

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kegiatan magang dilaksanakan di bawah supervisi langsung Bapak Hardiyanto Saputro selaku Manager IT Infrastructure, serta mendapat arahan dan bimbingan dari tim IT Infrastructure dalam menjalankan tugas-tugas yang diberikan. Seluruh aktivitas, hasil pekerjaan, dan proses koordinasi yang berkaitan dengan pelaksanaan tugas tersebut dilaporkan serta dikomunikasikan melalui grup WhatsApp internal tim.

#### 3.2 Tugas dan Uraian Kerja Magang

Selama melaksanakan kegiatan magang di PT Porto Indonesia Sejahtera, penulis diberikan berbagai tugas yang berhubungan dengan pengembangan sistem serta dokumentasi kegiatan harian dan bulanan. Rincian tugas selama periode magang dapat dilihat pada Tabel 3.1.

Table 3.1 Waktu Pelaksanaan Magang Perusahaan

Week	Perkenalan	Membangun Sistem Deteksi objek berbasis AI	Monthly Report
1			
2		Eksplor AI untuk deteksi api	
3		Pengumpulan data yang digunakan	
4		Eksplor model yang akan digunakan	
5		Pelatihan Model Yolov8	
6		Melakukan Testing menggunakan Vidio dan Gambar	
7		Pengumpulan data person dan truck	
8		Normalisasi kelas	
9		Pelatihan Model Yolov8	

10			
11			
12			
13		Perbaiki Model	

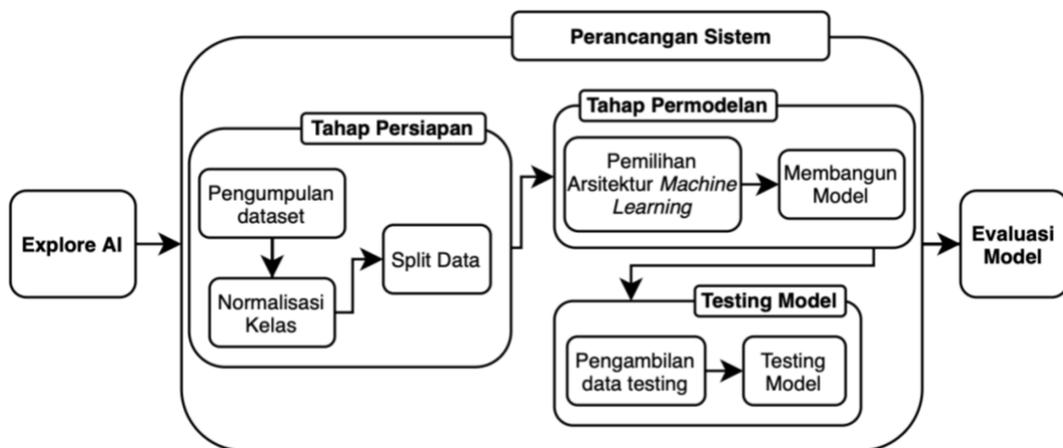
### 3.2.1 Perkenalan

Pada minggu pertama pelaksanaan magang di PT Porto Indonesia Sejahtera, penulis diperkenalkan dengan lingkungan kerja, khususnya Divisi IT Infrastruktur sebagai divisi tempat penulis ditempatkan. Selain itu, penulis juga diperkenalkan dengan beberapa divisi lainnya untuk memahami struktur organisasi dan alur kerja perusahaan secara umum.

Penulis juga diajak berkeliling untuk melihat fasilitas dan mengenal lebih jauh aktivitas yang dilakukan di perusahaan. Setelah sesi perkenalan, penulis diberikan beberapa akses yang dibutuhkan untuk mendukung pekerjaan, seperti akses jaringan Wi-Fi internal, akun email perusahaan melalui Zimbra, serta akses ke sistem manajemen perusahaan seperti Odoo dan DMS (Document Management System).

### 3.2.2 Membangun Sistem Deteksi Objek Berbasis AI

Selama masa magang, penulis mendapat tanggung jawab untuk mengerjakan project pengembangan sistem deteksi kebakaran berbasis AI dengan memanfaatkan model *object detection YOLO*. Sistem ini dirancang untuk mendeteksi keberadaan api, orang dan truck secara otomatis melalui input video dari kamera pengawas (*CCTV*). Berikut alur yang penulis lakukan dalam pengembangan system deteksi objek berbasis *AI*, yang dapat dilihat pada Gambar 3.1



Gambar 3.1 Alur dalam pengembangan system deteksi objek

Sebelum membangun sistem deteksi objek, penulis mulai dari mengeksplorasi konsep dan pendekatan *AI* yang dapat digunakan untuk mendeteksi objek, baik dari segi teori maupun praktik. Kegiatan ini meliputi studi pustaka, mengamati contoh sistem yang sudah ada, serta analisis potensi penerapan *computer vision* untuk tugas tersebut.

