

Deteksi Kelelahan *Real-Time* Berbasis Pengolahan Citra Digital Sebagai Media Deteksi Perilaku Pengguna

Real-Time Fatigue Detection Based on Digital Image Processing as a Medium for User Behavior Detection

Rifki Noviandra Lestari¹, Dinda Ayu Permatasari^{*2}, Gillang Al Azhar³

Politeknik Negeri Malang, Jl. Soekarno Hatta No.9, (0341) 404420

E-mail : Rnoviandra3@gmail.com¹, dinda_ayu@polinema.ac.id^{*2}, gillang_al_azhar@polinema.ac.id³

Received 26 June 2026; Revised 28 July 2026; Accepted 30 July 2026

Abstrak - Kelelahan merupakan faktor tertinggi terjadinya kecelakaan kerja, hal ini disebabkan karena pendeteksian dan penanganan yang sangat lambat akibat tidak adanya penanganan dini ketika terjadi kelelahan. Oleh karena itu, dibutuhkan sebuah alat otomatis yang dapat mendeteksi terjadinya kelelahan kemudian memberikan alarm agar kelelahan segera bisa ditangani. Kelelahan ini dapat dideteksi melalui fitur fisik berupa gerakan kepala, gerakan mulut, dan posisi mata. Parameter-parameter ini dapat dideteksi menggunakan pengolahan citra digital, dimana aktivitas pengguna akan dipantau melalui kamera kemudian akan diklasifikasikan menggunakan metode YoloV8s-Pose dan Random Forest. Dengan sistem yang dirancang dan pengujian yang telah dilakukan, didapati akurasi dan tingkat kepercayaan sistem yang sangat tinggi yaitu pada kisaran akurasi 95% dengan Tingkat kepercayaan 0.83. Notifikasi telegram juga mempersingkat waktu pendeteksian dan penanganan terjadinya kelelahan. Oleh karena itu, dapat disimpulkan sistem ini secara akurat dan efektif berhasil mendeteksi dan menangani kelelahan dengan waktu yang singkat.

Kata kunci - *Deteksi Objek, Kelelahan, Pengolahan Citra Digital, Sistem Peringatan Dini Otomatis, YoloV8*

Abstract - Fatigue is the leading cause of occupational accidents, primarily due to delayed detection and a lack of early intervention. Therefore, an automated system is required to detect fatigue and trigger an alarm to enable immediate response. This condition can be identified through physical features such as head movements, mouth movements, and eye states. These parameters are extracted using digital image processing, where user activity is monitored via a camera and classified using YOLOv8s-Pose and Random Forest algorithms. Based on the system design and subsequent testing, the system demonstrated high performance, achieving an accuracy of approximately 95% with a confidence score of 0.83. Furthermore, the integration of Telegram notifications significantly reduces the time required for detection and response. In conclusion, the proposed system accurately and effectively detects and manages fatigue within a minimal timeframe.

Keywords – *Automatic Warning System, Digital Image Processing, Fatigue, Object Detection, YoloV8*

1. PENDAHULUAN

Keselamatan dan Kesehatan Kerja (K3) adalah pilar fundamental bagi keberlangsungan operasi industri. Namun, hingga kini kecelakaan kerja masih menjadi masalah yang mengancam dunia kerja di Indonesia. Data menunjukkan bahwa meskipun upaya K3 terus ditingkatkan, angka kasus kecelakaan kerja nasional masih berada pada tingkat yang mengkhawatirkan. Sebagai contoh, Badan Penyelenggara Jaminan Sosial Ketenagakerjaan (BPJS Ketenagakerjaan) mencatat bahwa pada tahun 2023, jumlah kasus kecelakaan kerja mencapai angka yang signifikan, yakni

370.747 kasus. Fluktuasi dan tingginya angka ini mengindikasikan bahwa resiko di lingkungan kerja masih sangat besar dan mendesak untuk ditelaah lebih dalam.

Kecelakaan kerja merupakan hasil dari interaksi kompleks berbagai faktor yang dapat dikelompokkan menjadi kondisi tidak aman dan tindakan tidak aman [1]. Tidak hanya itu, dalam teori Domino H.W. Heinrich disebutkan bahwa 88% kecelakaan kerja diakibatkan oleh kesalahan manusia. Kontribusi dari faktor manusia seringkali jauh lebih dominan dalam memicu insiden. Dari berbagai variabel yang memengaruhi kinerja manusia, kelelahan kerja (*work fatigue*) menempati posisi sentral sebagai faktor utama kegagalan kinerja yang berujung pada kecelakaan kerja [1][2][3].

Pada penelitian terdahulu telah dibuat alat pendeteksi kelelahan dengan menggunakan sensor-sensor yang terpasang pada dada pengguna [4]. Penelitian ini dapat membuktikan bahwa kelelahan dapat dideteksi melalui tanda-tanda yang dihasilkan oleh tubuh. Namun, rancangan alat tersebut dapat membuat pengguna tidak nyaman saat dipakai dengan durasi yang lama. Tidak hanya itu, keakuratan sensor yang digunakan juga dapat terpengaruh oleh kondisi lingkungan kerja yang tidak kondusif dan penuh gangguan sehingga menurunkan *confident rate* dalam pendeteksian kelelahan.

Oleh karena itu dibuat sebuah alat deteksi kelelahan berbasis pengolahan citra digital untuk memantau perilaku pengguna pada saat melakukan aktivitas secara terus menerus dan *real-time*. Penggunaan citra digital akan mengurangi *noise-noise* yang diakibatkan oleh kondisi lapangan yang kurang memadai. Sistem juga harus dipasangkan atau diintegrasikan dengan IoT (*Internet of Things*) yang nantinya akan berfungsi sebagai notifikasi dini terjadinya kelelahan supaya dapat dilakukan penindakan langsung sebelum terjadinya kecelakaan kerja.

Penelitian ini bertujuan untuk mengimplementasikan metode pengolahan citra digital untuk dapat mendeteksi terjadinya kelelahan serta mengukur seberapa efektif metode digunakan pada kondisi aktual di lingkungan kerja. Dengan adanya sistem ini, kecelakaan kerja dapat ditanggulangi sehingga frekuensi kecelakaan kerja yang terjadi dapat ditekan hingga nilai yang sangat rendah.

Penelitian ini akan dibagi menjadi beberapa tahap yaitu, perencanaan sistem, implementasi metode dan pengujian, analisa keluaran dan kekurangan sistem, pembuatan dan pencatatan laporan performa sistem. Dengan penggunaan pengolahan citra digital, deteksi kelelahan secara otomatis direncanakan memiliki nilai akurasi di atas 80%. Hal ini ditetapkan guna memastikan terjadinya kelelahan benar-benar dideteksi dengan tepat dan sesuai sehingga tidak terjadi kecelakaan kerja kembali akibat kesalahan deteksi oleh sistem.

2. METODE PENELITIAN

Penelitian mengenai alat pendeteksi kelelahan telah banyak dilakukan karena urgensinya yang sangat penting dalam menyelesaikan kasus-kasus kecelakaan karena kelelahan. Secara global, kelelahan memberikan kontribusi sebesar 50% terhadap kecelakaan kerja [5]. Dari hal ini terlihat bahwa kelelahan adalah masalah yang benar-benar harus diselesaikan. Kelelahan ini dapat diselesaikan dengan menggunakan teknologi deteksi kelelahan, terdapat beberapa cara untuk menanggulangi resiko kelelahan, salah satunya adalah dengan dibuatnya teknologi untuk mendeteksi kelelahan tersebut secara otomatis [6].

