

Alat Penghitung Benih Ikan Berbasis Kecerdasan Buatan dan Database

Eko Rudiawan Jamzuri¹, Fajar Rahmatullah²¹

¹Politeknik Negeri Batam, Batam, Indonesia

*Email: eko.rudiawan@gmail.com

Abstract— Konsumsi ikan yang tinggi di Indonesia mendorong pertumbuhan usaha budidaya ikan, di mana pembenihan ikan menjadi langkah awal yang sangat penting. Namun, metode manual dalam menghitung benih ikan sering kali tidak efisien dan memakan waktu. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan sistem penghitung benih ikan otomatis yang memanfaatkan teknologi computer vision berbasis kecerdasan buatan. Penelitian ini adalah kelanjutan dari kegiatan *Project-Based Learning (PBL)* di Politeknik Negeri Batam, dengan fokus pada perbaikan akurasi dan kualitas model deteksi. Dalam penelitian ini, metode *object detection* diterapkan menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur *Single Shot MultiBox Detector (SSD) MobileDet*. Sistem ini dirancang untuk mendeteksi benih ikan dalam gambar dengan memanfaatkan pemrosesan citra yang memperhitungkan pengaruh warna air dan pencahayaan. Evaluasi model menunjukkan peningkatan performa yang signifikan dalam parameter yang tersedia melalui *tensorboard* seperti *Mean Average Precision (mAP)*, *Learning Rate*, dan *Total Loss* selama proses pelatihan. Pengujian dilakukan dalam wadah transparan berwarna putih dengan air jernih untuk meningkatkan kualitas deteksi. Meskipun model yang dilatih hanya dengan menggunakan laptop dengan bantuan *CPU*, penelitian ini menunjukkan hasil yang cukup baik. Namun terdapat penurunan kemampuan deteksi ikan pada jarak 40-50 cm dengan skala rata-rata deteksi antara 3 dari 10. Faktor-faktor yang mempengaruhi deteksi termasuk batasan dataset yang terbatas pada jarak pelatihan, kondisi pencahayaan, dan keterbatasan sumber daya pelatihan. Penelitian ini menyimpulkan bahwa untuk mengoptimalkan deteksi benih ikan dengan *computer vision*, diperlukan peningkatan variasi dan kualitas dataset serta penggunaan sumber daya pelatihan yang lebih memadai.

