

# Detection of Facial Fatigue in High-Risk Workers Using YOLOv4 [Deteksi Kelelahan Wajah Pada Pekerja Beresiko Tinggi Menggunakan YOLOv4]

Ahmad Rendra Fajaresta<sup>1)</sup>, Rohman Dijaya<sup>\*2)</sup>, Yulian Findawati<sup>3)</sup>, Metatia Intan Mauliana<sup>4)</sup>

<sup>1)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

<sup>2)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

<sup>3)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

<sup>4)</sup>Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

\*Email Penulis Korespondensi: rohman.dijaya@umsida.ac.id

**Abstract.** Work fatigue is a condition where the capacity and focus of workers decrease, affecting their ability to perform tasks. Workers need to be monitored periodically and in real-time so that their fatigue levels can be addressed by supervisors. This study proposes a mechanism for monitoring worker fatigue using real-time digital image processing through a webcam or CCTV. The study utilizes the YOLOv4 framework for facial recognition. The objective is to detect fatigue signs on workers' faces using YOLO. The research comprises several stages, including system analysis, dataset collection, data preprocessing, YOLO network configuration, data training, testing, and implementation. A total of 500 images were used for training, with 2 classes and 4 criteria. The 2 classes consist of normal and fatigue, while the 4 criteria include tired eyes, normal eyes, normal mouth, and yawning mouth. Based on the real-time testing conducted on a webcam, the system successfully detected faces with a best mAP score of 98.30%.

**Keywords** - Fatigue; Face; YOLOv4; Face Detection; Image Processing.

**Abstrak.** Kelelahan kerja merupakan kondisi kemampuan kapasitas dan daya fokus pekerja mengalami penurunan yang mempengaruhi kemampuan pekerja untuk melakukan tugas. Pekerja harus dipantau secara berkala dan realtime sehingga tingkat kelelahan pekerja dapat ditindaklanjuti oleh pengawas. Penelitian ini mengusulkan mekanisme pengawasan kelelahan pekerja menggunakan metode pengolahan citra digital secara realtime melalui Webcam atau CCTV. Penelitian ini menggunakan framework YOLOv4 untuk pengenalan wajah. Tujuan Penelitian ini untuk pendekripsi kelelahan wajah pada pekerja menggunakan YOLO. Penelitian ini terdiri dari beberapa tahap, yaitu analisa sistem, pengumpulan dataset, preproses data, konfigurasi jaringan YOLO, pelatihan data, pengujian, dan implementasi. Terdapat 500 citra yang digunakan dalam pelatihan dengan 2 kelas dan 4 kriteria. 2 kelas yang terdiri dari normal dan kelelahan. Dari setiap kelas terdiri 4 kriteria diantaranya mata lelah, mata normal, mulut normal, mulut menguap. Berdasarkan hasil pengujian yang dilakukan secara realtime pada Webcam, sistem berhasil mendekripsi wajah dengan nilai mAP terbaik sebesar 98.30%.

**Kata Kunci** - Kelelahan; Wajah; YOLOv4; Deteksi Wajah; Pengolahan Citra.

## I. PENDAHULUAN

Kelelahan kerja merupakan suatu kondisi dimana kemampuan dan daya tahan seseorang dalam bekerja mengalami penurunan. Istilah kelelahan mengacu pada kondisi yang mempengaruhi kemampuan tenaga kerja untuk melakukan suatu tugas, yang mengakibatkan berkurangnya kapasitas dan daya fokus kerja[1]. Kelelahan pada seorang pekerja ditandai dengan penurunan fokus kerja akibat pekerjaan yang monoton, tuntutan untuk melakukan pekerjaan dengan cepat, postur kerja yang tidak sesuai, status gizi pekerja tidak normal, status psikologis pekerja, faktor usia, rutinitas sarapan, dan beban kerja yang begitu besar sehingga tubuh pekerja melebihi batas kemampuan untuk bekerja. Oleh karena itu, perlu dilakukan pemulihan melalui istirahat[2].

Faktor kelelahan merupakan faktor yang tidak dapat dihindari karena seringkali para pekerja tidak mengetahui secara pasti apakah tubuhnya dalam keadaan lelah atau tidak dan juga karena tidak adanya sistem peringatan waktu istirahat. Aspek yang dapat menunjukkan kelelahan seorang pekerja adalah ekspresi wajah, karena ketika seseorang sedang lelah maka ekspresi wajahnya sulit untuk dimanipulasi[3]. Kelelahan pekerja dapat diklasifikasikan sebagai kelelahan ringan dengan gejala kelelahan seperti hilang konsentrasi, mengantuk, dan sakit kepala yang tidak dapat dijelaskan, sedangkan kelelahan yang terlihat ditandai dengan mata lelah dan mulut menguap. Oleh karena itu, perlu dilakukan pemantauan kelelahan kerja karyawan secara *realtime* untuk menghindari penurunan kinerja atau resiko kecelakaan kerja[4]. Salah satu solusi yang dapat digunakan untuk mengatasi masalah tersebut adalah dengan menggunakan teknologi pengolahan citra (*Image Processing*) menggunakan metode *YOLO* dan memanfaatkan kamera CCTV atau Webcam yang terpasang di zona inspeksi.

*Image processing* atau pengolahan citra adalah suatu proses komputasi yang melibatkan manipulasi dan analisis gambar untuk memperbaiki, menganalisis, dan mengubah informasi visual dalam gambar dua dimensi[5]. Salah satu

aplikasi penting dari pengolahan citra adalah dalam deteksi tanda-tanda kelelahan pada wajah manusia. Dalam hal ini, pengolahan citra digunakan untuk menganalisa ekspresi wajah, gerakan mata, atau gerakan mulut guna mengidentifikasi tanda-tanda kelelahan[6]. Metode yang banyak digunakan dalam deteksi kelelahan melalui pengolahan citra melibatkan algoritma seperti *YOLO*.

*YOLO* dikembangkan oleh Joseph Redmon[7] pada tahun 2016, adalah algoritma populer dalam bidang *computer vision* untuk pendekripsi objek *realtime*. Pendekatan *YOLO* membagi citra menjadi kisi (*grid*) dan secara simultan menghasilkan prediksi kotak pembatas dan kelas objek. Kecepatan tinggi menjadi keunggulan utama *YOLO* dengan kecepatan 45 *frame* per detik. Seiring dengan perkembangannya, *YOLOv4* diperkenalkan oleh Alexey Bochkovskiy[8] pada tahun 2020 sebagai versi yang ditingkatkan dari algoritma *YOLO*. *YOLOv4* menawarkan peningkatan signifikan dalam performa dan akurasi dengan menggunakan arsitektur *backbone* yang lebih kompleks, teknik augmentasi data yang lebih baik, dan metode pengoptimalan yang ditingkatkan. Selain itu, *YOLOv4* juga mengintegrasikan fitur-fitur canggih seperti pengenalan pola dan deteksi kelelahan.

