

Penerapan Algoritma *You Only Look Once* v8 untuk Mendeteksi Sandi Bendera Semaphore

Dito Chandrasekar, Diky Wardhani

Teknologi Informasi, Universitas Siber Indonesia

dito.yogaswara@cyber-univ.ac.id, diky.wardhani@cyber-univ.ac.id

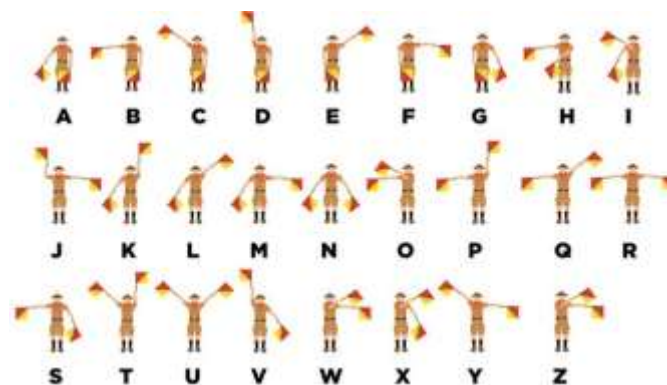
Abstrak

Semaphore merupakan salah satu metode komunikasi visual jarak jauh yang menggunakan isyarat berupa bendera, tongkat, tangan kosong, atau sarung tangan untuk mengirim dan menerima pesan. Sistem ini biasanya memanfaatkan posisi dan arah gerakan kedua tangan untuk merepresentasikan huruf-huruf alfabet. Teknik komunikasi ini telah lama digunakan, terutama dalam dunia maritim dan kegiatan kepramukaan, karena sangat efektif ketika komunikasi suara atau alat komunikasi elektronik tidak dapat digunakan. Penelitian ini bertujuan untuk membantu proses identifikasi gerakan sandi semaphore secara otomatis dan akurat dengan memanfaatkan teknologi deep learning, khususnya menggunakan algoritma YOLO v8 (You Only Look Once version 8) untuk mendeteksi objek visual. Metode ini dipilih karena mampu melakukan deteksi secara real-time dengan tingkat akurasi tinggi. Dataset penelitian terdiri dari 3.458 citra digital yang merepresentasikan huruf semaphore dari A hingga Z, yang kemudian diklasifikasikan ke dalam 26 kelas. Dalam pengujian pertama, dengan pembagian data sebesar 60% untuk pelatihan, 20% untuk validasi, dan 20% untuk pengujian, model menghasilkan akurasi (P) 98,2%, recall (R) 96,4%, dan mean average precision (mAP) 99%. Uji kedua dengan rasio 70%:20%:10% memberikan hasil akurasi 98,5%, recall 99%, dan mAP 99,5%. Sementara itu, uji ketiga dengan rasio 80%:10%:10% menunjukkan hasil yang juga sangat baik, yaitu akurasi 98,3%, recall 98,8%, dan mAP 99,5%. Hasil ini menunjukkan bahwa penerapan YOLO v8 sangat efektif untuk mendeteksi dan mengenali gerakan semaphore secara visual dengan tingkat ketepatan yang sangat tinggi.

Kata kunci: Semaphore, YOLO v8, Deep Learning, Deteksi Objek, Citra Digital.

1. Latar Belakang

Semaphore merupakan cara untuk berkomunikasi jarak jauh dengan menggunakan bendera semaphore dengan 45 cm x 45 cm diikat pada tongkat berukuran 60 cm (Rachmad & Fuad, 2015). Di Indonesia, keterampilan semaphore dianggap sebagai sebuah keterampilan wajib yang harus dikuasai oleh setiap anggota dalam kegiatan pramuka. Bendera semaphore yang digunakan di Indonesia umumnya berwarna merah dan kuning (Madore, 2015). Gambar menunjukkan bendera semaphore. Menurut statistik dari (Moscow: Federal Service for the Supervision of Transport, Department of State Marine and River Supervision, 2017), sekitar setengah dari insiden di laut terjadi karena alasan navigasi, setengahnya lagi diantaranya adalah kurangnya informasi interaksi antar kapal yang letaknya berdekatan.



