

# *Training Custom Model Deteksi Udang menggunakan YOLOv8*

Muhamad Irfan Maulana dan Rina Noviana

Magister Manajemen Sistem Informasi, Universitas Gunadarma  
Jl. Margonda Raya No. 100 Pondok Cina Depok Jawa Barat  
E-mail: muhamadirfan45513@gmail.com, rina\_n@staff.gunadarma.ac.id

## **Abstrak**

Perkembangan teknologi informasi yang semakin pesat membuat perkembangan teknologi di dunia computer vision semakin berkembang. Saat ini, penggunaan teknologi computer vision secara masif digunakan hampir di setiap kehidupan manusia yang tentunya sangat membantu manusia dalam melakukan pekerjaan dan aktivitas sehari-hari. Salah satu objek dari visi komputer adalah deteksi objek. Salah satu komoditas perikanan yang berkembang pesat di Indonesia adalah udang. Teknologi pendeteksian objek dapat membantu para petambak udang dalam mendeteksi udang dan menentukan jumlah udang yang terdeteksi. Penelitian ini akan membuat model pelatihan khusus untuk mendeteksi udang dengan menggunakan YOLOv8. Metode penelitian dalam penelitian ini dibagi menjadi 5 tahapan yaitu pengumpulan dataset, data annotations (labelling), pre-processing, training dan testing model, serta evaluasi. Tujuan penelitian ini adalah untuk melakukan training custom model deteksi udang menggunakan YOLOv8 yang diharapkan nantinya custom model yang dihasilkan dapat diimplementasikan dalam aplikasi pendeteksi udang serta penghitungan jumlah udang yang terdeteksi. Dataset yang digunakan adalah dataset dengan dua kelas, yaitu udang dan bukan\_udang dengan jumlah data citra sebanyak 6630 data. Proses anotasi citra dilakukan dengan menggunakan roboflow. Proses training dataset dilakukan dengan menggunakan Google Colaboratory, YOLOv8, dan hasil yang didapat disimpan di Google Drive. Pelatihan dilakukan dengan total 300 epoch. Nilai rata-rata precision sebesar 92.57%, sedangkan nilai recall sebesar 88.2%. Akurasi model diperoleh nilai 93.2%. Hasil ini menggambarkan bahwa model relatif stabil dalam mengklasifikasikan objek dengan tepat.

**Kata kunci** :Deteksi citra, YOLOv8, Google Colab, Roboflow, Udang.

## **Pendahuluan**

Salah satu komoditas akuakultur [1] yang berkembang sangat pesat di Indonesia saat ini adalah udang. Udang menjadi salah satu komoditas ekspor potensial yang ada di Indonesia [2]. Selain itu, udang juga banyak dibudidayakan oleh para peternak daerah di Indonesia. Berdasarkan data dari Kementerian Kelautan dan Perikanan (KKP) [3], pada tahun 2020, produktivitas budidaya udang di Indonesia berkisar antara 10 – 50 ton/hektar/siklus. Pada tahun 2020 produksi budidaya udang di Indonesia mencapai 911,2 ribu ton, udang juga memberikan kontribusi terhadap total volume ekspor hasil perikanan sebesar 18,95% [4].

Perkembangan teknologi informasi yang sangat pesat menyebabkan semakin banyaknya perkembangan teknologi di dunia computer vision maupun machine learning. Saat ini, penggunaan teknologi computer vision maupun machine learning digunakan secara masif hampir disetiap kehidupan manusia yang tentunya sangat membantu manu-

sia dalam melakukan pekerjaan maupun kegiatan sehari-hari. Salah satu objek dari computer vision adalah object detection. Object detection merupakan salah satu penerapan dari teknologi deep learning yang merupakan turunan dari machine learning yang menggunakan Artificial Neural Network (ANN) [5]–[7]. Penggunaan object detection membantu tugas manusia dalam mendeteksi objek [8]. Teknologi object detection dapat membantu peternak udang dalam mendeteksi udang [9] serta menentukan jumlah udang. Penerapan object detection dapat dilakukan melalui berbagai algoritma deep learning yang ada [10]. Algoritma You Only Look Once (YOLO) merupakan salah satu contoh dari Algoritma Deep Learning yang banyak digunakan dalam object detection [11], [12]. Algoritma YOLO merupakan algoritma object detection yang diklaim memiliki kecepatan dan akurasi yang paling cepat dibandingkan dengan algoritma object detection lainnya [13]. YOLOv8 adalah keluarga terbaru model Deteksi Objek berbasis YOLO dari Ultralytics yang memberikan kinerja canggih. Model

YOLOv8 lebih cepat dan lebih akurat dibandingkan dengan versi-versi sebelumnya sehingga sangat cocok diterapkan sebagai algoritma object detection [14]–[16].