Kelelahan ini dapat dideteksi melalui kondisi fisik mulai dari raut wajah, postur tubuh, dan variabilitas detak jantung. Akurasi pendeteksian kelelahan menggunakan kombinasi fitur fisik seperti fitur wajah, fitur mata, atau kombinasi dari fitur-fitur fisik tersebut memiliki akurasi deteksi kelelahan sebesar 99.59% [7]. Dari data tersebut dapat dikatakan pendeteksian kelelahan otomatis menggunakan ekstraksi fitur kamera sangatlah memungkinkan dan dapat menyelesaikan masalah dengan efisien.

Metode penelitian ini meliputi perancangan sistem dengan mengintegrasikan Algoritma YoloV8 dan Klasifikasi *Random Forest* sebagai sistem deteksi kelelahan berdasarkan aktivitas pengguna khususnya di bagian gerakan kepala dan ekspresi wajah. Gerakan kepala akan dideteksi

melalui jarak antara telinga dengan bahu kemudian untuk ekspresi wajah akan dideteksi melalui kedipan mata dan gerakan menguap. Selain itu terdapat perancangan elektronik dan mekanik sistem untuk memastikan metode deteksi dan klasifikasi memiliki dampak yang riil dalam mengurangi dan menanggulangi adanya kecelakaan akibat kelelahan.

2.1. Kelelahan

Kelelahan merupakan kondisi subjektif yang bersifat multidimensi dan dapat melemahkan kualitas hidup seseorang [8]. Secara umum, kelelahan ditandai oleh penurunan kemampuan fisik, terutama ketika otot tidak lagi mampu menghasilkan kekuatan yang dibutuhkan selama aktivitas. Selain itu, kelelahan memiliki gejala yang beragam karena dapat dipengaruhi oleh faktor fisik, kognitif, maupun emosional sesuai kondisi medis yang dialami. Secara fisiologis, kelelahan terjadi melalui mekanisme kompleks yang mencakup kegagalan pada sistem saraf pusat (kelelahan sentral) maupun pada otot atau transmisi saraf ke otot (kelelahan perifer). Kondisi ini juga sering diiringi perubahan psikofisiologis, seperti peningkatan aktivitas sistem saraf simpatis pada denyut nadi, yang menunjukkan adanya stres atau peningkatan arousal tubuh.

Penelitian terdahulu juga menjelaskan bahwa kelelahan dapat dideteksi melalui gerakan-gerakan tubuh. Kelelahan dapat dideteksi melalui tanda fisik berupa gerakan mengganggu kepala berulang dan durasi kedipan mata [9]. Gejala lain kelelahan yaitu lama waktu mata tertutup lebih dari 80% bagian selama satu menit lebih dari 15% mengindikasikan kelelahan [10].

2.2. Algoritma YOLO (You Only Look Once)

Algoritma YOLO (*You Only Look Once*) [11] merupakan salah satu terobosan fundamental dalam bidang *real-time object detection*. Inovasi utama YOLO terletak pada pendekatannya yang memformulasikan masalah deteksi objek, yang sebelumnya dilakukan dalam beberapa tahap terpisah, menjadi tugas regresi tunggal pada seluruh citra. Filosofi “You Only Look Once” merujuk pada kemampuan algoritma memproses seluruh citra hanya dalam satu kali forward pass melalui jaringan saraf tiruan. Hal ini menghilangkan kebutuhan akan mekanisme *Region Proposal* yang lambat dan kompleks, sehingga secara drastis meningkatkan kecepatan inferensi, menjadikan YOLO sangat ideal untuk aplikasi *real-time* [12][13].

Secara mekanisme, YOLO bekerja dengan membagi citra *input* menjadi sel-sel grid $S \times S$. Setiap sel grid bertanggung jawab untuk mendeteksi objek yang titik pusatnya berada di dalam sel tersebut. Setiap sel kemudian memprediksi sejumlah kotak pembatas (*bounding box*) dan skor keyakinan (*confidence score*) yang menunjukkan kemungkinan adanya objek dan akurasi posisinya, serta probabilitas kelas objek yang terdeteksi.

Pendeteksian objek di dalam Algoritma YOLO dimulai dari pengambilan *raw data* citra kemudian dilakukan perhitungan nilai tiap grid dan pada tahap akhir dilakukan pembuatan *bounding box* dan penentuan *confident rate* [14].

2.3. Perancangan YoloV8

Pada perancangan YoloV8 [11], keperluan deteksi adalah aktivitas pengguna mulai dari mata, mulut, dan gerakan kepala. Dengan keperluan ini, maka digunakan YoloV8 *Pose Estimation* agar keluaran lebih maksimal. Pada dasarnya, YoloV8 memiliki 3 bagian pemrosesan yang nantinya akan berfungsi untuk memproses citra dan melakukan deteksi sesuai dengan *weight dataset* yang telah dibuat. Pada bagian pertama terdapat bagian *bottleneck* yang berfungsi sebagai bagian penerima masukan citra mentah dan melakukan ekstraksi fitur seperti tepi, tekstur dan bentuk melalui beberapa lapisan konvolusi. Bagian *bottleneck* terdiri dari *Modified CSPDarknet53*, *C2fModule* (*Cross-Stage Partial Bottleneck with 2 Convolutions*) [11], dan *SPPF* (*Spatial Pyramid Pooling - Fast*) [15][16]. Berikut adalah penjelasan mengenai bagian-bagian *bottleneck* pada YoloV8:

1. Modified CSPDarknet53

Pada bagian ini, logika *Cross Stage Partial* difokuskan pada strategi membagi dan menaklukkan untuk peta fitur. Ketika terdapat peta fitur yang masuk, peta ini akan dibagi menjadi 2 bagian. Salah satu bagian dilanjutkan ke bagian *Processing Branch* dan bagian lain disalurkan ke *Skip Branch* kemudian menggabungkannya kembali pada akhir proses. Hal ini bertujuan agar model dapat mengingat *weight* di *layer* yang semakin dalam.

2. C2f Module

Peta fitur yang telah diproses oleh bagian *Modified CSPDarknet53*, akan melalui konvolusi 1×1 untuk penyesuaian dimensi *channel*. Kemudian keluaran akan di pisah menjadi 2 bagian, dimana setengah bagian dimasukkan ke kumpulan *bottleneck* dan setengah bagian lain menjadi nilai identitas. Selanjutnya seluruh keluaran *bottleneck* akan dihubungkan dan disatukan sehingga keluaran dari *C2f Module* memiliki nilai informasi yang beragam dari tiap-tiap *layer* proses.

3. Spatial Pyramid Pooling – Fast

Setelah melalui *C2f Module*, peta fitur memiliki banyak informasi namun memiliki resolusi yang sempit, sedangkan deteksi yang dilakukan memiliki resolusi yang bermacam-macam. SPPF akan membuat peta fitur dengan jumlah *pooling* yang berbeda-beda dengan menggunakan 5×5 *pooling* secara konstan. Hal ini menjadikan keluaran terbagi menjadi 5×5 , 9×9 , dan 13×13 *pooling* yang membuat model mempelajari lebih banyak fitur karena memiliki resolusi yang berbeda-beda dan mendetail.

