

# Pembuatan Sistem Presensi Berbasis Pengenalan Wajah Dengan Metode You Only Look Once

## Version 8

### (Studi Kasus: Event Organizer SHAF Management)

Muhamad Saiful Adim<sup>1</sup>, Andi Iwan Nurhidayat<sup>2</sup>

Program Studi D4 manajemen Informatika, Universitas Negeri Surabaya  
Jl. Prof. Moch Yamin, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231

<sup>1</sup>muhamad.19009@mhs.unesa.ac.id

<sup>2</sup>andy134k5@unesa.ac.id

**Abstrak**—Kemajuan dalam bidang teknologi sangatlah cepat dari tahun ke tahun terjadi banyak inovasi yang diciptakan dalam bidang teknologi berbasis face detection dan objek detection pada kehidupan manusia dalam mendeteksi objek wajah. Saat pada zaman terjadinya virus covid 19 teknologi berbasis face detection banyak dipakai dalam melakukan mendeteksi objek wajah apakah objek wajah menggunakan masker atau tidak. Saat setelah zaman virus covid 19 selesai maka perkembangan face detection semakin berkembang maka perkembangan tersebut dapat diterapkan dalam penggunaan sistem presensi yang sangat efektif dalam Perusahaan dikarenakan sebelum ada sistem presensi Perusahaan masih melakukan pencatatan berbasis manual dengan mendata tanganin buku presensi karyawan karena sebab itu metode You Only Look Once (YOLO) digunakan dalam pengenalan wajah dengan tingkat akurasi yang cukup baik yang akan diterapkan di Event Organizer SM dalam melakukan presensi secara cepat di lingkungan event. Hasil yang didapatkan bahwa hasil pendeteksi wajah ditemukan dengan hasil nilai yang dimunculkan dengan nilai Precision 92 %, hasil Recall 77% dan hasil mAP59 sebesar 86% yang dilakukan dengan hasil baik untuk melakukan pendeteksian wajah kru SHAF Management (SM). Progress in the field of technology is very fast, from year to year there are many innovations created in the field of technology based on face detection and object detection in human life in detecting facial objects. During the era of the Covid-19 virus, face detection-based technology was widely used to detect facial objects, whether the facial object was wearing a mask or not. When the era of the Covid 19 virus is over, the development of face detection continues to develop, this development can be applied in the use of a very effective presence system in the Company because before there was a presence system, the Company still carried out manual-based records by registering employee attendance books because of that, the You Only method Look Once (YOLO) is used in facial recognition with a fairly good level of accuracy which will be applied in Event Organizer SM to make a quick presence in the event environment. The results obtained were that the face detection results were found with a Precision value of 92%, a Recall result of 77% and an mAP59 result of 86% which was carried out with good results for detecting the faces of the SHAF Management (SM) crew.

**Kata kunci**—YOLO, face detection, object detection, presensi

#### I. PENDAHULUAN

Perkembangan teknologi zaman sekarang berkembang cukup cepat dan juga semakin canggih. Banyak tercipta hal-hal yang terjadi sehingga menjadi mutakhir dan dapat membantu manusia dalam kehidupan sehari-hari. Misalkan di jaman dulu kita harus menggambar secara manual dengan kanvas untuk mendapatkan hasil gambar seseorang, sekarang bisa menggunakan kamera dalam permasalahan tertentu seperti identifikasi personal dalam dunia kerja, contohnya dalam sistem presensi. Teknologi ternyata mampu memberikan Solusi dalam permasalahan presensi menggunakan fitur pengenalan wajah dengan tingkat akurasi tinggi, misalnya SSD.

Single Shot Detector (SSD) merupakan metode pengenalan objek pada suatu objek gambar menggunakan single deep neural network sekaligus menjadi algoritma deteksi objek yang populer dikarenakan mudah digunakan dengan hasil akurasi yang baik. Metode SSD membutuhkan dataset yang cukup banyak dalam kinerja dari Face Recognition seperti deep neural network. Oleh karena itu, penelitian ini menggunakan metode YOLO yang membutuhkan lebih sedikit dataset yang dipakai.

You Only Look Once (YOLO) merupakan algoritma yang dikembangkan untuk mendeteksi objek secara real-time. Sistem pendeteksi dilakukan menggunakan repurpose classifier atau localizer untuk melakukan pendeteksi[1]. model diterapkan pada sebuah citra di beberapa lokasi. Berdasarkan metode di atas maka pendeteksi wajah menggunakan kamera webcam berbasis wajah menggunakan YOLO yang dapat mendeteksi wajah secara akurat.

SHAF Management ingin mendirikan kantor baru dalam melakukan aktivitas dalam pembuatan event jejepangan di kota Surabaya. Maka SHAF Management harus membutuhkan sistem presensi yang efisien dalam

manajemen kehadiran kru SHAF Management didalam Perusahaan secara otomatis dan akurat.

## II. LANDASAN TEORI

### A. Penelitian Terdahulu

1) Susanti dkk, 2023 berjudul "Sistem Absensi Mahasiswa Berbasis Pengenalan Wajah Menggunakan Algoritma YOLOv5". Hasil dari penelitian tersebut menunjukkan bahwa Sistem Absensi Mahasiswa Universitas Bina Insan berbasis pengenalan wajah menggunakan algoritma YOLOv5 menghasilkan sebuah sistem yang menampilkan data kehadiran Mahasiswa Fakultas Teknik sesuai jam perkuliahan[2].

2) Zophie & Himawan, 2022 berjudul "Implementasi algoritma you only look once (YOLO) menggunakan web camera untuk mendeteksi objek statis dan dinamis" Menurut Zophie dan Himawan, akurasi objek yang sebesar 69,56%, dengan kesalahan klasifikasi 26,09%, dan ketelitian 94,12%, dipengaruhi oleh intensitas cahaya yang mempengaruhi nilai matrix RGB dari webcam[3].

3) Khairunnas dkk, 2021 berjudul Pembuatan modul deteksi objek manusia menggunakan metode YOLO untuk mobile Robot. Hasil menunjukkan bahwa YOLO memproses 904 gambar dalam 116 detik dengan akurasi 87,03%, mampu mendeteksi single object dan multiple object pada berbagai sudut dan jarak[4].

4) Sauqi, 2022 berjudul Deteksi Kendaraan Menggunakan Algoritma Yolo Only Look Once (YOLO) V3. Penelitian ini menghasilkan dua jaringan deteksi kendaraan menggunakan algoritma YOLO V3 dengan skenario pelabelan kelas yang berbeda berdasarkan sudut pengambilan citra. Dari 10 frame uji, diperoleh nilai performa tinggi dengan Precision sempurna, Recall di atas 70%, dan F1 Score di atas 80%.