Gambar 1. Sandi Bendera Semaphore

Komunikasi dengan Semaphore sangat dibutuhkan dalam waktu darurat. Namun, karena pengurangan waktu pelatihan untuk petugas sinyal junior, pelatihan spesialis yang relevan telah dibatalkan. Sebagai penggantinya, digunakan sambungan lampu sorot dalam kode Morse. Tapi ini tidak cukup. Misalnya, komunikasi semaphore diperlukan jika ada matahari yang sangat cerah (Serebryany & Zhdanov, 2018). Penelitian tentang deteksi sandi bendera semaphore dilakukan untuk meningkatkan keselamatan navigasi dan meningkatkan keandalan transmisi informasi. Adapun cara menerjemahkan berbagai macam abjad dalam semaphore dengan pengolahan citra bisa menggunakan metode pendeteksi objek. Pendeteksi objek merupakan salah satu bidang pada computer vision.

Beberapa penelitian telah dilakukan selama dekade terakhir mengenai otomatisasi pendeteksian objek pengenalan sandi bendera semaphore. Pada penelitian sebelumnya menggunakan sensor kinect dan menghasilkan rekonisasi gambar semaphore. Selain sensor kinect, pengenalan sandi bendera semaphore juga dapat dilakukan dengan algoritma seperti *You Only Look Once* (YOLO). YOLO (You Only Look Once) framework telah meraih popularitas karena kesesuaiannya yang sangat baik dalam kecepatan dan akurasi. Fitur ini memungkinkan pendeteksian objek dalam gambar secara efisien dan handal (Binuri et al., 2023).

Setelah mengetahui hal tersebut, pendeteksian sandi semaphore pada penelitian ini akan dilakukan dengan algoritma *You Only Look Once* (YOLO) Versi 8. Arsitektur YOLOv8 merupakan versi pengembangan dari YOLOv7 yang mendatangkan beberapa penyempurnaan dan keunggulan baru. Keunggulan utama YOLOv8 adalah dalam kenaikan performa dan akurasi pendeteksian objek.

2. Metode Penelitian

Penelitian ini merupakan penelitian kuantitatif yang mementingkan kedalaman data dan diperlukannya merekam data sebesar mungkin dari populasi yang luas. Fokus penelitian kuantitatif diidentifikasi sebagai proses kerja yang berlangsung secara ringkas, terbatas, dan memilah-milah permasalahan menjadi bagian yang dapat diukur atau dinyatakan dalam angka-angka. (Purnia & Alawiyah, 2020). Penelitian ini menggunakan eksperimen yang merupakan jenis penelitian kuantitatif yang sangat efektif untuk mengukur sebab akibat dan bisa disebut sebagai eksperimen lingkungan. Penelitian ini bertujuan untuk mengetahui dan mengukur tingkat akurasi yang dihasilkan dari pendeteksi objek.

Pengumpulan data pada penelitian ini berupa citra digital yang diambil melalui smartphone. Jumlah citra yang terkumpul sebanyak 3.458 citra dengan format jpeg, yang terdiri dari 26 kelas sandi semaphore. Adapun data yang telah terkumpul akan dijadikan sebuah dataset dengan 26 kelas yaitu huruf A-Z. Dataset yang ada diperoleh secara berkala dalam dua bulan. Teknik analisis data pada penelitian ini adalah mengimplementasikan CRISP-DM. Proses metode CRISP-DM yang diterapkan pada penelitian ini, yakni *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, serta *Evaluation*.

3. Hasil dan Diskusi

Business Understanding

Semaphore seringkali digunakan saat keadaan darurat jika komunikasi elektronik susah untuk dilakukan di laut maupun pegunungan. Namun, terkadang masih terjadi kesalahan untuk membedakan berbagai abjad sandi bendera semaphore. Hal ini membuat implementasi deteksi objek sandi bendera semaphore dapat berguna dalam keadaan darurat maupun untuk sarana pembelajaran. Pada penelitian ini yaitu deteksi objek sandi bendera semaphore dilakukan dengan mengimplementasikan algoritma YOLOv8, algoritma YOLO yang cukup baru dan lebih akurat dari versi sebelumnya. Implementasi YOLOv8 digunakan untuk mengelola dataset deteksi sandi bendera semaphore agar memperoleh hasil tingkat akurasi tinggi dari pendeteksian 26 abjad semaphore. Dengan memanfaatkan teknologi ini dapat membantu dalam pengenalan sandi bendera semaphore dengan lebih mudah dan meminimalisir kesalahan dalam pengenalan abjad bendera semaphore tersebut.

