# Implementasi YOLOv12 dan Slicing Inference untuk Meningkatkan Efisiensi Penghitungan Benih Udang

1st Habibi Wahyu Saputra
*Department of Computer Engineering*
*Telkom University*
Surabaya, Indonesia
habibiwahyusaputra@student.telkomuniversity.ac.id

2nd Susijanto Tri Rasmana
*Department of Computer Engineering*
*Telkom University*
Surabaya, Indonesia
susijanto@telkomuniversity.ac.id

3rd Eka Sari Oktarina
*Department of Computer Engineering*
*Telkom University*
Surabaya, Indonesia
ekasario@telkomuniversity.ac.id

***Abstrak* — Penghitungan benih udang pada tahap *post-larva* merupakan fase penting dalam manajemen budidaya untuk memastikan kepadatan tebar yang optimal. Namun, metode manual saat ini dinilai tidak efisien dan rentan kesalahan akibat karakteristik benih yang mikroskopis, tubuh semi-transparan, dan perilaku bergerombol. Penelitian ini bertujuan mengatasi permasalahan tersebut dengan mengembangkan sistem penghitungan otomatis berbasis *Deep Learning* menggunakan algoritma YOLOv12. Sistem diimplementasikan dalam arsitektur client-server, di mana model di-deploy sebagai layanan backend yang dapat diakses melalui API. Fokus utama penelitian adalah mengevaluasi efektivitas metode input dan konfigurasi hyperparameter pada dataset terbatas berjumlah 141 citra beresolusi tinggi. Hasil pengujian menunjukkan penerapan metode *Slicing Inference* memberikan peningkatan kinerja signifikan dibandingkan metode standard *resize*, dengan peningkatan *Mean Average Precision* (mAP@0.5) dari 40,8% menjadi 85,5%. Pada validasi data uji *black-box* hingga kepadatan 771 ekor, sistem mampu mencapai rata-rata akurasi penghitungan sebesar 83,70%. Dengan waktu inferensi rata-rata 11,53 detik per citra, sistem ini menawarkan keseimbangan antara presisi dan kecepatan sebagai solusi untuk digitalisasi industri akuakultur.**

***Kata kunci* — benih udang, deteksi objek, YOLOv12, slicing inference, penghitungan otomatis**

## I. PENDAHULUAN

Keberhasilan budidaya udang vaname (*Litopenaeus vannamei*) sangat bergantung pada manajemen presisi di fase pembenihan, di mana penghitungan benur (benih udang tahap *post-larva*) menjadi titik krusial untuk memastikan estimasi kepadatan tebar yang akurat dan mencegah kerugian ekonomi [1]. Namun, metode penghitungan manual yang umum digunakan saat ini dinilai tidak efisien secara waktu dan rentan terhadap kesalahan manusia. Tantangan utama dalam otomatisasi proses ini terletak pada karakteristik fisik benur yang berukuran mikroskopis, tubuh semi-transparan yang menyatu dengan air, serta perilaku bergerombol yang menyebabkan tumpang tindih objek [2].

Penerapan teknologi pengolahan citra digital berbasis *Deep Learning* telah terbukti mampu meningkatkan efisiensi penghitungan benih. Implementasi algoritma YOLOv5 pada perangkat *mobile* menunjukkan tingkat akurasi yang tinggi, namun pendekatan ini seringkali terkendala oleh kebutuhan dataset latih yang sangat besar untuk mencapai generalisasi yang baik [3]. Di sisi lain, pendekatan segmentasi regional yang memecah citra menjadi bagian-bagian lokal terbukti efektif menangani tantangan deteksi objek kecil pada kondisi kepadatan tinggi [2]. Hingga saat ini, belum ada studi yang mengeksplorasi potensi arsitektur terbaru YOLOv12 yang dilengkapi mekanisme *Area Attention* dan *Residual Efficient Layer Aggregation Network* (R-ELAN) untuk mengatasi permasalahan deteksi objek biologis yang samar [4].