Setelah melalui *bottleneck*, citra akan masuk ke bagian *neck* yang berfungsi sebagai penerima seluruh peta fitur yang dihasilkan oleh *backbone* untuk memastikan model memiliki detail dengan resolusi tinggi dan memiliki informasi yang lengkap. Pada bagian ini terdapat PAN (*Path Aggregation Network*) yang dibagi menjadi 2 jalur yaitu *Top-Down Path* yang berfungsi untuk memasukkan informasi dari *deep layer* ke *layer* yang lebih dangkal namun memiliki resolusi yang lebih tinggi dan *Bottom-Up Path* berfungsi memasukkan kembali sinyal lokalisasi dari *layer* dangkal ke *layer* yang lebih dalam. Logika ini disebut dengan *Dual Flow Architecture* yang memungkinkan model untuk melakukan prediksi *multi scale* agar dapat melakukan deteksi objek kecil diantara banyaknya objek yang lebih besar.

Pada bagian akhir pemrosesan terdapat bagian *Head* yang berfungsi sebagai pembuat keputusan *final* yang akan disebut sebagai *predictions*. Pada YoloV8 model yang digunakan pada *head* adalah model *decoupled* yang dimana pendeteksian akan dilakukan secara paralel dalam mendeteksi apa objek yang dideteksi dan dimana letak objek tersebut. Deteksi yang dilakukan juga menggunakan model *Anchor Free Detection* dimana model ini akan mendeteksi objek tanpa harus mempedulikan *pre-defined box*, sehingga model dapat menyesuaikan bentuk *box* sesuai dengan objek yang dideteksi. Pada bagian ini digunakan 3 macam perhitungan untuk menentukan probabilitas *class*, *bounding box*, hingga *keypoint*. Perhitungan yang digunakan adalah *Binary Class Entropy* (BCE), *Complete Intersection over Union* (CIoU), dan *Distribution Focal Loss* (DFL) [15]. Berikut adalah rumus perhitungan tiap perhitungan yang digunakan:

1. Binary Class Entropy (BCE)

$$L_{cls} = -\frac{1}{N_{pos}} \sum_{j=1}^{N_{pos}} \left[t_j \log(\sigma(\hat{y}_j)) + (1 - t_j) \log(1 - \sigma(\hat{y}_j)) \right] \quad (1)$$

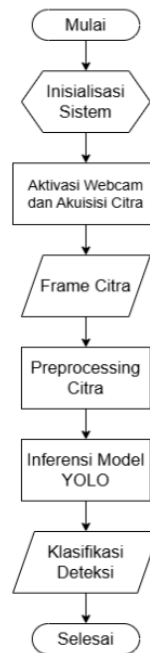
2. Complete Intersection over Union

$$L_{box} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (2)$$

3. Distribution Focal Loss

$$L_{dfl} = -[(y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})] \quad (3)$$

Selanjutnya adalah proses pengambilan gambar dan pendeteksian gambar dengan alur sebagai berikut:

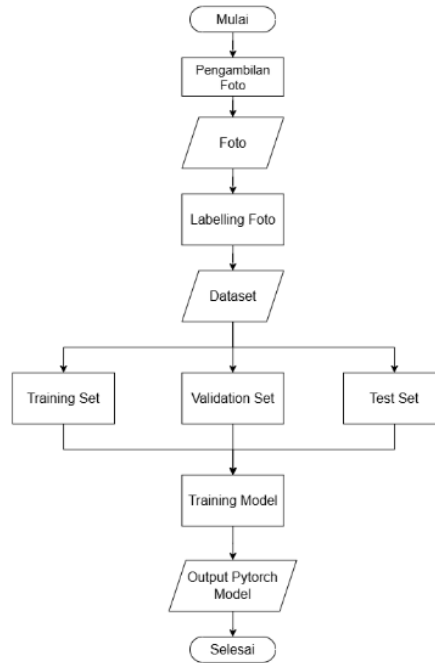


Gambar 1. Diagram Alir *Image Processing*

Pada Gambar 1, dijabarkan diagram alir dari *Image Processing* dimana sistem akan dimulai dengan inisialisasi sistem kemudian dilakukan akuisisi citra melalui *webcam*. *Frame* citra yang tertangkap kemudian akan dilakukan *preprocessing* terlebih dahulu sebelum masuk ke proses inferensi Model Yolo. Setelah melalui inferensi maka akan keluar hasil deteksi berupa klasifikasi objek dengan keluaran *bounding box* dan *keypoints* model.

Berdasarkan keperluan deteksi objek dan pose estimation, maka digunakan *dataset* dengan pembagian yang tepat dan label yang sesuai. Dataset yang dibuat harus memperhitungkan jarak dan intensitas cahaya karena hal ini mempengaruhi *confidence score* dari hasil deteksi [17]. Gambar 2 adalah alur pembuatan *dataset* untuk keperluan *training* data Algoritma Yolo.

Pembuatan *dataset* dimulai dengan melakukan pengambilan citra mentah, kemudian citra akan diberikan label sesuai dengan parameter deteksi dan estimasi pose yang diperlukan. Citra yang telah diberikan label akan dimasukkan sebagai *dataset* dengan perbandingan nilai *training* sebesar 70%, *valid* sebesar 20%, dan *test* sebesar 10% dari keseluruhan data. Setelah *dataset* dibuat, akan dilakukan *training* model yang akan menghasilkan hasil akhir berupa file PyTorch. Keluaran hasil *training dataset* harus dianalisa terlebih dahulu, untuk mengetahui tingkat *Recall*, *Precision*, dan *Confidence* dari model yang telah dilatih [18]. *Recall* adalah rasio dari jumlah prediksi positif yang benar terhadap total jumlah elemen positif yang sebenarnya dalam data.

Gambar 2. Diagram Alir Pembuatan *Dataset*

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

Untuk *Precision* adalah rasio dari jumlah prediksi positif yang benar terhadap total jumlah prediksi positif yang dibuat oleh model. Nilai ini akan menilai seberapa akurat sistem dalam melakukan pendeteksian objek.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

Perhitungan parameter terakhir adalah *Confidence*. Parameter ini adalah nilai yang menunjukkan seberapa yakin model terhadap prediksi yang dihasilkan untuk suatu objek saat deteksi dilakukan.

$$Confidence = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (6)$$

Pada Algoritma Yolo, *confidence score* adalah probabilitas bahwa kotak prediksi mengandung objek tertentu. *Confidence* digunakan untuk memutuskan apakah prediksi suatu objek cukup kuat untuk dianggap benar [19]. Prediksi dengan nilai *confidence* yang tinggi menunjukkan keyakinan tinggi model terhadap keberadaan objek dalam *bounding box* tersebut. Setelah hasil didapatkan, parameter yang dihasilkan dari pendeteksian adalah klasifikasi mata tertutup dan terbuka, mulut terbuka atau tertutup, dan *keypoints*. Klasifikasi mata tertutup dan terbuka akan dilakukan dengan melakukan identifikasi terbuka atau tertutupnya kelopak mata. Kemudian, klasifikasi mulut terbuka dan mulut tertutup dilakukan menggunakan deteksi *keypoints* mulut bagian atas (x_1, y_1) dan mulut bagian bawah (x_2, y_2) dengan perhitungan *Euclidean* [20] sebagai berikut:

$$Mouth_{closed} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \geq 30px \quad (7)$$