Data Understanding

Dalam penelitian ini dilakukan pengumpulan data terlebih dahulu. Objek diambil dari dua subjek berupa citra digital dari 26 jenis abjad semaphore pada dua orang yang diambil melalui dua yang tidak relevan atau berpotensi mengganggu proses modelling akan dibersihkan. yang terkumpul dan terpilih berjumlah 1456 data. Data yang sudah terkumpul akan di upload kedalam platform Roboflow untuk dilakukan kegiatan anotasi data dengan membagi data yang ada kedalam 26 kelas berbeda abjad sandi bendera semaphore dari huruf A sampai Z. Dengan adanya tahapan ini, diharapkan model deteksi dapat memberikan hasil yang terbaik dalam keakuratan dan dapat diandalkan untuk membantu pendeteksian objek sandi bendera semaphore.

Data Preparation



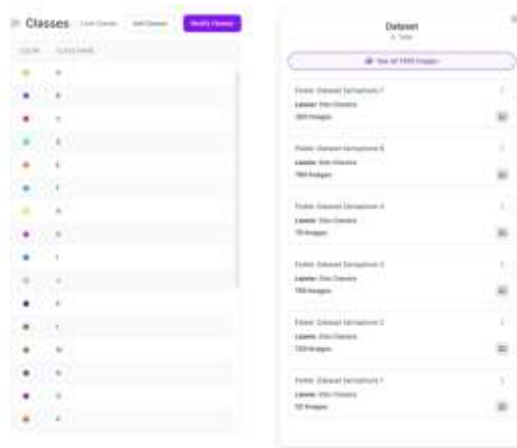
Gambar 2. Dataset pada File Explorer

Dalam tahap ini, proses pemilihan dan pengolahan data terjadi yang nantinya akan diperlukan dalam tahap modelling sehingga tahap modelling yang dilakukan dapat memberi hasil yang akurat sesuai dengan target yang diinginkan. Beberapa proses yang dilakukan pada tahap data preparation sebelum melanjutkan ke tahap selanjutnya, yakni Resizing image

Tabel 1. *Resizing Image*

Sebelum	Sesudah
	
	

Resizing image dilakukan secara manual pada fitur edit yang tersedia pada personal computer dengan melakukan cropping pada beberapa citra untuk memiliki ukuran 3:4. Tujuan dilakukannya *resizing* adalah untuk memperbanyak citra dan menaugmentasi citra yang ada. Tahap selanjutnya adalah *labeling*. Dalam tahap *labeling*, penelitian ini memanfaatkan *platform Roboflow* untuk melakukan kegiatan anotasi data dengan melabeli huruf A-Z pada setiap citra. Sebelum melakukan kegiatan anotasi data, perlu dilakukannya pembuatan proyek baru dalam *Roboflow*, memberi nama proyek, dan memilih tipe deteksi yang akan dilakukan selama penelitian.







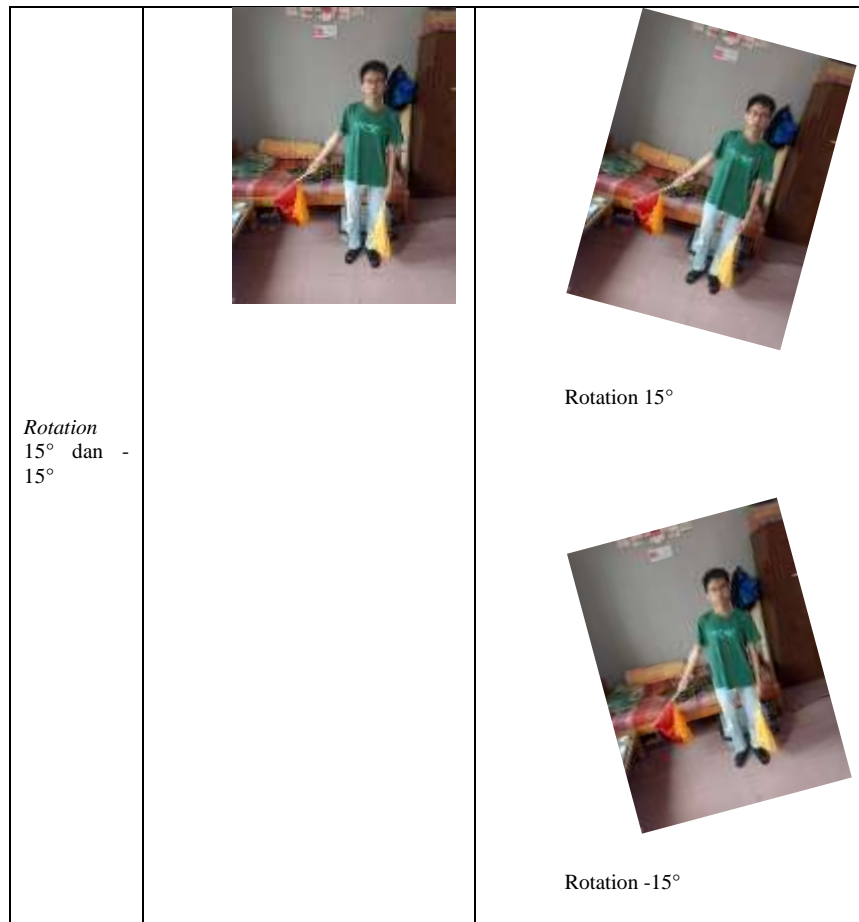
Gambar 3. Pembuatan Kelas pada Roboflow dan Dataset pada Roboflow

