAF
21 Maret 2024	ACC BAB I	
17 April 2024	TENTUKAN VARIAN ALGORITMA YANG DIGUNAKAN	
27 April 2024	ACC BAB II	
06 Mei 2024	REVISI BAB III	
14 Mei 2024	ACC BAB III	
24 Juli 2024	REVISI BAB IV	
25 Juli 2024	ACC BAB IV	
26 Juli 2024	ACC BAB V	
29 Juli 2024	ACC PAPER	

Ketentuan :

- Bimbingan dengan dosen pembimbing minimal 8 (delapan) kali
- Formulir Bimbingan wajib diisi dan ditunjukkan ke dosen pembimbing tiap bimbingan
- Formulir Bimbingan Skripsi ini dianggap sah, apabila telah mendapatkan Pengesahan dari dosen pembimbing
- Jika tidak memenuhi ketentuan di atas, skripsi dapat **DIBATALKAN**

Medan, 14 Mei 2024

Dosen Pembimbing,



ROBET, M.KOM

Mahasiswa,

EVELYN GLENISZ