Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

Journal Homepage: http://journal.lembagakita.org/index.php/jtik

Optimasi Deteksi Objek Dengan Segmentasi dan Data Augmentasi Pada Hewan Siput Beracun Menggunakan Algoritma *You Only Look Once* (YOLO)

Dadang Iskandar Mulyana [1], Reyga Ferdiansyah Putra [2*]

[1,2] Program Studi Teknik Informatika, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, Kota Jakarta Timur, Daerah Khusus Ibu kota Jakarta, Indonesia.*

a r t i c l e i n f o

Article history:
Received 14 July 2023
Received in revised form
29 September 2023
Accepted 20 November
2023
Available *online* January 2024

DOI:
https://doi.org/10.35870/jtik.v8i1.1391.

Keywords:
Object detection; Optimization; Segmentation; Augmentation; Yolo (You Only Look Once).

Kata Kunci:
Optimasi; Deteksi Objek; Segmentasi; Augmentasi; Yolo (You Only Look Once).

a b s t r a c t

Significant progress has been achieved in visual detection, and the abundance of remarkable models has been proposed. Object detection is an important task in various popular fields such as medical diagnosis, robot navigation, autonomous driving, augmented reality, and more. This research aims to develop an optimized object detection model with segmentation and augmentation using the YOLO (You Only Look Once) algorithm for recognizing 10 types of toxic snails in images and videos. The dataset consists of 5,720 images that have been augmented using Roboflow, divided into 5,000 images for training, 480 images for validation, and 240 images for testing. With a model training of 50 epochs, YOLOv8 Box_Curve F1-Confidence achieved "0.98 at 0.625", Precision Confidence "1.00 at 0.997", Precision Recall "0.987 mAP@0.5", and Recall Confidence "1.00 at 0.000". Mask_Curve, YOLOv8 achieved "0.98 at 0.625", Precision Confidence "1.00 at 0.997", Precision Recall "0.986 mAP@0.5", and Recall Confidence "1.00 at 0.000".

a b s t r a k

Kemajuan signifikan telah diperoleh dalam deteksi visual, dan kelimpahan dari model yang luar biasa telah diusulkan. Deteksi objek adalah tugas penting dibanyak bidang populer seperti diagnosis medis, navigasi robot, penggerak matic otomatis, augmented reality, dan sebagainya. Penelitian ini bertujuan untuk dapat mengembangkan dan mengevaluasi model Optimasi deteksi objek dengan segmentasi dan augmentasi menggunakan algoritma YOLO (You Only Look Once) untuk pengenalan 10 jenis siput beracun pada gambar dan video. Dengan jumlah dataset sebanyak 5.720 gambar yang sudah di augmentasi lewat Roboflow dan terbagi menjadi 5.000 gambar data training, 480 gambar data validation dan 240 gambar data testing. Dengan pelatihan model 50 epoch, YOLOv8 Box_Curve F1-Confidence mendapatkan "0.98 at 0.625", Precision Confidence "1.00 at 0.997", Precision Recall "0.987 mAP@0.5", dan Recall Confidence "1.00 at 0.000". Mask_Curve YOLOv8 mendapatkan "0.98 at 0.625", Precision Confidence "1.00 at 0.997", Precision Recall "0.986 mAP@0.5", dan Recall Confidence "1.00 at 0.000".

Corresponding Author. Email: reygafp@gmail.com [2].

ACM Computing Classification System (CCS)
EBSCOhost
Communication and Mass Media Complete (CMMC)

1. Latar Belakang

Dalam beberapa tahun terakhir, kemajuan signifikan telah diperoleh dalam deteksi visual, dan kelimpahan dari model yang luar biasa telah diusulkan [1]. Deteksi objek adalah tugas penting dibanyak bidang populer seperti diagnosis medis, navigasi robot, penggerak matic otomatis, *augmented reality*, dan sebagainya [2]. Oleh karena itu, teknologi deteksi objek dan *instance segmentation* pada citra digital dapat digunakan untuk mempermudah dan meningkatkan pengenalan siput beracun. Untuk segmentasi gambar video, partisi yang diperoleh untuk frame saat ini dapat digunakan untuk menghasilkan penanda untuk frame berikutnya [3].

Metode YOLO *(You Only Look Once)* adalah metode pendeteksian objek populer yang mendukung sistem dapat bekerja secara real-time [4]. YOLO memproses gambar secara real-time pada empat puluh lima (45) *frames per second* [5]. YOLO merupakan salah satu algoritma *one stage detector* untuk deteksi objek. Pada YOLO tidak diperlukan pemilihan *region*. Setiap citra pada YOLO akan dibentuk *grid* dengan ukuran S x S, dimana setiap grid pada citra akan memprediksi *bounding box* B dan Nilai probabilitas kelas C. Terdapat lima prediksi dalam *bounding box* B yaitu *confidence score* (p), x, y, w, dan h. Nilai *confidence score* menyatakan ada atau tidaknya objek pada grid tertentu. Nilai x dan y menyatakan titik pusat dari suatu objek, w dan h merupakan lebar dan tinggi *bounding box* dari objek [6]. Dalam konteks YOLO *(You Only Look Once),* yang merupakan algoritma populer untuk deteksi objek *real-time* dalam penglihatan komputer, augmentasi digunakan untuk meningkatkan kinerja model dan memperluas keragaman data pelatihan. Rotasi, flip, dan penambahan noise acak adalah hal biasa metode augmentasi data [7][8].