Setelah project berhasil dibuat, tahap selanjutnya adalah membuat kelas yang berbeda untuk pendeteksian. Kelas yang akan dibuat dalam penelitian ini adalah 26 kelas berbeda yang merupakan huruf A sampai Z sandi bendera *semaphore*. Tahap selanjutnya adalah mengupload citra yang telah ada ke dalam *Roboflow* dan simpan citra yang ada untuk dapat melakukan ke tahap selanjutnya yaitu kegiatan *labeling* atau anotasi data. Data dianotasi dengan menggunakan fitur *bounding box* dan memilih label untuk citra sesuai dengan 26 kelas yang sudah dibuat. Tujuan dari anotasi adalah pengenalan objek pada komputer.

Citra yang sudah dilabeli ke dalam 26 kelas, selanjutnya akan dijadikan sebuah *dataset* seperti pada gambar di atas untuk dilakukan tahapan selanjutnya. Tahap selanjutnya adalah *Augmentation*. Augmentasi dalam penelitian ini dilakukan dengan 2 cara, yaitu proses augmentasi manual yang dilakukan pada fitur edit yang ada pada *personal computer*, dan augmentasi otomatis yang akan dilakukan pada platform *Roboflow* setelah *dataset* sudah dilakukan anotasi. Augmentasi seperti dalam tabel dilakukan dengan fitur generate dalam platform *Roboflow*, yaitu sebagai berikut:

Tabel 2. Augmentasi Otomatis Roboflow

Teknik	Sebelum	Sesudah
<i>Cropping</i> dengan memotong sebagian citra ke dalam aspek rasio 16:9		
<i>Stretch</i> gambar ke ukuran 640x640		



Augmentasi yang dipilih pada platform *Roboflow* untuk penelitian ini adalah *cropping* sebagian citra ke dalam aspek rasio 16:9, meregangkan gambar ke dalam ukuran 640x640, penambahan *rotation* 15° secara *clockwise* dan *counter clockwise*. Penambahan augmentasi pada *Roboflow* membuat jumlah dari data yang ada menjadi lebih banyak dan mencapai lebih dari 3.000 data berupa citra digital yang awalnya hanya 1456 data. Dalam tahap data *training* dan data *testing*, *dataset* yang sudah ter-*generate* dibagi 3 rasio data latih. Berikut ini merupakan tabel dari pembagian batch size dan *epoch* pada ke 3 rasio data latih.

Tabel 3. Pembagian *Image Size* dan *Epoch* Data Latih

Rasio Data	<i>Epoch</i>	<i>Image Size</i>
60:20:20	50	800
70:20:10	50	800
80:10:10	50	800