### B. Analisis Penelitian

"Sebagai besar dalam penelitian terdahulu yang menyoroti dalam penggunaan algoritma YOLO terhadap presensi yang telah dibuat terdahulu, hasil penelitian penulis memiliki penambahan fitur dan metode yang lebih lanjut secara konsisten dalam pembuatan sistem presensi dengan hasil berbeda dengan hasil penelitian terdahulu dengan fitur yang lebih baik." Dalam penelitian terdahulu dengan penelitian penulis sekarang, terjadi perbedaan dari algoritma YOLO yang digunakan perbedaan dari penelitian terdahulu versi 5 dan penulis menggunakan versi 8. Hasil penelitian terdahulu hanya membaca akurasi dalam menebak orang dan hasil penelitian penulis akan menghasilkan penyimpanan foto dan tercatat dengan pencatatan berbentuk excel dalam presensi yang dilakukan 1 hari penuh.

### C. Dasar Teori

1) Citra Digital: Citra Digital merupakan gambar dua dimensi yang dapat ditampilkan dalam layar komputer sebagai distrik nilai digital dalam berbentuk pixel dengan tinjauan secara matematis, Arti dari kata Citra adalah sebuah

fungsi dari sebuah kontinum intensitas cahaya dalam bidang berbasis dua dimensi[5].

2) Pengolahan Citra: Pengolahan Citra merupakan keberhasilan sebuah proses manipulasi digital dari gambar visual dalam meningkatkan kualitas gambar, dengan mengekstraksi informasi yang berguna, atau mengubah citra agar lebih sesuai dengan kebutuhan tertentu. Tujuan utama dari pengolahan citra adalah mendapatkan informasi yang lebih bermakna dari gambar-gambar secara visual, seperti meningkatkan ketajaman, mengurangi noise, meningkatkan kontras, atau mengidentifikasi objek dalam gambar[6].

3) MAP (Mean Average Precision): MAP merupakan metrik yang banyak digunakan dalam mendeteksi objek yang menggabungkan presisi dan penarikan untuk mengevaluasi keakuratan data model. ini dalam menghitung dari presisi rata-rata untuk setiap kelas dan kemudian melakukan perhitungan secara rata-rata dari semua kelas[7].

4) Precision: *Precision* merupakan pengukuran dari akurat dari prediksi yang benar dengan menunjukkan rasio prediksi yang benar dari jumlah total prediksi keseluruhan[7].

5) Recall: *Recall* merupakan nilai yang mengevaluasi dari kemampuan model algoritma untuk menangkap semua contoh yang sesuai dengan kelas dalam mewakili rasio prediksi yang benar dengan jumlah total data yang di latih[7].

6) You Look Only Version 8 (YOLOv8): YOLOv8 (You Only Look Once Version 8) merupakan pembaruan algoritma YOLO yang telah diluncurkan pada tahun 2020. Dengan YOLOv8 membawa peningkatan dibandingkan dari versi sebelumnya, termasuk dalam jaringan algoritma yang lebih efisien, dan peningkatan prediksi secara multi-skala, dan sistem Anchor Free Detectio[1] yang lebih baru di YOLOv7[8].

7) Python: Python merupakan salah satu bahasa memprograman tingkat tinggi yang berfungsi berorientasi dengan objek dengan sistem dinamis. Python sangat relatif sederhana karena memerlukan data yang unik dengan berfokus pada keterbacaan. Pengembang dapat membaca dan juga dapat di jemahkan oleh kode python jauh lebih mudah dari bahasa pemrograman lain. Selain itu Python dapat mendukung dari pemakaian modul dan juga paket digunakan kembali dalam berbagai proyek.

8) OpenCV: Open Computer Vision (OpenCV) adalah sebuah library open source yang difungsikan dalam melakukan pengolahan citra. Dimaksudkan agar komputer memiliki kemampuan yang sangat mirip dengan cara pengolahan visual yang ditangkap oleh visual manusia. OpenCV telah menyediakan banyak algoritma visi komputer dasar. OpenCV juga menyediakan sebuah modul pendeteksian objek yang menggunakan metode computer vision.

9) Computer Vision: Computer Vision merupakan cabang ilmu komputer yang dikhususkan untuk memberikan kemampuan pada komputer dalam melihat dan memproses sebuah informasi visual seperti sebuah gambar dan video. Computer Vision memiliki tujuan untuk mengembangkan algoritma dan teknologi yang dapat untuk memahami, menganalisis, dan memanipulasi dari sebuah gambar dan sebuah video secara otomatis.

10) Google Collab: Collaboratory atau Collab merupakan sebuah produk yang diciptakan oleh Google Research. Collab memungkinkan semua orang dapat menulis dan menjalankan kode bahasa python arbitrer melalui website atau browser tanpa harus menggunakan spek laptop yang sangat tinggi hanya perlu akses internet saja, dan sangat cocok digunakan untuk analisis data, percobaan dalam machine learning, serta Pendidikan[9].

11) Visual Studio Code: Visual Studio Code merupakan aplikasi editor kode open source yang dikembangkan oleh perusahaan Microsoft. Visual Studio Code yang dirancang akan diberikan pengalaman atas pengeditan kode yang cepat, ringan, dan dapat disesuaikan oleh dengan kebutuhan pengguna. Dengan Visual Studio Code merupakan aplikasi penulisan kode program yang memiliki banyak bahasa pemrograman seperti C++, C#, Java, JavaScript, Python, dan masih banyak lagi bahasa pemrograman yang didukung, Visual Studio Code mendukung berbagai fitur seperti debugging, linting, pengisian kode otomatis, dan kontrol versi dengan memiliki ekstensi yang dapat diperluas untuk menyesuaikan fungsionalitas pengguna. Berdasarkan survey dari Stack Overflow, Visual Studio Code merupakan aplikasi editor program terpopuler di kalangan developer pemula dan profesional.

12) Webcam: Webcam merupakan perangkat keras atau hardware didalam laptop terbaru yang difungsikan menjadi alat merekam gambar dan suara secara digital dengan tujuan untuk melakukan kegiatan video konferensi, siaran langsung, atau digunakan sebagai panggilan video. Webcam secara umumnya dihubungkan ke komputer atau laptop secara kabel dengan menggunakan kabel usb dengan menggunakan perangkat lunak yang disediakan dalam pembelian webcam tersebut[10].