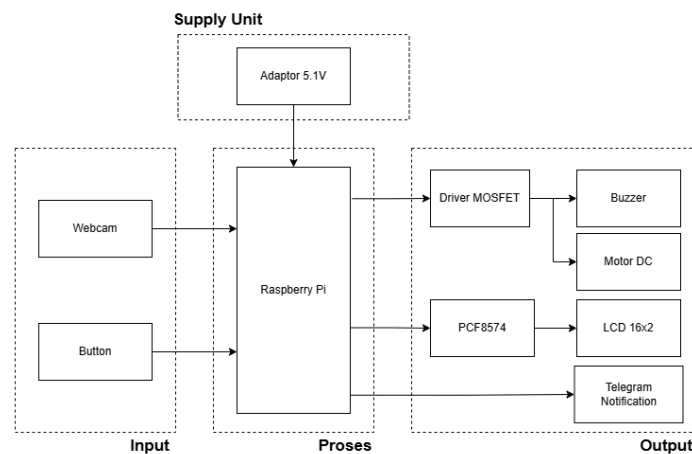
Dengan menggunakan persamaan 6, dapat diketahui jika jarak antara titik mulut bagian atas dan bagian bawah berjarak lebih atau sama dengan 30 piksel maka sistem akan mengklasifikasikan sebagai mulut terbuka. Tidak hanya itu, klasifikasi mulut terbuka akan terjadi jika pengguna mengarahkan tangannya ke mulut. Selanjutnya untuk menentukan jarak kemiringan dan posisi kepala, digunakan *keypoints* yang terletak pada telinga kanan (x_3, y_3) dan ujung bahu kanan (x_4, y_4) juga telinga kiri (x_5, y_5) dan bahu kiri (x_6, y_6). Perhitungan jarak tersebut digunakan dengan persamaan *Euclidean* berikut:

$$\text{Head Tilt} = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} \text{ atau } \sqrt{(x_6 - x_5)^2 + (y_6 - y_5)^2} \quad (8)$$

Dari perhitungan 6, data yang akan diambil untuk menentukan kemiringan kepala didapatkan melalui pemilihan nilai terkecil diantara kedua persamaan. Setelah keseluruhan parameter klasifikasi didapatkan, maka data akan langsung dikirimkan ke pemrosesan selanjutnya sebagai data perhitungan kelelahan.

2.4. Perancangan Elektronik Sistem

Pada Gambar 2, dijabarkan mengenai keperluan elektronik dari sistem yang akan dibuat. Terdapat 4 blok utama yaitu, blok *input*, blok *proses*, blok *output*, dan blok suplai. Catu daya sistem berasal dari Adaptor 5.1V yang digunakan untuk suplai tegangan Raspberry Pi 5. Selanjutnya terdapat 2 komponen input yaitu *webcam* yang akan tersambung ke *Port* USB Raspberry Pi 5 dan *Button* yang akan tersambung dengan *Pin* GPIO Raspberry Pi 5. Pada bagian proses digunakan Raspberry Pi 5 sebagai pusat pengolahan seluruh data mulai dari pengambilan data citra, proses inferensi YOLO, dan klasifikasi *Random Forest*. Pada blok terakhir yaitu blok *output* terdapat 3 komponen *output* yaitu motor DC dan Buzzer yang akan tersambung dengan *Driver* MOSFET, kemudian terdapat LCD 16x2 yang disambungkan dengan PCF8574 *Module* agar dapat berkomunikasi dengan Raspberry Pi 5 melalui komunikasi I2C (*Inter-Integrated Circuit*).



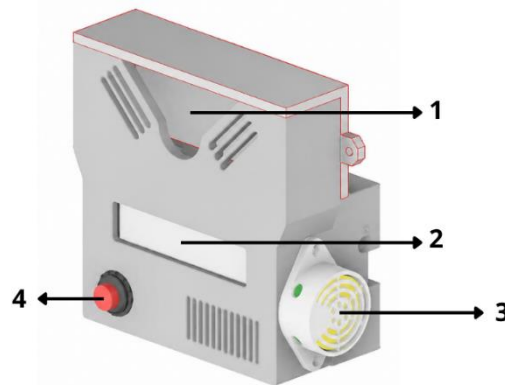
Gambar 3. Diagram Blok Sistem Elektronik

Webcam berfungsi sebagai komponen penangkap citra awal, selanjutnya citra akan diteruskan ke Raspberry Pi 5 dan akan dikelola menggunakan Algoritma Yolo agar dapat diketahui atau diidentifikasi object yang terdapat didalamnya. Fungsi *Button* adalah untuk melakukan konfirmasi setelah pengguna sadar dari kelelahan yang telah dialami dan nilai-nilai sistem deteksi akan *direset*. Pada bagian output, *Driver* MOSFET berfungsi untuk mengontrol Buzzer dan Motor DC sehingga dapat menyala ketika terjadi kelelahan dan dapat memberikan stimulasi kepada pengguna agar dapat tersadar dari kelelahannya. Selanjutnya LCD 16x2 digunakan untuk menampilkan keseluruhan nilai deteksi agar dapat dipantau secara langsung. Notifikasi Telegram akan dikirimkan setelah terjadi kelelahan, hal ini digunakan untuk

memberikan peringatan kepada orang yang bertanggung jawab terhadap pengguna agar dapat melakukan Tindakan secepatnya tanpa harus *stand-by* di dekat pengguna.

2.5. Perancangan Mekanik

Perancangan mekanik digunakan untuk memastikan komponen-komponen disusun dan ditata se-ergonomis mungkin agar keluaran yang dihasilkan bisa lebih maksimal dan tidak mempersulit pengguna dalam menggunakan dan pengoperasian alat. Perancangan mekanik system dapat dilihat pada Gambar 3.



Gambar 4. Perancangan Mekanik dan Peletakan Komponen

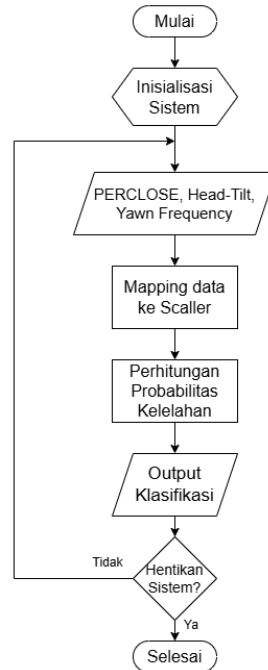
Bahan *box* yang digunakan adalah Filamen PLA (*Polylactic Acid*) dan dicetak menggunakan mesin 3D Printer. Bahan ini memiliki berat yang ringan namun kokoh sehingga sangat cocok digunakan dalam system yang akan dibuat. Dengan dimensi P x L x T, sudah terdapat beberapa ventilasi untuk memastikan terdapat sirkulasi udara agar komponen dan sirkuit tidak mengalami *overheat*. Selain itu penataan komponen juga dirincikan pada Gambar 2 dengan deskripsi sebagai berikut:

1. *Holder webcam*
2. *LCD 16x2*
3. *Buzzer*
4. *Button*

Peletakan komponen diatur sedemikian rupa untuk memastikan pengguna dapat dengan mudah mengoperasikan alat yang telah dibuat. Tidak hanya itu Motor DC juga diletakkan pada sebuah gelang yang nantinya akan digunakan oleh pengguna pada pergelangan tangan kanan sehingga dapat berfungsi sebagai sumber stimulasi getaran untuk menyadarkan pengguna.