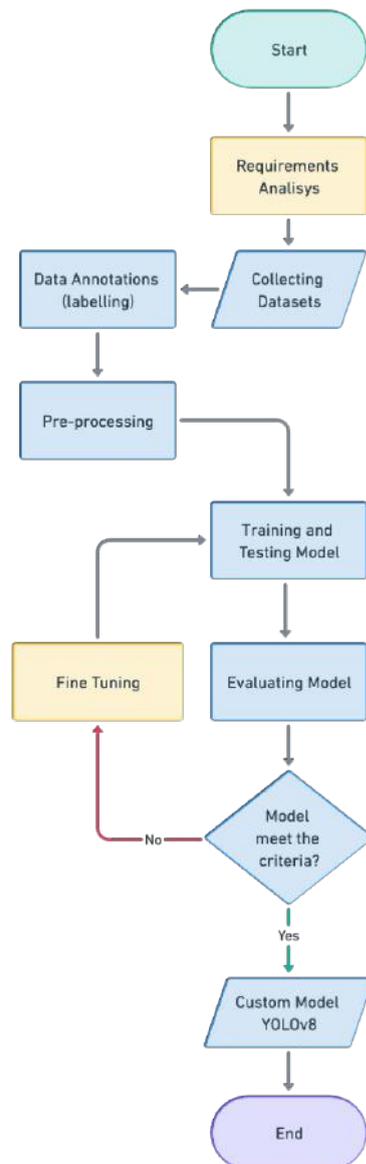
Pada penelitian sebelumnya yang telah dilakukan oleh Siska Armalivia, Zahir Zainuddin, Andani Achmad, dan Muh. Arief Wicaksono pada tahun 2021 yang berjudul “Automatic Counting Shrimp Larvae Based You Only Look Once (YOLO)”, telah merancang dan menerapkan algoritma YOLOv3 untuk mendeteksi larva udang serta menghitung jumlah larva udang yang terdeteksi. Meskipun penelitian tersebut cukup lengkap dan terarah, akan tetapi versi YOLO yang digunakan merupakan versi YOLO yang sudah lama [17]. YOLOv8 [15] adalah keluarga terbaru model Deteksi Objek berbasis YOLO dari Ultralytics yang memberikan kinerja canggih. Model YOLOv8 [15] lebih cepat dan lebih akurat dibandingkan dengan

versi-versi sebelumnya sehingga sangat cocok diterapkan sebagai algoritma object detection [16].

Berdasarkan penjabaran tersebut, maka dilaksanakan penelitian yang berjudul “Custom Model Shrimp Detector Training using YOLOv8” yang diharapkan nantinya custom model yang dihasilkan dapat diimplementasikan dalam aplikasi pendeteksi udang serta penghitung jumlah udang yang terdeteksi sehingga dapat membantu petani udang dalam melakukan budidaya udang.

## Metode Penelitian

Penelitian ini terdiri dari 5 tahapan yaitu pengumpulan dataset, data annotations (labelling), pre-processing, training dan testing model, evaluasi. Alur tahapan penelitian ini terdapat pada Gambar 1.



Made with Whimsical

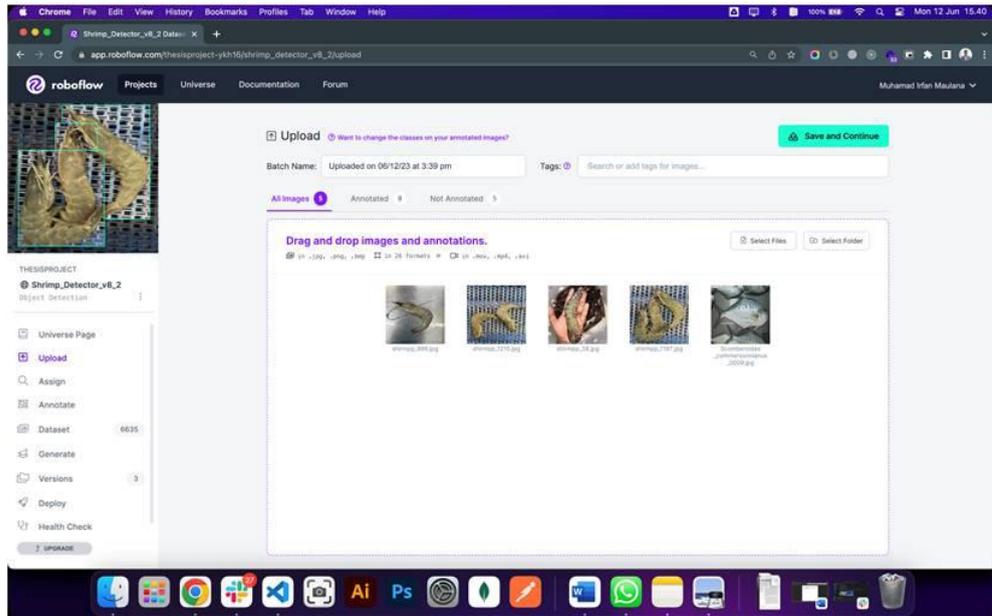
Gambar 1: Peta Pikiran

## Pengumpulan Dataset

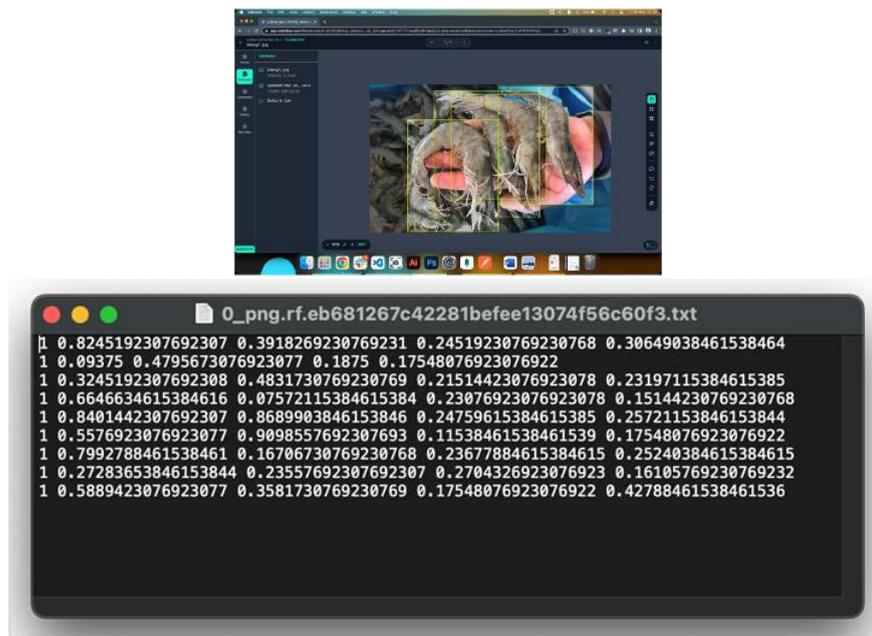
Pada penelitian ini, dilakukan pengumpulan dataset yang berupa data citra udang dan juga data citra bukan udang. Citra udang dan juga citra bukan udang tersebut didapatkan dari beberapa sumber di internet, website Kaggle, dan juga roboflow [18]. Pengumpulan dataset merupakan langkah awal sebelum YOLO diajarkan untuk mengenal benda-benda yang harus dideteksi [19].