Permasalahan teknis mendasar dalam deteksi objek mikro adalah hilangnya detail fitur visual saat citra resolusi tinggi mengalami kompresi ukuran (*resizing*) untuk menyesuaikan *input layer* model [5]. Untuk mengatasi degradasi informasi tersebut, metode *Slicing Inference* diterapkan dengan memproses citra dalam bentuk potongan-potongan (*slices*) yang saling tumpang tindih, sehingga resolusi asli objek tetap terjaga selama proses inferensi [5]. Integrasi metode ini dengan arsitektur *attention-centric* diharapkan mampu meningkatkan sensitivitas deteksi pada dataset yang terbatas.

Penelitian ini bertujuan mengimplementasikan dan mengevaluasi kinerja deteksi YOLOv12 yang dikombinasikan dengan teknik *Slicing Inference* untuk penghitungan benur udang otomatis. Sistem dibangun dengan arsitektur *client-server* untuk menangani beban komputasi visual secara terpusat, memungkinkan aksesibilitas yang fleksibel melalui API. Kontribusi utama penelitian ini terletak pada analisis komparatif efektivitas metode input serta validasi ketahanan model terhadap variasi kepadatan populasi dalam skenario pengujian *black-box*.

## II. KAJIAN TEORI

### A. YOLOv12: *Attention-Centric Object Detection*

YOLOv12 merupakan evolusi terbaru dari keluarga algoritma You Only Look Once yang memperkenalkan pendekatan attention-centric untuk menyeimbangkan efisiensi dan akurasi *[4]*. Berbeda dengan pendahulunya yang sangat bergantung pada Convolutional Neural Network (CNN) standar, YOLOv12 mengintegrasikan mekanisme Area Attention untuk memproses receptive field yang besar dengan biaya komputasi yang lebih rendah. Arsitektur ini juga menerapkan Residual Efficient Layer Aggregation Network (R-ELAN) pada backbone untuk mengatasi gradient bottlenecks dan meningkatkan penggunaan kembali fitur (feature reuse) . Inovasi arsitektur ini memungkinkan model mengekstrak fitur yang lebih kaya dari objek, yang sangat relevan untuk mendeteksi objek kecil dengan fitur visual terbatas seperti benih udang.

### B. Mekanisme *Slicing Inference*

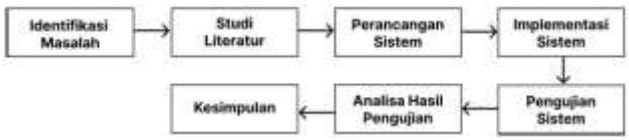
*Slicing Inference* merupakan teknik pra-pemrosesan yang diadopsi untuk mengatasi degradasi performa model deteksi objek pada citra resolusi tinggi. Pada pendekatan konvensional, citra input berdimensi besar dipaksa mengecil (*downsampling*) secara drastis untuk menyesuaikan lapisan input model, yang mengakibatkan hilangnya detail piksel pada objek mikroskopis [5]. Metode slicing bekerja dengan mempartisi citra asli menjadi matriks potongan (*slices*) $S = \{S_1, S_2, \dots, S_N\}$ dengan resolusi yang dipertahankan, sehingga fitur visual objek tetap tajam saat proses inferensi.

Komponen penting dalam metode ini adalah penerapan *Overlap Ratio*, yaitu area irisan antar potongan yang berfungsi untuk mencegah terpotongnya objek yang berada di garis batas potongan. Tanpa mekanisme *overlap*, objek di tepian akan terdeteksi parsial atau hilang karena fitur visualnya tidak utuh. Setiap potongan diproses secara independen oleh model, menghasilkan himpunan deteksi lokal yang kemudian dipetakan kembali ke sistem koordinat citra asli. Untuk menangani redundansi deteksi yang muncul di area *overlap*, algoritma pasca-pemrosesan seperti *Non-Maximum Suppression* (NMS) diterapkan guna menyeleksi *bounding box* dengan skor terbaik dan mengeliminasi duplikasi, memastikan jumlah akhir objek merepresentasikan populasi sebenarnya secara akurat.