Pada penelitian sebelumnya terkait pengolahan citra juga telah dilakukan oleh Fudholi[9] yang berjudul deteksi indikasi kelelahan menggunakan *Deep Learning*. Pada penelitian ini deteksi kelelahan dilakukan pada siswa menggunakan metode *ResNet50 CNN* dengan dataset video yang membuat klasifikasi biner dengan dua kategori hasil, mulut diam dan menguap. Didapatkan hasil berlabel 0 atau mulut diam dengan nilai *f1-score* sebesar 0,79 dan berlabel 1 atau mulut terbuka dengan *f1-score* sebesar 0,74. Adapun penelitian lainnya yang dilakukan oleh Setyaningsih[10] yang berjudul *YOLOv4* dan *Mask R-CNN* untuk deteksi kerusakan pada karung komoditi. Pada penelitian ini membandingkan dua metode yaitu *YOLOv4* dan *Mask R-CNN* dengan dataset keadaan karung komoditi yang memiliki dua kelas objek, karung bagus dan karung berlubang. Hasil menunjukkan bahwa model *YOLOv4* dapat bekerja lebih baik dengan akurasi 96,8%, sedangkan model *Mask R-CNN* dengan akurasi 65,78% pada data uji yang sama.

Berdasarkan latar belakang di atas, maka penulis mengusulkan penelitian yang berjudul Deteksi Kelelahan Wajah Pada Pekerja Beresiko Tinggi Menggunakan *YOLOv4*. Penelitian ini akan dirancang memanfaatkan *framework* *YOLOv4* untuk mengidentifikasi wajah dari pekerja yang membutuhkan fokus tinggi, yang nantinya dapat mengurangi resiko kelelahan pada saat bekerja. Rancangan deteksi kelelahan wajah akan diimplementasikan pada kamera Webcam atau jangkauan kamera CCTV. Penelitian ini dilakukan untuk membantu pihak pengawas dalam melakukan pengawasan pada pekerja agar mengurangi beban kerja dan menghindari kecelakaan kerja karena kelelahan.

## II. METODE

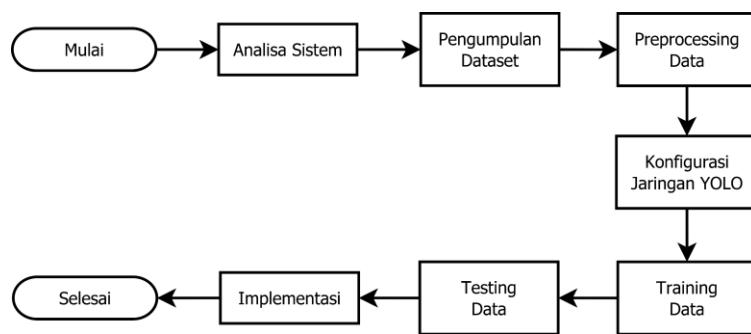
Proses perancangan sistem pendekripsi kelelahan wajah dimulai dengan melakukan analisis sistem yang bertujuan untuk memahami masalah yang dihadapi dan menentukan solusi yang tepat. Selanjutnya, dilakukan pengumpulan dataset yang berkaitan dengan gambar wajah yang menunjukkan tanda-tanda kelelahan. Data kemudian dilakukan *preprocessing* untuk mempersiapkan data masukan sebelum digunakan dalam konfigurasi jaringan *YOLO*. Tahap selanjutnya adalah melakukan pelatihan data pada model yang sudah dikonfigurasi, dan melakukan pengujian terhadap data uji. Setelah model dilatih dan diuji, maka dapat dilakukan implementasi pada sistem pendekripsi kelelahan wajah yang sesuai dengan tujuan dari sistem yang dibangun. Oleh karena itu, tahapan ini harus dilakukan secara sistematis dan cermat agar hasil yang dihasilkan dapat optimal dan sesuai dengan kebutuhan, seperti ditunjukkan dalam Gambar 1. Adapun tahap pelatihan dan pengujian pada perancangan sistem ini dilakukan menggunakan perangkat keras dan perangkat lunak dengan spesifikasi seperti dalam Tabel 1 dan Tabel 2.

### A. Analisa Sistem

Pada tahap awal penelitian deteksi kelelahan wajah pekerja beresiko tinggi menggunakan *YOLOv4* ini, ada beberapa persyaratan yang perlu diperhatikan. Pertama, saat mengumpulkan data citra wajah, diperlukan gambar wajah yang menunjukkan dengan jelas posisi mata dan mulut wajah. Selanjutnya, data citra wajah yang terkumpul akan dibagi menjadi dua bagian. Tujuannya adalah untuk digunakan sebagai data latih dan data pengujian dalam proses mendekripsi wajah serta mengklasifikasikan mata dan mulut yang terdeteksi.

Data latih yang digunakan adalah dataset publik yang tersedia dalam format gambar (.jpg) yang diambil dari sumber terbuka Kaggle. Parameter yang menjadi fokus adalah ekspresi wajah berdasarkan mata dan mulut. Kemudian dataset citra diberi anotasi atau pelabelan citra sesuai dengan format *YOLO* dan disimpan dalam format file teks berekstensi .txt. Anotasi tersebut mengklasifikasikan target ke dalam empat kelas, yaitu mata lelah, mata normal, mulut normal, dan mulut menguap. Dari empat kelas tersebut yang nantinya akan menentukan konfigurasi jaringan *YOLOv4* pada tahap *preprocessing* data.

Setelah proses pengumpulan dan anotasi data selesai, tahap selanjutnya adalah melakukan pelatihan menggunakan *Google Colab*. Dalam tahap ini, *framework* *YOLO* digunakan untuk menentukan lokasi di mana ekspresi tertentu muncul pada citra dan mengklasifikasikan objek tersebut. Secara umum, citra merupakan masukan yang diberikan untuk dilatih algoritma *YOLOv4* dan sebagai outputnya, diperoleh citra yang memiliki vektor kotak pembatas dan prediksi kelas. Melalui tahap pelatihan ini, model belajar untuk mengenali dan memahami berbagai ekspresi wajah yang diinginkan berdasarkan mata dan mulut.