### 3.2.2.1 Tahap Persiapan

Setelah memahami konsep dasar, penulis melanjutkan dengan pengumpulan dataset yang diperlukan untuk pelatihan model. Namun, dalam proses pelabelan data, penulis mengalami kendala karena keterbatasan sumber daya. Oleh karena itu, penulis memutuskan untuk menggunakan dataset publik yang tersedia di platform Roboflow.com, yang telah dilabeli dan siap digunakan untuk pelatihan model. Rincian lengkap mengenai dataset dan distribusinya disajikan pada Tabel 3.2.

Table 3.2 Pemilihan dataset Gambar dan Label

Kelas	Jumlah Gambar	Jumlah Label
Api	3.550	3.550
Orang	3.550	3.550
Truk	3.550	3.550

Berikut merupakan contoh gambar dari objek api, orang dan truk, yang dapat dilihat pada Gambar 3.2.



(A)



(B)



(C)

Gambar 3.2 Contoh gambar objek: (A) Api, (B) Orang dan (C) Truk

Label merupakan informasi anotasi yang digunakan untuk menandai objek dalam sebuah gambar, sehingga model deteksi objek dapat mengenali jenis objek (kelas) dan lokasinya di dalam gambar. Dalam konteks *object detection*, label tidak hanya menunjukkan kategori objek (misalnya api, orang, atau truk), tetapi juga menentukan posisi objek tersebut melalui bounding box berupa koordinat. Label berfungsi sebagai ground truth yang digunakan pada proses pelatihan (training) untuk membantu model belajar menghubungkan citra dengan informasi yang benar. Berikut merupakan contoh dari isi label dapat dilihat pada Gambar 3.3.

```
project ll > truck > train > labels > ≡ 5f630068e794c
1
2 0 0.57734375 0.55 0.778125 0.865625
3
```

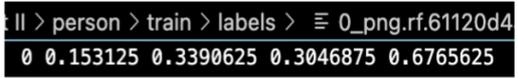
Gambar 3.3 Contoh label

Label biasanya disimpan dalam format .txt dengan struktur data yang mencakup:

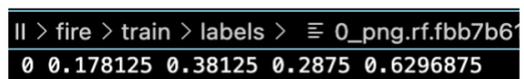
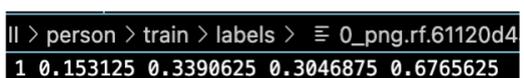
- ID kelas objek: 0
- Titik tengah bounding box (x, y): 0.57734375 0.55
- Lebar dan tinggi bounding box: 0.778125 0.865625

Setelah proses pengumpulan data selesai, ditemukan bahwa setiap kelas objek memiliki ID yang sama. Hal ini dapat menyebabkan model mengalami kebingungan dalam membedakan jenis objek. Oleh karena itu, diperlukan proses normalisasi kelas, yaitu memberikan identitas unik pada setiap jenis objek agar model dapat memahami dan membedakan data

dengan benar. Berikut merupakan contoh label sebelum dan sesudah dilakukan normalisasi kelas dapat dilihat pada Gambar 3.4 dan Gambar 3.5.

- fire → class ID 0 
- person → class ID 0 
- truck → class ID 0 

Gambar 3.4 Sebelum dilakukan normalisasi class

- fire → class ID 0 
- person → class ID 1 
- truck → class ID 2 

Gambar 3.5 Setelah dilakukan normalisasi class

Langkah selanjutnya adalah membagi data menjadi tiga bagian utama, yaitu data training, data validasi, dan data uji, untuk memastikan model dapat dilatih, dievaluasi, dan diuji secara efektif. Pembagian data dilakukan secara manual dengan rasio 70:20:10, di mana 70% digunakan untuk data training, 20% untuk data validasi, dan 10% untuk data uji. Rasio ini dipilih karena banyak penelitian sebelumnya menunjukkan bahwa pembagian tersebut mampu menghasilkan model dengan performa yang baik dan tingkat akurasi tinggi, bahkan mendekati 100% pada beberapa kasus. Pembagian dataset untuk masing-masing bagian dapat dilihat pada Tabel 3.3.

Table 3.3 Pembagian Data Training, Validasi dan Uji

Kelas	Train	Val	Test
Api	2.485	710	355
Orang	2.485	710	355
Truk	2.485	710	355

### 3.2.2.2 Tahap Permodelan

Setelah dataset tersedia dan dilakukan normalisasi kelas serta pembagian data, penulis melakukan eksplorasi terhadap beberapa model deteksi objek berbasis *deep learning*, dan memutuskan untuk menggunakan YOLOv8. Pemilihan model YOLOv8 didasarkan pada hasil studi literatur yang menunjukkan bahwa model ini memiliki tingkat akurasi tinggi dan telah banyak digunakan dalam penelitian sebelumnya untuk tugas serupa. Pelatihan dilakukan dengan ukuran gambar (image size) 640 selama 20 epoch, dan model dilatih menggunakan Google Colab.