Keyword: *Computer Vision, Object Detection, Database, Ikan.*

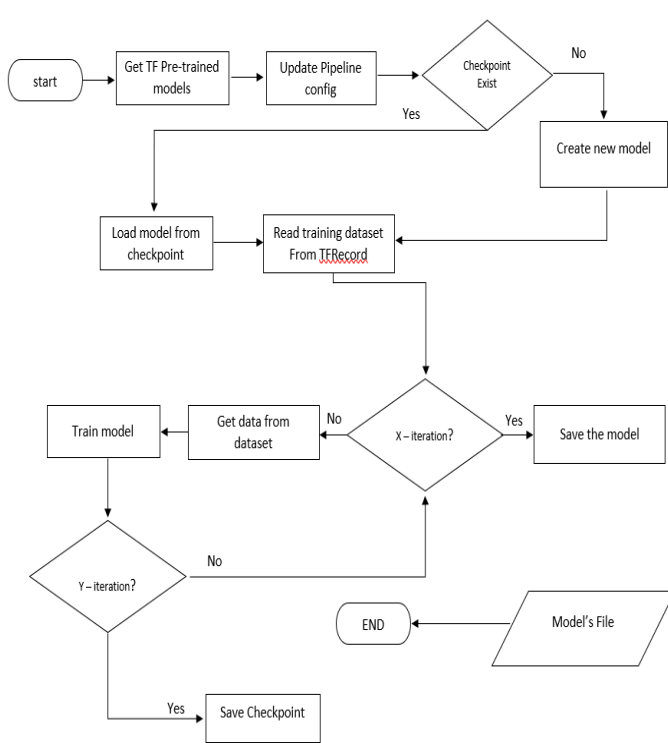
1. INTRODUCTION

Tingginya konsumsi ikan di Indonesia menjadikan usaha budidaya ikan sebagai sektor yang menjanjikan dan strategis untuk pertumbuhan ekonomi dan ketahanan pangan. Dalam budidaya ikan, pembenihan ikan adalah tahap awal yang sangat penting, karena menentukan kualitas dan kuantitas benih yang akan dibudidayakan. Pembenihan ikan melibatkan berbagai proses, termasuk pemilihan induk, pembuahan, dan pemeliharaan benih. Salah satu langkah dalam proses ini adalah penghitungan benih ikan, yang sering kali dilakukan secara manual dengan menghitung benih satu per satu. Metode manual ini tidak hanya memerlukan waktu yang lama tetapi juga berisiko terhadap kesalahan manusia dan tidak efisien, terutama dalam skala besar. Seiring dengan perkembangan teknologi, terdapat kebutuhan mendesak untuk mengimplementasikan solusi yang lebih efisien dan akurat dalam penghitungan benih ikan. Untuk itu, teknologi *computer vision* dan kecerdasan buatan (*artificial intelligence*) menawarkan solusi yang menjanjikan. Sistem penghitung benih ikan otomatis yang menggunakan teknologi ini dapat mempercepat proses penghitungan, mengurangi kesalahan, dan meningkatkan akurasi hasil penghitungan. Penelitian ini merupakan kelanjutan dari kegiatan *Project-Based Learning (PBL)* yang dilakukan di Politeknik Negeri Batam, di mana penelitian sebelumnya menghadapi kendala terkait akurasi dan kualitas model deteksi. Penelitian ini bertujuan untuk mengatasi kendala tersebut dengan meningkatkan kualitas model deteksi dan akurasi penghitungan benih ikan. Dalam penelitian ini, metode penghitungan benih ikan otomatis diterapkan menggunakan teknologi *computer vision* dengan teknik *object detection* berbasis *Convolutional Neural Network (CNN)* dan arsitektur *Single Shot MultiBox Detector (SSD) MobileDet*. Teknologi *computer vision* memungkinkan sistem untuk menganalisis citra dan mendeteksi benih ikan dengan memanfaatkan fitur visual seperti bentuk dan ukuran. Arsitektur *SSD MobileDet* dipilih karena kemampuannya dalam melakukan deteksi objek secara *real-time* dengan akurasi tinggi dan efisiensi komputasi yang baik. Metode ini melibatkan beberapa langkah utama, mulai dari pengambilan citra benih ikan dalam wadah,

pemrosesan citra untuk identifikasi objek, hingga penghitungan jumlah benih ikan yang terdeteksi. Sistem ini dirancang untuk bekerja dalam berbagai kondisi, dengan mempertimbangkan faktor-faktor lingkungan seperti warna air dan pencahayaan. Penggunaan wadah transparan berwarna putih dan air jernih diharapkan dapat meningkatkan kualitas citra dan visibilitas benih ikan, sehingga mempermudah proses deteksi. Pengembangan teknologi ini tidak hanya bertujuan untuk meningkatkan efisiensi dan akurasi dalam penghitungan benih ikan, tetapi juga untuk memberikan solusi yang lebih praktis. Dengan menggunakan teknologi *computer vision*, sistem penghitung benih ikan otomatis ini diharapkan dapat meningkatkan produktivitas dan mempercepat proses pembenihan ikan.