Modeling

Dalam tahap modelling, kegiatan yang dilakukan adalah pembuatan model dengan menggunakan algoritma YOLOv8 untuk menjalankan proses pendeteksian sandi bendera *semaphore* dengan melakukan proses-proses yang akan ada pada *Google Colab*.

```
38 epochs completed in 1.357 hours.
nglizer stopped from /mnt/defect/train/wgts/last.pt, 22.4MB
nglizer stopped from /mnt/defect/train/wgts/last.pt, 22.4MB

Validating /mnt/defect/train/wgts/last.pt...
Validation metrics: 0.184 AP@0.5, 0.202 AP@0.75, 0.211 AP@1.0, 0.211 AP@0.5:0.95, 0.211 AP@0.5:0.95, 0.211 AP@0.5:0.95
Model summary (FLOPs): 184.1 GFLOPs, 1113888 parameters, 9 gradients, 23.1 GB VRAM
Class      Images  Instances  BoxP  S  mAP0.5  mAP0.5-0.95  mAP0.5:0.95  mAP0.5:0.95
all        200      200      0.184  0.202  0.211  0.211  0.211
0          200      17      0.066  1  0.005  0.002  0.002
8          200      10      0.077  1  0.006  0.003  0.003
9          200      11      1  0.000  0.000  0.000  0.000
10         200      9      0.070  1  0.005  0.002  0.002
11         200      8      1  0.017  0.005  0.000  0.000
12         200      7      0.062  1  0.005  0.000  0.000
13         200      13      0.066  1  0.006  0.000  0.000
14         200      14      1  0.000  0.000  0.000  0.000
15         200      14      0.062  1  0.005  0.000  0.000
16         200      13      0.067  1  0.005  0.000  0.000
17         200      14      1  0.000  0.000  0.000  0.000
18         200      13      0.060  1  0.005  0.002  0.002
19         200      9      0.070  1  0.005  0.002  0.002
20         200      11      0.00  1  0.000  0.000  0.000
21         200      13      0.060  1  0.005  0.000  0.000
22         200      14      0.062  1  0.005  0.000  0.000
23         200      12      1  0.012  0.005  0.000  0.000
24         200      9      0.00  1  0.000  0.000  0.000
25         200      13      0.060  0.000  0.000  0.000  0.000
26         200      14      0.062  1  0.005  0.002  0.002
27         200      6      0.065  1  0.005  0.000  0.000
28         200      11      0.062  1  0.005  0.000  0.000
29         200      12      1  0.012  0.005  0.000  0.000

Speed: 4.0ms preprocess, 7.2ms inference, 0.0ms loss, 1.0ms postprocess per image
Results saved to /mnt/defect/train
Learn more at https://docs.ultralytics.com/models/train
```

Gambar 4. Proses Training YOLOv8

Gambar di atas menunjukkan proses *training* untuk 50 *epochs* pada YOLOv8 yang telah selesai. Gambar di bawah ini menampilkan citra yang sudah terdeteksi melalui tahap *testing* yang ditandai dengan label pada setiap kelasnya. Jumlah foto yang terdapat pada hasil deteksi terdapat foto yang tersimpan pada *file explorer*.



Gambar 5. Hasil Citra

Gambar di bawah ini menunjukkan tahapan validasi, tahapan ini memungkinkan untuk menilai kualitas model yang telah dilatih. Dalam YOLOv8, validasi mengacu pada mengevaluasi kinerja model deteksi objek yang telah dilatih. Berikut merupakan analisis dari gambar

```
Validating /mnt/defect/train/wgts/last.pt...
Validation metrics: 0.184 AP@0.5, 0.202 AP@0.75, 0.211 AP@1.0, 0.211 AP@0.5:0.95, 0.211 AP@0.5:0.95, 0.211 AP@0.5:0.95
Model summary (FLOPs): 184.1 GFLOPs, 1113888 parameters, 9 gradients, 23.1 GB VRAM
Class      Images  Instances  BoxP  S  mAP0.5  mAP0.5-0.95  mAP0.5:0.95  mAP0.5:0.95
all        200      200      0.184  0.202  0.211  0.211  0.211
0          200      17      0.066  1  0.005  0.002  0.002
8          200      10      0.077  1  0.006  0.003  0.003
9          200      11      1  0.000  0.000  0.000  0.000
10         200      9      0.070  1  0.005  0.002  0.002
11         200      8      1  0.017  0.005  0.000  0.000
12         200      7      0.062  1  0.005  0.000  0.000
13         200      13      0.066  1  0.006  0.000  0.000
14         200      14      1  0.000  0.000  0.000  0.000
15         200      14      0.062  1  0.005  0.000  0.000
16         200      13      0.067  1  0.005  0.000  0.000
17         200      14      1  0.000  0.000  0.000  0.000
18         200      13      0.060  1  0.005  0.002  0.002
19         200      9      0.070  1  0.005  0.002  0.002
20         200      11      0.00  1  0.000  0.000  0.000
21         200      13      0.060  1  0.005  0.000  0.000
22         200      14      0.062  1  0.005  0.000  0.000
23         200      12      1  0.012  0.005  0.000  0.000
24         200      9      0.00  1  0.000  0.000  0.000
25         200      13      0.060  0.000  0.000  0.000  0.000
26         200      14      0.062  1  0.005  0.002  0.002
27         200      6      0.065  1  0.005  0.000  0.000
28         200      11      0.062  1  0.005  0.000  0.000
29         200      12      1  0.012  0.005  0.000  0.000

Speed: 3.0ms preprocess, 11.0ms inference, 0.0ms loss, 1.0ms postprocess per image
Results saved to /mnt/defect/train
Learn more at https://docs.ultralytics.com/models/train
```