Alasan penulis menggunakan 20 epoch adalah untuk menghindari overfitting dan mengoptimalkan waktu pelatihan, mengingat sumber daya yang digunakan terbatas. Berdasarkan studi literatur, jumlah epoch yang terlalu besar pada dataset terbatas dapat menyebabkan model belajar berlebihan (overfit) terhadap data latih, sehingga tidak generalisasi dengan baik pada data validasi dan uji. Selain itu, pada tahap ini penulis tidak lagi menerapkan augmentasi data karena dataset yang digunakan sudah melalui proses augmentasi sebelumnya, sehingga variasi data sudah cukup untuk melatih model dengan baik. Potongan code pelatihan model dapat dilihat pada Gambar 3.6.

```
[ ] #n, s, m, l, xl
!yolo task=detect mode=train model=yolov8m.pt
data=/content/drive/MyDrive/data.yaml
epochs=20
imgsz=640
```

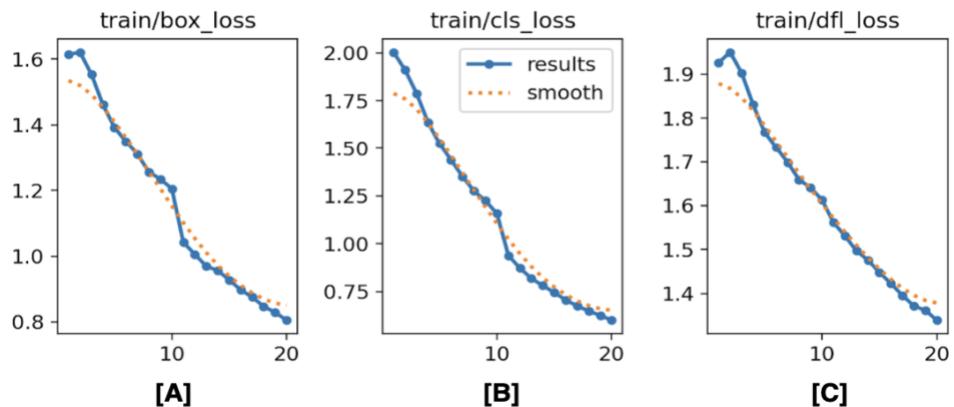
Gambar 3.6 Pelatihan model YOLOv8

Setelah proses pelatihan selesai, tahap selanjutnya adalah menganalisis performa model YOLOv8 Medium berdasarkan metrik dan nilai loss yang dihasilkan. Analisis ini mencakup tiga komponen utama, yaitu Train Loss, Validation Loss, dan Metrik Evaluasi, yang bertujuan untuk menilai sejauh mana model mampu belajar dan menggeneralisasi terhadap data.

- **Train Box Loss:** Metrik train box loss mengukur perbedaan antara bounding box yang diprediksi dengan bounding box sebenarnya dari objek pada data pelatihan. Nilai box loss yang lebih rendah menunjukkan bahwa bounding box prediksi model lebih sesuai dengan bounding box sebenarnya.
- **Train Class Loss:** Metrik train class loss mengukur perbedaan antara probabilitas kelas yang diprediksi dengan label kelas sebenarnya dari objek pada data pelatihan. Nilai class loss yang lebih rendah menunjukkan bahwa probabilitas kelas yang diprediksi model lebih sesuai dengan label kelas sebenarnya.
- **Train DFL Loss:** Metrik train DFL (Dynamic Feature Learning) loss mengukur perbedaan antara feature maps yang diprediksi dengan feature maps sebenarnya dari objek pada data pelatihan. Nilai DFL loss yang lebih rendah menunjukkan bahwa feature maps prediksi model lebih sesuai dengan feature maps sebenarnya.
- **Metrics Precision (B):** Metrik precision (B) mengukur proporsi deteksi true positive dibandingkan dengan semua bounding box yang diprediksi. Nilai precision yang lebih tinggi berarti model lebih baik dalam mengidentifikasi deteksi benar dan meminimalkan false positive.
- **Metrics Recall (B):** Metrik recall (B) mengukur proporsi deteksi true positive dibandingkan dengan semua bounding box sebenarnya. Nilai recall yang lebih tinggi berarti model lebih baik dalam mendeteksi semua objek yang benar dan meminimalkan false negative.