**Gambar 1.** Alur kerja penelitian**Tabel 1.** Spesifikasi Perangkat Keras

No	Aspek	Spesifikasi
1	<i>Processor</i>	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz
2	<i>Memory</i>	4 GB RAM
3	<i>Storage</i>	SSD 512 GB
4	<i>Graphics Card</i>	Nvidia GeForce MX230
5	<i>Operating System</i>	Windows 11 Home Single Language 64-bit

**Tabel 2.** Spesifikasi Perangkat Lunak

No	Aspek	Perangkat Lunak
1	Bahasa Pemrograman	Python versi 3.10.11
2	Software pelatihan dan uji Data	Google Colaboratory
3	<i>Open Source Dataset</i>	Kaggle

Setelah pelatihan selesai, tahap pengujian dilakukan menggunakan 2 skenario yaitu skenario menggunakan citra baru dan skenario secara *realtime* dengan menggunakan model *YOLOv4* yang telah dilatih sebelumnya. Citra yang diuji akan melalui proses deteksi dan klasifikasi menggunakan model *YOLOv4*. Sedangkan secara *realtime*, proses pengujian akan menggunakan Webcam memanfaatkan *Computer Vision*[11]. Hal ini memungkinkan pengguna untuk melihat dan memantau secara *realtime* deteksi ekspresi wajah pekerja yang menggambarkan tingkat kelelahan mereka). Dengan menggunakan pendekatan ini, penelitian ini berhasil mengimplementasikan mekanisme pengawasan kelelahan pekerja secara *realtime* melalui pengolahan citra digital.

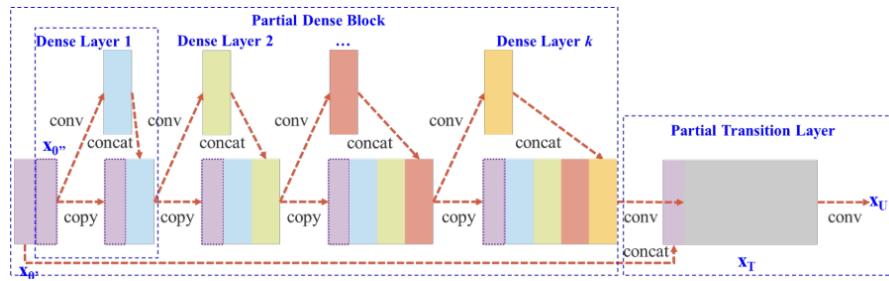
### B. Pengumpulan Dataset

Pada tahap pengumpulan data untuk proses pelatihan, dilakukan dengan memperoleh data citra ekspresi wajah yang terlihat jelas terutama pada area mata dan mulut melalui situs web Kaggle. Dataset yang diambil berjumlah 500 file(tanpa label) dengan format jpg[12]. Dataset dibagi menjadi 2 kategori yang terdiri dari data latih dengan jumlah 450 citra dan data uji dengan jumlah 50 citra. Tujuan dari pengambilan data ini adalah untuk memperoleh dataset yang beragam dan mencakup variasi ekspresi wajah yang berbeda untuk melatih model deteksi kelelahan menggunakan *YOLOv4*. Dengan mengumpulkan data dari sumber yang relevan dan berkualitas, diharapkan dapat membantu dalam mengembangkan model deteksi kelelahan wajah melalui identifikasi mata dan mulut pekerja dengan lebih akurat dan efektif. Tabel 3 pada kolom gambar merupakan contoh gambar dataset yang ada pada situs web Kaggle.

### C. Preprocessing Data

#### Pelabelan Citra

Pada tahap *preprocessing*, langkah awal yang dilakukan adalah memproses data citra dengan memperkecil ukuran berkas setiap file menjadi antara 50 dan 800 kilobyte, namun tetap dalam ukuran piksel yang sesuai. Setelah citra diproses ke ukuran yang lebih kecil, selanjutnya dilakukan anotasi data pada masing-masing citra. Anotasi data merupakan proses pemberian label pada citra untuk menyimpan informasi citra tersebut[13]. Proses anotasi data dilakukan dengan cara melabeli mata dan mulut sesuai kriteria label yang ditentukan, yaitu mata lelah, mata normal, mulut normal, dan mulut menguap. Pelabelan menggunakan *LabelImg* dengan metode *tagging* pada seluruh gambar data latih. Hasil anotasi berupa kotak pembatas (*bounding box*) yang sesuai dengan lebar, tinggi objek, dan nama label dari setiap objek citra. Informasi dari hasil anotasi berupa koordinat kotak pembatas yang merepresentasikan mata dan mulut pada citra dengan format label *YOLO* yang disimpan dalam file berekstensi .txt, seperti contoh Tabel 3.



Gambar 2. Arsitektur CSP[14]

Tabel 3. Anotasi Citra Wajah dan File Ekstensi

No	Gambar	Koordinat box	Target label	
			Biner	Teks Label
1		0.426852 0.414323 0.246296 0.061979	0	Mata lelah
		0.439352 0.514583 0.156481 0.076042	3	Mulut menguap
2		0.489955 0.462838 0.234375 0.077703	2	Mata normal
		0.504464 0.686655 0.129464 0.065878	1	Mulut normal

### Kloning dan membangun *CSPDarknet-53*

*CSPDarknet53* adalah arsitektur jaringan saraf konvolusi yang digunakan dalam *YOLOv4* untuk deteksi objek *realtime* yang efisien dan akurat. Pada *YOLOv4*, *CSPDarknet53* berperan sebagai jaringan dasar yang menggabungkan konsep *Convolutional Spatial Pyramid (CSP)* dengan arsitektur *Darknet-53*[14]. Tujuan dari penggabungan ini adalah untuk meningkatkan performa model dalam hal akurasi deteksi dan efisiensi komputasi, melampaui kemampuan *Darknet-53* sebelumnya. Keunggulan utama *CSPDarknet53* terletak pada efisiensi komputasi yang lebih tinggi dan optimalisasi proses pelatihan serta inferensi yang lebih cepat[15].

Dalam arsitektur *CSPDarknet53*, aliran data dibagi menjadi dua cabang yang berbeda menggunakan konsep *CSP* untuk mempercepat komputasi seperti yang ditunjukkan pada Gambar 2. Terdapat beberapa blok konvolusi bertingkat yang bertujuan untuk mengekstraksi fitur dari gambar input[16]. Setiap blok konvolusi terdiri dari beberapa lapisan konvolusi dengan fungsi aktivasi yang diikuti oleh lapisan pengelompokan (*pooling*) untuk mengurangi dimensi data. Selanjutnya, hasil dari setiap blok konvolusi digabungkan menggunakan metode *Concatenation* dan dilakukan operasi konvolusi kembali[17]. Dengan pendekatan ini, *CSPDarknet53* mampu menghasilkan representasi fitur yang kuat dan menjaga efisiensi komputasi dalam proses deteksi objek.