Gambar 6. Validasi YOLOv8

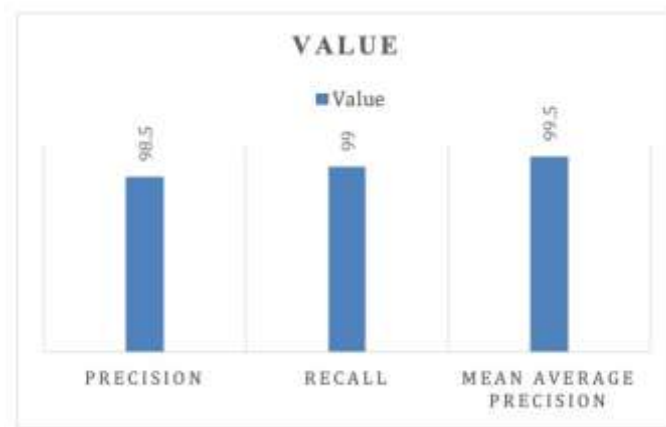
Evaluation

Dalam penelitian ini 3 kali proses *training* sudah dilakukan dari 3 rasio data yang sudah ditentukan dan hasil akurasi tertinggi adalah 70%:20%:10%. Pada rasio data ini, dilakukan pemilihan *epoch* sebanyak 50 dengan *image size* 800.

Tabel 4. Evaluasi Hasil Training

Evaluasi Hasil <i>Training</i>					
Tahap <i>Experiment</i>	Rasio Data	<i>Epoch</i> dan <i>Image Size</i>	Nilai Akurasi "P"	Nilai <i>Recall</i> "R"	Nilai mAP
Percobaan Pertama	60%:20%:20%	50 <i>Epoch</i> dan 800 <i>imgsz</i>	98.2	96.4	99.0
Percobaan Kedua	70%:20%:10%	50 <i>Epoch</i> dan 800 <i>imgsz</i>	98.5	99.0	99.5
Percobaan Ketiga	80%:10%:10%	50 <i>Epoch</i> dan 800 <i>imgsz</i>	98.3	98.8	99.5

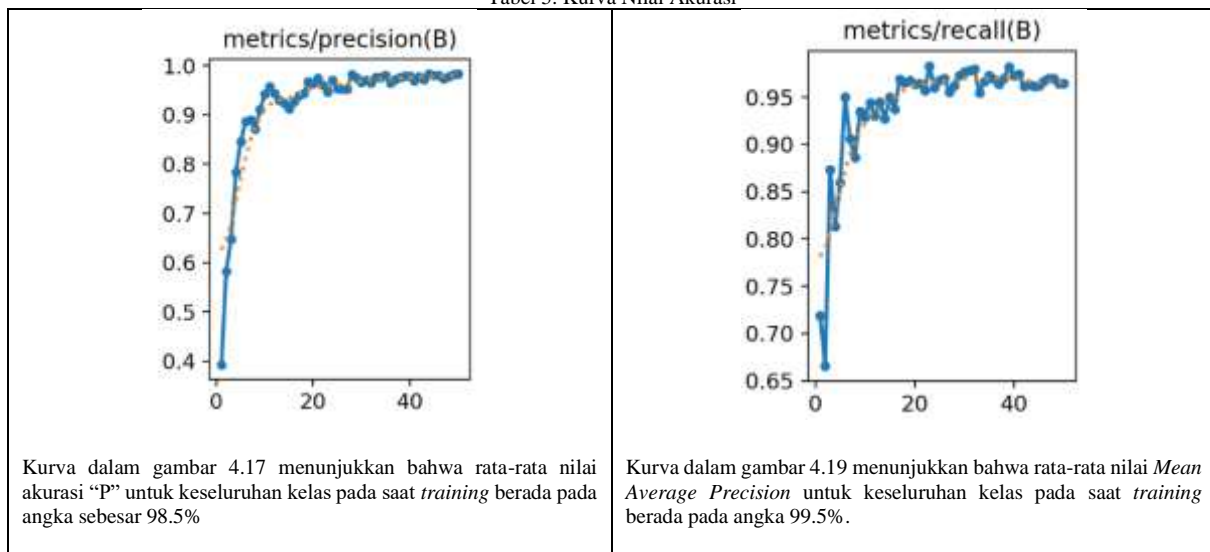
Hasil dari pendeteksian objek sudah terlihat dalam bentuk kurva seperti pada gambar-gambar berikut yang merupakan kurva dari uji pada rasio data.