Pada Gambar 2, merupakan proses pengumpulan dataset. Data berupa citra yang telah diper-

oleh di-upload ke dalam roboflow [18] agar dapat dilakukan proses anotasi data serta pre-processing nantinya. Pada saat pengumpulan data, diharuskan untuk mengumpulkan data berupa citra yang memiliki resolusi tinggi agar proses pendeteksian dapat dilakukan dengan baik dan dapat menampilkan koordinat bounding-box dengan baik pada masing-masing objek. Jika citra yang digunakan mempunyai resolusi yang rendah, maka hasil deteksi menjadi kurang maksimal. Dataset dikategorikan menjadi dua kelas, yaitu kelas shrimp dan not-shrimp.



Gambar 2: Pengumpulan dataset



Gambar 3: Pengumpulan dataset

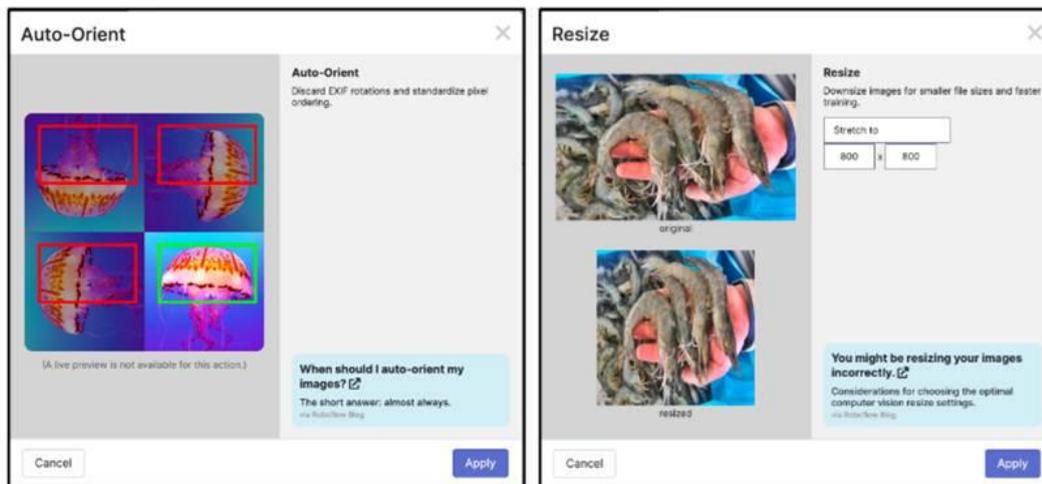
Data Annotation (Labelling) Pada tahap ini, data awal yang berupa data mentah akan diolah dengan melakukan data annotation, yaitu proses labelling. Labelling merupakan proses pemberian anotasi atau label kelas pada objek yang ingin dideteksi pada suatu gambar. Proses labelling dilakukan dengan memberikan bounding box dan kelas pada seluruh objek anggota masing-masing kelas data [20]. Dalam penelitian ini, proses data annotation atau labelling dilakukan menggunakan tools website roboflow [18].

Pada Gambar 3, bounding box yang dihasilkan dalam YOLO akan memiliki data berupa `<class>`

`<x center>` `<y center>` `<width>` `<height>` yang memiliki nilai dari 0 hingga 1 dengan class yang memiliki indeks integer [21]. Satu citra (1 data) dapat memiliki lebih dari 1 bounding box dan dipisahkan oleh garis baru.

## Pre-processing

Tahap pre-processing ini dilakukan menggunakan tools website roboflow [18] ketika akan membuat dataset. Pre-processing yang dilakukan pada penelitian ini adalah penyesuaian orientasi gambar dan juga resolusi input.