Pada tahap ini, dilakukan pengambilan kode sumber (*source code*) *CSPDarknet53* dari repositori yang telah disediakan. Selain itu, juga diambil file konfigurasi *yolo.cfg* yang berisi pengaturan arsitektur model *YOLOv4*, serta file *yolo.conv* yang berisi parameter-parameter awal yang digunakan dalam proses pelatihan model. Pengambilan dilakukan melalui metode kloning repositori menggunakan perintah *git clone* atau mengunduh file zip dari repositori yang tersedia. Tujuan dari tahap ini adalah untuk mendapatkan kode sumber dan konfigurasi yang akan digunakan dalam membangun jaringan *CSPDarknet53*.

Setelah berhasil mengkloning repositori, tahap selanjutnya adalah membangun (*build*) arsitektur *CSPDarknet53*. Pada tahap ini, dilakukan kompilasi kode sumber menggunakan alat (*tools*) seperti *CMake* atau *Makefile* yang telah disediakan dalam repositori. Proses pembangunan akan menghasilkan file biner eksekusi yang dapat digunakan untuk melakukan deteksi pada gambar atau video input. Dengan tahap kloning dan pembangunan *CSPDarknet53* yang telah dilakukan, jaringan *CSPDarknet53* siap digunakan dalam tahapan berikutnya pada proses deteksi kelelahan wajah pada pekerja berisiko tinggi menggunakan *YOLOv4*. File konfigurasi *yolo.cfg* dan *yolo.conv* yang diambil akan disesuaikan terlebih dahulu untuk memastikan bahwa model *YOLOv4* yang dibangun memiliki struktur yang sesuai dan parameter awal yang tepat untuk proses pelatihan dataset.

### D. Konfigurasi Jaringan *YOLO*

Pada tahap ini dilakukan konfigurasi *YOLO* pada file *yolo.cfg* yang sudah diambil dari proses kloning *CSPDarknet53* sebelumnya. Dalam penelitian ini, konfigurasi *hyperparameter* pada file *yolo.cfg* disesuaikan, seperti ditunjukkan dalam Tabel 4. *Batch size* sebesar 64 digunakan untuk memproses 64 gambar dalam setiap iterasi pelatihan. Pembagian (*subdivisions*) sebanyak 16 kali dilakukan dalam satu iterasi untuk mengoptimalkan penggunaan memori GPU[18]. Resolusi gambar masukan adalah 416x416 piksel dengan 3 saluran warna (RGB), sementara augmentasi data seperti rotasi dan variasi warna diterapkan untuk memperkaya variasi data pelatihan.

**Tabel 4.** Konfigurasi *Hyperparameter*

No	Hyperparameter	Nilai
1	<i>Batch size</i>	24
2	<i>Subdivisions</i>	16
3	<i>Width</i>	416
4	<i>Height</i>	416
5	<i>Channels</i>	3
6	<i>Angle</i>	0
7	<i>Saturation</i>	1.5
8	<i>Exposure</i>	1.5
9	<i>Hue</i>	0.1
10	<i>Learning rate</i>	0.001
11	<i>Momentum</i>	0.949
12	<i>Burn in</i>	1000
13	<i>Max batches</i>	6000
14	<i>Steps</i>	4800, 5400
15	<i>Scales</i>	0.1, 0.1
16	<i>Cutmix</i>	1
17	<i>Mosaic</i>	1
18	<i>Size</i>	1
19	<i>Stride</i>	1
20	<i>Pad</i>	1
21	<i>Filters</i>	27
22	<i>Mask</i>	0,1,2; 3,4,5; 6,7,8
23	<i>Anchors</i>	12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
24	<i>Classes</i>	4

*Learning rate* sebesar 0.001 dengan *momentum* 0.949 digunakan dalam algoritma optimasi. Kebijakan pembelajaran (*learning policy*) melibatkan *burn-in* sebesar 1000 iterasi dan maksimum iterasi sebesar 6000. Pengurangan *learning rate* dilakukan pada langkah-langkah 4800 dan 5400 dengan skala pengurangan 0.1. Teknik augmentasi data *CutMix* dan *Mosaic* juga diterapkan untuk meningkatkan variasi data pelatihan[19].

Arsitektur model *YOLOv4* terdiri dari *convolutional layer* dengan ukuran 1x1 dan 27 filter, diikuti oleh *layer YOLO*. *Layer YOLO* memiliki tiga bagian, masing-masing dengan mask dan anchors yang berbeda[20]. *Mask* digunakan untuk menentukan bagian grid yang bertanggung jawab dalam mendekripsi objek, sementara *anchors* adalah ukuran kotak pendekripsi yang dirancang untuk menangkap variasi skala dan rasio aspek objek yang ingin didekripsi, dan umumnya dipilih berdasarkan ukuran objek yang ada dalam dataset pelatihan[21]. Model dilatih untuk mendekripsi 4 kelas yang terkait dengan kelelahan wajah pekerja sesuai label yang telah ditentukan.

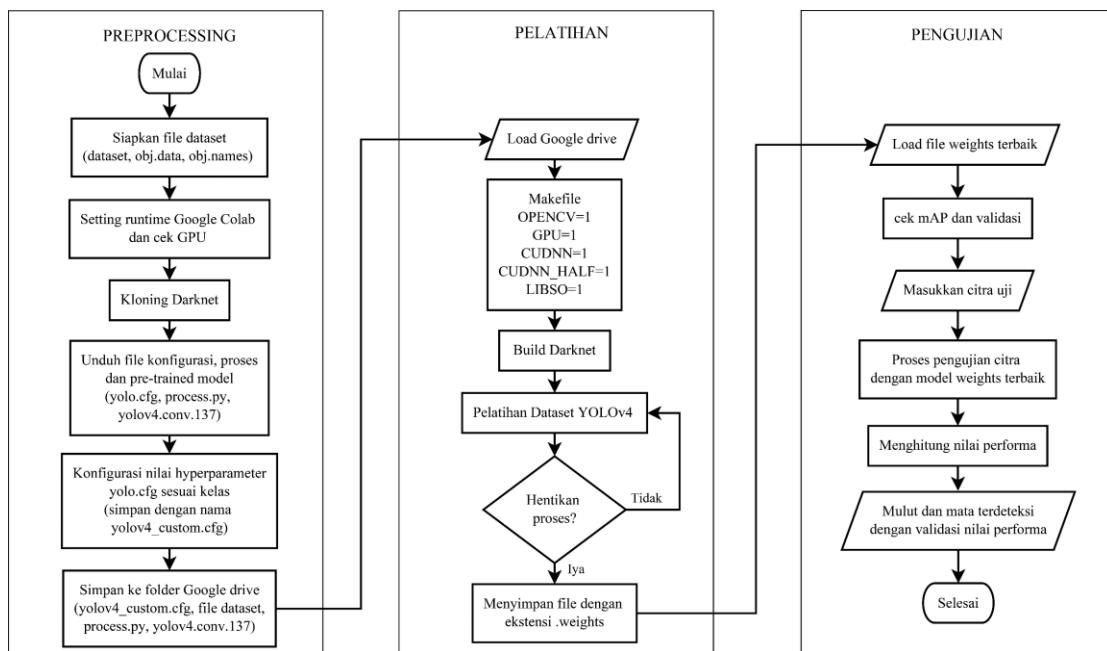
Setelah melakukan konfigurasi *hyperparameter* jaringan *YOLO*, hasil dari konfigurasi tersebut disimpan kembali dalam file berekstensi .cfg dan diberi nama *yolov4\_custom.cfg*. Kemudian, file konfigurasi dan dataset diunggah dalam 1 folder di google drive dengan nama *yolov4*.