13) CUDA: CUDA atau Compute Unified Device Architecture merupakan arsitektur yang memanfaatkan GPU dengan komputasi paralel dari CUDA. Arsitektur CUDA dapat dimungkinkan dalam pengembangan perangkat lunak yang berjalan pada GPU buatan NVIDIA dengan bahasa program yang mirip dengan bahasa program C yang sudah banyak digunakan. Arsitektur CUDA memiliki tiga bagian dasar dalam mengembangkan sistem dalam memanfaatkan kemampuan komputasi secara efisien dari kartu grafik buatan NVIDIA. CUDA membagi device ke grid, block, dan thread dalam struktur yang seperti gambar 2.11. Karena ada banyak thread dalam satu blok dan jumlah block dalam satu grip dan sejumlah grid didalam satu GPU. Karena proses paralel dapat tercapai dengan menggunakan suatu arsitektur yang sangat besar dan efisien[11].

14) Flask: Flask merupakan micro-framework pada bahasa python yang dapat memungkinkan pengguna dalam membaca dan memahami semua source code. Flask memiliki tiga dependensi utama. Routing, Debugging, dan subsistem Web Server Gateway Interface (WSGI) berasal dari Werkzeug, dukungan template disediakan oleh Jinja2 dan integrasi command-line berasal dari klik. Dependensi ini ditulis oleh Armin Ronacher, pencipta Flask. Flask tidak

mendukung basis data, verifikasi format web, serta sertifikasi pengguna atau tugas lainnya[12].

15) SHAF Management: Sebuah Event Organizer jejepangan yang berfokus di kota Surabaya didirikan pada tahun 2020 dengan dibimbing oleh Andre Prasetyo Gondo dan ketua SHAF Management bernama Shafry Ardi Kurniawan. Event pertama SHAF Management berlokasi di BG Junction dengan nama event Surabaya Trinity Show pada tanggal 29 November 2020. SHAF Management sebagai salah satu Event Organizer yang terbaik di kota Surabaya dengan menyuguhkan berbagai konsep event keren.

### III. METODE PENELITIAN

#### A. Gambaran Umum Sistem

Sistem presensi berbasis pengenalan wajah dengan metode you only look once version 8 adalah sistem yang dibuat untuk melakukan pencatatan dan pemantauan kehadiran kru secara real-time dengan melakukan pencatatan melalui teknologi yang menggunakan algoritma dari biometric dalam mengidentifikasi melalui pemindaian wajah.

Sistem ini menggunakan webcam dan melalui metode YOLOv8 untuk mengidentifikasi wajah untuk memverifikasi identitas wajah. Dengan akurasi tinggi dan kemampuan untuk memproses data secara real-time, sistem ini meningkatkan efisiensi dan keamanan dalam pengelolaan kehadiran di berbagai lingkungan selain di SHAF Management.

#### B. Kebutuhan Sistem

##### 1) Perangkat Keras

Proses pembuatan sistem presensi berbasis pengenalan wajah dengan metode You Only Look Once Version 8 Dalam pengembangan sistem di rancang dari perangkat yang dipakai dalam pengembangan sistem seperti di table berikut :

TABEL I  
SPESIFIKASI PERANGKAT KERAS

| No | Nama Perangkat                    | Jenis Perangkat                       |
|----|-----------------------------------|---------------------------------------|
| 1  | Intel Core i5-7200U CPU @ 2.50GHz | Perangkat keras (Prosesor)            |
| 2  | RAM 8 GB                          | Perangkat keras (RAM)                 |
| 3  | SSD 256 GB                        | Perangkat keras (Memori penyimpanan)  |
| 4  | NVIDIA Geforce 940MX              | Perangkat keras (Kartu Grafik)        |
| 5  | Windows 10                        | Perangkat lunak (sistem operasi)      |
| 6  | NVIDIA CUDA                       | Perangkat lunak (sistem kartu Grafik) |

##### 2) Perangkat Lunak

Proses pembuatan sistem presensi berbasis pengenalan wajah dengan metode You Only Look Once Version 8 Dalam pengembangan sistem di rancang menggunakan bantuan aplikasi yang memudahkan dalam pengembangan sistem seperti di table berikut:

TABEL II  
PERANGKAT LUNAK YANG DIGUNAKAN

| No | Nama aplikasi yang digunakan | Fungsi aplikasi yang digunakan                                                                                      |
|----|------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 1  | Visual Studio Code           | Untuk menulis kode program dari sistem presensi.                                                                    |
| 2  | Anaconda                     | Untuk membuat sistem environment dalam menjalankan sistem.                                                          |
| 3  | Google Collab                | Untuk melakukan proses pelatihan algoritma YOLOv8.                                                                  |
| 4  | Labelling                    | Untuk melakukan label dalam melakukan mengidentifikasi kelas dalam wajah untuk dijadikan bahan pelatihan algoritma. |
| 5  | Command Prompt               | Untuk memberikan perintah menjalankan proses pengaktifan sistem presensi.                                           |
| 6  | Xampp                        | Untuk membuat database untuk dimasukkan dalam sistem presensi.                                                      |
| 7  | Microsoft Edge               | Program browser untuk menampilkan sistem presensi.                                                                  |
| 8  | Panda                        | Sistem untuk membuat pencatatan dari sistem presensi dengan output data excel.                                      |
| 9  | Shapely                      | Untuk menampilkan proses bounding box untuk zona presensi.                                                          |
| 10 | OpenCV                       | Untuk mengolah tampilan dari kode python.                                                                           |
| 11 | Flask                        | Untuk mengubah kode program berbasis python dapat digunakan menjadi sebuah website.                                 |
| 12 | CUDA                         | Untuk memudahkan proses pengolahan penangkapan objek wajah dapat di tangkap dan digunakan secara efektif.           |

### C. Tahapan Penelitian

Penelitian akan melewati beberapa tahapan dalam Perumusan Masalah, Landasan Teori, Pengumpulan Data, Pelatihan Algoritma, Rancangan sistem akan diuji, Implementasi Sistem dan Pengujian Sistem.

### D. Uraian Penelitian

#### 1. Perumusan Masalah

Pada proses tahapan perumusan masalah, penulis melakukan pencarian permasalahan yang akan diangkat menjadi tema penelitian ini, tema yang akan diangkat adalah Presensi Berbasis Pengenalan Wajah, didasarkan permintaan ketua SHAF Management dalam pembangun sistem presensi kru dalam kebutuhan SHAF Management, bagaimana cara membangun modal sistem dalam presensi yang berbasis pengenalan objek wajah dengan efektif dan tingkat akurasi pengenalan objek wajah baik. Pada tahapan ini akan ditentukan dari tujuan penelitian tersebut adalah pembuatan model sistem presensi berbasis pengenalan wajah yang akan dibuat.