- Metrics mAP50 (B): Metrik mAP50 (B) mengukur mean average precision model pada berbagai kategori objek, dengan ambang intersection-over-union (IoU) sebesar 50%. Nilai mAP50 yang lebih tinggi menunjukkan bahwa model lebih baik dalam mendeteksi dan melokalisasi objek secara akurat di berbagai kategori.
- Metrics mAP50-95 (B): Metrik mAP50-95 (B) mengukur mean average precision model pada berbagai kategori objek, dengan ambang IoU berkisar antara 50% hingga 95%. Nilai mAP50-95 yang lebih tinggi menunjukkan bahwa model lebih baik dalam mendeteksi dan melokalisasi objek secara akurat di berbagai kategori dengan rentang ambang IoU yang lebih luas.

Berikut merupakan metrik dan nilai loss dari model yolov8 medium yang sudah dilatih dapat dilihat pada Gambar 3.7 (train loss), 3.8 (val loss) dan 3.9 (Metrik Evaluasi).

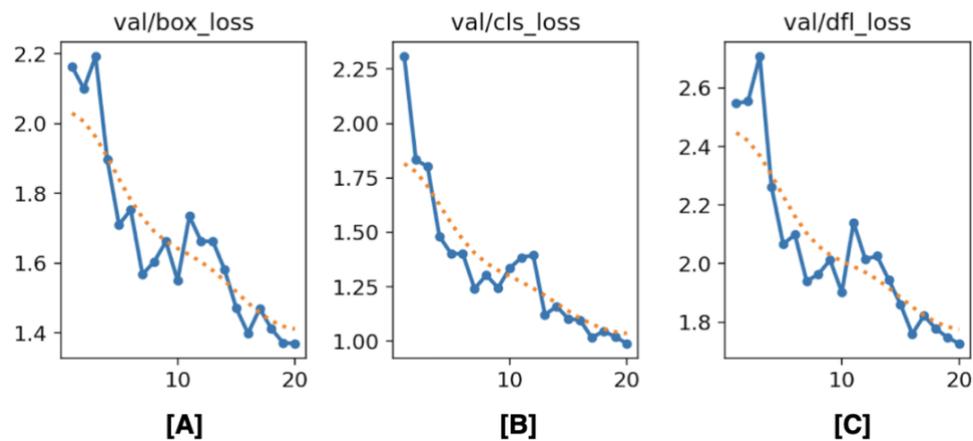


Gambar 3.7 Train Loss: [A] Box Loss [B] CLS LOSS [C] DFL Loss

Gambar 3.7 menunjukkan grafik train loss yang terdiri dari Box Loss, CLS Loss dan DFL Loss. Berikut merupakan analisis dari setiap metrik:

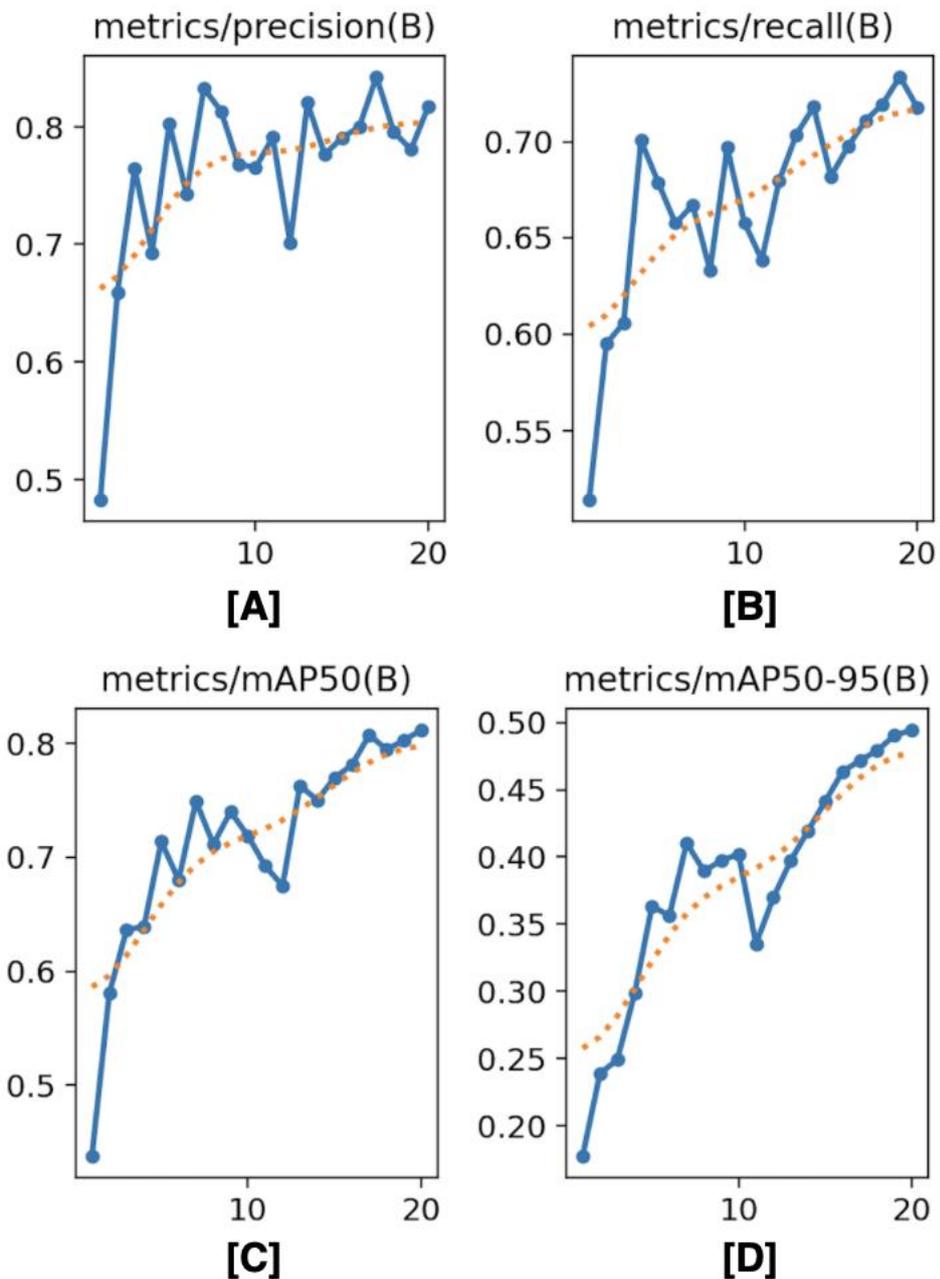
- Box Loss: Terlihat penurunan yang konsisten dari sekitar 1.6 menjadi 0.8 pada akhir pelatihan. Hal ini menunjukkan bahwa prediksi posisi bounding box semakin mendekati ground truth seiring bertambahnya epoch.

- Class Loss: Turun signifikan dari sekitar 2.0 menjadi 0.8, menunjukkan model semakin akurat dalam mengklasifikasikan objek.
- DFL Loss: Mengalami penurunan dari 1.9 menjadi 1.3, menandakan kemampuan model dalam mempelajari fitur objek semakin baik.



Gambar 3.8 Val Loss: [A] Box Loss [B] CLS IOSS [C] DFL Loss

Dapat dilihat pada Gambar 3.7 menunjukkan grafik train loss yang terdiri dari Nilai box loss, class loss, dan DFL loss pada data validasi juga menurun secara konsisten, meskipun dengan fluktuasi. Pola penurunan ini mengindikasikan bahwa model tidak mengalami overfitting yang signifikan karena tren validasi sejalan dengan tren pelatihan.



Gambar 3.9 Matrik Evaluasi: [A] Precision (B) [B] Recall (B) [C] mAP50 (B) [D] mAP50-90 (B)

Gambar 3.9 menunjukkan grafik matrik evaluasi yang terdiri dari Precision (B), Recall (B), mAP50 dan mAP50-90 (B). Berikut merupakan analisis dari setiap metrik:

- Precision (B): Mengalami peningkatan dari sekitar 0.55 pada awal pelatihan menjadi 0.80, menunjukkan model semakin baik dalam mengurangi false positive.

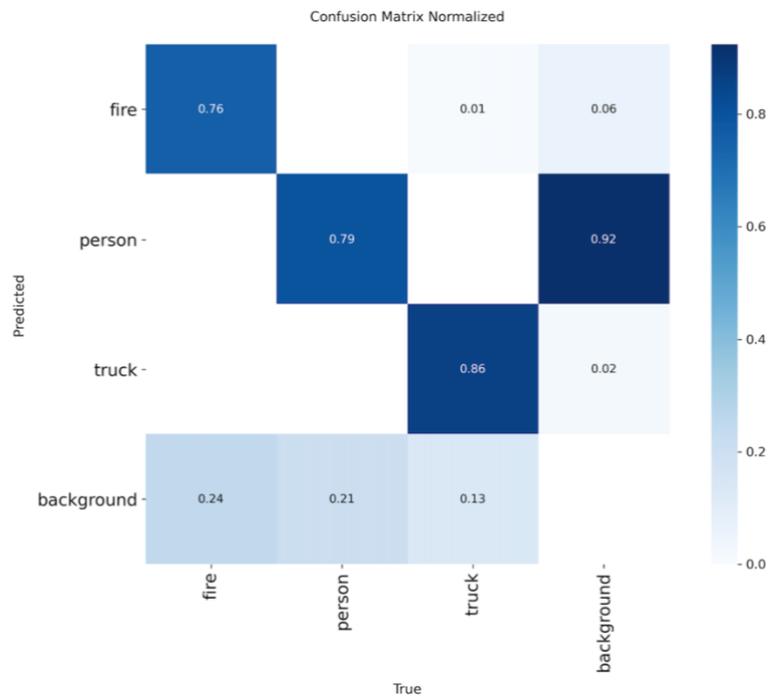
- Recall (B): Meningkat dari 0.48 menjadi 0.72, artinya model semakin mampu mendeteksi objek yang benar (true positive).
- mAP50 (B): Naik dari 0.50 menjadi sekitar 0.82, menandakan model dapat mendeteksi objek dengan tingkat akurasi yang tinggi pada ambang IoU 50%.
- mAP50-95 (B): Berhasil meningkat dari 0.20 menjadi 0.49, menunjukkan kemampuan model mendeteksi dan melokalisasi objek pada berbagai ambang IoU.