2. METHOD

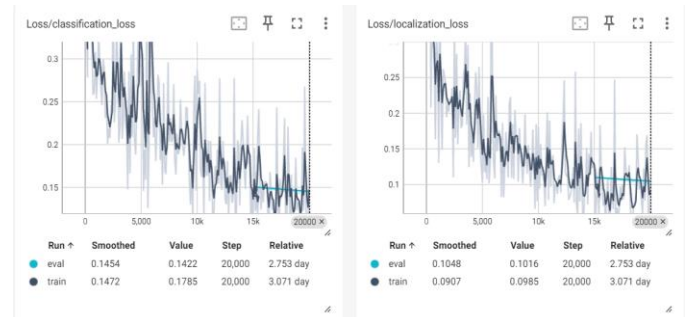
a. Perancangan Diagram Blok Sistem



Gambar 1. Flowchart Training Model

Pada penelitian ini proses pelatihan data dimulai dengan menyiapkan model pra-terlatih *Tensorflow* yang disediakan dan dapat diunduh di *website* resmi *Tensorflow*, kemudian dilanjutkan dengan mengatur *pipeline config* yang akan digunakan untuk mengatur alur kerja dan konfigurasi yang dibutuhkan dalam proses pelatihan. Selanjutnya *Checkpoint* digunakan untuk memulai pelatihan dari kondisi tertentu, baik dari awal pelatihan maupun dari model yang telah dilatih sebelumnya. Kemudian sistem akan menginisiasi untuk membaca dataset yang telah dikumpulkan dalam format *TFRecord*. Kemudian proses pelatihan dilanjutkan dengan mengecek apakah jumlah iterasi (X) yang telah ditentukan sudah tercapai dimana pada penelitian ini jumlah iterasi yang ditentukan adalah sebanyak 20000 iterasi. Jika jumlah iterasi

(X) sudah tercapai, proses pelatihan selesai dan model akan disimpan sebagai file model yang final. Namun jika jumlah iterasi (X) belum tercapai, sistem akan mulai melakukan kembali proses pelatihan menggunakan data yang diambil dari dataset. Ketika jumlah iterasi keseluruhan (Y) yang telah ditentukan sudah tercapai, sistem akan menyimpan *checkpoint* dari model yang sedang dilatih untuk dapat digunakan Kembali jika pelatihan dihentikan atau dilanjutkan. Dalam melakukan pelatihan terdapat parameter pendukung yang digunakan sebagai acuan dalam menilai kualitas model, seperti *Mean Average Precision(mAP)*, *Loss* dan *Learning Rate*.

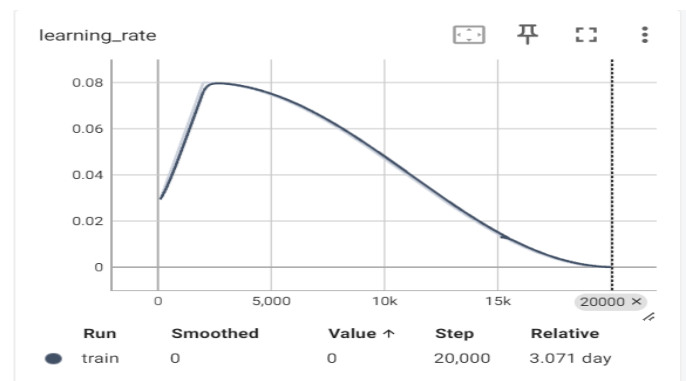


Gambar 2. Grafik Loss Hasil Penelitian

Loss adalah metrik yang digunakan untuk mengukur seberapa baik atau buruk prediksi model. Dalam hal ini semakin kecil nilai *loss* maka semakin baik pula performa model [6]. Terdapat berbagai jenis *loss* yang dapat digunakan, oleh karena pengujian ini adalah deteksi objek, *loss* yang digunakan adalah *classification loss*, *localization loss*, dan *regularization loss*.

1. *Classification loss* : berfungsi untuk mengukur seberapa akurat model dalam mengklasifikasikan objek ke dalam kategori yang benar sesuai dataset.
2. *Localization loss* : berfungsi untuk mengukur seberapa akurat model dalam menentukan lokasi objek di dalam gambar.