2.6. Perancangan Random Forest

Klasifikasi dibutuhkan untuk menggabungkan parameter hasil deteksi citra kemudian menentukan keputusan terhadap data yang telah diterima. *Random Forest* menggunakan pendekatan *bagging* (*bootstrap aggregating*) [21]. Selain itu, fitur yang digunakan pada tiap *node* dipilih secara acak sehingga tiap-tiap *decision tree* memiliki struktur yang beragam, hal ini membuat resiko model mengalami *overfitting* berkurang. Dalam proses prediksi, *Random Forest* menentukan hasil klasifikasi berdasarkan hasil keluaran dari seluruh pohon pada kasus klasifikasi atau berdasarkan nilai rata-rata pada tiap regresi [22]. Algoritma ini dapat membuat keputusan dengan akurasi yang sangat tinggi yang berada pada kisaran 85,4% – 97,98% [4]. Berikut adalah alur klasifikasi random forest dengan data deteksi citra melalui kamera:

Gambar 5. Perancangan Klasifikasi *Random Forest*

Pada Gambar 5, proses klasifikasi akan dimulai dengan proses inisiali sistem, dimana akan dilakukan penentuan jenis *scaler* untuk penyamaan *range* data tiap-tiap parameter. Selanjutnya akan dilakukan pengambilan data berdasarkan parameter deteksi yaitu nilai PERCLOS, *Head-Tilt*, dan *Yawn Frequency*. Data-data ini akan diambil melalui hasil deteksi melalui kamera. Kemudian data tersebut akan dimasukkan ke dalam *scaler* untuk dilakukan penyesuaian dan penyamaan range nilai parameter [23]. Pada parameter PERCLOS, data keluaran klasifikasi akan dihitung melalui persamaan berikut:

$$PERCLOS = \left(\frac{N_{closed}}{\frac{FPS}{60}} \right) \times 100\% \quad (9)$$

Dari persamaan 7, didapatkan persentase PERCLOS didapatkan melalui pengambilan jumlah deteksi atau klasifikasi mata tertutup kemudian dibagi besarnya *Frame Per Second* (FPS) dan dibagi oleh 60, untuk menentukan panjangnya PERCLOS per menit [23][24]. Selanjutnya perhitungan akan dikalikan dengan 100% untuk penentuan besarnya persentase PERCLOS per menit. Data selanjutnya adalah data parameter *Yawn Frequency*. Data ini dihitung melalui hasil klasifikasi mulut terbuka dengan perhitungan sebagai berikut:

$$Yawn_{Frequency} + 1 \text{ Ketika } \Delta t_{mouth_open} - (t_{sekarang} - t_{mulai}) > 2s \quad (10)$$

Frekuensi menguap akan bertambah ketika terjadinya kondisi mulut terbuka atau mulut tertutup tangan dengan *threshold* sebesar 2 detik. Selanjutnya data perhitungan *PERCLOS* dan *Yawn_Frequency* akan dikirimkan ke *Random Forest* beserta data keluaran kemiringan kepala. *Random Forest* akan memperhitungkan probabilitas kelelahan dengan perhitungan sebagai berikut:

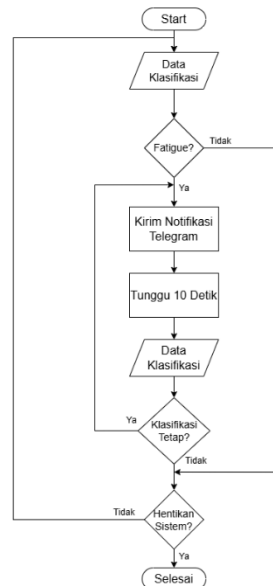
$$P(c|x) = \frac{1}{B} \sum_{b=1}^B I(\widehat{y}_b = c) \quad (11)$$

Dari persamaan 7, dapat diketahui bahwa nilai *confidence* pada *Random Forest* menggunakan nilai rata-rata dari keseluruhan *Decision Tree* yang digunakan dalam *training*. Ketika perhitungan selesai dilakukan maka keluaran akan berbentuk data klasifikasi dan nilai *confidence* dari pendeteksian tersebut.

2.7. Perancangan Pengiriman Notifikasi Telegram

Pengiriman notifikasi tidak menggunakan *Telegram Bot*, namun menggunakan *API Development Tools* yang dimiliki oleh Telegram ini sendiri. Penggunaan fitur ini mempermudah adanya pengiriman pesan karena memiliki sistem autentikasi permanen tanpa ada batasan waktu *expired session*. Namun kelemahannya pengiriman pesan hanya dapat dikirimkan untuk nomor yang terdaftar pada kontak pemilik *API Development Tools*. Walaupun terlihat seperti kekurangan yang sangat buruk, namun hal ini dapat menjadi komponen *safety* karena hanya pengguna-pengguna yang terdaftar dan teridentifikasi yang dapat menerima pesan tersebut.

Pada sistem, notifikasi ini dikirimkan untuk memberikan pesan kepada penerima bahwa terjadi kelelahan atau *fatigue* terhadap pengguna alat. Sehingga dapat dilakukan penindakan secepatnya sebelum adanya hal-hal yang tidak diinginkan. Berikut adalah alur pengiriman notifikasi yang dilakukan sistem:



Gambar 6. Perancangan Pengiriman Notifikasi Telegram

Pada Gambar 6, dijelaskan diagram alir proses pengiriman notifikasi melalui Telegram yang dimulai dari proses pengambilan data hasil klasifikasi oleh algoritma *Random Forest*. Sistem terlebih dahulu menerima input berupa data klasifikasi kondisi pengguna. Selanjutnya dilakukan pengecekan terhadap hasil klasifikasi tersebut. Apabila hasil klasifikasi menunjukkan kondisi *fatigue*, maka sistem akan melanjutkan ke proses pengiriman notifikasi Telegram ke nomor atau akun yang telah ditentukan sebelumnya. Namun, apabila hasil klasifikasi tidak menunjukkan kondisi *fatigue*, maka sistem akan masuk ke proses pengondisian penghentian sistem.

Setelah notifikasi Telegram berhasil dikirim, sistem akan menunggu selama 10 detik sebelum melakukan pengambilan data klasifikasi kembali. Setelah jeda waktu selesai, sistem kembali menerima input data klasifikasi hasil *Random Forest* dan melakukan pengecekan apakah kondisi *fatigue* masih terdeteksi. Jika kondisi *fatigue* masih berlangsung, maka sistem akan kembali mengirimkan notifikasi Telegram. Sebaliknya, jika kondisi *fatigue* sudah tidak terdeteksi, maka sistem akan masuk ke proses pengondisian penghentian sistem.

Pada proses pengondisian penghentian sistem, pengguna diberikan pilihan apakah sistem akan dihentikan atau tetap dijalankan. Jika pengguna memilih untuk menghentikan sistem, maka proses

akan berakhir. Namun, jika sistem tidak dihentikan, maka sistem akan kembali melakukan pengambilan data klasifikasi dan mengulangi proses pemantauan kondisi pengguna secara berkelanjutan.