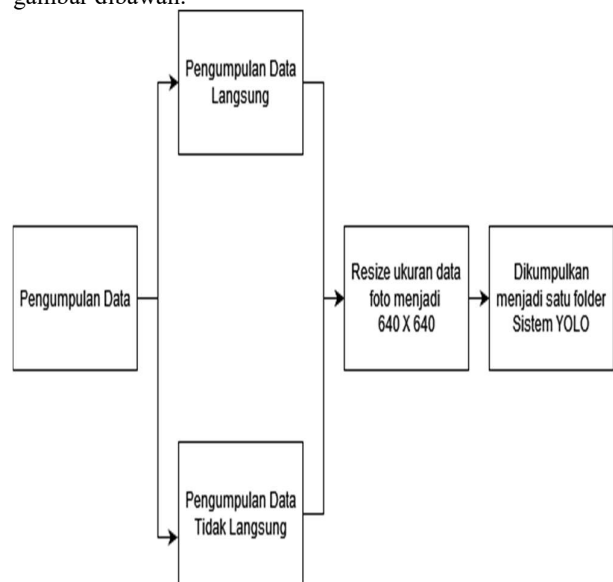
#### 2. Landasan Teori

Pada proses tahapan dari landasan teori dalam mengenai penelitian terdahulu yang akan dipelajari oleh penulis dalam Presensi Pengenalan Wajah. Selanjutnya akan dilakukan pembuatan kajian Pustaka dalam referensi

penelitian terdahulu yang berkaitan dengan penelitian yang akan dipakai oleh penulis. Untuk mempelajari dalam mengolah teori yang berkaitan dengan penelitian yang sesuai dengan metode yang dipakai di penelitian tersebut. Penulis melakukan pengumpulan berbagai Pustaka yang dicari melalui berbagai macam sumber yang diambil dalam berbagai situs yang berhubungan dalam mempelajari teori yang berkaitan seperti, Video youtube yang membahas mengenai teori YOLOv8 dari channel Indonesia hingga channel luar negeri, jurnal laporan Skripsi dan laporan Tugas Akhir yang membahas penerapan YOLO. Jurnal, laporan skripsi dan laporan Tugas Akhir didapatkan melalui situs Google Scholar.

#### 3. Pengumpulan Data

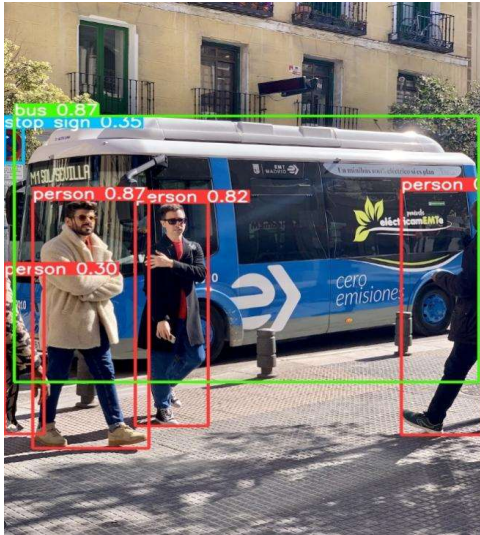
Sebelum melakukan pembuatan sistem maka dilakukan pengumpulan data terlebih dahulu secara langsung atau tidak langsung dalam pengambilan data objek wajah yang akan di training. pengumpulan data secara langsung menggunakan metode pemotretan wajah secara langsung ditempat terhadap wajah orang yang akan ditrain data foto objek tersebut. Tahapan pengumpulan data tidak langsung menggunakan foto yang telah dikumpulkan di dalam grup Line SHAF Management. Setelah mendapatkan data yang akan train maka dikelola dulu seperti alur di gambar dibawah.



Gambar 1 Alur Pengumpulan data

#### 4. Pelatihan Algoritma

Tahapan Pelatihan algoritma akan menggunakan pelatihan terhadap algoritma YOLOv8 dengan menggunakan Ultralytics dan diproses melalui Google Colab untuk melakukan pelatihan Algoritma. Dalam proses pelatihan algoritma harus melakukan penginstalan algoritma YOLOv8 di dalam google colab dengan data yang telah di sediakan oleh Ultralytics secara gratis untuk mengetahui apakah algoritma YOLOv8 sudah terinstall di google colab dengan benar. Dalam pengujian dalam algoritma YOLOv8 dengan menampilkan gambar bus dibawah ini.

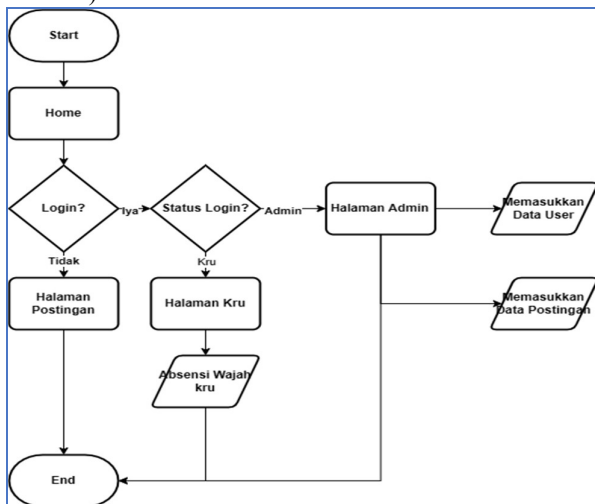


Gambar 2 Gambar Hasil Dari YOLOv8

##### 5. Rancangan Sistem Akan Diuji

Proses Rancangan Sistem yang akan diuji harus di sesuaikan dengan sistem yang akan diterapkan secara langsung dengan membuat desain flowchart sistem dan low fidelity sistem yang akan di uji.

###### 1) Flowchart Alur Sistem



Gambar 3 Flowchart Alur Sistem

##### 6. Implementasi Sistem

Proses implementasian sistem dengan penggunaan metode YOLOv8 untuk melakukan presensi pengenalan wajah dalam sistem. Dengan menggabungkan sistem presensi berbasis YOLOv8 dengan Flask sebagai framework website yang berbasis python.