Secara keseluruhan, grafik menunjukkan tren yang positif dengan penurunan nilai loss dan peningkatan metrik evaluasi. Hal ini mengindikasikan bahwa model YOLOv8 yang digunakan berhasil belajar dengan baik dari dataset yang diberikan, tanpa indikasi overfitting yang signifikan. Hasil ini dapat menjadi dasar untuk melanjutkan ke tahap pengujian model pada data uji.

### 3.2.2.3 Evaluasi Uji Model

Data uji terdiri dari dua bagian. Bagian pertama adalah 10% dari hasil pembagian data sebelumnya, yang sepenuhnya terpisah dari data pelatihan dan validasi. Hasil pengujian pada data ini ditampilkan dalam bentuk *Confusion Matrix* dan *mAP Metrics* untuk menilai akurasi dan kemampuan deteksi model. Bagian kedua adalah data video yang diambil langsung dari *CCTV*, yang digunakan untuk mengevaluasi performa deteksi model secara *real-time*, sehingga dapat menilai sejauh mana model bekerja dalam kondisi sebenarnya.

- Evaluasi Uji: *Confusion Matrix* dan *mAP Metrics*



Gambar 3.10 Confusion Matrix

Dapat dilihat pada Gambar 3.10 menunjukkan *Confusion Matrix* ternormalisasi yang digunakan untuk mengevaluasi kinerja model *YOLOv8* Medium pada data uji. Matriks ini memperlihatkan perbandingan antara label sebenarnya (*True*) dan hasil prediksi model (*Predicted*) untuk empat kelas, yaitu fire, person, truck, dan background. Berikut meruapan Hasil analisis dari data uji berdasarkan *Confusion Matrix*:

1. Kelas api memiliki tingkat prediksi benar sebesar 0.76, yang berarti model dapat mendeteksi objek api dengan cukup baik, meskipun masih terdapat kesalahan deteksi yang diklasifikasikan sebagai background (0.24) dan sebagian kecil sebagai kelas lain.
2. Kelas orang memiliki akurasi prediksi sebesar 0.79, namun terdapat kesalahan klasifikasi yang cukup besar ke kategori background (0.21), yang mengindikasikan model masih kesulitan membedakan objek manusia dari latar belakang.

3. Kelas truk menunjukkan hasil prediksi yang baik dengan nilai 0.86, menandakan model cukup akurat dalam mendeteksi truk, meskipun terdapat kesalahan kecil ke kategori background (0.13).
4. Untuk kelas background, model menunjukkan tingkat prediksi yang sangat tinggi sebesar 0.92, yang menunjukkan kemampuan model untuk mengenali latar belakang dengan baik, meskipun masih ada sedikit kesalahan klasifikasi terhadap kelas objek lainnya.

Secara keseluruhan, Confusion Matrix ini mengindikasikan bahwa model YOLOv8 mampu mengenali objek dengan cukup baik, namun masih terjadi mis-klasifikasi terutama antara objek yang berukuran kecil atau berada pada latar belakang yang kompleks.

Tabel 3.4 menampilkan mAP Metrics, yang memuat metrik precision (P), recall (R), mAP50, dan mAP50-95:

*Table 3.4 mAP Metrics*

<b>Kelas</b>	<b>Precision (p)</b>	<b>Recall (R)</b>	<b>mAP50</b>	<b>mAP50-90</b>
All	0.817	0.718	0.812	0.495
Api	0.812	0.624	0.763	0.389
Orang	0.725	0.715	0.755	0.445
Truk	0.914	0.816	0.918	0.640

Secara keseluruhan, model menunjukkan kinerja yang cukup baik dengan nilai rata-rata precision sebesar 0.817, recall 0.718, mAP50 sebesar 0.812, dan mAP50-95 sebesar 0.495. Hal ini mengindikasikan bahwa model mampu mendeteksi objek dengan tingkat akurasi tinggi pada ambang IoU 50%, meskipun akurasi menurun pada rentang IoU yang lebih ketat (50–95%). Performa per kelas menunjukkan variasi hasil sebagai berikut:

1. Kelas fire memiliki nilai precision 0.812 dan recall 0.624, dengan mAP50 sebesar 0.763 dan mAP50-95 sebesar 0.389. Hal ini

menunjukkan bahwa meskipun model cukup baik mendeteksi api, tingkat deteksi yang terlewat (false negative) masih tinggi.