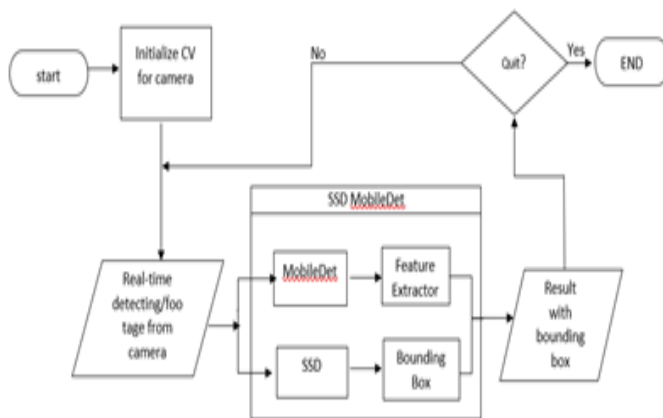
Secara keseluruhan grafik menunjukkan penurunan nilai *loss* yang menandakan performa model dalam masa pelatihan semakin meningkat.



Gambar 3. Grafik Learning Rate Hasil Penelitian

Learning rate adalah parameter yang digunakan untuk mengontrol ukuran langkah (*step*) yang diambil saat melakukan pembaruan bobot model berdasarkan gradien loss. Pada grafik terlihat sumbu X menunjukkan langkah pelatihan dan sumbu Y menunjukkan nilai dari *learning rate*. Pada awal pelatihan, *learning rate* meningkat dari nilai awalnya hingga mencapai puncak di langkah ke-5000 dengan nilai 0,08. Setelah mencapai puncak, *learning rate* pelatihan model ini menurun secara bertahap hingga akhir di langkah ke-20000 dan mendekati nilai 0. Peningkatan *learning rate* pada awal pelatihan bertujuan untuk mempercepat proses pembelajaran pelatihan di awal. Kemudian setelah mencapai puncak, *learning rate* turun secara bertahap yang bertujuan agar model dapat melakukan penyesuaian dengan lebih halus dan tepat serta menghindari *overshooting*, yang dapat menyebabkan fluktuasi dalam masa pelatihan.

b. Flowchart Metode Deteksi



Gambar 2. Flowchart Metode Deteksi

Pada penelitian ini metode yang digunakan adalah *Tensorflow* dengan menggunakan *SSD MobileDet* sebagai model pelatihan. Proses deteksi dimulai dengan menginisiasi *Computer Vision (CV)* dengan melibatkan penggunaan Pustaka *OpenCV* dari *Python* untuk mengakses dan mengatur kamera yang akan digunakan untuk pengambilan video secara *real-time*. Selanjutnya input gambar akan di transfer ke jaringan *CNN*. Kemudian jaringan akan menghasilkan beberapa *Feature maps* dengan resolusi yang berbeda. Setiap *Feature map cell* memiliki *anchor boxes* dengan berbagai aspek rasio dan skala. Untuk setiap *anchor boxes*, *SSD* memprediksi pergeseran posisi (*offsets*) dan skala untuk menyesuaikan box dengan objek yang terdeteksi serta *confidence score* untuk setiap kelas objek. Selanjutnya *Feature Extractor* akan menghasilkan *Feature Maps* yang berisi informasi yang diperlukan untuk mendeteksi objek. *SSD* akan menggunakan *Feature Maps* ini untuk membuat *Bounding Boxes*. Kemudian hasil deteksi akan ditampilkan dengan *Bounding Boxes* yang menunjukkan lokasi benih ikan dalam frame.

c. Perancangan Software

Dalam Penelitian ini penulis menggunakan *software PyQt5* dan *QtDesigner* beserta *Library* yang tersedia untuk mendesain awal tampilan antarmuka atau *Graphic User Interface (GUI)*.