Beberapa Penelitian menggunakan alghoritma Yolo telah dilakukan dengan menggunakan berbagai metode. Seperti penelitian oleh Azizah & Fatichah (2023) Menerapkan Augmentasi Model Warna HSV pada Gambar Mushaf pada 17 data uji pada model YOLOv7, YOLOv6, dan YOLOv5 mendapatkan akurasi 80%, 69%, dan 71% [4]. Penelitian oleh Zhao & Zhu (2023) mendapatkan akurasi 59,00% dari dataset SeaDronesee berisi 14.227 gambar [9]. Penelitian yang dilakukan Wang, Bochkovskiy, & Liao (2023) mendapati akurasi maksimumYOLOv7 seebsar 56,8% AP [10]. Sejalan dengan penelitian oleh Bakirman (2023) menghasilkan akurasi YOLOv7 dan YOLOv5 sebesar 68,11% dan 69,69% [11]. Sapitri *et al* (2023) melakukan 40 video ekokardiografi janin dilatih dengan arsitektur YOLOv7 menghasilkan presisi rata-rata tertinggi sebesar 82,10%. Hasil penelitian ini juga sejalan dengan penelitian Siddique *et al* (2023) dengan menggunakan Yolov7 mendapatkan akurasi lebih timggi yaitu sebesar 97% [13]. Penelitian lainnya dengan penggunaan *Yolo V7 Deep Learning Algorithm* juga didapati oleh Chao *et al* (2023) Dimana diketahui Model Yolov7 mendapatkan akurasi tertinggi 94,69% [14].

Berdasarkan penelitian dan permasalahan yang terkait sebelumnya, Penulis akan menggabungkan konsep-konsep optimasi, segmentasi, data augmentasi, dan pendekatan YOLO *(You Only Look Once)* untuk mengembangkan metode yang lebih canggih dalam deteksi dan segmentasi siput beracun. Sehingga muncul ide penulis untuk membuat penelitian dengan Optimasi Deteksi Objek dengan Segmentasi dan Data Augmentasi Pada Hewan Siput Beracun Menggunakan Algoritma *You Only Look Once*. Penelitian ini diharapkan dapat mengembangkan dan mengevaluasi model deteksi objek dengan segmentasi yang efektif dan akurat untuk pengenalan 10 jenis siput beracun pada gambar dan video.

2. Metode Penelitian

Perangkat keras yang digunakan untuk mengimplementasikan sistem adalah sebagai berikut:

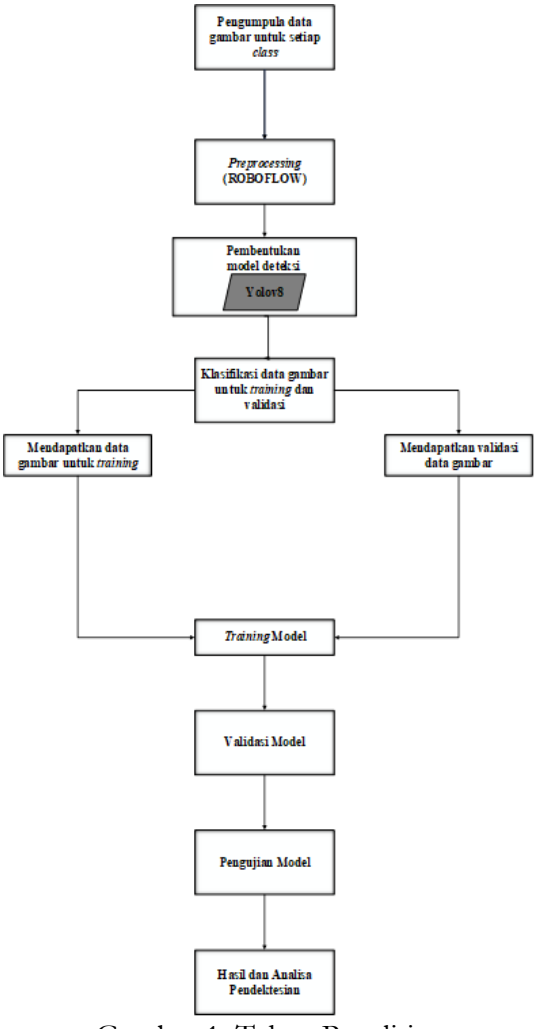
Tabel 1. Spesfikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1.	*Processor*	*Intel Core* I5
2.	*Memory*	DDR3 12 GB
3.	*Hardisk* HDD	1 TB
4.	*Hardisk* SDD	130 GB

Perangkat lunak yang digunakan untuk membangun dan mengimplementasikan sistem adalah sebagai berikut:

Tabel 2. Perangkat Lunak Yang Digunakan

No	Perangkat Keras	Kegunaan
1.	*Google Colab*	Memprogram, menjalankan, dan menguji pengembangan model
2.	*Roboflow*	Pengembang dalam mengelola dan mengoptimalkan dataset, serta mempercepat siklus pengembangan model visual komputer.



Gambar 1. Tahap Penelitian

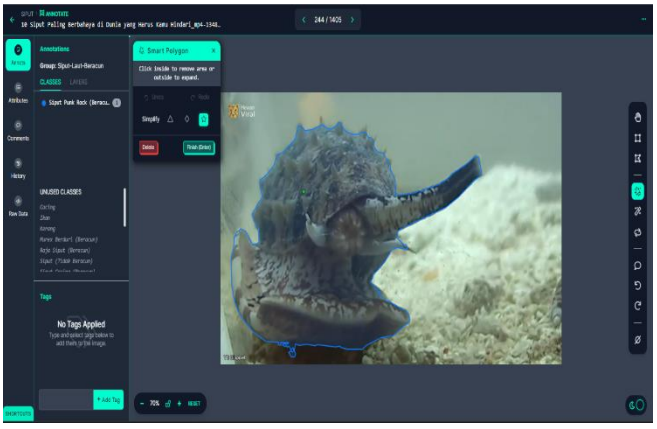
1) Pengumpulan *dataset*
Pertama-tama, mempersiapkan *dataset* yang berisi gambar-gambar siput beracun yang diambil melalui Google. *Dataset* dalam penelitian ini bersifat *public*.