## E. Training Data

Proses pelatihan pada *YOLOv4* dimulai dengan memuat file-file pada folder *yolov4*, seperti *yolov4\_custom.cfg*, dataset, *yolov4.conv.137*, dan *process.py* dari *Google Drive*. Selanjutnya, dilakukan konfigurasi menggunakan *Makefile* dengan parameter tertentu seperti *OPENCV=1*, *GPU=1*, *CUDNN=1*, *CUDNN\_HALF=1*, dan *LIBSO=1* untuk memaksimalkan penggunaan *OpenCV*, memanfaatkan GPU, dan mengaktifkan fitur-fitur yang terkait dengan pustaka *CUDNN*. Setelah itu, dilakukan pembangunan *Darknet* sebagai kerangka kerja yang digunakan untuk melatih model *YOLOv4*.

Selanjutnya, dilakukan pelatihan dataset *YOLOv4* berdasarkan model *Darknet* yang telah dibangun. Proses ini melibatkan pemrosesan dataset yang telah disiapkan dan mengajarkan model untuk mengenali dan memprediksi objek dalam dataset tersebut. Pelatihan dataset dapat dilakukan beberapa kali dengan meningkatkan jumlah iterasi untuk meningkatkan performa model. Evaluasi kinerja model dilakukan secara berkala dan keputusan diambil untuk melanjutkan atau menghentikan proses pelatihan.

Setelah pelatihan selesai, model yang telah dilatih disimpan sebagai file dengan ekstensi .weights. Proses pelatihan menyimpan *weights* file setiap 1000 iterasi, selanjutnya akan disimpan setiap 112 iterasi sebagai *last\_weights* dan disimpan setiap mendapatkan nilai performa *mAP* terbaik sebagai *best\_weights*. File tersebut nantinya akan digunakan dalam tahap-tahap selanjutnya sesuai dengan alur kerja seperti ditunjukkan dalam Gambar 3.



Gambar 3. Alur kerja proses

#### F. Testing Data

Berdasarkan Gambar 3, tahap pengujian *YOLOv4* dimulai dengan memuat file *weights* terbaik yang dihasilkan dari tahap pelatihan sebelumnya. Kemudian, evaluasi *mAP* dan validasi model dilakukan untuk memastikan kualitas dan keandalannya. Nilai *mAP* (*mean Average Precision*) merupakan nilai rata-rata dari presisi rata-rata atau *Average Precision* (*AP*) yang berfungsi untuk menilai sejauh mana performansi file bobot hasil pelatihan (*weights*) pada tahap pelatihan data[22]. Dengan menggunakan metrik ini, dapat diketahui seberapa baik performa model dalam mendeteksi objek secara keseluruhan.

Selanjutnya, citra uji dimasukkan ke dalam sistem untuk diuji. Proses pengujian ini menggunakan model *weights* terbaik pada citra uji untuk mendeteksi objek, khususnya fokus pada deteksi mulut dan mata dalam konteks deteksi kelelahan atau normal. Setelah terdeteksi, performa model dievaluasi berdasarkan hasil validasi dan dilakukan perhitungan nilai performa. Proses pengujian ini bertujuan untuk memastikan bahwa model yang dilatih dapat dengan akurat dan efektif mendeteksi tanda-tanda kelelahan pada wajah pekerja.

Pada tahap pengujian, dilakukan penggunaan kriteria yang didasarkan pada jurnal penelitian oleh A. K. A. Tianto yang berjudul Faktor-Faktor yang Berhubungan dengan Keluhan Kelelahan Mata pada Pekerja Kantor X Karangayär[23]. Penelitian tersebut menyatakan bahwa mata lelah dan mulut menguap merupakan tanda-tanda kelelahan saat bekerja, sedangkan kondisi normal dapat dilihat ketika mata dan mulut tidak mengalami perubahan dari kondisi normalnya. Dalam pengujian ini, kriteria tersebut digunakan sebagai acuan dalam mengkategorikan kelelahan atau kondisi normal pada wajah pekerja.

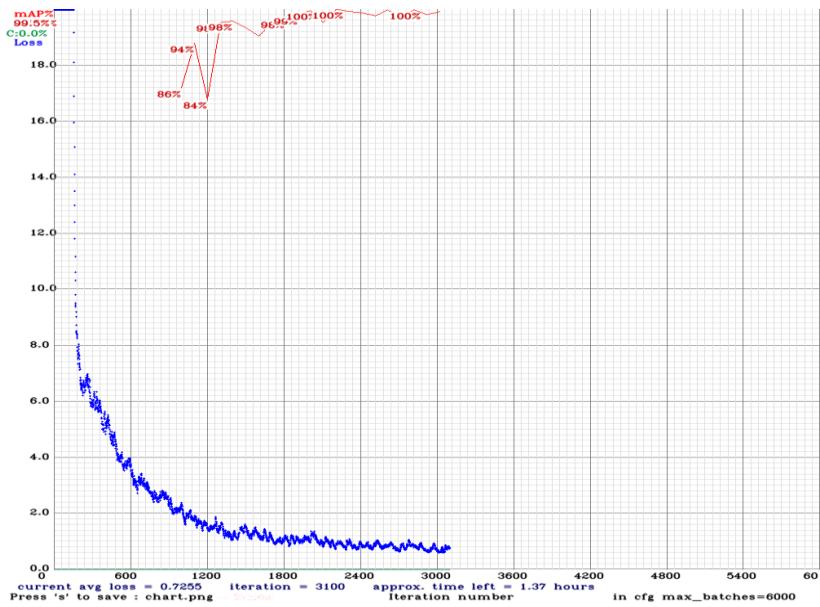
#### G. Implementasi

Setelah menyelesaikan tahap pelatihan dan pengujian, langkah terakhir adalah mengimplementasikan program pada Webcam atau DroidCam sebagai simulasi pemantauan CCTV untuk mendapatkan citra wajah secara *realtime*. Citra wajah pekerja yang diambil dari Webcam atau CCTV kemudian diproses menggunakan model algoritma *YOLOv4* untuk mendeteksi tanda-tanda kelelahan dari mata dan mulut. Implementasi ini melibatkan skenario penggunaan citra baru yang diunggah ke sistem dan skenario pemanfaatan Webcam sebagai simulasi kamera CCTV. Hasil deteksi akan ditampilkan secara visual dengan menggambarkan kotak pembatas (*bounding boxes*) di sekitar area yang terdeteksi. Semua skenario implementasi akan dilakukan di *Google Colaboratory*.