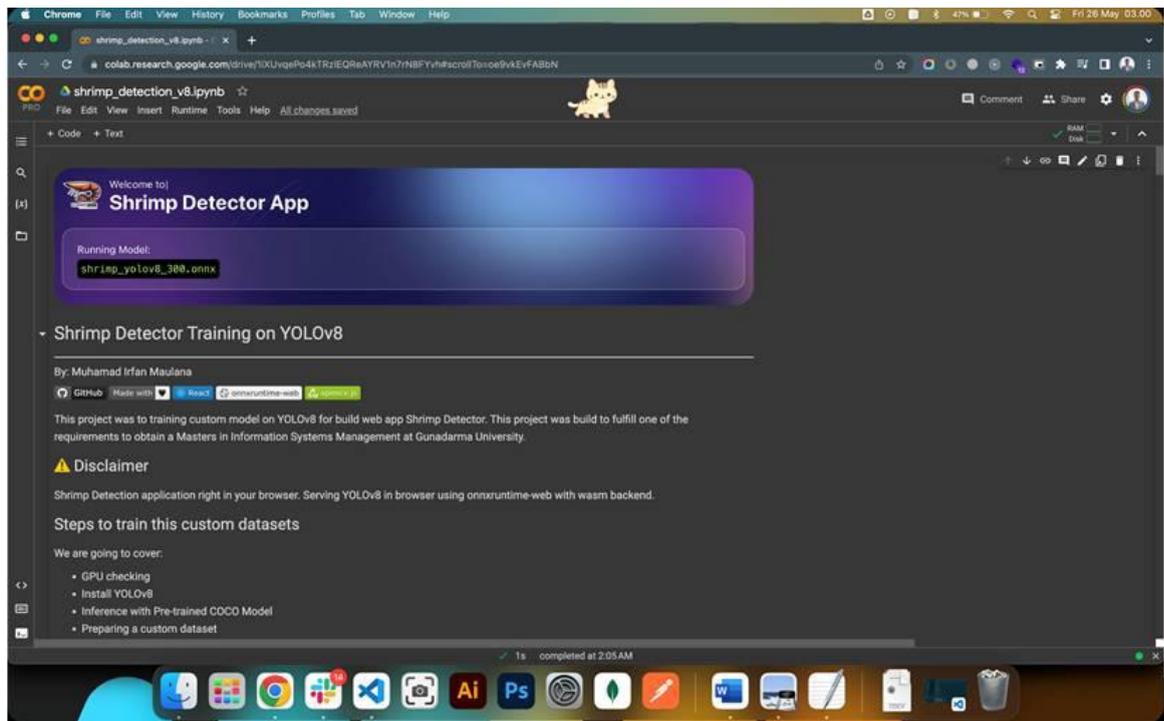
Gambar 4: Pre-processing

Penyesuaian orientasi citra perlu disesuaikan karena Ketika sebuah citra ditangkap, maka citra tersebut akan berisi metadata yang menentukan orientasi yang harus ditampilkan bersifat relative terhadap bagaimana piksel disusun pada memori. Hal tersebut akan disimpan dalam bidang orientasi EXIF yang akan mempercepat proses encoding citra Ketika pengambilan citra sehingga kamera dapat secara efisien mengambil sampel data dari sensornya tanpa artefak yang tidak diinginkan. Hal tersebut berarti bahwa sebagian besar kamera menyimpan piksel gambar yang persis sama, baik kamera diorientasikan dalam mode lanskap atau potret, kamera hanya membalik sedikit untuk memberi sinyal kepada pengguna apakah akan menampilkan piksel apa adanya atau memutarinya 90 atau 180 derajat saat menampilkan gambar. Sayangnya, hal ini dapat menyebabkan masalah jika aplikasi yang menampilkan gambar tidak mengetahui metadata dan menampilkan gambar secara naif tanpa memperhatikan orientasi EXIF-nya. Oleh sebab itu, dalam penelitian ini, salah satu tahap pre-processing yang digunakan adalah menyesuaikan auto-orient. Mengubah resolusi gambar adalah langkah pre-processing yang penting dalam computer vision. Pada prinsipnya, model machine learning dan deep learning memi-

liki proses trainig yang lebih cepat pada gambar yang lebih kecil. Gambar input yang dua kali lebih besar mengharuskan jaringan deep learning untuk belajar dari piksel empat kali lebih banyak. Selain itu, banyak arsitektur model deep learning mengharuskan gambar memiliki ukuran yang sama dan gambar mentah yang dikumpulkan pada umumnya memiliki resolusi yang berbeda. Oleh karena itu, pada tahap pre-processing diperlukan penyesuaian resolusi gambar.

## Training dan Testing

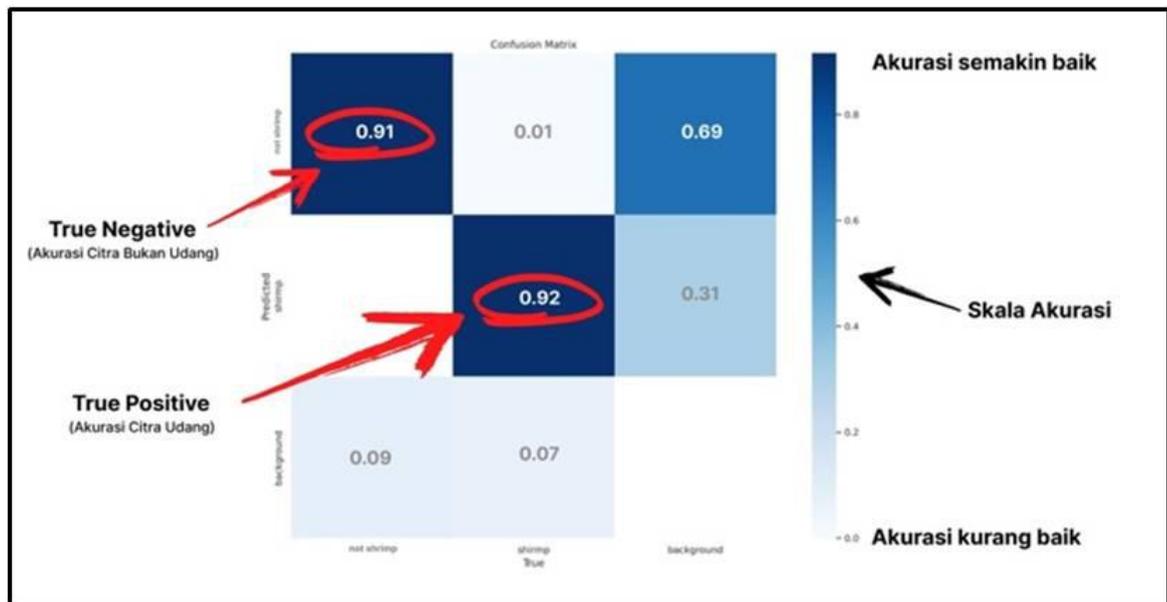
Model Tahapan training dan testing model pada penelitian ini dilakukan menggunakan tools Google Colab Pro. Google Colab adalah sebuah IDE untuk pemrograman Python dimana pemrosesan akan dilakukan oleh server Google yang memiliki perangkat keras dengan performa yang tinggi [22]. Tahapan training custom model yang dilakukan pada google colab memiliki beberapa Langkah yaitu: proses mounting Google Drive [23], GPU checking, instalasi YOLOv8 [15], menyiapkan custom dataset, dan memulai proses custom training. Proses training yang dilakukan menggunakan Google Colab versi Pro dengan pengaturan GPU tipe T4 dan High-RAM agar memperlancar dan mempercepat proses training.