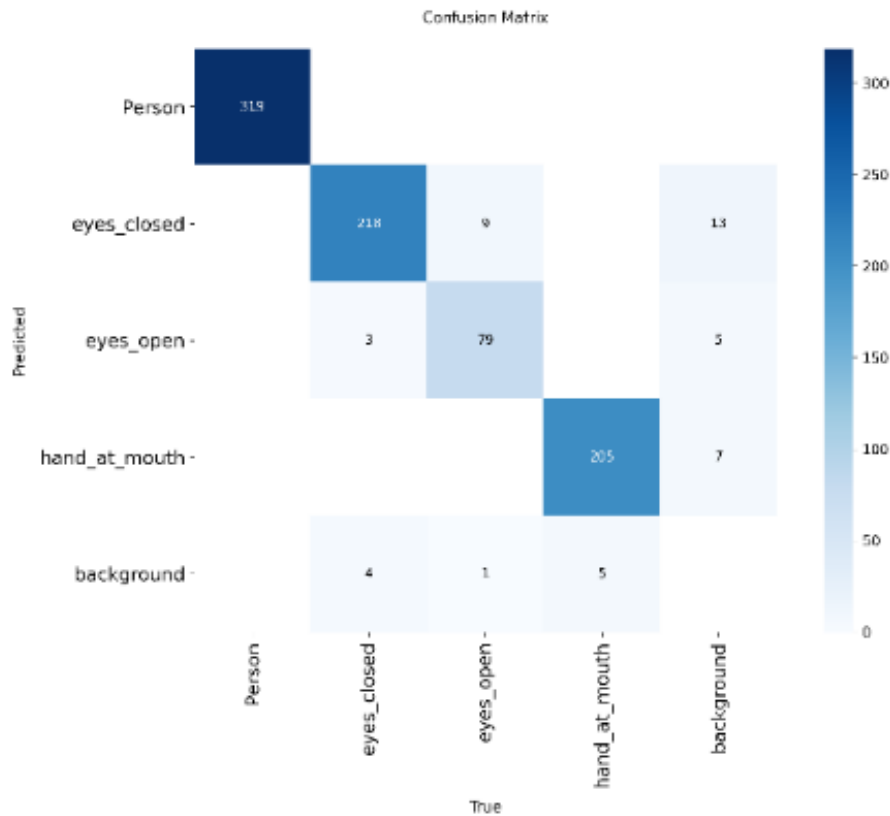
3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mendapatkan hasil berupa data dan keluaran yang dapat dianalisis. Pengujian *webcam* dilakukan dengan tujuan memastikan *webcam* dapat digunakan dan berfungsi dengan baik Ketika dihubungkan dengan Raspberry Pi 5. Pengujian akan dilakukan dan diuji dengan 10 objek yang berbeda dengan kondisi pencahayaan dan jarak yang berbeda-beda. Setiap pengujian akan dilakukan 5 kali pengulangan, untuk memastikan sistem secara stabil dan akurat dalam mendeteksi fitur-fitur kelelahan pengguna.

Tabel 1. Pengujian *Webcam*

Uji	Hasil	Piksel	Ukuran
1		1080p	310.2 kB
2		1080p	296.7 kB
3		1080p	283.7 kB
4		1080p	366.9 kB
5		1080p	388.5kB

Hasil keluaran pengujian *webcam* menunjukkan bahwa *webcam* bekerja dengan baik dan bisa digunakan untuk akuisisi citra mentah. Selanjutnya dilakukan *training* model menggunakan tipe model YoloV8s-Pose. Dengan *dataset* sebanyak 3046 data yang dibagi menjadi 2628 *training set*, 319 *validation set*, dan 99 *test set*. Dari training tersebut didapatkan grafik *confusion matrix* [19] yang berisi data mengenai kinerja dan kebingungan model pada saat dilakukan *training*.



Gambar 7. Confusion Matrix

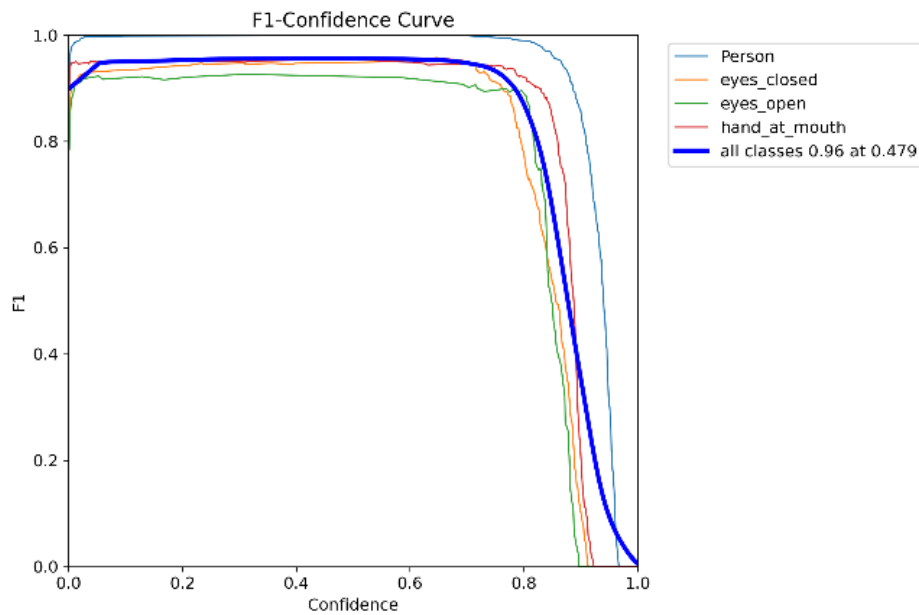
Dari matriks pada Gambar 7, dengan menggunakan perhitungan pada persamaan 4 dan 5, didapatkan nilai *Recall* tiap *class* sebagai berikut:

1. *Person*: 1.00
2. *Eyes_open*: 0.89
3. *Eyes_closed*: 0.97
4. *Hand_at_mouth*: 0.98

Keluaran nilai *Recall* menunjukkan model dapat mengidentifikasi semua *class* beserta *keypoint* yang telah ditentukan pada tiap-tiap data validasi dan tes dengan rata-rata nilai 96. Selanjutnya didapatkan nilai *Precision* tiap *class* sebagai berikut:

1. *Person*: 1.00
2. *Eyes_open*: 0.91
3. *Eyes_closed*: 0.91
4. *Hand_at_mouth*: 0.97

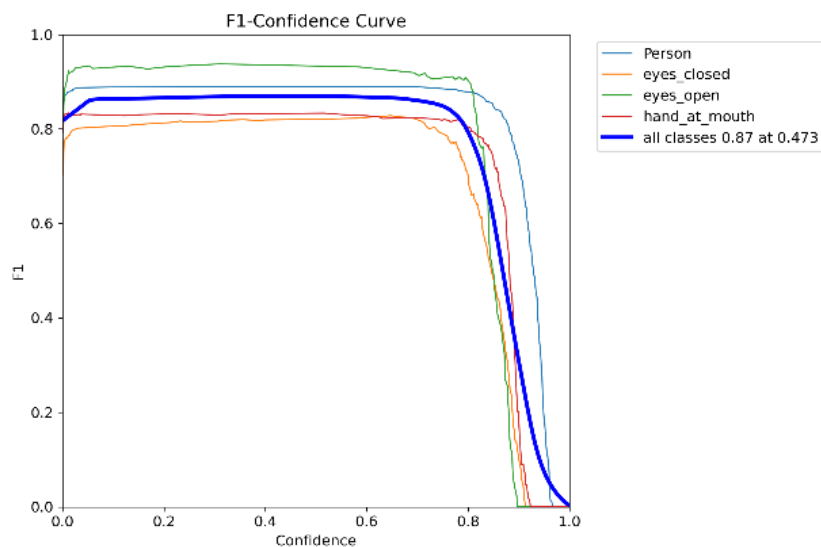
Keluaran nilai *Precision* menunjukkan model dapat memprediksi semua *class* beserta *keypoint* yang telah ditentukan pada tiap-tiap data validasi dan tes dengan rata-rata nilai 0.94. Model yang di-*training* memiliki kinerja yang sangat optimal dalam melakukan deteksi fitur yang telah ditetapkan sebelumnya. Dengan hasil tersebut dapat dipastikan model akan dapat mendeteksi objek pada citra yang nantinya akan diambil dari *webcam*.