### III. HASIL DAN PEMBAHASAN

#### A. Hasil pelatihan

Proses pelatihan dataset dilakukan menggunakan *Google Colaboratory* yang berlangsung selama 10 jam untuk menghasilkan model file pelatihan baru. Model file pelatihan baru akan disimpan oleh sistem dengan nama *yolov4\_custom\_last.weights*. File tersebut yang digunakan untuk melakukan tahap pengujian citra baru. Hasil dari pelatihan menghasilkan sebuah grafik proses pelatihan yang ditunjukkan dalam Gambar 4. Tabel 5 merupakan hasil analisis dari grafik training dan hasil evaluasi model *yolov4\_custom\_last.weights*.



Gambar 4. Grafik hasil pelatihan menggunakan YOLOv4

Tabel 5. Performa evaluasi model YOLOv4

No	Hasil	Skor
1	Iterasi	3100
2	Max batches	6000
3	Average Loss	0.7255
4	Precision	96%
5	Recall	99.9%
6	F1-score	98%
7	Average IoU	78.73%
8	mAP@0.50	99.5%

Hasil pelatihan algoritma YOLOv4 menunjukkan rata-rata *loss* 0.7255 setelah mencapai iterasi ke-3100. Meskipun *loss* ini masih rendah, proses pelatihan masih berlangsung dan belum mencapai iterasi maksimum yang ditetapkan. Hal ini menunjukkan kemampuan YOLOv4 dalam mempelajari dan menyesuaikan pola dataset yang digunakan. Estimasi waktu yang diperlukan untuk menyelesaikan pelatihan pada iterasi ke-6000 adalah sekitar 1.37 jam. Namun, tergantung pada performa perangkat keras yang digunakan dan kecepatan GPU *Google Colaboratory*.

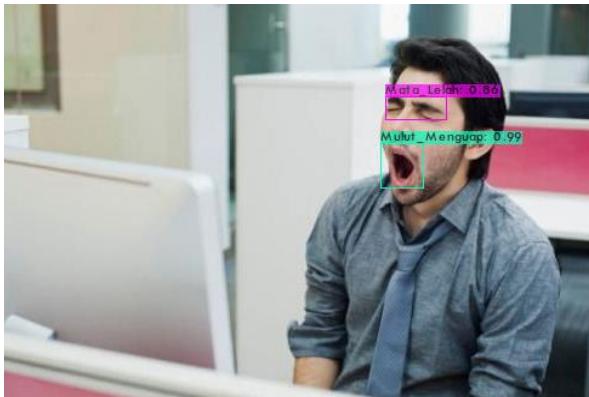
Dalam file konfigurasi(yolov4\_custom.cfg), *max\_batches* ditetapkan sebesar 6000, pengaturan ini menunjukkan harapan mencapai titik konvergensi setelah 6000 iterasi. Berdasarkan perkiraan waktu yang tersisa untuk menyelesaikan pelatihan hingga mencapai iterasi maksimum, hal ini akan memungkinkan algoritma YOLOv4 untuk mencapai hasil yang lebih optimal dan stabil. Selain itu, hasil pelatihan juga menunjukkan bahwa algoritma YOLOv4 memiliki tingkat akurasi yang tinggi dengan mencapai *mAP* sebesar 99.5%.

Pada evaluasi model yolov4\_custom\_last.weight yang ditunjukkan dalam Tabel 5, didapatkan performa yang baik dalam deteksi objek pada citra. Presisi mencapai 96%, menunjukkan tingkat akurasi deteksi objek yang tinggi. *Recall* sebesar 99.9% menunjukkan kemampuan algoritma dalam mendeteksi objek secara keseluruhan. *F1-Score* mencapai 98%, menunjukkan keseimbangan yang baik antara presisi dan *recall*. Rata-rata *IoU* sebesar 78.73% menunjukkan ketepatan kotak deteksi objek dalam menutupi objek sebenarnya. *mAP@0.50* mencapai 99.5%, menunjukkan akurasi deteksi objek yang baik pada berbagai ukuran dan bentuk. Dapat disimpulkan bahwa model YOLOv4 memiliki performa yang baik dalam deteksi objek pada citra. Tingkat *precision*, *recall*, *F1-Score*, *Average IoU*, dan *mAP* yang tinggi menunjukkan kemampuan algoritma dalam mengenali dan mendeteksi objek secara akurat.

## B. Hasil pengujian

### Skenario pengujian menggunakan citra baru

Berdasarkan laporan hasil pengujian menggunakan citra baru, terdapat beberapa analisis yang dapat diambil dan ditunjukkan dalam Tabel 6. Pengujian dilakukan terhadap dua citra baru menggunakan model *yolov4* yang sudah dilatih (yolov4\_custom\_last.weights) sebelumnya. Algoritma YOLOv4 mampu mendeteksi mata dan mulut serta memiliki perolehan *mAP* yang signifikan seperti yang ditunjukkan dalam Gambar 5 dan Gambar 6. Berikut penjelasan analisis dari hasil perolehan *mAP* pada dua citra yang diuji.



Gambar 5. Hasil Pengujian Citra Baru (kelelahan)



Gambar 6. Hasil Pengujian Citra Baru (normal)

Tabel 6. Hasil Pengujian Citra Baru

No	Gambar	Label terdeteksi	Skor (%)
1	Gambar 5	Mata_Lelah	86%
		Mulut_Menguap	99%
2	Gambar 6	Mata_Normal	73%
		Mulut_Normal	89%

Pertama, pada Gambar 5 dengan keluaran mata lelah sebesar 86% dan mulut menguap sebesar 99%, hasil tersebut menunjukkan bahwa algoritma *YOLOv4* memiliki tingkat keberhasilan yang tinggi dalam mendeteksi tanda-tanda kelelahan pada wajah pekerja. Deteksi mata lelah sebesar 86% menunjukkan kemampuan model dalam mengenali mata yang menunjukkan gejala kelelahan. Selain itu, deteksi mulut menguap sebesar 99% menunjukkan sensitivitas yang baik dalam mendeteksi mulut yang mengalami kelelahan. Mata lelah dan mulut menguap dapat mengindikasikan tingkat kelelahan yang tinggi pada pekerja sehingga pekerja pada Gambar 5 perlu segera diperhatikan dan ditangani.