2. Kelas person memperoleh precision 0.725 dan recall 0.715, dengan mAP50 sebesar 0.755. Meskipun akurasi cukup baik, kompleksitas latar belakang dan ukuran objek yang bervariasi kemungkinan menyebabkan kesalahan deteksi.
3. Kelas truck menunjukkan kinerja terbaik dengan precision 0.914, recall 0.816, dan mAP50 0.918, menandakan model sangat andal dalam mendeteksi truk.

Selain itu, kecepatan pemrosesan model juga terbilang efisien, dengan rata-rata waktu 0.4 ms untuk preprocessing, 21.6 ms untuk inference, dan 1.4 ms untuk postprocessing per gambar, sehingga model berpotensi untuk diimplementasikan dalam sistem deteksi real-time.

- Evaluasi Uji: data real yang diambil dari CCTV langsung

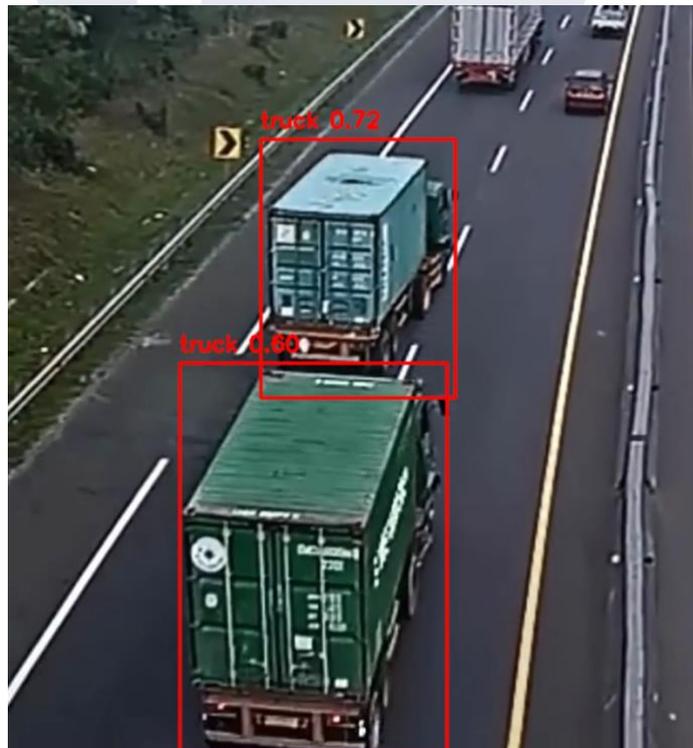
Model yang telah dilatih kemudian diuji menggunakan data uji berupa gambar dan video untuk mengevaluasi performa deteksi secara *real-time*. Untuk pengujian deteksi objek orang dan api, penulis menggunakan dataset berupa video yang diambil secara langsung melalui CCTV dengan mensimulasikan terjadinya kebakaran. Sedangkan untuk pengujian deteksi truk, penulis menggunakan video yang diambil dari sumber terbuka (YouTube).

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



*Gambar 3.11* Sampel testing video – Orang, Api dan truk

Untuk melakukan pengujian, penulis menggunakan script Python yang telah disesuaikan agar dapat membaca model best.pt dan mendeteksi objek api pada input gambar maupun video. Berikut adalah hasil pengujian model deteksi api, orang dan truk seperti yang diperlihatkan pada Gambar 3.4.



*Gambar 3.12* Hasil testing Truck



Gambar 3.13 Hasil testing Api dan Orang

Berdasarkan hasil yang ditampilkan pada Gambar 3.12 dan 3.13, model mampu mendeteksi objek truk dan orang dengan cukup baik. Namun, terdapat kendala di mana objek api terkadang terdeteksi sebagai objek orang. Menurut analisis penulis, hal ini kemungkinan disebabkan oleh proses normalisasi label yang belum dilakukan secara menyeluruh atau merata pada seluruh data, sehingga menyebabkan ambiguitas dalam proses pelatihan model. Oleh karena itu saat ini penulis sedang mencoba melakukan perbaikan terhadap dataset, terutama peninjauan ulang terhadap label untuk memastikan bahwa setiap objek telah diberi label yang tepat sesuai dengan kelasnya. Setelah dataset diperbaiki dan dianggap sudah sesuai, penulis akan melakukan pelatihan ulang (retraining) model menggunakan data yang telah diperbarui.