## III. METODE

### A. Desain Sistem

Penelitian ini mengembangkan sistem penghitungan benih udang otomatis berbasis *client-server*. Alur penyelesaian masalah dimulai dari identifikasi kendala penghitungan manual, perancangan sistem, pelatihan, hingga pengujian sistem. Tahapan lengkap penelitian digambarkan pada Gambar 1.




*GAMBAR 1*
*(ALUR PENYELESAIAN MASALAH)*

### B. Pengumpulan Dataset

Sumber data primer diperoleh melalui pengambilan citra langsung benih udang vaname (Litopenaeus vannamei) fase post-larva. Pengambilan data dilakukan dalam lingkungan terkontrol menggunakan wadah kontainer putih semi-transparan dengan kedalaman air 1,5 cm. Citra diakuisisi menggunakan kamera smartphone dengan resolusi tinggi 4000x3000 pixel. Total dataset berjumlah 141 citra, yang dibagi menjadi 99 citra latih (70%), 28 citra validasi (20%), dan 14 citra uji (10%) untuk mengevaluasi generalisasi model.

### C. Implementasi Algoritma Slicing Inference

Metode *Standard Resize* menyebabkan hilangnya detail fitur visual pada objek mikroskopis karena downsampling resolusi yang ekstrem. Untuk mengatasi hal tersebut, penelitian ini menerapkan algoritma *Slicing Inference* yang bekerja melalui tiga tahapan sistematis:

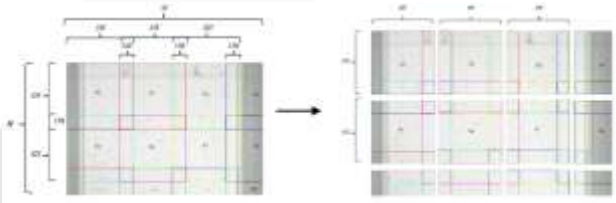
*i. Slicing Citra*

Citra asli berukuran $W$ x $H$ tidak diproses secara utuh, melainkan dipotong menjadi beberapa potongan (slices) $s_1, s_2, \dots, s_N$ dengan dimensi tetap $SW$ x $SH$. Jumlah potongan $N$ ditentukan secara matematis untuk menjamin cakupan area menyeluruh menggunakan Persamaan (1).

$$\mathrm{N} = \left(1 + \left\lceil \frac{W - SW}{Stride_W} \right\rceil\right) \times \left(1 + \left\lceil \frac{H - SH}{Stride_H} \right\rceil\right) \qquad (1)$$

Dimana nilai pergeseran atau *stride* horizontal ($Stride_W$) dan vertikal ($Stride_H$) dihitung berdasarkan *Overlap Ratio* ($OR$) menggunakan Persamaan (2):

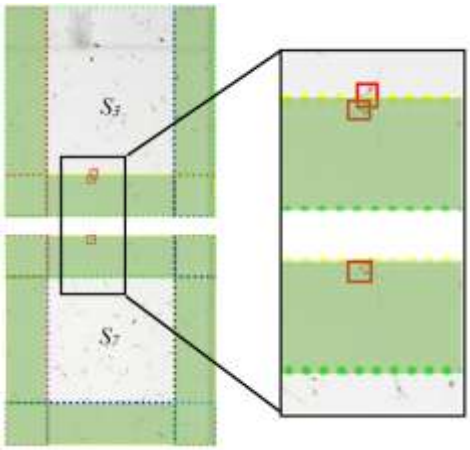
$$Stride_W = SW \times (1 - OR) \qquad (2)$$
$$Stride_H = SH \times (1 - OR)$$



*GAMBAR 2*
*(ILUSTRASI MEKANISME SLICING CITRA DENGAN PENERAPAN OVERLAP)*

ii. Mekanisme Overlap dan Inferensi Lokal

Penerapan *Overlap Ratio* sangat penting untuk mencegah terjadinya objek terpotong pada area perbatasan antar *slice*. Tanpa area tumpang tindih, benur yang berada di tepian *slice* akan kehilangan sebagian fitur visualnya dan gagal dideteksi oleh model. Dengan adanya *overlap*, objek tersebut akan muncul secara utuh setidaknya pada satu *slice* tetangga. Setiap potongan kemudian diproses secara independen oleh model YOLOv12 untuk menghasilkan prediksi koordinat lokal.