Kedua, pada Gambar 6 dengan keluaran mata normal 73% dan mulut normal 89%, hasil tersebut menunjukkan bahwa model *YOLOv4* juga dapat membedakan antara kelelahan dan kondisi normal pada wajah pekerja. Deteksi mata normal sebesar 73% menunjukkan kemampuan model dalam mengenali mata yang tidak menunjukkan tanda-tanda kelelahan. Deteksi mulut normal sebesar 89% menunjukkan bahwa model dapat mengenali mulut dalam keadaan normal, yang dapat membantu membedakan antara kelelahan dan keadaan normal pada pekerja. Mata dan mulut normal dapat mengindikasikan pekerja tersebut dalam keadaan normal sehingga pekerja tidak memerlukan tindakan khusus.

#### Skenario pengujian secara *realtime*

Hasil pengujian secara *realtime* dengan skenario pengujian menggunakan webcam sebagai CCTV dalam deteksi kelelahan wajah pekerja beresiko tinggi menggunakan *YOLOv4* telah menghasilkan hasil yang signifikan dengan penjabaran seperti yang ditunjukkan dalam Tabel 7. Hal ini dapat ditunjukkan melalui hasil pengujian seperti yang ditunjukkan dalam Gambar 7 dan Gambar 8. Pada kondisi pertama, algoritma berhasil memperoleh *mAP* (*mean Average Precision*) mata normal sebesar 98.30% dan *mAP* mulut normal sebesar 98.07%. Hasil ini menunjukkan tingkat akurasi yang tinggi dalam mendeteksi keadaan mata dan mulut pekerja dalam keadaan normal. Selain itu, sistem juga mampu memberikan klasifikasi yang tepat, yaitu "pekerja dalam kondisi aktif", yang menunjukkan bahwa pekerja sedang dalam kondisi baik dan tidak mengalami tanda-tanda kelelahan.

Pada kondisi kedua, algoritma berhasil mencapai *mAP* mata lelah sebesar 98.28% dan *mAP* mulut menguap sebesar 98.04%. Hal ini menunjukkan bahwa kemampuan sistem deteksi *YOLOv4* dalam mendeteksi keadaan mata lelah dan mulut menguap pada pekerja dengan tingkat akurasi yang tinggi. Selain itu, klasifikasi yang dihasilkan oleh sistem, yaitu "pekerja dalam kondisi lelah", sesuai dengan kondisi yang sebenarnya. Hal ini menunjukkan kemampuan algoritma *YOLOv4* dalam mengenali dan mengklasifikasikan kelelahan pada wajah pekerja beresiko tinggi.

Hasil pengujian ini membuktikan bahwa algoritma *YOLOv4* mampu memberikan performa yang baik dalam mendeteksi kelelahan wajah secara *realtime*. Tingkat akurasi yang tinggi dalam mengenali mata normal dan mulut normal, serta mata lelah dan mulut menguap, bersama dengan klasifikasi yang tepat, menunjukkan kehandalan algoritma ini dalam mengenali dan mengindikasikan tanda-tanda kelelahan pada wajah pekerja sesuai dengan klasifikasi yang ditentukan. Dibandingkan dengan penelitian terdahulu yang dijadikan sebagai rujukan pada latar belakang, penelitian ini memiliki label yang lebih banyak namun nilai *mAP* pada performa pengujian algoritma *YOLOv4* jauh lebih baik serta mampu mengklasifikasi kesimpulan dari performa pengujian dengan tepat. Hasil ini memiliki potensi besar untuk mendukung pengawasan kelelahan pekerja, memungkinkan respons yang cepat dan tepat untuk mengurangi risiko kecelakaan serta menjaga kesehatan dan keselamatan kerja pekerja beresiko tinggi.



Gambar 7. Hasil Pengujian secara *realtime* (Normal)



Gambar 8. Hasil Pengujian secara *realtime* (Kelelahan)

Tabel 7. Hasil Pengujian Secara *Realtime*

No	Gambar	Label terdeteksi	Skor (%)	Hasil Klasifikasi
1	Gambar 7	Mata_Normal	98.30%	Pekerja Dalam Kondisi Aktif
		Mulut_Normal	98.07%	
2	Gambar 8	Mata_Lelah	98.28%	Pekerja Dalam Kondisi Lelah!
		Mulut_Menguap	98.04%	

#### IV. KESIMPULAN

Berdasarkan hasil pelatihan dan pengujian yang dilakukan, dapat disimpulkan bahwa algoritma *YOLOv4* memiliki performa yang baik dalam deteksi kelelahan pada wajah pekerja. Proses pelatihan menghasilkan model dengan tingkat akurasi yang tinggi, ditunjukkan oleh presisi sebesar 96%, *recall* sebesar 99.9%, *F1-Score* sebesar 98%, *Average IoU* sebesar 78.73%, dan *mAP@0.50* mencapai 99.5%. Hasil pengujian yang dilakukan menggunakan citra baru dan secara *realtime* juga menunjukkan kemampuan algoritma dalam mengenali dan mengklasifikasikan tanda-tanda kelelahan dengan tingkat akurasi yang signifikan. Algoritma *YOLOv4* mampu mencapai *mAP* tertinggi sebesar 99% ketika dilakukan pengujian pada citra baru dan mencapai *mAP* tertinggi sebesar 98.30% ketika dilakukan pengujian secara *realtime* menggunakan webcam. Hal ini memiliki potensi besar untuk mendukung upaya pengawasan dan pencegahan kelelahan pada pekerja beresiko tinggi dengan respons yang cepat dan tepat. Dengan demikian, algoritma *YOLOv4* dapat dijadikan sebagai alat yang efektif untuk meningkatkan pengawasan kelelahan pekerja, kesehatan dan keselamatan kerja pekerja, serta mengurangi risiko kecelakaan yang disebabkan oleh kelelahan.