Tabel 3. *Dataset* 10 Jenis Siput Beracun

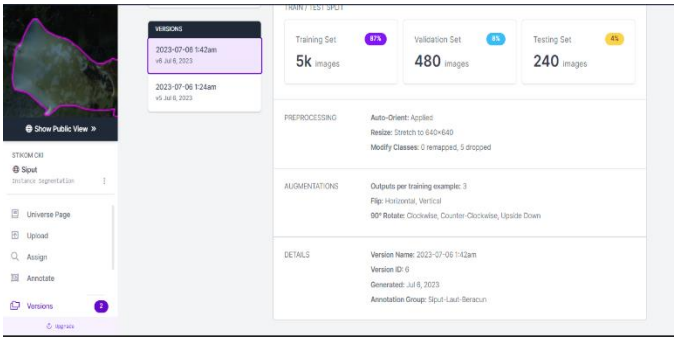
No	Nama Siput	Gambar
1.	Murex Berduri (Beracun)	
2.	Raja Siput (Beracun)	
3.	Siput Pengebor Cangkang (Beracun)	
4.	Siput Kerucut (Beracun)	
5.	Siput Pembunuh (Beracun)	
6.	Siput Darat Raksasa Afrika (Beracun)	

7.	Siput Pemakan Daging (Beracun)	
8.	Siput Samudera Biru (Beracun)	
9.	Siput Cacing (Beracun)	
10.	Siput Punk Rock (Beracun)	

2) Melakukan *preprocessing* dengan anotasi atau label image pada suatu object dengan mask *instance segmentation*. Memilih *smart polygon* pada fitur di roboflow agar otomatis menganotasikan objek siput beracun seperti pada gambar 2. Dan melakukan *augmentation*, yang digunakan flip horizontal dan vertical, Lalu menggunakan 90 derajat rotate dengan *clockwise, counter – clockwise,* dan *upside down*. Dengan metode ini maka jumlah dataset yang didapatkan semakin banyak dan memadai. Dan pada gambar 3, menunjukkan jumlah 5.000 gambar untuk data training, 480 gambar untuk data *validation* dan 240 gambar untuk data *testing*.



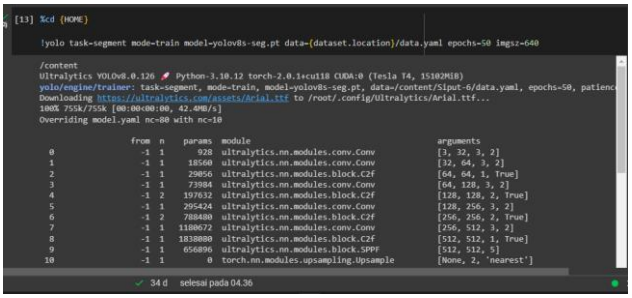
Gambar 2. Anotasi pada citra Siput beracun



Gambar 3. *Augmentation* di Roboflow

3) Mengekstrak file dataset atau memanggil dataset dalam roboflow dengan memasukkan API KEY seperti pada gambar 4. Setelah dataset gambar disiapkan, kita dapat melatih model deteksi objek dan *instance segmentation* menggunakan arsitektur YOLOv8. Proses pelatihan melibatkan beberapa langkah, seperti memilih hyperparameter yang sesuai, memilih dataset pelatihan dan validasi, dan melakukan pelatihan pada model yang ditunjukkan pada gambar 5. Sesuaikan parameter model seperti ukuran jendela geser, jumlah lapisan konvolusi, dan jumlah neuron di lapisan terakhir.

Gambar 4. Pemograman *Import Dataset* Dari Roboflow ke Google Colab YOLOv8

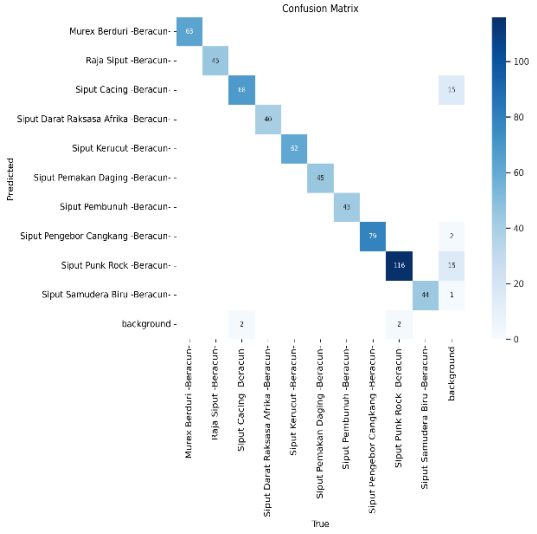


Gambar 5. Pemograman Pelatihan Model Pada YOLOv8

4) Evaluasi Model: Parameter penilaian untuk menguji akurasi model dan tingkat confident dalam melakukan klasifikasi terhadap data baru [15]. Setelah model dilatih, model dapat dievaluasi dengan menggunakan dataset yang tidak pernah dilihat sebelumnya *(testing set)*. Model dapat dievaluasi dengan menggunakan metrik evaluasi seperti *Confusion Matrix, Results, Precision, Recall, dan F1-Score.*

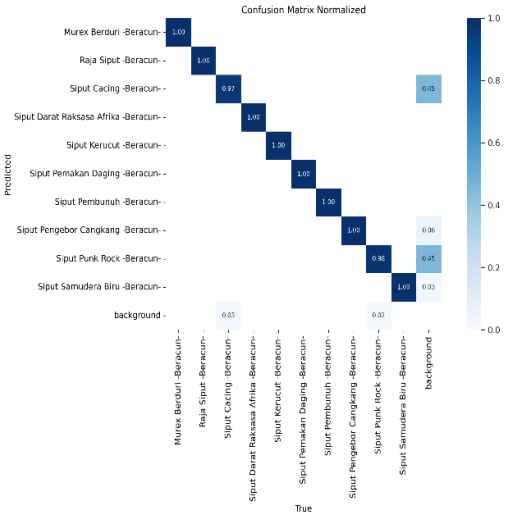
Deteksi Objek dan Segmentasi: Setelah model siap digunakan, kita dapat melakukan deteksi objek dan instance segmentation pada gambar siput beracun menggunakan model tersebut. Untuk deteksi objek, model akan memberikan bounding box yang menunjukkan lokasi dari objek pada gambar, sedangkan untuk instance segmentation, model akan memberikan mask yang menunjukkan lokasi dari objek pada gambar.