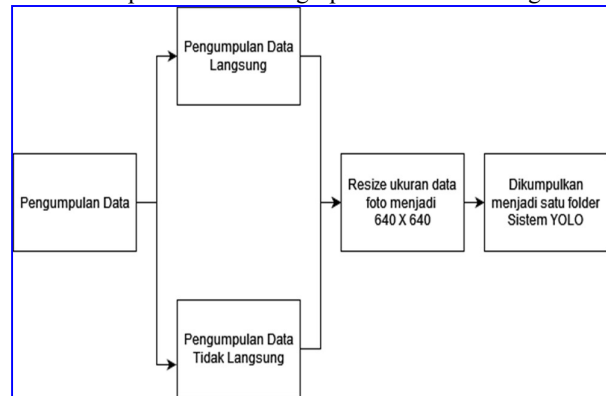
##### 7. Pengujian Sistem

Setelah sistem selesai dikembangkan maka sistem akan dilakukan pengujian oleh penulis apakah dapat melakukan pencatatan dan verifikasi menggunakan wajah yang telah dilatih dan mendapatkan hasil percobaan dalam performa dalam penangkapan objek.

#### IV. HASIL DAN PENELITIAN

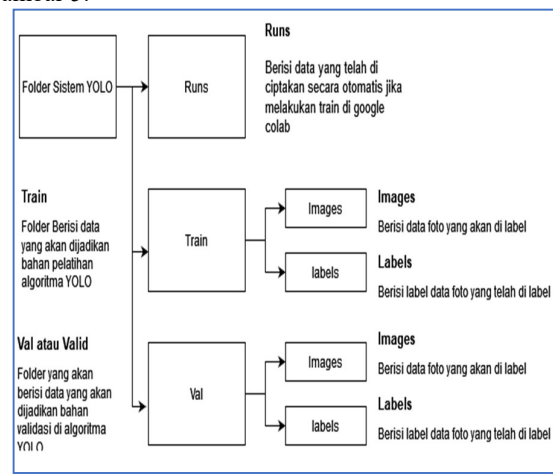
##### A. Pengumpulan Data

Dalam pembuatan sistem dilakukan pengumpulan data secara langsung dan secara tidak langsung dalam pengambilan data objek wajah yang akan di training. pengumpulan data secara langsung menggunakan metode pemotretan wajah secara langsung ditempat terhadap wajah orang yang akan di training data foto objek tersebut. Tahapan pengumpulan data tidak langsung menggunakan foto yang telah dikumpulkan di dalam grup Line SHAF Management.



Gambar 4 Alur Pengumpulan Data

Pada Gambar 4 menjelaskan saat melakukan pengumpulan data foto objek wajah dengan memisahkan data foto objek yang akan di training dan data foto objek yang akan dijadikan bahan validasi dalam 1 folder sistem yang akan dimasukkan ke dalam folder YOLOv8 seperti di Gambar 5.



Gambar 5 Penjelasan Folder Algoritma YOLOv8

Pada Gambar 5 menjelaskan Dimana folder YOLO berisi dengan folder Runs, Train dan Val dalam folder YOLO. Folder Runs akan terisi otomatis jika selesai melakukan training dengan algoritma YOLO. Folder Train berisi tentang data objek wajah yang akan diberi label menggunakan sistem labeling akan disimpan di data labels berisi koordinat bounding box yang dibuat di labeling dengan id class yang telah dibuat.

## B. Pelabelan Objek

Setelah pengumpulan data, dilanjutkan dengan pelabelan objek pada data yang telah dikumpulkan di dataset. Dikarenakan pada sistem ini menggunakan algoritma YOLO v8, maka pelabelan harus dilakukan dengan format YOLO dikarenakan di pelabelan menggunakan LabelImg harus disesuaikan dengan format YOLO dikarenakan LabelImg format Default nya adalah Poscol VOC. Setelah format telah diganti menjadi YOLO maka dilanjutkan dengan proses Pelabelan Objek. Proses pelabelan objek wajah dilakukan dari bagian dahi hingga dagu. Proses pelabelan pada penelitian ini menggunakan aplikasi LabelImg (Tzutalin,2015) dengan format anotasi YOLO. Hasil pelabelan disimpan penulis di pada folder YOLO di folder labels di images dan valid. Berikut adalah Gambar 10 dalam proses pelabelan objek wajah dan Gambar 6 merupakan isi file dari data txt yang didapatkan dari setelah selesai proses pelabelan.



Gambar 6 Proses Pelabelan Objek

adim1.txt - Notepad  
File Edit Format View Help  
0 0.610156 0.412500 0.204687 0.425000

Gambar 7 Hasil dari labeling

Pada Gambar 7 terdapat hasil isi dari data txt setelah proses pelabelan objek yaitu :

1. Nomer pertama merupakan nomer kelas yang di tentukan oleh sistem yang disesuaikan dengan list kelas yang telah di tentukan. Penjelas kelas seperti berikut :

TABEL III  
Tabel Kelas ID

| ID | Nama    | ID | Nama   | ID | Nama   |
|----|---------|----|--------|----|--------|
| 0  | Adim    | 10 | Dhani  | 20 | Riyud  |
| 1  | Alvin   | 11 | Dito   | 21 | Sevin  |
| 2  | Arame   | 12 | Fauzan | 22 | Shafry |
| 3  | Arie    | 13 | Fiqi   | 23 | Udin   |
| 4  | Atok    | 14 | Mori   |    |        |
| 5  | Bima    | 15 | Nopal  |    |        |
| 6  | Bulan   | 16 | Qolbi  |    |        |
| 7  | Chandra | 17 | Rafi   |    |        |
| 8  | Chita   | 18 | Reyhan |    |        |
| 9  | Daniel  | 19 | Rifqi  |    |        |

2. Nomer kedua dan nomer ketiga menjelaskan koordinat x dan y objek yang telah di label.

3. Nomer empat dan nomer lima yang menjelaskan lebar dan tinggi objek yang telah di label.

## C. Split Data

Split data dilakukan menggunakan website Roboflow dataset yang sudah dilakukan pelabelan maka akan dilakukan resize dan pembagian 3 jenis data yaitu 80 % data train, 13 % data valid dan 7 % data test. untuk memudahkan penulis dalam pembagian data label secara cepat tanpa harus memilah satu-satu.



Gambar 8 Split Data Dataset YOLOv8

## D. Pelatihan Algoritma

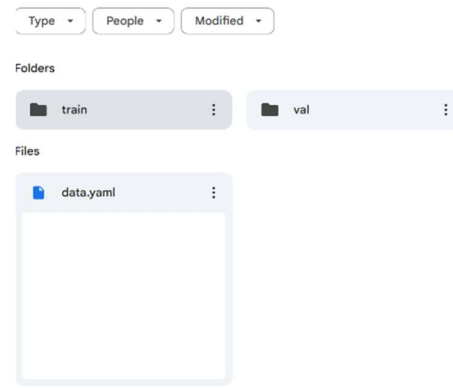
Setelah melakukan pelabelan data objek wajah dan pembagian data objek wajah, maka dilanjutkan pelatihan algoritma YOLO bertujuan untuk membuat sistem YOLO dapat menyesuaikan oleh data objek wajah yang dikumpulkan dan meningkatkan performa algoritma dalam memprediksi klasifikasi objek wajah. Pelatihan algoritma dilakukan dengan menggunakan platform Google Collaboratory karena mudah digunakan dan tidak membebani laptop penulis. Sebelum melakukan pelatihan algoritma YOLO, terlebih dahulu sebelum melakukan pelatihan algoritma YOLO dilakukan pemasangan sistem algoritma YOLOv8 seperti di Gambar 9.