Gambar 8. *F1-Confidence Score Box*

Pada Gambar 8, dijabarkan keluaran nilai *f1-confidence* yang menunjukkan nilai keseimbangan antara *precision* dan *recall* pada nilai *confidence* yang berbeda-beda. Pada deteksi objek model memiliki nilai *F1* yang sangat tinggi yaitu diatas 0.9 dari titik 0 hingga pada titik *confidence* mendekati 0.8 setelah itu model mengalami penurunan *F1 Score* pada kondisi setelah 0.8. Hal ini membuktikan bahwa model sangat bisa diandalkan dalam mendeteksi adanya objek kelelahan pada kondisi *confidence score* 0.00 hingga 0.85.

Selanjutnya selain deteksi dengan keluaran *box*, pada *f1-confidence* yang ditunjukkan pada Gambar 8, nilai *f1-score* berada kondisi yang sangat baik, dengan nilai di atas 0.8 pada *threshold confidence* sebesar 0.8. Hal ini menunjukkan keluaran dari *training* model YoloV8s-Pose sangatlah baik dalam melakukan deteksi fitur.



Gambar 9. *F1-Confidence Score Pose*

Setelah dilakukan *training*, langkah selanjutnya adalah *deploy* model kedalam Raspberry Pi 5 sehingga pengujian deteksi aktual dapat dilakukan dan dapat dianalisa untuk menentukan hasil keluaran sistem keseluruhan.

Tabel 2. Pengujian Deteksi Fitur Fisik Kelelahan

Uji	Parameter	Hasil	Confidence
1	Mata Terbuka dan Pose		0.72
2	Mata Terbuka dan Pose		0.76
3	Mata Tertutup dan Pose		0.74
4	Mata Tertutup dan Pose		0.70
5	Tangan Di Mulut dan Pose		0.76

Dari hasil pengujian pada Tabel 2, terlihat model dapat mendeteksi fitur-fitur yang telah ditentukan untuk deteksi fitur fisik kelelahan. Dengan data diatas didapatkan keluaran yang akurat pada *bounding box* dimana model dapat mendeteksi keempat kelas dengan nilai *confidence* berkisar pada nilai 0.70 hingga 0.85. Pada bagian *pose estimation* dapat dilihat seluruh titik *keypoints* berada pada titik yang benar, yaitu berada pada telinga, bahu, dan mulut. Tidak hanya itu, perpindahan posisi objek tidak mempengaruhi *confidence score* yang dihasilkan, hal ini membuktikan kinerja deteksi sudah akurat dan stabil. Selanjutnya akan dilakukan pengujian keseluruhan sistem menggunakan klasifikasi *random forest* sehingga parameter-parameter deteksi dapat diklasifikasikan menjadi kondisi kelelahan ataupun kondisi normal.

Tabel 3. Pengujian Keseluruhan Sistem

Uji	Hasil	Data	Proba
1		Perclos: 3.2% Yawn: 5 Head_Tilt: 190.5	0.02 (Normal)
2		Perclos: 8.2% Yawn: 5 Head_Tilt: 169.1	0.05 (Normal)
3		Perclos: 31.4% Yawn: 5 Head_Tilt: 107.3	0.54 (Fatigue)
4		Perclos: 16.5% Yawn: 2 Head_Tilt: 120.2	0.19 (Normal)
5		Perclos: 40.7% Yawn: 7 Head_Tilt: 70.9	0.83 (Fatigue)

Hasil pengujian keseluruhan pada Tabel 3, menunjukkan bahwa parameter kelelahan yang diklasifikasikan pada *Random Forest* akan diklasifikasikan sebagai kelelahan jika terdapat dua atau lebih parameter diatas ambang batas kelelahan yang telah ditentukan, yaitu PERCLOS diatas 15%, frekuensi menguap diatas 4, dan posisi kepala yang mendekati bahu dengan jarak kurang dari 5 cm. *Random Forest* akan mengklasifikasikan kelelahan ketika nilai probabilitas klasifikasi diatas 0.50, sehingga dengan *threshold* waktu sebesar 2 detik ketika klasifikasi ditentukan dan tidak berubah, maka dapat dipastikan kelelahan dapat dideteksi dengan akurat dan stabil oleh sistem ini.

Tabel 4. Notifikasi Kelelahan pada Telegram

Uji	Hasil	Data	Kondisi
1	Tidak ada notifikasi	Tidak ada notifikasi	Normal
2	Tidak ada notifikasi	Tidak ada notifikasi	Normal
3		Time: 10:26:33 Proba: 0.54	Fatigue
4	Tidak ada notifikasi	Tidak ada notifikasi	Normal
5		Time: 12:03:00 Proba: 0.83	Fatigue

Ketika terjadi kelelahan, dapat dilihat dari Tabel 4 bahwa seluruh kejadian kelelahan akan dikirimkan melalui telegram dilengkapi dengan gambar deteksi. Notifikasi hanya dikirimkan pada saat terjadi kelelahan sehingga jika tidak terjadi kelelahan tidak akan ada pesan yang dikirimkan melalui telegram. Dengan adanya hal ini, penanganan kelelahan dapat dilakukan sesegera mungkin agar tidak terjadi kecelakaan kerja yang terjadi akibat kelelahan yang tidak ditangani.

4. KESIMPULAN

Melalui pengambilan data yang telah dilakukan, sistem dapat mendeteksi fitur-fitur fisik kelelahan menggunakan metode YOLO. Akurasi deteksi rata-rata berada pada nilai 95% dilihat dari hasil keluaran training melalui nilai *recall* dan *precision* dengan nilai *confidence* rata-rata sebesar 0.82. Melalui klasifikasi *Random Forest*, kelelahan dapat dideteksi apabila minimal dua parameter diatas ambang kelelahan dengan *threshold* waktu stabil selama 2 detik. Hal ini menunjukkan bahwa sistem dapat mendeteksi kelelahan dengan akurat dan stabil walaupun dengan objek yang berbeda-beda. Pengiriman notifikasi pada telegram memiliki waktu delay yang sangat minim sehingga terjadinya kelelahan langsung bisa terdeteksi jarak jauh dan dapat segera dilakukan penanganan. Namun, kedepannya parameter kelelahan yang dideteksi akan lebih sempurna jika dipadukan dengan kondisi detak jantung pengguna, agar kelelahan tidak hanya dideteksi dari tampak luar pengguna saja sehingga dapat meningkatkan ketelitian dan keakuratan deteksi kelelahan kedepannya.

REFERENSI

- [1] Moch. F. H. Tiway, R. A. Ratriwardhani, F. Ayu, dan Moch. Sahri, “Analisis Kecelakaan Kerja pada Divisi Metal Stamping di PT. XYZ Menggunakan Metode Systematic Cause Analysis Technique (SCAT),” *Jurnal Keselamatan Kesehatan Kerja dan Lingkungan*, vol. 6, no. 2, hlm. 167–177, Okt 2025, doi: 10.25077/jk3l.6.2.167-177.2025.
- [2] N. Christiani Berek, L. Putu Ruliati, H. J. Ndun, dan D. J. Nabuasa, “The Influence of Workload, Work Fatigue, and Work Behavior on Work Accidents for Female Workers in Informal Sectors in Kupang City,” *Jurnal Ergonomi Indonesia*, vol. 8, no. 1, hlm. 30–36, doi: 10.24843/JEI.2022.v08.i01.p04.