3. Hasil dan Pembahasan



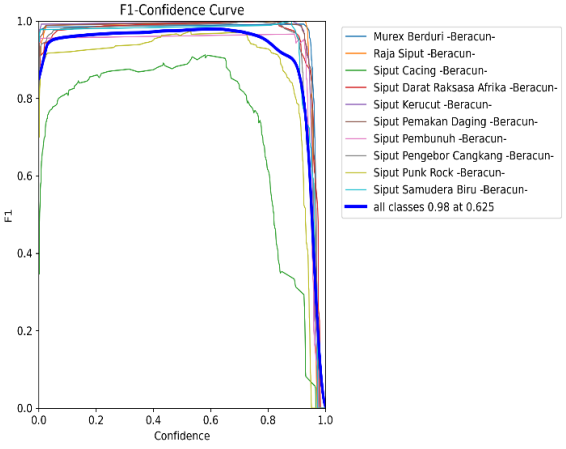
Gambar 6. *Confusion Matrix*

Confusion matrix digunakan untuk mengevaluasi kinerja suatu model klasifikasi dengan membandingkan hasil prediksi model dengan label sebenarnya dari objek. Yang ditunjukkan pada gambar 6, terdapat 10 *class* siput beracun dalam *confusion matrix*.



Gambar 7. *Confusion Matrix Normalized*

Confusion matrix normalized, juga dikenal sebagai confusion matrix yang dinormalisasi, adalah representasi confusion matrix yang menggambarkan proporsi atau persentase dari kesalahan klasifikasi dalam setiap kelas. Hal ini membantu dalam memperoleh pemahaman yang lebih baik tentang kinerja model klasifikasi dalam mengklasifikasikan berbagai kelas seperti yang ditunjukkan pada gambar 7. *Box curve* adalah kurva yang menunjukkan hubungan antara nilai *precision* dan *recall* pada tingkat IoU *(Intersection over Union)* tertentu untuk setiap kelas objek yang dideteksi oleh model deteksi objek seperti YOLO *(You Only Look Once)*.

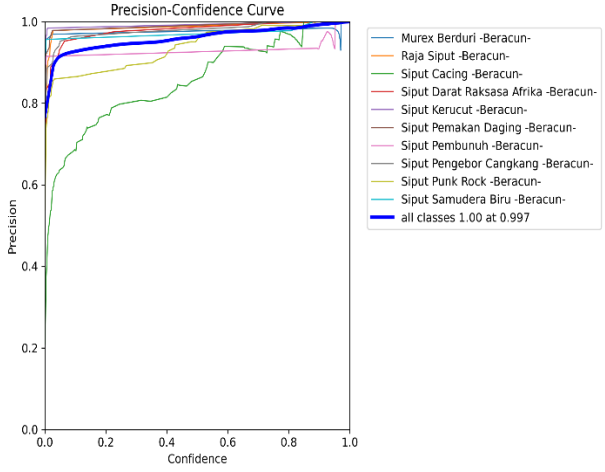


Gambar 8. *Box*_F1-*Confidence Curve*

F1 *confidence curve* adalah kurva yang menunjukkan hubungan antara F1 *score* dan tingkat kepercayaan *(confidence level)* untuk semua kelas objek yang dideteksi oleh model deteksi objek seperti YOLO *(You Only Look Once)*. Setiap titik pada kurva menunjukkan F1 *score* pada suatu tingkat kepercayaan tertentu. F1 *score* adalah *matrix* evaluasi yang mengukur keseimbangan antara presisi *(precision)* dan *recall* pada model deteksi objek. Sedangkan tingkat kepercayaan *(confidence level)* adalah tingkat keyakinan model dalam memprediksi suatu objek yang dinyatakan sebagai nilai probabilitas.

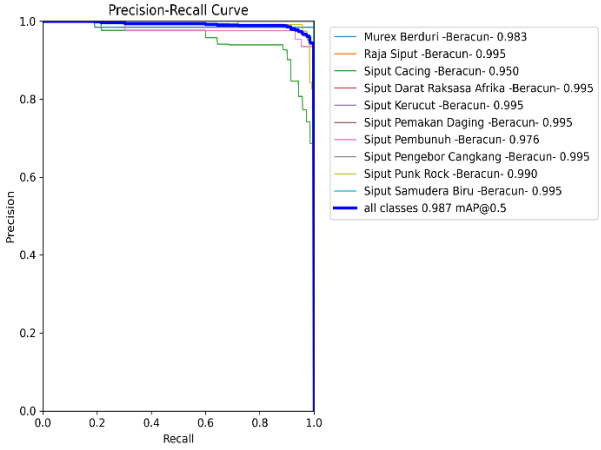
Seperti pada gambar 8, deteksi objek dan segmentasi mendapatkan "*all classes* = 0.98" mengindikasikan bahwa semua kelas objek yang dideteksi oleh model YOLOv8 memiliki nilai F1 *score* sebesar 0.98. Nilai ini menunjukkan bahwa model memiliki performa deteksi objek yang sangat baik dan akurat untuk semua kelas objek. Sedangkan "*at* 0.625" mengindikasikan bahwa tingkat kepercayaan *(confidence level)* yang memberikan F1 *score* sebesar 0.98 adalah sebesar 0.625.

Artinya, ketika model YOLOv8 memberikan prediksi dengan tingkat kepercayaan *(confidence level)* sebesar 0.625, maka F1 *score* yang dihasilkan untuk semua kelas objek adalah sebesar 0.98. Secara umum, semakin tinggi nilai F1 *score* dan semakin rendah nilai tingkat kepercayaan *(confidence level)* yang diperlukan untuk mencapai F1 *score* tersebut, semakin baik performa deteksi objek dari model tersebut. Oleh karena itu, F1 *confidence curve* dapat digunakan sebagai alat untuk memilih tingkat kepercayaan yang optimal pada model deteksi objek.