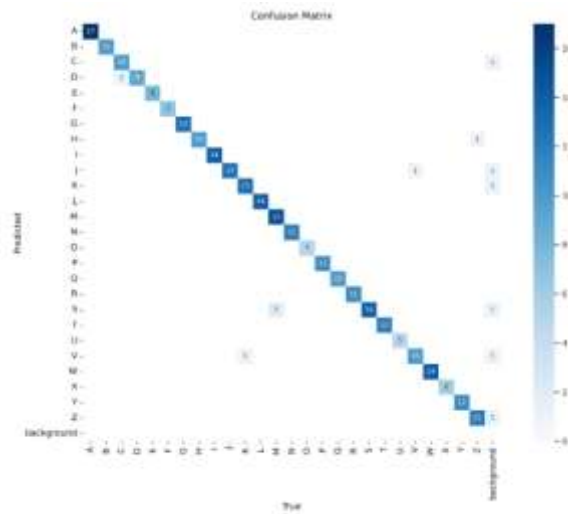
Gambar 7. Chart Hasil Training Rasio 70%:20%:10%

Gambar 7. Diatas merupakan hasil training rasio 70%:20%:10% diketahui bahwa Precision memiliki nilai 98.5, Recall memiliki nilai 99 dan Mean Average Precision memiliki nilai 99.5

Tabel 5. Kurva Nilai Akurasi



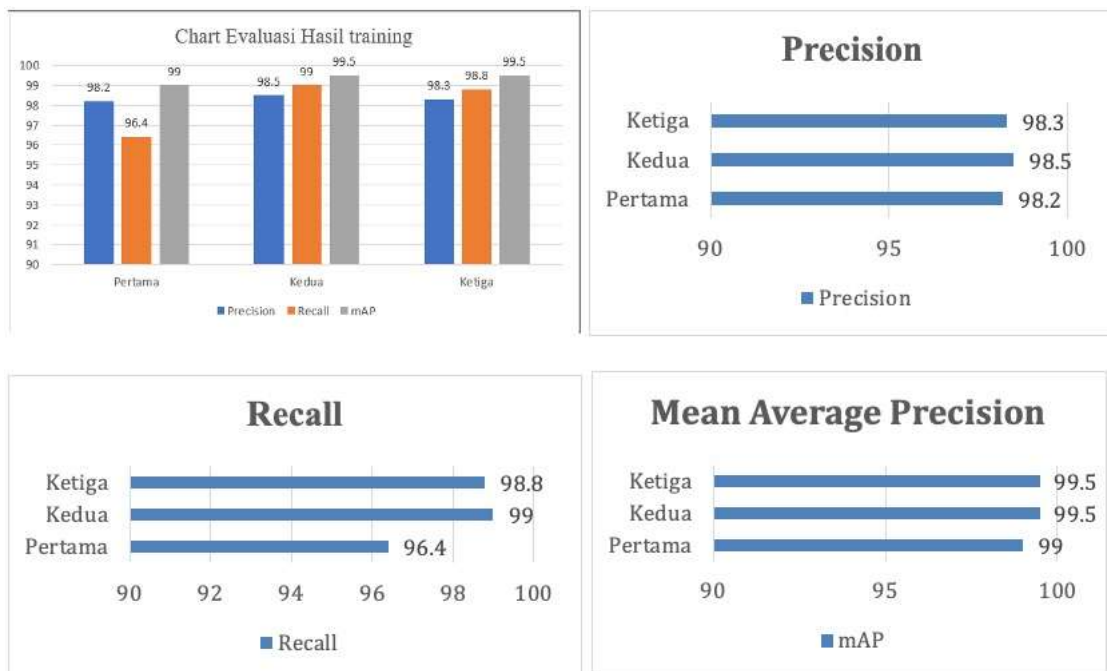
Pengujian akurasi model dilakukan dengan cara mendeteksi pada *dataset testing*. Setiap kali model berhasil mendeteksi objek sandi bendera *semaphore* dengan tingkat akurasi diatas 90%. Maka dari itu, model dianggap berhasil dalam mendeteksi objek dengan akurat.