Gambar 5: Google Colab

Pada gambar 5, merupakan tampilan lembar kerja google colab yang menjadi tempat untuk melakukan training custom model YOLOv8. Training dilakukan dengan memanfaatkan pre-trained model YOLOv8. Pada penelitian ini menggunakan pre-trained model yolo8n.pt. Proses training dimulai dengan menggunakan command yolo dari ultralytics. Proses training dilakukan dengan memanggil pre-trained model yaitu yolo8n.pt dan juga data

yang merupakan datasets yang sebelumnya sudah di import ke dalam google colab [22]. Selanjutnya, diperlukan juga penyesuaian epoch. Pada penelitian ini melakukan training dengan epoch sebanyak 300 epoch. Epoch tersebut merupakan proses iterasi dalam proses training custom model yang dijalankan. Selain itu, juga dilakukan penyesuaian resolusi dari input image menjadi ukuran 800x800 pixel.



Gambar 6: Grafik Confusion Matrix yang dihasilkan dari proses training

Custom datasets yang telah dibuat sebelumnya akan ditraining menggunakan pre-trained model dari YOLOv8 agar dapat dikenali masing-masing kelas dari dataset yang digunakan dalam prses pembuatan custom model dan menghasilkan output berupa banyaknya iterasi (300 iterasi), sebuah custom model dengan format .pt, dan sebuah grafik confusion matrix, serta grafik result dari proses training. Custom model dengan format .pt yang dihasilkan nantinya akan digunakan untuk implementasi deteksi objek berdasarkan kelas-kelas yang sudah ditentukan. Hasil akurasi citra udang (true positive) sebesar 0,92 dan akurasi citra bukan udang (true negative) sebesar 0,91. Rata-rata nilai precision yang didapatkan dari custom model tersebut yaitu sebesar 92,57%, sedangkan nilai recall diperoleh rata-rata sebesar 88,2%. Akurasi dari model diperoleh nilai sebesar 92%. Akurasi merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data. Precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Performa dari custom model yang dihasilkan divisualisasikan menggunakan grafik confusion matrix pada Gambar 6.

### Evaluasi

Pada tahap ini dilakukan evaluasi atas model yang telah dibuat dan melalui proses training serta test-

ing. Pada saat proses training, terdapat proses pengenalan masing-masing kelas yang ditentukan oleh nilai presisi dari masing-masing kelas tersebut. Nilai presisi tersebut ditentukan menggunakan confusion matrix [24] yang merupakan daftar data kelas yang di setiap kelas, data aktual diklasifikasikan sehingga dapat diamati kategori sampel mana yang mudah bingung dalam jaringan yang dimodifikasi. Dalam confusion matrix, baris mewakili kategori sebenarnya dari gambar uji. Kolom menunjukkan kelas citra uji dibagi dengan jaringan dites yang sebenarnya. Setelah mendapatkan hasil deteksi, kemudian akan dihitung nilai akurasi untuk setiap hasil deteksi yang dihasilkan menggunakan rumus (1).

$$\text{accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

Keterangan:

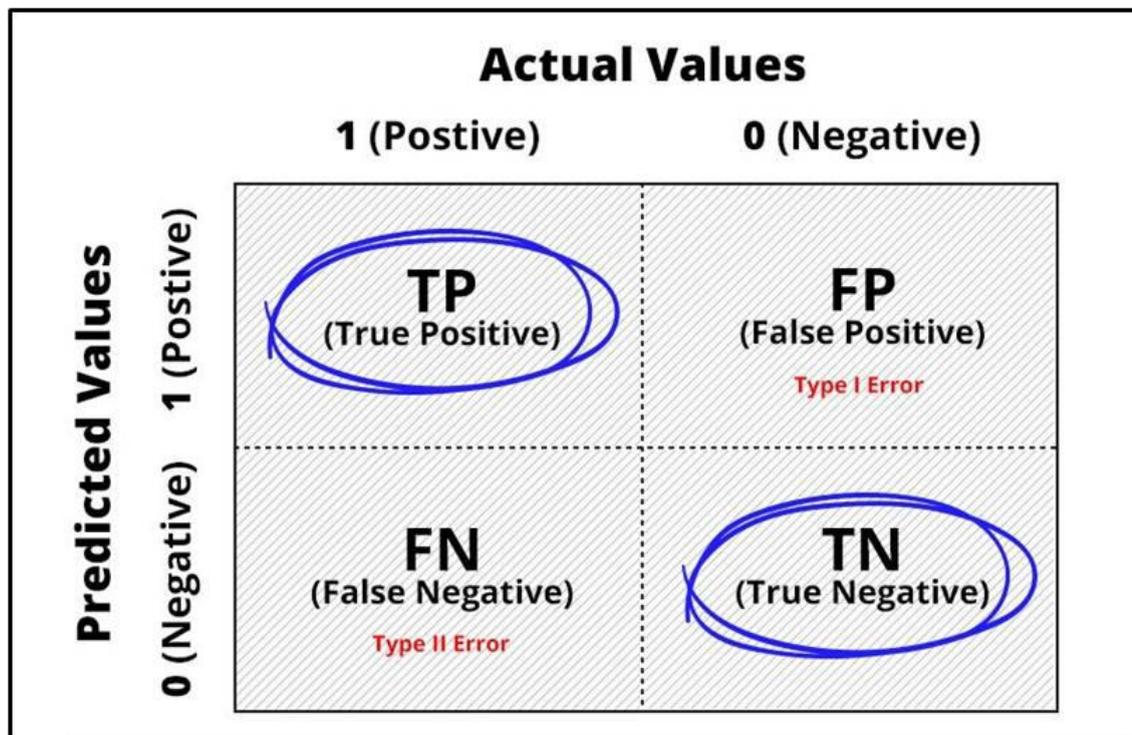
TP : True Positif

TN : True Negatif

FP : False Positif

FN : False Negatif

Accuracy [25] menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Maka, accuracy merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dengan kata lain, accuracy merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya). Nilai accuracy dapat diperoleh dengan persamaan (1).



Gambar 7: Confusion Matrix

Pada Gambar 7, True Positive merupakan jumlah objek yang terdeteksi, sedangkan True Negative merupakan jumlah objek yang tidak terdeteksi. Untuk mengetahui jumlah aktual objek yang ada dapat diketahui melalui dari penjumlahan total kolom actual positive yaitu True Positive ditambahkan dengan False Negative. Sedangkan jumlah actual objek yang tidak terdeteksi dapat diketahui melalui penjumlahan total kolom actual negative, yaitu False Positive ditambahkan dengan True Negative.

Pada tahap ini juga akan ditentukan apakah model yang terbentuk sudah sesuai dengan kriteria yang telah ditentukan dan dibutuhkan atau belum. Jika belum sesuai, maka harus dilakukan proses fine tuning, akan tetapi jika sudah sesuai dengan kriteria, maka proses training selesai dan didapatkan custom yolov8 model dengan format .pt.

## Hasil dan Pembahasan

Dalam penelitian ini, dataset yang digunakan dibuat menggunakan roboflow dengan kumpulan citra udang dan bukan udang yang berasal dari beberapa sumber di internet, website Kaggle, dan juga roboflow.

### Hasil Pengumpulan Dataset

Berdasarkan proses pengumpulan datasets pada alur proses penelitian, maka didapatkan data sebanyak 6630 data dengan dua kelas yaitu shrimp dan not shrimp. Tabel 1 merupakan contoh hasil pengumpulan data.

Tabel 1: Contoh Hasil Pengumpulan Data

No	Image	Resolution (px)	Format
1		416x416	.jpg
2		416x416	.jpg
3		1024x593	.jpg

Setelah dilakukan pengumpulan dataset, maka dilakukan tahapan anotasi citra atau sering disebut proses labelling. Proses anotasi citra tersebut dilakukan menggunakan tools roboflow dengan dua

kelas yaitu shrimp dan not shrimp. Pada Tabel 2 terdapat hasil proses anotasi citra.

Tabel 2: Hasil Proses Anotasi Citra

No	Image	Class
1		shrimp
2		shrimp
3		Shrimp and Not-shrimp

Setelah dilakukan proses anotasi citra, maka proses selanjutnya adalah tahap pre-processing. Tahapan pre-processing ini juga dilakukan menggunakan tools roboflow. Tahapan pre-processing yang dilakukan adalah auto-orient, resizing-image menjadi resolusi 800x800 pixel, dan juga peningkatan exposure dari bounding-box dengan peningkatan sebesar 25%. Pada Tabel 3 terdapat hasil pre-processing citra.

Tabel 3: Gambar Hasil Pre-processing

No	Hasil pre-processing	Pre-processing
1		<ul style="list-style-type: none"> <li>- Resolusi menjadi 800x800px</li> <li>- Auto-orient</li> <li>- Exposure Bounding-box 25%</li> </ul>
2		<ul style="list-style-type: none"> <li>- Resolusi menjadi 800x800px</li> <li>- Auto-orient</li> <li>- Exposure Bounding-box 25%</li> </ul>
3		<ul style="list-style-type: none"> <li>- Resolusi menjadi 800x800px</li> <li>- Auto-orient</li> <li>- Exposure Bounding-box 25%</li> </ul>

## Pengujian Deteksi Menggunakan Citra Udang dan Bukan Udang

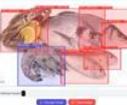
Percobaan yang dilakukan adalah mencoba menguji untuk mendeteksi objek yang didalamnya terdapat image udang dan bukan udang. Terdapat tiga data percobaan yang mengandung objek udang dan bukan udang. Daftar image yang menjadi objek percobaan terdapat pada Tabel 4.

Tabel 4: Contoh Hasil Pengumpulan Data

No	Image	Resolution (px)	Image Notes
1		747x747	Udang dan Cumi
2		750x500	Udang dan berbagai jenis ikan
3		1024x593	Udang, ikan, dan kepiting

Selanjutnya, dari daftar gambar pada Tabel 4 tersebut, dilakukan percobaan mendeteksi gambar-gambar tersebut menggunakan aplikasi website shrimp-detector yang telah dibangun. Tabel 5 merupakan hasil pengamatannya.