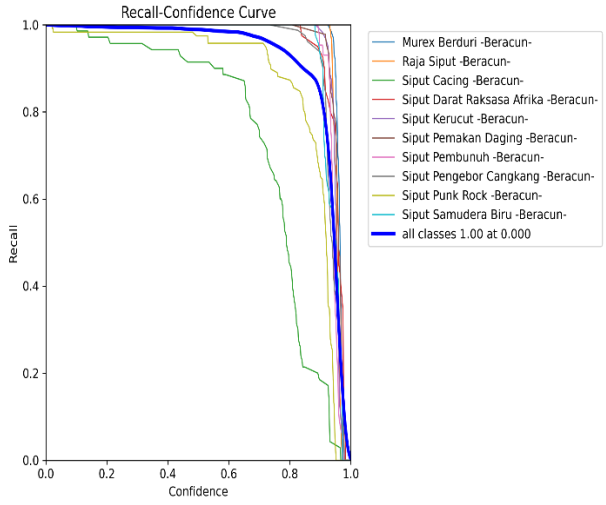
Gambar 9. *Box_Precision-Confidence Curve*

Pada gambar 9, ditunjukkan *precision confidence curve*, "*all classes* 1.00" mengindikasikan bahwa semua kelas objek yang dideteksi oleh model YOLOv8 memiliki nilai presisi sebesar 1.00. Nilai ini menunjukkan bahwa model memiliki performa deteksi objek yang sangat akurat dan tidak membuat kesalahan dalam melakukan prediksi. Sedangkan "*at* 0.997" mengindikasikan bahwa tingkat kepercayaan *(confidence level)* yang memberikan presisi sebesar 1.00 adalah sebesar 0.997. Artinya, ketika model YOLOv8 memberikan prediksi dengan tingkat kepercayaan *(confidence level)* sebesar 0.997, maka semua prediksi tersebut akan menjadi benar atau tidak ada kesalahan dalam prediksi yang dilakukan.

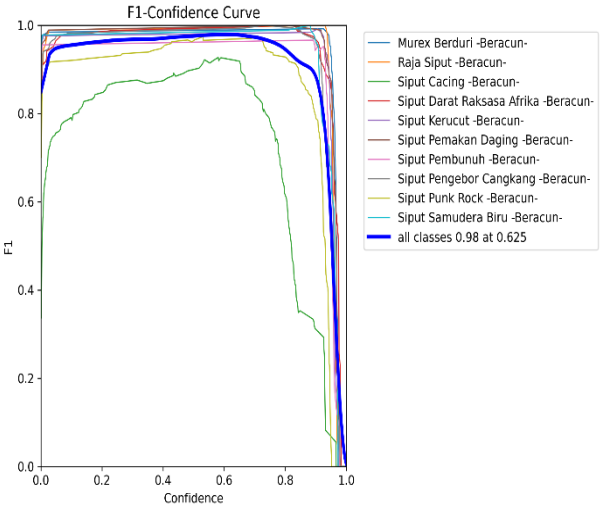


Gambar 10. *Box_Precision-Recall Curve*

Dan juga *precision recall curve* "*all classes* 0.987" mengindikasikan bahwa nilai rata-rata presisi dan recall untuk semua kelas objek adalah sebesar 0.987. Ini menunjukkan bahwa model memiliki performa deteksi objek yang cukup tinggi untuk semua kelas objek yang ada. Sedangkan "mAP@0.5" adalah nilai rata-rata dari *Average Precision* (AP) pada tiap kelas objek yang dihitung dengan menggunakan IoU *threshold* sebesar 0.5. IoU *threshold* adalah parameter yang digunakan untuk menentukan apakah kotak pembatas *(bounding box)* yang dihasilkan oleh model deteksi objek benar-benar menangkap objek yang sebenarnya. mAP@0.5 mengukur sejauh mana model deteksi objek mampu menghasilkan kotak pembatas yang akurat pada suatu tingkat IoU tertentu.



Gambar 11. *Box_Recall-Confidence Curve*

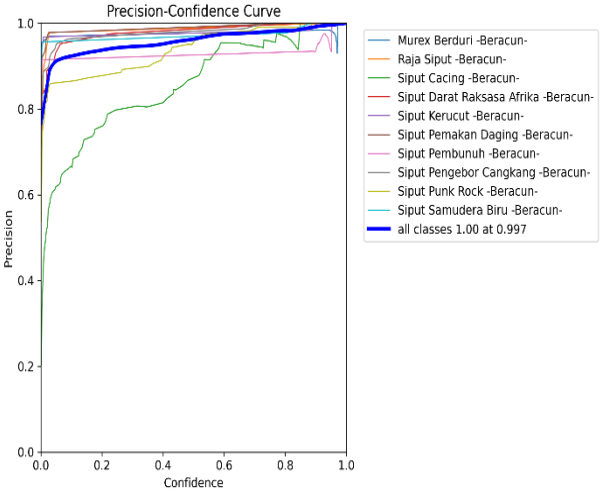


Gambar 12. *Mask*_F1-*Confidence Curve*

Dalam *recall confidence curve*, "*all classes* 1.00" mengindikasikan bahwa nilai recall untuk semua kelas objek adalah sebesar 1.00. Ini menunjukkan bahwa model mampu menemukan semua objek yang sebenarnya untuk setiap kelas objek yang ada. Sedangkan "at 0.000" menunjukkan bahwa recall dihitung pada nilai *confidence threshold* sebesar 0.000. Hal ini menunjukkan bahwa model mampu menemukan semua objek yang sebenarnya bahkan pada kasus-kasus di mana confidence threshold sangat rendah atau ketat.