Gambar 8. Confusion Matrix Sandi Semaphore

Gambar 8 merupakan tingkat akurasi hasil evaluasi deteksi objek sandi bendera *semaphore*. Semakin gelap warna hasil evaluasi, maka semakin tinggi tingkat akurasinya. Hasil pendeteksian menunjukkan bahwa telah di deteksi dengan tingkat akurasi yang tinggi.

Tabel 6. Hasil Evaluasi



4. Kesimpulan

Kesimpulan yang didapat berdasarkan hasil dari penelitian eksperimen ini adalah sebagai berikut: 1). Pembuatan model YOLO v8 untuk mendeteksi objek berupa sandi bendera *semaphore* telah berhasil dilakukan dengan melakukan pendeteksian sandi bendera *semaphore* pada sebuah *dataset* yang di proses menggunakan algoritma YOLO v8 pada *Google Colab* hingga hasil pendeteksian didapat, 2). YOLO v8 dapat mendeteksi dengan akurat setelah 3 kali percobaan. Percobaan kedua berhasil mendeteksi jenis kematangan jerawat seperti pada tabel. Hasil deteksi objek sandi bendera memiliki tingkat akurasi “P” 98.5% pada rasio data *training*, *valid*, dan *testing* 70%:20%:10% dan dapat disimpulkan bahwa model ini bekerja dengan baik.

Referensi

1. Amazon Web Services. (n.d.) Apa itu komputasi cloud?1. Diakses pada 15 Agustus 2024, dari <https://aws.amazon.com/id/what-is/data-augmentation/>
2. Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform* (pp. 59-64). Berkeley, CA: Apress.
3. Bisong, E., & Bisong, E. (2019). *Python. Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, 71-89.
4. Brad Dwyer, James Gallagher (Mar 16, 2023). *Getting Started with Roboflow. Roboflow Blog: https://blog.roboflow.com/getting-started-with-roboflow/*
5. Dompeipen, T. A., & Sompie, S. R. U. (2020). Penerapan *computer vision* untuk pendeteksian dan penghitungan jumlah manusia. *Jurnal Teknik Informatika*, 15(4), 1-12.
6. Dwiyanto, R., Widodo, D. W., & Kasih, P. (2022, November). Implementasi Metode *You Only Look Once* (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 6, No. 3, pp. 102-104).
7. Eriana, E. S., & Zein, A. (2023). *Artificial Intelligence (AI)*.
8. Hussain, M. (2023). YOLO-v1 to YOLO-v8, *the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines*, 11(7), 677.
9. Jiang, X., Hadid, A., Pang, Y., Granger, E., & Feng, X. (Eds.). (2019). *Deep learning in object detection and recognition*.
10. Karlina, O. E., & Indarti, D. (2020). Pengenalan objek makanan cepat saji pada video dan real time webcam menggunakan metode you look only once (yolo). *Jurnal Ilmiah Informatika Komputer*, 24(3), 199-208.
11. Madore, A..2015. *Semaphore Notes, Canada's Naval History, Canadian War Museum*.
12. Müller, A. C., & Guido, S. (2017). *Introduction to machine learning with Python: a guide for data scientists. O'Reilly Media, Inc.*
13. Polamuri, Dr & Kumbhkar, Makhan & Daniel, Dr. (2022). *Introduction to Deep Learning*.
14. Purnia, D. S., & Alawiyah, T. (2020). Metode Penelitian Strategi Menyusun Tugas Akhir. *Yogyakarta: Graha Ilmu*.
15. Rachmad, A., & Fuad, M. (2015). *GEOMETRY ALGORITHM ON SKELETON IMAGE BASED SEMAPHORE GESTURE RECOGNITION. Journal of Theoretical & Applied Information Technology*, 81(1).
16. Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(2).
17. Semaphore. 2024. Pada KBBI Daring. Diambil 30 May 2024, dari <https://kbbi.kemdikbud.go.id/entri/semaphore>
18. Serebryany N S and Zhdanov B B 2018 *Handbook of the signalman* (Moscow: Voen)
19. Sugiyono, D. (2013). Metode penelitian kuantitatif, kualitatif dan R & D/Sugiyono. *Bandung: Alfabeta*.
20. Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.
21. Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.