Tabel 5: Hasil Deteksi

No	Image	Expected Result		Result	
		Shrimp	Not-shrimp	Shrimp	Not-Shrimp
1		6	8	6	2
2		5	4	6	4
3		2	5	1	5

Pengujian Hasil Akurasi Citra Udang dan Bukan Udang Pada tahap ini dilakukan proses pengujian hasil akurasi deteksi citra udang. Proses

pengujian ini bertujuan untuk memeriksa performa deteksi aplikasi website yang dibuat. Pengujian dilakukan dengan melakukan deteksi citra udang dan bukan udang berdasarkan Tabel 4 sebelumnya. Hasil pengujian deteksi akan berfokus pada penghitungan objek yang terdeteksi. Hasil pengujian tersebut akan disajikan dalam bentuk tabel dan Confusion Matrix.

		Actual	
		1 (Positive)	0 (Negative)
Predicted	1 (Positive)	<b>6 (True Positive)</b>	6 (False Positive)
	0 (Negative)	0 (False Negative)	<b>2 (True Negative)</b>

Gambar 8: Confusion Matrix dari Citra 1

Pada Gambar 8, Jumlah udang yang terdeteksi adalah 6 yang disebut sebagai True Positive. Sedangkan jumlah bukan udang yang terdeteksi adalah 2 yang disebut sebagai True Negative. Jumlah aktual udang yang ada dapat dihitung dari penjumlahan total kolom actual positive yaitu sebanyak 6 (TP) + 0 (FN) = 6. Jumlah aktual bukan udang yang ada dapat dihitung dari penjumlahan total kolom actual negative yaitu sebanyak 6 (FP) + 2 (TN) = 8.

		Actual	
		1 (Positive)	0 (Negative)
Predicted	1 (Positive)	<b>6 (True Positive)</b>	0 (False Positive)
	0 (Negative)	1 (False Negative)	<b>4 (True Negative)</b>

Gambar 9: Confusion Matrix dari Citra 2

Pada Gambar 9, Jumlah udang yang terdeteksi adalah 6 yang disebut sebagai True Positive. Sedangkan jumlah bukan udang yang terdeteksi adalah 4 yang disebut sebagai True Negative. Jumlah aktual udang yang ada dapat dihitung dari penjumlahan total kolom actual positive yaitu sebanyak 6 (TP) + 1 (FN) = 7. Jumlah aktual bukan udang yang ada dapat dihitung dari penjumlahan total kolom actual negative yaitu sebanyak 0 (FP) + 4 (TN) = 4.

		Actual	
		1 (Positive)	0 (Negative)
Predicted	1 (Positive)	<b>1 (True Positive)</b>	0 (False Positive)
	0 (Negative)	1 (False Negative)	<b>5 (True Negative)</b>

Gambar 10: Confusion Matrix dari Citra 3

Pada Gambar 10, Jumlah udang yang terdeteksi adalah 1 yang disebut sebagai True Positive. Sedangkan jumlah bukan udang yang terdeteksi adalah 5 yang disebut sebagai True Negative. Jumlah aktual udang yang ada dapat dihitung dari penjumlahan total kolom actual positive yaitu sebanyak 1 (TP)

+ 1 (FN) = 2. Jumlah aktual bukan udang yang ada dapat dihitung dari penjumlahan total kolom actual negative yaitu sebanyak 0 (FP) + 5 (TN) = 5.

Masing-masing citra yang terdapat pada Tabel 5, selanjutnya dihitung akurasi menggunakan rumus akurasi pada persamaan (1). Pada Tabel 6 terdapat hasil perhitungan akurasi citra udang yang telah dideteksi menggunakan confusion matrix.

Tabel 6: Hasil Perhitungan Akurasi Citra Udang

No	Citra	Hasil Kalkulasi Akurasi
1	Citra 1	$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$ $accuracy = \frac{6+2}{6+2+0+6} \times 100\%$ $accuracy = \frac{8}{14} \times 100\%$ $accuracy = 57,14\%$
2	Citra 2	$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$ $accuracy = \frac{6+4}{6+4+1+0} \times 100\%$ $accuracy = \frac{10}{11} \times 100\%$ $accuracy = 90,9\%$
3	Citra 3	$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$ $accuracy = \frac{1+5}{1+5+1+0} \times 100\%$ $accuracy = \frac{6}{7} \times 100\%$ $accuracy = 85,71\%$

Hasil pengujian akurasi deteksi yang telah dilakukan memiliki tingkat persentase keberhasilan yang variatif dari masing-masing tabel. Dari hasil pengujian, maka diperoleh beberapa hal:

1. Persentase akurasi keberhasilan pengujian citra 1 sebesar 57,14%.
2. Persentase akurasi keberhasilan pengujian citra 2 sebesar 90,9%.
3. Persentase akurasi keberhasilan pengujian citra 3 sebesar 85,71%

Persentase keberhasilan pengujian objek shrimp dan not-shrimp yang terdeteksi menggunakan tiga gambar memperoleh hasil akurasi sebesar 79,91%. Selain itu, tingkat akurasi keberhasilan tertinggi yaitu Gambar 2 dengan tingkat akurasi sebesar 90,9%.

## Penutup

Pada penelitian ini, custom model shrimp detector yang menjadi output penelitian berhasil ditraining menggunakan YOLOv8. Custom model tersebut berhasil mendeteksi objek berupa udang dan bukan udang serta menghitung jumlah udang yang terdeteksi. Custom model dibangun menggunakan custom dataset dengan dua kelas yaitu shrimp dan not-shrimp sebanyak 6630 data. Proses training custom

model membutuhkan waktu selama 7,5 jam dan dilakukan menggunakan Google Colab Pro. Training dilakukan dengan jumlah epoch sebanyak 300 epoch. Rata-rata nilai precision didapatkan sebesar 92,57%, sedangkan nilai recall diperoleh rata-rata sebesar 88,2%. Akurasi dari model diperoleh nilai sebesar 93,2%. Hasil tersebut menggambarkan bahwa model cukup stabil dalam mengklasifikasikan objek secara tepat. Hasil pengujian dari confusion matrix terhadap citra hasil deteksi menghasilkan nilai akurasi sebesar 79,91%.