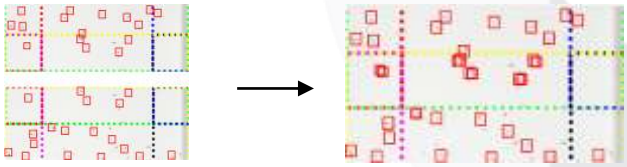



*GAMBAR 3*
*(ILUSTRASI PENERAPAN OVERLAP RATIO)*

*GAMBAR 4*
*(ILUSTRASI PROSES INFRENESI PADA TIAP POTONGAN CITRA)*

iii. Rekonstruksi dan *Post-Processing*

Hasil deteksi dari seluruh potongan dipetakan kembali ke sistem koordinat citra asli. Konsekuensi dari penerapan *overlap* adalah munculnya redundansi, di mana satu objek benur dapat terdeteksi berulang kali pada *slice* yang berbeda. Oleh karena itu, diterapkan algoritma *Non-Maximum Suppression* (NMS) pada tahap akhir. NMS berfungsi menyeleksi *bounding box* dengan skor kepercayaan tertinggi dan mengeliminasi kotak lain yang memiliki nilai *Intersection over Union* (IoU) tinggi di area yang sama, sehingga memastikan setiap benur hanya dihitung satu kali.



*GAMBAR 5*
*(ILUSTRASI PEMETAAN KEMBALI HASIL DETEKSI)*

*GAMBAR 6*
*(ILUSTRASI NMS)*

D. Arsitektur dan Konfigurasi Model

Sistem menggunakan arsitektur YOLOv12 varian *medium* yang diinisialisasi dengan bobot *random*. Proses pelatihan dilakukan menggunakan *platform cloud* dengan konfigurasi *hyperparameter* hasil optimasi: *batch size* 16, *learning rate* 0,001, dan *optimizer* SGD. Pelatihan model dibatasi dengan mekanisme *Early Stopping* (*patience* 40 *epoch*) untuk mencegah *overfitting* dan memastikan efisiensi waktu komputasi.

E. Kebutuhan Penelitian

Seluruh proses pelatihan model dilakukan menggunakan *platform cloud computing* dengan dukungan GPU akselerator untuk mempercepat komputasi. Adapun pengambilan dataset dilakukan menggunakan kamera utama *smartphone* Xiaomi Redmi Note 9 dengan resolusi 48MP (f/1.8) dan sensor 1/2.0" untuk memastikan detail objek mikroskopis benur tertangkap dengan tajam.

F. Skenario Pengujian

*i. Uji Model*

Mengevaluasi kinerja deteksi menggunakan metrik *Mean Average Precision* (mAP@0.5), *Precision*, *Recall*, dan *F1-Score*. Pada tahap ini, dilakukan perbandingan performa antara metode *Standard Resize* dan *Slicing Inference*.

*ii. Uji Sistem*

Menguji fungsionalitas API *backend* menggunakan data baru dengan variasi kepadatan benur sejumlah 50 - 771 ekor dan kondisi pencahayaan yang berbeda untuk mengukur ketahanan sistem.