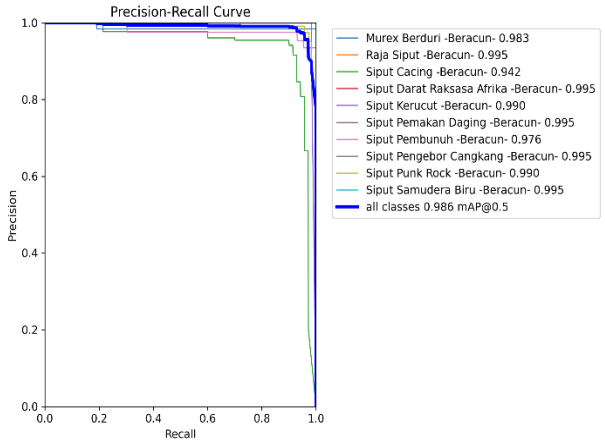
Mask curve adalah kurva yang menunjukkan hubungan antara nilai precision dan recall pada tingkat IoU *(Intersection over Union)* tertentu untuk deteksi objek dengan segmentasi piksel *(instance segmentation)* pada setiap kelas objek yang dideteksi oleh model seperti *Mask* R-CNN. *Mask* R-CNN adalah salah satu model deteksi objek yang menggabungkan deteksi objek dan segmentasi piksel untuk menghasilkan deteksi objek yang lebih akurat.

Pada F1 *confidence mask curve* pada gambar 10 , YOLOv8 mendapatkan "*all classes* 0.98 *at* 0.625".



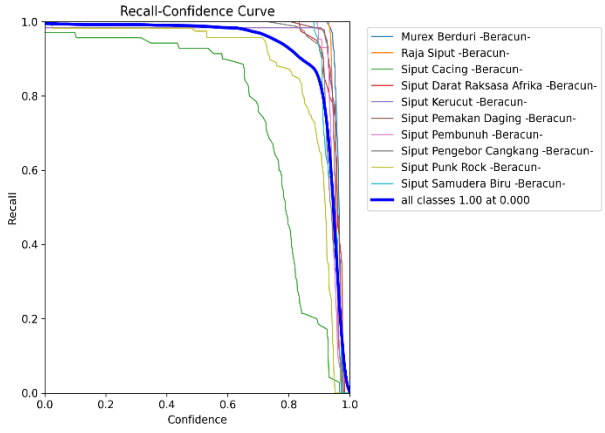
Gambar 13. *Mask_Precision-Confidence Curve*

Precision confidence mask curve mendapatkan " *all classes* 1.00 *at* 0.997".



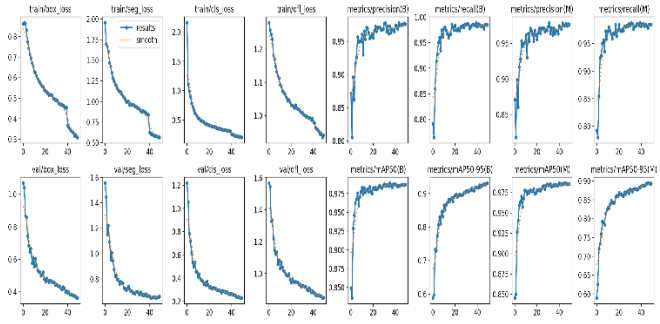
Gambar 14. *Mask_Precision-Recall Curve*

Precision recall mask curve mendapatkan " *all classes* 0.986 mAP@0.5 ".



Gambar 15. *Mask_Recall-Confidence Curve*

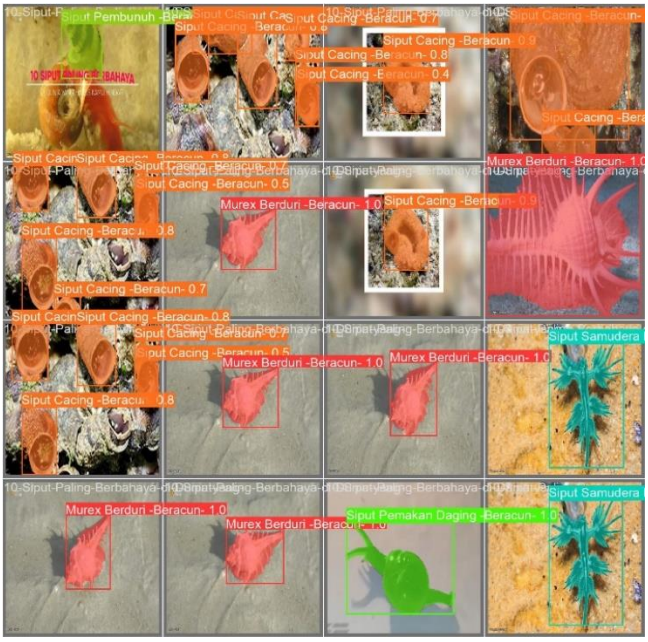
Dan *recall confidence mask curve* mendapatkan " *all classes* 1.00 *at* 0.000".



Gambar 16. *Results*

Pada gambar 16. adalah hasil dari pelatihan model sebanyak 50 *epoch*. model tersebut dapat digunakan untuk mendeteksi objek pada gambar atau video yang belum pernah dilihat sebelumnya. Proses deteksi objek melibatkan melewati gambar atau video melalui model YOLOv8, di mana model tersebut akan menentukan lokasi dan label kelas objek pada gambar atau video tersebut.

Setelah model siap digunakan, kita dapat melakukan deteksi objek dan *instance segmentation* pada gambar siput beracun menggunakan model tersebut. Untuk deteksi objek, model akan memberikan *bounding box* yang menunjukkan lokasi dari objek pada gambar, sedangkan untuk *instance segmentation*, model akan memberikan mask yang menunjukkan lokasi dari objek pada gambar.



Gambar 17. *Val_batch* YOLOv8

Val_batch pada YOLO *(You Only Look Once)* adalah parameter yang mengatur jumlah data gambar yang diproses dalam satu iterasi ketika melakukan evaluasi pada model. Secara lebih spesifik, *val_batch* digunakan saat melaksanakan evaluasi model pada data validasi. Dalam pelatihan *deep learning* , setiap iterasi pada model biasanya melibatkan penghitungan gradien dan pembaharuan bobot berdasarkan data pelatihan. Namun, ketika melakukan evaluasi pada data validasi, pembaharuan bobot tidak dilakukan. Sebaliknya, model hanya digunakan untuk memperoleh prediksi pada data validasi dan kemudian menghitung metrik evaluasi seperti akurasi atau AP *(Average Precision)*.