Gambar 4. Rancangan awal GUI

GUI ini berfungsi sebagai jembatan antara pengguna dan sistem, membuat teknologi ini menjadi lebih mudah diakses dan digunakan. Hal ini membantu pengguna dalam mengoperasikan alat dengan lebih efisien, mengurangi kompleksitas, dan memastikan bahwa semua data yang diperlukan disajikan dengan cara yang mudah dipahami.

d. Perancangan Mekanikal

Dalam melakukan perancangan mekanikal penelitian ini menggunakan material *aluminium profile* yang dapat diatur ke atas dan kesamping untuk memastikan penempatan dan stabilitas kamera yang optimal dalam pengambilan deteksi gambar atau video benih ikan. Pemanfaatan mekanikal ini dimulai dengan menentukan jarak ukur antara kamera dengan wadah yang berisi benih ikan.

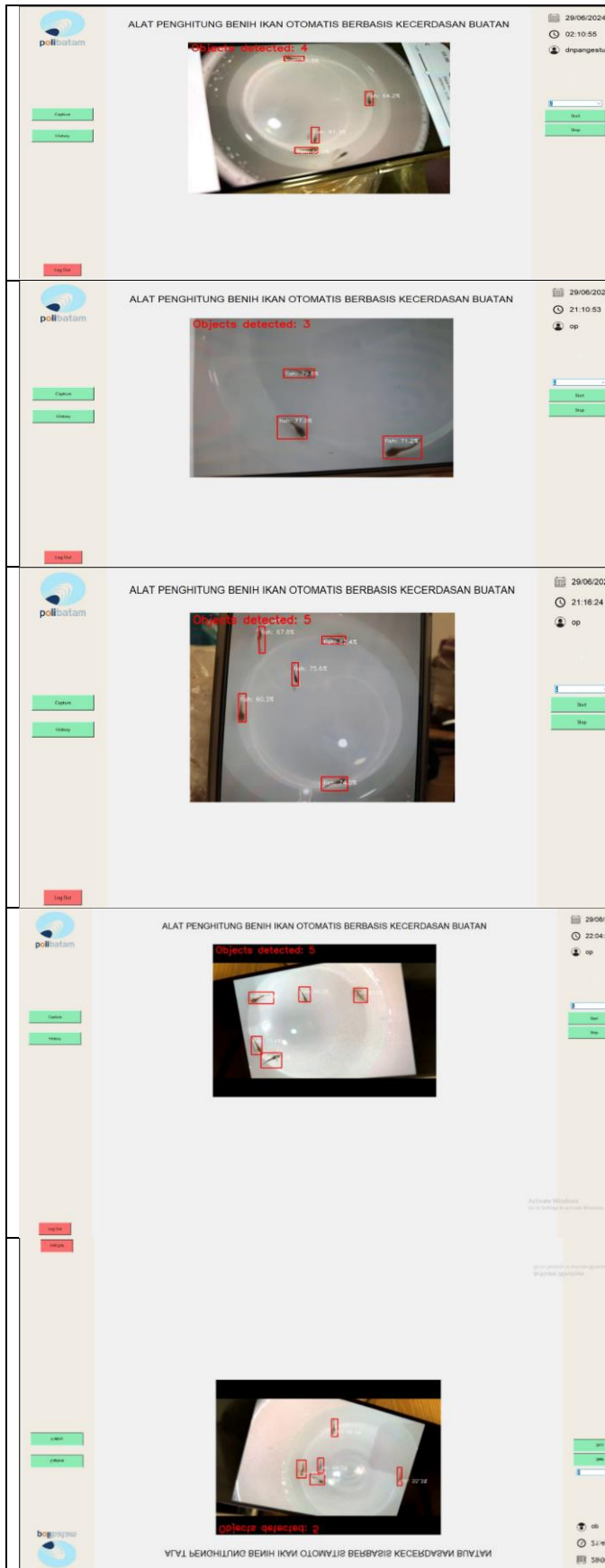


Gambar 5. Tampak atas kamera ke wadah

3. HASIL DAN PEMBAHASAN

Hasil dan pembahasan berisi data yang harus diketahui dalam penelitian ini. Pengambilan data memiliki tujuan untuk membuktikan bahwa penelitian ini sesuai dengan tujuan yang diharapkan. Berikut merupakan data hasil pengujian yang telah dilakukan berdasarkan model *tensorflow object detection* dan dataset sebanyak 7000 sampel gambar dengan jumlah step iterasi sebanyak 20000. Metode ini diimplementasikan menggunakan CPU dengan mendeteksi benih ikan yang berada dalam suatu wadah yang di deteksi dari sisi atas wadah menggunakan *web-camera*.