Gambar 9 Pemasangan Sistem YOLOv8

Setelah, melakukan pemasangan Algoritma YOLOv8 selanjutnya memasukkan data objek wajah yang telah di labeling ke dalam google drive penulis seperti Gambar 10.

My Drive > YOLOV8\_SM >



Gambar 10 Input Data Ke Dalam Google Drive

Selanjutnya, akan dilakukan pemasangan melalui Google Drive dengan *virtual machine* dari *runtime google colab* seperti Gambar 11.

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Gambar 11 Kode Program Mount Google Drive

Setelah melakukan Program Mount Google Drive dilanjutkan dengan melakukan pengecekan Terhadap penginstall YOLOv8 apakah berhasil atau gagal seperti Gambar 12.

```
import ultralytics
ultralytics.checks()

Ultralytics YOLOv8.2.20 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 28.9/78.2 GB disk)
```

Gambar 12 Pengecekan Sistem Algoritma YOLOv8

Setelah mengecek algoritma YOLOv8 maka melakukan pengambilan data YOLO seperti Gambar 13.

```
[ ] from ultralytics import YOLO
```

Gambar 13 Pengambilan Algoritma YOLOv8

Setelah pengambilan data YOLO dilanjutkan dengan pengambilan model sistem algoritma YOLO sesuai Gambar 14.

```
[ ] model = YOLO("yolov8n.pt")

Downloading https://github.com/ultralytics/assets/releases/download/v8.0.0/yolov8n.pt to 'yolov8n.pt'...
100% 49.7M/49.7M [00:00<00:00, 319MB/s]
```

Gambar 14 Melakukan Pengambilan Model YOLOv8

Di lanjutkan dalam meminta data.yaml dari algoritma YOLOv8 seperti Gambar 15.

```
[ ] !touch data.yaml
```

Gambar 15 Mendapatkan data.yaml

Setelah mendapatkan data.yaml maka melakukan konfigurasi untuk dapat di kelola oleh google colab dengan mengisi seperti di Gambar 16.

```
data.yaml X
1 train: /content/drive/MyDrive/YOLOv8_SM/train
2 val: /content/drive/MyDrive/YOLOv8_SM/val
3
4 nc: 24
5 names: ['Adim', 'Alvin', 'Arame', 'Arie', 'Atok', 'Bima',
6         'Bulan', 'Chandra', 'Chita', 'Daniel', 'Dhani', 'Dito',
7         'Fauzan', 'Fiqui', 'Mori', 'Nopal', 'Qolbi', 'Rafi', 'Reyhan',
8         'Rifqi', 'Riyud', 'Sevin', 'Shafry', 'Udin']
9
```

Gambar 16 Setting Data.Yaml

Setelah konfigurasi Data.yaml selesai, maka dilanjutkan pada tahap training data melalui algoritma YOLO dengan bantuan sistem Ultralytics seperti di Gambar 17 berikut:

```
!yolo detect train data=/content/drive/MyDrive/YOLOv8_SM/data.yaml

Ultralytics YOLOv8.2.20 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=/content/drive/MyDrive/YOLOv8_SM/data.y
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100% 755k/755k [00:00<00:00, 83.5MB/s]
Overriding model.yaml nc=80 with nc=24

from n params module arguments
0 -1 1 464 ultralytics.nn.modules.conv.Conv [3, 16, 3, 2]
1 -1 1 4672 ultralytics.nn.modules.conv.Conv [16, 32, 3, 2]
2 -1 1 7360 ultralytics.nn.modules.block.C2f [32, 32, 1, True]
3 -1 1 18560 ultralytics.nn.modules.conv.Conv [32, 64, 3, 2]
4 -1 2 49664 ultralytics.nn.modules.block.C2f [64, 64, 2, True]
5 -1 1 73984 ultralytics.nn.modules.conv.Conv [64, 128, 3, 2]
6 -1 2 197632 ultralytics.nn.modules.block.C2f [128, 128, 2, True]
7 -1 1 295424 ultralytics.nn.modules.conv.Conv [128, 256, 3, 2]
8 -1 1 460288 ultralytics.nn.modules.block.C2f [256, 256, 1, True]
9 -1 1 164608 ultralytics.nn.modules.block.SPPF [256, 256, 5]
10 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
11 [-1, 6] 1 0 ultralytics.nn.modules.conv.Concat [1]
12 -1 1 148224 ultralytics.nn.modules.block.C2f [384, 128, 1]
13 -1 1 0 torch.nn.modules.upsampling.Upsample [None, 2, 'nearest']
14 [-1, 4] 1 0 ultralytics.nn.modules.conv.Concat [1]
15 -1 1 37248 ultralytics.nn.modules.block.C2f [192, 64, 1]
```

Gambar 17 Training Algoritma YOLOv8.

#### E. Hasil Pelatihan Algoritma YOLOv8

Dalam melakukan pelatihan algoritma YOLOv8, Penulis melakukan pelatihan algoritma dengan epoch berbeda dengan ingin menemukan hasil terbaik dari pelatihan algoritma menggunakan bantuan Google colab. Penulis juga melakukan pelatihan algoritma dengan Epoch 100 dengan menunjukkan hasil nilai Precision 92 persen, nilai recall 77 persen dan nilai mAP50 86 persen. hasil seperti di Tabel IV.hasil seperti tabel berikut:

TABEL IV  
Hasil Data Pelatihan Algoritma Dengan Epoch 100

| Hasil Data pelatihan algoritma berbasis YOLO  |        |          |           |        |       |
|-----------------------------------------------|--------|----------|-----------|--------|-------|
| Epoch yang dilakukan kepada data sebanyak 100 |        |          |           |        |       |
| Class                                         | Images | Intances | Precision | Recall | mAP50 |
| All                                           | 65     | 226      | 0.928     | 0.775  | 0.862 |
| Adim                                          | 65     | 8        | 0.952     | 0.875  | 0.89  |
| Alvin                                         | 65     | 13       | 0.724     | 0.769  | 0.82  |
| Arame                                         | 65     | 3        | 0.985     | 1      | 0.995 |
| Arie                                          | 65     | 20       | 0.929     | 0.85   | 0.929 |
| Atok                                          | 65     | 1        | 0.88      | 1      | 0.995 |
| Bima                                          | 65     | 13       | 0.862     | 0.484  | 0.732 |
| Bulan                                         | 65     | 10       | 0.861     | 0.7    | 0.805 |

|         |    |    |       |       |       |
|---------|----|----|-------|-------|-------|
| Chandra | 65 | 6  | 1     | 0.739 | 0.851 |
| Chita   | 65 | 5  | 0.912 | 0.8   | 0.878 |
| Daniel  | 65 | 7  | 1     | 0.92  | 0.995 |
| Dhani   | 65 | 7  | 0.996 | 0.857 | 0.873 |
| Dito    | 65 | 17 | 1     | 0.65  | 0.969 |
| Fauzan  | 65 | 10 | 0.962 | 1     | 0.995 |
| Fiqi    | 65 | 8  | 0.948 | 0.75  | 0.842 |
| Mori    | 65 | 6  | 1     | 0.563 | 0.751 |
| Nopal   | 65 | 12 | 0.911 | 0.917 | 0.928 |
| Qolbi   | 65 | 10 | 0.908 | 0.993 | 0.922 |
| Rafi    | 65 | 10 | 0.943 | 0.7   | 0.968 |
| Reyhan  | 65 | 5  | 0.927 | 0.8   | 0.808 |
| Rifqi   | 65 | 8  | 1     | 0.615 | 0.731 |
| Riyud   | 65 | 6  | 0.774 | 0.579 | 0.705 |
| Sevin   | 65 | 18 | 0.916 | 0.778 | 0.81  |
| Shafry  | 65 | 20 | 0.947 | 0.775 | 0.742 |
| Udin    | 65 | 3  | 0.946 | 0.875 | 0.749 |

#### F. Hasil Prediksi Algoritma Terhadap Objek Wajah

TABEL V  
Tabel Hasil Pelatihan Algoritma Dengan Epoch 100

| No | Gambar Yang Ditest                                                                  | Class Yang Ditampilkan | Class Yang Sebenarnya |
|----|-------------------------------------------------------------------------------------|------------------------|-----------------------|
| 1  |   | Arie                   | Arie                  |
| 2  |  | Arie                   | Arie                  |
| 3  |  | Arie                   | Arie                  |
| 4  |  | Shafry                 | Shafry                |

|   |                                                                                      |                      |                      |
|---|--------------------------------------------------------------------------------------|----------------------|----------------------|
| 5 |    | Sevin                | Sevin                |
| 6 |    | Sevin                | Sevin                |
| 7 |    | Shafry               | Shafry, Rifqi        |
| 8 |   | Chandra, Nopal, Arie | Chandra, Nopal, Arie |
| 9 |  | Adim                 | Adim                 |

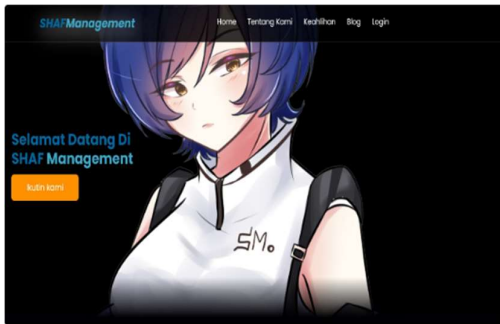
Di tabel diatas menjelaskan hasil dari pengetesan pendeteksi wajah dengan mencoba dengan cara membaca foto yang telah dimasukkan untuk di ujicoba.dengan hasil sedikit kesalahan mendeteksi kelas berada di nomer 7 tetap melakukan kesalahan membaca lebih buruk yaitu tidak terbaca wajah Rifqi dan tidak menampilkan kelas data sama sekali.

#### G. Hasil Penerapan Sistem YOLOv8 Berbasis Website

Setelah melakukan pelatihan algoritma YOLOv8 maka dilanjutkan ke tahap pembuatan sistem presensi berbasis YOLOv8 dengan bantuan flask untuk dikembangkan dalam pemrograman website. Di akan menjelaskan hasil desain

pembuatan program berbasis website dengan penggunaan Flask.

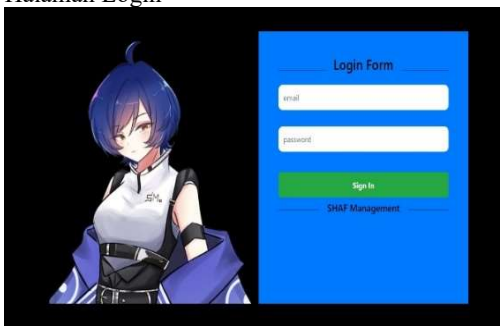
- Halaman Home



Gambar 18 Halaman Home

Halaman Home sebagai halaman awal dari sistem presensi yang berisi Home, Tentang Kami, Event Kami, Blog dan Login sebelum halaman sistem presensi SHAF Management.

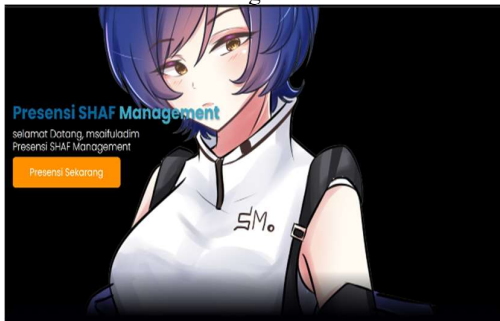
- Halaman Login



Gambar 19 Halaman Login

Halaman Login berfungsi sebagai tempat mengakses fitur presensi yang akan dibuka melalui username dan password dari admin. Dikarenakan masih menggunakan alur 1 pintu untuk menghindari kesalahan dalam mendeteksi dan selalu digunakan oleh pengawasan admin..

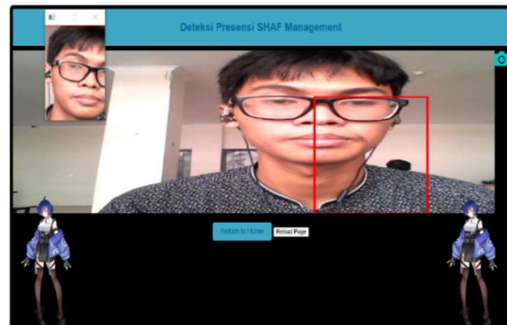
- Halaman Selamat Datang



Gambar 20 Halaman Selamat Datang

Halaman Selamat Datang berfungsi sebagai halaman penyemangat untuk admin setelah melakukan login untuk dapat mengakses halaman presensi SHAF Management.