## IV. HASIL DAN PEMBAHASAN

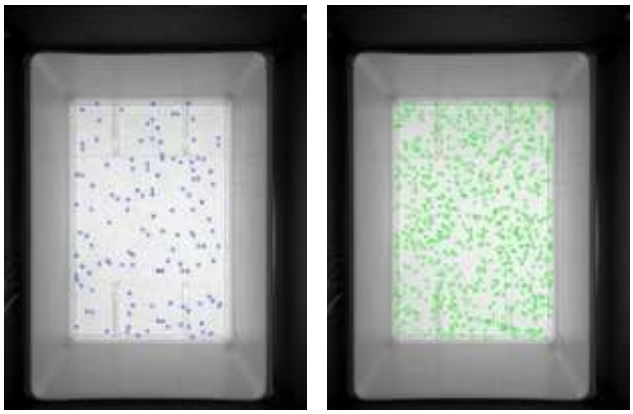
A. Analisis Perbandingan Metode Input

Evaluasi utama penelitian ini membandingkan kinerja dua metode *pre-processing* yaitu *Standard Resize* dan *Slicing Inference*. Hasil pengujian pada Tabel 1 menunjukkan kesenjangan performa yang signifikan. Metode *Standard Resize* gagal mendeteksi sebagian besar objek benur, dengan capaian *mAP*@0.5 hanya 40,8% dan *Recall* 35,4%. Hal ini disebabkan oleh distorsi ekstrem saat citra resolusi tinggi sebesar 4000x3000 pixel diperkecil menjadi 640x640 pixel, yang menghilangkan detail visual benur.

Sebaliknya, metode *Slicing Inference* terbukti lebih baik dengan *mAP*@0.5 mencapai 85,5% dan *Recall* 83,5%1. Mekanisme pemotongan citra berhasil mempertahankan resolusi asli, sehingga fitur morfologi benur tetap tajam. Meskipun waktu inferensi meningkat menjadi 11,53 detik per citra akibat proses penggabungan hasil deteksi, *trade-off* ini sepadan dengan lonjakan akurasi yang dihasilkan. Perbedaan visual hasil deteksi kedua metode dapat dilihat pada Gambar 7.

*TABEL 1*
*(Perbandingan kinerja metode Resize vs Slicing)*

| Metode | Precision | Recall | F1-Score | mAP@0.5 | Rata2 Waktu Inferensi |
|---|---|---|---|---|---|
| Resize | 63.5% | 35.4% | 45.5% | 40.8% | 0.76 detik |
| Slicing | 86.5% | 83.5% | 85.0% | 85.5% | 11.53 detik |



(a) (b)

*GAMBAR 7*
*(Perbandingan visual hasil deteksi: (a) metode resize gagal mendeteksi mayoritas objek; (b) metode slicing inference berhasil mendeteksi benur dengan akurat)*

B. Optimasi Hyperparameter

Untuk memaksimalkan kinerja model YOLOv12, dilakukan *tuning* bertahap pada *Batch Size* dan *Learning Rate*.

Penentuan *Batch Size* dilakukan dengan menguji variasi nilai 8, 16, dan 32 untuk mencari keseimbangan antara stabilitas gradien dan efisiensi waktu. Tabel 2 menunjukkan perbandingan kinerja ketiga konfigurasi tersebut.

*TABEL 2*
*(Perbandingan pengaruh konfigurasi batch size)*

| Batch Size | Precision | Recall | F1-Score | mAP@0.5 | Durasi *Training* |
|---|---|---|---|---|---|
| 8 | 85.9% | 83.2% | 84.5% | 85.0% | 11280 detik |
| 16 | 86.5% | 83.5% | 85.0% | 85.5% | 10500 detik |
| 32 | 86.0% | 83.0% | 84.5% | 84.8% | 10260 detik |
| 64 | Tidak berjalan karena *Out Of Memory* | | | | |

Hasil menunjukkan bahwa *batch size* 8 membutuhkan waktu pelatihan terlama (11.280 detik) akibat frekuensi pembaruan bobot yang terlalu sering. Sebaliknya, pada *batch size* 32 terjadi sedikit penurunan performa deteksi menjadi 84,8% *mAP*@0.5. Batch size 16 memberikan hasil optimal dengan *mAP* tertinggi (85,5%) dan durasi pelatihan yang lebih efisien (10.500 detik), sehingga dipilih sebagai parameter tetap.