Pada penelitian berikutnya dapat diharapkan melanjutkan pengembangan algoritma *YOLOv4* dengan melibatkan lebih banyak data pelatihan yang representatif, termasuk variasi kondisi kelelahan pada wajah pekerja. Hal ini akan membantu meningkatkan kemampuan algoritma dalam mengenali dan mengklasifikasikan tanda-tanda kelelahan secara lebih akurat. Selain itu, penting untuk melakukan evaluasi kinerja secara berkala dan memperbarui model dengan dataset yang lebih baru untuk memastikan performa yang optimal. Selanjutnya, implementasi algoritma *YOLOv4* dapat dikombinasikan dengan sistem monitoring dan alarm yang dapat memberikan peringatan dini kepada pekerja dan pengawas ketika tanda-tanda kelelahan terdeteksi. Dengan demikian, pengawasan kelelahan pekerja dapat dilakukan secara proaktif, meminimalkan risiko kecelakaan, dan meningkatkan kesehatan serta keselamatan kerja pekerja beresiko tinggi.

#### REFERENSI

- [1] M. J. L. Gaol, A. Camelia, and A. Rahmiwati, “Analisis Faktor Risiko Kelelahan Kerja Pada Karyawan Bagian Produksi PT. Arwana Anugrah Keramik, Tbk,” *J. Ilmu Kesehat. Masy.*, vol. 9, no. 1, pp. 53–63, Mar. 2018, doi: 10.26553/jikm.2018.9.1.53-63.
- [2] B. A. Deyulmar, Suroto, and I. Wahyuni, “Analisis Faktor-Faktor Yang Berhubungan Dengan Kelelahan Kerja Pada Pekerja Pembuat Kerupuk Opak Di Desa Ngadikerso, Kabupaten Semarang,” *J. Kesehat. Masy.*, vol. 6, no. 4, pp. 278–285, 2018, doi: 10.14710/jkm.v6i4.21428.
- [3] S. Firdaus and K. D. Artika, “Deteksi Kelelahan Pengemudi Mobil Menggunakan Citra Wajah,” *Elem. J. Tek. MESIN*, vol. 8, no. 1, p. 16, Jun. 2021, doi: 10.34128/je.v8i1.154.
- [4] L. Lady and A. S. Wiyanto, “TINGKAT KELELAHAN KERJA PADA PEKERJA LUAR RUANGAN DAN PENGARUH LINGKUNGAN FISIK TERHADAP PENINGKATAN KELELAHAN,” *J. Ind. Serv.*, vol. 5, no. 1, pp. 58–64, Oct. 2019, doi: 10.36055/jiss.v5i1.6504.

- [5] F. O. Reynaldi, "Analisa Performa Arsitektur Mobilenetv1 Dan Resnet Menggunakan Meta-Learning Dalam Mendeteksi Objek Hewan Kucing," *Indones. J. Bus. Intell.*, vol. 4, no. 1, p. 1, 2021, doi: 10.21927/ijubi.v4i1.1686.
- [6] N. Irtija, M. Sami, and M. A. R. Ahad, "Fatigue Detection Using Facial Landmarks," *Int. Symp. Affect. Sci. Eng.*, vol. ISASE2018, no. November, pp. 1–6, 2018, doi: 10.5057/isase.2018-C000041.
- [7] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," Apr. 2018, doi: 10.48550/arXiv.1804.02767.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, doi: 10.48550/arXiv.2004.10934.
- [9] D. H. Fudholi, R. A. N. Nayoan, M. Suyuti, and R. Rahmadi, "Deteksi Indikasi Kelelahan Menggunakan Deep Learning," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 5, pp. 1–9, 2021, doi: 10.30645/j-sakti.v5i1.292.
- [10] E. R. Setyaningsih and M. S. Edy, "YOLOv4 dan Mask R-CNN Untuk Deteksi Kerusakan Pada Karung Komoditi," *Teknika*, vol. 11, no. 1, pp. 45–52, 2022, doi: 10.34148/teknika.v11i1.419.
- [11] D. J. P. Manajang, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor," *J. Tek. Inform.*, vol. 8, no. 4, pp. 1821–1831, Dec. 2021, doi: doi.org/10.35793/jti.15.3.2020.29775.
- [12] Timj, "Fatigue Dataset," *Kaggle*, 2021. <https://www.kaggle.com/datasets/timmjy/fatigue-detection>.
- [13] M. Sarosa and N. Muna, "Implementasi Algoritma You Only Look Once ( Yolo ) Untuk Implementation of You Only Look Once ( Yolo ) Algorithm for," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, pp. 787–792, 2021, doi: 10.25126/jtiik.202184407.
- [14] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, vol. 2020-June, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.
- [15] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.
- [16] N. Chen, Y. Man, and Y. Sun, "Abnormal Cockpit Pilot Driving Behavior Detection Using YOLOv4 Fused Attention Mechanism," *Electronics*, vol. 11, no. 16, p. 2538, Aug. 2022, doi: 10.3390/electronics11162538.
- [17] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 13024–13033, doi: 10.1109/CVPR46437.2021.01283.
- [18] R. S. Huda, R. Wulanningrum, and D. Swanjaya, "PEMANFAATAN YOLOV4 UNTUK DETEKSI PELANGGARAN HELM DAN MASKER SERTA IDENTIFIKASI PELAT NOMOR MENGGUNAKAN TESSERACT-OCR," *JOUTICA*, vol. 7, no. 2, p. 596, Sep. 2022, doi: 10.30736/informatika.v7i2.873.
- [19] Q. Chen and Q. Xiong, "Garbage Classification Detection Based on Improved YOLOV4," *J. Comput. Commun.*, vol. 08, no. 12, pp. 285–294, 2020, doi: 10.4236/jcc.2020.812023.
- [20] S. Jupiyandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, and I. Cholissodin, "Pengembangan Deteksi Citra Mobil Untuk Mengetahui Jumlah Tempat Parkir Menggunakan Cuda Dan Modified Yolo Development of Car Image Detection To Find Out the Number of Parking Space Using Cuda and Modified Yolo," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, pp. 413–419, 2019, doi: 10.25126/jtiik.201961275.
- [21] A. F. Fandisyah, N. Iriawan, and W. S. Winahju, "Deteksi Kapal di Laut Indonesia Menggunakan YOLOv3," *J. Sains dan Seni ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373520.v10i1.59312.
- [22] T. A. A. H. Kusuma, K. Usman, and S. Saidah, "PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD," *J. Tek. Inform.*, vol. 2, no. 1, pp. 57–66, Feb. 2021, doi: 10.20884/1.jutif.2021.2.2.77.
- [23] A. K. A. Tianto, I. Qadrijati, and S. Haryati, "Faktor-Faktor Yang Berhubungan Dengan Keluhan Kelelahan Mata Pada Pekerja Kantor X Karanganyar," *J. Kesehat. Masy.*, vol. 1, no. 2, pp. 58–66, 2023, doi: 10.14710/jkm.v1i1.36786.

**Conflict of Interest Statement:**

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.