- Halaman Presensi



Gambar 21 Halaman Presensi

Presensi berfungsi melakukan sebagai sistem presensi wajah kru SHAF Management saat memasuki kegiatan yang dibawah naungan SHAF Management untuk ditangkap data wajah kru untuk di jadikan bukti kru telah melakukan presensi. Wajah kru harus melewati atau berada di kotak warna merah yang difungsikan menjadi zona pendeteksi yang hanya mendeteksi wajah dari kru Shaf Management.

- Laporan Presensi Kru

| Class Name | Status  | Entry Time | Video Status |
|------------|---------|------------|--------------|
| Adim       | Present | 11:00:00   | Entrance     |
|            |         |            |              |
|            |         |            |              |
|            |         |            |              |

Gambar 22 Laporan Presensi Kru

Di Gambar 22 merupakan hasil pencatatan kehadiran Kru yang telah dicatat oleh sistem dan dikelola sebagai laporan tercatat melalui Excel.

- Hasil Presensi Kru



Gambar 23 Hasil Presensi Kru

Di Gambar 23 merupakan hasil penangkapan objek wajah yang telah melakukan presensi dengan wajah dengan hasil gambar objek wajah yang sesuai kelas di name dan status objek wajah apakah dia presensi saat jam datang atau saat jam pulang.

## V. KESIMPULAN

#### A. Simpulan

Dari pelatihan algoritma YOLOv8 yang sudah dilakukan oleh penulis dengan perbedaan epoch dapat ditemukan dari pelatihan algoritma YOLOv8 dengan epoch 25 menemukan bahwa data nilai yang dihasilkan dengan nilai Precision 87%, nilai recall menunjukkan nilai 75% dan nilai mAP50 menunjukkan nilai 82%. Hasil pelatihan algoritma YOLOv8 dengan epoch 50 menghasilkan nilai Precision 78%, nilai recall 80% dan nilai mAP50 84%. Hasil pelatihan algoritma YOLOv8 dengan epoch 75 menghasilkan nilai dengan nilai Precision 90%, nilai recall 75% dan nilai mAP50 85. dan pelatihan algoritma YOLOv8 dengan epoch 100 menghasilkan nilai Precision 92%, nilai recall 77% dan nilai mAP50 dengan hasil 86%. Hasil pelatihan algoritma YOLOv8 dilakukan tes pengenalan kelas dengan contoh gambar diuji dengan 2 kelompok pengetasan seperti pembacaan kelas yang benar dengan pembacaan salah atau tidak terbaca kelas dengan epoch yang berbeda-beda. Ditemukan Kesimpulan bahwa algoritma YOLOv8 yang dilatih dan digunakan tidak dapat membaca kelas yang dimana gambar yang diuji menggunakan masker, kepala dimiringkan 90 derajat dan jarak yang cukup jauh dalam pembacaan kelas. Dengan mengimplementasikan hasil pelatihan YOLOv8 kedalam sistem presensi website dengan framework Flask penulis dapat mengetahui kendala yang akan terjadi didalam proses pencatatan berbasis wajah dengan kendala yang akan terjadi. Flask memudahkan dalam menjalankan algoritma YOLOv8 yang dapat diakses melalui website.

#### B. Saran

Saran yang diajukan oleh penulis dalam penelitian selanjutnya adalah penerapan secara luas dalam penerapan algoritma YOLOv8 dapat diterapkan dalam kehidupan manusia sebagai pendeteksi api, pendeteksi orang merokok, pendeteksi masker, mendeteksi huruf Bahasa Jepang dan sebagai kamera pengaman dalam lingkungan perumahan atau apartemen dengan mengambil dataset yang cukup banyak. Dengan adanya sistem yang dibuat oleh penulis dapat memudahkan dalam pencatatan kehadiran karyawan secara cepat dan otomatis daripada pencatatan secara konvensional yang masih diterapkan hingga sekarang yang menyebabkan tidak efisien dalam pencatatan karyawan.

#### REFERENSI

- [1] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, pp. 91–99, Nov. 2023, doi: 10.34010/komputa.v12i2.10654.
- [2] L. Susanti, N. K. Daulay, and B. Intan, "Sistem Absensi Mahasiswa Berbasis Pengenalan Wajah Menggunakan Algoritma YOLOv5," *JURIKOM (Jurnal Riset Komputer)*, vol. 10, no. 2, p. 640, Apr. 2023, doi: 10.30865/jurikom.v10i2.6032.
- [3] J. Zophie, H. Himawan Triharminto, D. Elekronika, and A. Angkatan Udara, "Implementasi Algoritma You Only Look Once (YOLO)." [4] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," *Jurnal Teknik ITS*, vol. 10, no. 1, Aug. 2021, doi: 10.12962/j23373539.v10i1.61622.
- [5] M. Riadi, "Pengolahan Citra Digital - KajianPustaka." Accessed: Jun. 25, 2024. [Online]. Available: <https://www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html>
- [6] S. Sulistiyanti, F. A. Setyawan, and M. Komarudin, *PENGOLAHAN CITRA DASAR DAN CONTOH PENERAPANNYA*. 2016.
- [7] Rina, "Memahami Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-Score untuk Evaluasi Model Klasifikasi | by Rina | Medium." Accessed: Jun. 25, 2024. [Online]. Available: <https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf>
- [8] Deci Team Marketing Team, "The History of YOLO Object Detection Models from YOLOv1 to YOLOv8 | Deci." Accessed: Jun. 25, 2024. [Online]. Available: <https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/>
- [9] G. I. E. Soen, M. Marlina, and R. Renny, "Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants," *JITU : Journal Informatic Technology And Communication*, 2022, [Online]. Available: <https://api.semanticscholar.org/CorpusID:269201668>
- [10] H. Riyadi, "Pengertian WEBCAM Adalah : Fungsi dan Cara Kerja Webcam." Accessed: Jun. 25, 2024. [Online]. Available: <https://www.nesabamedia.com/pengertian-webcam-dan-fungsi-webcam/>
- [11] B. Kurniawan, T. B. Adji, and N. A. Setiawan, "Analisis Perbandingan Komputasi GPU dengan CUDA dan Komputasi CPU untuk Image dan Video Processing," 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:67404931>
- [12] M. Surya Putri, "Flask dan Cara Instalasi di Windows | by Monita Surya Putri | Medium." Accessed: Jun. 25, 2024. [Online]. Available: <https://monitas.medium.com/flask-dan-cara-instalasi-di-windows-d0ee6d429f53>