## Daftar Pustaka

- [1] S. Rejeki, R. W. Aryati dan L. L. Widowati, "Pengantar Akuakultur", Semarang: UNDIP Press Semarang, 2019.
- [2] S. S. Fatimah, S. Marwanti, and S. Supardi, "Export Performance of Indonesian Shrimp In the United States During 2009-2017: A Constant Market Share Model Approach", Jurnal Sosial Ekonomi Kelautan dan Perikanan, vol. 15, no. 1, pp. 57-67, doi: <http://dx.doi.org/10.15578/jsekp.v15i1.7677>, Jun. 2020.
- [3] Agus Suherman, "2020, KKP Targetkan Konsumsi Ikan 56,39 KG", Kementerian Kelautan dan Perikanan Republik Indonesia, diakses daring pada: <https://kkp.go.id/artikel/16451-2020-kkp>, 2020.
- [4] Anonim, "Budidaya Udang Vaname di Tambak Milenial. Situbondo", Balai Perikanan Budidaya Air Payau Situbondo, Kementerian Kelautan dan Perikanan, diakses daring pada: <https://kkp.go.id/djpb/bpbapsitubondo/artikel/34255-budidaya-udang-vaname-di-tambak-milenial-millennial-shrimp-farming-msf>, 2021.
- [5] Lusiana Rahma, Hadi Syaputra, A.Haidar Mirza dan Usan Dian Purnamasari, "Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO(You Only Look Once)", Jurnal Nasional Ilmu Komputer, vol. 2, no. 3, 2021.
- [6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review", IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11. Institute of Electrical and Electronics Engineers Inc., pp. 3212-3232, doi: 10.1109/TNNLS.2018.2876865, Nov. 01, 2019.
- [7] A. R. Pathak, M. Pandey and S. Rautaray, "Application of Deep Learning for Object Detection", in Procedia Computer Science, Elsevier B.V., pp. 1706-1717. doi: 10.1016/j.procs.2018.05.144, 2018.

- [8] D. Putra dan Westriningsih, "Pengolahan citra digital", Yogyakarta: ANDI, 2010.
- [9] Supono, "Teknologi Produksi Udang", Bandar Lampung: Universitas Negeri Lampung, 2017.
- [10] Y. Wang, H. Wang and Z. Xin, "Efficient detection model of steel strip surface defects based on YOLO-V7", *IEEE Access*, doi: 10.1109/ACCESS.2022.3230894, 2022.
- [11] P. Jiang, D. Ergu, F. Liu, Y. Cai and B. Ma, "A Review of Yolo Algorithm Developments", *Procedia Comput Sci*, vol. 199, <https://doi.org/10.1016/j.procs.2022.01.135>, 2022.
- [12] N. Buhl, "YOLO models for Object Detection Explained [YOLOv8 Updated]", diakses daring pada: <https://encord.com/blog/yolo-object-detection-guide/#h2>, February 2023.
- [13] M. A. bin Zuraimi and F. H. K. Zaman, "Vehicle Detection and Tracking using YOLO and DeepSORT", 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 23–29, 2021.
- [14] Z. Wei, M. Chang and Y. Zhong, "Fruit Freshness Detection Based on YOLOv8 and SE attention Mechanism", *Academic Journal of Science and Technology*, vol. 6, no. 1, 2023.
- [15] J. Solawetz and Francesco, "What is YOLOv8? The Ultimate Guide", diakses daring pada: <https://blog.roboflow.com/whats-new-in-yolov8/#yolov8-accuracy-improvements>, January 2023.
- [16] Ultralytics Team, "Introducing Ultralytics YOLOv8", diakses daring pada: <https://ultralytics.com/article/Introducing-Ultralytics-YOLOv8>, Oct 2023.
- [17] S. Armalivia, Z. Zainuddin, A. Achmad and Muh. A. Wicaksono, "Automatic Counting Shrimp Larvae Based You Only Look Once (YOLO)", 2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), pp. 1–4, 2021.
- [18] Anonym, "Roboflow Overview", Roboflow, diakses daring pada: <https://docs.roboflow.com/>, 2023.
- [19] A. R. Wasril, M. S. Ghozali dan M. B. Mustafa, "Pembuatan Pendeteksi Obyek dengan Metode You Only Look Onc (YOLO) untuk Automated Teller Machine (ATM)", *Majalah Ilmiah UNIKOM*, DOI 10.34010/MIU.V17I1.2240, 2019.
- [20] A. N. Sugandi dan B. Hartono, "Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO", in *Prosiding The 13th Industrial Research Workshop and National Seminar*, Bandung, Jul. 2022.
- [21] A. Menon, B. Omman and A. S, "Pedestrian Counting Using Yolo V3", 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), pp. 1–9, 2021.
- [22] R. G. Guntara, "Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7", *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 55–60, Jan. 2023.
- [23] S. Tata, A. Popescul, M. Najork and M. Colagrosso, "Quick Access: Building a Smart Experience for Google Drive", *KDD 2017 Applied Data Science Paper*, Aug. 2017.
- [24] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir and A. ul Haq, "Object Detection through Modified YOLO Neural Network", *Scientific Programming*, vol. 2020, Article ID 8403262, 10 pages, <https://doi.org/10.1155/2020/8403262>, 2020.
- [25] K. S. Nugroho, "Confusion Matrix untuk Evaluasi Model pada Supervised Learning", *Medium*, diakses daring pada: <https://ksnugroho.medium.com/confusion-matrix-untuk-evaluasi-model-pada-supervised-machine-learning-bc4b1ae9ae3f>, 2019.