a. Hasil Deteksi 5 ekor benih dengan interval jarak 25-45cm

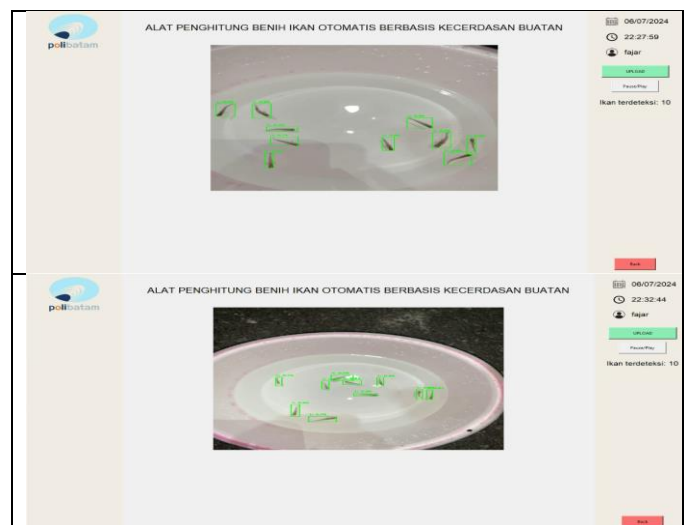


Tabel 1. Hasil Deteksi 5 Ekor Benih Ikan

Pembahasan dari hasil deteksi 5 benih ekor ikan :

1. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 25cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 2 benih ikan terdeteksi.
2. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 25cm menunjukkan sedikit perbedaan hasil antara jumlah ikan aktual dan uji web cam dimana sebanyak 3 benih ikan terdeteksi.
3. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 30cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 5 benih ikan terdeteksi.
4. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 35cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 5 benih ikan terdeteksi.
5. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 45cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 5 benih ikan terdeteksi.

b. Hasil Deteksi 10 Benih Ikan Dengan Interval 25-50cm





Tabel 2. Hasil Deteksi 10 Benih Ekor Ikan

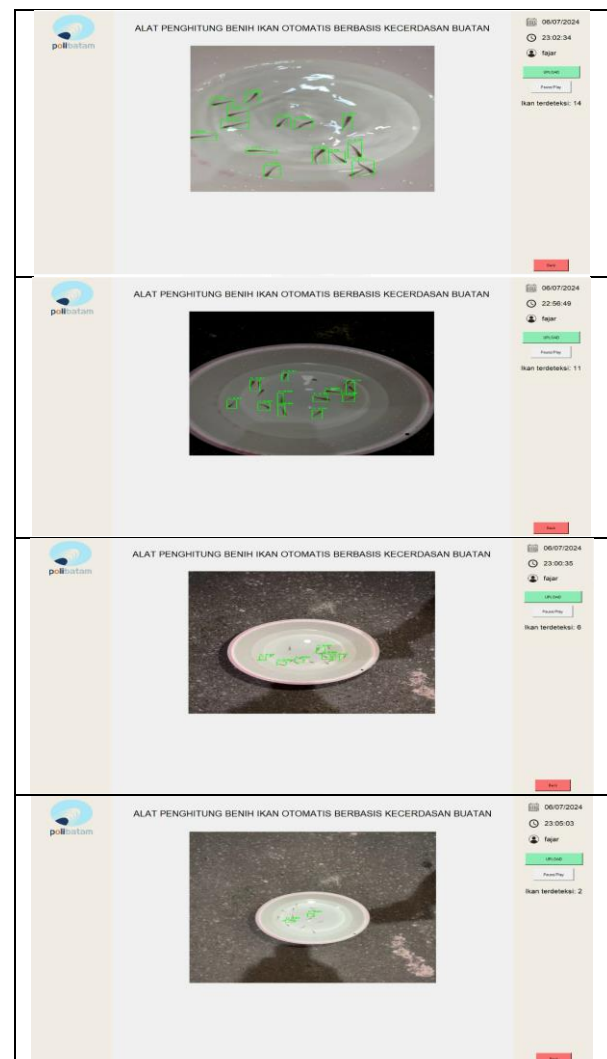
Pembahasan Hasil Deteksi 10 Ekor Benih Ikan :

1. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 20cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 10 ekor benih ikan terdeteksi.
2. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 25cm menunjukkan sedikit perbedaan hasil antara jumlah ikan aktual dan uji web cam dimana sebanyak 10 ekor benih ikan terdeteksi.
3. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 30cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 10 ekor benih ikan terdeteksi.
4. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model

Tensorflow. Dari hasil uji deteksi diatas dengan jarak uji sejauh 35cm menunjukkan hasil yang sesuai antara jumlah ikan aktual dan uji web cam dimana sebanyak 10 ekor benih ikan terdeteksi.

5. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 40cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 10 ekor benih ikan mengalami penurunan akurasi sehingga hanya dapat mendeteksi sebanyak 5 ekor benih ikan.
6. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 50cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 10 ekor benih ikan mengalami penurunan akurasi sehingga hanya dapat mendeteksi sebanyak 5 ekor benih ikan.

c. Hasil Deteksi 15 Benih Ikan Dengan Interval 15-50cm



Tabel 3. Hasil Deteksi 15 Benih Ekor Ikan

Hasil Pembahasan Deteksi 15 Ekor Benih Ikan :

1. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 20cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 15 ekor benih ikan mendeteksi sebanyak 11 ekor benih ikan.
2. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 25cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 15 ekor benih ikan mendeteksi sebanyak 14 ekor benih ikan.
3. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 40cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 15 ekor benih ikan mendeteksi sebanyak 6 ekor benih ikan.
4. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 50cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 15 ekor benih ikan mendeteksi sebanyak 2 ekor benih ikan

d. Hasil Deteksi 20 Benih Ikan Dengan Interval 20-50cm



Tabel 4. Hasil Deteksi 20 Benih Ekor Ikan

Hasil Pembahasan Deteksi 20 Ekor Benih Ikan :

1. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 20cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 19 ekor benih ikan.
2. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 25cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 18 ekor benih ikan.
3. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 30cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 16 ekor benih ikan.
4. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 35cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 12 ekor benih ikan.
5. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 40cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 7 ekor benih ikan.

6. Pengujian pengambilan data ini dilakukan secara *real-time* dengan memperhatikan gerakan ikan selama 1 menit untuk menguji akurasi kamera yang telah dilatih dengan model *Tensorflow*. Dari hasil uji deteksi diatas dengan jarak uji sejauh 50cm menunjukkan bahwa hasil deteksi yang dapat dicapai antara jarak dengan jumlah ikan sebanyak 20 ekor benih ikan mendeteksi sebanyak 5 ekor benih ikan