- Evaluasi Uji: Limitasi Model YOLOv8 Medium

Meskipun model YOLOv8 Medium menunjukkan kinerja yang cukup baik pada pengujian, terdapat beberapa keterbatasan yang perlu diperhatikan, yaitu:

1. Dataset yang digunakan untuk pelatihan masih terbatas, baik dari segi jumlah gambar maupun variasi kondisi lingkungan. Sebagian besar data diambil dari sumber terbuka, sehingga belum sepenuhnya merepresentasikan kondisi nyata di lapangan.
2. Model masih mengalami kesalahan klasifikasi ketika objek berada pada latar belakang yang kompleks atau kondisi pencahayaan yang buruk. Misalnya, objek api terkadang terdeteksi sebagai objek orang.
3. Model YOLOv8 cenderung kurang optimal mendeteksi objek dengan ukuran kecil, terutama ketika objek jauh dari kamera atau tertutup sebagian oleh objek lain.
4. Proses pelabelan yang tidak konsisten pada sebagian data menyebabkan ambiguitas saat pelatihan, yang berpengaruh pada akurasi prediksi. Hal ini terlihat dari kesalahan deteksi api menjadi orang pada pengujian real-time.
5. Pada pengujian real-time, kualitas kamera CCTV dapat memengaruhi hasil deteksi. Kamera dengan resolusi rendah atau pencahayaan yang kurang memadai dapat menurunkan akurasi deteksi.

Dari keterbatasan di atas, diharapkan hasil ini dapat menjadi referensi bagi mahasiswa atau peneliti yang akan melakukan penelitian maupun pengembangan proyek serupa di masa mendatang.

### **3.1.1 Monthly Report**

Selama program magang berlangsung, penulis diwajibkan untuk membuat Monthly Report sebagai bentuk pelaporan rutin atas pekerjaan yang telah dilakukan dalam satu bulan. Laporan ini berisi pembaruan aktivitas, progress project, serta target bulanan. Laporan disusun dalam format presentasi PowerPoint dan terlebih dahulu diperiksa oleh Manajer IT Infrastructure beberapa hari sebelum jadwal presentasi bulanan. Setelah melalui proses pengecekan, seluruh laporan dari tim IT Infrastructure akan dikompilasi menjadi satu file presentasi utama.

Selanjutnya, presentasi gabungan tersebut akan dipaparkan dalam rapat koordinasi bulanan yang dihadiri oleh seluruh anggota divisi IT, termasuk direksi IT dan divisi-divisi terkait lainnya. Berikut merupakan contoh presentasi penulis terkait proyek yang dikerjakan selama periode laporan bulanan:



*Gambar 3.14 Presentasi Monthly Report*

### **3.2 Kendala yang Ditemukan**

Selama menjalani kerja magang di PT Porto Indonesia Sejahtera, penulis menghadapi beberapa kendala yang mempengaruhi kelancaran pelaksanaan tugas. Berikut ini adalah beberapa kendala utama yang penulis temukan selama proses magang berlangsung:

- Penulis mengalami kesulitan dalam memahami tugas dan proses kerja di divisi IT Infrastruktur karena keterbatasan pengalaman.
- Proses pelabelan data untuk pelatihan model menjadi tidak efektif karena jumlah data yang sangat banyak, sehingga tidak memungkinkan dilakukan secara manual.
- Waktu pelatihan model machine learning sangat lama akibat penggunaan CPU lokal yang memiliki keterbatasan dalam kapasitas komputasi.

### **3.3 Solusi atas Kendala yang Ditemukan**

Untuk mengatasi kendala-kendala tersebut, penulis mengambil beberapa langkah strategis dan praktis yang bertujuan untuk meningkatkan efektivitas dan efisiensi dalam menjalankan tugas selama magang. Berikut adalah solusi yang diterapkan:

- Penulis melakukan eksplorasi secara mandiri dan aktif berdiskusi dengan tim IT Infrastruktur untuk memperdalam pemahaman terkait tugas dan proses kerja yang dijalankan.
- Penulis memanfaatkan dataset yang sudah dilabeli dan tersedia di platform Roboflow guna mengurangi beban pelabelan data secara manual dan mempercepat proses persiapan data.
- Untuk mengatasi keterbatasan waktu pelatihan, penulis memanfaatkan layanan Google Colab yang menyediakan fasilitas GPU, sehingga proses training dapat berjalan lebih cepat dan efisien.