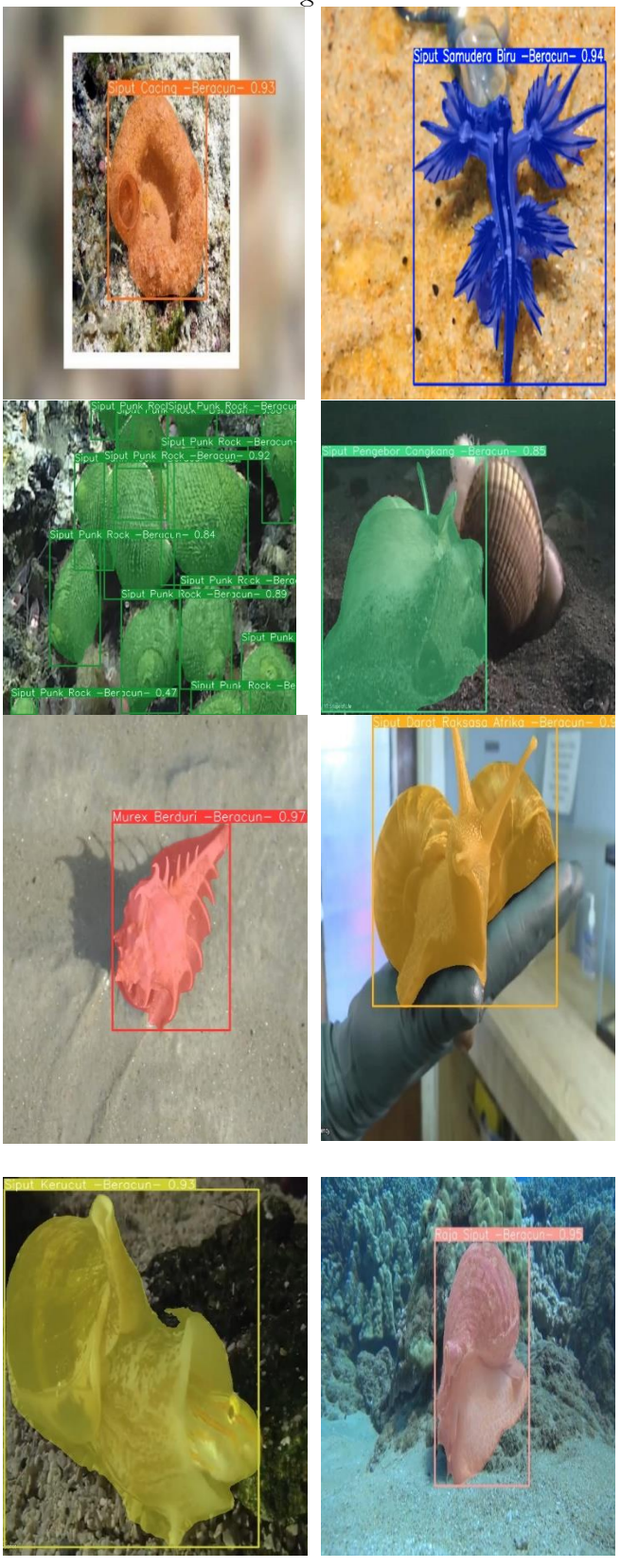
Dalam YOLO, *val_batch* dapat diatur sesuai dengan kapasitas memori GPU yang digunakan untuk menjalankan evaluasi model. Semakin besar *val_batch*,

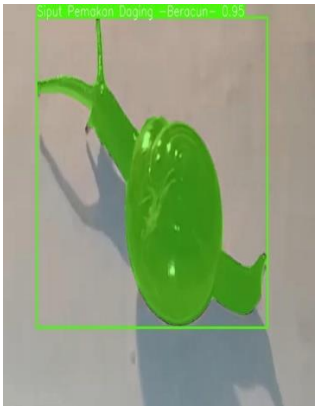
semakin banyak data gambar yang dapat diproses dalam satu iterasi sehingga evaluasi dapat dilakukan dengan lebih cepat. Namun, *val_batch* yang terlalu besar juga dapat menyebabkan kekurangan memori dan memperlambat evaluasi. Secara umum, *val_batch* pada YOLO dapat diatur dengan nilai antara 1 hingga sekitar 64, tergantung pada kapasitas memori GPU yang tersedia dan ukuran gambar yang digunakan pada data validasi.

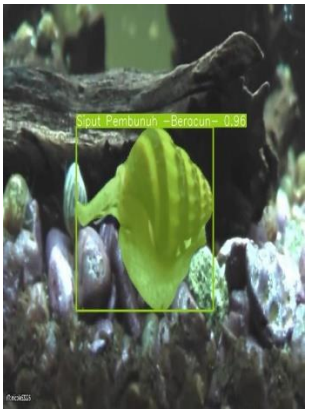
Gambar 18. *Train_batch* YOLOv8

Train_batch pada YOLO *(You Only Look Once)* adalah parameter yang mengatur jumlah data gambar yang diproses dalam satu iterasi ketika melakukan pelatihan pada model. Secara lebih spesifik, *train_batch* digunakan saat melaksanakan pelatihan pada model YOLO. Dalam pelatihan *deep learning* , setiap iterasi pada model biasanya melibatkan penghitungan gradien dan pembaharuan bobot berdasarkan data pelatihan. *Train_batch* mengontrol jumlah data gambar yang diproses dalam satu iterasi sehingga dapat mempengaruhi waktu pelatihan dan memori yang dibutuhkan. Semakin besar nilai *train_batch*, semakin banyak data gambar yang dapat diproses dalam satu iterasi, yang dapat mempercepat waktu pelatihan. Namun, nilai *train_batch* yang terlalu besar juga dapat menyebabkan kekurangan memori dan memperlambat pelatihan. Selain itu, nilai *train_batch* juga dapat mempengaruhi stabilitas pelatihan. Saat menggunakan nilai *train_batch* yang besar, gradien yang dihitung dapat menjadi lebih tidak stabil dan menyebabkan pelatihan gagal. Oleh karena itu, nilai *train_batch* yang optimal harus dipilih berdasarkan kapasitas memori dan kestabilan pelatihan.