4. KESIMPULAN DAN SARAN

a. Kesimpulan

Berdasarkan hasil penelitian ini, implementasi teknologi *computer vision* dalam penghitungan benih ikan terbukti dapat mengatasi kendala yang sering muncul dalam metode penghitungan manual. Dalam penelitian ini, pelatihan model dilakukan menggunakan laptop dengan CPU, namun berhasil menghasilkan data latih yang cukup baik. Evaluasi menggunakan *Tensorboard* menunjukkan peningkatan performa model selama masa pelatihan, dengan parameter seperti *Mean Average Precision (mAP)*, *Learning Rate*, dan *total Loss* yang menunjukkan adanya perbaikan yang signifikan dari awal hingga akhir pelatihan. Selain itu, untuk meningkatkan kualitas deteksi, pengujian harus dilakukan dalam wadah transparan berwarna putih yang diisi dengan air jernih. Penelitian ini juga mengidentifikasi beberapa faktor yang mempengaruhi kemampuan deteksi kamera, terutama pada jarak 40-50cm. Hasil uji menunjukkan bahwa deteksi ikan mengalami penurunan kemampuan deteksi dengan skala rata – rata antara 3-4 dari 10. Beberapa faktor yang mempengaruhi meliputi batasan dataset yang dilatih dengan jarak tidak lebih dari 50cm di setiap variasi jumlah ikan, kondisi pencahayaan yang dapat mempengaruhi visibilitas objek oleh kamera, serta keterbatasan sumber daya membatasi pelatihan model hanya dengan menggunakan laptop dengan CPU. Penelitian ini menyiratkan bahwa untuk mengoptimalkan deteksi benih ikan menggunakan *computer vision*, diperlukan peningkatan dalam variasi dan kualitas dataset dan sumber daya yang memadai untuk pelatihan model.

b. Saran

Agar penelitian yang dilakukan dapat ditingkatkan atau dikembangkan lebih baik kedepannya, dalam hal ini penulis memiliki beberapa saran sebagai berikut :

1. Dalam melakukan pelatihan dataset latih sebaiknya menggunakan *device* yang dapat mengkombinasikan kinerja CPU dan GPU agar menghasilkan data hasil uji latih yang lebih akurat dan baik.
2. Dalam melakukan pengumpulan dataset sebaiknya berikan lebih banyak variasi gambar objek agar sistem dalam melakukan deteksi dengan lebih baik.
3. Pengujian ini terbukti dapat mendeteksi benih ikan dalam kondisi pencahayaan yang cukup baik. Oleh karena itu, hindari melakukan pengujian deteksi di daerah yang minim pencahayaan.
4. Cobalah untuk memahami setiap parameter dalam pelatihan model machine learning, seperti yang ditampilkan pada grafik-grafik di atas. Hal ini juga berfungsi untuk meningkatkan performa model, membantu proses

pengembangan dan menghasilkan model akhir yang lebih baik.

REFERENSI

- [1] L. R.T, "FishCountingIEEE," *Automated fish fry counting and schooling behavior analysis using computer vision*, pp. 2-5, 2017.
- [2] K. Sivaraman and K. Azaraffali, "Network Failure Detection and Diagnosis by Analyzing Syslog and SNS Data: Applying Big Data Analysis to Network Operations," *IEEE access*, pp. 883-887, 2019.
- [3] G. P. L. K. K. M. A. L. Chen Liang-Chieh, "Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units," pp. 9-11, 2020.
- [4] Y. Ç. Aktaş, "Toward Data Science," 31 January 2022. [Online]. Available: <https://towardsdatascience.com/object-detection-with-convolutional-neural-networks-c9d729eecd18>.
- [5] A. A.F, "Estimasi Citra Kedalaman (Depth Estimation) Menggunakan Kamera Monocular," *Tugas Akhir*, pp. 9-10, 2020.
- [6] M. Corrales and D. Kadowaki, "TensorBoard: Exploring Different Types of Graphs for Deep Learning Models," *Tensorboard*, 2019.
- [7] "Unsupervised Monocular Depth Estimation with Left-Right Consistency," *Unsupervised Monocular Depth Estimation with Left-Right Consistency*, pp. 270-279, 2017.
- [8] B. G.J, "UCL_Unsupervised Monocular Depth Estimation with Left-Right Consistency," *UCL Visual Computing*, pp. 270-279, 2017.
- [9] K. S and S. S, "Manual vs Automatic Counting of Fish Eggs dan Larvae: Implications for Stock Assessment," *Aquaculture International*, 2020.
- [10] J. S. Jane Doe, "The Importance of Data Preparation in Machine Learning: Understanding the Data Pipeline," *Journal of Machine Learning Research*, 2021.