Selanjutnya, pengujian *Learning Rate* (LR) dilakukan pada nilai 0.01, 0.001, dan 0.0001. Sebagaimana ditampilkan pada Tabel 3, LR 0.001 menghasilkan performa terbaik dengan *loss* validasi terendah.

*TABEL 3*
*(Perbandingan konfigurasi learning rate)*

| *Learning Rate* | *Precision* | *Recall* | *F1-Score* | mAP@0.5 | *Loss* |
|---|---|---|---|---|---|
| 0.01 | 82.4% | 88.1% | 85.1% | 84.1% | 2.35 |
| 0.001 | 86.5% | 83.5% | 85.0% | 85.5% | 2.09 |
| 0.0001 | 86.1% | 83.5% | 84.8% | 85.4% | 2.07 |

Penggunaan LR besar (0.01) menyebabkan fenomena *overshooting*, yang diindikasikan oleh tingginya nilai *loss* validasi mencapai 2,35 serta ketidakstabilan yang teramati selama proses pelatihan. Sebaliknya, penggunaan LR 0.001 menunjukkan penurunan nilai *loss* yang signifikan dan konsisten, menandakan bahwa model mampu mencapai titik konvergensi secara optimal tanpa terjebak dalam fluktuasi yang berlebihan.

*C. Cross Validation*

Validasi K-Fold Untuk memastikan model tidak mengalami *overfitting* pada pembagian data tertentu, diterapkan metode *5-Fold Cross Validation*. Hasil pada Tabel 4 menunjukkan rentang deviasi performa yang kecil, dengan selisih mAP terendah dan tertinggi hanya berkisar 3,4%. Konsistensi ini mengindikasikan bahwa arsitektur YOLOv12 dengan *Slicing Inference* memiliki stabilitas yang baik terhadap variasi input data.

*TABEL 4*
*(Hasil pengujian cross validoation)*

| *Fold* | *mAP@0.5* | *Precision* | *Recall* | *F1-Score* | Waktu training |
|---|---|---|---|---|---|
| 1 | 80.22% | 83.46% | 80.23% | 81.81% | 30600 detik |
| 2 | 81.06% | 84.04% | 78.92% | 78.92% | |
| 3 | 82.84% | 84.03% | 82.26% | 83.14% | |
| 4 | 83.48% | 84.30% | 82.57% | 83.43% | |
| 5 | 83.65% | 84.27% | 81.73% | 82.98% | |
| Rata-rata | 82.25% | 84.02% | 81.14% | 82.55% | |

D. Pengujian Sistem

Pengujian tahap akhir dilakukan untuk memvalidasi ketahanan sistem backend terhadap variasi kepadatan populasi menggunakan data uji baru. Pengujian melibatkan rentang kepadatan mulai dari 50 hingga 771 ekor per citra.

Hasil pada Tabel 2 menunjukkan bahwa sistem mampu mempertahankan akurasi rata-rata sebesar 83,70%. Meskipun terdapat sedikit penurunan performa pada kepadatan tertinggi yaitu 771 ekor, sistem tetap mampu mendeteksi lebih dari 500 objek dengan tepat. Hal ini membuktikan bahwa metode *Slicing Inference* efektif mengatasi masalah oklusi (tumpang tindih) pada populasi padat, menjadikan sistem ini layak diimplementasikan sebagai alat bantu hitung di lingkungan budidaya nyata.