- [3] A. TASDELEN dan A. ÖZPINAR, “The Impacts of Mental and Physical Fatigue of Employees on the Perception Level and the Risk of Accident,” *European Journal of Science and Technology*, Nov 2020, doi: 10.31590/ejosat.820997.
- [4] J. Halomoan, K. Ramli, D. Sudiana, T. S. Gunawan, dan M. Salman, “ECG-Based Driving Fatigue Detection Using Heart Rate Variability Analysis with Mutual Information,” *Information*, vol. 14, no. 10, hlm. 539, Okt 2023, doi: 10.3390/info14100539.
- [5] A. Agustin, T. Ihsan, dan R. A. Lestari, “Gambaran Faktor-Faktor Yang Mempengaruhi Kelelahan Kerja Pada Pekerja Industri Tekstil Di Indonesia: Review,” *Jurnal Keselamatan Kesehatan Kerja dan Lingkungan*, vol. 2, no. 2, hlm. 138–151, Okt 2021, doi: 10.25077/jk3l.2.2.138-151.2021.
- [6] E. R. Justitian, I. Y. Purbasari, dan F. T. Anggraeny, “Perbandingan Akurasi Deteksi Kelelahan pada Pengendara Menggunakan YOLOv3-Tiny YOLOv4-Tiny,” *Jurnal Informatika dan Sistem Informasi*, vol. 3, no. 1, hlm. 21–30, Apr 2022, doi: 10.33005/jifosi.v3i1.440.
- [7] O. Khunpisuth, T. Chotchinasri, V. Koschakosai, dan N. Hnoohom, “Driver Drowsiness Detection Using Eye-Closeness Detection,” dalam *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, 2016, hlm. 661–668. doi: 10.1109/SITIS.2016.110.
- [8] H. M. Abd-Elfattah, F. H. Abdelazeim, dan S. Elshennawy, “Physical and cognitive consequences of fatigue: A review,” *J. Adv. Res.*, vol. 6, no. 3, hlm. 351–358, Mei 2015, doi: 10.1016/j.jare.2015.01.011.
- [9] R. Hooda, V. Joshi, dan M. Shah, “A comprehensive review of approaches to detect fatigue using machine learning techniques,” *Chronic Dis. Transl. Med.*, vol. 8, no. 1, hlm. 26–35, Mar 2022, doi: 10.1016/j.cdtm.2021.07.002.
- [10] Ł. Dziuda *dkk.*, “Evaluation of a Fatigue Detector Using Eye Closure-Associated Indicators Acquired from Truck Drivers in a Simulator Study,” *Sensors*, vol. 21, no. 19, hlm. 6449, Sep 2021, doi: 10.3390/s21196449.
- [11] M. Talib, A. H. Y. Al-Noori, dan J. Suad, “YOLOv8-CAB: Improved YOLOv8 for Real-time object detection,” *Karbala International Journal of Modern Science*, vol. 10, no. 1, Jan 2024, doi: 10.33640/2405-609X.3339.
- [12] N. A. Megantara dan E. Utami, “Object Detection using YOLOv8: A Systematic Review,” *SISTEMASI*, vol. 14, no. 3, hlm. 1186, Mei 2025, doi: 10.32520/stmsi.v14i3.5081.
- [13] D. J. Marcelleno dan M. P. K. Putra, “PERFORMANCE EVALUATION OF YOLOV8 IN REAL-TIME VEHICLE DETECTION IN VARIOUS ENVIRONMENTAL CONDITIONS,” *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 1, hlm. 269–279, Feb 2025, doi: 10.52436/1.jutif.2025.6.1.3916.
- [14] S. Jupiandi, F. R. Saniputra, Y. Pratama, M. R. Dharmawan, dan I. Cholissodin, “Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 4, hlm. 413, Jul 2019, doi: 10.25126/jtiik.2019641275.
- [15] Vivi Afifah dan S. Erniwati, “YOLOv8 for Object Detection: A Comprehensive Review of Advances, Techniques, and Applications,” *IJACI: International Journal of Advanced Computing and Informatics*, vol. 2, no. 1, hlm. 53–61, Jul 2025, doi: 10.71129/ijaci.v2i1.pp53-61.
- [16] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, dan P. Allen, “Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy,” *Vehicles*, vol. 6, no. 3, hlm. 1364–1382, Agu 2024, doi: 10.3390/vehicles6030065.
- [17] D. A. Permatasari, H. Herwandi, D. S. Ma’arif, dan A. Ramelan, “Rancang Bangun Alat Sistem Absensi Mahasiswa menggunakan Face Recognition dengan Metode YOLO

- berbasis Raspberry Pi,” *JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika dan Komputer)*, vol. 6, no. 2, hlm. 114–124, Des 2024, doi: 10.26905/jasiek.v7i2.14144.
- [18] S. W. Sidehabi, I. Indrabayu, E. Warni, dan S. A. Bake, “Comparative Performance Analysis of YOLO and Faster R-CNN in Detecting Species and Estimating the Weight of Grouper and Snapper Fish Using Digital Images,” *JOIV: International Journal on Informatics Visualization*, vol. 9, no. 4, hlm. 1430, Jul 2025, doi: 10.62527/joiv.9.4.3121.
- [19] F. Jamaludin, A. Asriyanik, dan A. Pambudi, “PENERAPAN YOLO (YOU ONLY LOOK ONCE) UNTUK DETEKSI ETIKA BERBUSANA DI UNIVERSITAS MUHAMMADIYAH SUKABUMI,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, hlm. 10623–10629, Sep 2024, doi: 10.36040/jati.v8i5.11124.
- [20] Dantanarayana, H. G., & Huntley, J. M. (2017). Object recognition and localization from 3D point clouds by maximum-likelihood estimation. *Royal Society Open Science*, 4(8). <https://doi.org/10.1098/rsos.160693>
- [21] Y. Resti, C. Irsan, J. F. Latif, I. Yani, dan N. R. Dewi, “A Bootstrap-Aggregating in Random Forest Model for Classification of Corn Plant Diseases and Pests,” *Science and Technology Indonesia*, vol. 8, no. 2, hlm. 288–297, Apr 2023, doi: 10.26554/sti.2023.8.2.288-297.
- [22] T.-H. Lee, A. Ullah, dan R. Wang, “Bootstrap Aggregating and Random Forest,” 2020, hlm. 389–429. doi: 10.1007/978-3-030-31150-6_13.
- [23] A. Khoirunnisa dan N. G. Ramadhan, “Improving malaria prediction with ensemble learning and robust scaler: An integrated approach for enhanced accuracy,” *JURNAL INFOTEL*, vol. 15, no. 4, hlm. 326–334, Nov 2023, doi: 10.20895/infotel.v15i4.1056.
- [24] Abe, T. PERCLOS-based technologies for detecting drowsiness: Current evidence and future directions. *Sleep Adv.* **2023**, 4, zpad006.
- [25] Liu, G., Wu, K., Lan, W., & Wu, Y. (2025). YOLO-FDCL: Improved YOLOv8 for Driver Fatigue Detection in Complex Lighting Conditions. *Sensors*, 25(15). <https://doi.org/10.3390/s25154832>