Tabel 4. *Testing set* YOLOv8







Dengan teknik *instance segmentation*, kita dapat mengenali siput beracun pada citra digital dan video. Dengan akurasi yang tinggi dan menghindari kesalahan identifikasi yang dapat berdampak pada keselamatan manusia dan lingkungan.

4. Kesimpulan

Dalam kesimpulan, penelitian ini memberikan kontribusi dalam pengembangan model deteksi objek dengan menerapkan teknik optimasi, segmentasi, dan data augmentasi. Hasilnya menunjukkan bahwa model deteksi objek dan segmentasi dapat digunakan untuk mengenali dan memisahkan objek siput beracun secara akurat. Penerapan teknik augmentasi data juga meningkatkan kinerja deteksi dan segmentasi pada hewan siput beracun. Dengan pelatihan model sebanyak 50 *epoch,* nilai YOLOv8 *Box_Curve* F1-*Confidence* mendapatkan "0.97 *at* 0.722", *Precision Confidence* "1.00 *at* 0.985", *Precision Recall* "0.981 mAP@0.5", dan *Recall Confidence* "1.00 *at* 0.000". Dan pada *Mask_Curve* YOLOv8 mendapatkan "0.97 *at* 0.722", *Precision Confidence* "1.00 *at* 0.985", *Precision Recall* "0.977 mAP@0.5", dan *Recall Confidence* "1.00 *at* 0.000".

5. Daftar Pustaka

[1] He, W., Huang, Z., Wei, Z., Li, C. and Guo, B., 2019. TF-YOLO: An improved incremental network for real-time object detection. *Applied Sciences*, *9*(16), p.3225. DOI: https://doi.org/10.3390/app9163225.

[2] Fang, W., Wang, L. and Ren, P., 2019. Tinier-YOLO: A real-time object detection method for constrained environments. *Ieee Access*, *8*, pp.1935-1944. DOI: https://doi.org/10.1109/ACCESS.2019.2961959.

[3] Meyer, F., 2019. Watersheds and Flooding: a Segmentation Golden Braid

[4] Azizah, A.N. and Fatichah, C., 2023. Tajweed-YOLO: Object Detection Method for Tajweed by Applying HSV Color Model Augmentation on Mushaf Images. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *7*(2), pp.236-245. DOI: https://doi.org/10.29207/resti.v7i1.4739.

[5] Khairunnas, K., Yuniarno, E.M. and Zaini, A., 2021. Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot. *Jurnal Teknik ITS*, *10*(1), pp.A50-A55. DOI: http://dx.doi.org/10.12962/j23373539.v10i1.61622.

[6] Thoriq, M.Y.A., Siradjuddin, I.A. and Permana, K.E., 2023. Deteksi Wajah Manusia Berbasis One Stage Detector Menggunakan Metode You Only Look Once (Yolo). *Jurnal Teknoinfo*, *17*(1), pp.66-73. DOI: https://doi.org/10.33365/jti.v17i1.1884.

[7] Li, W., Chen, C., Zhang, M., Li, H. and Du, Q., 2018. Data augmentation for hyperspectral image classification with deep CNN. *IEEE Geoscience and Remote Sensing Letters*, *16*(4), pp.593-597. DOI: https://doi.org/10.1109/LGRS.2018.2878773.

[8] Zhang, M., Li, W. and Du, Q., 2018. Diverse region-based CNN for hyperspectral image classification. *IEEE Transactions on Image Processing*, *27*(6), pp.2623-2634. DOI: https://doi.org/10.1109/TIP.2018.2809606.

[9] Zhao, L. and Zhu, M., 2023. MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography. *Drones*, *7*(3), p.188. DOI: https://doi.org/10.3390/drones7030188.

[10] Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M., 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7464-7475). Source code is released in https://github.com/WongKinYiu/yolov7.

[11] BAKIRMAN, T., 2023. An Assessment of YOLO Architectures for Oil Tank Detection from SPOT Imagery. *International Journal of Environment and Geoinformatics*, *10*(1), pp.9-15. DOI: https://doi.org/10.30897/ijegeo.1196817.

[12] Sapitri, A.I., Nurmaini, S., Rachmatullah, M.N., Tutuko, B., Darmawahyuni, A., Firdaus, F., Rini, D.P. and Islami, A., 2023. Deep learning-based real time detection for cardiac objects with fetal ultrasound video. *Informatics in Medicine Unlocked*, *36*, p.101150. DOI: https://doi.org/10.1016/j.imu.2022.101150.

[13] Siddique, S., Islam, S., Neon, E.E., Sabbir, T., Naheen, I.T. and Khan, R., 2023. Deep learning-based bangla sign language detection with an edge device. *Intelligent Systems with Applications*, *18*, p.200224. DOI: https://doi.org/10.1016/j.iswa.2023.200224Ref.

[14] Cao, L., Zheng, X. and Fang, L., 2023. The semantic segmentation of standing tree images based on the Yolo V7 deep learning algorithm. *Electronics*, *12*(4), p.929. DOI: https://doi.org/10.3390/electronics12040929.

[15] Yudianto, M.R.A., Kusrini, K. and Al Fatta, H., 2020. Analisis Pengaruh Tingkat Akurasi Klasifikasi Citra Wayang dengan Algoritma Convolitional Neural Network. *(JurTI) Jurnal Teknologi Informasi*, *4*(2), pp.182-191. DOI: https://doi.org/10.36294/jurti.v4i2.1319.