*TABEL 5*
*(Hasil uji validasi system pada variasi kepadatan ebnur)*

| No | Jumlah Manual | Rata-rata Jumlah Prediksi | Rata-rata Selisih | Rata-rata Akurasi | Rata-rata Waktu Respon |
|---|---|---|---|---|---|
| 1 | 50 | 45.52 | 4.48 | 91.04% | 31.98 detik |
| 2 | 100 | 84.47 | 15.52 | 84.47% | 34.64 detik |
| 3 | 150 | 126.30 | 23.69 | 84.20% | 33.35 detik |
| 4 | 200 | 159.71 | 40.28 | 79.85% | 34.84 detik |
| 5 | 250 | 199.10 | 50.90 | 79.64% | 32.58 detik |
| 6 | 300 | 251.75 | 48.25 | 83.91% | 36.22 detik |
| 7 | 350 | 299.62 | 50.37 | 85.60% | 32.96 detik |
| 8 | 400 | 332.75 | 67.25 | 83.18% | 33.55 detik |
| 9 | 500 | 408.80 | 91.20 | 81.76% | 32.70 detik |
| 10 | 771 | 642.66 | 128.33 | 83.35% | 34.98 detik |
| Rata-rata | | | | 83.70% | 38.47 detik |

## V. KESIMPULAN

Penelitian ini berhasil mengimplementasikan sistem penghitungan benih udang otomatis berbasis *Deep Learning* menggunakan arsitektur YOLOv12 yang terintegrasi dalam layanan *backend*. Berdasarkan hasil evaluasi, penerapan metode *Slicing Inference* terbukti menjadi solusi paling efektif untuk mengatasi tantangan deteksi objek mikroskopis, dengan peningkatan signifikan pada *Mean Average Precision* (mAP@0.5) dari 40,8% (metode *resize*) menjadi 85,5%.

Konfigurasi pelatihan optimal dicapai menggunakan model YOLOv12m dengan *Batch Size* 16 dan *Learning Rate* 0,001, yang memberikan keseimbangan terbaik antara stabilitas gradien dan kecepatan konvergensi. Pada pengujian validasi sistem dengan variasi kepadatan hingga 771 ekor per citra, sistem mampu mempertahankan rata-rata akurasi sebesar 83,70%. Meskipun metode ini membutuhkan waktu inferensi rata-rata 11,53 detik per citra yang lebih lama dibandingkan metode standar, tingkat presisi yang dihasilkan menjadikannya solusi yang valid dan andal untuk menggantikan proses penghitungan manual di industri akuakultur.

Pengembangan selanjutnya disarankan untuk meningkatkan kuantitas dataset hingga 500-1000 citra dengan variasi kondisi lingkungan guna meningkatkan ketahanan model. Selain itu, eksplorasi implementasi langsung pada perangkat menggunakan *framework* seperti TensorFlow Lite sangat direkomendasikan untuk menghilangkan latensi jaringan di area tambak yang minim sinyal.

## REFERENSI


[1] S. Arsad *et al.*, “The Application of Microalgae Feeding Regime on Whiteleg Shrimp Culture in Each Stage: A Mini Review,” *Sains Malays.*, vol. 52, no. 1, pp. 1–16, Jan. 2023, doi: 10.17576/jsm-2023-5201-01.

[2] H. Duan *et al.*, “Shrimp Larvae Counting Based on Improved YOLOv5 Model with Regional Segmentation,” *Sensors*, vol. 24, no. 19, p. 6328, Sep. 2024, doi: 10.3390/s24196328.

[3] S. Y. Asmak, D. D. Rizaldi, R. A. F. Saputra, A. P. Abseno, V. R. Hananto, and E. S. Oktarina, “A Mobile App for Counting Shrimp Larvae Based on the YOLO V5 Method,” *Journal of Computer Electronic and Telecommunication*, vol. 5, no. 2, Dec. 2024, doi: 10.52435/complete.v5i2.647.

[4] Y. Tian, Q. Ye, and D. Doermann, “YOLOv12: Attention-Centric Real-Time Object Detectors,” Feb. 2025, doi: https://doi.org/10.48550/arXiv.2502.12524.

[5] F. C. Akyon, S. Onur Altinuc, and A. Temizel, “Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection,” in *2022 IEEE International Conference on Image Processing (ICIP)*, IEEE, Oct. 2022, pp. 966–970. doi: 10.1109/ICIP46576.2022.9897990.